

SZÉCHENYI ISTVÁN EGYETEM
MŰSZAKI TUDOMÁNYI KAR
JEDLIK ÁNYOS GÉPÉSZ, INFORMATIKAI ÉS VILLAMOSMÉRNÖKI
INTÉZET
MATEMATIKA ÉS SZÁMÍTÁSTUDOMÁNY TANSZÉK



MŰSZAKI INFORMATIKA SZAK
Szoftverfejlesztő szakirány

DIPLOMAMUNKA

Alkalmazásfejlesztés cakePHP keretrendszerrel

Barkóczy Roland
mérnök-informatikus

2007

Feladat:

Webes áruház kialakítása MVC tervezési minta alapján, cakePHP keretrendszer felhasználásával.

Részfeladatok:**a) A szakdolgozat elkészítéséhez szükséges résztvevőképesítések:**

A kijelölt szakirodalom (MVC tervezési minta, cakePHP keretrendszer) megismerése, és a feladathoz szükséges módszerek tanulmányozása. Hasonló keretrendszerek bemutatása. Az alkalmazás megtervezése, kivitelezése és tesztelése, felhasználói dokumentáció készítése.

b) A szakdolgozat főbb részei:

Elméleti összefoglaló az MVC tervezési mintáról. PHP alapú keretrendszerek áttekintése. A CakePHP bemutatása. Alkalmazás tervezése: adatbázis tervezése, funkciók specifikálása. Alkalmazás implementálása. Felhasználói dokumentáció.

Belső konzulens: Fehérvári Arnold SZE GIVI Matematika és Számítástudomány Tanszék

Külső konzulens: Nagy Gusztáv tanszéki mérnök, KEFO GAMF Informatikai Tanszék

A diplomamunka benyújtásának határideje: 2007. május 31.

Győr, 2007. május 25.

.....

tanszékvezető

.....

szakfelelős

.....

szakirányfelelős

A diplomamunkát ellenőriztem: *beadható* – *nem adható be*

dátum

belső konzulens aláírása

A diplomamunkát ellenőriztem: *beadható* – *nem adható be*

dátum

külső konzulens aláírása

Bíráló adatai:

.....

.....

A diplomamunka minősítése

A bíráló javaslata:

érdemjegy

dátum

bíráló aláírása

A Tanszék javaslata:

érdemjegy

dátum

tanszékvezető aláírása

Az ÁVB határozata:

érdemjegy

dátum

ÁVB-elnök aláírása

BARKÓCZI ROLAND

ALKALMAZÁSFEJLESZTÉS CAKEPHP KERETRENDSZERREL

ÖSSZEFOGLALÁS

Manapság a statikus, brosúraszerű honlapok mellett egyre nagyobb az igény a Web alapú alkalmazások készítésére, mint például:

- elektronikus kereskedelemmel kapcsolatos feladatok megoldása egyedi igényekhez igazítva,
- ügyfelekkel való on-line kapcsolattartás (Customer Relationship Management),
- ügyviteli rendszerek kialakítása,
- tartalomkezelő rendszerek.

Az ilyen nagy, adatbázis-alapú rendszerek gyors, hibamentes fejlesztéséhez és karbantartásához nélkülözhetetlen egy jó keretrendszer és objektum-orientált programtervezési minta.

Szakdolgozatom témájául egy webes kereskedelmi alkalmazás megvalósítását választottam. A programban lehetőség van a vállalkozás termékskálájának rögzítésére és megjelenítésére, a vevők adatainak regisztrációjára, a termékek megrendelésére és véleményezésére, valamint a rendelések nyomon követésére. A bolt adminisztrátorai karbantarthatják a termékek adatbázisát és kezelhetik a rendeléseket.

A dolgozat elején szeretném bemutatni a Model–Nézet–Vezérlő (Model-View-Controller) tervezési mintát és annak előnyeit, majd egy rövid áttekintést olvashatunk a különböző, nyílt forráskódú, PHP alapú keretrendszerekről. Ezután a választott keretrendszer, a CakePHP kerül bemutatásra. Röviden olvashatunk a rendszer telepítéséről és felépítéséről. A következő fejezetben az alkalmazás fejlesztési dokumentációját olvashatjuk, kiemelve a legfontosabb funkciókat és az érdekesebb megoldásokat. A dolgozatot a felhasználói dokumentáció zárja.

Szakdolgozatomat és a forráskódok tanulmányozását ajánlom mindazoknak, akiket érdekel a webes fejlesztés PHP nyelven, és szeretnének gyorsan és hatékonyan alkalmazásokat fejleszteni.

ROLAND BARKÓCZI

APPLICATION DEVELOPMENT WITH CAKEPHP FRAMEWORK

SUMMARY

Nowadays, among the static, brochure-like websites, there are increasing demands for the web based applications, like:

- customized solutions for electronic commerce,
- customer relationship management,
- management software's
- content management systems.

A well-organized framework and an object oriented program design pattern are indispensable for the quick, fail-safe development of these kinds of expansive, DB-based applications.

The subject of my thesis is the development of a web-based e-commerce system. It is possible to record and display the data of the product-range. The customers can sign-up and order the products, write comments about them and follow-up their orders. The web shop's administrators can update the data of products and maintain the orders.

In the first part of the thesis I would like to introduce the Model-View-Controller design pattern and its advantages, then a short description can be read about the different, open-source, PHP-based frameworks. After, the chosen framework, the CakePHP will be introduced. We can read briefly about the installation and structure of the system. The next chapter contains the development documentation. The most important functions and the more interesting solutions are highlighted. Lastly the user documentation can be found.

I recommend studying my thesis and the source codes for the programmers who are interested in the PHP based, quick and powerful web-development.

TARTALOMJEGYZÉK

1. BEVEZETÉS	5
2. MVC TERVEZÉSI MINTA	6
3. PHP KERETRENDSZEREK.....	8
3.1. Prado	8
3.2. Seagull	8
3.3. Zephyr	9
3.4. Symfony.....	10
4. CAKEPHP.....	11
4.1. Követelmények	11
4.2. Telepítés.....	11
4.3. Beállítások	12
4.4. Gyorsváz	13
4.5. Modellek	13
4.5.1. Kapcsolatok.....	14
4.5.2. Az Appmodel osztály metódusai	15
4.6. Vezérlők.....	16
4.6.1. Metódusok.....	17
4.6.2. Változók	19
4.7. Nézetek	19
4.8. Helperek és komponensek	20
5. ALKALMAZÁSFEJLESZTÉS: WEBES ÁRUHÁZ.....	21
5.1. Funkciók	21
5.2. Adatmodell	22
5.3. Fejlesztői környezet	24
5.4. Alkalmazás felépítése	24
5.4.1. Felhasználók	24

TARTALOMJEGYZÉK

5.4.2. Termékek	26
5.4.3. Termékfotók.....	28
5.4.4. Termékek véleményezése	29
5.4.5. Kategóriák.....	29
5.4.6. A kosár	30
5.4.7. Rendelések	30
5.4.8. Kódtáblák karbantartása.....	31
5.4.9. Menüszerkezet és jogosultságok.....	32
5.4.10. Kliensoldali programozás	32
6. FELHASZNÁLÓI DOKUMENTÁCIÓ	33
6.1. Rendszerkövetelmények	33
6.2. Publikus felület.....	33
6.2.1. Felhasználói felület	33
6.2.2. Termékek tallózása	34
6.2.3. Kosár	35
6.2.4. Regisztráció.....	35
6.2.5. Vásárlás.....	35
6.2.6. Saját rendelések ellenőrzése.....	36
6.3. Adminisztrációs felület.....	36
6.3.1. Kategóriák.....	36
6.3.2. Termékek	37
6.3.3. Termékfotók.....	38
6.3.4. Rendelések	38
6.3.5. Áfakulcsok, szállítási és fizetési módok	39
6.3.6. Dokumentumok.....	39
7. ÖSSZEFOGLALÁS	40
8. IRODALOMJEGYZÉK	41
9. MELLÉKLET.....	42
9.1. Vezérlők forrásállományai.....	42

TARTALOMJEGYZÉK

9.1.1. categories_controller.php	42
9.1.2. Delivery_methods_controller.php.....	44
9.1.3. orders_controller.php	45
9.1.4. pages_controller.php	50
9.1.5. payment_methods_controller.....	51
9.1.6. product_comments_controller.php	53
9.1.7. product_photos_controller.php	53
9.1.8. products_controller.php	56
9.1.9. Shopping_carts_controller.php	60
9.1.10. taxes_controller.php.....	62
9.1.11. Users_controller.php	63
9.2. Modellek.....	66
9.2.1. category.php	66
9.2.2. delivery_method.php.....	68
9.2.3. order.php	68
9.2.4. order_comments.php.....	70
9.2.5. ordered_product.php	70
9.2.6. page.php	70
9.2.7. payment_method.php.....	71
9.2.8. product.php	71
9.2.9. product_comment.php	72
9.2.10. product_photo.php	73
9.2.11. shopping_cart.php.....	73
9.2.12. tax.php.....	73
9.2.13. user.php.....	73
9.3. Nézetek	75
9.3.1. category.....	75
9.3.2. delivery_method.....	77
9.3.3. elements	78
9.3.4. layout.....	80

TARTALOMJEGYZÉK

9.3.5. order	83
9.3.6. page	91
9.3.7. Payment_method	92
9.3.8. admin_add.html	92
9.3.9. product	93
9.3.10. product_photo	102
9.3.11. shopping_cart	106
9.3.12. tax	108
9.3.13. user	110

1. BEVEZETÉS

Gyakorló PHP fejlesztőként gyakran ütköztem abba a problémába, hogy nagyon hasonló, de bonyolult feladatokat kellett szinte nulláról kezdve felépíteni. A monolit, strukturális kód használatának hátrányai hamar megmutatkoznak, amikor egy komolyabb alkalmazást kell elkészíteni, dokumentálni és karbantartani. Az alkalmazás üzleti logikája és a kliens oldali (leggyakrabban html és JavaScript) kód összekeveredik, a program növekedésével a hibák megjelenése és azok elhárítására fordított idő exponenciálisan növekszik. Nem is beszélve a későbbi bővítések nehézségéről.

Felismerve ezt, egy megfelelő módszert kerestem a gyors, hatékony, hibamentes és skálázható PHP nyelvű alkalmazások fejlesztéséhez. Először egy saját magam által írt osztályokból álló kódot használtam, amely nagyrészt kiiktatta a hibalehetőségek melegágyát jelentő SQL lekérdezések írását. Azonban rábukkantam az MVC programozási mintára és vele együtt a CakePHP keretrendszerre, melynek alapjai nagyon hasonlóak voltak az én elgondolásomhoz, csak éppen profi módon voltak kivitelezve. Ekkor a saját fejlesztés folytatása helyett a CakePHP használatának megismerésére fordítottam az időmet, melynek az előnyei már a legelején jelentkeztek: ezek közül a legfontosabb, hogy a nemzetközi és magyar CakePHP felhasználók nagyon segítőkészek, így az első lépések nem is voltak olyan nehezek, és a gyorsan jelentkező sikerek további tanulásra inspiráltak.

2007 elején hozták létre a magyar CakePHP oldalt, ahol nagy lelkesedéssel kezdődött az angol nyelvű dokumentáció fordítása. Ekkor fogalmazódott meg bennem az a gondolat, hogy a szakdolgozatom témájaként egy CakePHP segítségével fejlesztett alkalmazás írását válasszam. Próbáltam egy nem túl egyszerű, de nem is túl komplex témát választani, amelyet később továbbfejlesztve akár kereskedelmi forgalomba is lehetne hozni. Így esett a választásom a manapság egyre népszerűbb e-kereskedelem egy lehetséges, web alapú implementációjára.

2. MVC TERVEZÉSI MINTA

Az MVC (model–view–controller, modell–nézet–vezérlő) egy jól használható, kipróbált módszer arra, hogy hogyan válasszuk szét a felhasználói felületet és az üzleti logikát. Az elsődleges cél az, hogy a felhasználói felület megjelenítéséért felelős kódot teljesen elkülönítsük. Ezáltal annak módosítása, kiegészítése nem vonja maga után az üzleti logikát megtestesítő kód módosítását vagy megismétlését.

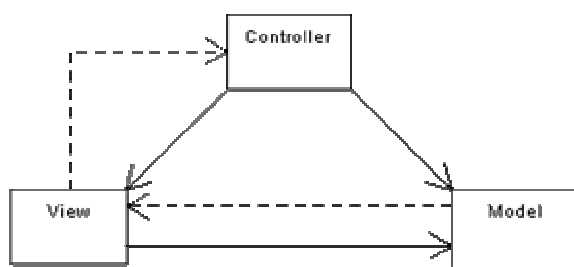
A módszer lényege az, hogy a hagyományos eljárás alapú programok *adatbevitel*, *adatfeldolgozás*, *eredmény megjelenítése* feladatait leképezzék a grafikus felhasználói felülettel rendelkező programokban:

adatbevitel → *adatfeldolgozás* → *eredmény megjelenítése*
Controller (Vezérlő) → *Model (Modell)* → *View (Nézet)*

A **Controller** dolgozza fel a felhasználói adatbevitelt. Parancsokká és függvényhívássökká képezi le azokat. Ezek a parancsok fogják előidézni az adatok módosítását, törlését, vagy a nézetek megváltozását. Például, ha a felhasználó rákattint a menü egyik elemére, akkor egy controller fogja meghatározni, hogy ennek hatására mi is történjen.

A **Model** reprezentálja az üzleti logikát, feladata értelmet adni a nyers adatoknak. Ez az egység felelős egy számla áfatartalmának és végösszegének kiszámolásáért. A model tudja, hogy melyik vevő fogja kifizetni a számlát, és ha szükséges, azt is, hogy éppen születésnapja van-e ennek a vevőnek.

A **View** feladata a felhasználói felület megjelenítése. Ez formokat, táblázatokat, linkeket, gombokat, szövegeket jelent. Ezek az elemek megjelenhetnek egy szabványos HTML-oldalon, azonban ha egy Java nyelvű alkalmazásról van szó, akkor az AWT vagy a Swing könyvtárra épülő grafikus felületet jelent.



A Model, View és Controller kapcsolatai.

A folytonos vonal a közvetlen, a szaggatott pedig az indirekt kapcsolatokat mutatja [8]

Vegyük egy konkrét gyakorlati PHP fejlesztői példát. Feladatunk egy felmérés eredményeinek megjelenítése táblázatban.

Hagyományos módszerrel a PHP-forrásunk tartalmaz egy SQL-lekérdezést, majd az adatokat egy ciklussal kiírja egy táblázatba. Ha az ügyfelünk ugyanezen adatokat egy grafikonon is látni szeretné, akkor létrehozunk egy másik fájlt, ami szintén tartalmazni fogja az SQL-kérést, és az adatok megjelenítését.

Az MVC tervezési minta ezzel szemben tartalmazni fog egy modellt, ami felelős az adatbázis kezeléséért, és két nézetet (views), ami felelős az adatok megjelenítéséért. Azt, hogy melyik nézetet kell megjeleníteni, a controller dönti el a felhasználó kívánsága szerint. Ha további nézetekre van szükségünk (pl. kördiagram vagy esetleg egy szöveges magyarázatokkal ellátott kimutatás), akkor csak a megfelelő nézetfájlokat kell elkészíteni, amelyek a modelltől származó adatokat használják fel.

Ezt a tervezési mintát Trygve Reenskaug írta le 1979-ben, amikor grafikus felületű SmallTalk alkalmazásokon dolgozott a Xeroxnál. A SmallTalk MVC nagy sikert aratott, rengeteg grafikus keretrendszerben került felhasználásra. Néhány példa:

- Microsoft Foundation Classes (MFC)
- Java Swing
- QuickTime Toolkit
- Business Server Pages (SAP)

MVC keretrendszerek különböző nyelveken:

- Ruby: **Ruby on Rails**
- Python: **Django**
- Perl: **Catalyst**
- Coldfusion: **Fusebox**

Sok PHP nyelvű keretrendszer létezik, a legfontosabbakat a következő fejezetben fogjuk górcső alá venni.

3. PHP KERETRENDSZEREK

Jó néhány MVC-t megvalósító keretrendszer fejlesztésének kezdtek neki a lelkes programozó csapatok – mindenki a saját elképzelése szerint, különböző létező rendszerek által inspirálva. Vannak, akik a Microsoft .NET környezetet vették alapul, mások a Ruby on Rails-t (<http://www.rubyonrails.org/>) próbálták PHP-ben implementálni, megint mások Java Struts nyomdokain haladtak. Rengeteg kezdeményezést van, lássunk néhány jó keretrendszert, amelyek fejlesztése jelenleg is folyik, és támogatottsága széles körű.

3.1. Prado

Ez a keretrendszer tulajdonképpen az ASP.NET megvalósítása PHP-ben. A lelkes fejlesztőcsapatnak köszönhetően már a 3.0-s verziót tölthetjük le, sőt elérhető a 3.1-es alfa változat is. A fejlesztést a PRADOSOFT végzi, Qiang Xue vezetésével.

Fő jellemzői:

- Szükséges PHP verzió: 5.1
- Eseményvezérelt programozás
- Gazdag HTML komponens készlet: formelemek, varázslók, táblázatok stb.
- Támogatja az i18n-t és a L10n-t
- AJAX-támogatás
- Webszolgáltatások kezelése
- Testre szabható hibaüzenetek és kivételkezelés
- Biztonsági intézkedések: védelem XSS támadások ellen, sütivédelem stb.
- Honlap: <http://www.pradosoft.com/>

3.2. Seagull

PEAR (PHP Extension and Application Repository) csomagokra épülő keretrendszer. A londoni székhelyű Seagull System (<http://seagullsystems.com>) nevű cég fejleszti. Legfrissebb stabil változat a 0.6.1-es.

Jellemzők:

- Szükséges PHP verzió: 4.2-től
- Szoros kapcsolat a PEAR könyvtárakkal

- Beépített egyszerű tartalomkezelő rendszer és egyéb modulok:
 - Felhasználók karbantartása
 - RSS
 - Hírlevélszerkesztő
 - Belső üzenetkezelő modul
 - Menüszerkezet-kezelő
 - Beépített WYSIWYG szerkesztők
- Könnyen beilleszthetők a külső fejlesztők által készített programok
- Kiforrott, robusztus rendszer (fejlesztése 2001-ben kezdődött)
- XHTML és CSS 2.1 támogatás
- Támogatott adatbázisok: MySQL, PostgreSQL, MS SQL Server, Oracle, MaxDB, DB2
- Honlap: <http://www.seagullproject.org>

3.3. Zephyr

Smarty-ra és AdoDB-re épülő keretrendszer Indiából. A dokumentáció bevezetője a rendszer problémamentes telepítését, könnyű tanulását és alkalmazások, modulok gyors fejlesztését ígéri. Legfrissebb változata a 2.0 beta számot viseli. A fejlesztői csoport magját három indiai fiatalember alkotja, vezetőjük Hasin Hazder (<http://hasin.wordpress.com/>)

Jellemzői:

- Szükséges PHP verzió: 5.1-től
- Smarty (<http://smarty.php.net/>) sablonmotorra épülő architektúra
- AdoDB (<http://adodb.sourceforge.net/>) használata az adatkapcsolati rétegben
- Cpaint (<http://cpaint.wiley14.com/>) használata az AJAX-funkciók megvalósításához
- Prototype könyvtár (<http://www.prototypejs.org/>) a front-end rétegben
- Cron támogatás: az ismétlődően és rendszeresen végrehajtandó feladatok futtatásának ütemezése
- PEAR csomagok integrálásának támogatása
- Honlap: <http://zephyr-php.sourceforge.net/>

3.4. Symfony

A francia Sensio (<http://www.sensio.com>) cég által fejlesztett Ruby on Rails szerű keretrendszer. Fabien Potencier indította a projektet még 2003-ban, az első hivatalos változat 2005-ben jelent meg. Jelenleg egy nagy létszámú fejlesztői közösség dolgozik a keretrendszer újabb verzióján.

Jellemzői:

- Szükséges PHP verzió: 5.1-től
- Adatbázis-független
- Könnyű telepíthetőség
- PEAR támogatása
- Beépített tesztelő, hibakereső és dokumentációkészítő eszközök
- AJAX-implementáció
- Skálázható rendszer, külső fejlesztők moduljai könnyen integrálhatóak
- Honlap: <http://www.symfony-project.com/>

4. CAKEPHP

Ebben a fejezetben röviden bemutatom a cakePHP telepítését és fő funkcióit, jellemzőit.

4.1. Követelmények

A cakePHP futtatásához a következő szoftverkörnyezetre van szükség:

- Webszerver (például Apache)
- PHP 4.3.2 vagy újabb verzió
- Adatbázisszerver: mySQL vagy PostgreSQL

4.2. Telepítés

A keretrendszer legfrissebb változata a <http://cakeforge.org/projects/cakephp/> webcímen érhető el. A szakdolgozat készítésének időpontjában (2007. március) ez a 1.1.13.4450 verziószámot viselte.

Töltsük le a <http://cakephp.org/downloads> címről a nekünk tetsző változatot, majd a zip fájl tartalmát másoljuk a webszerverre.

A cakePHP könyvtárszerkezete nagyon jól szervezett, de emellett rugalmasan alakítható. A letöltött tömörített mappa tartalma a következő:

```
/app  
/cake  
/docs  
/vendors  
.htaccess  
index.php
```

Az /app könyvtár tartalmazza az alkalmazásunk elemeit, csak ennek a könyvtárnak a fájljait kell módosítanunk. A /cake könyvtárban találjuk a rendszerfájlokat, gyakorlatilag sosem kell módosítanunk a tartalmát. Ennek köszönhetően a cakePHP frissítésekor nem kell aggódnunk amiatt, hogy a saját munkánkat véletlenül felülírjuk. A /vendors könyvtárba helyezhetjük el az alkalmazásunkban használni kívánt, külső fejlesztők által készített osztályokat (például phpMailer.class.php). A /docs könyvtár néhány txt dokumentumot tartalmaz, szükségtelen felmásolni a szerverünkre. Az /app/tmp könyvtárra mindenképpen adjunk írási jogot, mivel a rendszer a működése során létrejövő átmeneti fájlokat itt tárolja. A webhelyünkhöz tartozó gyökérkönyvtárat (DocumentRoot) állítsuk az /app/webroot mapára.

A cakePHP természetesen támogatja a szép webcímek használatát. Ez azt jelenti, hogy a webcímeink felépítése a következőképpen nézhetnek ki: <http://domain.hu/controller/action/p/a/r/a/m/s>, szemben a hagyományos URL formátummal, ahol egy webcím valahogy így néz ki: <http://domain.hu/index.php?controller=controller&action=action¶m1=param...>

A szép webcímeknek több előnyük is van: egyrészt könnyebben megjegyezhetőek, rövidebbek és a keresők is jobb találati helyezést adnak honlapunknak, ha már az URL is tartalmazza a kulcsszót.

Emiatt engedélyezzük a `mod_rewrite` modul használatát, amennyiben apache webszervert használunk.

Van mód ettől eltérő könyvtárstruktúra létrehozására is. Bővebb információ a <http://manual.cakephp.org/chapter/installing> webcímen érhető el.

4.3. Beállítások

A cakePHP beállítása roppant egyszerű: mindössze az adatbázis-kapcsolatot kell megadnunk, illetve néhány konfigurációs fájl tartalmát kell módosítanunk.

Az adatbázis-kapcsolat beállítását az `/app/config/database.php` fájl tartalmazza. Első telepítéskor ez a fájl még nem létezik, a `database.php.default` fájl átnevezésével hozhatjuk létre. Ebben a fájlban van deklarálva a `DATABASE_CONFIG` osztály, melynek publikus változói tartalmazzák az adatbázis-kapcsolat létrehozásához szükséges adatokat: meghajtó, kapcsolat típusa, host, felhasználónév, jelszó, adatbázisnév, valamint a táblaelőtagok, amennyiben vannak. Ez utóbbinak a célja az, hogy egy adatbázisban ugyanannak az alkalmazásnak több változatát lehessen üzemeltetni. Az itt meghatározott előtag hozzá lesz illesztve a táblanevekhez minden lekérdezésben.

Lehetőség van több adatbázis-kapcsolat egyidejű megadására. A `database.php.default` fájlban van egy `$test` változó is deklarálva, ha például egy fejlesztői adatbázist kívánunk használni az éles rendszer helyett. Ebben az esetben, a különböző modellekben a `var $useDbConfig = 'test';` változó inicializálásával áttérhetünk a `$test` változóban leírt adatbázis-kapcsolatra.

Az adatbázis tervezésénél ajánlott betartani az alábbi konvenciókat:

- A táblanevekhez angol nyelvű főneveket használjunk többes számban. Például: **users, articles**.
- Minden táblának legyen egy „id” nevű elsődleges kulcsa.
- Az idegen kulcsok neve legyen angol főnév egyes számban, utaljon a kapcsolódó tábla nevére. Ezután legyen egy aláhúzás, majd az „id” szó. Például: `user_id`
- Amennyiben a tábla tartalmaz „created” vagy „modified” nevű mezőket, a cakePHP automatikusan be fogja állítani az értéküket értelemszerűen: a record

létrehozásakor a „created” mezőt, update kérés esetén pedig a „modified” mezőt.

A **core.php** fájlban további globális konstansokat és változókat állíthatunk be, amelyek az egész rendszer működését meghatározzák. Ez egyes változók funkciói és lehetséges értékei a fájlban lévő megjegyzésekben részletesen le vannak írva, itt csak a legfontosabbat szeretném kiemelni:

- **DEBUG:** a hibakövetéshez szükséges információk monitorozására szolgál. Négy lehetséges értéke van:
 - 0: nincsenek debug információk
 - 1: a debug() és a pr() függvények hívásainak eredménye megjelenik a kimenetben
 - 2: az előző, plusz a kimenetben megjelennek az SQL-kérések és azok eredményei.
 - 3: az előző, plusz az adott modellhez tartozó objektum változói és függvényeinek listája.

Részletes leírás a <http://manual.cakephp.org/chapter/configuration> oldalon található.

4.4. Gyorsváz

Az alkalmazások fejlesztésének kezdeti fázisában, amikor még esetleg az adatbázis-struktúra is képlékeny, gyakran szükséges az egyes táblákhoz tartozó négy alap adatbázisművelet (rekord létrehozása, módosítása, megjelenítése és törlése) elvégzése. A cakePHP gyorsváz (scaffolding) képességével ezek a funkciók hozzáadása egy vezérlőhöz mindössze egyetlen változó deklarációjával elérhető.

Az adott vezérlőhöz a **var \$scaffold;** sor hozzáadásával kapunk egy listát az adott adattábláról: megjelenik az összes rekord és mező, valamint a rekordhoz tartozó funkciók linkjei: megjelenítés, módosítás és törlés. Rekord hozzáadásához és módosításához szükséges űrlap is automatikusan létrejön.

További információ a <http://manual.cakephp.org/chapter/scaffolding> oldalon található.

4.5. Modellek

Mint ahogy szó volt erről a második fejezetben, a modellekben valósítjuk meg az alkalmazásunk logikáját. A modellek felelősek az adatok kezeléséért, illetve itt implementálhatjuk az egyes üzleti funkciókat.

A modellek mindig az AppModel osztály leszármazottai. Az osztály nevét nagybetűvel kezdjük és általában a kapcsolódó tábla nevének egyes számú alakja. Például: az **User** mo-

dell az **users** táblának adataiért felelős. A modellek az `/app/models/` könyvtárban találhatóak. A fájl nevének meg kell egyeznie a modell nevének kisbetűs változatával.

4.5.1. Kapcsolatok

A modellek között kapcsolatokat hozhatunk létre. Például webáruházunk vevőit az **users** tábla tartalmazza, míg a vevők rendeléseit az **orders** tábla. E két tábla között van egy 1:N kapcsolat, – az **orders** tábla **user_id** mezője mutatja, hogy az adott rekord melyik vevőhöz tartozik. A modellben ezt a kapcsolatot könnyen leírhatjuk, ezáltal a kérdéses SQL-lekérdezések automatikusan létrejönnek. Összesen négy fajta kapcsolatot definiálhatunk: a **\$hasOne** az 1:1, a **\$hasMany** az 1:N, a **\$hasAndBelongsToMany** (HABTM) pedig az N:M típusú kapcsolatokért felelős. A negyedik, a **\$belongsToMany** az 1:1 vagy 1:N kapcsolatok jobb oldalán lévő tábláknál állíthatjuk be, hogy melyik táblához kapcsolódik.

