

**ТЕХНОЛОГИЧЕСКО УЧИЛИЩЕ ЕЛЕКТРОННИ СИСТЕМИ
към ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - СОФИЯ**

ДИПЛОМНА РАБОТА

Тема: Уеб базирана социална мрежа с геотаргетиране 1

Дипломант:

Жоро Йорданов

Научен ръководител:

Марио Пешев

С О Ф И Я

2 0 1 1

У В О Д

Геотаргетирането (geotargeting) в интернет маркетинга е метод за установяване на геолокацията [1] на потребителите на уеб приложение и доставяне на специфично съдържание спрямо реалното им физическо местонахождение. То може да бъде установено по критерии като държава, град, пощенски код, IP адрес, ISP и други.

Именно локализацията на съдържанието е най-разпространеното приложение на геотаргетирането. Много компании предоставят различна информация в уеб сайтовете си в зависимост от държавността на посетителите си – показват съответните валути, цени на продуктите, цени за доставки, най-близък магазин и т.н. Бизнеси в ниши като, например, заведения за хранене, магазини и други, могат да рекламират таргетирано до потребителите, които са в близост до реалното местоположение на техния продукт или услуга.

Геотаргетирането е широко употребявано и във филтрирането на интернет съдържание в дадени държави. Уеб услуги за излъчване на видео като BBC iPlayer [23], Hulu [24] и Vevo [25] ограничават съдържанието си за дадени страни, защото нямат лиценз за излъчване на видеата в съответните държави.

Услугите базирани на местонахождение (Location-Based Services) предоставят информация за обекти близо до локацията на техния потребител. Най-често тези услуги се използват през мобилни устройства – телефони, таблети и преносими компютри. Информацията, която се предоставя може да бъде, например, къде се намира най-близкия ресторант, хотел, финтес център и т.н., или пък къде се намират в момента приятелите на потребителя. Поради тази причина LBS набират все по-голяма популярност. Според проучване на Gartner [2] се очаква през 2012 г. броят на потребителите на подобни услуги да нарастне на 526 млн., спрямо 96 млн. за 2009 г. Според статия на eTimes™ [3] от Април 2010 г.

“3/4 от потребителите у нас ползват мобилен интернет”. Анкетата е проведена сред над 1000 българи на възраст до 35 г. и резултатите сочат, че 40% от анкетираните използват мобилен интернет всеки ден. Според прогнози от телекомуникационен доклад за България на анализаторската група Business Monitor International броят абонати на високоскоростен интернет у нас ще достигне 1,3 млн. до 2013 г. Предвижда се голяма част от тях да ползват услугата мобилен интернет.

Резултатите от тези проучвания показват добра перспектива за развитието на мобилни услуги свързани с геотаргетиране.

Целта на настоящата дипломната работа е, да предостави възможност на потребителите си, да намират информация за бизнеси (обекти / услуги), релевантни спрямо текущото физическо местонахождение на въпросния потребител.

Уеб услугата ще предоставя възможности на собствениците на бизнеси да ги промотират, чрез редица функционалности изградени за целта.

Потребителите ще могат да споделят физическото си местонахождение със своите приятели; ще могат да отбелязват къде (в кой бизнес обект / място) се намира в момента. Съответно собствениците на заведения ще могат да създават промоционални игри, чрез които да правят отстъпки за, например, потребител отбелязал се в тяхното кафене 30 пъти този месец.

Чрез създаване и бъдещо използване на API, услугата ще може да бъде развита и чрез десктоп или мобилни приложения за различните операционни системи. В бъдеще потребителите ще могат да отбелязват физическата си локация и чрез мобилно приложение или мобилен уеб сайт, изграден с помощта на API.

П Ъ Р В А Г Л А В А

ПОДОБНИ ПРИЛОЖЕНИЯ С ГЕОТАРГЕТИРАНЕ И ГЕОТАГИНГ. ИЗБОР НА РАЗВОЙНИ СРЕДСТВА И СРЕДИ

1.1. Преглед на съществуващи подобни услуги

1.1.1. Foursquare

Foursquare [14] е един от феномените в интернет в последно време. Представлява location-based service, който дава възможност на потребителите си да споделят физическото си местонахождение чрез мобилния си телефон. Така потребителите трупат посещения (т.нар. “check-in”) на определени места (кафенета, университети, паркове и т.н.) и отключват значки. Игровият елемент е един от основните характеристики на повечето LBS. Именно чрез повече посещения на дадени места, потребителите на Foursquare могат да отключват всякакви забавни значки (achievements), които “зарибяват” потребителите да продължават да споделят местонахождението си.

Foursquare има мобилни приложения за iPhone, BlackBerry, Android, Palm, и Ovi (Nokia).

Когато потребителят инсталира приложението на телефона си, то засича местонахождението му и предлага да се отбележи в обектите, които са в близост до него.

Потребителите могат да добавят приятели и да следят къде се намират те. Чрез интеграция с Facebook и Twitter споделянето на местонахождението достига до повече хора.

При редовно отбелязване в някое място, потребителят може да стане “Майор”, докато някой друг по-редовно отбелязващ се, не го измести. Собствениците на съответния бизнес могат да дават отстъпки или други награди на “майорите” си в социалната мрежа.

Foursquare предлага платени възможности на собственици на бизнеси и марки [7]. Една от тях е създаването на специални съвети – Tips, като например класацията на The New York Times за най-интересните места по време на Зимните олимпийски игри във Ванкувър през 2010 г. Друг пример е Windows Live Photo Gallery, които помогнаха на хората, които ги следят, да намерят интересни места за снимане в техните градове. Така наречените Tips могат да бъдат и приканващи към действие: например “Отидете тук. Поръчайте си това. А сега си поръчайте допълнително.” [7].

Foursquare предоставя възможност и за създаване на специални значки, които хората да могат да спечелят. Пример за подходяща употреба на тази функционалност са Lucky Magazine, които дават специална значка на всеки, отбелязали се в избрани модни бутици. Toys’R’Us (верига магазини за детски играчки), наградиха всички лоялни клиенти със значка, даваща им специални отстъпки на Черен петък (петъкът след Деня на благодарността в САЩ).

Foursquare първоначално е реализиран на PHP (работещо на Apache сървър) с MySQL база от данни, но в последствие е пренаписан на Scala с Lift Web Framework и PostageSQL.

Изглед на началната страница на продукта е показан във фиг. 1.1.

1.1.2. Gowalla

Gowalla [15] е един от двама основни конкуренти на Foursquare. Концепцията ѝ в общи линии е доста подобна на тази на Foursquare. Предлага приложения за Apple iPhone, Android, BlackBerry, Palm и Apple iPad. Има и мобилна версия на уеб сайта.

Регистрацията в Gowalla отнема секунди, изисква само основна информация.

Дизайнът на сайта и приложенията е доста приятен, със свежи цветове и интересни, специално направени стикери (еквивалент на на значките в Foursquare).

Gowalla също показва интересни места наблизо, както и разстоянието им спрямо моментното потребителско. Социалният елемент с приятелството присъства и тук. Има възможност за интеграция с Facebook, Twitter, Foursquare и Tumblr.

Passport stamps са като “значките” във Foursquare [26]. При посещение на дадено място, при дадени условия (брой на посещенията, време на посещенията и т.н.) потребителят получава стикер, който се запазва в неговия “паспорт”. Така например, ако потребителят се отбележи за пръв път в Ню Йорк, САЩ, той ще получи стикер в паспорта си, който донякъде ще отговаря на идеята на истинските паспорти.

Потребителите на Gowalla носят виртуални предмети, които събират и респективно оставят на дадени места. За съжаление обаче предметите не са релевантни към местата в които се намират, а също така нямат стойност в реалния свят.

Собствениците на бизнеси имат възможност да организират кампании, при която клиентите им да печелят специално създадени стикери. Например Disney създадоха подобна кампания, в която имаше над 100 стикера свързани с техния парк. Споделянето на снимки и коментари за отделните атракции се оказва добра реклама, според маркетинг специалистите [26].

На фиг. 1.2 е показан изглед от началната страница на Gowalla.

1.1.3. Facebook Places

Най-бързо развиващата се социална мрежа Facebook [16] също предоставя възможност за geo tagging. Така всеки потребител може да види, кои от неговите приятели във Facebook са наблизо, след като отбележи че е на дадено място. Услугата предоставя възможност и за намиране на изгодни търговски оферти, например намаления от различни магазини в близост до неговото местоположение, като тези места се отбелязват със специална икона. Най-голямото предимство на Facebook

Places спрямо Foursquare и Gowalla е голямата база данни с потребители (около 500млн.), с които стартират.

На фиг. 1.3 е показан изглед на страницата на Facebook Places.

1.1.4. Yelp

Yelp [17] е социална мрежа насочена към потребителски ревята на места и локално търсене на такива. Създадена през 2004 в Сан Франциско, САЩ, в момента мрежата има около 40 млн. посещения на месец, а годишният ѝ доход е около \$30 млн.

Цялостната концепция на Yelp е съсредоточена върху ревятата, които потребителите пишат за места (ресторанти, магазини или други обекти, извършващи услуги), които са посетили. Подобни сайтове са реализирани още през 2000 г., но въпреки това Yelp успява да пробие в световен мащаб поради няколко причини. На първо място Yelp въвежда социалния елемент. Поради факта, че едва ли за потребителите е голям стимулът да напишат ревято за заведението и оmlета, който са хапнали днес, Yelp предоставя възможност за гласуване за написаното ревято. Потребителите могат да следят други потребители и да оценяват ревятата им.

Хората написали реалните си имена, редовно добавящи снимки на местата, които ревятират и разбира се, с качествени ревята могат да попаднат в елитния Yelp отряд (Yelp Elite Squad) и да бъдат поканени на събития, организирани от Yelp [8].

Модерацията в сайта е на ниво - не се срещат “ревята” от типа “Яко е.” или “Не струва”. Текстове копирани дума по дума от сайта на даденото място също са рядкост. Тоест ревятата в мрежата са достоверни.

Друга причина, поради която Yelp се налага на пазара е, че още от самото начало рекламните банери в сайта са минимално количество. Цялостния бизнес модел на услугата се базира върху допълнителни

възможности за собствениците на обекти. Именно собствениците могат да заплатят за показването на обекта им по-напред при търсене по дадени критерии. Платена функционалност е и подчертаването (highlight) на вече написани ревюта, както и възможност за отговоряне на ревютата от страна на собственика [8].

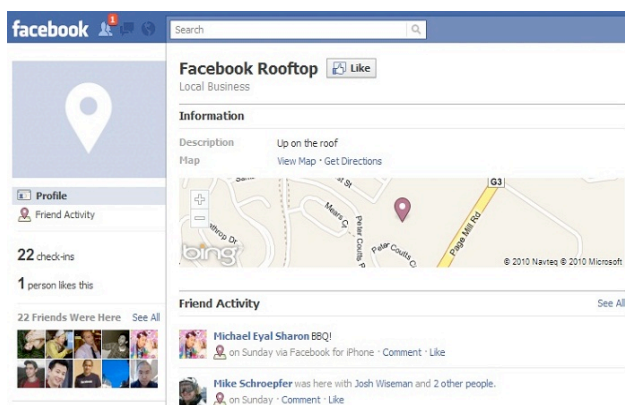
На фиг. 1.4 е показан изглед на страница за преглед на място.



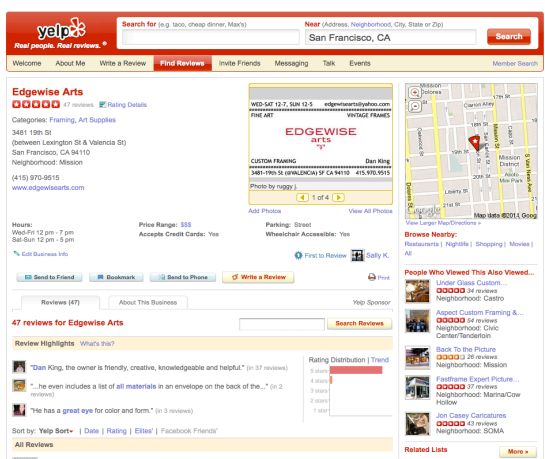
Фиг. 1.1. Foursquare



Фиг. 1.2. Gowalla



Фиг. 1.3. Facebook Places



Фиг. 1.4. Yelp

1.2. Развойни средства и среди

1.2.1. Panic Coda

Coda [18] е комерсиално приложение за разработка на уеб софтуер под Mac OS, създадено от Panic през 2007 г. Още същата година печели наградата от Apple за приложение, предлагащо най-добро потребителско изживяване (Apple Design Award for Best User Experience).

Опростеният интерфейс е един от големите плюсове на Coda.

Компанията Panic има разработено отлично приложение за FTP достъп – Transmit, ядрото на което е интегрирано в Coda.

Контролът на кода е лесен, защото в приложението има вграден Subversion source control модул, чрез който и синхронизацията с екипа, работещ по проекта става лесно и интуитивно.

Поддържа и “Clips”. Това са парчета често използван код, който може да се извиква чрез клавишна комбинация. Тази функционалност е изключително полезна, при изграждането на CSS стилове. Чрез нея дизайнерът задава HEX или RGB стойностите на цветовете, ползвани в дизайна и по време на разработката може да ги извика чрез, например, linkColor + клавиш Tab.

Code navigator е полезен за намиране на определени части от код при огромно количество редове.

Чрез добавка за компресиране на CSS, могат да се спестят ценни kb от размера на крайния CSS файл. Това се случва, като благодарение на добавката автоматично се премахнат всички празни пространства – нови редове и спейсове.

1.2.2. Adobe Photoshop

Въпреки, че продуктът на Adobe [19] насочен за уеб дизайн разработка и векторна графика е Adobe Fireworks, Photoshop върши отлична работа в уеб дизайна. Въпреки, че има и алтернативи, разпространяващи се свободно (например GIMP), Photoshop остава

стандарт в уеб дизайн индустрията. В по-голямата част от обяви за работа изискванията са, кандидатът да е запознат с Photoshop, т.е. по-голяма част от фирмите го използват. Fireworks остава на заден план, поради няколко причини: по-сложни Photoshop файлове не винаги се отварят коректно в Fireworks; Anti-aliasing на шрифтовете не е достатъчно добър; Липсват голяма част от инструментите в Photoshop, т.е. винаги ще има повод да се използва и Photoshop, когато се работи с Fireworks. В повечето случаи Photoshop би бил достатъчен за ефективна разработка на уеб, докато Fireworks, често би се нуждал от Photoshop при разработка на по-сложни графики.