Példánkban az **Order** modell az **User** modellhez tartozik, így a modellt a következőképpen definiáljuk:

```
class Order extends AppModel
{
    var $name = 'Order';
    var $belongsToMany = array('User' =>
        array('className' => 'User',
            'conditions' => '',
            'order' => '',
            'foreignKey' => 'user_id'
        ));
}
```

Ezáltal amikor a modell egy lekérdező függvényén keresztül (lásd lejjebb) adatokat kérünk le az **orders** táblából, akkor a kapcsolódó rekord is lekérdezésre kerül a **users** táblából is.

Lássunk egy példát, a **\$hasMany** kapcsolatra. A vevőknek egy vagy több rendelése lehet. Hogy ezeket a rendeléseket hozzákapcsoljuk az **User** modellhez, a modellt a következőképpen implementáljuk:

```
class User extends AppModel
{
    var $name = 'Order';
    var $hasMany = array("Order" =>
        array('className' => Order,
            'conditions' => '',
```

CAKEPHP

```
'order'           => 'name ASC',  
'foreignKey'      => 'user_id',  
'dependent'      => true,  
'exclusive'      => false,  
'finderQuery'    => ''));  
}
```

Ezáltal az User modellben automatikusan rendelkezésre állnak a vevők rendelései is.

A <http://manual.cakephp.org/chapter/models> linken a modellek közötti kapcsolatokról bővebben olvashatunk.

4.5.2. Az Appmodel osztály metódusai

Az AppModel osztályban rengeteg hasznos függvényt implementáltak a cakePHP fejlesztői. Ezek közül néhány fontosabb függvény:

```
findAll ([string $conditions, [array $fields, [string $order, [int $limit, [int $page, [int $recursive]]]]]])
```

Az egyik leggyakrabban használt metódus. Egy, a \$conditions változóban meghatározott feltétel szerinti SQL-lekérdezést valósít meg, és egy asszociatív tömbben adja vissza annak eredményt. Az \$order paraméterben adható meg, hogy mely mezők szerint rendezze az eredményhalmazt. Amennyiben a \$limit paraméter be van állítva, ennyi darab rekordot ad vissza a \$page oldaltól kezdve. Amennyiben a modellben vannak kapcsolatok (pl. \$hasMany), a \$recursive paraméter beállításával szabályozhatjuk, hogy a kapcsolat megjelenjen-e az eredményben, illetve milyen mélységben: a kapcsolódó modellnek is lehetnek további kapcsolatai, így egy egész láncolatot hozhatunk létre. Ebben a paraméterben állíthatjuk be, hogy a kapcsolati láncnak hányadik láncszeméig kell a lekérdezést elvégezni.

```
generateList([string $conditions, [string $order, [int $limit, [int $page, [string $keyPath, [string $valuePath]]]]]])
```

Ez a \$conditions-ban megadott feltételek szerint, a \$page laptól kezdődően, \$limit számú rekordból álló asszociatív tömbbel tér vissza. A tömb kulcsa a \$keyPath-ban, az értéke pedig a \$valuePath-ban meghatározott mezők lesznek. A függvénnyel gyorsan készíthetünk listát a HTML select taghez.

Az adatok mentéséhez a

```
save (array $data, [boolean validate = true, [array $fieldlist]])
```

ahol a \$data tartalmazza az adatokat, a \$validate mutatja meg, hogy van-e szükség az adatok validálására, illetve a \$fieldlist-ben fel lehet sorolni a mentéskor engedélyezett mezőket. Amennyiben a rekord mentése sikeres volt, true-val tér vissza, ellenkező esetben false-szal.

A \$data tömbnek a következő formátumban kell rendelkezésre állni:

```

Array
(
    [modelName] => Array
        (
            [fieldname1] => 'value'
            [fieldname2] => 'value'
        )
)

```

Célszerűen a HTML-lapon lévő formoknak úgy kell nevet adnunk, hogy a form beküldésekor ne kelljen „kézzel” létrehozni a fenti adatstruktúrát. Például a felhasználói regisztrációkor a login név input mezője a következőképpen néz ki:

```

<input type="text" id="UserLogin" value="" size="35"
name="data[User][login]"/>

```

A kontrollerek tárgyalásánál még visszatérünk a formból érkező adatok feldolgozására.

A frissen beillesztett rekord – amennyiben a tábla elsődleges kulcsa automatikusan kap értékét – azonosítóját a **getLastInsertId()** függvény adja vissza a mentés után.

A rekordok törléséhez a **del()** függvény használható.

```
del (mixed $id, [boolean $cascade=false])
```

ahol az **\$id** a törölni kívánt rekord azonosítója. A **\$cascade** paraméter igaz értéke esetén a kapcsolódó modellekben a megfelelő rekordok is törlődnek. Sikeres törlés esetén igazzal tér vissza, egyébként hamissal.

Ezekon a legfontosabb függvényeken kívül a **Model** osztály – mint ahogy a fejezet elején említettem – rengeteg metódust tartalmaz. Ezek között van több, a lekérdezéseket segítő függvény, hasznosak lehetnek a callback függvények, melyek a törlés, mentés, lekérdezés előtt vagy után hívódnak meg. Amennyiben az adattáblánk jelölésrendszere nem egyezik meg a cakePHP kódolási konvencióival, szabadon beállíthatjuk a modellhez kapcsolódó táblát, annak az azonosítóját stb.

Az osztály teljes leírása a http://api.cakephp.org/class_model.html címen érhető el.

4.6. Vezérlők

A cakePHP-ban a vezérlők az **appController** osztályból származnak. Feladatuk a felhasználótól érkező parancsok feldolgozása, és a megfelelő nézet segítségével az eredmények megjelenítése. A kódolási konvenció szerint a vezérlők neve mindig többes számban álló angol főnév. A saját vezérlőink az **/app/controllers/** mappában helyezkednek el. A fájljaink nevének szigorúan kell követniük a csakePHP kódolási előírásait – minden controllernek a **names_controller.php** alakot kell követniük.

A vezérlő metódusaiban vannak megvalósítva az egyes felhasználói funkciók, mint események. Például:

```
class UsersController extends AppController
{
    ...
    function signup() {
    ...
    }
    function login() {
    ...
    }

    function details($userid) {
    ...
    }
}
```

Ebben a példában a felhasználó regisztrálását a `signup()`, a beléptetését a `login()`, valamint a felhasználó adatainak megjelenítését a `details()` metódus végzi el. Az egyes funkciókhoz tartozó linkek a következőképpen néznek ki:

<http://www.pelda.hu/users/signup>

<http://www.pelda.hu/users/login>

<http://www.pelda.hu/users/details/134>

4.6.1. Metódusok

Az `appController` osztálynak van néhány saját metódusa. Ezek közül a leggyakrabban használt metódusok a következők:

set (string \$var, mixed \$value)

Ezzel a függvénnyel tudunk adatokat küldeni a nézetek számára. A `$var` paraméterben állíthatjuk be a változó nevét, a `$value`-ban pedig a változó értékét adhatjuk meg. Például a **set('szin','kék')**; hívásával a kontrollerhez tartozó nézetfájlban elérhető a `$szin` változó, aminek az értéke 'kék' lesz.

render (string \$action, string \$layout, string \$file)

A függvény automatikusan meghívódik az esemény lefutása után, mégpedig az esemény nevével megegyező nézetfájl hívja meg. Speciális esetekben e függvény segítségével eltérhetünk ettől, az esemény kódjában bárhol meghívhatjuk, és az `$action` paraméterben meg-

adott nézetfájl kerül megjelenítésre. A `$layout` paraméterben adhatjuk meg, hogy melyik elrendezésfájlt használja.

`redirect(string $url)`

az `$url` paraméterben megadott helyre irányíthatjuk a felhasználót. Az `$url` lehet relatív elérési út vagy abszolút hivatkozás.

`flash(string $message, string $url, int $pause)`

a függvény célja megegyezik az előbbivel, viszont a **`$pause`** paraméterben meghatározott másodpercig megjeleníti a **`$message`** üzenetet, mielőtt átugrik az **`$url`**-ben meghatározott linkre.

`requestAction(string $url, array $extra)`

ez a függvény az `$url`-ben meghatározott vezérlő megfelelő eseményének eredményével tér vissza. Példaként lássuk a `Users` kontroller egy metódusát, ami lekéri a szőke hajú felhasználókat:

```
function getblonds()
{
return $this->User->findAll(array('haircolor'=>'blond'));
}
```

Egy másik kontrollerben, ahol szükségünk van a szőke hajúak listájára, a következőképpen kaphatjuk meg ezt:

```
function anyfunction()
{
...
$szokek = $this->requestAction('/users/getblonds');
...
}
```

Ezt a metódust nagyon gyakran az elemek (`elements` – lásd a nézetek című alfejezetben) adatainak feltöltésére használjuk.

Ebben az osztályban is vannak `callback` függvények. Egyik legfontosabb a **`beforefilter()`**. Ez a függvény a felhasználói esemény előtt hívódik meg.

Az `/app/app_controller.php` fájlban kiegészíthetjük a `cakePHP` `appController` osztályát. Ez esetben az összes vezérlőnk örökölni fogja az itt implementált függvényeket és beállított változókat, így az itt szereplő `beforefilter()` az alkalmazásunk összes kontrollerének összes eseménye előtt le fog futni. Remek lehetőség, hogy globális változókat állítsunk be, vagy hogy jogosultságokat kezeljünk.

A `beforefilter()` függvényen kívül a következő `callback` függvények érhetőek el:

- `afterFilter()` – értelemszerűen a felhasználói esemény *után* kerül meghívásra,
- `beforeRender()` – a tartalom kiírása előtt hajtódik végre.

4.6.2. Változók

A saját vezérlőink működését a következő változókkal szabhatjuk teste:

string \$name: a vezérlő nevét tartalmazza.

array \$uses: a vezérlőben használt modelleket lehet beállítani.

array \$helpers: a vezérlőben használni kívánt segítők (helperek) listája. Az egyik legfontosabb, leggyakrabban használt helper a `htmlHelper`, amely a különböző HTML-objektumok (linkek, formok, képek) létrehozásában nyújt segítséget.

string \$layout: a használni kívánt elrendezés-sablon neve.

boolean \$autoRender: alapértelmezett értéke igaz, azaz az események végén mindig meghívásra kerül a render függvény. Hamisra állítva, az eseményekben nem történik meg automatikusan a megjelenítés.

array \$beforeFilter: a kívánt felhasználói események lefutása előtt meghívandó callback függvények listáját állíthatjuk be.

array \$components: a használni kívánt komponensek listája.

4.7. Nézetek

A nézet nem más, mint egy HTML-kódot tartalmazó sablon. Általában egy vezérlő eseményéhez egy nézetfájl tartozik, de előfordul, hogy egy sem, vagy éppen kettő vagy három. Alap esetben a nézetfájl neve megegyezik az esemény nevével. Az elhelyezkedésük mindig a `/app/views/controller/` mappában van. Kötelezően `.html` kiterjesztést kell használni.

A nézetekben csak a vezérlő `set()` metódusával átadott változókat érhetjük el.

A nézetfájlok csak a vezérlőtől kapott adatok megjelenítéséért felelősek, a honlap egésznek felépítése nem feladatuk. A honlap szerkezetét az úgynevezett layout fájlok írják le, amelyek az `/app/views/layouts` mappában találhatóak. Itt több layoutot is létrehozhatunk, és a vezérlőben a `$layout` változóval, vagy a `setLayout()` metódussal beállíthatjuk, hogy melyiket használja. Alapértelmezettként a `default.html` fájlt használja a rendszer.

A honlapoknak vannak olyan részei, melyek minden egyes funkcionál megjelennek. Ilyen például a menü vagy a belépés/kilépés form, esetleg a hirdetések. Ezeket a csakePHP terminológiában `elements`-eknek (magyarul elemek) nevezzük és az `/app/views/elements` mappában találhatóak. Szintén `.html` kiterjesztésű sablonfájlokról van szó. A layout fájlban tudjuk elhelyezni az elemek hivatkozásait, mégpedig a `$this->renderElement()` függvénnyel. Vigyázni kell arra, hogy az elemekben a layoutban meglévő változók közül egy

sem érhető el. Általában a vezérlők `$requestAction()` metódusát használjuk az elem változóinak feltöltésére – mint ahogyan azt a vezérlőkről szóló fejezetben láthattuk.

4.8. Helperek és komponensek

A komponensek és a helperek közös jellemvonásokkal bírnak: bizonyos célfeladatokat ellátó objektumokról van szó, amelyeket az alkalmazásunkban bárhol használhatunk. A komponensek a vezérlőkhöz kötődnek szorosan, míg a helperek (talán legjobban „segítőnek” lehetne fordítani) a nézetekben kerülnek felhasználásra. A komponensek az `/app/controller/component` könyvtárban találhatóak, míg a helpereknek az `/app/views/helpers` könyvtárban a helyük. Mind a komponenseket, mind a helpereket a kérdéses vezérlőben kell deklarálni, mégpedig a `var $components = array()` és a `var $helpers = array();` tömbökben.

Az egyik leggyakrabban használt komponens a `Session`. Ezzel a komponenssel a munkamenetünket kezelhetjük: lekérdezhetjük annak állapotát, adatokat írhatunk be, illetve olvashatunk ki stb.

A <http://cakeforge.org/snippet/browse.php?by=lang&lang=5> honlapon rengeteg közzétett komponens található, nagyjából látható, hogy minden felmerülő, általános problémára lehet komponens készíteni. Példaként említhetném a `pagination` komponenset (<http://cakeforge.org/snippet/detail.php?type=package&id=16>), amely a nagy rekordszámú találati listák oldalakra rendezését könnyíti meg. Ezen osztály használatával egy vezérlőben mindössze 3 sor kóddal intézhetjük el a lapozás egyébként eléggé nehézkes feladatát.

CakePHP-ban két nagyon fontos helper létezik: a `HTML` és az `AJAX`. A `HTML` helper segítségével a nézetekben formákat, formelemeket, linkeket, képeket hozhatunk létre, míg az `AJAX` helperrel a `Prototype.js` (<http://www.prototypejs.org/>) kliens oldali könyvtárra épülő `AJAX`-hívásokat lehet könnyen beilleszteni.

Szakdolgozatomban készítettem egy saját helpert, amit „`S`” helpernek neveztem el, az angol `service` szóból. Ebben a segítőben többek között a következő hasznos függvények állnak rendelkezésre

```
string $s:: toMoney(mixed $money)
```

ahol a `$money` egy egész vagy valós szám. Visszatér a szám pénzfórmátumával, pl. `12345` → `12 345 Ft`.

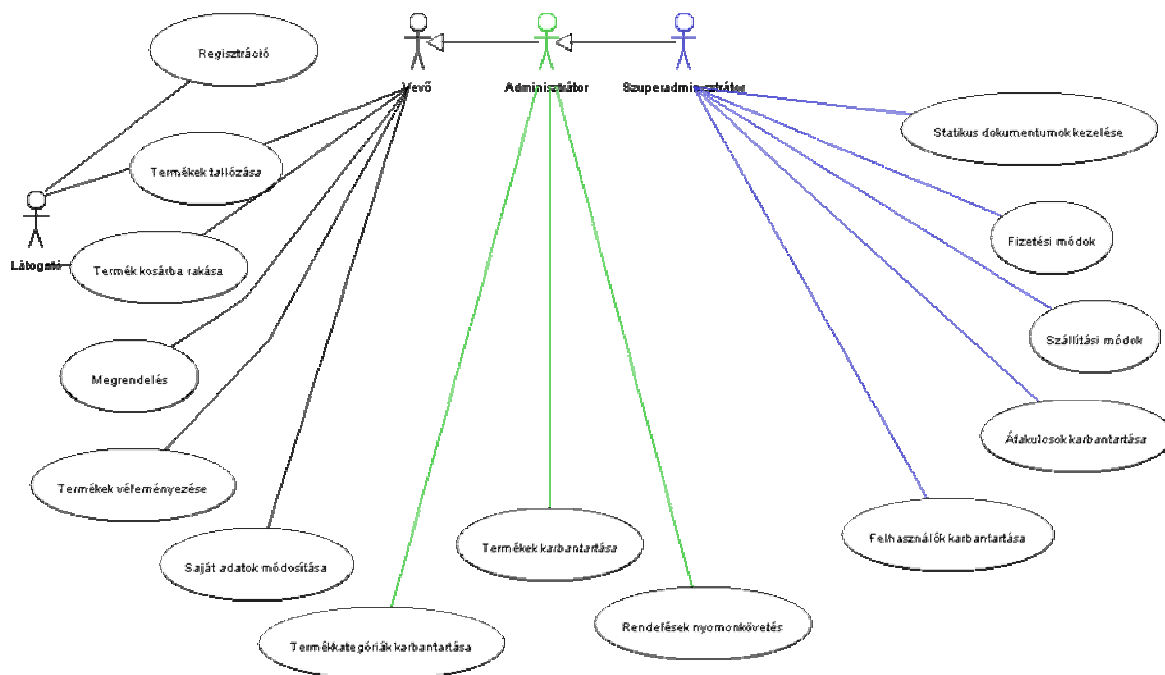
```
string $s:: cutesize(int $number)
```

ahol a `$number` egy egész szám. Visszatér a szám 1024-es váltószámú normálalakjával és a megfelelő prefixumával, pl. `2048` → `2kbytes`

5. ALKALMAZÁSFEJLESZTÉS: WEBES ÁRUHÁZ

5.1. Funkciók

Az alkalmazásunk Use-Case diagramja:



Mint ahogy az ábráról leolvashatjuk, 14 funkció lesz megvalósítva a webshopban, és ezt a 14 funkciót négy különböző típusú felhasználó használhatja. Legkevesebb joga a látogatónak van: mindössze tallózni tudja a termékeket, a kosárba teheti azokat, és regisztrálhat. A regisztrációval vevővé válik, és elérhető lesz számára a kosár tartalmának megrendelése, illetve a termékek véleményezése is. A saját adatok módosítása funkció is megilleti őt, hiszen személyes adataiban beálló változásokat az eladó tudtára kell hozni.

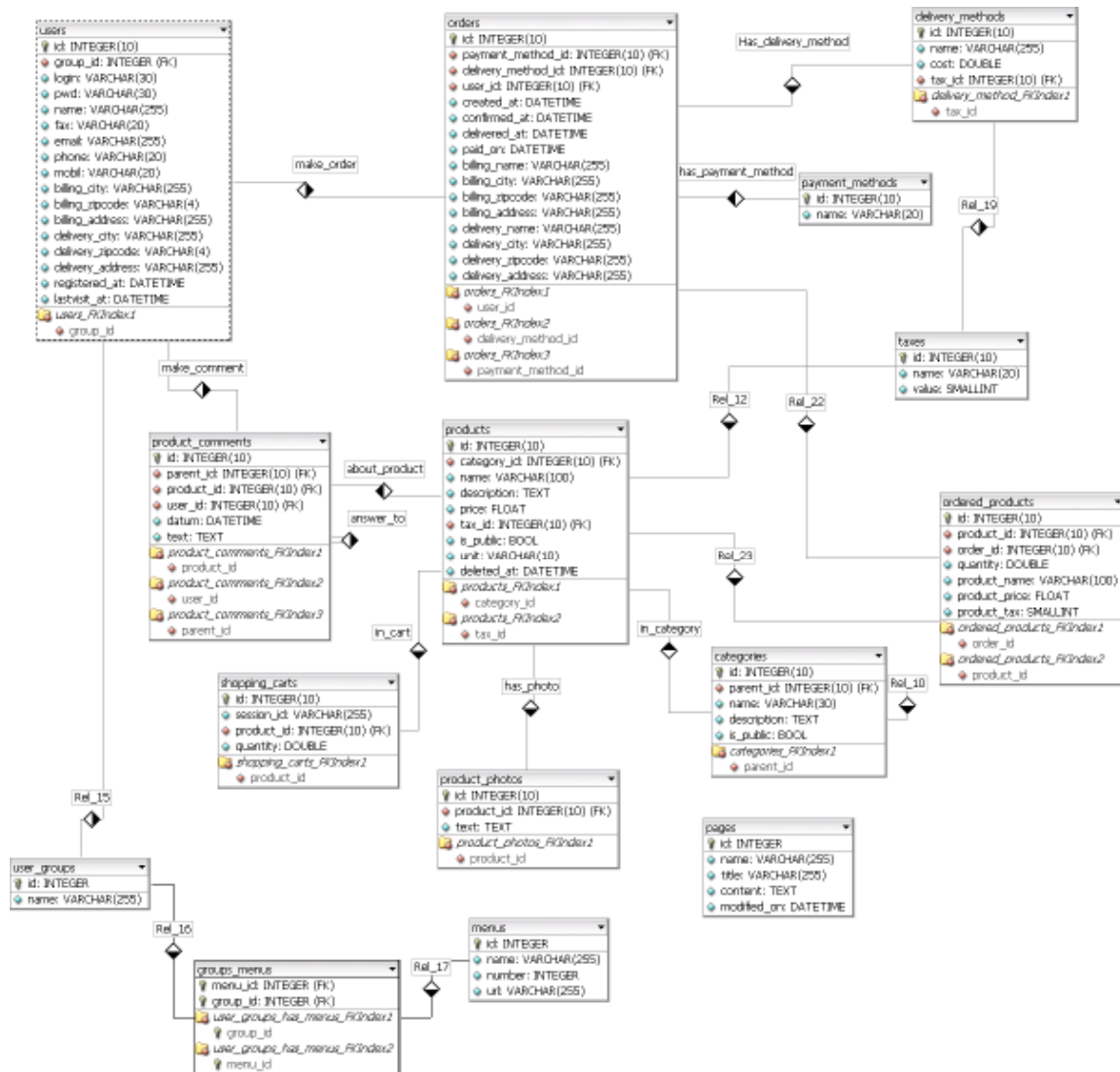
Az adminisztrátorok tudják karbantartani a termékkategóriákat, amely egy hierarchikus felépítésű struktúra. A termékeket ezekbe a kategóriákba lehet szervezni. A termékek karbantartása is az adminisztrátor feladata. A beérkezett rendelések nyomon követése, kezelése, szintén adminisztrátori hatáskör. De természetesen az adminisztrátorok számára elérhető az összes vevői funkció is.

A szuperfelhasználó a webshop korlátlan ura. A felhasználók karbantartásával áttekintheti a vevői listákat, illetve a megfelelő felhasználóknak adminisztrátori vagy szuperadmi-

nisztrátori jogokat adhat. Ő tarthatja karban a bolt alapvető beállításait: az áfakulcsok értékét, a szállítási és fizetési opciókat, illetve a honlap statikus dokumentumait is ez a típusú felhasználó kezelheti.

5.2. Adatmodell

A felhasználói funkciók alapján elkészíthetjük az adatmodellt, illetve az egyed-kapcsolat diagrammot. Az egyes entitások mezőlistáját az E-K diagrammon tudjuk nyomon követni:



Az adatmodell kialakításakor megpróbáltam betartani a CakePHP irányelveit: a táblák neve többes számú angol főnév lett, a mezőneveket szintén angol nevet kaptak. A táblák elsődleges kulcsa mindig „id” névvel vannak jelölve. A kapcsolódó tábláknál az idegen kulcs a *tablaneve_id* jelölést követi.

Az egyes táblák funkciója:

- **users:** A felhasználók személyes adatait tartalmazza
- **products:** a termékek adatai és ára.
- **product_comments:** A felhasználók véleményei a termékekről
- **product_photos:** termékek fényképeit tartalmazó tábla
- **categories:** termékkategóriák.
- **shopping_carts:** a felhasználók bevásárlókocsija. Mivel ezt a funkciót használhatják a látogatók is, így nincs az users táblához csatolva. Az egyes kosarakat a munkamenet azonosító alapján különböztetjük meg.
- **orders:** a rendeléseket tartalmazó tábla. A vevők saját adatai, a név, számlázási cím, szállítási cím első látásra redundáns módon van tárolva. De erre több okból szükség van:
 1. A vevő nem biztos, hogy saját részére rendel.
 2. A vevő a rendelés után megváltoztathatja az adatait, viszont a rendelést ez nem szabad, hogy érintse: a későbbi rendelések visszakövetése esetén kell, hogy tudjuk, a rendelés időpontjában mik voltak a vevő adatai.
- **ordered_products:** az egy rendeléshez tartozó termékek listája. Itt is külön el van tárolva a rendelt termék neve, a rendelés időpontjában lévő ára, amiatt, hogy a termékek változtatása/törlése ne érintse a régi rendeléseket.
- **taxes:** forgalmi adókulcsok
- **payment_methods:** lehetséges fizetési módok
- **delivery_methods:** lehetséges szállítási módok
- **user_groups:** felhasználói csoportok
- **menus:** a webshop menürendszerét tartalmazza
- **groups_menus:** a user_groups és a menus táblák közötti N:M kapcsolat kapcsolótáblája. Ez a tábla tartalmazza, melyik menüpontot melyik felhasználói csoport láthatja.
- **Pages:** a webshop statikus dokumentumai, pl. vásárlási feltételek stb.

5.3. Fejlesztői környezet

A program szerver oldalon Apache 1.3-as webservert, PHP4 értelmezőt és MySQL 4 adatbázisszerveret igényel.

Kliens oldalon FireFox böngésző 2.0-s változata szükséges a hibátlan működéshez.

A fájlok szerkesztéséhez Scite (<http://www.scintilla.org>) szövegszerkesztőt használtam. Az adatmodell egyed-kapcsolat diagrammjának szerkesztését a DB Designer 4 (<http://fabforce.net>) programmal oldottam meg, az Use-Case diagrammot az ArgoUML (<http://argouml.tigris.org/>) szoftverrel szerkesztettem, Microsoft Windows XP operációs rendszer alatt.

5.4. Alkalmazás felépítése

Röviden tekintsük át az egyes funkciókat, illetve azokat megvalósító vezérlőket, modelleket és nézeteket.

5.4.1. Felhasználók

A felhasználók modelljét az /app/models/user.fájlbán lévő User nevű objektumban írjuk le. Ez a fájl csak a táblára vonatkozó validációs szabályokat tartalmazza: mely mezők milyen feltételekkel tudják elfogadni az adatokat.

Az /app/controller/users_controller.php tartalmazza a felhasználókat érintő összes függvényt. A CakePHP erejét demonstrálandó, lássuk a felhasználói regisztrációt lekezelő függvényt:

```

1: function signup() {
2:     $this->layout="public";//layout file neve
3:     $this->set("target","/users/signup/");
4:     $this->set("Felirat","Új felhasználó rögzítése");
5:     if (!empty($this->data)) {
6:         $this->data["User"]["registered_at"]=date("Y-m-d h:i:s");
7:         if ($this->User->save($this->data)) {
8:             $this->flash('Felhasználó rögzítve!','/users/index/');
9:             exit();
10:        }
11:    }

```

Lássuk a függvény felépítését sorról sorra:

- A 3. és 4. sorban beállítjuk a nézet számára a \$target és a \$Felirat változót

- Ha érkezett POST adat, azaz megnyomták a regisztráció gombját, akkor a 6. sorban beállítjuk, hogy mikor is volt a regisztráció, majd a 7. sorban egyetlen egy függvényhívással rengeteg dolog történik: a beérkezett adatokat leellenőrzi a CakePHP, hogy megfelel-e a modellben definiált feltételeknek, majd létrehozza a megfelelő INSERT SQL utasítást, gondosan ügyelve az idézőjelekre, és le is futtatja azt. Ha az adatok érvényesítése és az SQL-utasítás futtatása sikeres volt, a Model::save() függvény igazgal tér vissza, és a Controller::flash() függvény hatására 3 másodpercig megjeleníti a „Felhasználó rögzítve feliratot”, majd a /users/index linkre ugrik.
- Amennyiben hamissal tért vissza a save() függvény, a függvényhez tartozó nézet ismét megjelenik, és automatikusan megjelennek a hibás adatbevitelre utaló felhasználót tájékoztató üzenetek is.

A html kimenetet az /app/views/users/signup.html tartalmazza. Ez a fájl írja le a formot, ez tartalmazza a kliens oldali validációt tartalmazó javascriptet, valamint ez írja ki a hibaüzeneteket is.

Azt hiszem, ez a függvény nagyon jól demonstrálja a Cake erejét: egy 17 mezős tábla INSERT és UPDATE parancsainak elkészítése sok időbe telik, nem beszélve arról, hogy amennyiben az adatmodellben változás történik, ezeket az SQL-kéréseket szintén módosítani kell. Esetünkben elég a nézetben hozzátenni a megfelelő beviteli mezőt, az SQL-kérések automatikusan kiegészülnek a megfelelő értékekkel. **A fejlesztés során egyetlen egy SQL-kérést nem kellett leírni!** Ezzel rengeteg időt spórolhatunk meg, és rengeteg hibalehetőséget zárhatunk ki.

Az adatok érvényesítését a <http://bakery.cakephp.org/articles/view/improved-advance-validation-with-parameters> linken olvasható cikk alapján oldottam meg, felhasználva az itt található kódokat. A megoldás egyszerűsége mellett leginkább az tetszett, hogy egy mezőhöz sok szabályt lehet beállítani. Például a login név választásához összesen négy szabályt definiáltam:

- Nem lehet üres
- Csak alfanumerikus karaktereket tartalmazhat
- Egyedinek kell lennie, azaz az adatbázisban nem lehet még egy felhasználó ugyanazzal a login névvel
- A hosszának 6 és 50 karakter közé kell esnie.

Természetesen az egyes szabályokhoz külön-külön lehet beállítani a hibaüzeneteket is.

A felhasználók többi függvénye és azok feladatai:

- **admin_index():** az adminisztrációs felületen elérhető funkció a felhasználók listájának előállításához
- **admin_add():** új felhasználó hozzáadása az admin felületen.

- **admin_show(\$id):** \$id azonosítójú felhasználó adatlapjának megtekintése
- **admin_del(\$id):** \$id azonosítójú felhasználó törlése
- **admin_edit(\$id):** \$id azonosítójú felhasználó adatainak módosítása
- **login():** felhasználó bejelentkezése. Ha sikeres volt a bejelentkezés, azaz megfelelő login/jelszó párossal azonosítja magát a felhasználó, akkor az aktuális munkamenetbe feljegyzésre kerül a felhasználó összes adata, valamint az, hogy melyik felhasználói csoportba tartozik. Ez utóbbi a jogosultságok kezelésénél és a menüszerkezet kiírásánál kap szerepet.
- **logout():** kijelentkezés.

A függvények a `cake.aer.hu/users/függvény_neve/p/a/r/a/m/s` formátumú linkekkel érhetőek el, ahol a `p/a/r/a/m/s` a függvény paramétereit jelenti. Kivétel ez alól az `admin_előtaggal` rendelkező függvények: ezek ugyanis az adminisztrációs felülethez tartoznak, ezeket a <http://cake.aer.hu/admin/users/függvényneve> címen lehet elérni. Az adminisztrátori előtagot a `/app/config/core.php` fájlban lévő `CAKE_ADMIN` konstans tartalmazza.

5.4.2. Termékek

A termékek modelljében létre kell hoznunk a kapcsolódó táblákhoz való asszociációkat, mint ahogy a 4.5.1-es fejezetben láttuk. Ez a megoldás rengeteg gondot levesz a vállunkról: amikor lekérünk egy terméket az adatbázisból, az összes itt definiált adat is megjelenik, azaz nem csak a `tax_id`, hanem az ehhez a forgalmi adóhoz tartozó név és érték is. Mindezt egy egyszerű `$this->Product->read(null, $id)` függvénnyel, ami az alábbi struktúrájú tömböt adja vissza:

```
Array
(
    [Product] => Array
        (
            [id] => 5
            [category_id] => 28
            [name] => Canon EOS 350D kit (18-55 mm)
            [description] => Canon ...
            [price] => 122000
            [tax_id] => 1
            [is_public] => 1
            [unit] => db
            [deleted_at] => 0000-00-00 00:00:00
        )
    [Category] => Array
        (
            [id] => 28
            [parent_id] => 27
            [name] => Digitális fényképezőgépek
            [description] =>
            [is_public] => 1
        )
    [Tax] => Array
        (
            [id] => 1
            [name] => 20% Áfa
        )
)
```

```

        [value] => 20
    )
[0] => Array
(
    [gross] => 146400
    [taxvalue] => 24400
)
[Photo] => Array
(
    [0] => Array
    (
        [id] => 9
        [product_id] => 5
        [text] => Előnézet
    )
    [1] => Array
    (
        [id] => 10
        [product_id] => 5
        [text] => Hátnézet
    )
    [2] => Array
    (
        [id] => 11
        [product_id] => 5
        [text] => Bal nézet, de milyen
    )
)
[Comment] => Array
(
    [0] => Array
    (
        [id] => 3
        [product_comments_to] => 0
        [product_id] => 5
        [user_id] => 2
        [datum] => 2007-05-25 01:43:10
        [text] => Nagyon jó termék.
    )
)
)

```

Tehát egy tömbben megkapjuk az összes adatot, de még a számított bruttó árat és a fizetendő áfát is. Ahogy látjuk, itt érhetjük el a termékről formált véleményeket is. Emiatt a termékek véleményezését vezérlő kontrollerben nincs szükség külön a vélemények megjelenítését végző függvényre.

A controllerben lévő funkciók:

- **index (\$category_id=1):** a \$category_id-jű termékek listázása a publikus oldalon.
- **admin_index (\$category_id=1):** a \$category_id-jű termékek listázása az admin oldalon.
- **admin_trash():** a termékek végleges fizikai törlése helyett szemeteskukát használunk. A szemetkukában lévő termékek listázása.
- **admin_add():** új termék hozzáadása.
- **admin_edit():** termék adatainak módosítása.

- **admin_show():** termék adatlapjának megjelenítése.
- **admin_setstatus():** a termék megjelenítése/elrejtése a publikus oldalon. Ez a függvény az admin/products/index oldalról hívódik meg AJAX-kérésen keresztül.
- **admin_del():** termék törlése. Nem fizikai törlésről, hanem a deleted_at mező beállításáról van szó. A termék ezáltal – logikailag – a szemetesbe kerül, ahonnan később vissza lehet állítani.
- **admin_undel():** termék visszaállítása a szemetesből.
- **admin_undelall():** az összes termék visszaállítása a szemetesből
- **details(\$id):** az \$id azonosítójú termék részletes leírásának megjelenítése a publikus oldalon.

5.4.3. Termékfotók

Minden termékhez tetszőleges számú fotót lehet hozzáfűzni. A fotókhoz leírást lehet csatolni magyarázandó, mit is láthatunk a fényképen. A termékfotók modelljében bevezetésre került egy del(\$id) nevű függvény. Ugyanis ebben a modellben nem csupán az adatbázisból, hanem a merevlemezről is le kell törölni a fényképet. A fotó feltöltését viszont technikai okok miatt a vezérlőben kellett megoldani: sokkal praktikusabb volt így.

A termékfotók feltöltését igyekeztem a Prototype.js kódkönyvtárra épített, látványos, felhasználóbarát módon megoldani:

- A feltöltéskor a form targetje egy rejtett iframe. Amikor feltöltődött egy fájl, akkor a felöltést végző rutin (admin_upload()) JavaScript callback függvényen keresztül üzen a fő alkalmazásnak a feltöltés állapotáról. A fő alkalmazásban lévő JavaScript függvény az eredmény szerint vagy hibaüzenetet küld a felhasználónak (amennyiben nem sikerült a fájlt felölteni, vagy a fájl nem felelt meg a feltételeknek), vagy frissíti a fájllistát, és tájékoztatja a felhasználót, hogy minden rendben.
- Drag&drop technika a fotók törléséhez
- In-place editor a fotók leírásának szerkesztéséhez

A vezérlő funkciói:

- **admin_index(\$product_id):** a \$product_id azonosítójú termékhez tartozó fotók megjelenítése.
- **admin_getphotos(\$product_id):** a \$product_id azonosítójú termék fotólistájának lekérése AJAX-kérésre. Ekkor a honlap többi része nem töltődik le, csak a megfelelő nézet.

- **admin_upload(\$product_id):** a \$product_id azonosítójú termékhez fotó feltöltése
- **admin_changetext(\$id):** az in-place editor mentésekor meghívódó függvény, ami elmenti az \$id azonosítójú fotóhoz tartozó szöveget.
- **admin_del(\$id):** az \$id azonosítójú fotó törlése
- **slideshow(\$product_id,\$id=null):** a \$product_id termékhez tartozó fényképek megjelenítése, az \$id fotóval kezdődően.

5.4.4. Termékek véleményezése

A vevők elmondhatják a véleményüket a termékről a honlapon. Az ehhez a funkcióhoz szükséges controller a product_comments_controller.php fájlban van definiálva. A vezérlő funkciói:

- **add(\$id=0):** az \$id azonosítójú termékhez való hozzászólás rögzítése.

5.4.5. Kategóriák

A termékek hierarchikus kategóriákba, egy egygyökerű fába vannak rendezve. A tábla önhivatkozó, a parent_id mező mutatja meg, hogy az adott kategóriának melyik a szülője. A fa gyökere az 1-es azonosítójú, Webshop nevű kategória. A teljes fa lekérdezése ez esetben csak rekurzív függvénnyel lehetséges. A szükséges függvényeket a kategóriákat kezelő modellben valósítottam meg:

- **getRoute(\$id):** a gyökértől az \$id azonosítójú kategóriaelemhez vezető elemeket adja vissza egy tömbben.
- **routeToHtml(\$id, \$link):** feladata az \$id azonosítójú elemhez vezető út navigációs célra használt linkjeinek létrehozása. A \$link változóban a /controller/action párost lehet átadni.
- **getTreeForSelect(\$forbidden=false, \$id = 0, \$level = 0):** a html select mező elemeinek feltöltésére használható formában visszaadja a fa összes elemét, melynek az \$id a szülője. A függvény rekurzívan kerül meghívásra. A \$forbidden azonosítójú elem és annak gyermekei nem kerülnek bele az eredményhalmazba. Erre a kategória módosításánál van szükség: logikailag nem megengedhető, hogy egy kategóriaelemnek önmaga vagy valamelyik gyermeke legyen a szülője.
- **getTree(\$id=0, \$is_public=false):** visszatér az \$id gyökerű fát tartalmazó hash tömbbel. Az eredménytömb indexe a kategóriaelem azonosítója, az értéke pedig az elem összes adatát tartalmazó tömb. A gyermekelemek a *Children* indexű tömbben helyezkednek el.

- **getAllChildren(\$id):** Visszatér az \$id elemből induló fának az összes elemével egy egydimenziós tömbben.
- **setPublic (\$parent, \$is_public):** a \$parent azonosítójú elem és annak összes gyermekének láthatóságát állítja be. Ha az \$is_public hamis, akkor az összes gyermeknek is beállítja ezt az értékét hamisra. Amennyiben az \$is_public igaz, csak a \$parent elem láthatóságát állítja igazra.

5.4.6. A kosár

A vásárlók a kiválasztott terméket a kosárba tehetik. A kosár tartalmát szabadon lehet módosítani: terméket betenni és kivenni, illetve a kosárban lévő termék mennyiségét módosítani lehet. A kosár működéséért a shopping_carts_controller osztály függvényei felelősek.

- **setting(\$product_id=null, \$quantity=0):** tulajdonképpen ez a függvény valósítja meg egyedül az összes funkciót: a product_id azonosítójú termékből \$quantity mennyiséget rak a kosárba. Ha a quantity nulla, akkor törli a terméket a kosárból. Ezt a függvényt kizárólag AJAX-kéréseken keresztül hívjuk meg, amikor a felhasználó a kosárba szeretné rakni a kiválasztott terméket, vagy a termék mennyiségét szeretné változtatni.
- **refresh():** a kosár tartalmának felfrissítését végző függvény. Ez a függvény is AJAX-kérésre fut le. CakePHP technikai érdekességként szeretném megemlíteni a Configure::write('debug', '0'); statikus függvény hívását az első sorban, amire azért van szükség, hogy a CakePHP debug üzenetek ne jelenjenek meg feleslegesen a függvény kimenetében. Mivel most sima HTML kimenetet eredményez ez a függvényünk, csupán esztétikai problémát okoznának a debug üzenetek (a kosárban, a termékek listája alatt megjelenne az összes SQL-lekérdezés, ami felesleges és zavaró). Azonban, ha a kimenet JSON formátumú adat vagy XML lenne, amit kliens oldali JavaScript dolgoz fel, ezek az üzenetek teljesen elrontanák az adatok formátumát, és a feldolgozás sikertelen lenne. Ám ezzel a függvényhívással futás közben kapcsolhatjuk ki a CakePHP debug üzeneteit.

5.4.7. Rendelések

A vevők a kosárba tett termékeket megrendelhetik. Amennyiben látogató szeretne rendelni, a program a vevői regisztrációs formra irányítja elsőként az adatainak rögzítése céljából. A rendeléskor meg kell adni a számlázási és szállítási adatokat, a fizetés és a termék átvételének módját. Az orders_controller függvényei:

- **make():** ez a függvény végzi a rendelést. Ellenőrzi, hogy a felhasználó be van-e jelentkezve, a rendelés lezárásakor létrehozza a szükséges bejegyzést az orders táblába és áttölti a kosár tartalmát az ordered_products táblába.
- **myorders():** a vevők a saját rendeléseit nyomon követő funkció.

- **admin_index(\$type="new")**: az adminisztrátor számára jeleníti meg a beérkezett rendeléseket. A \$type paraméter szerint állítja össze a listát:
 - new: csak az új rendeléseket jeleníti meg
 - delivery: szállításra váró rendelések
 - payment: fizetésre váró rendelések
 - closed: lezárt rendelések
 - egyéb más: minden rendelés
- **admin_details(\$id=null)**: az \$id azonosítójú rendelés részletes adatainak megjelenítését végző függvény
- **admin_confirm(\$id=null), admin_deliver(\$id=null), admin_pay(\$id=null)**: az \$id azonosítójú rendelés visszaigazol, kiszállított, kifizetett státusát beállító függvények
- **setDM(\$id=null)**: Az \$id azonosítójú rendelés szállítási módját beállító segéd-függvény. AJAX-on keresztül hívódik meg a termékek rendelésekor.

5.4.8. Kódtáblák karbantartása

A program futtatásához szükséges kódtáblák (forgalmi adók, szállítási és fizetési módok) CRUD¹ műveleteit a TaxesController, PaymentMethodsController és a DeliveryMethodsController osztályok végzik a hozzájuk tartozó Tax, PaymentMethod és DeliveryMethod modellek segítségével.

A CakePHP egyik erőssége az ilyen jellegű, egyszerű alkalmazások készítésében rejlik: a szükséges fájlokat (a kontrollert, a modellt és a nézeteket) a parancssorból indított varázsló néhány egyszerű kérdésre válaszolva „megsüti” (bake) nekünk, mint egy sütit (cake). A <http://nio.infor96.com/webbaker/> honlapon található a Bake-hez készült webes felület is, amivel könnyűszerrel készíthetők el a szükséges fájlok.

Meg kell adnunk a modell nevét (pl. Tax), milyen típusú „sütést” szeretnénk, esetünkben ez most a „popular”. Valamint megadhatjuk, hogy milyen actions-okat szeretnénk, vesszővel elválasztva felsorolhatjuk a szükséges függvényeket. Egy gombnyomásra létrejön a modell, a vezérlő és az összes nézetfájl.

¹ A négy alapvető adatbázis-művelet, Create, Read, Update és Delete angol szavaiból képzett mozaikszó. Forrás: [http://en.wikipedia.org/wiki/CRUD_\(acronym\)](http://en.wikipedia.org/wiki/CRUD_(acronym)).

5.4.9. Menüszerkezet és jogosultságok

A honlap menüszerkezetét a menus tábla tartalmazza. A menüszerkezethez szükséges adatok előállításáért a Menu modell `getTree($group, $user_group, $parent_id=0)` metódusa felelős. Visszatér a HTML formázott menüszerkezettel. Paraméterek:

- `$group`: a menücsoporthoz kódja
- `$user_group`: a felhasználói csoport
- `$parent_id`: a gyökérelem azonosítója

A jogosultságot az `/app/app_controller.php` fájlban lévő `AppController` osztályban definiált `beforefilter()` függvényben ellenőrizzük. Ez a függvény minden kontroller minden függvényének lefutása előtt hívódik meg, így ideális hely annak eldöntésére, hogy az adott funkciót egyáltalán elérheti-e a felhasználó. A `groups_menus` kapcsolótáblában van feljegyezve, hogy melyik csoportnak melyik menüponthoz van joga. A program megnézi, hogy melyik volt az igényelt funkció, annak mi az azonosítója a menus táblában, és kikeresi, hogy van-e `groups_menus` táblában ezzel az azonosítóval és a felhasználó csoportjának azonosítójával rendelkező sor. Amennyiben igen, a program futása a kívánt felhasználói funkcióval folytatódik. Amennyiben a felhasználó számára nem engedélyezett menüpontot próbál elérni, a program egy hibüzenet kíséretében a főoldalra irányítja.

5.4.10. Kliensoldali programozás

A PHP-kód kimenete XHTML 1.0 transitional dokumentumtípust használ. Az alkalmazás megjelenését külső CSS 2.1 fájlok szabályozzák. A felhasználóbarát viselkedést, az AJAX-hívásokat, a fogd-és-vidd megoldást a `Script.aculo.us` (<http://script.aculo.us/>) és a `Prototype.js` (<http://www.prototypejs.org/>) nevű JavaScript kódkönyvtárak segítségével oldottam meg.

Az alkalmazást Firefox 2.0 böngészőhöz fejlesztettem, Webdeveloper toolbar és Firebug beépülő komponensek használatával, melyek a fejlesztést nagyban támogatják. Mivel szakdolgozatomban a CakePHP alkalmazásfejlesztés a célja, és nem a kliens oldali megoldások bemutatása, így a fejlesztés során nem koncentráltam arra, hogy a program más böngészőben is rendesen fusson. Így például Internet Explorer alatt néhány hiba fel-lephet.

6. FELHASZNÁLÓI DOKUMENTÁCIÓ

6.1. Rendszerkövetelmények

Az alkalmazás zökkenőmentes használatához a következő rendszerkomponensek szükségesek:

- FireFox 2.0 webböngésző, JavaScript futtatásának és sütik (Cookies) használatának engedélyezésével
- Legalább 1024×768-as képernyőfelbontás és 16 bites színmélység
- Széles sávú internetkapcsolat

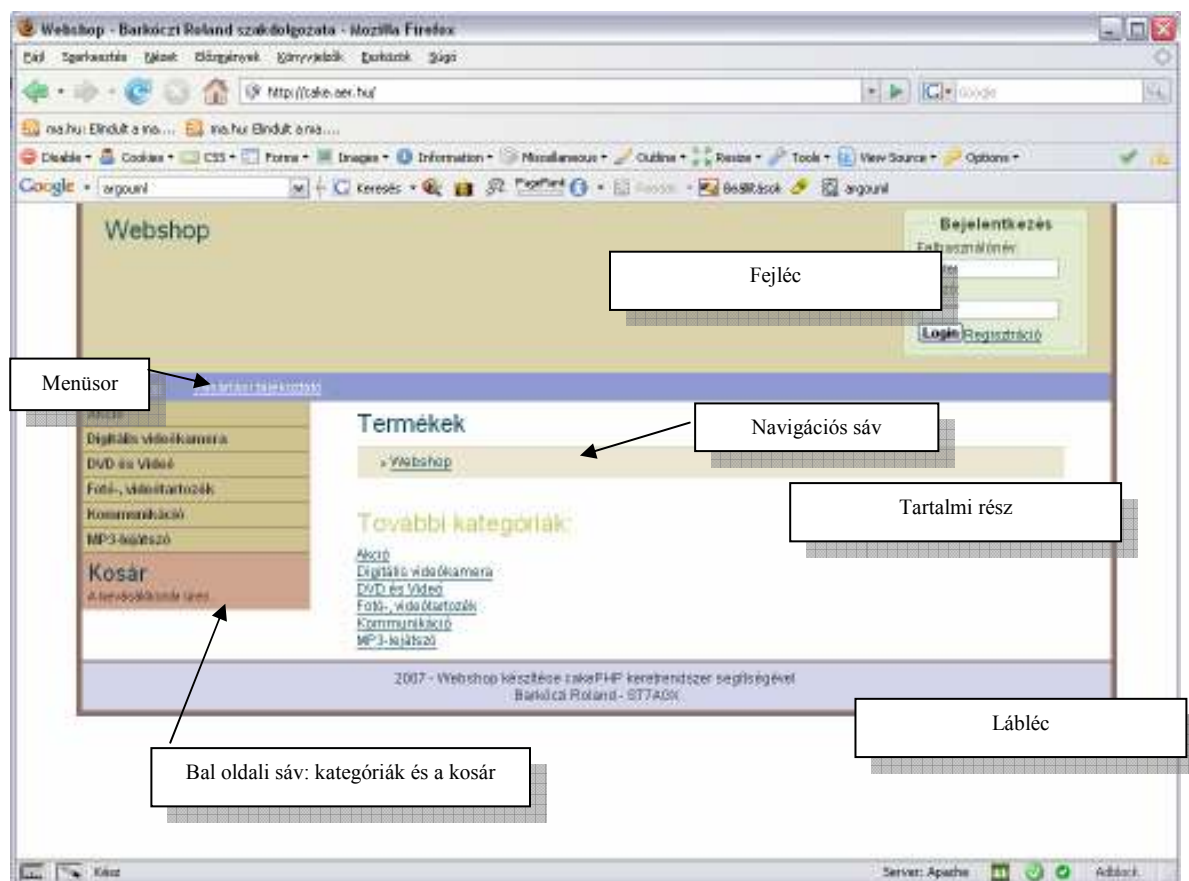
6.2. Publikus felület

A program használatához navigáljunk a <http://cake.aer.hu> oldalra.

6.2.1. Felhasználói felület

A felhasználói felület a következő részekre tagolódik:

- Feljéc a belépéshez szükséges beviteli mezőkkel és a regisztrációhoz szükséges linkkel
- Menü
- Bal oldali terület a termékkategóriákkal és a kosárral
- Navigációs sáv
- Tartalmi rész
- Lábléc



Az alkalmazás kezdőlapja

6.2.2. Termékek tallózása

Válasszunk kategóriát a bal oldali sávon lévő menüből vagy a tartalmi részen lévő „További kategóriák” felirat alatti listából. A navigációs sávon láthatjuk, hogy a kategóriák hierarchiájában éppen hol vagyunk.

A választott kategóriába tartozó termékek megjelennek a tartalmi részen egy összefoglaló táblázatban. A táblázatban csak a termék fotója, a neve, bruttó és nettó ára, valamint a kosárba tevéshez szükséges link szerepel. Amennyiben ismerjük a terméket, már itt is a kosárba tehetjük. Ehhez kattintsunk a „Kosárba” linkre, majd a megjelenő beviteli mezőben adjuk meg a kíván mennyiséget és kattintsunk az „Ok” gombra. Ekkor a bal oldali sávon, a kosár tartalma automatikusan frissül.

A termék részletes adatlapjának megtekintéséhez kattintsunk a termék nevére. Ekkor megjelenik a részletes leírás, az ár, a termék fotója, valamint a vásárlók véleménye a termékről. A fotóra kattintva a felbukkanó ablakban végiglapozhatjuk a termékhez tartozó összes fotó.

Amennyiben elnyerte tetszésünket a termék, az adatlapon lévő Kosárba linkre kattintva, az előbb ismertetett módszer segítségével az elektronikus kosarunkba rakhatjuk azt.



Termék kosárba rakása

A termék részletes adatlapjáról a „Vissza” gombbal vagy a navigációs sáv segítségével léphetünk vissza.

6.2.3. Kosár

A termékek böngészése során mindig látható a kosár tartalma. A kosár tartalmának ellenőrzéséhez és annak módosításához kattintsunk a „Kosár megnyitása” linkre. A megjelenő táblázatban módosíthatjuk a vásárolni kívánt termékek mennyiségét: írjuk be a kívánt mennyiséget és nyomjuk le az Enter vagy Tab billentyűt.

Amennyiben egy terméket mégsem szeretnénk megrendelni, kattintsunk a termék neve alatti „Törlés” linkre, vagy írjunk nullát a rendelni kívánt mennyiség mezőjébe.

Amennyiben meg szeretnénk rendelni a kosárban lévő termékeket, be kell jelentkezünk vagy – amennyiben még nincsen meg a belépéshez szükséges jelszavunk – regisztrálnunk kell.

6.2.4. Regisztráció

A termékek rendeléséhez meg kell adni adatainkat. Kattintsunk a fejlécben a „Regisztráció” linkre. Töltsük ki a megjelenő űrlapot. A hibás adatbevitelre a program megfelelő hibüzenetekkel reagál.

6.2.5. Vásárlás

Regisztráció után lehetőség nyílik a kosárban lévő termékek megrendelésére. Kattintsunk a kosár táblázata alatt vagy a menüsorban lévő Pénztár linkre.

A megjelenő űrlapon a szükség szerint módosítsuk a szállítási és számlázási címet, válasszuk ki a kívánt fizetési és szállítási módot és ellenőrizzük a megrendelni kívánt termékeket. Amennyiben minden rendben van, kattintsunk a „Megrendelem” feliratú gombra.

6.2.6. Saját rendelések ellenőrzése

A menüben válasszuk a „Rendelések” menüpontot. Ekkor megjelenik az összes rendelésünk táblázata, ahol nyomon követhetjük, hogy a rendelést mikor regisztrálták, mikor igazolták vissza, mikor szállították ki, és mikor érkezett meg a számla ellenértéke. A „részletek” linkre kattintva ellenőrizhetjük az egyes rendelések adatait.

6.3. Adminisztrációs felület

Az adminisztrációs felület a <http://cake.aer.hu/admin> címen érhető el. Az itteni funkciók csak bejelentkezett adminisztrátorok és szuperadminisztrátorok számára érhető el.

6.3.1. Kategóriák

Itt tarthatjuk karban a termékkategóriákat. A megjelenő fában a jelölőnégyzet ki és bepipálásával szabályozhatjuk, hogy a kategória megjelenjen-e a publikus felületen. A kategóriaelem adatainak szerkesztéséhez kattintsunk a jegyzetömböt szimbolizáló ikonra, a törléshez pedig a piros X-re.

A „Tevékenységek” feliratú mezőben található opció segítségével adhatunk hozzá a fához új elemet. A megjelenő űrlapon adjuk meg a kategória nevét, válasszuk ki a szülőjét, és opcionálisan csatoljunk egy leírást a kategóriához.



Termékkategóriák

6.3.2. Termékek

Ebben a menüpontban tarthatjuk karban a webshopban árulni kívánt termékeket.

Termékek

Tevékenységek

Keresés:

Webshop
▼

[+ Új termék hozzáadása](#)

[Kuka megtekintése](#)

Termék	<input type="checkbox"/> Státusz	Műveletek
Canon EOS 350D kit (18-55 mm)	✓	<input type="button" value="▶"/> <input type="button" value="✖"/>
Panasonic DMC-FZ50, fekete	✓	<input type="button" value="▶"/> <input type="button" value="✖"/>
Fujifilm FinePix S6500fd	✓	<input type="button" value="▶"/> <input type="button" value="✖"/>

Termékek listája

A „Tevékenységek” területen lévő űrlap segítségével kereshetünk terméket név szerint, illetve szűrhetünk bizonyos kategóriákra. Itt kapott helyet az új termék rögzítéséhez szükséges link és a Kuka megtekintése link is.

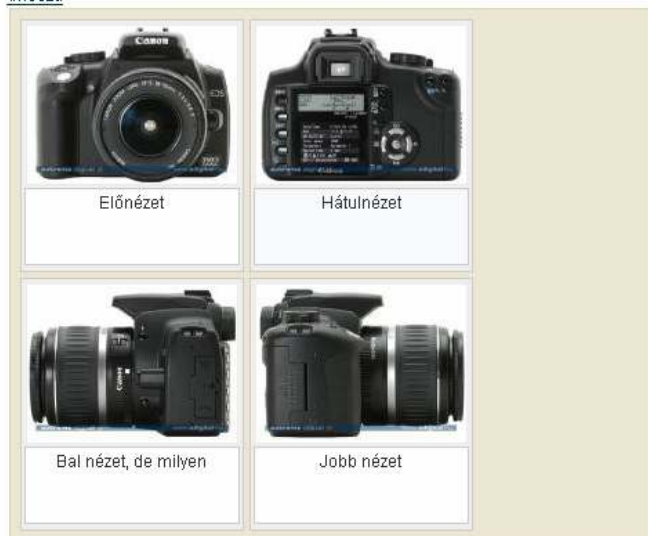
A termékek nevére kattintva megnézhetjük annak részletes adatait. A státusz jelölőnégyzettel szabályozhatjuk a termék publikusságát (erre például átmeneti készlethiány miatt lehet szükség, amikor a termékből nem lehet rendelni, de törölni sem akarjuk).

A jegyzetomb ikonra kattintva módosíthatjuk, a piros X-szel törölhetjük a terméket. (A továbbiakban, más menüpontokban ez a két ikon mindig ezt a két műveletet fogja jelenteni, így nem kerülnek részletezésre). A termék törlése nem jelent fizikai törlést, hanem a szemeteskukába kerül, ahonnan később – szükség esetén – vissza lehet azt állítani.