1.2.3. MAMP

MAMP [5] е пакет от open source софтуер, обикновено използван заедно, за разработване на уеб приложения работещи на сървъри, използващи Apple Mac OS X. Абривиатурата идва от Mac OS X (операционната система), Apache (уеб сървър), MySQL (сървърът за базата данни), PHP, Perl или Python (програмните езици използвани за уеб разработката, в случая – PHP).

MAMP се използва предимно за разработване на уеб приложения, а не като основен сървър, върху който да работят.

Apache HTTP Server е уеб сървър с отворен код за Unix, Windows, Novell, NetWare и други платформи. Apache има огромен принос за първоначалното разрастване на World Wide Web и чрез него работят над 70% от сайтовете (включително Wikipedia). Счита се от много специалисти за платформа, според която се разработват и оценяват другите уеб сървъри.

1.2.4. Subversion

Subversion [6] (съкр. SVN) е програмен продукт за управление на софтуерните версии. Използва се главно за поддръжката на настоящи и

минали версии на файлове за изходен код, уеб страници и документация. Главната му цел е била да замести използваната широко до момента на излизането си, програма за управлението на версии с името Concurrent Versions System (съкр. CVS).

Subversion е широко използвана система в обществото на свободния изходен код и се използва от редица известни проекти, включително Apache Software Foundation, KDE, Free Pascal, FreeBSD, GCC, Python, Django, Ruby, Mono, SourceForge.net, ExtJS and Tigris.org.

Subversion е лицензирана под лиценза на Apache, което я прави продукт с отворен изходен код.

1.2.5. Dropbox

Dropbox [20] е услуга за синхронизиране и споделяне на файлове между няколко компютъра. Работата върху файловете качени в Dropbox може да продължи дори офлайн, като промените ще бъдат синхронизирани веднага след връзка с интернет. Синхронизират се само промените във файловете, а не целите файлове, което спестява време. Безплатният пакет предлага 2 GB пространство, което е достатъчно за хостването на SVN repository на проекта на настоящата дипломна работа.

1.2.6. База данни от споделен хостинг

За да бъде БД на приложението една и съща за всички, като по този начин всяка промяна в БД е мигновена и не изисква допълнителна синхронизация при промяна в структурата или съдържанието на БД използваме споделен хостинг от Superhosting [21], който предоставя MySQL БД. Това допълнително улеснява и ускорява процеса на разработка като помага например за по-бързо тестване и откриване на бъгове.

1.2.7. MySQL Workbench Community Edition

MySQL Workbench [22] е програма за визуално проектиране на база данни, разработена от MySQL. Тя е наследника на проекта DBDesigner4. MySQL Workbench Community Edition е под GPL лиценз. Програмата предоставя графична среда за работа с MySQL сървъри и бази от данни. Програмата напълно поддържа всички версии на MySQL след версия 5.1. MySQL Workbench има три основни функционалности:

- SQL разработка – позволява създаването управление на връзки с MySQL сървъри. Позволява конфигуриране на параметрите за връзката и предоставя възможността да бъдат изпълнявани SQL заявки на всяка от връзките чрез вграден SQL редактор. Този модул замества функционалността, предоставяна преди от приложението Query Browser.
- Моделиране на БД – позволява създаването на модели на БД графично, както и прехвърлянето от схема на БД към MySQL сървър и обратно. Също така, редактиране на всички аспекти от БД чрез подробния редактор на таблици. Той предоставя лесен за използване интерфейс за редакция на таблици, колони, индекси, тригери, процедури и други.
- Администрация на MySQL сървър – позволява създаване и администриране на инстанции на MySQL сървъри. Този модул замества функционалността, предоставяна преди от приложението MySQL Administrator.

В Т О Р А Г Л А В А

ИЗИСКВАНИЯ КЪМ ГЕОТАРГЕТИНГ УЕБСАЙТА. АЛГОРИТЪМ. СРЕДСТВА ЗА РАЗРАБОТКА. СТРУКТУРА НА БАЗА ДАННИ

2.1. Изисквания към социалната мрежа с геотаргетиране

Уебсайтът трябва да предоставя следните възможности:

- 1.Регистрация и автентикация на потребители.
- 2.Възстановяване на забравена парола.
- 3.Редакция на потребителски профил:
 - 3.1. Промяна на парола
 - 3.2. Промяна на email адрес
 - 3.3. Добавяне на аватар
4. Потребителите трябва да имат възможност да добавят места чрез помощта на Google Maps. Добавените събития трябва да съдържат следната информация:
 - 4.1. Име
 - 4.2. Местонахождение (адрес)
 - 4.3. Описание
 - 4.4. Описателни тагове
 - 4.5. Категории
 - 4.6. Снимки
5. Потребителите трябва да могат да редактират добавени места.
6. Потребителите трябва да могат да коментират вече добавени места и да могат да добавят снимки към коментарите си.
7. Потребителите трябва да могат да отбелязват, че са посетили или ще посетят дадено място.
8. Потребителите трябва да могат да запазват дадени места, отбелязвайки ги като “Любими”.

9. Потребителите трябва да могат да създават, редактират и коментират събития, съдържащи следната информация:

- 9.1. Начало на събитието
- 9.2. Край на събитието (незадължително)
- 9.3. Описание на събитието
- 9.4. Място на събитието (свързвайки го с вече добавено място)
- 9.5. Снимка към събитието
- 9.6. Тип на събитието – публично или частно

10. Потребителите трябва да могат да следят активността на който и да е потребител, с цел информирание за неговите местонахождения, събитията на които ще отиде, коментарите му, и други. Ако потребителят е направил профила си частен, то тази информация трябва да се показва само на онези, които са лично одобрени от него.

11. Потребителите трябва да могат да обновяват статуса и местоположението си, както и да прикачват снимки към него.

12. Потребителите трябва да могат да търсят места по предварително зададени критерии – име, град, адрес, категория, тагове.

13. Потребителите трябва да могат да виждат хората, с публични профили, отбелязали се в близост до тях.

14. Потребителите, които всъщност са собственици или работници в дадено място, трябва да имат възможност да се автентифицират като такива. След ръчно одобрение от екипа на сайта, въпросните потребители ще имат допълнителни възможности за разширяване информацията за тяхното място. Те ще могат да:

- 14.1. Променят вида на страницата с мястото им
- 14.2. Създават официални събития
- 14.3. Добавят официални снимки
- 14.4. Отговарят на потребителски коментари
- 14.5. Добавят менюта (разписания) на предлаганите услуги

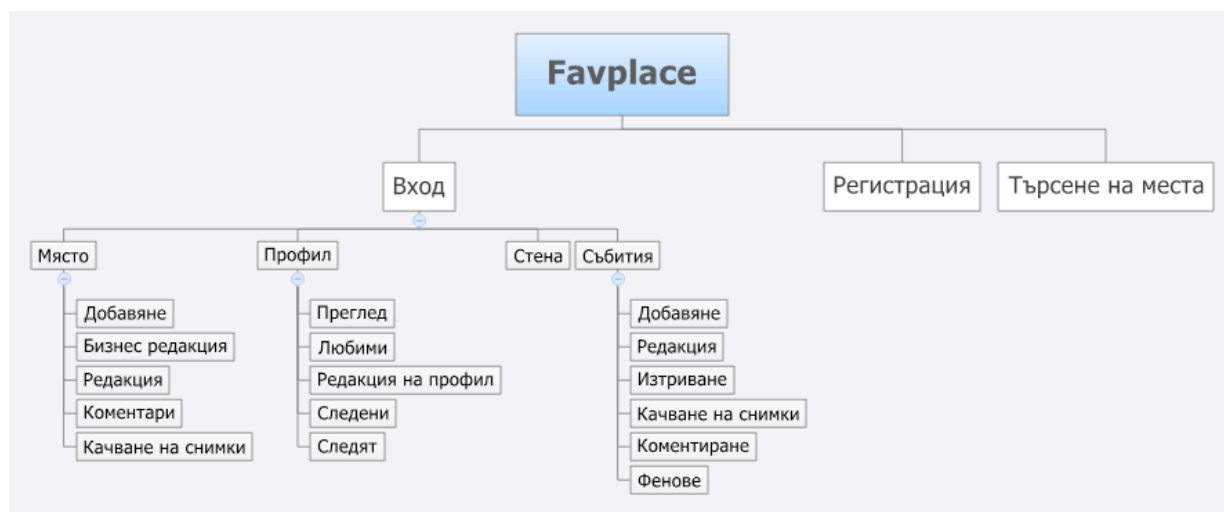
15. Уеб приложението трябва да предоставя API, с което да осигури разрастването, популяризирането и сътрудничеството с други уеб сайтове или приложения.

16. Уеб приложението трябва да работи под следните (или по-нови) интернет програми: Firefox 3.5, Internet Explorer 7, Safari 4, Chrome, Opera. На потребителите на Internet Explorer 6 трябва да бъде изведено информационно съобщение, за това защо е необходимо да обновят до по-актуална версия.

Задачата на настоящия дипломен проект е да се създаде графичен интерфейс, който да се реализира чрез HTML, CSS, jQuery. Като втора задача е зададена цялостната реализация (front-end и back-end) на модулите за регистрация и за бизнес места (т. 1 и т. 14 от изискванията).

2. Описание на алгоритъма на уеб приложението

Основният алгоритъмът на уеб приложението е показан на фиг. 2.1.



Фиг. 2.1. Алгоритъм на уебсайта.

Потребителят трябва да се регистрира, за да може да използва основната функционалност. Преди това може само да търси места и да прегледа малка информация за даден профил. След регистрация, потребителят се идентифицира с потребителско име и парола. После може да редактира информацията за профила си, както и да смени своята парола, email и аватар. Има възможност за добавяне или редактиране на

дадено място, както и за качване на снимки или коментиране. Потребителят може да следи други потребители, да види техните любими места, кои хора следят, и от кои хора са следени. Потребителя също така може да създава събития, след което да ги редактира. Ако събитието е публично всеки може да го коментира и да качва снимки за него, както и да види кои са феновете на събитието. Ако потребителя потвърди, че е собственик (или работник) на дадено място, то той ще може да добавя допълнителна информация за мястото. Ще може да създава менюта и официални събития, да променя изгледа на мястото, да добавя официални снимки и да отговаря на потребителски коментари от името на мястото.

2.3. Избор на средства за разработка

2.3.1. Apache

Apache е нашият избор за уеб сървър, защото има отлична интеграция с PHP, мощен е, гъвкав, HTTP/1.1 съвместим, имплементира най-съвременните протоколи, има висока степен на конфигурация и възможност за разширение, с отворен код е, и с неограничаващ лиценз.

Apache е тестван както от разработчици, така и от потребители от много време и работи на над 120 милиона интернет сървъри (до Април 2010), което е над полованата от уеб сървърите в света [5].

2.3.2. PHP

PHP е относително семпъл, но мощен език и е с отворен код. Той има много добра имплементация на обектно ориентираното програмиране. PHP също така притежава отлична поддръжка за базата данни MySQL и е най-популярният сървърен скриптов език. Популярната блогинг платформа Wordpress например, използва перфектната комбинация от PHP и MySQL.

2.3.3. MySQL

MySQL [35] е най-популярната база данни с отворен код, заради високата си производителност, надеждност и лекота на използване. Много от най-големите и най-бързо растящи организации включително Facebook, Google и Adobe разчитат на MySQL.

MySQL предоставя най-доброто по отношение на скалируемост и може да покрие и най-изискващите производителност очаквания, от която и да е система чрез нейните високоскоростни зареждания на различните услуги, кешове в паметта, текстови индекси и други подобряващи производителността механизми.

MySQL предлага и един от най-мощните енджини за транзакции на пазара. Включва пълна поддръжка на ACID транзакции, неограничено заключване на редове, възможност за дистрибутиране на транзакциите и поддръжка на многоверсионни транзакции.

MySQL е дефакто стандартът за сайтове с голям трафик, защото има високоефективен енджин за обработка на заявките, невероятно бързо добавяне на данни и силна поддръжка на често използвани в уеб функции като бързо търсене по големи текстови низове.

MySQL предоставя мощни механизми, осигуряващи самооторизирани потребители да имат достъп до MySQL сървър, както и мощни алгоритми за криптиране и декриптиране на данните.

2.3.4. Codeigniter

Codeigniter [27] е PHP фреймуърк. Неговата цел е по-бързо и надеждно писане на код, отколкото ако се пише на чисто PHP, като предоставя богат набор от библиотеки с често използвани функции.

CodeIgniter е безплатен и е лицензиран под Apache/BSD-style open source лиценз. Фреймуърка е наистина много лек. Ядрото на системата изисква само няколко много малки библиотеки. Това е в рязък контраст с много други библиотеки изискващи значително повече ресурси. Допълнителни библиотеки се зареждат динамично, когато е необходимо, така че базовата система е изключителна малка и бърза. Фреймуърка използва и подхода Model-View-Controller, който позволява разделение между бизнес логиката и изгледа на приложението. Също така CodeIgniter е бърз – изключително бърз. По времето на писане на дипломната работа, това е най-бързият PHP фреймуърк.