6.3.3. Termékfotók

Canon EOS 350D kit (18-55 mm) fényképek

[«Vissza](#)



Fénykép feltöltése

Fájl:

Megjegyzés:

A termékhez tartozó fotók kezelése

A fényképezőgépre kattintva tölthetjük fel a termékhez tartozó fotókat. Új fotó feltöltéséhez válasszuk ki a tallózás gomb segítségével a kívánt fotót a számítógépünkről, adjuk meg a fotó leírását, majd kattintsunk a mentés gombra. A fotók szövegeinek módosításához, kattintsunk a módosítani kívánt szövegre, majd az „Ok” gombbal menthetjük a kívánt változtatásokat. Egy fotó törléséhez húzza a fotót a szemeteskukára.

A „Vissza” gombbal visszatérhetünk a terméklistához.

6.3.4. Rendelések

A rendszerben rögzített rendelések karbantartására szolgáló menüpont. Külön listázhatóak az új, szállításra váró, fizetésre váró, a lezárt vagy az összes rendelés.

A rendelés részletes adatainak megjelenítéséhez és nyomtatásához kattintson a „részletek” gombra. A különböző tevékenységekhez (Visszaigazolás, Kiszállítás, Fizetve) értelemszerűen kattintson a táblázatban a megfelelő linkre.

Beérkezett rendelések

[Mind](#) | [Újak](#) | [Szállításra várók](#) | [Fizetésre várók](#) | [Lezárt rendelések](#)

Új rendelések

Találatok száma: 10

Azonosító	Megrendelő	Rendelés dátum	Visszaigazolás dátuma	Szállítás dátuma	Fizetve	
8	Barkóczy Roland, 8500 Pápa, Szladik u.	2007-05-13 05:57:17	Visszaigazolás	Kiszállítás	Fizetés	Részletek
9	Barkóczy Roland, 8500 Pápa, Szladik u.	2007-05-13 06:12:38	Visszaigazolás	Kiszállítás	Fizetés	Részletek
10	Barkóczy Roland, 8500 Pápa, Szladik u.	2007-05-13 06:14:31	Visszaigazolás	Kiszállítás	Fizetés	Részletek
11	Barkóczy Roland, 8500 Pápa, Szladik u.	2007-05-13 06:19:18	Visszaigazolás	Kiszállítás	Fizetés	Részletek
12	Barkóczy Roland, 8500 Pápa, Szladik u.	2007-05-13 06:26:15	Visszaigazolás	Kiszállítás	Fizetés	Részletek
13	Barkóczy Roland, 8500 Pápa, Szladik u.	2007-05-13 06:26:41	Visszaigazolás	Kiszállítás	Fizetés	Részletek
14	Barkóczy Roland, 8500 Pápa, Szladik u.	2007-05-13 10:01:40	Visszaigazolás	Kiszállítás	Fizetés	Részletek
15	Barkóczy Roland, 8500 Pápa, Szladik u.	2007-05-13 10:03:15	Visszaigazolás	Kiszállítás	Fizetés	Részletek
16	Barkóczy Roland, 8500 Pápa, Szladik u.	2007-05-22 02:33:09	Visszaigazolás	Kiszállítás	Fizetés	Részletek

Új rendelések listája

6.3.5. Áfakulcsok, szállítási és fizetési módok

Ezekben a menüpontokban tarthatjuk karban a megfelelő segédlistákat. Mindhárom menüpont felépítése szinte teljesen azonos, így kezelésük is.

Szállítási módok

Tevékenységek

[+Új fizetési mód hozzáadása](#)

Fizetési mód neve	Műveletek
Kézpénz	<input type="checkbox"/> <input type="checkbox"/>
Banki átutalás	<input type="checkbox"/> <input type="checkbox"/>
Csekk	<input type="checkbox"/> <input type="checkbox"/>
Bankkártya	<input type="checkbox"/> <input type="checkbox"/>

Fizetési módok az adatbázisban

6.3.6. Dokumentumok

A honlap statikus dokumentumait lehet itt karbantartani. Jelenleg egy dokumentum van rögzítve, a Vásárlási feltételek, igény szerint több dokumentumot lehet létrehozni, és azokat megjeleníteni a publikus oldalon.

7. ÖSSZEFOGLALÁS

A fejlesztés folyamán alaposan megismerkedhettünk a CakePHP keretrendszer előnyeivel és hátrányaival. A CakePHP ígéretéhez híven tényleg lehetővé teszi a gyors alkalmazásfejlesztést, hiszen egy egyszerű, egy-két táblának kezelését végző modulok fejlesztése – még Bake nélkül is – mindössze néhány tíz percet vesz igénybe, aminek nagy részét a nézetfájlok testre szabásával kell eltölteni. A bonyolultabb modulokhoz, mint például a hierarchikus adatok kezelése, már a modelleket kell kiegészíteni, de kis gyakorlattal ez is egyszerű feladat. A modellek összekapcsolásának (hasMany, hasOne stb.) lehetőségével nagyon gyorsan hozhatunk létre komplex lekérdezéseket. A beépített debug funkciók pedig felbecsülhetetlen értékű támogatást nyújtottak a fejlesztés során.

Hátrányként említeném meg a html segítő eléggé kezdetleges megoldásait. A HTML-űrlapelemek létrehozása nem eléggé támogatott, itt sokat kell még fejleszteni, hogy egy igazán jól használható segítőt kapjunk. Illetve van néhány idegesítő megoldás is benne:

- A `html->selectTag()`, ami egy `select` mezőt hoz létre a megadott hash-ből, alapszintűen létrehoz egy üres opciót is a lista elején.
- Az paraméterekben átadott szövegek HTML-entitásait (<, >, & stb.) átkonvertálja, és nem entitásként jeleníti meg, pl. a > jel helyett > jelenik meg a kimenetben.

Ami a megvalósított alkalmazást illeti, egyáltalán nem alkalmas kereskedelmi forgalomba hozásra, de nagyon jó kiinduló alap lehet egy kereskedelmi termék fejlesztéséhez, például a következő funkciók megvalósításával:

- A felhasználók regisztrációjakor szükséges lenne leellenőrizni a megadott e-mail cím helyességét egy megerősítő linket tartalmazó levél küldésével. A felhasználó csak ezen validációs linkre való kattintás után tudnának belépni és vásárolni.
- A rendelésről szintén kellene e-mailt küldeni mind a vevőnek, mind az adminisztrátornak
- A rendelések állapotainak változásakor (visszaigazolás, kiszállítás, fizetés rögzítése) szintén e-mailben kellene tájékoztatni a vevőt.
- A funkciók egy része nem működik rendesen Internet Explorer 6 alatt. Ennek oka a nevezett böngésző rossz megvalósításában keresendő, mivel az XHTML, a CSS és a JavaScript értelmezője sem működik a vonatkozó szabványok ajánlásainak megfelelően. De a széles körű elterjedése és használata miatt szükséges lenne a kliens oldali programozást erre a böngészőre is optimalizálni.

8. IRODALOMJEGYZÉK

- [1] DeBolt, Virginia.: *HTML és CSS szerkesztés stílusosan*. Kiskapu kiadó, 2005.
- [2] George Schlossnagle: *PHP fejlesztés felsőfokon*. Kiskapu kiadó, 2005.
- [3] Jeffrey D. Ullman – Jennifer Widom: *Adatbázis rendszerek*. Panem kft. 1998
- [4] PHP MVC Frameworks: http://www.phpwact.org/php/mvc_frameworks
- [5] CakePHP hivatalos honlap: <http://cakephp.org/>
- [6] CakePHP felhasználói dokumentáció: <http://manual.cakephp.org>
- [7] CakePHP API dokumentáció: <http://api.cakephp.org>
- [8] Model-view-controller : <http://en.wikipedia.org/wiki/Model-view-controller>
- [9] Prado keretrendszer: <http://www.pradosoft.com/>
- [10] Seagull: <http://www.seagullproject.org>
- [11] Zephyr: <http://zephyr-php.sourceforge.net/>
- [12] Symphony: <http://www.symfony-project.com>

9. MELLÉKLET

9.1. Vezérlők forrásállományai

9.1.1. categories_controller.php

```
<?php
class CategoriesController extends ApplicationController
{
    var $name="Category";
    var $helpers = array('Html', 'Javascript', 'Ajax', "S", 'Form', 'Error');
    var $uses=array("Category", "Product", "ProductPhoto");

/**
 * Function admin_index($parent_id)
 * Kategóriák listájának megjelenítése
 * Paramléterek:
 * $parent_id: a szülő kategóriájának azonosítója
 */
    function admin_index($parent_id=1){
        $this->set("parent_id", $parent_id);
        $this->set("categories", $this->Category->getTree(1));
    } //end index()

/**
 * Function admin_add($parent_id)
 * Új kategória mentése
 * Paramléterek:
 * $parent_id: a szülő kategóriájának azonosítója
 */
    function admin_add($parent_id=1){
        $this->set("Felirat", "Új kategória létrehozása");
        $this->set("target", "/admin/categories/add");
        $this->set("sup", $this->Category->getTreeForSelect());
        if (!empty($this->data)){
            if ($this->Category->save($this->data)){
                $this->flash('Kategória
mentve!', '/admin/categories/index/'. $this->data["Category"]["parent_id"]);
            }
        }
        $this->data["Category"]["parent_id"] = $parent_id;
    } //end admin_add()

/**
 * Function admin_edit($id)
 * Kategória adatainak módosítása
 * Paramléterek:
 * $id: a kategória azonosítója
 */
    function admin_edit($id=0){
        $this->set("Felirat", "Kategória adatainak módosítása");
        $this->set("target", "/admin/categories/edit/".$id);
        $this->set("sup", $this->Category->getTreeForSelect($id));
        if (empty($this->data)){
            $this->Category->id = $id;
            $this->data = $this->Category->read();
        } else{
            $this->data["Category"]["id"]=$id;
            if ($this->Category->save($this->data['Category'])){

```

MELLÉKLET

```

        $this->flash('Adatok
mentve.', '/admin/categories/index/'. $this->data["Category"]["parent_id"]);
    }
}
    $this->render("admin_add");
} //end admin_edit()

/**
 * Function admin_setstatus($id)
 * Kategória elrejtése/megjelenítése
 * Paramléterek:
 * $id: a kategória azonosítója
 */
function admin_setstatus($id){
    $this->cacheAction = false;
    $this->Category->Id = $id;
    $cat = $this->Category->read('is_public', $id);
    Configure::write('debug', '0');
    $is_public = $cat["Category"]["is_public"];
    if($is_public==1)$is_public =0; else $is_public =1;
    $this->Category->setPublic($id,$is_public);
    $this->Category->cacheQueries=false;
    $this->set("categories", $this->Category->getTree(1));
    $this->render("categoritable", "ajax");
} //end setstatus()

/**
 * Function admin_del($id)
 * Kategória törlése
 * Paramléterek:
 * $id: a kategória azonosítója
 */
function admin_del($id){
    $this->Category->id = $id;
    $children = $this->Category->getAllChildren($id); //összes gyermeket is
törölni kell
    $children[$id]=true; //hozzáadjuk magát a kategóriát is
    if(is_array($children)){ //ha vannak gyermekek
        foreach($children as $key=>$value){ //bejárjuk őket
            $product = $this->Product->findAll("category_id=$key"); //kigyűjtjük
az egyes kategóriákban lévő termékeket
            if(is_array($product)) //ha van termék, akkor azt töröljük, ami ugy
azok deleted_at mezőjének beállítását jelenti
                {
                    foreach($product as $p)
                        $this-
>requestAction('/admin/products/del/'. $p["Product"]["id"], array("return"));
                }
            $this->Category->del($key);
        }
    }
    $this->flash('Kategóriátörölve.', '/admin/categories/index/');
} //end del();

/**
 * Function getpubliclist()
 * Visszatér a publikus kategóriák listájával
 * Paramléterek:
 * nincs
 */
function getpubliclist()
{
    return $this->Category->getTree(1, true);
}

/**
 * Function welcome()

```



```

* adminisztrációs felület köszönő
* Paramléterek:
* nincs
*/
function welcome(){
}
}
?>

```

9.1.2. Delivery_methods_controller.php

```

<?php
class DeliveryMethodsController extends ApplicationController
{
    var $name="DeliveryMethod";
    var $helpers = array('Html', 'Javascript', 'Ajax', "S", "Form", "Error");
    var $uses=array("DeliveryMethod", "Tax");

/**
 * Function admin_index()
 * szállítási módok listáját adja vissza
 * Paramléterek:
 * nincs
 */
function admin_index(){
    $this->set("methods", $this->DeliveryMethod->findAll());
} //end index()

/**
 * Function admin_add()
 * Szállítási mód hozzáadása
 * Paramléterek:
 * nincs
 */
function admin_add(){
    if (empty($this->data)){
        $this->set("taxes", $this->Tax->
>generateList(null, "name", null, "{n}.Tax.id", "{n}.Tax.name"));
        $this->set("Felirat", "Új szállítási mód létrehozása");
        $this->set("target", "/admin/delivery_methods/add");
    }
    else
    {
        if ($this->DeliveryMethod->save($this->data))
        {
            $this->flash('Új szállítási mód
mentve!', '/admin/delivery_methods/index/');
        }
    }
} //end add()

/**
 * Function admin_edit($id=null)
 * Szállítási mód módosítáa
 * Paramléterek:
 * (int) $id: metódus azonosítója
 */
function admin_edit($id=null){
    if (empty($this->data)){

```

MELLÉKLET

```
        $this->set("taxes", $this->Tax-
>generateList(null, "name", null, "{n}.Tax.id", "{n}.Tax.name"));
        $this->set("Felirat", "Szállítási mód módosítása");
        $this->set("target", "/admin/delivery_methods/edit/$id");
        $this->DeliveryMethod->id = $id;
        $this->data = $this->DeliveryMethod->read();
        $this->render("admin_add");
    }
    else
    {
        $this->data["DeliveryMethod"]["id"]=$id;
        if ($this->DeliveryMethod->save($this->data))
        {
            $this->flash('Szállítási mód
mentve!', '/admin/delivery_methods/index/');
        }
    }
} //end edit()

/**
 * Function admin_del($id=null)
 * Szállítási mód törlése
 * Paramléterek:
 * (int) $id: metódus azonosítója
 */
function admin_del($id=null){
    $this->DeliveryMethod->id=$id;
    $this->DeliveryMethod->del();
    $this->flash('Szállítási mód
törölve!', '/admin/delivery_methods/index/');
} //end del()
}
?>
```

9.1.3. orders_controller.php

```
<?php
class OrdersController extends ApplicationController
{
    var $name="Order";
    var
    $uses=array("Order", "OrderedProduct", "User", "Product", "ShoppingCart", "Del
iveryMethod", "PaymentMethod");
    var $helpers = array('Html', 'Javascript', 'Ajax', "S",
"Form", "Error");

/**
 * Function make(void)
 * termékek megrendelése
 * Paramléterek:
 * nincs
 */
function make(){
    $this->layout="public";
    $user = $this->Session->read("User");
    if($user){//ha be van jelentkezve
        //szükséges változók beállítása a nézetekhez
        $this->set("user", $this->User->find(array("id"=>$user["id"])));
        $items = $this->ShoppingCart-
>findAll(array("ShoppingCart.session_id"=>session_id()),
```

MELLÉKLET

```

null,
null,
null,
null,
2);
$this->set("items", $items);

$delivery_methods = $this->DeliveryMethod->findAll();
$this->set("delivery_methods", $delivery_methods);

$payment_methods = $this->PaymentMethod->findAll();
$this->set("payment_methods", $payment_methods);

if(!empty($this->data)){//megrendelés
    $this->data["Order"]["user_id"]=$user["id"];
    $this->data["Order"]["created_at"]=date("Y-m-d h:i:s");
    if($this->Order->save($this->data){
        $this->data["OrderedProduct"]["order_id"]=        $this->Order-
>getLastInsertId();

        //termékek áttöltése a bevásárlókosárból
        $items                =                $this->ShoppingCart-
>findAll(array('session_id'=>session_id()),
            null,
            null,
            null,
            null,
            2);
        if(is_array($items))
            foreach($items as $item){
                $this->OrderedProduct->id=null;
                $this->data["OrderedProduct"]["product_id"]                =
$item["ShoppingCart"]["product_id"];
                $this->data["OrderedProduct"]["quantity"]                =
$item["ShoppingCart"]["quantity"];
                $this->data["OrderedProduct"]["product_name"]                =
$item["Product"]["name"];
                $this->data["OrderedProduct"]["product_price"]                =
$item["Product"]["price"];
                $this->data["OrderedProduct"]["product_tax"]                =
$item["Product"]["Tax"]["value"];
                $this->OrderedProduct->save($this->data);
            }
        //bevásárlókocsi ürítése
        $this->ShoppingCart->execute("delete from shopping_carts
where session_id = '".session_id()."'");
        $this->flash('Vásárlás rendben', "/products");
        return false;
    }
}

if(!empty($this->data["Order"]["delivery_method_id"])){
    $selectedDM = $this->DeliveryMethod->findById($this-
>data["Order"]["delivery_method_id"]);
    $this->set("selectedDM", $selectedDM);
}
if(empty($this->data["Order"]["billing_name"]))
    $this->data["Order"]["billing_name"]=$user["name"];
if(empty($this->data["Order"]["billing_zipcode"]))
    $this-
>data["Order"]["billing_zipcode"]=$user["billing_zipcode"];

```

MELLÉKLET

```

        if(empty($this->data["Order"]["billing_city"]))
            $this->data["Order"]["billing_city"]=$user["billing_city"];
        if(empty($this->data["Order"]["billing_address"]))
            $this->data["Order"]["billing_address"]=$user["billing_address"];

        if(empty($this->data["Order"]["delivery_name"]))
            $this->data["Order"]["delivery_name"]=$user["name"];
        if(empty($this->data["Order"]["delivery_zipcode"]))
            $this->data["Order"]["delivery_zipcode"]=$user["delivery_zipcode"]?
            $user["delivery_zipcode"]:$user["billing_zipcode"];
        if(empty($this->data["Order"]["delivery_city"]))
            $this->data["Order"]["delivery_city"]=$user["delivery_city"]?
            $user["delivery_city"]:$user["billing_city"];
        if(empty($this->data["Order"]["delivery_address"]))
            $this->data["Order"]["delivery_address"]=$user["delivery_address"]?
            $user["delivery_address"]:$user["billing_address"];

        $this->render("make");
    }
    else $this->render("make_login");
} //end index

/**
 * Function myorders()
 * Saját rendelések nyomkövetése
 * Paramléterek:
 * nics
 */
function myorders(){
    $this->layout="public";
    $user=$this->Session->read("User");
    $this->set("orders",$this->Order->findAll(
ray("user_id"=>$user["id"]),
null,
array("Order.created_at
asc")));
}

/**
 * Function details($order_id=null)
 * Saját rendelés részleteinek megtekintéses
 * Paramléterek:
 * (int) $order_id: rendelés azonosítója
 */
function details($order_id=null){
    $this->layout="public";
    $user=$this->Session->read("User");
    $this->set("order",$this->Order->find(
array("user_id"=>$user["id"],"Order.id"=>$order_id)));
}

/**
 * Function admin_index($type)
 * Rendelés listája és adminisztrációja
 * Paramléterek:

```

MELLÉKLET

```

* (string) $type: milyen listát állítson össze
* all: mindn rendelés listázása
* new: csak az új rendelések (azaz a confirmed_at=0);
* delivery: kiszállításra várók (azaz a delivered_at = 0);
* payment: fizetésre várók (azaz a paid_on=0)
* closed: lezárt rendelések (azaz minden dátum mezője ki van töltve)
*/
function admin_index($type="new"){
    $this->Session->write("ordersadmin",$type);
    if($type=="new"){
        $cond = "Order.confirmed_at = 0";
        $this->set("felirat","Új rendelések");
    }
    elseif($type=="delivery"){
        $cond = "Order.delivered_at = 0";
        $this->set("felirat","Szállításra váró rendelések");
    }
    elseif($type=="payment"){
        $cond = "Order.paid_on = 0";
        $this->set("felirat","Fizetésre váró rendelések");
    }
    elseif($type=="closed"){
        $cond = array("Order.paid_on != 0 and Order.delivered_at != 0 and
confirmed_at!= 0 ");
        $this->set("felirat","Lezárt rendelések");
    }
    else {
        $cond= "";
        $ord = "";
        $this->set("felirat","Minden rendelés");
    }
    $this->set("orders",$this->Order->findAll($cond));
}

/**
* Function admin_details($id)
* Rendelés részletes adatai
* Paramléterek:
* (int) $id: rendelés azonosítója
*/
function admin_details($id=null){
    $order = $this->Order->find(array("Order.id"=>$id));
    if($order){
        $this->set("ordersadmin",$this->Session->read("ordersadmin"));
        $this->set("order",$order);
    }
    else{
        $this->Flash("Hibás rendelési szám","/orders/admin/".$this->
>Session->read("ordersadmin"));
        return false;
    }
}

/**
* Function admin_confirm($id)
* Rendelés visszaigazolása
* Paramléterek:
* (int) $id: rendelés azonosítója
*/

```

MELLÉKLET

```

function admin_confirm($id=null) {
    $order = $this->Order->find(array("Order.id"=>$id));
    if($order) {
        $this->Order->id = $id;
        $this->Order->saveField("confirmed_at", date("Y-m-d H:i:s"));
        //id jön, hogy e-mailt küldönk az ügyfélnek, hogy rendben van a
rendelés
        $this->Flash("Rendelés                visszaigazolása                rend-
ben", "/admin/orders/index/". $this->Session->read("ordersadmin"));
    }
    else
        $this->Flash("Hibás                rendelési
szám", "/admin/orders/index/". $this->Session->read("ordersadmin"));
    return false;
}

/**
 * Function admin_deliver($id)
 * Rendelés Kiszállításaása
 * Paramléterek:
 * (int) $id: rendelés azonosítója
 */
function admin_deliver($id=null) {
    $order = $this->Order->find(array("Order.id"=>$id));
    if($order) {
        $this->Order->id = $id;
        $this->Order->saveField("delivered_at", date("Y-m-d H:i:s"));
        //id jön, hogy e-mailt küldönk az ügyfélnek, hogy rendben van a
rendelés
        $this->Flash("Rendelés                kiszállítása                rend-
ben", "/admin/orders/index/". $this->Session->read("ordersadmin"));
    }
    else
        $this->Flash("Hibás                rendelési
szám", "/admin/orders/index/". $this->Session->read("ordersadmin"));
    return false;
}

/**
 * Function admin_pay($id)
 * Rendelés Kiszákifizetése
 * Paramléterek:
 * (int) $id: rendelés azonosítója
 */
function admin_pay($id=null) {
    $order = $this->Order->find(array("Order.id"=>$id));
    if($order) {
        $this->Order->id = $id;
        $this->Order->saveField("paid_on", date("Y-m-d H:i:s"));
        //id jön, hogy e-mailt küldönk az ügyfélnek, hogy rendben van a
rendelés
        $this->Flash("Rendelés                kiszállítása                rend-
ben", "/admin/orders/index/". $this->Session->read("ordersadmin"));
    }
    else
        $this->Flash("Hibás                rendelési
szám", "/admin/orders/index/". $this->Session->read("ordersadmin"));
    return false;
}

/**
 * Function setDM($id)

```

MELLÉKLET

```
* szállítási mód beállítása
* Paramléterek:
* (int) $id: szállítási mód azonosítója
*/
function setDM($id=null){
    $items = $this->ShoppingCart-
>findAll(array("ShoppingCart.session_id"=>session_id()),
    null,
    null,
    null,
    null,
    2);
    $this->set("items",$items);
    $selectedDM = $this->DeliveryMethod->findById($id);
    $this->set("selectedDM",$selectedDM);
    $this->render('account','ajax');
}
}
?>
```

9.1.4. pages_controller.php

```
<?php
class PagesController extends ApplicationController
{
    var $name="Page";
    var $helpers = array(
ray('Html','Javascript','Ajax','S','Form','Error');

/**
 * Function index($id=null)
 * Statikus tartalom megjelenítése
 * Paramléterek:
 * (int) $id: lap azonosítója
 */
    function index($id=null){
        $this->layout="public";
        $page = $this->Page->findById($id);
        if($page) $this->set("page",$page);
        else $this->render("404");
    } //end index()
/**
 * Function admin_index()
 * Dokumentumok listáját adja vissza
 * Paramléterek:
 * nincs
 */
    function admin_index(){
        $this->set("pages",$this->Page->findAll());
    } //end index()

/**
 * Function admin_add()
 * dokumentum hozzáadása
 * Paramléterek:
 * nincs
 */
    function admin_add(){
```

MELLÉKLET

```
$this->set("Felirat", "Új dokumentum létrehozása");
$this->set("target", "/admin/pages/add");
if (!empty($this->data)) {
    $this->data["Page"]["modified_on"] = date("Y-m-d H:i:s");
    if ($this->Page->save($this->data)) {
        $this->flash('Új dokumentum
mentve!', '/admin/pages/index/');
    }
}

} //end admin_add()

/**
 * Function admin_edit($id=null)
 * dokumentum módosítása
 * Paramléterek:
 * (int) $id: dokumentum azonosítója
 */
function admin_edit($id=null) {
    $this->set("Felirat", "Dokumentum módosítása");
    $this->set("target", "/admin/pages/edit/$id");
    if (empty($this->data)) {
        $this->Page->id = $id;
        $this->data = $this->Page->read();
        $this->render("admin_add");
    }
    else
    {
        $this->data["Page"]["id"] = $id;
        $this->data["Page"]["modified_on"] = date("Y-m-d H:i:s");
        if ($this->Page->save($this->data))
        {
            $this->flash('Dokumentum mentve!', '/admin/pages/index/');
        }
    }
} //end admin_edit()

/**
 * Function admin_del($id=null)
 * dokumentum törlése
 * Paramléterek:
 * (int) $id: dokumentum azonosítója
 */
function admin_del($id=null) {
    $this->Page->id = $id;
    $this->Page->del();
    $this->flash('Dokumentum törölve!', '/admin/pages/index/');
} //end del()

}
?>
```

9.1.5. payment_methods_controller

```
<?php
class PaymentMethodsController extends ApplicationController
{
    var $name = "PaymentMethod";
    var $helpers = array('Html', 'Javascript', 'Ajax', "S", "Form", "Error");
    // var $uses = array("ShoppingCart", "Product", "Tax");
}
```


MELLÉKLET

```
/**
 * Function admin_index()
 * fizetési módok listáját adja vissza
 * Paramléterek:
 * nincs
 */
function admin_index(){
    $this->set("methods",$this->PaymentMethod->findAll());
}

/**
 * Function add()
 * fizetési mód hozzáadása
 * Paramléterek:
 * nincs
 */
function admin_add(){
    $this->set("Felirat","Új fizetési mód létrehozása");
    $this->set("target","/admin/payment_methods/add");
    if ($this->PaymentMethod->save($this->data))
    {
        $this->flash('Új fizetési mód
mentve!','/admin/payment_methods/index/');
    }

}

/**
 * Function admin_edit($id=null)
 * Fizetési mód módosítáa
 * Paramléterek:
 * (int) $id: metódus azonosítója
 */
function admin_edit($id=null){
    $this->set("Felirat","Fizetési mód módosítása");
    $this->set("target","/admin/payment_methods/edit/$id");
    if (empty($this->data)){
        $this->PaymentMethod->id = $id;
        $this->data = $this->PaymentMethod->read();
        $this->render("admin_add");
    }
    else
    {
        $this->data["PaymentMethod"]["id"]=$id;
        if ($this->PaymentMethod->save($this->data))
        {
            $this->flash('Fizetési mód
mentve!','/admin/payment_methods/index/');
        }
    }
}

/**
 * Function admin_del($id=null)
 * Fizetési mód törlése
 * Paramléterek:
 * (int) $id: metódus azonosítója
 */
function admin_del($id=null){
```

```

        $this->PaymentMethod->id=$id;
        $this->PaymentMethod->del();
        $this->flash('Fizetési mód törölve!',
'/admin/payment_methods/index/');
    }//end del()
}
?>

```

9.1.6. product_comments_controller.php

```

<?php
class ProductCommentsController extends ApplicationController
{
    var $name="ProductComment";
    var $helpers = array('Html', 'Javascript', 'Ajax', "S", 'Form', 'Error');
    var $uses=array("Product", "ProductComment");

    /**
     * Function add()
     * komment hozzáadása a termékhez
     * Paramléterek:
     * nincs
     */
    function add($id=0){
        $this->layout="public";
        $user = $this->Session->read("User");
        $this->Product->id = $this->data['ProductComment']['product_id'];
        $product = $this->Product->read();
        if($this->data && !empty($user) && !empty($product)){
            $this->data['ProductComment']['user_id']=$user["id"];
            $this->data['ProductComment']['datum']=date("Y-m-d h:i:s");
            if ($this->ProductComment->save($this->data['ProductComment']))
            {
                $this->flash('Adatok mentve.', '/products/details/'. $this->data["ProductComment"]["product_id"]);
                exit();
            }
            $this->requestAction("/products/details/" . $this->data["ProductComment"]["product_id"]);
            exit();
        } //end add()
    }
}
?>

```

9.1.7. product_photos_controller.php

```

<?php
class ProductPhotosController extends ApplicationController
{
    var $name="ProductPhoto";
    var $uses=array("ProductPhoto", "Product");
    var $helpers = array('Html', 'Javascript', 'Ajax', "S");

```

MELLÉKLET

```

    var $path =    "/img/photos/";
/**
 * Function admin_index($product_id)
 * A $product_id azonosítójú termékhez tartozó fényképek megjelenítése és
 feltöltése
 * Paramléterek:
 * (int) $product_id: a termék azonosítója
 */
function admin_index($product_id){
    $this->Product->id=$product_id;
    $product= $this->Product->read();
    $this->set("product_id",$product_id);
    if(!$product){//nincs ilyen termék
        $this->flash("Nincs ilyen termék!","/admin/products/index");
    }
    else{
        $this->set("product",$product);
        $this->set("path",$this->path);
        $photos = $this->ProductPhoto-
>findAll("product_id=$product_id");
        $this->set("photos",$photos);
    }
}

/**
 * Function admin_getphotos($product_id)
 * A $product_id azonosítójú termékhez tartozó fényképek listájának ajax
 lekérése és megjelenítése
 * Paramléterek:
 * (int) $product_id: a termék azonosítója
 */
function admin_getphotos($product_id){
    $this->set("path",$this->path);
    $photos = $this->ProductPhoto-
>findAll("product_id=$product_id");
    $this->set("photos",$photos);
    Configure::write('debug', '0');
    $this->render("productphotos","ajax");
}

/**
 * Function admin_upload($product_id)
 * A $product_id azonosítójú termékhez tartozó fényképek feltöltése
 * Paramléterek:
 * (int) $product_id: a termék azonosítója
 */
function admin_upload($product_id){
    $this->Product->id=$product_id;
    $product= $this->Product->read();
    if(!$product){//nincs ilyen termék
        $this->flash("Nincs ilyen termék!","/products/index");
    }
    else{
        $error=0;
        $fajl = $this->data["ProductPhoto"]["img"];
        $fn = $fajl['name'];//fajlnev
        $fe = strtolower(substr($fn,-3,3));
        $fs = $fajl['size'];
        if($fn=="")$error="Hiba! Nincs fájl kiválasztva!";
        elseif($fe!="jpg")$error="Hiba! Csak jpg fájlokat lehet

```

MELLÉKLET

```

feltölteni!";
    elseif($fs>200000)$error="Hiba! Maximális méret: 200kB";
    elseif($fajl["error"])$error="Sikertelen feltöltés!";
    if(!$error){
        $this->ProductPhoto->save($this->data);
        move_uploaded_file($fajl["tmp_name"],WWW_ROOT.$this-
>path.$this->ProductPhoto->getLastInsertId().".jpg");
    }
    else $this->set("error",$error);
    $this->render("admin_upload");
}
}

/**
 * Function admin_changetext($id)
 * Az $id azonosítójú fényképhez tartozó szöveg megváltoztatása, AJAX-on
 keresztül
 * Paramléterek:
 * (int) $id: a fénykép azonosítója
 */
function admin_changetext($id){
    Configure::write('debug', '0');
    $id = substr($id,4);
    $this->ProductPhoto->id= $id;
    $desc = htmlspecialchars(urldecode($_POST["text"]));
    $desc = iconv("UTF-8","ISO-8859-2",$desc);
    $this->ProductPhoto->saveField("text",$desc);
    $this->set("desc",$desc);
    $this->render("admin_changetext","ajax");
}

/**
 * Function admin_del($id)
 * Az $id azonosítójú fényképtörlése
 * Paramléterek:
 * (int) $id: a fénykép azonosítója
 */
function admin_del($id){
    $id = substr($id,2);
    $this->ProductPhoto->id= $id;
    $product_id=$this->ProductPhoto->read("product_id");
    $this->ProductPhoto->delete($id);
    unlink(WWW_ROOT.$this->path.$id.".jpg");
    $this->render("empty","ajax");
}

/**
 * Function slideshow($product_id,$id=null)
 * Az $product_id azonosítójú termékhez tartozó fényképek lapozgatása az
 $id azonosítójú fényképpel kezdve
 * Paramléterek:
 * (int) $product_id: termék azonosítója
 * (int) $id: a fénykép azonosítója
 */
function slideshow($product_id,$id=null)
{
    Configure::write('debug', '0');
    $this->layout="slideshow";
    $this->Product->id=$product_id;
    $product = $this->Product->read();
    if(!$product){//nem létező termékre való hivatkozás
        $this->set("error","Nem létező termék");
    }
}

```

```

        else{
            $this->set("product", $product);
            $this->set("path", $this->path);
            $photos = $this->ProductPhoto-
>findAll("product_id=$product_id");
            $this->set("photos", $photos);
            $this->set("current", $id);
        }
    }
}
?>

```

9.1.8. products_controller.php

```

<?php
class ProductsController extends ApplicationController
{
    var $name="Product";
    var
    $uses=array("Product", "Category", "Tax", "ProductPhoto", "ProductComment");
    var $helpers = ar-
ray('Html', 'Javascript', 'Ajax', "S", "Error", 'Form');

/**
 * Function index($category_id=1)
 * Termékek listájának megjelenítése
 * Paramléterek:
 * (int) $category_id: a termék kategóriájának azonosítója, alapértel-
mezett értéke 1.
 * A Search nevű formból is kaphat paramétereket:
 * (string) data["Search"]["keyword"]: kulcsszó szerinti keresés
 * (string) data["Search"]["category_id"]: termékkategória, amiben keres
 */
    function index($category_id=1){
        $this->set("route", $this->Category-
>routeToHtml($category_id, "/products/index"));
        $this->set("category_children", $this->Category-
>getChildren($category_id, true));

        $this->set("path", $this->ProductPhoto->path);
        $this->set("products", $this->Product-
>findAll(array("category_id"=>$category_id, "Product.is_public"=>"1"),
array("*",
"(100+Tax.value)/100*Product.price as gross",
"Tax.value/100*Product.price as taxvalue")
));
        $this->set("category_id", $category_id);
        $this->layout="public";
    } //end index()

/**
 * Function admin_index($category_id=1)
 * Termékek listájának megjelenítése
 * Paramléterek:
 * (int) $category_id: a termék kategóriájának azonosítója, alapértel-
mezett értéke 1.
 * A Search nevű formból is kaphat paramétereket:

```

MELLÉKLET

```

* (string) data["Search"]["keyword"]: kulcsszó szerinti keresés
* (string) data["Search"]["category_id"]: termékkategória, amiben keres
*/
function admin_index($category_id=1){
    if(!empty($this->data))//keresés történt
    {
        $cond = "";
        if($this->data["Search"]["keyword"])
            $cond[]='Product.name like "'. $this->
>data["Search"]["keyword"].' "%';
        if($this->data["Search"]["category_id"]>1){
            $cond[] = ' Product.category_id = ' . $this->
>data["Search"]["category_id"];
            $cond[] = ' Product.deleted_at = 0';
        }
        else $cond[] = ' Product.deleted_at = 0';
        $this->set("products", $this->Product->findAll($cond));
        $this->set("category_id", $this->
>data["Search"]["category_id"]);
    }
    else{
        if($category_id==1)$this->set("products", $this->Product->
>findAll("deleted_at=0"));
        else $this->set("products", $this->Product->
>findAll("category_id = $category_id"));
        $this->set("category_id", $category_id);
    }
    $this->set("categories", $this->Category->getTreeForSelect());
}

} //end admin_index()

/**
 * Function admin_trash( )
 * A szemétben lévő termékek listája
 * Paramléterek:
 * void
 * A Search nevű formból is kaphat paramétereket:
 * (string) data["Search"]["keyword"]: kulcsszó szerinti keresés
 */
function admin_trash(){
    if(!empty($this->data))//keresés történt
    {
        $cond = "";
        if($this->data["Search"]["keyword"])
            $cond[]='Product.name like "'. $this->
>data["Search"]["keyword"].' "%';
        }
        $cond[] = ' Product.deleted_at != 0';
        $this->set("products", $this->Product->findAll($cond));
    }
}

/**
 * Function admin_add($category_id=1)
 * Új termék rögzítése
 * Paramléterek:
 * (int) $category_id: a termék kategóriájának azonosítója, alapértel-
mezett értéke 1.
 */
function admin_add($category_id=1){
    $this->set("Felirat", "Új termék létrehozása");
}

```

MELLÉKLET

```

        $this->set("target", "/admin/products/add/" . $category_id);
        $this->set("category_id", $category_id);
        $this->set("categories", $this->Category-
>getTreeForSelect(1));
        $this->set("taxes", $this->Tax-
>generateList(null, "name", null, "{n}.Tax.id", "{n}.Tax.name"));
        if (empty($this->data)) {
            $this->data["Product"]["category_id"] = $category_id;
        }
        else {
            if ($this->Product->save($this->data)) {
                $this->flash('Termék
mentve!', '/admin/products/index/' . $this->data["Product"]["category_id"]);
            }
        }
    } //end admin_add()

/**
 * Function admin_edit($id=1)
 * Termék Módosítása
 * Paramléterek:
 * (int) $id: a termék azonosítója
 */
function admin_edit($id=0) {
    $this->set("Felirat", "Termék módosítása");
    $this->set("target", "/admin/products/edit/" . $id);
    $this->set("categories", $this->Category->getTreeForSelect(1));
    $this->set("taxes", $this->Tax-
>generateList(null, "name", null, "{n}.Tax.id", "{n}.Tax.name"));
    $this->set("category_id", $this->data["Product"]["category_id"]);
    if (empty($this->data))
    {
        $this->Product->id = $id;
        $this->data = $this->Product->read();
    }
    else
    {
        $this->data["Product"]["id"] = $id;
        if ($this->Product->save($this->data['Product']))
        {
            $this->flash('Adatok
mentve.', '/admin/products/index/' . $this->data["Product"]["category_id"]);
            exit();
        }
    }
    $this->render("admin_add");
} //end admin_edit()

/**
 * Function admin_show($id)
 * Termék adatlapjának megjelenítése
 * Paramléterek:
 * (int) $id: a termék azonosítója
 */
function admin_show($id=null, $category_id) {
    if($id) $this->Product->id=$id;
    $this->set("category_id", $category_id);
    $product = $this->Product->read(
array(

```

MELLÉKLET

```

        """,
        "(100+Tax.value)/100*Product.price as gross",
        "Tax.value/100*Product.price as taxvalue"));
        $this->set("product",$product);
    }//end admin_show()

/**
 * Function admin_setstatus($id)
 * Termék megjelenítése, elrejtése
 * Paraméterek:
 * (int) $id: a termék azonosítója
 */
function admin_setstatus($id=null){
    $this->Product->id=$id;
    $p = $this->Product->read("is_public",$id);
    $is_public = $p["Product"]["is_public"];
    if($is_public)$is_public=0;else $is_public=1;
    $this->Product->saveField("is_public",$is_public);
    $this->render("empty");
}//end setstatus()

/**
 * Function admin_del($id)
 * Termék törlése. Nem történik fizikai törlés, csak a deleted_at mező
 megkapja az aktuális értéket,
 * és átkerül a 2-es, azaz a kukka kategóriába.
 * Paraméterek:
 * (int) $id: a termék azonosítója
 */
function admin_del($id){
    $this->Product->id=$id;
    $c = $this->Product->field("category_id");
    $this->Product->saveField("deleted_at",date("Y-m-d H:i:s"));
    //$this->Product->saveField("category_id",2);
    $this->flash("Törlés rend-
ben!","/admin/products/index/".$c["Product"]["category_id"]);
}//end del()

/**
 * Function admin_undel($id)
 * Termék visszavétele a kukából
 * Paraméterek:
 * (int) $id: a termék azonosítója
 */
function admin_undel($id){
    $this->Product->id=$id;
    $c = $this->Product->field("category_id");
    $this->Product->saveField("deleted_at",0);
    $this->flash("Visszaállítás rend-
ben!","/admin/products/index/".$c["Product"]["category_id"]);
}//end undel()

/**
 * Function admin_undelall( )
 * Összes termék visszavétele a kukából
 * Paraméterek:
 * nincs
 */
function admin_undelall( ){

    $this->Product->execute("update products set deleted_at = 0");

```


MELLÉKLET

```
        $this->flash("Visszaállítás rendben!", "/admin/products/index/");
    } //end undel()

/**
 * Function details( $id)
 * Termék adatlapjának megtekintése a kukából
 * Paraméterek:
 * (int) $id: termék azonosítója
 */
function details($id){
    $this->layout="public";
    if($id) $this->Product->id=$id;
    $product = $this->Product->read(
        array(
            "*",
            "(100+Tax.value)/100*Product.price as gross",
            "Tax.value/100*Product.price as taxvalue"));
    $this->set("product", $product);
    $this->set("comments", $this->ProductComment-
>findAll("product_id=$id"));
    if($product){
        $this->set("route", $this->Category-
>routeToHtml($product["Product"]["category_id"], "/products/index"));
        $this->set("path", $this->ProductPhoto->path);
    }else{
        $this->flash("Érvénytelen azonosító!", "/products/");
        exit();
    }
} //end details()

}
?>
```

9.1.9. Shopping_carts_controller.php

```
<?php
class ShoppingCartsController extends ApplicationController
{
    var $name="ShoppingCart";
    var $helpers = array('Html', 'Javascript', 'Ajax', "S", "Form", "Error");
    var $uses=array("ShoppingCart", "Product", "Tax");
/**
 * Function setting($product_id=null, $quantity=null)
 * Termék hozzáadása a bevásárlókocsihoz.
 * Paramléterek:
 * $product_id: termék azonosítója
 * $quantity: mennyiség, ha amennyiség nulla, akkor töröljük is a sort
 */
    function setting($product_id=null, $quantity=0){
        Configure::write('debug', '0');
        $item = $this->ShoppingCart-
>find(array("ShoppingCart.session_id"=>session_id(), "ShoppingCart.product
_id"=>$product_id));
        $quantity = 1* $quantity;
        if($quantity<0) $quantity=0;
        $product = $this->Product-
>findAll(array("Product.id"=>$product_id));
        if($product){
            if($item){
                $this->ShoppingCart->id=$item["ShoppingCart"]["id"];
            }
        }
    }
}
```

MELLÉKLET

```

        if($quantity){
            $item["ShoppingCart"]["quantity"] = $quantity;
            unset($item["ShoppingCart"]["id"]);
            $this->ShoppingCart->save($item);
        }
        else{//nulla a mennyiség törölni kell
        {
            $this->ShoppingCart->del();
        }
    }else{
        if($quantity){
            $item["ShoppingCart"]["quantity"] = $quantity;
            $item["ShoppingCart"]["session_id"] = session_id();
            $item["ShoppingCart"]["product_id"] = $product_id;
            $this->ShoppingCart->save($item);
        }
    }
}
else $this->set("err", "Nem létező termék");
}
}
/**
 * Function refresh()
 * Visszaadja az aktuális bevásárlókosár tartalmát
 * Paramléterek:
 * nincs
 */
function refresh(){
    Configure::write('debug', '0');
    $items = $this->ShoppingCart-
>findAll(array("ShoppingCart.session_id"=>session_id()),
    null,
    null,
    null,
    null,
    2);
    if(is_array($items))
    $this->set("items", $items);
    $this->render('refresh', 'ajax');
}

function refreshtable(){
    Configure::write('debug', '0');
    $items = $this->ShoppingCart-
>findAll(array("ShoppingCart.session_id"=>session_id()),
    null,
    null,
    null,
    null,
    2);
    if(is_array($items))
    $this->set("items", $items);
    $this->render('shopping_cart_table', 'ajax');
}
/**
 * Function index()
 * Bevásárlókosár kezelése
 * Paramléterek:
 * nincs
 */
function index(){

```

```

        $this->layout="public";
        $items = $this->ShoppingCart-
>findAll(array("ShoppingCart.session_id"=>session_id()),
        null,
        null,
        null,
        null,
        2);
        if(is_array($items))
        $this->set("items",$items);
    }
}
?>

```

9.1.10. taxes_controller.php

```

<?php
class TaxesController extends ApplicationController
{
    var $name="Tax";
    var $helpers = array('Html','Javascript','Ajax');
/**
 * Function admin_index(void)
 * Áfakulcsok táblázatának megjelenítése
 * Paramléterek:
 * nincs
 */
    function admin_index()
    {
        $this->set("taxes",$this->Tax->findAll());
    }
/**
 * Function admin_add(void)
 * Új adókulcs rögzítése
 * Paramléterek:
 * nincs
 */
    function admin_add()
    {
        if (empty($this->data)){
            $this->set("Felirat","Új adókulcs létrehozása");
            $this->set("target","/admin/taxes/add");
        }
        else
        {
            if ($this->Tax->save($this->data))
            {
                $this->flash('Adókulcs mentve!','/admin/taxes/index/');
            }
        }
    }
} //end admin_add()

/**
 * Function admin_edit($id=null)
 * Adókulcs módosítása
 * Paramléterek:
 * (int) $id: áfakulcs azonosítója
 */
    function admin_edit($id=null)

```

MELLÉKLET

```
{
    if (empty($this->data)){
        $this->set("Felirat","Adókulcs módosítása");
        $this->set("target","/admin/taxes/edit/$id");
        $this->Tax->id = $id;
        $this->data = $this->Tax->read();
        $this->render("admin_add");
    }
    else
    {
        $this->data["Tax"]["id"]=$id;
        if ($this->Tax->save($this->data))
        {
            $this->flash('Adókulcs mentve!','/admin/taxes/index/');
        }
    }
} //end admin_edit()

/**
 * Function admin_del($id=null)
 * Adókulcs törlése
 * Paramléterek:
 * (int) $id: áfakulcs azonosítója
 */
function admin_del($id=null)
{
    $this->Tax->id=$id;
    $this->Tax->del();
    $this->flash('Adókulcs törölve!','/admin/taxes/index/');
}
}
?>
```

9.1.11. Users_controller.php

```
<?php
class UsersController extends ApplicationController
{
    var $name="User";
    var $uses=array("User");
    var $helpers = array('Html','Javascript','Ajax','S', "Form","Error");
/**
 * Function admin_index(void)
 * Felhasználók listájának megjelenítése
 * Paramléterek:
 * nincs
 * A Search nevű formból is kaphat paramétereket:
 * (string) data["Search"]["keyword"]: kulcsszó szerinti keresés
 */
    function admin_index(){
        $cond=array();
        if(!empty($this->data))//keresés történt
        {
            $cond[]='User.name                like                "'. $this->
>data["Search"]["keyword"].' "%';
        }
        $this->set("users",$this->User->findAll($cond));
    } //end index
/**
 * Function signup(void)
```

MELLÉKLET

```

* Felhasználó regisztrációja
* Paramléterek:
* nincs
*/
function signup(){
    $this->layout="public";//a publikus oldalelrendezést kell használni
    $this->set("target","/users/signup/");
    if (empty($this->data)){
        $this->set("Felirat","Új felhasználó rögzítése");
    }
    else{
        $this->data["User"]["registered_at"]=date("Y-m-d h:i:s");
        if ($this->User->save($this->data)){
            $this->flash('Felhasználó rögzítve!','/users/index/');
        }
    }
}
}

/**
* Function admin_add(void)
* Felhasználó rögzítése
* Paramléterek:
* nincs
*/
function admin_add(){
    $this->set("target","/admin/users/add/");
    if (empty($this->data)){
        $this->set("Felirat","Új felhasználó rögzítése");
    }
    else{
        $this->data["User"]["registered_at"]=date("Y-m-d h:i:s");
        if ($this->User->save($this->data)){
            $this->flash('Felhasználó rögzítve!','/admin/users/index/');
        }
    }
}
}

/**
* Function admin_show($id)
* Felhasználó adatlapjának megtekintése
* Paramléterek:
* (int) $id: felhasználó azonosítója
*/
function admin_show($id=null){
    $this->User->id = $id;
    $this->set("user",$this->User->read());
}

/**
* Function admin_del($id)
* Felhasználó törlése
* TODO: érdemes volna itt is a deleted_at mezőt bevezetni
* Paramléterek:
* (int) $id: felhasználó azonosítója
*/
function admin_del($id){
    $this->User->id = $id;
    $this->User->del();
    $this->flash('Felhasználó törölve!','/admin/users/index/');
}
}

```

MELLÉKLET

```

/**
 * Function admin_edit($id)
 * Felhasználó adatainak szerkesztése
 * Paramléterek:
 * (int) $id: felhasználó azonosítója
 */
function admin_edit($id=null){
    if (empty($this->data)){
        $this->User->id = $id;
        $this->data = $this->User->read();
        if(!$this->data) $this->flash('Nincs ilyen azonosítójú
felhasználó!', '/admin/users');
    }
    else{
        if ($this->User->save($this->data['User'])){
            $this->flash('Adatok mentve.', '/admin/users');
        }
    }
} //end admin_edit()

/**
 * Function login()
 * Felhasználó bejelentkezése
 * Paramléterek:
 * nincs
 */
function login()
{
    $this->set('error', false);
    if (!empty($this->data))
    {
        $someone = $this->User->findByLogin($this->
data['User']['login']);

        if(!empty($someone['User']['pwd']) && $someone['User']['pwd']
== $this->data['User']['pwd'])
        {
            $this->Session->write('User', $someone['User']);
            $this->redirect($this->data['User']['url']);
        }
        else
        {
            $this->flash("A bejelentkezés nem sikerült.", "/");
        }
    }
}

/**
 * Function logout()
 * Felhasználó kijelentkezése
 * Paramléterek:
 * nincs
 */
function logout()
{
    $this->Session->delete('User');
    $this->redirect('/');
}
}
?>

```

9.2. Modellek

9.2.1. category.php

```

<?php
class Category extends AppModel
{
    var $name = 'Category';
    var $validate;
    var $hasMany = array("Product" =>
        array('className' => 'Product',
              'conditions' => '',
              'order' => 'name ASC',
              'foreignKey' => 'category_id',
              'dependent' => true,
              'exclusive' => false,
              'finderQuery' => ''));

    function loadValidation(){
        $this->validate = array(
            'name' => array(
                'required' => array(
                    'method' => VALID_NOT_EMPTY,
                    'message' => 'Nem adott meg adatot.',
                )));
    }

    function getRoute($id){
        $result = array();
        if($id)$result[]=$id;
        $current= $this->read(null,$id);
        $parent_id = $current["Category"]["parent_id"];
        while($parent_id){
            $result[]=$parent_id;
            $current= $this->read(null,$parent_id);
            $parent_id = $current["Category"]["parent_id"];
        }
        return array_reverse($result);
    }

    function routeToHtml($id, $link){
        $route = $this->getRoute($id);
        $result = "<ul>";
        if(is_array($route))
            foreach($route as $key=>$value)
            {
                $current= $this->read("name",$value);
                $result.="<li>          &raquo;          <a
href='\".$link.\"/\".$value.\">\".$current["Category"]["name"].\"</a></li>";
            }
        $result .= "</ul>";
        return $result;
    }

    function getTreeForSelect($forbidden=false, $id = 0, $level = 0){
        $tab = str_repeat("-", $level*4);
        $items = $this->findAll("parent_id=$id", null, "name");
        $result =array();

```

MELLÉKLET

```

    foreach($items as $item)
    {
        if($forbidden && ($item["Category"]["id"]== $forbidden)) con-
        tinue;
        if(count($item["Product"]))          $szam          =          "
        (".count($item["Product"]).")";
        else $szam = "";
        $result[$item["Category"]["id"]]      =
        $tab.$item["Category"]["name"].$szam;
        if($this->findCount("parent_id      =
        ".$item["Category"]["id"]))
        $result                               += $this->getTreeForSelect($forbidden,
        $item["Category"]["id"], $level+1);
    }
    return $result;
}

function getTree($id=0, $is_public=false){
    $items = $this->getChildren($id,$is_public);
    $result =array();
    if(is_array($items))
    foreach($items as $item)
    {
        $result[$item["id"]] = $item;
        if($children = $this->getTree($item["id"],$is_public))
        $result[$item["id"]]["Children"] = $children;
    }
    return $result;
}

function getChildren($id,$is_public=false){
    $result = false;
    $cond["parent_id"]=$id;
    if($is_public)$cond["is_public"]=1;
    $items = $this->findAll($cond,null,"name");
    foreach($items as $item)
    {
        $result[$item["Category"]["id"]] = $item["Category"];
    }
    return $result;
}

function getAllChildren($id){
    $items = $this->getChildren($id);
    $result =array();
    if(is_array($items))
    foreach($items as $item)
    {
        $result[$item["id"]] = $item;
        if($children = $this->getChildren($item["id"]))
        $result+=$children;
    }
    return $result;
}

/**
 * function setPublic
 * set the publicity of all children
 */

```


MELLÉKLET

```
function setPublic ($parent, $is_public)
{
    $this->id=$parent;
    $this->saveField("is_public",$is_public);
    if($is_public==0){//we have to hide all the children
    $items = $this->getAllChildren($parent);
    foreach($items as $item)
        {
            $this->id = $item["id"];
            $this->saveField("is_public",$is_public);
        }
    }
    else{//we have to unhide all the parents
    $items = $this->getRoute($parent);
    foreach($items as $item)
        {
            $this->id = $item;
            $this->saveField("is_public",$is_public);
        }
    }
}
}
?>
```

9.2.2. delivery_method.php

```
<?php
class DeliveryMethod extends AppModel
{
    var $name = 'DeliveryMethod';
    var $belongsTo = array('Tax'=>
        array('className'=>'Tax',
            'conditions' => '',
            'order' => '',
            'foreignKey' => 'tax_id'
        )
    );
}
?>
```

9.2.3. order.php

```
<?php
class Order extends AppModel
{
    var $name = 'Order';
    var $belongsTo = array('User'=>
        array('className'=>'User',
            'conditions' => '',
            'order' => '',
            'foreignKey' => 'user_id'
        ),
        'DeliveryMethod'=>
        array('className'=>'DeliveryMethod',
            'conditions' => '',
            'order' => '',
            'foreignKey' => 'delivery_method_id'
        ),
    );
}
```

MELLÉKLET

```

        'PaymentMethod'=>
        array('className'=>'PaymentMethod',
            'conditions' => '',
            'order'      => '',
            'foreignKey' => 'payment_method_id'
        ),
    );
    var $hasMany = array("OrderedProduct"=>
        array('className'      => 'OrderedProduct',
            'conditions'      => '',
            'order'           => '',
            'limit'           => '',
            'foreignKey'      => 'order_id',
            'dependent'       => true,
            'exclusive'       => false,
            'finderQuery'     => ''
        )
    );

    var $validate;

    function loadValidation(){

        $this->
    >validate['payment_method_id']['required']['method']=VALID_NOT_EMPTY;
        $this->
    >validate['payment_method_id']['required']['message']='Válasszon fizetési
    módot!';
        $this->
    >validate['delivery_method_id']['required']['method']=VALID_NOT_EMPTY;
        $this->
    >validate['delivery_method_id']['required']['message']='Válasszon
    szállítási módot!';

        $this->
    >validate['billing_name']['required']['method']=VALID_NOT_EMPTY;
        $this->validate['billing_name']['required']['message']='Kérem,
    adja meg a nevet!';
        $this->
    >validate['billing_zipcode']['required']['method']=VALID_NOT_EMPTY;
        $this->validate['billing_zipcode']['required']['message']='Kérem,
    adja meg az irányítószámot!';
        $this->
    >validate['billing_city']['required']['method']=VALID_NOT_EMPTY;
        $this->validate['billing_city']['required']['message']='Kérem,
    adja meg a települést!';
        $this->
    >validate['billing_address']['required']['method']=VALID_NOT_EMPTY;
        $this->validate['billing_address']['required']['message']='Kérem,
    adja meg a címet!';

        $this->
    >validate['delivery_name']['required']['method']=VALID_NOT_EMPTY;
        $this->validate['delivery_name']['required']['message']='Kérem,
    adja meg a nevet!';
        $this->
    >validate['delivery_zipcode']['required']['method']=VALID_NOT_EMPTY;
        $this->
    >validate['delivery_zipcode']['required']['message']='Kérem, adja meg az
    irányítószámot!';
        $this->
    
```

```

>validate['delivery_city']['required']['method']=VALID_NOT_EMPTY;
    $this->validate['delivery_city']['required']['message']='Kérem,
adja meg a települést!';
    $this-
>validate['delivery_address']['required']['method']=VALID_NOT_EMPTY;
    $this-
>validate['delivery_address']['required']['message']='Kérem, adja meg a
címet!';
    }

}
?>