2.3.5. XHTML 1.0 Transitional

XHTML (eXtensible HyperText Markup Language) [36] е XML markup език за разработване на уеб страници. XHTML е преформулировка на HTML, като при XHTML има малко по-строго обособени правила за писане на код. Тези правила не са много на брой, но олесняват писането и най-вече четенето и пренаписването на вече написан код. Някои от тях са са:

1. Поставяне на валиден Doctype (Document Type Definiton), който указва на браузъра какъв код ще бъде визуализиран. В случая на XHTML 1.0 Transitional, доктайпът трябва да изглежда като показания на фиг. 2.2.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0  
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-  
transitional.dtd">
```

Фиг. 2.2. Document Type Definition

2. Всички тагове се пишат с малки букви.
3. Атрибутите на таговете се ограждат с кавички (атрибут="стойност").
4. Отварящите тагове трябва да отговарят на затварящите. Не трябва да има преплитане на елементи (тагове).
5. Елементите с id трябва да се използват само веднъж в страница.

6. Всички елементи трябва да имат затварящи тагове, дори и единичните такива, като например: `
` или ``

Спряхме се на XHTML 1.0 Transitional, защото въпреки навлизането на, все още неофициалният, HTML5 поддръжката му изисква допълнително настройване чрез JavaScript, което не е особено подходящо за по-големи уеб приложения.

2.3.6. CSS3

Cascading Style Sheets [28] е език за стилизиране на markup код (какъвто е XHTML), като чрез CSS се оформя форматирането и структурата на XHTML елементите. От 2002 г. насам CSS се е превърнал в основно средство за създаване на графичния изглед на уеб приложенията. Преди 2002 г. за целта са използвани HTML таблици, но това не е рационален вариант при наличието на CSS.

Версия 3 все още не е излязла официално, но въпреки това се използва в редица големи уеб сайтове. Една от основните причини да изберем CSS3, въпреки факта, че все още е в разработка е, че чрез него намалихме размера и количеството използвани изображения в конвертирането на дизайна от графична концепция (графичен файл) в структурирана markup страница. Избегнахме много проблеми като например заоблените ъгли, които чрез CSS3 могат да се постигнат с 2 свойства (property) на елемента, докато при CSS 2.1 за да се постигне този ефект са необходими 2 картинки, 4 CSS класа и 4 HTML елемента.

CSS3 поддържа и сянка на текста, сянка на елемент, прозрачност на фон, използване на шрифтове, които не са от стандартните използвани в операционните системи (т.нар. “not internet safe fonts”), както и доста други дребни подобрения.

Някои от изброените горе свойства не се рендират при Internet Explorer, но само по себе си това не е голям поради две причини:

интерфейсът изглежда добре, дори да не е със заоблени ъгли или със сенки на кутиите; в някои от случаите на употреба на такива CSS3 свойства, може да се приложи JavaScript фикс, който рендира съответните такива.

2.3.7. jQuery JavaScript Library

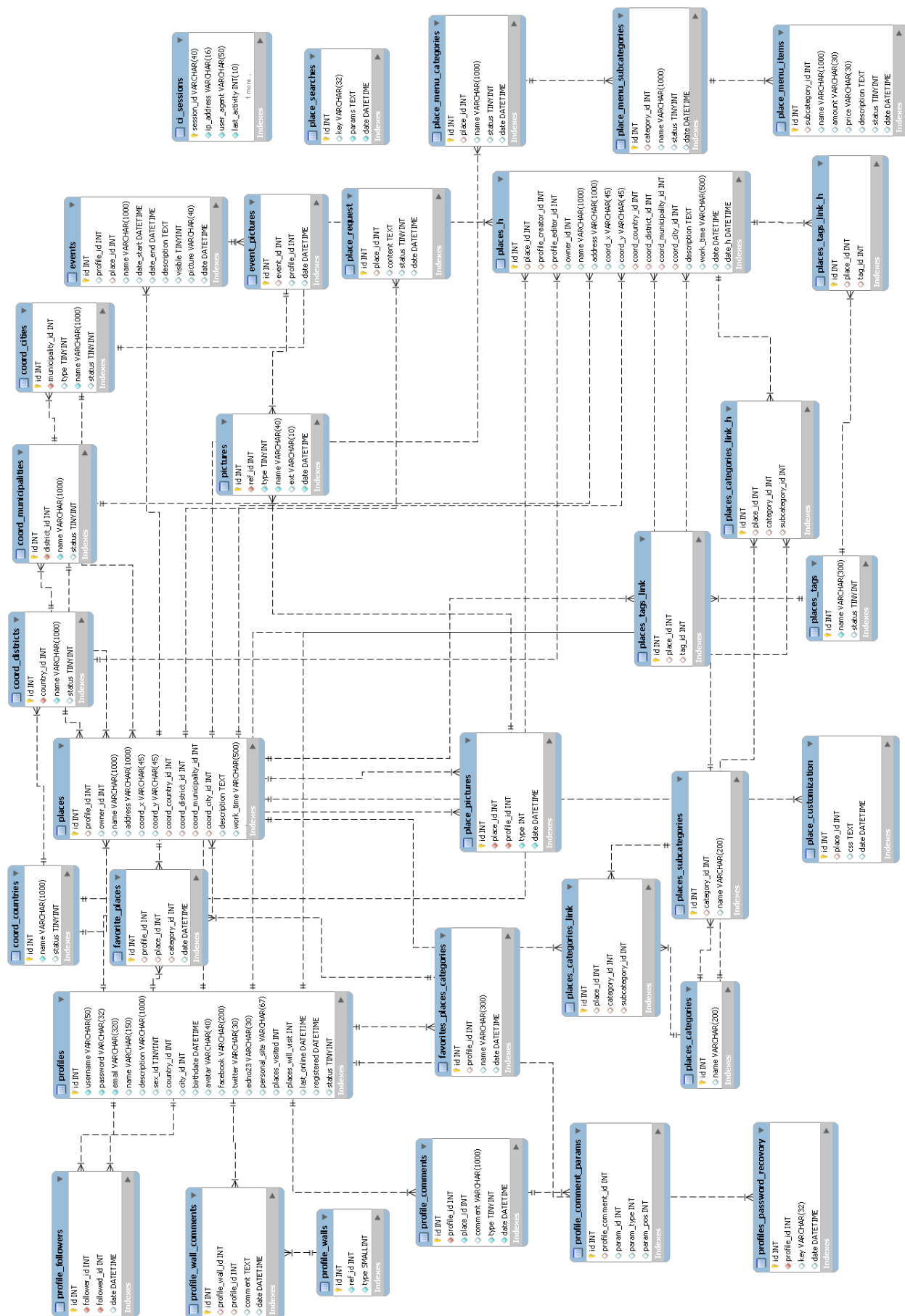
jQuery [29] е JavaScript библиотека, при която са преодолените различията при браузърите (cross-browser), работеща е със CSS3 свойства и е изключително оптимизирана и лека. Редица големи компании като Google, Dell, Digg, Mozilla.org, WordPress, Drupal я използват в уеб сайтовете си [29]. Друго предимство на библиотеката е, че чрез нея с лекота се работи с DOM елементи, а синтаксисът, с който това се случва е CSS-базиран. Достигането на елемент може да стане по редица начини, като някои по-често използвани са чрез: Class селектори (.class), ID селектори (#id), Child селектори – за достъп до елементите, които са деца на даден елемент в документа (.parent .child), Attribute селектори (:input) и други.

jQuery може да се зареди по два начина – локално за съответния уеб сайт или чрез Google, където е хостната библиотеката. Двата начина имат своите предимства и недостатъци. При локално зареждане всички файлове на уеб сайта се намират на едно място и са лесни за достъп. При глобалното зареждане обаче скоростта на зареждане на библиотеката ще бъде по-висока, защото най-вероятно, потребителят вече е посещавал друг сайт, в който е използвана същата библиотека и тя вече се е кеширала.

jQuery е мощен инструмент за създаване на интерактивни JavaScript модули с огромно количество отлично написани плъгини лицензирани под MIT. Неслучайно слоганът на jQuery е “write less, do more”.

2.4. Структура на базата данни на приложението

Структурата на базата данни е изобразена на графиката от фиг. 2.3.



Фиг. 2.3. Структура на базата данни.

Описание на таблиците от фиг. 2.3:

- **ci_sessions** – съдържа информация за сесията на потребителите
- **coord_countries** – съдържа списък с държави
- **coord_districts** – съдържа списък с областите към дадена държава
- **coord_municipalities** – съдържа списък с общините към дадена област
- **coord_cities** – съдържа списък с градовете към дадена община
- **events** – съдържа списък на събитията
- **event_pictures** – съдържа списък на снимките асоциирани към дадено събитие и определен потребител
- **profiles** – съдържа информация за потребителите
- **profile_followers** – показва кой потребител, кого следи
- **places** – съдържа информация за местата
- **favorite_places** – съдържа информация, кой потребител, кои места харесва и в коя категория е всяко място
- **favorite_places_categories** – съдържа категориите, създадени от даден потребител за разпределение на любимите места
- **pictures** – списък със снимки, асоциирани към място, събитие или друго чрез полето ref_id.
- **place_customization** – съдържа информацията за изгледа на дадено място
- **place_menu_categories** – съдържа главните категории за менюто на дадено място
- **place_menu_subcategories** – съдържа подкатегиите към дадена главна категория
- **place_menu_items** – съдържа продуктите или услугите в дадена подкатегория
- **place_pictures** – съдържа връзката между дадено място и качените наведнъж (една или повече) снимки

- **place_request** – съдържа заявките за асоцииране на даден потребител като собственик на определено място
- **place_searches** – съдържа списъка на търсенията (и техните параметри) извършени от потребителите
- **places_categories** – съдържа списъка на главни категории за местата
- **places_subcategories** – съдържа списъка на подкатегиите към дадена главна категория
- **places_tags** – съдържа списъка с тагове за местата
- **places_categories_link** – съдържа списъка на подкатегиите и категории асоциирани към дадено място
- **places_tags_link** – съдържа списъка с тагове асоциирани към дадено място
- **profile_comments** – съдържа списъка на коментари на даден потребител
- **profile_comment_params** – съдържа списъка на „споменавания“ асоциирани към даден коментар
- **profile_walls** – съдържа списък на събитията
- **profile_wall_comments** – съдържа списък с коментари за дадено събитие
- **profiles_password_recovery** – списък с нови пароли, генерирани в резултат на заявка от страна на потребителите за възстановяване на забравена парола.
- **places_h** – съдържа история на редакциите на информацията за дадено място
- **places_categories_link_h** – съдържа история на редакциите на категориите за дадено място
- **places_tags_link_h** – съдържа история на редакциите на тагове за дадено място

ТРЕТА ГЛАВА

РЕАЛИЗАЦИЯ НА ГРАФИЧНИЯ ИНТЕРФЕЙС НА СОЦИАЛНАТА МРЕЖА. ЦЯЛОСТНА РЕАЛИЗАЦИЯ НА МОДУЛИ РЕГИСТРАЦИЯ И БИЗНЕС

3.1. Обща стратегия за разработване на уеб сайта

3.1.1. Основна задача

Моята основна задача в този дипломен проект е създаването на запазен знак, графичен дизайн и потребителски интерфейс за уеб страниците. След приключването на концепцията в графичния редактор, конвертирах картинките в работещ markup код. За целта използвах няколко практики, които според моя опит като front-end developer, са най-удачни за разработване на сравнително мащабни уеб проекти. Практиките са описани в т. 3.1.2, т.3.1.3 и т. 3.1.4.

Задачата за реализация на front-end и back-end частта на формата за регистрация, както и за някои от бизнес модулите също е моя.

3.1.2. Оптимизация за браузъри

Според статистиката от [10] пазарният дял на Microsoft Internet Explorer 6 в България не надвишава 6%. За щастие преобладават по-съвременни браузъри, поддържащи голяма част от CSS3 свойствата. Тъй като по принцип MSIE 6 има проблеми с рендирането на много от CSS елементите, за да се коригират тези проблеми в изгледа на страницата, се добавя голямо количество допълнителен код, използват се така наречените “хакове”. Това усложнява и забавя разработката и пърформънса на уеб сайта. Разбира се, при добро желание тези проблеми могат да бъдат коригирани, но моето виждане е, че в случая това е излишно. Цялостната идея на настоящия дипломен проект е предназначена за хора с по-съвременни виждания. Тоест потребител, все

още използващ браузър излязал преди 10 години, едва ли ще използва смартфон за отбелязване в уеб сайта. С други думи, потребителите използващи MSIE 6 не са нашата таргет група. А и в духа на интернет кампаниите, които насърчават спирането на поддръжката за този браузър, настоящия дипломен проект също няма да го поддържа. На тези 6%, използващи IE 6 ще бъде показано съобщение, което накратко обяснява защо, как и откъде да обновят браузър клиента си.

3.1.3. Графичен формат на изображенията в дизайна

При конвертирането на дизайна от графичния редактор в markup код, един от основните моменти е “нарязването” на отделните елементи от дизайна и пресъздаването им в HTML, с помощта на CSS.

Начините това да бъде направено са много (обяснил съм концепцията на CSS спрайтовете в 3.1.4), но правилните са няколко.

Изборът на частта от картинката, която да се използва в CSS трябва да се обмисли. Неправилно е, например ако имаме кутия, която има заоблени ъгли, сянка и преливка за фон, и е с размер 600x400 точки, да я маркираме цялата и да я запааметим както е (независимо в какъв формат). В този случай ако съдържанието на картинката надвиши 400 пиксела по височина, то ще трябва да излезе от “кутията”, която картинката изобразява. Затова конкретната ситуация може да бъде решена, като се отреже лента широка (ако преливката на фона е вертикална) или висока (ако е хоризонтална) 1 пиксел и след това чрез CSS се зададе за фон, който да се повтаря по хоризонтала, или вертикала. Сянката и заоблените ъгли също могат да се добавят посредством CSS.

Картинката от горната ситуация може да бъде запазена като JPG, GIF или PNG файл. В папката с картинки от дизайна се срещат и трите формата.

Графичните файлове от тип JPG поддържат много цветове и имат високо ниво на компресия. Подходящи за използването на фотографии, елементи с много цветове или преливки от един цвят към друг.

GIF файловете са подходящи в ситуации, при които имаме елементи с ясно изразени ъгли или елементи с малко цветове (GIF форматът поддържа само 256 цвята, но съответно размерите му като kb са по-малки). Поддържа прозрачност.

Най-често използвани в настоящия проект са 32 Bit PNG (Portable Network Graphics) файловете. Имат добра компресия, поддържат алфа прозрачност (при GIF един пиксел е или прозрачен или не, докато тук има процент на прозрачност). Подходящи са използване в CSS спрайтове (т. 3.1.4).

Допреди 2-3 години PNG файловете бяха избягвани, защото Internet Explorer 6 не ги поддържа добре. Има проблеми с прозрачността и цветовете. Но поради факта, че в настоящия проект IE 6 е игнориран, то PNG тук е най-често срещан.

В крайна сметка всеки файл е различен формат, като съм се стремял да запазя размера на файла възможно най-нисък – като сменя файловия формат или помисля как да изрежа по-малка част от дадения елемент и след това го пресъздам с помощта на CSS.

3.1.4. Употреба на CSS sprites

Ползването на CSS спрайтове е чудесна техника за увеличаване зареждането на уеб приложението. Концепцията им се състои в това, че всички графики от дадена група елементи (например всичко иконки на дадена страница) се групират в един файл (най-често прозрачен PNG) и след това, чрез CSS се създават класове за отделните икони, като за фон им се задава този графичен файл. Чрез пропъртите `background-position` се променя позицията на картинката за всеки един клас, като по този начин се вижда само частта от фона, в която е разположена

графиката на съответната иконка. Фиг. 3.1 и фиг. 3.2 изобразяват някои от спрайтовете използвани в текущия проект.

Предимството на този метод е, че се намалява количеството HTTP заявки към сървъра. Ако имаме 15 файла – за всяка иконка в проекта, това означава 15 отделни заявки. Докато ако ги комбинираме в един файл – заявката е само една, което значително подобрява скоростта на зареждане на страницата. В голяма част от ситуациите се намалява и размера на файловете, т.е. ако например тези 15 картинки по отделно са с общ размер 100 kb, то в един правилно създаден спрайт ще бъдат 60-70 kb.

Недостатъците на спрайтовете са два. Първият е, че при IE 6 не могат да бъдат използвани, но в случая това не ни притеснява. Основният недостатък е, че не могат да бъдат използвани за елементи, при които е необходимо да се повтаря фонът (например input полета или textarea).



Фиг. 3.1. PNG файл използван като sprite.



Фиг. 3.2. Прозрачен PNG sprite.

3.2. Потребителски интерфейс

3.2.1. Графичен дизайн и типография

Логото на проекта е разработено във векторен формат, за да бъде лесно скалируемо и използваемо при необходимост за използването му в печатни медии, например за рекламни материали. Шрифтът, с който е изписано името на проекта – “Favplace” е Qlassik Bold и е разрешен за свободна употреба. Цветът на шрифта е преливка от тъмно към светло кафяво, но основният кафяв цвят е C: 75%, M: 68%, Y: 67%, K: 90%, а подчертавката е светложелта: C: 52%, M: 16%, Y: 100%, K: 1%.



Фиг. 3.3. Логото на проекта.

Цветовете използвани в потребителския интерфейс са светли и приветливи. Светлосиньо е използвано (hex: #1A74C7) за по-важни връзки, бутони и други елементи, с които потребителят може да взаимодейства пряко. Според редица проведени тестове за ползваемост на интернет страници, синьото е най-подходящият цвят за означаването на връзки. Когато потребителят взаимодейства с някой от активните елементи, цветът му се променя в зелен (hex: #7A901C). Лоша практика е използването на чисто черен цвят върху бял фон. Именно заради това

основният цвят на текста е тъмно сив (hex: #444), в повечето случаи разположен върху бял или светлосив фон.

В типографията са използвани несерифните шрифтове Myriad Pro с неговите вариации (удебелен, изтънен и т.н.), Arial и Tahoma, а за основен серифен шрифт – Georgia.

3.2.2. Основна конструкция

Интерфейсът е замислен така, че да се вижда нормално, без нужда от хоризонтално скролиране на страницата, при всички потребители с разделителна способност на екрана равна или по-висока от 1024 точки по хоризонтала.

Ширината на основните елементи в уеб приложението е фиксирана и не надвишава 960 пиксела. Този размер е оптималният за монитори с резолюция 1024x768 пиксела. Числото 960 е кратно на 3, 4, 5, 6, 8, 10, 12, 15 и 16, което прави възможно използването на така наречените “решетки” за построяването на основната конструкция на приложението. В някои блокове и списания насочени към уеб дизайна се дискутира въпроса, дали не е дошло време да се започне разработка, съобразена с разделителни способности над 1024x768 пиксела. Камерън Мол, уеб дизайнер, автор и говорител, включен в статия на списание .net “Топ 20 уеб дизайнера в света”, предлага използването на фиксирани ширини 1020, 1040 или 1080, вместо 960 [13]. Въпреки че на теория при по-голям размер може да се събере повече и по-прегледна информация, 960 пиксела остава подходящ избор, поради няколко причини. Според статистика на W3Counter.com за декември 2010г. [9] на първо място по разпространеност е резолюцията 1024x768. Въпреки нарастващия брой потребители с по-големи размери монитори (или просто по-високи резолюции), голяма част от тях не използват браузърите си на цял екран.

За по-голямата част от проекта интерфейсът е разделен на две колони – лявата 650 пиксела, а дясната 300 пиксела. Дясната колона е

съобразена с това, че най-разпространения рекламен формат в българското интернет пространство са рекламните банери с размер 300x250 точки.

Центрирането на цялата страница е реализирано чрез известна CSS техника, при която в един главен контейнер с фиксирана ширина се побира цялото съдържание, което трябва да бъде в центъра.

Поради факта, че височината на някои страници (като например формата за регистрация и формата за вход) не образува вертикален скролбар, то долният колон титул (наричан още “футър”) се измества нагоре в страницата, а под него остава празно място, което е объркващо за потребителя. Този проблем е преодолян чрез друга CSS техника [30], която представлява това:

```
1 html,
2 body {
3     height: 100%; }
4
5 .wrap {
6     width: 960px;
7     min-height: 100%;
8     height: auto !important;
9     height: 100%;
10    margin: 0 auto -140px; /* negative value is same as #footer 's
height + padding */ }
11
12 #footer {
13     height: 110px;
14     position: relative; }
15
16 .footer-push { height: 110px; /* same height as #footer */ }
```

Фиг. 3.4. CSS за “Sticky footer”.

```
1 <body>
2     <div class="wrap">
3         Main content
4
5         <div class="footer-push"></div>
6     </div><!-- / .wrap -->
7
8     <div id="footer">
9         Footer's content
10    </div><!-- / #footer -->
11 </body>
```

Фиг. 3.5. HTML структура за “Sticky footer”.

По този начин цялото основно съдържание (почти целия сайт) ще бъде поставен в контейнера с клас wrap. Независимо от височината на

съдържанието в този контейнер, `<div>` с `id="footer"` винаги ще се намира най-долу на страницата.

Друг немаловажен момент на този етап от интерфейса е постоянното визуализиране на вертикален скролбар, независимо от количеството съдържание. Тоест, чрез задаването на следното:

```
1 body { overflow-y: scroll; }
```

Фиг. 3.6. CSS свойство за елемента *body*.

По този начин потребителят ще вижда лентата за вертикално превъртане на всяка страница. Така при преход от страница без нужда от скрол към страница със скрол, няма да се получи стесняване на страницата ~20 пиксела и съответно съдържанието няма да създава илюзията, че се размества. Тази техника е използвана в много от уеб приложенията на Google.

3.2.3. Стилизация на полетата за въвеждане на информация

Основната конструкция на форма за попълване се дели на две колони. В лявата се намира името на полето, което е разположено в дясната колона. Под всяко поле с ненаатрапчив по размер и цвят шрифт е обяснено какво точно трябва да бъде съдържанието му – фиг. 3.7.

Въпреки, че говорим за колони, то в `markup` кода въпросната структура е реализирана, чрез контейнери (`<div>`). Основната структура, използвана и във вътрешните страници е следната:

E-mail

Вашата електронна поща.

Фиг. 3.7. Структура на *input* поле.

```
1 <div class="input-holder">
2   <label class="label">E-mail</label>
3   <input type="text" name="email" class="inputText" />
4   <small>Вашата електронна поща.</small>
5 </div>
```

Фиг. 3.8. HTML код за контейнер с *input* поле.

Чрез фиг. 3.9 стилизираме горния HTML (фиг. 3.8) до вида, който е на фиг. 3.7. По този начин тази и всички останали полета за писане, придобиват стила на проекта.

```
1 .input-holder {
2     width: 560px;
3     float: left;
4     padding: 10px 0px;
5     border-bottom: 1px solid #eaeaea;
6     position: relative; }
7
8 .input-holder:last-child {
9     border: 0px;
10    padding-bottom: 0px; }
11
12 .label {
13     width: 90px;
14     margin-top: 8px;
15     padding-right: 10px;
16     float: left;
17     text-align: right;
18     font-size: 13px;
19     font-weight: bold;
20     color: #5a5a5a; }
21
22 .input-holder small {
23     margin-left: 100px;
24     margin-top: 7px;
25     float: left;
26     display: block; }
27
28 .inputText {
29     background: #fff url(../images/input.gif) left top repeat-x;
30     width: 598px;
31     padding: 10px;
32     display: block;
33     float: left;
34     overflow: hidden;
35     border: 1px solid #d0d0d0;
36     font-family: Arial, Tahoma, Verdana, sans-serif;
37     font-size: 12px;
38     -moz-border-radius: 3px;
39     -webkit-border-radius: 3px;
40     border-radius: 3px; }
41
42 .inputText:focus {
43     border: 1px solid #e8da7b;
44     -moz-box-shadow: #e8da7b 0px 0px 5px;
45     -webkit-box-shadow: #e8da7b 0px 0px 5px;
46     box-shadow: #e8da7b 0px 0px 5px; }
```

Фиг 3.9. CSS код за стилизация на input полета.

С добавянето на картинка 1х6 пиксела, повтаряща се по хоризонтала, в съчетание с бял цвят за фон и сива граница (border) на полето, създаваме ефект на вдлъбнатост, който в уеб дизайна е признат за всички видове полета за писане (бутоните, от своя страна, имат ефект на изпъкналост). Ъглите на формите са заоблени чрез CSS3 свойства, което

олекотява визуализацията на формите, защото не са използвани картинки с фиксирани размери (респ. графични файлове с по-големи размери).

На 8-ми ред, от предоставения във фиг. 3.9 код, е използван псевдо-селекторът `:last-child`, който задава специални свойства на последния елемент от дадена група елементи. В случая последния контейнер с клас `input-holder` няма да има долна граница, както и отстояние от долния му край до съдържанието му. Това е направено с цел поставяне на стилизиран бутон за изпращане на данните, под всяка една форма за попълване.

На 42-ри ред от фиг. 3.9 е използван друг псевдо-селектор `:focus`. Свойствата, които са му зададени се изпълняват, само когато потребителят е фокусирал дадения елемент – в този случай, когато курсорът е поставен в полето за въвеждане на информация. Отново чрез CSS3 свойства, то е стилизирано така, че когато потребителят пише, около полето да има жълто “сияние”, което му подсказва в кое поле се намира курсора.

За стилизирането на полета от тип текст и парола, както и за всички `textarea` е използван само CSS. Но не всички видове `input` полета могат да се стилизират по този начин. Radio бутони и `checkbox`, форми за добавяне на файл, както и списъците за избор са стилизирани с помощта на jQuery плъгин наречен Uniform [33] и разпространяващ се под MIT лиценз [12]. Скриптът работи за всички видове елементи, но в настоящия проект е използван единствено за `checkbox`, `radio button`, и `file input` полета, поради факта, че останалите видове елементи могат да бъдат стилизирани само чрез CSS (което означава по-бързо зареждане на страниците).

Някои форми за избор на дата са създадени, с помощта jQuery UI [31] – библиотека с основни интерфейс компоненти, разпространяваща се под MIT лиценз [12]. Българизацията на модула “Календар” е написана от Стоян Кьосев [32].

Публичност ☒ За всички
☐ Само за приятели

Добави снимка

Фиг. 3.10. Стилизирани radio бутони и форма за добавяне на файл.

Град

Фиг. 3.11. Стилизиран select списък.

Дата и час

Май 2011

По	Вт	Ср	Че	Пе	Съ	Не
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Фиг. 3.12. Избор на дата чрез jQuery UI.

3.2.4. Навигация и потребителски панел

След няколко на брой промени, интерфейсът на навигационната лента придоби моментния си вид, който е максимално опростен и интуитивен.

Навигационната лента използва цялата ширина на потребителския браузър, но самите елементи винаги са центрирани в решетката с ширина 960 точки.

Основните бутони в навигацията са, както следва:

1. Favplace Logo – отвежда към началната страница
2. Места – показва каталог с места

3. Събития – показва каталог със събития

4. Помощ – показва раздел за въпроси и отговори, където чрез снимки (а въбдеще и чрез видео) е обяснена функционалността на целия уеб сайт, както и на всеки отделен модул.

5. Ако потребителя се е автентифицирал му се появява още един бутон - Мои места, чрез който с помощта на падащо меню, има достъп до бизнес администрацията на негови места, ако притежава такива.

Бутоните за регистрация и вход в системата са с отличителни цветове, интуитивно подсказващи за наличието и местоположението си.

Когато потребителят се идентифицира, на мястото на бутоните за регистрация и вход се появява панел с аватара и псевдонима на потребителя. Този панел уведомява потребителя за наличие на активност, по някакъв начин свързана с него (например ако друг потребител го е отбелязал (tag) в публикация). За да се запази олекотената и неангажираща навигация, бутоните свързани с менажирането на профила на потребителя се появяват единствено при клик върху полето с псевдонима. По този начин, чрез падащото меню, потребителя има бърз и ненатрапчив достъп до:

1. Известия за активност свързана с него
2. Преглед на профила си
3. Хората, които следи, както и тези, които го следят
4. Страница за настройка на профила
5. Бутон за изход от профила

6. Поле с автоматично допълване на написаното (autocomplete), което е предназначено за лесна смяна на текущата локация на потребителя, за да може да му бъде предлагано съдържание, което е релевантно на моментното му физическо местонахождение. Полето е вмъкнато в този панел, за да бъде максимално бързо и лесно за достъпване и промяна.

Markup кода и PHP скрипта за визуализация на падащото меню на бутона “Мои места” са показани на фиг. 3.13:

```

1 <? if ($this->user_logged): ?>
2 <li>
3     <a id="my-places-btn"><b>Мои Места</b></a>
4     <div id="my-places-box" class="shadow">
5
6         <? foreach ($owner_places->result() as $owner->place): ?>
7             <a href="place/<?=$owner_place->id;
?>/<?=$url_title($owner_place->name); ?>" class="place">
8                 name); ?>" />
9                 <b>Place name</b>
10            </a>
11        <? endforeach; ?>
12
13        <a href="help/business" class="btn30">Нямаш места?</a>
14    </div><!-- / #my-places-box -->
15 </li>
16 <? endif; ?>

```

Фиг. 3.13. Падащо меню “Мои места”.

Контейнерът с id="my-places-box" по подразбиране е скрит. Когато се кликне на върху линка с id="my-places-btn" се задейства следната jQuery функция:

```

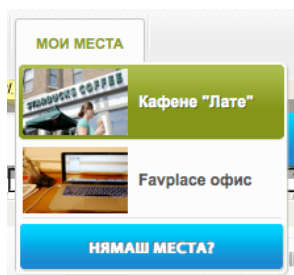
1 $(document).ready(function () {
2     ...
3     /*-----
4         My places - dropdown
5     -----*/
6     $("#my-places-btn").click(function () {
7         if(!($("#my-places-btn").hasClass('act')) {
8             $("#my-places-btn").addClass("act");
9         } else {
10             $("#my-places-btn").removeClass("act");
11         }
12
13         $('#my-places-box').slideToggle('fast');
14     });
15     ...
16 });

```

Фиг. 3.14. jQuery скрипт за падащото меню “Мои места”.

Ако бутонът #my-places-btn е активен (т.е. няма class="act"), то такъв му се добавя и обратното. След това се задейства функцията slideToggle(), която разпъва контейнера #my-places-box.

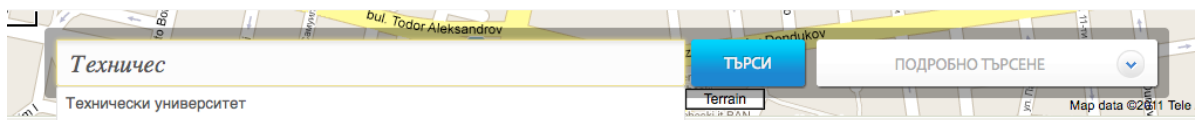
По подобен начин е реализирано и падащото меню за бутоните за потребителския профил, както и това за подробното търсене.



Фиг. 3.15. Падащо меню със статус *hover* на първия елемент.

3.2.5. Панел за търсене

Панелът за търсене се намира под навигационната лента. Той е разположен върху активна Google карта. Полето за търсене е сравнително голямо, лесно забележимо. Чрез jQuery UI е интегриран модул за автоматично довършване на фразите (autocomplete) - фиг. 3.16.



Фиг. 3.16. jQuery UI Autocomplete.

Когато потребителят фокусира полето за въвеждане на текст, стойността по подразбиране (“Какво търсиш?”) се изтрива и потребителят може директно да започне да пише. Когато не въведе нищо, но разфокусира полето, то отново придобива стойността по подразбиране. Това е реализирано, чрез кода от фиг 3.17.