```

9.2.4. order_comments.php

```

<?php
class OrderComment extends AppModel
{
    var $name = 'OrderComments';
}
?>

```

9.2.5. ordered_product.php

```

<?php
class OrderedProduct extends AppModel
{
    var $name = 'OrderedProduct';
    var $belongsTo = array('Product'=>
        array('className'=>'Product',
              'conditions' => '',
              'order'      => '',
              'foreignKey' => 'product_id'
        ));
}
?>

```

9.2.6. page.php

```

<?php
class Page extends AppModel
{
    var $name = 'Page';
    var $validate;

    function loadValidation(){

        $this->validate['name']['required']['method']=VALID_NOT_EMPTY;
        $this->validate['name']['required']['message']='Nem adott meg
adatot.';

        $this->validate['title']['required']['method']=VALID_NOT_EMPTY;
        $this->validate['title']['required']['message']='Nem adott meg
adatot.';
    }
}
?>

```

9.2.7. payment_method.php

```
<?php
class PaymentMethod extends AppModel
{
    var $name = 'PaymentMethod';
    var $validate;
    function loadValidation(){
        $this->validate = array(
            'name' => array(
                'required' => array(
                    'method' => VALID_NOT_EMPTY,
                    'message' => 'Nem adott meg nevet.',
                )
            )
        ); //end array
    } //end function
}
?>
```

9.2.8. product.php

```
<?php
class Product extends AppModel
{
    var $name = 'Product';
    var $belongsTo = array('Category'=>
        array('className'=>'Category',
            'conditions' => '',
            'order' => '',
            'foreignKey' => 'category_id'
        ),
        'Tax'=>
            array('className'=>'Tax',
                'conditions' => '',
                'order' => '',
                'foreignKey' => 'tax_id'
            )
    );
    var $hasMany = array("Photo"=>
        array('className' => 'ProductPhoto',
            'conditions' => '',
            'order' => '',
            'limit' => '',
            'foreignKey' => 'product_id',
            'dependent' => true,
            'exclusive' => false,
            'finderQuery' => ''
        ),
        "Comment"=>array('className' => 'ProductCom-
ment',
            'conditions' => '',
            'order' => '',
            'limit' => '',
            'foreignKey' => 'product_id',
            'dependent' => true,
            'exclusive' => false,
            'finderQuery' => ''
        )
    );
};
```

```

var $validate;
function loadValidation(){
    $this->validate = array(
        'name' => array(
            'required' => array(
                'method' => VALID_NOT_EMPTY,
                'message' => 'Nem adott meg terméknevet.',
            )
        ),
        'price' => array(
            'required' => array(
                'method' => VALID_NOT_EMPTY,
                'message' => 'Nem adott meg árat.',
            ),
            'validnumber' => array(
                'method' => VALID_FLOAT,
                'message' => 'Érvénytelen karakterek a megadott adatban.',
            ),
        ),
        'unit' => array(
            'required' => array(
                'method' => VALID_NOT_EMPTY,
                'message' => 'Nem adott meg mértékegységet.',
            )
        ),
    );//end array
};//end function

}
?>

```

9.2.9. product_comment.php

```

<?php
class ProductComment extends AppModel
{
    var $name = 'ProductComment';
    var $belongsTo = array('Product'=>
        array('className'=>'Product',
            'conditions' => '',
            'order' => '',
            'foreignKey' => 'product_id'
        ),
        'User'=>
        array('className'=>'User',
            'conditions' => '',
            'order' => '',
            'foreignKey' => 'user_id'
        )
    );

    var $validate;
    function loadValidation(){
        $this->validate = array(
            'text' => array(
                'required' => array(
                    'method' => VALID_NOT_EMPTY,
                    'message' => 'Nem adott meg szöveget.')
            )
        )
    }
}

```

```

    );//end array
  }//end function
}
?>

```

9.2.10. product_photo.php

```

<?php
class ProductPhoto extends AppModel
{
    var $name = 'ProductPhoto';
    var $path= "img/photos/";

    function delete($id=null){
        if($id){
            $this->del($id);
            $file = WWW_ROOT.$this->path.$id.".jpg";
            return @unlink($file);
        }
    }
}
?>

```

9.2.11. shopping_cart.php

```

<?php
class ShoppingCart extends AppModel
{
    var $name = 'ShoppingCart';
    var $belongsTo = array('Product'=>
        array('className'=>'Product',
            'conditions' => '',
            'order'      => '',
            'foreignKey' => 'product_id'
        ));
}
?>

```

9.2.12. tax.php

```

<?php
class Tax extends AppModel
{
    var $name = 'Tax';
}
?>

```

9.2.13. user.php

```

<?php
class User extends AppModel
{
    var $name = 'User';
    var $validate;
    function loadValidation(){
        $this->validate = array(
            'login' => array(

```

MELLÉKLET

```
'required' => array(
  'method' => VALID_NOT_EMPTY,
  'message' => 'Nem adott meg felhasználónevet.',
),
'word' => array(
  'method' => VALID_WORD,
  'message' => 'A megadott felhasználónév érvénytelen karaktere-
ket tartalmaz.',
),
'unique' => array(
  'method' => VALID_UNIQUE,
  'message' => 'A megadott felhasználónév már használatban van.'
),
'length_within' => array(
  'method' => VALID_LENGTH_WITHIN,
  'message' => 'A felhasználónév hossza 6 és 50 karakter közé
kell hogy essen.',
  'parameters' => array('min' => 6, 'max' => 50)
),
),
'name' => array(
  'required' => array(
    'method' => VALID_NOT_EMPTY,
    'message' => 'Nem adta meg a nevét.',
  ),
),
'email' => array(
  'required' => array(
    'method' => VALID_NOT_EMPTY,
    'message' => 'Nem adott meg e-mail címet.',
  ),
  'email' => array(
    'method' => VALID_EMAIL,
    'message' => 'A megadott e-mail cím formátuma nem megfelelő.'
  ),
),
),
'pwd' => array(
  'required' => array(
    'method' => VALID_NOT_EMPTY,
    'message' => 'Nem adott meg jelszót.',
  ),
  'length_within' => array(
    'method' => VALID_LENGTH_WITHIN,
    'message' => 'A jelszó minimum 8, maximum 50 karakter hosszú
lehet.',
    'parameters' => array('min' => 8, 'max' => 50)
  ),
  'confirmed' => array(
    'method' => VALID_CONFIRMED,
    'message' => 'A megadott jelszavak nem egyeznek.'
  ),
),
),
'billing_city' => array(
  'required' => array(
    'method' => VALID_NOT_EMPTY,
    'message' => 'Nem adott meg városnevet.',
  ),
),
'billing_zipcode' => array(
  'required' => array(
    'method' => VALID_NOT_EMPTY,
```

```

        'message' => 'Nem adta meg az irányítószámot.',
    )),
    'billing_address' => array(
        'required' => array(
            'method' => VALID_NOT_EMPTY,
            'message' => 'Nem adta meg a címet.',
        )),
    );
}
}
?>

```

9.3. Nézetek

9.3.1. category

9.3.1.1. admin_add.thtml

```

<h1><?php echo $Felirat;?></h1>

<form method="post" action="<?php echo $html->url($target)?>">
<?php
echo $error->allMessagesFor('Category');
echo "<fieldset>";
echo "<legend>Adatok</legend>";
echo $form->generateInputDiv("Category/name", "Kategória
neve:", true, null, 35);
echo $error->messageFor('Category/name');
echo $form->generateSelectDiv('Category/parent_id', "Szülő kategória:
", $sup );
echo $form->generateCheckboxDiv('Category/is_public', "Publikus?");
echo $form->generateAreaDiv($form->generateAreaDiv('Cate-
gory/description', "Leírás:", false, null, 32, 10);
echo "</fieldset>";
?>

<div class="submitblock">
<?php
echo $html->submit('Mentés');
echo $html->input('Category/cancel', array(
    "type"=>"button",
    "onclick"=>"location.href='/admin/categories/index'",
    "value"=>"Mégsem"));
?>
</div>
</form>

```

9.3.1.2. admin_index.thtml

```

<h1>Termékkategóriák</h1>
<div class='operations'>
<h2>Tevékenységek</h2>
<?php
$text = "Új kategória hozzáadása";
echo $html->link(
    $html->image("icon/edit_add.png", array("alt"=>$text)).$text,
    "/admin/categories/add/{$parent_id}",

```


MELLÉKLET

```
        null,
        null,
        false
    );
?>

</div>
<div id="categorytable">
<?php
if(count($categories)) include "categtable.shtml";
else echo "<div class='warning'>Nincs találat!</div>";
?>
</div>
```

9.3.1.3. categtable.shtml

```
<?php

function showItem($categories)
{
    $max = count($categories);
    $i=0;
    echo "<ul class='tree'>";
    foreach($categories as $item)
    {
        $i++;
        if($max==$i)$last = ' class="last"';
        else $last = "";
        $chk = array();
        if($item["is_public"]==1)$chk["checked"] =
"/img/chk_on.jpg";
        else $chk["checked"]="img/chk_off.jpg";

        $chk["onclick"] = "onclick = \"new
Ajax.Updater('categorytable','/admin/categories/setstatus/' . $item["id"] .
')\"";
        $checkbox = "<img class='chk'
src=\"{$chk["checked"]}\" {$chk["onclick"]} alt='chk'/>";
        $operations = " <a
href='/admin/categories/edit/{ $item["id"] }' ><img
src='/img/icon/edit.png' alt='Szerkesztés'/></a>";
        $operations.=" <a
href='/admin/categories/del/{ $item["id"] }' onclick=\"if(confirm('Biztos,
hogy törölni akarja?')){new
Ajax.Updater('categorytable','/admin/categories/del/' . $item["id"] .
')\"};\" ><img src='/img/icon/cancel.png' alt='Törlés'/></a>";

        echo "<li{$last}><div{$last}>$checkbox";
        echo $item["name"] . "&nbsp;"; $operations;

if(isset($item["Children"])) showItem($item["Children"]);
        echo "</div></li>";
    }
    echo "</ul>";

}

echo "<form name='form' action='/admin/categories/index'>";
showItem($categories);
```

```
    echo "</form>";
?>
```

9.3.1.4. welcome.thtml

```
<h1>Webshop adminisztráció felület</h1>
```

9.3.2. delivery_method

9.3.2.1. admin_add.thtml

```
<h1><?php echo $Felirat;?></h1>

<form method="post" action="<?php echo $html->url($target)?>">
<?php
echo $error->allMessagesFor('DeliveryMethod');
echo "<fieldset>";
echo "<legend>Adatok</legend>";
echo $form->generateInputDiv("DeliveryMethod/name", "Szállítási mód
neve:", true, null, 35);
echo $error->messageFor('DeliveryMethod/name');
echo $form->generateInputDiv("DeliveryMethod/cost", "Díja:", true, null, 35);
echo $error->messageFor('DeliveryMethod/cost');

echo $form->generateSelectDiv('DeliveryMethod/tax_id', "Áfa: ", $taxes );
echo $error->messageFor('DeliveryMethod/tax_id');
echo "</fieldset>"
?>
<div class="submitblock">
  <?php
    echo $html->submit('Mentés');
    echo $html->input('DeliveryMethod/cancel', array(
      "type"=>"button",
      "onclick"=>"location.href='/admin/delivery_method/'",
      "value"=>"Mégsem"));
  ?>
</div>
</form>
```

9.3.2.2. admin_index.thtml

```
<?php
  //echo $javascript->link("category.js");
?>
<h1>Szállítási módok</h1>

<div class='operations'>
  <h2>Tevékenységek</h2>
  <?php
    $text = "Új szállítási mód hozzáadása";
    echo $html->link(
      $html->image("icon/edit_add.png", array("alt"=>$text)).$text,
      "/admin/delivery_methods/add",
      null,
      null,
      false
    );
  ?>
</div>
```

```

?>
</div>
<table>
<?php
if(count($methods)){
    echo $html->tableHeaders(array("Szállítási mód
neve", "Költsége", "Műveletek"));
    foreach($methods as $method){
        $m=$method["DeliveryMethod"];
        $id=$m["id"];
        $t=$method["Tax"];
        $operations = $html->link($html->image("icon/edit.png",
ar-
ray("alt"=>"Szerkesztés")), "/admin/delivery_methods/edit/" . $id, null,
null, false );
        $operations .= $html->link($html->image("icon/cancel.png",
ar-
ray("alt"=>"Törlés")), "/admin/delivery_methods/del/" . $id, null, 'Biztos,
hogy törölni akarja?', false );

        $gross = $m["cost"]*(100+$t["value"])/100;
        $net = $m["cost"];

        echo $html->tableCells(
            array($m["name"],
                $s->toMoney($gross) . "<br/>" .
                $s->toMoney($net) . "+" . $t["name"],
                $operations));
    }
}
else echo "<div class='warning'>Nincs találat!</div>";
?>
</table>

```

9.3.3. elements

9.3.3.1. categories.thtml

```

<?php
$categories = $this->requestAction("/categories/getpubliclist");
function showItem($categories)
{
    $max = count($categories);
    $i=0;
    echo "<ul>";
    foreach($categories as $item)
    {
        echo "<li>";
        echo
href="/products/index/" . $item["id"] . "'>" . $item["name"] . "</a>";
        if(isset($item["Children"])){
            showItem($item["Children"]);
        }
        echo "</li>";
    }
    echo "</ul>";
}
}

```

MELLÉKLET

```
    echo "<div id='categories'>";
    showItem($categories);
    echo "</div>";
?>
<script type="text/javascript">
//
if (window.attachEvent) sfHover('categories');
//]]&gt;
&lt;/script&gt;</pre></div><div data-bbox="113 217 281 234" data-label="Section-Header"><h3>9.3.3.2. login.shtml</h3></div><div data-bbox="114 239 862 628" data-label="Text"><pre>&lt;form action="&lt;?php echo $html-&gt;url('/users/login'); ?&gt;" method="post"
id='login'&gt;
&lt;?php
    if($user = $session-&gt;read("User")){
        echo "Belépett felhasználó: &lt;b&gt;". $user["name"]. "&lt;/b&gt;&lt;br/&gt;". $html-
&gt;link("Kijelentkezés", "/users/logout");

        }else{ ?&gt;

&lt;fieldset&gt;
    &lt;legend&gt;Bejelentkezés&lt;/legend&gt;
&lt;div&gt;
    &lt;label for="username"&gt;Felhasználónév:&lt;/label&gt;
    &lt;?php echo $html-&gt;input('User/login', array('size' =&gt; 20)); ?&gt;
&lt;/div&gt;
&lt;div&gt;
    &lt;label for="password"&gt;Jelszó:&lt;/label&gt;
    &lt;?php echo $html-&gt;password('User/pwd', array('size' =&gt; 20));
    echo $html-
&gt;hidden("User/url", array("value"=&gt;$_SERVER["REQUEST_URI"]));
    ?&gt;
&lt;/div&gt;
&lt;div&gt;
    &lt;?php echo $html-&gt;submit('Login');
    echo $html-&gt;link("Regisztráció", "/users/signup");
    ?&gt;
&lt;/div&gt;
&lt;/fieldset&gt;

    &lt;? } ?&gt;
&lt;/form&gt;</pre></div><div data-bbox="113 643 287 659" data-label="Section-Header"><h3>9.3.3.3. menu.shtml</h3></div><div data-bbox="114 665 806 792" data-label="Text"><pre>&lt;ul id='mainmenu'&gt;
    &lt;li&gt;&lt;a href='/admin/categories/'&gt;Kategóriák&lt;/a&gt;&lt;/li&gt;
    &lt;li&gt;&lt;a href='/admin/products/'&gt;Termékek&lt;/a&gt;&lt;/li&gt;
    &lt;li&gt;&lt;a href='/admin/orders/'&gt;Rendelések&lt;/a&gt;&lt;/li&gt;
    &lt;li&gt;&lt;a href='/admin/users/'&gt;Felhasználók&lt;/a&gt;&lt;/li&gt;
    &lt;li&gt;&lt;a href='/admin/taxes/'&gt;Áfakulcs&lt;/a&gt;&lt;/li&gt;
    &lt;li&gt;&lt;a href='/admin/delivery_methods/'&gt;Szállítási módok&lt;/a&gt;&lt;/li&gt;
    &lt;li&gt;&lt;a href='/admin/payment_methods/'&gt;Fizetési módok&lt;/a&gt;&lt;/li&gt;
    &lt;li&gt;&lt;a href='/admin/pages/'&gt;Dokumentumok&lt;/a&gt;&lt;/li&gt;
&lt;/ul&gt;</pre></div><div data-bbox="473 861 500 878" data-label="Page-Footer"><p>79</p></div>
```

9.3.3.4. public_menu.shtml

```
<ul id='mainmenu'>
  <li><a href='/products/'>Webshop</a></li>
  <li><a href='/pages/index/1'>Vásárlási tájékoztató</a></li>
  <? if($session->read("User")){
    ?>
  <li><a href='/orders/make'>Pénztár</a></li>
  <li><a href='/orders/myorders'>Rendelések</a></li>
  <?
  }?>
</ul>
```

9.3.3.5. shopping_cart.shtml

```
<div id='shopping_cart'>
<h1>Kosár</h1>
<?php
$item = $this->requestAction("/shopping_carts/shortlist");
if(is_array($item)){
echo "<table>";
$sum=0;
foreach($item as $item){
  $cell[0] = $item["Product"]["name"];
  $cell[1] = $item["ShoppingCart"]["quantity"];
  $cell[2] = $item["Product"]["unit"];
  $cell[3] = $item["Product"]["price"]*$item["ShoppingCart"]["quantity"]*(100+$item["Product"]["Tax"]["value"])/100;
  $sum += $cell[3];
  $cell[2] = $s->toMoney(round($cell[3]));
  echo "<tr>";
  echo "<td class='item'>".$cell[0]."- ".$cell[1]."</td>";
  echo "<td class='price'>".$cell[3]."</td>";
  echo "</tr>";
}
echo "<tr><td>Összesen:</td><td class='price'>".$s->toMoney($sum)."</td></tr>";
echo "</table>";
} else echo "A bevásárlókosár üres.";
?>
</div>
```

9.3.4. layout

9.3.4.1. default.shtml

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Webshop Adminisztrációs felület - Barkóczy Roland szakdolgozata</title>
<link rel="shortcut icon" href="/favicon.ico" type="image/x-icon" />
<link rel="stylesheet" type="text/css" href="/css/cake.generic.css" />
<link rel="stylesheet" type="text/css" href="/css/tree.css" />
<link rel="stylesheet" type="text/css" href="/css/layout.css" />
<link rel="stylesheet" type="text/css" href="/css/inplace.css" />
```

MELLÉKLET

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<script type="text/javascript" src="/js/lib/prototype.js"></script>
<script type="text/javascript" src="/js/functions.js"></script>
<script type="text/javascript" src="/js/src/scriptaculous.js"></script>
</head>
<body>
  <div id="level0">
    <div id='header'>
      <? echo $this->renderElement('login');?>
      <h1>Welcome</h1>

    </div>
    <? echo $this->renderElement('menu');?>
    <div id="content">
      <?php echo $content_for_layout ?>
    </div>
    <div id="footer">
      2007 - Webshop készítése cakePHP keretrendszer segítségével <br/>
      Barkóczy Roland - ST7AGX
    </div>
  </div>
  <div id="ajax_request"><img src='/img/icon/ajax-loader.gif' alt="" />
  Kommunikáció a szerverrel</div>
  <script type="text/javascript">
  //globalhandlers regisztrálása
  Ajax.responders.register(myGlobalHandlers);
  Event.observe(document,
    'mousemove',
    function(event)
    {
      Element.setStyle('ajax_request',
        {
          left: (Event.pointerX(event)+15)+'px',
          top: (Event.pointerY(event)+15)+'px'
        });
    });
  </script>
</body>
</html>
```

9.3.4.2. public.thtml

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Webshop - Barkóczy Roland szakdolgozata</title>
<link rel="shortcut icon" href="/favicon.ico" type="image/x-icon" />
<link rel="stylesheet" type="text/css" href="/css/cake.generic.css" />
<link rel="stylesheet" type="text/css" href="/css/tree.css" />
<link rel="stylesheet" type="text/css" href="/css/public_layout.css" />
<link rel="stylesheet" type="text/css" href="/css/inplace.css" />
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<script type="text/javascript" src="/js/lib/prototype.js"></script>
<script type="text/javascript" src="/js/functions.js"></script>
<script type="text/javascript" src="/js/src/scriptaculous.js"></script>
</head>
<body>
```

MELLÉKLET

```
<div id="level0">
  <div id='header'>
    <? echo $this->renderElement('login');?>
    <h1>Webshop</h1>

  </div>
  <? echo $this->renderElement('public_menu');

?>
  <div id="left">
    <? echo $this->renderElement('categories');?>
    <div id='shopping_cart'>
      <? $this->requestAction("/shopping_carts/refresh");?>
    </div>
  </div>
  <div id="content">
    <?php echo $content_for_layout ?>
  </div>
  <div id="footer">
    2007 - Webshop készítése cakePHP keretrendszer segítségével <br/>
    Barkóczi Roland - ST7AGX
  </div>
</div>
<div id="ajax_request"><img src='/img/icon/ajax-loader.gif' alt=""/>
Kommunikáció a szerverrel</div>
<script type="text/javascript">
//globalhandlers regisztralasa
Ajax.Responders.register(myGlobalHandlers);
</script>
</body>
</html>
```

9.3.4.3. slideshow.thtml

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Proba</title>
<link rel="icon" href="/favicon.ico" type="image/x-icon" />
<link rel="shortcut icon" href="/favicon.ico" type="image/x-icon" />
<link rel="stylesheet" type="text/css" href="/css/cake.generic.css" />
<link rel="stylesheet" type="text/css" href="/css/slideshow.css" />
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<script type="text/javascript" src="/js/lib/prototype.js"></script>
<script type="text/javascript" src="/js/src/scriptaculous.js"></script>
</head>
<body>
  <div id="level0">
    <div id="content">
      <?php echo $content_for_layout ?>
    </div>
  </div>
</body>
</html>
```

9.3.5. order

9.3.5.1. account.thtml

```

<?php
echo "<table style='width:auto'>";
echo $html->tableHeaders(array("Tétel", "Mennyiség", "Áfa", "Nettó
egységár", "Ár"));

if(count($items)){
$s_gross=0;
$s_net = 0;
$s_tax = 0;
foreach($items as $item){

    $gross =
$item["Product"]["price"]*$item["ShoppingCart"]["quantity"]*(100+$item["P
roduct"]["Tax"]["value"])/100;
    $gross_e =
$item["Product"]["price"]*(100+$item["Product"]["Tax"]["value"])/100;
    $s_gross += $gross;
    $net = $item["Product"]["price"]*$item["ShoppingCart"]["quantity"];
    $net_e = $item["Product"]["price"];
    $s_net += $net;
    $tax =
$item["Product"]["price"]*$item["ShoppingCart"]["quantity"]*(($item["Produ
ct"]["Tax"]["value"])/100;
    $tax_e =
$item["Product"]["price"]*($item["Product"]["Tax"]["value"])/100;
    $s_tax += $tax;

    $cell[0] = $item["Product"]["name"];
    $cell[1] = $item["ShoppingCart"]["quantity"]."
".$item["Product"]["unit"];
    $cell[2] = $item["Product"]["Tax"]["name"];
    $cell[3] = "<div class='gross'>".$s-
>toMoney(round($gross_e))."</div>".$s->toMoney(round($net_e))."+".$s-
>toMoney(round($tax_e));

    $cell[4] = "<div class='gross'>".$s-
>toMoney(round($gross))."</div>".$s->toMoney(round($net))."+".$s-
>toMoney(round($tax));

    echo "<tr>";
    echo "<td class='item'>".$cell[0]."</td>";
    echo "<td class='quantity'>".$cell[1]."</td>";
    echo "<td class='quantity'>".$cell[2]."</td>";
    echo "<td class='price'>".$cell[3]."</td>";
    echo "<td class='price'>".$cell[4]."</td>";
    echo "</tr>";
}

if(isset($selectedDM) && $selectedDM["DeliveryMethod"]["cost"]!=0 ){//ha
vin kiválasztva delivery method és az nem nulla

echo "<tr>";
    $gross = $select-
edDM["DeliveryMethod"]["cost"]*(100+$selectedDM["Tax"]["value"])/100;
    $net = $selectedDM["DeliveryMethod"]["cost"];

```


MELLÉKLET

```
$tax=
edDM["DeliveryMethod"]["cost"]*($selectedDM["Tax"]["value"])/100;
$s_gross += $gross;
$s_net += $net;
$s_tax += $tax;

$cell[3] = $cell[4] = "<div class='gross'>".$s-
>toMoney(round($gross))."</div>".$s->toMoney(round($net))."+".$s-
>toMoney(round($tax));

echo "<td class='item'>Szállítás
(".$selectedDM["DeliveryMethod"]["name"]."):</td>";
echo "<td class='quantity'></td>";
echo "<td class='quantity'>".$selectedDM["Tax"]["name"]."</td>";
echo "<td class='price'>".$cell[3]."</td>";
echo "<td class='price'>".$cell[4]."</td>";
echo "</tr>";
}

echo "<tr><td colspan=4>Összesen:</td><td class='price'><div
class='gross'>".$s->toMoney(round($s_gross))."</div>
".$s->toMoney(round($s_net))." + ".$s->toMoney(round($s_tax))." Áfa
</td></tr>";
}

echo "</table>";
?>
```

9.3.5.2. admin_details.shtml

```
<h1>Rendelés adatai</h1>
<?php
if(is_array($order)){
    $o = $order["Order"];
    $p = $order["OrderedProduct"];
    $pm = $order["PaymentMethod"];
    $dm = $order["DeliveryMethod"];

    $confirmed_at = $o["confirmed_at"]!=0?$o["confirmed_at"]:"<a
href='/orders/confirm/'. $o["id"]."'>Visszaigazolás</a>";
    $delivered_at = $o["delivered_at"]!=0?$o["delivered_at"]:"<a
href='/orders/deliver/'. $o["id"]."'>Kiszállítás</a>";
    $paid_on = $o["paid_on"]!=0?$o["paid_on"]:"<a
href='/orders/pay/'. $o["id"]."'>Fizetés</a>";

    echo "<table>";
    echo $html->tableCells(array("Rendelés azonosítója:", $o["id"]));
    echo $html->tableCells(array("Számlázási cím:", $o["billing_name"].",
".$o["billing_zipcode"]." ".$o["billing_city"].",
".$o["billing_address"]));
    echo $html->tableCells(array("Szállítási cím:", $o["delivery_name"].",
".$o["delivery_zipcode"]." ".$o["delivery_city"].",
".$o["delivery_address"]));
    echo $html->tableCells(array("Szállítási mód:", $dm["name"]));
    echo $html->tableCells(array("Fizetési mód:", $pm["name"]));

    echo $html->tableCells(array("Rendelés dátuma:", $o["created_at"]));
    echo $html->tableCells(array("Visszaigazolás
```

MELLÉKLET

```

dátuma:", $confirmed_at));
    echo $html->tableCells(array("Szállítás dátuma:", $delivered_at));
    echo $html->tableCells(array("Fizetve:", $paid_on));

    echo "</table>";

if(count($order["OrderedProduct"])){
    $s_gross=0;
    $s_net = 0;
    $s_tax = 0;
    echo "<table style='width:auto'>";
    echo $html->tableHeaders(array("Tétel", "Menny-
iség", "Áfa", "Egységár", "Ár"));
    foreach($order["OrderedProduct"] as $item){

        $gross =
$item["product_price"]*$item["quantity"]*(100+$item["product_tax])/100;
        $gross_e = $item["product_price"]*(100+$item["product_tax])/100;
        $s_gross += $gross;
        $net = $item["product_price"]*$item["quantity"];
        $net_e = $item["product_price"];
        $s_net += $net;
        $tax =
$item["product_price"]*$item["quantity"]*($item["product_tax])/100;
        $tax_e = $item["product_price"]*($item["product_tax])/100;
        $s_tax += $tax;

        $cell[0] = $item["product_name"];
        $cell[1] = $item["quantity"];
        $cell[2] = $item["product_tax"]."%";
        $cell[3] = "<div class='gross'>".$s-
>toMoney(round($gross_e))."</div>".$s->toMoney(round($net_e))."+".$s-
>toMoney(round($tax_e));

        $cell[4] = "<div class='gross'>".$s-
>toMoney(round($gross))."</div>".$s->toMoney(round($net))."+".$s-
>toMoney(round($tax));

        echo "<tr>";
        echo "<td class='item'>".$cell[0]."</td>";
        echo "<td class='quantity'>".$cell[1]."</td>";
        echo "<td class='quantity'>".$cell[2]."</td>";
        echo "<td class='price'>".$cell[3]."</td>";
        echo "<td class='price'>".$cell[4]."</td>";
        echo "</tr>";
    }
    echo "<tr><td colspan=4>Összesen:</td><td class='price'><div
class='gross'>".$s->toMoney(round($s_gross))."</div>
".$s->toMoney(round($s_net))." + ".$s->toMoney(round($s_tax))." Áfa
</td></tr>";
    echo "</table>";
}
}
else {
    echo "<div class='warning'>Nincsenek ilyen azonosítójú
rendelés!</div>";
}
echo "<div class='center'><a