```

1 $(document).ready(function () {
2   ...
3   /*-----
4       Main Search Field - clear value on focus
5   -----*/
6   $('#mainSearch').focus(function() {
7     if (this.value == this.defaultValue) {
8       this.value = '';
9     }
10  });
11
12  $('#mainSearch').focusout(function() {
13    if (this.value == '') {
14      this.value = this.defaultValue;
15    }
16  });
17  ...
18 });

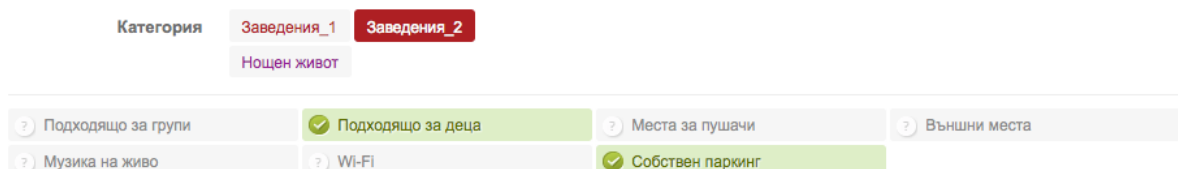
```

Фиг 3.17. jQuery за изчистване стойността на поле.

Бутонът “Подробно търсене” работи с функция подобна на тази от фиг. 3.14. При клик се разпъва форма за подробно търсене, чрез която

потребителят може да зададе търсене по град, категория или други критерии, направени на принципа на таговете.

Категориите и таговете всъщност са `checkbox` елементи, но са стилизирани, за да изглеждат по-изчистено. На фиг. 3.18 категорията “Заведения_2” е маркирана, както и таговете “Подходящо за деца” и “Собствен паркинг”.



Фиг. 3.18. Стилизирани `checkbox` елементи.

Подмяната на елементите е реализирана по следния начин:

```
1 <a href="#h_tags-1" class="fakecheck">Подходящо за деца</a>
2 <input type="checkbox" id="h_tags-1" value="1" />
```

Фиг. 3.19. HTML за стилизирани `checkboxes`.

```
1 $(document).ready(function () {
2   ...
3   /*-----
4     Fakechecks
5   -----*/
6   // check for what is/isn't already checked and match it on the
fake ones
7   $("input:checkbox").each(function() {
8     if((this.checked)) {
9       $("#fake"+this.id).addClass('fakechecked')
10    } else {
11      $("#fake"+this.id).removeClass('fakechecked');
12    }
13  });
14
15  // function to 'check' the fake ones and their matching
checkboxes
16  $("a.fakecheck").click(function(){
17    if(($this).hasClass('fakechecked')) {
18      $(this).removeClass('fakechecked')
19    } else {
20      $(this).addClass('fakechecked');
21    }
22    $(this.hash).trigger("click");
23    return false;
24  });
25  ...
26 });
```

Фиг. 3.20. `jQuery` скрипт за стилизирани `checkboxes`.

Чрез задаване на клас на контейнера, в който се намира кодът от фиг. 3.19 стилизираме всички линкове в него (през CSS), а `input` полетата

скриваме посредством `display: none`. С JavaScript от фиг. 3.20 проверяваме всички `input` полета от тип `checkbox` и ако са отбелязани (`checked="checked"`), то линкът (чийто анкор (`#something`) съвпада с `id` на скритото `input` поле) получава клас `fakechecked`. Тоест с кода от фиг. 3.20 добавяме и премахваме класове на линковете, като въпросните класове са стилизирани през CSS, а истинските `input` елементи остават скрити, но свързани с линковете, променящи състоянието им (`checked`) чрез клик върху тях (линковете).

3.2.6. Модул Активност

Формата за въвеждане на публикация е реализирана чрез `textarea` и `div`, скрит под нея. Потребителят пише във въпросната `textarea`, а нейното съдържание се копира в контейнера под нея, чийто текст е прозрачен, но позицията му напълно съвпада с тази на текста отгоре. Това е направено с цел отличаване на текста, който ще бъде превърнат в линк след публикуване на поста. Когато потребителят напише "@" ще му се отвори допълнителна форма, която има `autocomplete` функция. Там потребителят може да напише псевдонима на свой последовател (приятел) или пък името на дадено място. Например ако потребителя въведе "В момента с @hristo сме в @Бар Петък", то "hristo" и "Бар Петък" ще се оградят в `` таг в контейнера (`div`). Таговете `` имат зададен жълт фон. Реализацията и стилизацията на `markup` кода е демонстрирана във фиг. 3.21 и фиг. 3.22, а резултатът е показан с изображение във фиг. 3.23.

```
1 <div class="post-form-box">
2   <div class="txt-fake br4" id="comment_rich"></div>
3   <textarea id="comment_text" class="flexible-cbox br4"></textarea>
4 </div>
5
6 <div class="post-form-box">
7   <input id="comment_autocomplete" class="inputText" type="text" />
8   <input type="hidden" name="comment_json" />
9 </div>
```

Фиг. 3.21. HTML за формата за статуси.

```
1 /* Fake div */
2 #comment_rich {
3     background: #fff;
4     width: 600px;
5     *width: 602px; /* IE fix */
6     _width: 602px; /* IE fix */
7     height: 20px;
8     float: left;
9     padding: 11px 0 10px 11px;
10    overflow: hidden;
11    position: relative;
12    z-index: 1;
13    color: transparent;
14    letter-spacing: 0px;
15    line-height: 1.5em;
16    font-size: 12px;
17    font-family: Arial, sans-serif !important;
18    font-weight: normal;
19    vertical-align: text-bottom;
20    white-space: pre-wrap;
21    word-wrap: break-word;
22    /*color: red; // debug */ }
23
24 @media screen and (-webkit-min-device-pixel-ratio:0)
25 {
26     #comment_rich { width: 590px; } /* Webkit browsers fix */
27 }
28
29 /* Mentioned */
30 #comment_rich b {
31     background: #FBFFBF;
32     font-weight: normal;
33     -moz-border-radius: 2px;
34     -webkit-border-radius: 2px;
35     border-radius: 2px;
36     text-shadow: #fff 1px 1px 0px; }
37
38 /* Real textarea */
39 #comment_text{
40     background: url(../images/input.gif) left top repeat-x;
41     width: 590px;
42     height: 10px;
43     padding: 10px;
44     overflow: hidden;
45     position: absolute;
46     z-index: 2;
47     left: -1px;
48     border: 1px solid #d0d0d0;
49     letter-spacing: 0px;
50     line-height: 1.5em;
51     font-size: 12px;
52     font-family: Arial, sans-serif;
53     font-weight: normal;
54     vertical-align: text-bottom;
55     white-space: pre-wrap;
56     word-wrap: break-word;
57     color: #555; }
```

Фиг. 3.22. CSS за *textarea* и *div* елемента под нея.

В момента с yordanoff пишем книжното тяло на дипломната работа близо до @

НД

НДК

Фиг. 3.23. Textarea с отличаване на линковете.

3.3. Разработване на модул Бизнес

3.3.1. Заявка за Бизнес версия

На страниците за преглед на място се появяват рекламни полета, които представят идеята за бизнес версията – фиг. 3.24.

Starbucks II

ИНФО ГАЛЕРИЯ МЕНЮ СЪБИТИЯ КАРТА ФЕНОВЕ За собственици

Заведния
Кафенета, Дискотеки

Град София
бул. "Лорем ипсум"

www.starbucks.bg
02 / 987 87 77

Отворено: Пон-Нед, 11ч - 24ч
Цени: 10-20лв / човек
Паркинг: Собствен

42
Аз: 4 пъти
ТУК СЪМ

13
Аз: 22 Сен, 22:30
ОТИВАМ

8
Аз: (?)
СУПЕР Е

For those of you who have long been craving a sip of your favourite Starbucks coffee, the wait is finally over. You can already get your cup in the very centre of Sofia, at 62 General Yossif Gurko Street, right on the corner with Vassil Levski Boulevard. "We are proud that with the opening of our first shop in Bulgaria we will present Starbucks Coffee and the unique. [Чети нататък »](#)

Подходящо за групи: ✓
Подходящо за деца: ✗
Места за пушачи: ✓

Външни места: ✗
Музика: На живо
Wi-Fi: ✓

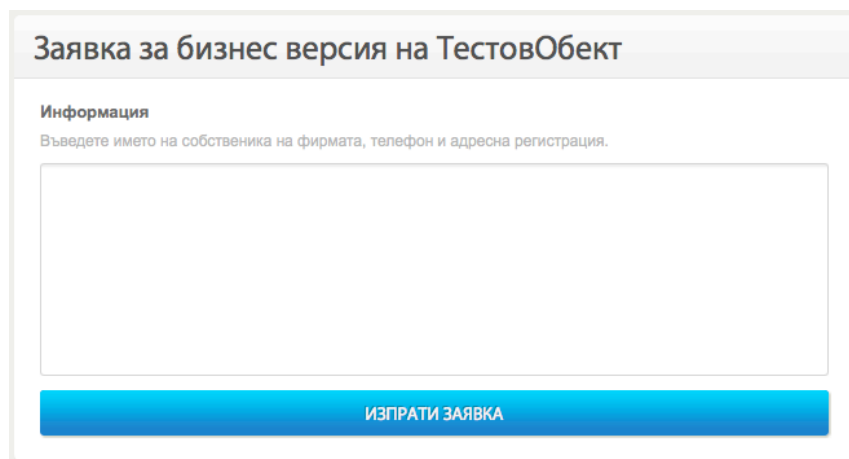
Ако това място е Ваше или просто работите тук,
ОТКЛЮЧЕТЕ БИЗНЕС ВЕРСИЯТА
Тя Ви предоставя редица страхотни възможности за управлението на вашето място.

НАУЧИ ПОВЕЧЕ

Favplace
BUSINESS

Фиг. 3.24. Преглед на място с промотиране на бизнес версията.

След като потребителят прегледа функционалностите, които му предлага разширената версия, той може да заяви, че е собственик на мястото, като попълни формата от фиг. 3.25.



Фиг. 3.25. Заявка за бизнес функционалност.

Реализацията на РНР функцията, обработваща информацията, въведена в textarea от фиг. 3.25 е показана на фиг. 3.26.

```
1 function request($place_id) {
2     $place = $this->Place_model->get($place_id);
3     if (!$place) redirect("");
4
5     $rules = array(
6         array("field" => "request", "rules" =>
7             "trim|required|max_length[5000]")
8     );
9     $this->form_validation->set_rules($rules);
10
11     if ($this->form_validation->run()) {
12         $this->Business_model->send_request($place_id);
13     }
14
15     $data["place"] = $place;
16     $this->load->view('business/request', $data);
17 }
```

Фиг. 3.26. Извадка от контролер business.php.

SQL заявката е показана на фиг. 3.27.

```
1 function send_request($place_id) {
2     $this->db->query("INSERT INTO place_request (place_id, content, date)
3         VALUES (?, ?, now())", array($place_id, $this->input-
4     >post("request")));
5 }
```

Фиг. 3.27. Извадка от модел business_model.php.

3.3.2. Добавяне на елементи в модул Меню

Менютата се състоят от няколко основни части - главна категория, подкатегория и артикули. Артикулите, от своя страна, имат заглавие, описание, количество и цена.

Главните категории се изпращат от формата като масив. Името на подкатегориите към дадена категория се образува `sub_cat_{индекс на главната категория}`. Това се прави за да се асоциира всяка подкатегория към дадена категория. Артикулите и имената им в HTML формата се образуват `item_{индекс на главната категория}_{индекс на подкатегорията}`. По същия начин се образуват и описанието, количеството и цената.

В контролера `business.php` създаваме функция `menu_edit`, която обработва данните от HTML формата.

Първо проверяваме дали текущият потребител е собственик на мястото (ред 4, фиг. 3.28). Ако е, създаваме масив с правила за валидация на формата и ги прилагаме. If конструкцията на ред 13 проверява дали формата е валидирана коректно и ако е – продължаваме с добавянето.

Взимаме главните категории и ги обхождаме и записваме в БД с цикъл `foreach`. След това взимаме всички подкатегории на текущата главна категория и ги записваме в БД. Накрая обикаляме всички артикули на текущата подкатегория и ги записваме в БД. Преди да запишем артикулите в базата данни обаче, чрез функцията `add_menu_item` от модела `business_model.php` (ред 15, фиг. 3.29), проверяваме дали всички полета са попълнени (условната конструкция на ред 33, фиг. 3.28).

```

1 function menu_edit($place_id)
2 {
3     // if the logged user is not the owner, exit
4     if (!$this->Business_model->is_place_owner($place_id))
5         redirect("");
6
7     $rules = array(
8         array("field" => "main_cat[]", "rules" =>
9             "trim|required|max_length[1000]")
10    );
11
12    $this->form_validation->set_rules($rules);
13
14    if ($this->form_validation->run()) {
15        // add category if any
16        $main_cats = $this->input->post("main_cat");
17        foreach ($main_cats as $key => $main_cat) {
18            $main_cat_id = $this->Business_model-
19>add_menu_category($place_id, $main_cat);
20
21            // add subcategory if any
22            $sub_cats = $this->input->post("sub_cat_$key");
23            if (!sizeof($sub_cats))
24                continue;
25
26            foreach ($sub_cats as $key2 => $sub_cat) {
27                $sub_cat_id = $this->Business_model-
28>add_menu_subcategory($main_cat_id, $sub_cat);
29
30                // get item arrays
31                $sub_cat_items = $this->input->post('item_' . $key . '_' .
32                $key2);
33                $sub_cat_amounts = $this->input->post('amount_' . $key .
34                '_' . $key2);
35                $sub_cat_prices = $this->input->post('price_' . $key . '_'
36                . $key2);
37                $sub_cat_descriptions = $this->input->post('description_'
38                . $key . '_' . $key2);
39
40                if (sizeof($sub_cat_items) == sizeof($sub_cat_amounts) &&
41                    sizeof($sub_cat_items) == sizeof($sub_cat_prices)
42                    &&
43                    sizeof($sub_cat_items) ==
44                    sizeof($sub_cat_descriptions) &&
45                    sizeof($sub_cat_items) != 0) { // check if size of
46                    arrays match and it's not zero
47                        foreach ($sub_cat_items as $key3 => $item) {
48                            $amount = $sub_cat_amounts[$key3];
49                            $price = $sub_cat_prices[$key3];
50                            $description = $sub_cat_descriptions[$key3];
51
52                            $this->Business_model->add_menu_item($sub_cat_id,
53                            $item, $amount, $price, $description);
54                        }
55                    }
56                }
57            }
58            redirect("business/edit/$place_id");
59        }
60        $this->load->view('business/menu/edit');
61    }
62 }

```

Фиг. 3.28. Част от контролера *business.php*.

```

1 function add_menu_category($place_id, $main_cat) {
2     $this->db->query("INSERT INTO place_menu_categories (place_id, name,
date)
3         VALUES (?, ?, now())", array($place_id, $main_cat));
4
5     return $this->db->insert_id();
6 }
7
8 function add_menu_subcategory($main_cat_id, $sub_cat) {
9     $this->db->query("INSERT INTO place_menu_subcategories (category_id,
name, date)
10        VALUES (?, ?, now())", array($main_cat_id, $sub_cat));
11
12    return $this->db->insert_id();
13 }
14
15 function add_menu_item($sub_cat_id, $item, $amount, $price,
description) {
16     $this->db->query("INSERT INTO place_menu_items
17         (subcategory_id, name, amount, price, description, date)
18         VALUES (?, ?, ?, ?, ?, now())", array($sub_cat_id, $item, $amount,
$price, $description));
19
20     return $this->db->insert_id();
21 }

```

Фиг. 3.29. Част от модела от *business_model.php*.

Пици		
Вегетариански		
Капричоза	600 гр	8.50 лв
пилешко филе кашкавал синьо сирене сметана заеленчуци, пилешко филе кашкавал синьо сирене сметана заеленчуци, пилешко филе кашкавал синьо сирене сметана заеленчуци		
Пеперони	600 гр	8 лв
пилешко филе, кашкавал, синьо сирене, easter egg, сметана, заеленчуци		
Калцоне	600 гр	7.80 лв
пилешко филе кашкавал синьо сирене сметана заеленчуци сметана		

Фиг 3.30. Изглед на меню.

ПРЕДЯСТЯ		
Аламинут		
Пържени картофи	250 гр	2.30 лв
Пържени картофи със сирене, подправки и други		

Фиг 3.31. Форма за добавяне или редакция на меню.

3.3.3. Модул за промяна на външния вид на места

Собствениците на места имат възможността да променят външния вид на страниците на обектите си. Чрез JavaScript инструмента, показан на фиг. 3.36, собствениците могат да променят цветовете, шрифтовете, сенките и т.н. в страницата на мястото им.

Чрез jQuery UI са създадени плъзгачи, които помагат на потребителя да настройва размери и отстояния (кодът от фиг. 3.32). Отново с помощта на jQuery UI е интегриран инструмент за избор на цветове (colorpicker).

```
1 $('#border-rounder').slider({
2   values: [8],
3   min: 0,
4   max: 20,
5   slide: function(event, ui) {
6       cssStuff['borderRadius'] = ui.value + "px";
7       createCSS();
8   }
9 });
```

Фиг. 3.32. JavaScript UI slider.

Когато потребителят премести плъзгача, се извиква функцията `createCSS()`, която пресъздава стиловете на елементите, които могат да се променят, взимайки стойността от конкретния плъзгач, списък за избор или текстово поле. CSS стиловете се съдържат в JavaScript масив.

Промените се изпращат до БД, чрез AJAX заявката от фиг. 3.33.

```
1 function saveStyle(place_id)
2 {
3     $.ajax({
4         type: "post",
5         url: "business/customization/"+place_id,
6         dataType: 'json',
7         data: {
8             "css":$("#customization-style").val()
9         },
10        success: function(data) {},
11        error: function (XMLHttpRequest, textStatus, errorThrown)
12        {
13            alert("error: "+errorThrown);
14        }
15    });
16 }
```

Фиг. 3.33. AJAX заявка.

AJAX заявката изпраща необходимите данни към сървъра, който изпълнява функцията от фиг. 3.34.

```
1 function customization($place_id) {
2     if (!$place_id)
3         redirect("");
4
5     $this->load->model("Place_model", "", TRUE);
6     $place = $this->Place_model->get($place_id);
7
8     if (!$place)
9         redirect("");
10
11     $rules = array(
12         array("field" => "css", "rules" => "trim|required")
13     );
14
15     $this->form_validation->set_rules($rules);
16
17     if ($this->form_validation->run()) {
18         $this->Business_model->save_style($place_id);
19         echo json_encode(array('status' => 'ok'));
20     } else {
21         $data['place'] = $place;
22         $data['place_favorites_count'] = $this->Place_model-
>get_favorites_count($place_id);
23         $data['place_tags'] = $this->Place_model-
>get_place_tags($place_id);
24         $data['customization'] = $this->Place_model-
>get_customization($place_id);
25
26         $this->load->view('business/customization/index', $data);
27     }
28 }
```

Фиг. 3.34. Извадка от контролер *business.php*.

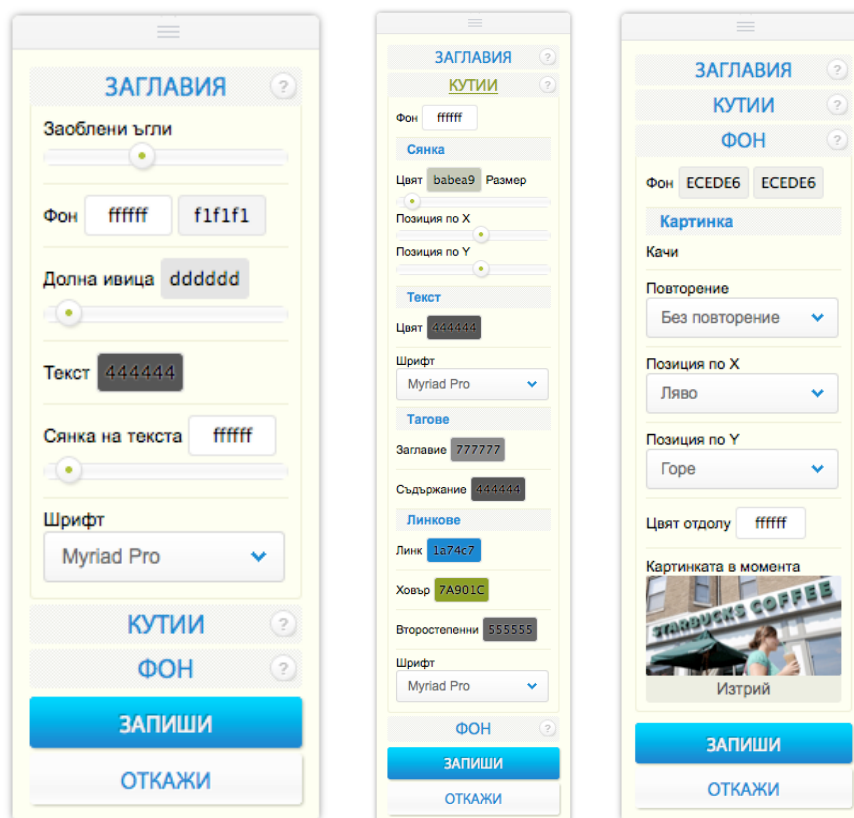
Заявките към БД са извадени във фиг. 3.35.

```

1 function save_style($place_id)
2 {
3     $query = $this->db->query("SELECT id FROM place_customization WHERE
place_id=?",
4     array($place_id));
5
6     if($query->row())
7     {
8         $customization_id = $query->row()->id;
9     }
10    else
11    {
12        $customization_id = 0;
13    }
14
15    if($customization_id)
16    {
17        $this->db->query("UPDATE place_customization SET css=? WHERE
id=?",
18        array($this->input->post("css"), $customization_id));
19    }
20    else
21    {
22        $this->db->query("INSERT INTO place_customization (place_id,
css, date) VALUES(?, ?, now())",
23        array($place_id, $this->input->post("css")));
24    }
25 }

```

Фиг 3.35. Извадка от модел *business_model.php*.



Фиг. 3.36. Инструменти за промяна на дизайна.

3.4. Реализация на модул Регистрация

Страницата със съответния модул е разделена на две колони. В лявата се намират формите за попълване, а в дясната, чрез jQuery скрипт, се ротират някои от основните функционалности на сайта. По този начин, докато се регистрира, потребителят ще може да прегледа защо и как точно би могъл да му бъде полезен сайта.

```
1 function index() {
2     $this->load->model("Signup_model", "", TRUE);
3
4     $rules = array(
5         array('field' => 'username', 'rules' =>
6             'alpha_numeric|required|min_length[3]|max_length[18]|callback__is_username_
7             free'),
8         array('field' => 'email', 'rules' =>
9             'trim|required|valid_email|callback__is_email_free'),
10        array('field' => 'password', 'rules' =>
11            'required|min_length[6]|max_length[50]'),
12        array('field' => 'password_r', 'rules' =>
13            'required|min_length[6]|max_length[50]|matches[password]'),
14    );
15
16    $this->form_validation->set_rules($rules);
17    if ($this->form_validation->run()) {
18        if ($this->Signup_model->register()) {
19            $this->load->view('signup/signup_success');
20        }
21        else
22            $this->load->view('signup/signup_error');
23    }
24    else {
25        $this->load->view('signup/signup');
26    }
27 }
28
29 // callback function
30 function _is_username_free() {
31     $this->load->model("Signup_model", "", TRUE);
32     $this->form_validation->set_message('_is_username_free', 'Този
33     псевдоним е зает, избири друг.');
```

```
34     return $this->Signup_model->is_username_free($this->input-
35     >post("username"));
36 }
37
38 // callback function
39 function _is_email_free() {
40     $this->load->model("Signup_model", "", TRUE);
41     $this->form_validation->set_message('_is_email_free', 'Вече има
42     регистрация с този email.');
```

```
43     return $this->Signup_model->is_email_free($this->input-
44     >post("email"));
45 }
```

Фиг. 3.37. Извадка от контролер *signup.php*.