```

```
href='/admin/orders/index/'. $ordersadmin.'">Vissza</a></div>";
?>
```

9.3.5.3. Admin_index.thtml

```
<h1>Beérkezett rendelések</h1>
<div class='operations'>
<a href='/admin/orders/index/all'>Mind</a> |
<a href='/admin/orders/index/new'>Újak</a> |
<a href='/admin/orders/index/delivery'>Szállításra várók</a> |
<a href='/admin/orders/index/payment'>Fizetésre várók</a> |
<a href='/admin/orders/index/closed'>Lezárt rendelések</a>
</div>
<h2><?php echo $felirat?></h2>
<?php
if(is_array($orders)&&count($orders)){
    echo "Találatok száma: ".count($orders);

    echo "<table style='width:auto'>";
    echo $html->tableHeaders(array("Azonosító", "Megrendelő", "Rendelés
dátum", "Visszaigazolás dátuma", "Szállítás dátuma", "Fizetve", ""));
    foreach($orders as $order){
        $o = $order["Order"];
        $p = $order["OrderedProduct"];
        $pm = $order["PaymentMethod"];
        $dm = $order["DeliveryMethod"];
        $confirmed_at = $o["confirmed_at"]!=0?$o["confirmed_at"]:"<a
href='/admin/orders/confirm/'. $o["id"]."'>Visszaigazolás</a>";
        $delivered_at = $o["delivered_at"]!=0?$o["delivered_at"]:"<a
href='/admin/orders/deliver/'. $o["id"]."'>Kiszállítás</a>";
        $paid_on = $o["paid_on"]!=0?$o["paid_on"]:"<a
href='/admin/orders/pay/'. $o["id"]."'>Fizetés</a>";

        $cell[0]=$o["id"];
        $cell[1]=$o["billing_name"].", ". $o["billing_zipcode"]."
". $o["billing_city"].", ". $o["billing_address"];
        $cell[3]=$o["created_at"];
        $cell[4]=$confirmed_at;
        $cell[5]=$delivered_at;
        $cell[6]=$paid_on;
        $cell[7]="<a
href='/admin/orders/details/'. $o["id"]."'>Részletek</a>";
        echo $html->tableCells($cell);
    }

    echo "</table>";

} else {
    echo "<div class='warning'>Nincs találat!</div>";
}
?>
```

9.3.5.4. Details.thtml

```
<h1>Rendelés adatai</h1>
<?php
if(is_array($order)){
    $o = $order["Order"];
    $p = $order["OrderedProduct"];
    $pm = $order["PaymentMethod"];
}
```

MELLÉKLET

```

$dm = $order["DeliveryMethod"];
$confirmed_at = $o["confirmed_at"]!=0?$o["confirmed_at"]:"-";
$delivered_at = $o["delivered_at"]!=0?$o["delivered_at"]:"-";
$paid_on = $o["paid_on"]!=0?$o["paid_on"]:"-";
echo "<table>";
echo $html->tableCells(array("Rendelés azonosítója:", $o["id"]));
echo $html->tableCells(array("Számlázási cím:", $o["billing_name"].",
". $o["billing_zipcode"]." " $o["billing_city"].",
". $o["billing_address"]));
echo $html->tableCells(array("Szállítási cím:", $o["delivery_name"].",
". $o["delivery_zipcode"]." " $o["delivery_city"].",
". $o["delivery_address"]));
echo $html->tableCells(array("Szállítási mód:", $dm["name"]));
echo $html->tableCells(array("Fizetési mód:", $pm["name"]));

echo $html->tableCells(array("Rendelés dátuma:", $o["created_at"]));
echo $html->tableCells(array("Visszaigazolás
dátuma:", $confirmed_at));
echo $html->tableCells(array("Szállítás dátuma:", $delivered_at));
echo $html->tableCells(array("Fizetve:", $paid_on));

echo "</table>";

if(count($order["OrderedProduct"])){
    $s_gross=0;
    $s_net = 0;
    $s_tax = 0;
    echo "<table style='width:auto'>";
    echo $html->tableHeaders(array("Tétel", "Menny-
iség", "Áfa", "Egységár", "Ár"));
    foreach($order["OrderedProduct"] as $item){
        $gross =
$item["product_price"]*$item["quantity"]*(100+$item["product_tax"])/100;
        $gross_e = $item["product_price"]*(100+$item["product_tax"])/100;
        $s_gross += $gross;
        $net = $item["product_price"]*$item["quantity"];
        $net_e = $item["product_price"];
        $s_net += $net;
        $tax =
$item["product_price"]*$item["quantity"]*($item["product_tax"])/100;
        $tax_e = $item["product_price"]*($item["product_tax"])/100;
        $s_tax += $tax;

        $cell[0] = $item["product_name"];
        $cell[1] = $item["quantity"];
        $cell[2] = $item["product_tax"]."%";
        $cell[3] = "<div class='gross'>".$s-
>toMoney(round($gross_e))."</div>".$s->toMoney(round($net_e))."+".$s-
>toMoney(round($tax_e));

        $cell[4] = "<div class='gross'>".$s-
>toMoney(round($gross))."</div>".$s->toMoney(round($net))."+".$s-
>toMoney(round($tax));

        echo "<tr>";
        echo "<td class='item'>".$cell[0]."</td>";
        echo "<td class='quantity'>".$cell[1]."</td>";
        echo "<td class='quantity'>".$cell[2]."</td>";

```

MELLÉKLET

```
        echo "<td class='price'>". $cell[3]. "</td>";
        echo "<td class='price'>". $cell[4]. "</td>";
        echo "</tr>";
    }

    if($dm["cost"]!=0 ){//ha vin kiválasztva delivery method és az nem
nulla
echo "<tr>";
    $gross = $dm["cost"]*(100+$taxes[$dm["tax_id"]])/100;
    $net = $dm["cost"];
    $tax= $dm["cost"]*($taxes[$dm["tax_id"]])/100;
    $s_gross += $gross;
    $s_net += $net;
    $s_tax += $tax;

    $cell[3] = $cell[4] = "<div class='gross'>". $s-
>toMoney(round($gross)). "</div>". $s->toMoney(round($net)). "+" . $s-
>toMoney(round($tax));

    echo "<td class='item'>Szállítás (". $dm["name"]. ")</td>";
    echo "<td class='quantity'> </td>";
    echo "<td class='quantity'>". $dm["Tax"]["name"]. "</td>";
    echo "<td class='price'>". $cell[3]. "</td>";
    echo "<td class='price'>". $cell[4]. "</td>";
    echo "</tr>";
}

    echo "<tr><td colspan=4>Összesen:</td><td class='price'><div
class='gross'>". $s->toMoney(round($s_gross)). "</div>
". $s->toMoney(round($s_net)). " + ". $s->toMoney(round($s_tax)). " Áfa
</td></tr>";
    echo "</table>";
}
}
else {
    echo "<div class='warning'>Nincsenek ilyen azonosítójú
rendelés!</div>";
}
echo "<div class='center'><a href='/orders/myorders'>Vissza</a></div>";
?>
```

9.3.5.5. Make.html

```
<h1>Termékek rendelése</h1>
<form method="post" action="/orders/make" style='width: 600px'>
<?
echo "<fieldset style='width: 588px'>";
echo "<legend>Vevő adatai</legend>";
echo $error->allMessagesFor('Order');
echo "<fieldset style='width: 280px; float: left'>";
echo "<legend>Számlázási cím</legend>";
echo $error->messageFor('Order/billing_name');
echo $form->generateInputDiv ("Order/billing_name", "Név:", true, "ok", 35);
echo $error->messageFor('Order/billing_zipcode');
echo $form->generateInputDiv ("Order/billing_zipcode", "Irányítószám:", true, "ok", 4);
echo $error->messageFor('Order/billing_city');
```

MELLÉKLET

```

echo          $form->generateInputDiv          ("Or-
der/billing_city", "Település:", true, "ok", 35);
echo $error->messageFor('Order/billing_address');
echo          $form->generateInputDiv          ("Order/billing_address", "Utca,
házszám:", true, "ok", 35);
echo "</fieldset>";

echo "<fieldset style='width: 280px; float: right'>";
echo "<legend>Szállítási cím</legend>";
echo $error->messageFor('Order/delivery_name');
echo $form->generateInputDiv ("Order/delivery_name", "Név:", true, "ok", 35);
echo $error->messageFor('Order/delivery_zipcode');
echo          $form->generateInputDiv          ("Or-
der/delivery_zipcode", "Irányítószám:", true, "ok", 4);
echo $error->messageFor('Order/delivery_city');
echo          $form->generateInputDiv          ("Or-
der/delivery_city", "Település:", true, "ok", 35);
echo $error->messageFor('Order/delivery_address');
echo          $form->generateInputDiv          ("Order/delivery_address", "Utca,
házszám:", true, "ok", 35);
echo "</fieldset>";
echo "</fieldset>";
echo "<fieldset style='width: 280px; float: left'>";
echo "<legend>Fizetés</legend>";
echo $error->messageFor('Order/payment_method_id');
echo $html->hidden("Order/payment_method_id", "");
    foreach($payment_methods as $payment_method){
        $pm = $payment_method["PaymentMethod"];
        echo          "<div>". $html->radio("Order/payment_method_id",          ar-
ray($pm["id"]=>$pm["name"])). "</div>";
    }
echo "</fieldset>";

echo "<fieldset style='width: 280px; float: right'>";
echo "<legend>Szállítás</legend>";
echo $html->hidden("Order/delivery_method_id", "");
echo $error->messageFor('Order/delivery_method_id');
    foreach($delivery_methods as $delivery_method){

        $dm = $delivery_method["DeliveryMethod"];
        $t = $delivery_method["Tax"];
        $name = $dm["name"]. " - Költség: ". $s-
>toMoney($dm["cost"]). "+" . $t["name"];
        echo          "<div>". $html->radio("Order/delivery_method_id",          ar-
ray($dm["id"]=>$name), null,
ray("onclick"=>"setDM(this.value)")). "</div>";
    }
echo "</fieldset>";

echo "<fieldset style='width: 588px; clear: both'>";
echo "<legend>Tételek</legend>";
echo "<div id='tetelek'>";
include "account.thtml";
echo "</div>";
echo "</fieldset>";
?>

<div class="submitblock">
    <?php
        echo $html->submit('MEGRENDELÉS');

```

```

?>
</div>
</form>
<script type="text/javascript">
//

function setDM(dm) {
    new Ajax.Updater('tetelek',
        '/orders/setDM/'+dm,
        {
            method: 'post'
        });
}

//]]&gt;
&lt;/script&gt;
</pre>
</div>
<div data-bbox="114 316 340 334" data-label="Section-Header">
<h3>9.3.5.6. Make_login.shtml</h3>
</div>
<div data-bbox="114 339 862 466" data-label="Text">
<pre>
&lt;h1&gt;Termékek rendelése&lt;/h1&gt;
&lt;p&gt;
Ha Ön még nem rendelkezik bejelentkezési azonosítóval, kérjük regisztrál-
jon a következő linkre kattintva:
&lt;a href='/users/signup'&gt;REGISZTRÁCIÓ&lt;/a&gt;
&lt;/p&gt;&lt;p&gt;
Amennyiben Ön már regisztrált ügyfelünk, kérjük jelentkezzen be webáru-
házunkba a címsorban lévő BEJELENTKEZÉS blokk mezőinek kitöltésével.
&lt;/p&gt;
</pre>
</div>
<div data-bbox="114 480 323 498" data-label="Section-Header">
<h3>9.3.5.7. Myorders.shtml</h3>
</div>
<div data-bbox="114 504 862 843" data-label="Text">
<pre>
&lt;h1&gt;Rendelések nyomonkövetése&lt;/h1&gt;
&lt;?php
if(is_array($orders) &amp;&amp; count($orders)) {
    echo "&lt;table&gt;";
    echo
    $html-&gt;tableHead-
ers(array("Azonosító", "Rendelés", "Visszaigazolás", "Kiszállítva", "Fizetve"
, ""));
    foreach($orders as $order) {
        $o = $order["Order"];
        $confirmed_at = $o["confirmed_at"]!=0?$o["confirmed_at"]:"-";
        $delivered_at = $o["delivered_at"]!=0?$o["delivered_at"]:"-";
        $paid_on = $o["paid_on"]!=0?$o["paid_on"]:"-";

        echo $html-&gt;tableCells(array(
            $o["id"],
            $o["created_at"],
            $o["confirmed_at"]?$o["confirmed_at"]:"- ",
            $o["delivered_at"]?$o["delivered_at"]:"- ",
            $o["paid_on"]?$o["paid_on"]:"- ",
            $html-&gt;link("részletek", "/orders/details/".$o["id"]))
        );
    }
    echo "&lt;/table&gt;";
}
else {
</pre>
</div>
<div data-bbox="473 862 500 879" data-label="Page-Footer">90</div>
```

```

    echo "<div class='warning'>Nincsenek rendelések!</div>";
}
?>

```

9.3.6. page

9.3.6.1. 404.shtml

```
<h1>A keresett oldal nem található</h1>
```

9.3.6.2. Admin_add.shtml

```

<h1><?php echo $Felirat;?></h1>

<form method="post" action="<?php echo $html->url($target)?>">
<?php
echo $error->allMessagesFor('Page');
echo "<fieldset>";
echo "<legend>Adatok</legend>";
echo $form->generateInputDiv("Page/name", "Dokumentum
neve:", true, null, 35);
echo $error->messageFor('Page/name');
echo $form->generateInputDiv("Page/title", "Címe:", true, null, 35);
echo $error->messageFor('Page/title');

echo $form->generateAreaDiv("Page/content", "Tartalom:", true, null, 60, 20);
echo $error->messageFor('Page/content');
echo "</fieldset>"
?>
<div class="submitblock">
    <?php
        echo $html->submit('Mentés');
        echo $html->input('Page/cancel', array(
            "type"=>"button",
            "onclick"=>"location.href='/admin/pages/'",
            "value"=>"Mégsem"));
    ?>
</div>
</form>

```

9.3.6.3. Admin_index.shtml

```

<h1>Dokumentumok</h1>

<div class='operations'>
    <h2>Tevékenységek</h2>
    <?php
        $text = "Új dokumentum hozzáadása";
        echo $html->link(
            $html->image("icon/edit_add.png", array("alt"=>$text)).$text,
            "/admin/pages/add",
            null,
            null,
            false
        );
    ?>

```



```

</div>
<table>
<?php
if(count($pages)){
    echo $html->tableHeaders(array("Dokumentum
címe", "Módosítva", "Műveletek"));
    foreach($pages as $page){
        $m=$page["Page"];
        $id=$m["id"];
        $operations = $html->link($html->image("icon/edit.png",
            ar-
ray("alt"=>"Szerkesztés")), "/admin/pages/edit/" . $id, null, null, false );
        $operations .= $html->link($html->image("icon/cancel.png",
            ar-
ray("alt"=>"Törlés")), "/admin/pages/del/" . $id, null, 'Biztos, hogy törölni
akarja?', false );

        echo $html->tableCells(
            array($m["name"],
                $m["modified_on"],
                $operations));
    }
}
else echo "<div class='warning'>Nincs találat!</div>";
?>
</table>

```

9.3.6.4. Index.shtml

```

<?php
echo $page["Page"]["content"];
?>

```

9.3.7. Payment_method

9.3.8. admin_add.shtml

```

<h1><?php echo $Felirat;?></h1>

<form method="post" action="<?php echo $html->url($target)?>">
<?php
echo $error->allMessagesFor('PaymentMethod');
echo "<fieldset>";
echo "<legend>Adatok</legend>";
echo $form->generateInputDiv("PaymentMethod/name", "Fizetési mód
neve:", true, null, 35);
echo $error->messageFor('PaymentMethod/name');
echo "</fieldset>";
?>
<div class="submitblock">
    <?php
    echo $html->submit('Mentés');
    echo $html->input('PaymentMethod/cancel', array(
        "type"=>"button",
        "onclick"=>"location.href='/admin/payment_methods/'",
        "value"=>"Mégsem"));

```

```

    ?>
</div>
</form>

```

9.3.8.1. admin_index.shtml

```

<?php
    //echo $javascript->link("category.js");
    ?>
<h1>Szállítási módok</h1>

<div class='operations'>
    <h2>Tevékenységek</h2>
    <?php
        $text = "Új fizetési mód hozzáadása";
        echo $html->link(
            $html->image("icon/edit_add.png",array("alt"=>$text)).$text,
            "/admin/payment_methods/add",
            null,
            null,
            false
        );
    ?>

</div>
<table>
<?php
if(count($methods)){
    echo $html->tableHeaders(array("Fizetési mód neve", "Műveletek"));
    foreach($methods as $method){
        $m=$method["PaymentMethod"];
        $id=$m["id"];
        $operations = $html->link($html->image("icon/edit.png",
            ar-
ray("alt"=>"Szerkesztés")), "/admin/payment_methods/edit/" . $id, null, null,
false );
        $operations .= $html->link($html->image("icon/cancel.png",
            ar-
ray("alt"=>"Törlés")), "/admin/payment_methods/del/" . $id, null,      'Biztos,
hogy törölni akarja?', false );

        echo $html->tableCells(
            array($m["name"],
                $operations));
    }
}
else echo "<div class='warning'>Nincs találat!</div>";
?>
</table>

```

9.3.9. product

9.3.9.1. admin_add.shtml

```

<h1><?php echo $Felirat;?></h1>

```

MELLÉKLET

```
<form method="post" action="<?php echo $html->url($target)?>">
<?php
echo $error->allMessagesFor('Product');
echo "<fieldset>";
echo "<legend>Adatok</legend>";
echo $form->generateInputDiv("Product/name", "Termék neve:", true, null, 35);
echo $error->messageFor('Product/name');
echo $form->generateSelectDiv('Product/category_id', "Kategória:",
", $categories );
echo $error->messageFor('Product/category_id');
echo $form->generateInputDiv("Product/price", "Termék ára:", true, null, 35);
echo $error->messageFor('Product/price');

echo $form->generateSelectDiv('Product/tax_id', "Áfakulcs:", $taxes );
echo $error->messageFor('Product/tax_id');
echo $form->generateInputDiv("Product/unit", "Mértékegység:", true, null, 5);
echo $error->messageFor('Product/unit');

echo $form->generateCheckboxDiv('Product/is_public', "Publikus?");
echo $form->generateAreaDiv('Product/description', "Leírás:", false, null, 32, 10);

echo "</fieldset>"
?>
<div class="submitblock">
  <?php
  echo $html->submit('Mentés');
  echo $html->input('Category/cancel', array(
    "type"=>"button",
    "onclick"=>"location.href='/admin/products/lista/'. $category_id.",
    "value"=>"Mégsem"));
  ?>
</div>
</form>
```

9.3.9.2. admin_index.shtml

```
<h1>Termékek</h1>
<div class='operations'>
  <h2>Tevékenységek</h2>
  <form id='search' method='post' action='/admin/products/index'>
<label for='SearchKeyword'>Keresés:</label>
<?php
echo $html->input('Search/keyword', array('size' => '40'));
echo $html->selectTag('Search/category_id', $categories, $category_id, null, null,
false);
echo $html->submit('OK');
?>
</form>

<?php
$text = "Új termék hozzáadása";
echo $html->link(
  $html->image("icon/edit_add.png", array("alt"=>$text)).$text,
  "/admin/products/add/{ $category_id }",
  null,
  null,
  false
);
```

```

    $text = "Kuka megtekintése";
    echo "<br/>".$html->link(
        $html->image("icon/trashcan_full.png",array("alt"=>$text)).$text,
        "/admin/products/trash",
        null,
        null,
        false
    );
?>
</div>

<div id="producttable">
<?php

if($products) include "producttable.shtml";
else echo "<div class='warning'>Nincs találat!</div>";
?>
</div>

```

9.3.9.3. admin_show.shtml

```

<h1><?php echo $product["Product"]["name"];?></h1>
<table>
  <tr>
    <td>Cikkszám: </td>
    <td><?php echo $product["Product"]["id"];?></td>
  </tr>
  <tr>
    <td>Ár: </td>
    <td><?php echo
    >toMoney($product["0"]["gross"])."</b><br/>";?>      (<?echo      "<b>"$.s-
    >toMoney($product["Product"]["price"])."              +      "$s-
    >toMoney($product["0"]["taxvalue"])." Áfa" ?></td>      ".s-
  </tr>
</table>

<div class='info'>
Leírás: <?php echo nl2br($product["Product"]["description"]);?>
</div>
<div class='center'>
<?php
  if($category_id=="trash")$link = "/admin/products/trash";
  else $link = "/admin/products/index/".$category_id;
  $text = " vissza a termékekhez";
  echo $html->link(
    $html->image("icon/1leftarrow.png",array("alt"=>$text)).$text,
    $link,
    null,
    null,
    false
  );
?>
</div>

```

9.3.9.4. admin_trash.html

```

<h1>Törölt termékek</h1>
<div class='operations'>
<form id='search' method="post" action="/admin/products/trash">

```

MELLÉKLET

```
<label for='SearchKeyword'>Keresés:</label>
<?php
echo $html->input('Search/keyword',array('size' => '40'));
echo $html->submit('OK');
?>
</form>
<?php
    $text = " vissza a termékekhez";
    echo $html->link(
        $html->image("icon/1leftarrow.png",array("alt"=>$text)).$text,
        "/admin/products",
        null,
        null,
        false
    );

?>
</div>
<div id="producttable">
<?php

if($products){
    echo "<table>";

        echo          $html->tableHeaders(array("Termék", "<a href
        ='/admin/products/undelall'>Műveletek</a>"));

        foreach($products as $product)
        {
            $p = $product["Product"];
            $id = $p["id"];
            $link = $html->link($html->link($p["name"], "/admin/products/show/" . $id . "/trash");
            $operations = $html->link($html->link($html->image("icon/folder_outbox.png",array("alt"=>"Visszaállítás")),
            "/admin/products/undel/$id",null, null, false);
            echo $html->tableCells(
                array($link,
                    $operations),
                array("id"=>"sor".$id));
        }
        echo "</table>";
    }
else echo "<div class='warning'>Nincs találat!</div>";
?>
</div>
```

9.3.9.5. details.thtml

```
<?php
    $p = $product["Product"]; //a termék adatai
    $ph = $product["Photo"]; //fotók
    $t = $product["Tax"]; //adók
    $calc = $product[0]; //számított értékek
echo "<div id='productdetails'>";

echo "<h1>{$p["name"]}</h1>";
```

MELLÉKLET

```

echo "<div id='route' class='operations'>{$route}</div>";

echo "<div id='price'>Eladási ár: <div class='gross'>".{$s-
>toMoney($calc["gross"])."</div>";
echo "<div class='net'>".{$s-
>toMoney($p["price"])."+".$t["name"]."</div>";
echo "<a href='#' onclick='addToCart(\".$p["id"].")'
id='pl\".$p["id"].">Kosárba</a>";
echo "</div>";
//foto kirakása
if(count($ph)){
    $f = "/" . $path . $ph[0] ["id"] . ".jpg";
    $img = "<img src='". $s->thumbURL($f,array("w"=>200)) . "'
alt='{ $p["name"] }' class='product_photo' onclick='openw(\"/product_photos/slideshow/{ $p["id"] } \")' />";
    echo "<div id='photo'>";
    echo $img;
    echo "<div class='hint'>Kattintson a fotóra a további képek meg-
jelenítéséhez! (új ablakban nyílik meg.)</div></div>";
}
echo $html->link("Vissza", "/products/index/" . $p["category_id"]);

echo "<div class='operations'><h2>Leírás:</h2>". $p["description"] . "</div>";

echo "<div class='operations'><h2>Vevői megjegyzések:</h2>";

if(count($comments)){
    foreach($comments as $c){
        echo "<div class='comment'>";
        echo "Név: <b>". $c["User"] ["login"] . "</b><br/>";
        echo "Dátum: <b>". $c["ProductComment"] ["datum"] . "</b><br/><br/>";
        echo $c["ProductComment"] ["text"];
        echo "</div>";
    }
}
else{
    echo "Még nem érkezett egy megjegyzés sem.";
}

if($session->read("User")){
    echo "<form action='/product_comments/add/' method='post'>";
    echo "<fieldset>";
    echo "<legend>Az Ön véleménye:</legend>";
    echo $form->generateAreaDiv('ProductCom-
ment/text', "Szöveg:", false, null, 32, 10);
    echo $error->messageFor('ProductComment/text');
    echo $html->hidden("ProductComment/product_id", array("value"=>$p["id"]));
    echo "<div class='submitblock'>";
    echo $html->submit('Mentés');

    echo $html->input('ProductComment/cancel', array(
        "type"=>"reset",
        "value"=>"Töröl"));
    echo "</div>";

    echo "</fieldset>";
    echo "</form>";
}
else{

```

MELLÉKLET

```

    echo "Megjegyzés írásához kérem jelentkezzen be.";
}
echo "</div>";
echo "</div>";

echo "<form id='quantity' action='index.php' style='position: absolute;
top:-500px; width: 120px'>";
echo "<label for='ShoppingCartQuantity'>Mennyiség:</label>";
echo $html->input('ShoppingCart/quantity',array("value"=>1,"size"=>3));
echo $html->input('ShoppingCart/submit',array("onclick"=>'send()', "type"=>"button", "value"=>"Ok"));
echo $html->input('ShoppingCart/cancel',array("onclick"=>'hideme()', "type"=>"button", "value"=>"Mégsem"));
echo $html->hidden("ShoppingCart/product_id",array("value"=>$p["id"]));
echo "</form>";
?>
<script type="text/javascript"><!--
function addToCart(product){
    $('#ShoppingCartProductId').value=product;
    link = $("p1"+product);
    $("quantity").style.top=(link.offsetTop-5)+'px';
    $("quantity").style.left=(link.offsetLeft-5)+'px';
    new Effect.Appear("quantity",{ duration: 0.5 });
}

function send(){
    product = $('#ShoppingCartProductId').value;
    quantity = parseInt($('#ShoppingCartQuantity').value);
    if(isNaN(quantity)){
        alert("Ez nem egy szám.");
        $('#ShoppingCartQuantity').select();
        return false;
    }
    else if(quantity>100 || quantity<1){
        alert("A rendelhető mennyiség min.1, max. 100 lehet.");
        $('#ShoppingCartQuantity').select();
        return false;
    }
    new Ajax.Request(
        '/shopping_carts/setting/'+product+'/'+quantity,
        {
            method: 'post',
            onComplete: refreshShoppingCart
        });
    hideme();
}

function hideme(){
    new Effect.Fade("quantity");
}
function openw(link)
{
    filewindow = window.open(link, "window", "height= 600,
width=800,scrollbars=yes");
}

```

MELLÉKLET

```
function refreshShoppingCart(){
    new Ajax.Updater('shopping_cart',
        '/shopping_carts/refresh',
        {
            method: 'post'
        });
}
hideme();
--></script>
```

9.3.9.6. index.shtml

```
<h1>Termékek</h1>
<div id='route' class='operations'>
<?php
    echo $route;
?>
</div>
<div id="producttable">
<?php

if($products){
    echo "<table>";
    echo $html->tableHeaders(array("Termék", "Ár", "Rendelés"));
    foreach($products as $product){
        $p = $product["Product"]; //a termék adatai
        $ph = $product["Photo"]; //fotók
        $t = $product["Tax"]; //adók
        $calc = $product[0]; //számított értékek
        //fénykép ha van
        if(count($ph)){
            $f = "/" . $path . $ph[0]["id"] . ".jpg";
            $img = "<img src='\" . $s->thumbURL($f, array("w"=>40)) . \"'
alt='{ $p["name"]}' class='product_photo' />";
        }
        else $img = "<img src='\" . $path . \"noimg.jpg'
alt='{ $p["name"]}' />";

        $cell[0] = $img . $html-
>link($p["name"], "/products/details/" . $p["id"]);
        $cell[1] = "<div class='gross'>\" . $s-
>toMoney($calc["gross"]) . "</div>";
        $cell[1] = "<div class='net'>\" . $s-
>toMoney($p["price"]) . "+" . $t["name"] . "</div>";
        $cell[2] = "<a href='#' onclick='addToCart(\" . $p["id"] . \")';return
false;' id='pl\" . $p["id"] . \">'>Kosárba</a>";
        echo $html->tableCells($cell);
    }

    echo "</table>";
}
else {
    if($category_children){ //ha vannak további alkategóriák
        echo "<h2>További kategóriák:</h2>";
        foreach($category_children as $item){
            echo $html-
>link($item["name"], "/products/index/" . $item["id"]) . "<br/>";
        }
    }
}
```