```
1 function register()  
2 {  
3     $this->db->query(  
4         "INSERT INTO profiles (username,password,email,registered,status)  
VALUES(?,md5(??),?,now(),0)",  
5         array($this->input->post("username"),$this->input->  
>post("password"),$this->input->post("email")));  
6  
7         return $this->db->insert_id();  
8 }
```

Фиг. 3.38. Извадка от модел *signup_model.php*.

Преди регистрацията, PHP скриптът проверява серия от правила за всяко едно от полетата (ред 4, фиг 3.37). Най-първо дали потребителското име е само букви и цифри, и дали е свободно, след това дали вече няма регистрация с въведения email. Накрая се проверява дали паролата и нейното потвърждение съвпадат. При SQL заявката за добавяне на регистрацията, полето status се въвежда да е 0, което означава че регистрацията все още не е потвърдена.

The image shows a web registration form titled "Присъедини се" (Join). It contains four input fields with labels and validation hints:

- Псевдоним** (Pseudonym): A text input field with a hint "Латински букви и цифри. От 3 до 18 символа." (Latin letters and digits. 3 to 18 symbols).
- E-mail**: A text input field with a hint "Вашата електронна поща." (Your email).
- Парола** (Password): A text input field with a hint "От 6 до 50 символа." (6 to 50 symbols).
- Парола, отново** (Password, again): A text input field with a hint "Потвърдете паролата си." (Confirm your password).

At the bottom left, there are links for "Вход" (Login) and "Забравена парола" (Forgot password). At the bottom right, there is a blue button labeled "РЕГИСТРАЦИЯ" (Registration).

Фиг. 3.39. Форма за регистрация.

Ч Е Т В Ъ Р Т А Г Л А В А

РЪКОВОДСТВО НА ПОТРЕБИТЕЛЯ

4.1. Изисквания на програмния продукт

- Apache версия 2.0 или по-нова с модула `mod_rewrite`.
- PHP версия 5.1.6 или по-нова.
- MySQL версия 5.0 или по-нова.
- Съвременен браузър – Firefox 3.5, Internet Explorer 7, Safari 4, Chrome, Opera 10 (приложението не работи при Microsoft Internet Explorer 6).

4.2. Инсталация на програмния продукт

- Качете цялото съдържанието на диска във `www/` директорията на сървъра.
- Отворете `/application/config/config.php` и заместете <http://localhost/> с адреса на който се намира Favplace:
`$config['base_url'] = http://mysite.com/`
- Създайте база данни за Favplace, както и MySQL потребител, който има пълни привилегии за достъпа и редактирането ѝ.
- Отворете `/application/config/database.php` и въведете вашите настройки за базата данни:
 - Хост:
`$db['default']['hostname'] = "localhost";`
 - Потребител:
`$db['default']['username'] = "favp_user";`
 - Парола:
`$db['default']['password'] = "123456";`
 - Име:
`$db['default']['database'] = "favplace";`

- Направете import на файла favplace.sql в root директорията на диска, за да се създадат таблиците в базата данни.
- Уверете се че папките /www/avatars и /www/pictures са с права за писане.

4.3. Използване на социалната мрежа

4.3.1. Регистрация

За да бъде използван програмния продукт пълноценно, потребителят трябва първо да се регистрира. За това е нужно да избере уникален за уеб сайта псевдоним, да посочи валиден email, на който ще му бъде изпратен линк за потвърждение на регистрацията, както и парола. Под всяко едно поле има кратко обяснение как трябва да се попълни, както на фиг. 4.1:

Присъедини се

Псевдоним
Латински букви и цифри. От 3 до 18 символа.

E-mail
Вашата електронна поща.

Парола
От 6 до 50 символа.

Парола, отново
Потвърдете паролата си.

[Вход](#) · [Забравена парола](#)

РЕГИСТРАЦИЯ

Фиг. 4.1. Регистрация на потребители.

4.3.2. Вход

След като потребителя се регистрира и потвърди регистрацията си, трябва да използва формата за вход, за да се идентифицира и да получи достъп до пълната функционалност на уеб сайта. Това става чрез въвеждане на потребителско име и парола.

4.3.3. Редакция на профил

Чрез редакцията на профил, може да се попълни по-подробна информация за профила на потребителя (истинско име, град, рождена дата и други), като по този начин приятелите му ще го намерят по-лесно.

Може да се посочи линк към профила в Twitter, Facebook и линк към собствения сайт на потребителя, ако има такъв.

По подразбиране уеб сайта се опитва да намери потребителския gravatar, използвайки имейл адреса, въведен при регистрацията. Ако потребителя няма gravatar или не иска да използва такъв, то той може да качи свой собствен аватар от формата на фиг. 4.2.

Фиг. 4.2. Добавяне на аватар.

Има възможност и за промяна на паролата и e-mail, който е даден при регистрацията. Промяната на паролата става, след като се въведе първо старата парола – фиг. 4.3.

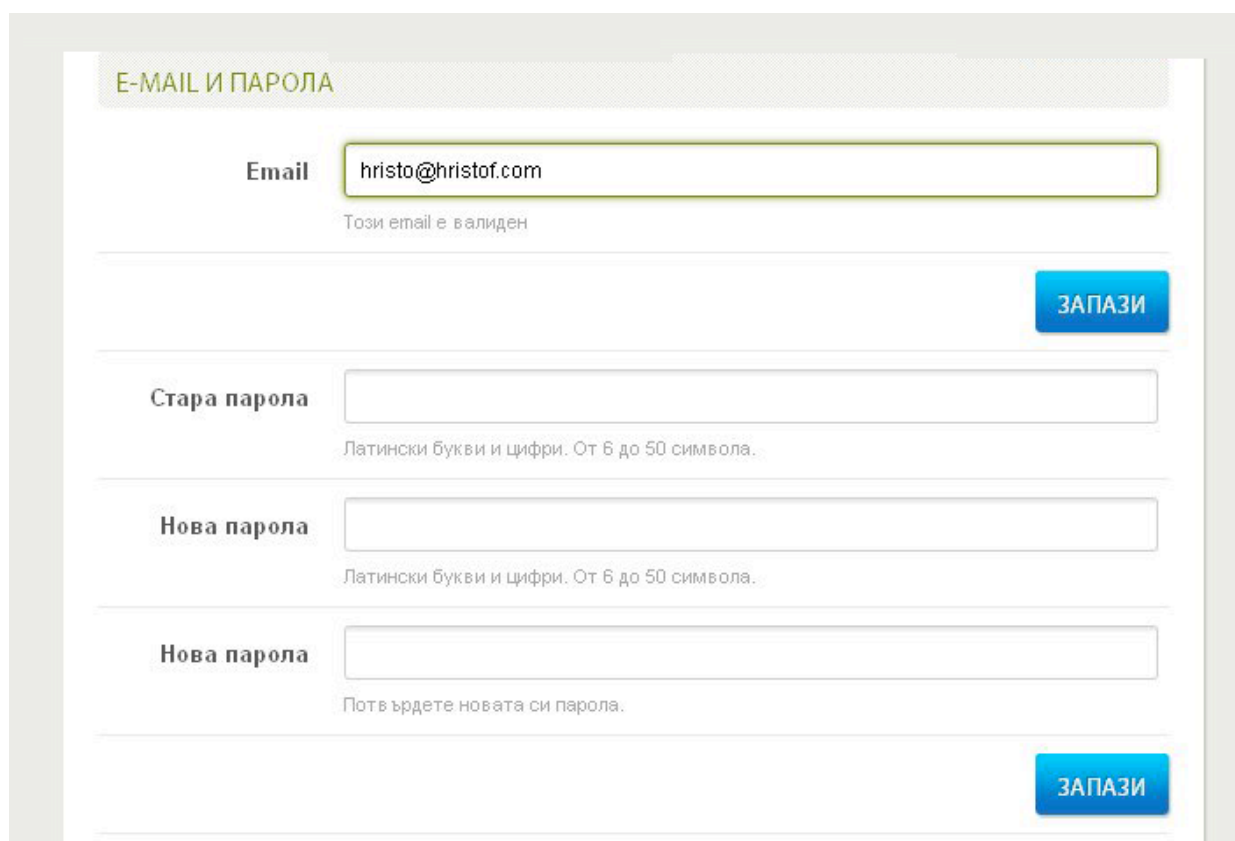
4.3.4. Търсене на места

Бързото търсене на място, което ще се използва най-често е търсене по име или адрес. Това може да стане от хедъра във всяка една страница

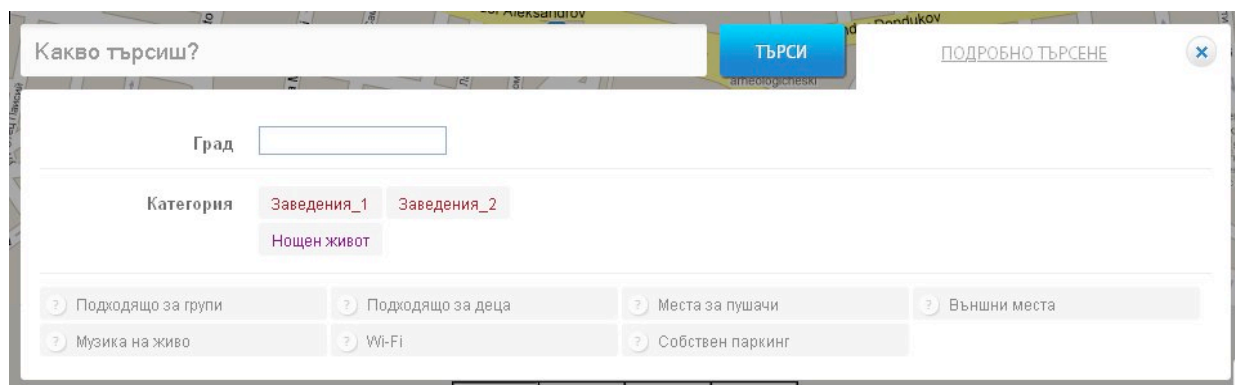
на уеб сайта. Още щом потребителя започне да въвежда какво търси, излизат предложения за това какво може би иска да напише.

Подробното търсене (фиг. 4.4) се зарежда при клик върху линка „Подробно търсене”. От там потребителя може да избере града, в който иска да търси, категориите места, които го интересуват, както и някои по-специфични особености на местата, като например дали са подходящи за пушачи и има ли WiFi.

След като приключи с въвеждането на търсените от него критерии, потребителят трябва да кликне бутона „Търси”.



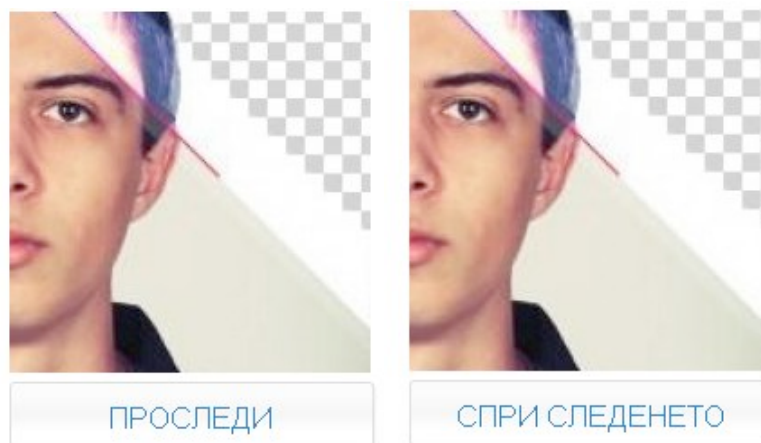
Фиг. 4.3. Промяна на email и парола.



Фиг. 4.4. Подробно търсене.

4.3.5. Проследяване на потребител

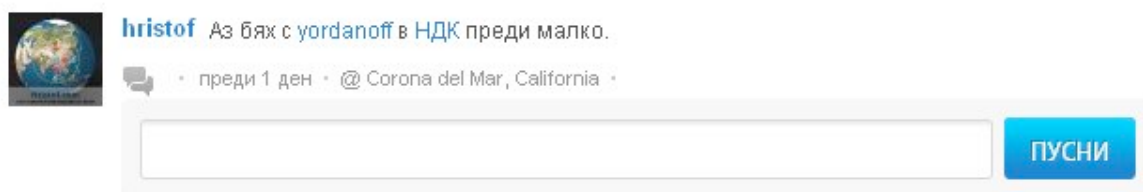
В профилната страница на всеки потребител има бутон “Проследи” или „Спри следенето”. Когато потребител следи друг потребител, то първия може да види активността на следения от него, като например къде последно е бил, какви снимки е качил и други. Съответно „Спри следенето” ще спре показването на активността на съответния потребител.



Фиг. 4.5. Проследяване на потребител.

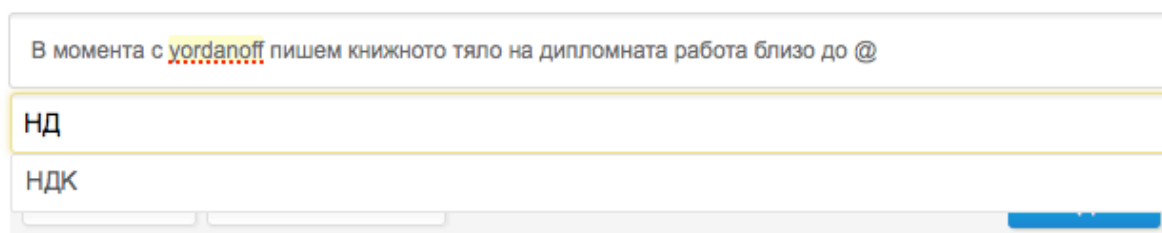
4.3.6. Използване на стената на началната страница

Стената показва различните видове потребителска активност, както на текущия потребител, така и на хората, които той следи. Там се появяват последните отбелязвания на хората, следени от потребителя, последните снимки качени от тях, събитията, на които ще присъстват и други. Също така се отбелязват и споменаванията на други потребители в дадени публикации. Например, ако някой напише на своята стена: „Запознах се с @user1 днес.”, то това ще се покаже и на стената на потребител с псевдоним *user1*. Също така, всяка една активност може да бъде коментирана.



Фиг. 4.6. Публикация със заредена форма за коментар.

За да обнови статуса си, потребителя трябва да използва формата, която е най-горе на стената на началната страница. За да спомене някой потребител, достатъчно е да напише знака „@”, и ще се покаже ново поле, в което трябва да напише псевдонима на потребителя или името на мястото, което иска да спомене. След като избере какво иска да спомене, то ще бъде автоматично добавено в полето за статуса. Накрая трябва само да се кликне бутона “Сподели”.



Фиг. 4.7. Публикуване на статуси, снимки, местоположение.

4.3.7. Добавяне на място

При добавяне на място, има няколко задължителни полета. Първо трябва да се попълни име на мястото. След това се пише адреса на мястото, като в същото време излизат предложения за адреса, който потребителя може би иска да напише. Другият вариант е потребителят да намери мястото на картата и да кликне върху него. Тогава адресът се попълва автоматично. Последното задължително нещо е да се избере поне една категория, към която спада това място.

След това следват няколко незадължителни полета, като: описание на мястото, работно време, особености на мястото (места за пушачи, WiFi) – фиг. 4.9.

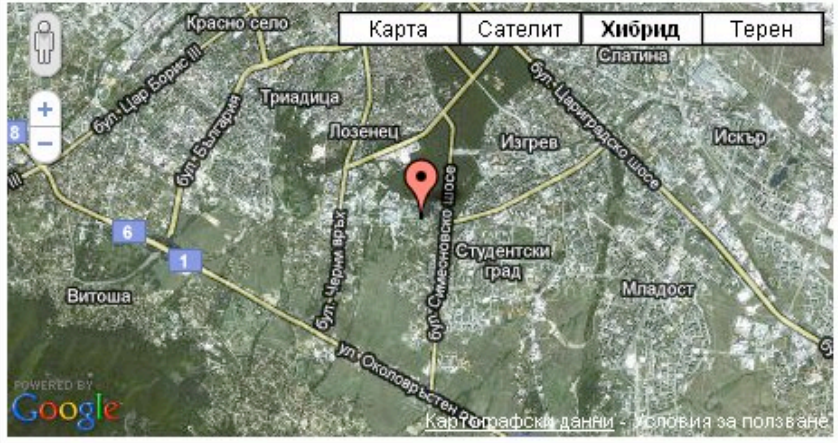
После потребителят, ако желае, има възможност да качи снимки за мястото.

ОСНОВНА ИНФОРМАЦИЯ

Име

Пример: Дон Домат 3 · Starbucks бул. Васил Левски · Fibank Младост

Адрес



Пример: гр. София, бул. България 32

Категория

Заведения_1

Заведения_2

Нощен живот

Фиг. 4.8. Въвеждане на име, избор на адрес и категория за мястото.

ДОПЪЛНИТЕЛНА ИНФОРМАЦИЯ

Описание

Работно време

Пример: Понеделник - Неделя, 11ч - 24ч

☒ Подходящо за групи

☒ Подходящо за деца

☒ Места за пушачи

☒ Външни места

☒ Музика на живо

☒ Wi-Fi

☒ Собствен паркинг

Фиг. 4.9. Въвеждане на описание, работно време и описателни тагове.

4.3.8. Добавяне на събитие

За да се създаде събитие, трябва или да се избере линка „Създай събитие” от страницата с листинг на събитията, или да се избере същия линк от страницата на някое място. След това се въвежда име на събитието. После се избира дата за начало, като се кликне в полето, или се напише ръчно във формат *ден от седмицата, ден.месец.година*. Ако е необходимо може да бъде избрана и дата за край на събитието, като се кликне върху линка „Добави край”, в дясно от полето за избор на начална дата. Ако потребителя идва от страницата на някое място, местоположението на събитието е сложено по подразбиране на това място. Но ако не идва от там, или желае да промени мястото, това може да стане от линка „Избери място”. Така ще се покаже карта и поле за въвеждане на име на мястото със autocomplete.