MELLÉKLET

```

else echo "<div class='warning'>Nincs találat!</div>";
}

echo "<form id='quantity' action='index.php' style='position: absolute;
top:-500px; width: 85px'>";
echo "<label for='ShoppingCartQuantity'>Mennyiség:</label>";
echo $html->input('ShoppingCart/quantity',array("value"=>1,"size"=>3));
echo $html->input('ShoppingCart/submit',array("onclick"=>'send()', "type"=>"button", "value"=>"Ok"));
echo $html->input('ShoppingCart/cancel',array("onclick"=>'hideme()', "type"=>"button", "value"=>"Mégsem"));
echo $html->hidden("ShoppingCart/product_id",array("value"=>$p["id"]));
;
echo "</form>"
?>
</div>
<script type="text/javascript"><!--
function addToCart(product){
    $('#ShoppingCartProductId').value=product;
    link = $("pl"+product);
    $("quantity").style.top=(findPosY(link)-5)+'px';
    $("quantity").style.left=(findPosX(link)-5)+'px';
    new Effect.Appear("quantity",{ duration: 0.5 });
}

function send(){
    var product = $('#ShoppingCartProductId').value;
    var quantity = parseInt($('#ShoppingCartQuantity').value);
    if(isNaN(quantity)){
        alert("Ez nem egy szám.");
        $('#ShoppingCartQuantity').select();
        return false;
    }
    else if(quantity>100 || quantity<1){
        alert("A rendelhető mennyiség min.1, max. 100 lehet.");
        $('#ShoppingCartQuantity').select();
        return false;
    }
    new Ajax.Request(
        '/shopping_carts/setting/'+product+'/'+quantity,
        {
            method: 'post',
            onComplete: refreshShoppingCart
        });
    hideme();
}

function hideme(){
    new Effect.Fade("quantity");
}
function openw(link)
{
    filewindow = window.open(link, "window", "height= 600,
width=800,scrollbars=yes");
}

```

MELLÉKLET

```
function refreshShoppingCart(){
    new Ajax.Updater('shopping_cart',
        '/shopping_carts/refresh',
        {
            method: 'post',
            onComplete: highliht
        });
}
function highliht(){
    alert('Termék berakása a kosárba rendben.');
```

9.3.9.7. producttable.shtml

```
<?php
    echo "<table id='producttable'>";
    $c = $html->checkbox("Product/is_public", null, ar-
ray("onclick"=>"chkall(this.checked)"));

    echo $html->tableHeaders(array("Termék", $c."Státusz", "Műveletek"));

    foreach($products as $product)
    {
        $p = $product["Product"];
        $id = $p["id"];
        $link = $html->link($p["name"], "/admin/products/show/" . $id . "/" . $category_id);

        $chk = array();
        if($p["is_public"]) $chk = array("checked"=>"checked");
        else $chk = array();
        $chk+=array("onclick"=>"new
Ajax.Request('/admin/products/setstatus/" . $id . "')");
        $chk+=array("class"=>"chk");
        $checkbox = $html->checkbox("Product/is_public" . $id, null,
$chk);

        $operations = $html->link($html->image("icon/camera.png",
array("alt"=>"Fényképek")), "/admin/product_photos/index/" . $id, null,
false);

        $operations .= $html->link($html->image("icon/edit.png",
array("alt"=>"Szerkesztés")), "/admin/products/edit/" . $id, null, false
);

        $operations .= $html->link($html->image("icon/cancel.png", array("alt"=>"Törlés")),
"/admin/products/del/" . $id, null, 'Biztos, hogy
törölni akarja?', false);
        echo $html->tableCells(
            array($link,
                $checkbox,
                $operations),
```

```

        array("id"=>"sor".$id));
    }
    echo "</table>";
?>
<script type="text/javascript"><!--
function chkall(status)
{
    items = $('producttable').getElementsByClassName('chk');
    items.each(function(item) {
        if(item.checked!=status) item.click();
    });
}
--></script>

```

9.3.10. product_photo

9.3.10.1. admin_changetext.shtml

```
<?php echo $desc ?>
```

9.3.10.2. admin_index.shtml

```

<script type="text/javascript"><!--
function get_photos()
{
    var myAjax = new Ajax.Updater('productphotos',
        '/admin/product_photos/getphotos/<?php echo $product_id?>',
        {
            method: 'post',
            onComplete: make_icon_drag
        });
}

function uploadsucceed()
{
    $('errorMessage').innerHTML = "<?php echo "Sikeres feltöltés!"?>";
    new Effect.Highlight('errorMessage', {startcolor:'#00AA00', end-
color:'#FFFFFF'});
    get_photos();
}

function startupload()
{
    $('errorMessage').innerHTML = "<img src='/img/icon/ajax-loader.gif'
alt=''/>"+"<?php echo "Feltöltés..."?>";
    //new Effect.Highlight('errorMessage', {startcolor:'#00AA00', end-
color:'#FFFFFF'});
}

function uploadfault(message)
{
    $('errorMessage').innerHTML = message;
    new Effect.Highlight('errorMessage', {startcolor:'#AA0000', end-
color:'#FFFFFF'});
}
--></script>
<h1><?php echo $product["Product"]["name"];?> fényképek</h1>
<?php

```

MELLÉKLET

```

echo                                                                                               $html-
>link("&laquo;vissza", "/admin/products/index/{product["Product"] ["catego
ry_id"]}", null, null, false);
include "form.shtml"?>
<div id="productphotos" class='info'>
    <?php include "productphotos.shtml"; ?>
</div class='close'></div>
</div>
<img id='trash' src='/img/icon/trashcan_empty.png' alt='trashcan'
style='' />

<script type="text/javascript">
<!--
function make_icon_drag()
{
    images = $('productphotos').getElementsByClassName("productphoto");
    images = $A(images);
    images.each( function(item)
        {
            new Draggable(item.id,
                {
                    revert: true,
                    ghosting: false
                });
        });

    desc = $('productphotos').getElementsByClassName("description");
    desc = $A(desc);
    desc.each( function(item)
        {
            new Ajax.InPlaceEditor(
                item.id,
                '/admin/product_photos/changetext/'+item.id,
                {
                    highlightcolor: "#c9D8E5",
                    cancelLink: false,
                    rows: 3,
                    cols: 14,
                    callback: function(form, value)
                    { return 'text=' + encodeURIComponent(value) }
                });
        });
    }

function init(){
    Droppables.add("trash",
        {
            accept: 'productphoto',
            onDrop: function(element, me)
            {
                if(confirm("Biztos, hogy törölni akarja?"))
                {
                    new
Ajax.Request('/admin/product_photos/del/'+ element.id,
                    {
                        method: 'post',
                        onComplete: get_photos
                    });
                }
            }
        });
}

```

```

    }
  });
}

init();
make_icon_drag();
new Draggable("trash",
  {revert: false,
   ghosting: false
  });
-->
</script>

```

9.3.10.3. admin_upload.shtml

```

<?php
if(isset($error)){
    $st = "<script type=\"text/javascript\"><!--\n";
    $st.= "window.parent.uploadfault('".$error."');";
    $st.= "\n--></script>";
}else{
    $st = "<script type=\"text/javascript\"><!--\n";
    $st.= "window.parent.uploadsuccess();";
    $st.= "\n--></script>";
}

echo $st;
?>

```

9.3.10.4. form.shtml

```

<div id="productphotos_form" style='float: right; width: 300px'>

<form method="post" action="/admin/product_photos/upload/<?php echo
$product_id?>" target='hiddenDiv' enctype="multipart/form-data"
style="width: 300px;">
<fieldset><legend>Fénykép feltöltése</legend>
<div class='inputblock'>
  <label for='ProductPhotoImg'>Fájl:</label>
  <?php
  if(!isset($error))$error="";
  echo $html->file('ProductPhoto/img');
  echo $html->tagErrorMsg('ProductPhoto/img', $error);
  echo $s->closeDiv();
  ?>
</div>
<div class='inputblock'>
  <label for='PhotoText'>Megjegyzés: </label>
  <?php
  echo
  >textarea('ProductPhoto/text', array("rows"=>"8", "cols"=>"30"));
  echo $s->closeDiv();
  echo $html->hidden('ProductPhoto/product_id', ar-
ray("value"=>$product_id));

  ?>
</div>
<div id='errorMessage'>&nbsp;</div>
<div class="submitblock">

```

MELLÉKLET

```
<?php
echo $html->submit('Mentés',array("onclick"=>"startupload()"));
echo $html->input('Category/cancel',array(
    "type"=>"button",
    "on-
click"=>"location.href='/admin/products/index/{$_product["Product"]["category_id"]}'",
    "value"=>"Vissza"));
?>
</div>
</fieldset>
</form>
<iframe id="hiddenDiv" name="hiddenDiv" style="width: 600px; height: 200px; border:1px solid black; display: none"></iframe>
</div>
```

9.3.10.5. productphotos.thtml

```
<?php
foreach($photos as $photo){

    $f = $path.$photo["ProductPhoto"]["id"].".jpg";
    $id= $photo["ProductPhoto"]["id"];
    echo "<div class='productphoto' id='pf{$id}'>";
    echo "    <img alt='Termékfotó' src='".$s-
>thumbURL($f,array("w"=>150))."'/>";
    echo "    <div class='description'
id='text{$id}'>".$photo["ProductPhoto"]["text"]."</div>";
    echo "</div>";
}
?>
<div class='close'></div>
```

9.3.10.6. slideshow.thtml

```
<?php
$p = $product["Product"];
echo "<h1>{$p["name"]}</h1>";
$max=0;
foreach($photos as $photo){
    $tmp[$max]=$photo["ProductPhoto"]["id"];
    $max++;
}
$max--;
if(!($cur = array_search($current,$tmp))) $cur=0;
if($cur == 0) $prev=$max;
else $prev=$cur-1;
if($cur==$max) $next=0;
else $next=$cur+1;

$link = "/product_photos/slideshow/".$p["id"]."/";
$prevLink = "<a href='".$link.$tmp[$prev]."' id='prev'><img
src='/img/icon/1leftarrow.png' alt='Előző'/></a> ";
$nextLink = "<a href='".$link.$tmp[$next]."' id='next'><img
src='/img/icon/1rightarrow.png' alt='Kövekező'/></a> ";
$closeLink = "<img src='/img/icon/cancel.png' onclick='window.close()'
alt='Bezár' style='cursor: pointer'/> ";

echo "<div id='main'>";
```

```

echo "<div id='navigation'>";
echo $prevLink.$closeLink.$nextLink;
echo "</div>";
    $f = $path.$tmp[$cur].".jpg";
    echo
        "<img src='".$s->thumbURL($f,array("w"=>400))."'
alt='{ $p["name"]}' class='product_photo' />";
echo "</div>";
echo "<div class='operations'>";
echo $photos[$cur]["ProductPhoto"]["text"];
echo "</div>";

?>

```

9.3.11. shopping_cart

9.3.11.1. index.shtml

```

<h1>Kosár tartalma:</h1>
<?php
if(count($items)){
echo "<div id='shopping_cart_table'>";

include "shopping_cart_table.shtml";

echo "</div>";

} else echo "A bevásárlókosár üres.";
?>
<script type="text/javascript"><!--
function send(product) {
    quantity = parseInt($('q'+product).value);
    if(isNaN(quantity)) {
        alert("Ez nem egy szám.");
        $('q'+product).focus();
        $('q'+product).select();
        return false;
    }
    else if(quantity==0 && !confirm("Biztos, hogy törölni szeretné?")) {
        return false;
    }
    else if(quantity>100 || quantity<0) {
        alert("A rendelhető mennyiség min.1, max. 100 lehet.");
        $('q'+product).value=1;
        $('q'+product).focus();
        $('q'+product).select();
        return false;
    }

    new Ajax.Request(
        '/shopping_carts/setting/'+product+'/'+quantity,
        {
            method: 'post',
            onComplete: refreshShoppingCart
        });
}

function del(product) {
    if(confirm("Biztos, hogy törölni akarja?")) {
        new Ajax.Request(
            '/shopping_carts/setting/'+product+'/0',

```

```

        {
            method: 'post',
            onComplete: refreshShoppingCart
        });
    }
}
function refreshShoppingCart() {
    new Ajax.Updater('shopping_cart_table',
        '/shopping_carts/refreshtable',
        {
            method: 'post'
        });

    new Ajax.Updater('shopping_cart',
        '/shopping_carts/refresh',
        {
            method: 'post'
        });
}
--></script>

```

9.3.11.2. refresh.shtml

```

<h1>Kosár</h1>
<?php
if(is_array($items) && count($items)) {

$sum=0;
foreach($items as $item) {
    $cell[0] = $item["Product"]["name"];
    $cell[1] = $item["ShoppingCart"]["quantity"].
    ".$item["Product"]["unit"];
    $cell[2] =
$item["Product"]["price"]*$item["ShoppingCart"]["quantity"]*(100+$item["P
roduct"]["Tax"]["value"])/100;
    $sum += $cell[2];
    $cell[2] = $s->toMoney(round($cell[2]));
    echo "<div class='item' id='sc_item{$item["Product"]["id"]} '>";
    echo $cell[0]. " - ".$cell[1];
    echo "<div class='price'>".$cell[2]. "</div></div>";
}
echo "<div class='item'><u>Összesen:</u><div class='price'>".$s->
toMoney($sum). "</div></div>";

echo "<div class='center'><b><a href='/shopping_carts/index'>Kosár megny-
itása&raquo;</a></b></div>";
} else echo "A bevásárlókosár üres.";
?>

```

9.3.11.3. setting.shtml

```

<?php
if(isset($err)) echo $err;
?>

```

9.3.11.4. shopping_cart_table.shtml

```

<?php

```


MELLÉKLET

```
if(count($items)){
    $s_gross=0;
    $s_net = 0;
    $s_tax = 0;
    echo "<table>";
    foreach($items as $item){
        $cell[0] = $item["Product"]["name"];
        $cell[0] .="<br/><a href='#' on-
click='del(\".$item["Product"]["id"].")'; return false;'>Törlés</a>";
        $cell[1] = "<input type='text' size='3' on-
change='send(\".$item["Product"]["id"].") '
id='q\".$item["Product"]["id"]." '
value='\".$item["ShoppingCart"]["quantity"]."'/>
\".$item["Product"]["unit"]";

        $gross =
$item["Product"]["price"]*$item["ShoppingCart"]["quantity"]*(100+$item["P
roduct"]["Tax"]["value"])/100;
        $s_gross += $gross;
        $net = $item["Product"]["price"]*$item["ShoppingCart"]["quantity"];
        $s_net += $net;
        $tax =
$item["Product"]["price"]*$item["ShoppingCart"]["quantity"]*( $item["Produ
ct"]["Tax"]["value"])/100;
        $s_tax += $tax;
        $cell[2] = "<div class='gross'>\".$s-
>toMoney(round($gross))."</div>\".$s-
>toMoney(round($net))."+".$item["Product"]["Tax"]["name"];

        echo "<tr>";
        echo "<td class='item'>\".$cell[0]."</td>";
        echo "<td class='quantity'>\".$cell[1]."</td>";
        echo "<td class='price'>\".$cell[2]."</td>";
        echo "</tr>";
    }
    echo "<tr><td colspan=2>Összesen:</td><td class='price'><div
class='gross'>\".$s->toMoney(round($s_gross))."</div>
\".$s->toMoney(round($s_net))." + \".$s->toMoney(round($s_tax))." Áfa
</td></tr>";

    echo "</table>";
    echo "<a href='/orders/make' id='order'>Pénztár&raquo;</a>";
}
else echo "A kosár üres";

?>
```

9.3.12. tax

9.3.12.1. admin_add.thtml

```
<h1><?php echo $Felirat;?></h1>
<form method="post" action="<?php echo $html->url($target)?>">

<div class='inputblock'>
    <label for='TaxName'>Adókulcs neve:</label>
    <?php
    echo $html->input('Tax/name',array('size' => '40'));
    echo $html->tagErrorMsg('Tax/name', 'Ezt a mezőt kötelező
```

```

kitölteni.')
```

 ?>

```

</div>

<div class='inputblock'>
  <label for='TaxValue'>Adókulcs:</label>
  <?php
    echo $html->input('Tax/value',array('size' => '40'));
    echo $html->tagErrorMsg('Tax/value', 'Ezt a mezőt kötelező
kitölteni.')
```

 ?>

```

</div>

<div class="submitblock">
  <?php
    echo $html->submit('Mentés');
    echo $html->input('Tax/cancel',array(
      "type"=>"button",
      "onclick"=>"location.href='/admin/taxes/index'",
      "value"=>"Mégsem"));
  ?>
</div>
</form>
```

9.3.12.2. admin_index.shtml

```

<?php
  //echo $javascript->link("category.js");
  ?>
<h1>Forgalmi adó kulcsok</h1>

<div class='operations'>
  <h2>Tevékenységek</h2>
  <?php
    $text = "Új adókulcs hozzáadása";
    echo $html->link(
      $html->image("icon/edit_add.png",array("alt"=>$text)).$text,
      "/admin/taxes/add",
      null,
      null,
      false
    );
  ?>

</div>
<table>
<?php
if(count($taxes)){
  echo $html->tableHeaders(array("Adókulcs neve", "Kulcs", "Műveletek"));

  foreach($taxes as $tax){
    $t=$tax["Tax"];
    $id=$t["id"];
    $operations = $html->link($html->image("icon/edit.png",
      ar-
ray("alt"=>"Szerkesztés")), "/admin/taxes/edit/".$id,null, null, false );
    $operations .= $html->link($html->image("icon/cancel.png",
      ar-
ray("alt"=>"Törlés")), "/admin/taxes/del/".$id,null, 'Biztos, hogy törölni
akarja?', false );
```

```

        echo $html->tableCells(
            array($t["name"],
                $t["value"],
                $operations));
    }
}
else echo "<div class='warning'>Nincs találat!</div>";
?>
</table>

```

9.3.13. user

9.3.13.1. admin_add.thtml

```

<h1>Felhasználó hozzáadása</h1>
<form method="post" action="<?php echo $html->url($target)?>">
<?php
echo $error->allMessagesFor('User');

echo "<fieldset>";
echo "<legend>Login adatok</legend>";
echo $error->messageFor('User/login');
echo $form-
>generateInputDiv("User/login", "Felhasználónév:", true, null, 35);

echo $error->messageFor('User/pwd');
echo $form->generateInputDiv
("User/pwd", "Jelszó:", true, "ok", 10, array("type"=>"password"));
echo $form->generateInputDiv ("User/pwd_confirmation", "Jelszó is-
métlése:", true, "ok", 10, array("type"=>"password"));
echo "</fieldset>";

echo "<fieldset>";
echo "<legend>Kapcsolat</legend>";
echo $error->messageFor('User/name');
echo $form->generateInputDiv ("User/name", "Név:", true, "ok", 35);
echo $error->messageFor('User/email');
echo $form->generateInputDiv ("User/email", "E-mail cím:", true, "ok", 35);
echo $form->generateInputDiv ("User/phone", "Telefonszám:", false, "ok", 35);
echo $form->generateInputDiv ("User/mobil", "Mobil:", false, "ok", 35);
echo $form->generateInputDiv ("User/fax", "Fax:", false, "ok", 35);
echo "</fieldset>";

echo "<fieldset>";
echo "<legend>Számlázási cím</legend>";
echo $error->messageFor('User/billing_city');
echo $form->generateInputDiv
("User/billing_city", "Település:", true, "ok", 35);
echo $error->messageFor('User/billing_zipcode');
echo $form->generateInputDiv
("User/billing_zipcode", "Írányítószám:", true, "ok", 10);
echo $error->messageFor('User/billing_address');
echo $form->generateInputDiv ("User/billing_address", "Utca,
házszám:", true, "ok", 35);
echo "</fieldset>";

echo "<fieldset>";

```

MELLÉKLET

```
echo "<legend>Szállítási cím</legend>";
echo "<div class='remark'>Amennyiben megegyezik a számlázási címmel,
hagyja üresen.</div>";
echo $form->generateInputDiv
("User/delivery_city", "Település:", false, "ok", 35);
echo $form->generateInputDiv
("User/delivery_zipcode", "Írányítószám:", false, "ok", 10);
echo $form->generateInputDiv ("User/delivery_address", "Utca,
házszám:", false, "ok", 35);
echo "</fieldset>";

?>
<div class="submitblock">
  <?php
    echo $html->submit('Mentés');
    echo $html->input('User/cancel', array(
      "type"=>"button",
      "onclick"=>"location.href='/admin/users/index'",
      "value"=>"Mégsem"));
  ?>
</div>
</form>
```

9.3.13.2. admin_edit.shtml

```
<h1>Adatak módosítása</h1>
<form method="post" action="<?php echo $html->url($target)?>">
<?php
echo $error->allMessagesFor('User');

echo "<fieldset>";
echo "<legend>Kapcsolat</legend>";
echo $error->messageFor('User/name');
echo $form->generateInputDiv ("User/name", "Név:", true, "ok", 35);
echo $error->messageFor('User/email');
echo $form->generateInputDiv ("User/email", "E-mail cím:", true, "ok", 35);
echo $form->generateInputDiv ("User/phone", "Telefonszám:", false, "ok", 35);
echo $form->generateInputDiv ("User/mobil", "Mobil:", false, "ok", 35);
echo $form->generateInputDiv ("User/fax", "Fax:", false, "ok", 35);
echo "</fieldset>";

echo "<fieldset>";
echo "<legend>Számlázási cím</legend>";
echo $error->messageFor('User/billing_city');
echo $form->generateInputDiv
("User/billing_city", "Település:", true, "ok", 35);
echo $error->messageFor('User/billing_zipcode');
echo $form->generateInputDiv
("User/billing_zipcode", "Írányítószám:", true, "ok", 10);
echo $error->messageFor('User/billing_address');
echo $form->generateInputDiv ("User/billing_address", "Utca,
házszám:", true, "ok", 35);
echo "</fieldset>";

echo "<fieldset>";
echo "<legend>Szállítási cím</legend>";
echo "<div class='remark'>Amennyiben megegyezik a számlázási címmel,
hagyja üresen.</div>";
echo $form->generateInputDiv
("User/delivery_city", "Település:", false, "ok", 35);
```

MELLÉKLET

```
echo                                                                 $form->generateInputDiv
("User/delivery_zipcode", "Irányítószám:", false, "ok", 10);
echo $form->generateInputDiv ("User/delivery_address", "Utca,
házszám:", false, "ok", 35);
echo "</fieldset>";
echo $html->hidden("User/id");
?>
<div class="submitblock">
  <?php
    echo $html->submit('Mentés');
    echo $html->input('User/cancel', array(
      "type"=>"button",
      "onclick"=>"location.href='/admin/users/index'",
      "value"=>"Mégsem"));
  ?>
</div>
</form>
```

9.3.13.3. admin_index.shtml

```
<h1>Felhasználók</h1>
<form id='search' method="post" action="/admin/users/index">
<label for='SearchKeyword'>Keresés névre:</label>
<?php
echo $html->input('Search/keyword', array('size' => '40'));
echo $html->submit('OK');

$text = "Új felhasználó hozzáadása";
echo "<br/>". $html->link(
  $html->image("icon/edit_add.png", array("alt"=>$text)).$text,
  "/admin/users/add/",
  null,
  null,
  false
);
?></form>

<?php
if(is_array($users)&&count($users)){
  echo "<table>";
  echo $html->tableHeaders(array("Név", "Műveletek"));
  foreach($users as $user){
    $u = $user["User"];
    $id=$u["id"];
    $operations = $html->link($html->image("icon/edit.png",
      ar-
ray("alt"=>"Szerkesztés")), "/admin/users/edit/" . $id, null, null, false );
    $operations .= $html->link($html->image("icon/cancel.png",
      ar-
ray("alt"=>"Törlés")), "/admin/users/del/" . $id, null, 'Biztos, hogy törölni
akarja?', false );
    echo $html->tableCells(
      array($html->link($u["name"],
"/admin/users/show/" . $u["id"]),
      $operations));
  }
  echo "</table>";
}
else echo $s->warning("Nincs találat!");
```

?>

9.3.13.4. admin_show.shtml

```
<h1>Felhasználó adatlapja</h1>
<table style='width:300px'>
<?php
$u = $user["User"];
echo $html->tableCells(array("Azonosító:", $u["id"]));
echo $html->tableCells(array("Felhasználónév:", $u["login"]));
echo $html->tableCells(array("Név:", $u["name"]));
$billing      =      $u["billing_zipcode"]."      ". $u["billing_city"].",
". $u["billing_address"];
if($u["delivery_zipcode"])
    $delivery  =      $u["delivery_zipcode"]."      ". $u["delivery_city"].",
". $u["delivery_address"];
else $delivery= $billing ;
echo          $html->tableCells(array("E-mail:", $html->
link($u["email"], "mailto:". $u["email"])));
echo $html->tableCells(array("Telefon:", $u["phone"]));
echo $html->tableCells(array("Mobil:", $u["mobil"]));
echo $html->tableCells(array("Fax:", $u["fax"]));
echo $html->tableCells(array("Számlázási cím:", $billing));
echo $html->tableCells(array("Szállítási cím:", $delivery ));
echo          $html->tableCells(array("Regisztráció
dátuma:", $u["registered_at"]));
?>
</table>
<?php
echo $html->link("Vissza", "/admin/users");
?>
```

9.3.13.5. signup.shtml

```
<h1>Regisztráció</h1>
<form method="post" action="<?php echo $html->url($target)?>">
<?php
echo $error->allMessagesFor('User');

echo "<fieldset>";
echo "<legend>Login adatok</legend>";
echo $error->messageFor('User/login');
echo          $form->
>generateInputDiv("User/login", "Felhasználónév:", true, null, 35);

echo $error->messageFor('User/pwd');
echo          $form->generateInputDiv
("User/pwd", "Jelszó:", true, "ok", 10, array("type"=>"password"));
echo $form->generateInputDiv ("User/pwd_confirmation", "Jelszó is-
métlése:", true, "ok", 10, array("type"=>"password"));
echo "</fieldset>";

echo "<fieldset>";
echo "<legend>Kapcsolat</legend>";
echo $error->messageFor('User/name');
echo $form->generateInputDiv ("User/name", "Név:", true, "ok", 35);
echo $error->messageFor('User/email');
echo $form->generateInputDiv ("User/email", "E-mail cím:", true, "ok", 35);
echo $form->generateInputDiv ("User/phone", "Telefonszám:", false, "ok", 35);
echo $form->generateInputDiv ("User/mobil", "Mobil:", false, "ok", 35);
```

MELLÉKLET

```

echo $form->generateInputDiv ("User/fax", "Fax:", false, "ok", 35);
echo "</fieldset>";

echo "<fieldset>";
echo "<legend>Számlázási cím</legend>";
echo $error->messageFor('User/billing_city');
echo $form->generateInputDiv
("User/billing_city", "Település:", true, "ok", 35);
echo $error->messageFor('User/billing_zipcode');
echo $form->generateInputDiv
("User/billing_zipcode", "Irányítószám:", true, "ok", 10);
echo $error->messageFor('User/billing_address');
echo $form->generateInputDiv ("User/billing_address", "Utca,
házzszám:", true, "ok", 35);
echo "</fieldset>";

echo "<fieldset>";
echo "<legend>Szállítási cím</legend>";
echo "<div class='remark'>Amennyiben megegyezik a számlázási címmel,
hagyja üresen.</div>";
echo $form->generateInputDiv
("User/delivery_city", "Település:", false, "ok", 35);
echo $form->generateInputDiv
("User/delivery_zipcode", "Irányítószám:", false, "ok", 10);
echo $form->generateInputDiv ("User/delivery_address", "Utca,
házzszám:", false, "ok", 35);
echo "</fieldset>";

?>
<div class="submitblock">
  <?php
    echo $html->submit('Mentés');
    echo $html->input('Country/cancel', array(
      "type"=>"button",
      "onclick"=>"location.href='/users/index'",
      "value"=>"Mégsem"));
  ?>
</div>
</form>

```