По подразбиране събитието е видимо за всички, но може да бъде направено видимо само за тези, които следят съответния потребител. По желание може да бъде въведено по-подробно описание и да бъде качена снимка, която да се показва в листинга на събитията.

Създай събитие

Какво?
Пример: D'n'B парти

Кога? 12:00 [Добави край](#)

Къде? ☒ НДК [Избери място](#)

Още инфо?

Публичност ☒ За всички ☐ Само за приятели

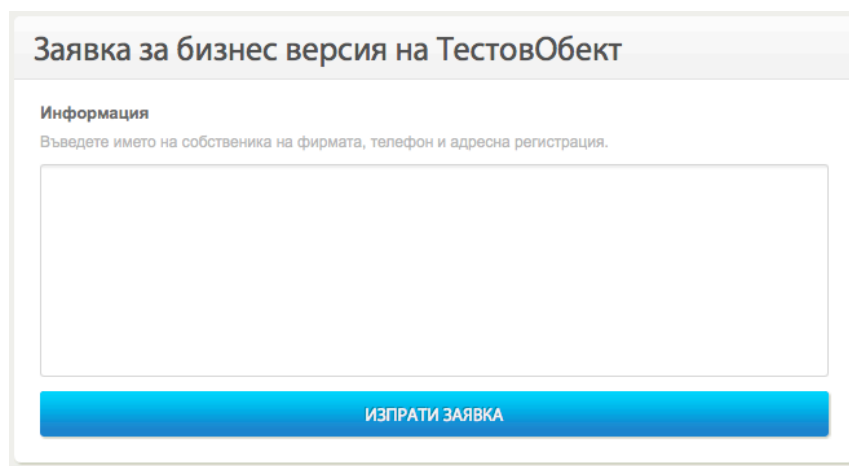
Добави снимка [Избери](#)

СЪЗДАЙ

Фиг. 4.10. Създаване на събитие.

4.3.9. Идентификация на собственици на места

За идентификация на потребителя като собственик на дадено място, той трябва да въведе част от информацията, която е използвана при регистрацията на фирмата му в търговския регистър [34] и е публично достъпна. След проверка в търговския регистър (извършена от администратор), потребителят може да бъде назначен като собственик на мястото, като по този начин ще получи достъп до бизнес функционалностите.



Фиг. 4.11. Заявка за бизнес функционалност.

4.3.10. Бизнес модул Меню

Менюто в началото съдържа една главна категория, една подкатегория и един продукт. Потребителят попълва информацията в съответните полета. Чрез бутон за добавяне на допълнителни елементи може динамично да увеличи броя на главните категории, подкатегориите или продуктите. Чрез този бутон ще му бъдат добавени нови полета, които ще трябва да попълни. Структурата на формата за добавяне или редакция на елементи от фиг. 4.12 наподобява тази на страницата с преглед на меню от фиг. 4.13 с цел максимално олесняване на потребителя при неговото ориентиране.

ПРЕДЯСТΙΑ		
Аламинут		
Пържени картофи	250 гр	2.30 лв
Пържени картофи със сирене, подправки и други		

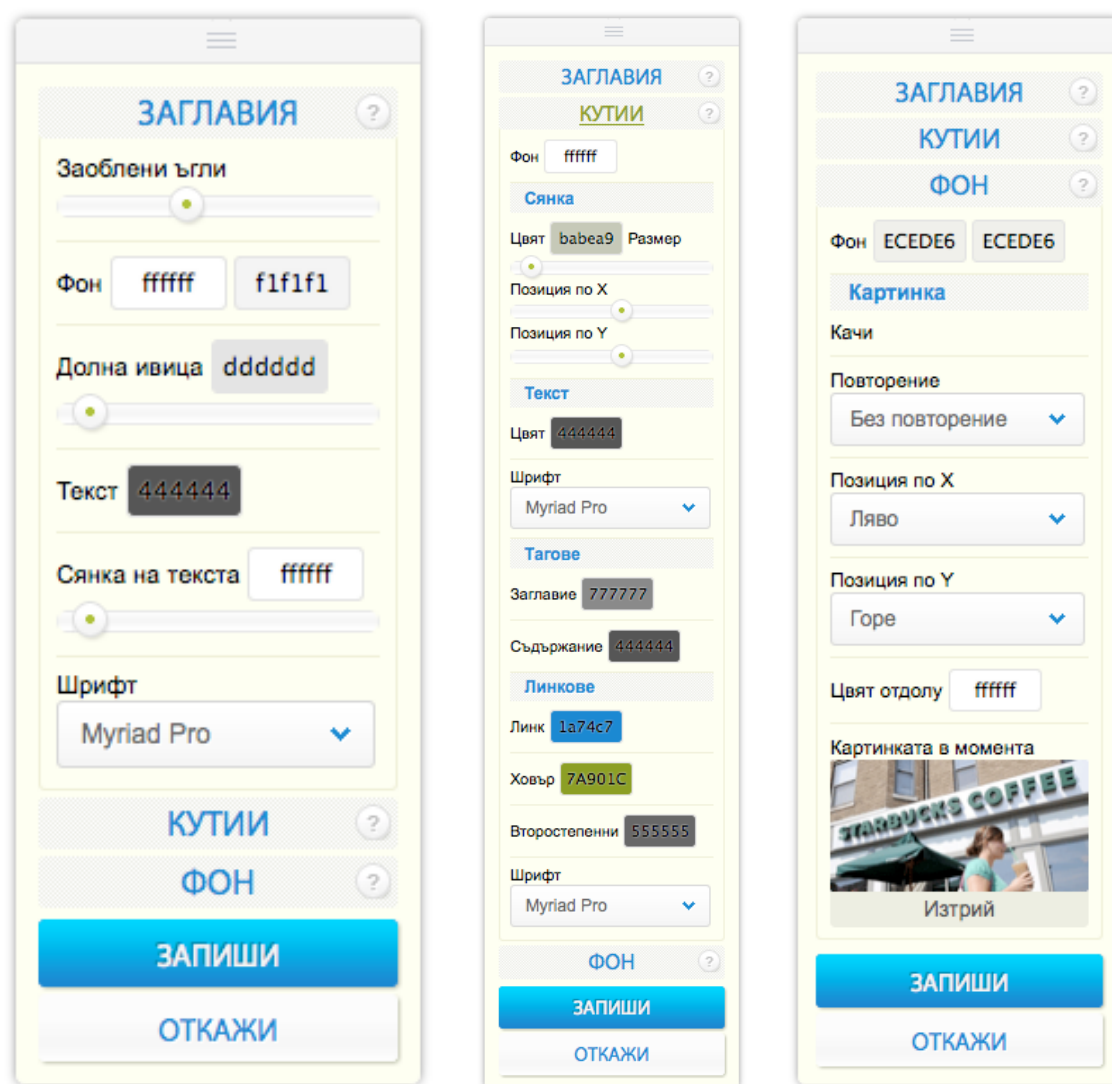
Фиг. 4.12. Редакция на меню.

Пици		
Вегетариански		
Капричоза	600 гр	8.50 лв
<i>пилешко филе кашкавал синьо сирене сметана заеленчуци, пилешко филе кашкавал синьо сирене сметана заеленчуци, пилешко филе кашкавал синьо сирене сметана заеленчуци</i>		
Пеперони	600 гр	8 лв
<i>пилешко филе, кашкавал, синьо сирене, easter egg, сметана, заеленчуци</i>		
Калцоне	600 гр	7.80 лв
<i>пилешко филе кашкавал синьо сирене сметана заеленчуци сметана</i>		

Фиг. 4.13. Преглед на меню.

4.3.11. Бизнес модул Визия

Чрез инструментите от фиг. 4.14. потребителят може да настрои начина по който да изглежда мястото му. Може да промени радиуса на заобляне на кутиите с информация. Има възможност за промяна на цвета на текста, кутиите, сенките и фона на страницата. Може да бъде настройвана и позицията на сенките на кутиите и части от текста. Собственикът може да променя и шрифтовете на отделни елементи от страницата.



Фиг. 4.14. Лента с инструменти за промяна на дизайна.

ЗАКЛЮЧЕНИЕ

Настоящият проект дава добра основа за развитие на реален, комерсиален уеб сайт.

Изчистеният и приятен дизайн, в комбинация с настоящата функционалност поставят добро ниво на потребителско изживяване.

За близко бъдеще е планирано закупуването на домейн имената favplace.bg и far.bg (по-кратко, подходящо за споделяне в микроблогинг платформи с ограничение в символите) и публикуването на доразработена версия на проекта. Благодарение на партньор, базиран в Сан Франциско, Калифорния, проектът има реални шансове да стартира предлагането на платени бизнес функционалности. За целта е планирана регистрацията на търговско дружество в България.

Към Favplace имат интерес и собствениците на най-популярната българска микроблогинг платформа. Планирана е интеграция с тяхната система.

В далечно бъдеще може да бъде добавен модул за превод на интерфейса, като по този начин Favplace ще може да бъде разпространен и в други балкански държави.

Бизнес версиите на местата могат да се доразработят, за да предлагат още интересни и полезни функционалности. Замислена е разработката на модул “Задачи”, чрез който, собствениците, закупили бизнес пакет за мястото си, ще могат да създават промоционални игри, предоставящи различни награди за хората, изпълнили критериите, зададени от собственика.

Модулът “Меню” може да се разшири и да предоставя различни, предварително създадени, темплейти за попълване. Възможността за свалянето им в PDF формат, ще бъде полезна за собствениците, които искат да принтират менюта за обекта си. На същия принцип може да бъде създаден модул за изработване на визитни картички и покани за събития.

Създаването на мобилна версия на уеб сайта е едно от основните неща, които ще бъдат направени. А чрез API, ще могат да бъдат разработени и мобилни приложения за iPhone, Android и Symbian, както и desktop клиенти.

Чрез Facebook Connect и Twitter API потребителите ще могат да се използват системата, без нужда от регистрация.

Работата по настоящата дипломна работа ни помогна да доразвием уменията си за работа в екип, синхронизацията чрез система за контрол на версиите и цялостното ни мислене при разработване на проект от такъв мащаб.

Използвана литература

1. ClientSideNews, Nov/Dec 2010 issue, Page 6 “You Can Really Do That? – The Power of Geolocation Technology” - <http://clientsidenews.com/downloads/CSNV10I6.pdf>
2. Gartner Identifies the Top 10 Consumer Mobile Applications for 2010 - <http://www.gartner.com/it/page.jsp?id=1230413>
3. Статия на eTimes.bg: “3/4 от потребителите у нас ползват мобилен интернет” - http://www.etimes.bg/articles/1009/34_ot_potrebiteelite_u_nas_polzvat_mobilen_internet.html
4. The Web Designer’s Guide to Comparing Photoshop and Fireworks - <http://webdesign.tutsplus.com/articles/general/the-web-designers-guide-to-comparing-photoshop-and-fireworks/>
5. MAMP: Mac, Apache, MySQL, PHP - <http://www.mamp.info/en/mamp/index.html>
6. Apache Subversion - <http://subversion.apache.org/>
7. Foursquare Business - <http://foursquare.com/business>
8. The New York Times “Why Yelp Works” - <http://bits.blogs.nytimes.com/2008/05/12/why-yelp-works>
9. W3Counter – Global Web Stats - <http://www.w3counter.com/globalstats.php>
10. Statcounter.com – Top 12 Browser Versions in Bulgaria from January 2010 to January 2011 - http://gs.statcounter.com/#browser_version-BG-monthly-201001-201101
11. jQuery – Official website – <http://jquery.com>
12. Информация за MIT лиценз - <http://creativecommons.org/licenses/MIT>
13. Cameron Mall – Is it time to move beyond 960? http://cameronmoll.com/archives/2009/04/is_it_time_to_move_beyond_960/
14. Foursquare - <http://foursquare.com>
15. Gowalla - <http://gowalla.com>
16. Facebook Places - <http://facebook.com/places>
17. Yelp – <http://yelp.com>
18. Panic Coda - <http://www.panic.com/coda/>
19. Adobe Photoshop – <http://adobe.com>
20. Dropbox – <http://dropbox.com>
21. Superhosting – <http://superhosting.bg>
22. MySQL Workbench - <http://wb.mysql.com>
23. BBC iPlayer – <http://bbc.co.uk/iplayer>
24. Hulu – <http://hulu.com>
25. Vevo – <http://vevo.com>
26. Gowalla Business - <http://gowalla.com/business>
27. Codeigniter – <http://codeigniter.com>
28. CSS3 - <http://www.css3.info>
29. jQuery – <http://jquery.com>
30. CSS Sticky Footer technique by Ryan Fait– <http://ryanfait.com/sticky-footer>
31. jQuery UI – <http://jqueryui.com>
32. Българизация на jQuery календар - <http://svest.org>
33. Uniform jQuery plugin - <http://pixelmatrixdesign.com/uniform>
34. Търговски регистър - <http://www.brra.bg>
35. MySQL - <http://www.mysql.com>
36. XHTML 1.0 Transitional - <http://www.w3.org/TR/xhtml1>

Съдържание

УВОД	2
ПЪРВА ГЛАВА	4
ПОДОБНИ ПРИЛОЖЕНИЯ С ГЕОТАРГЕТИРАНЕ И ГЕОТАГИНГ. ИЗБОР НА РАЗВОЙНИ СРЕДСТВА И СРЕДИ	4
1.1. Преглед на съществуващи подобни услуги.....	4
1.1.1. Foursquare	4
1.1.2. Gowalla	5
1.1.3. Facebook Places	6
1.1.4. Yelp	7
1.2. Развойни средства и среди.....	9
1.2.1. Panic Coda.....	9
1.2.2. Adobe Photoshop	9
1.2.3. MAMP	10
1.2.4. Subversion	10
1.2.5. Dropbox.....	11
1.2.6. База данни от споделен хостинг	11
1.2.7. MySQL Workbench Community Edition	12
ВТОРА ГЛАВА.....	13
ИЗИСКВАНИЯ КЪМ ГЕОТАРГЕТИНГ УЕБСАЙТА. АЛГОРИТЪМ. СРЕДСТВА ЗА РАЗРАБОТКА. СТРУКТУРА НА БАЗА ДАННИ	13
2.1. Изисквания към социалната мрежа с геотаргетиране	13
2.2. Описание на алгоритъма на уеб приложението	15
2.3. Избор на средства за разработка	16
2.3.1. Apache.....	16
2.3.2. PHP.....	16
2.3.3. MySQL.....	17
2.3.4. Codeigniter	17
2.3.5. XHTML 1.0 Transitional	18
2.3.6. CSS3	19
2.3.7. jQuery JavaScript Library	20
2.4. Структура на базата данни на приложението	20
ТРЕТА ГЛАВА	24
РЕАЛИЗАЦИЯ НА ГРАФИЧНИЯ ИНТЕРФЕЙС НА СОЦИАЛНАТА МРЕЖА. ЦЯЛОСТНА РЕАЛИЗАЦИЯ НА МОДУЛИ РЕГИСТРАЦИЯ И БИЗНЕС.....	24

3.1. Обща стратегия за разработване на уеб сайта.....	24
3.1.1. Основна задача	24
3.1.2. Оптимизация за браузъри	24
3.1.3. Графичен формат на изображенията в дизайна	25
3.1.4. Употреба на CSS sprites	26
3.2. Потребителски интерфейс	28
3.2.1. Графичен дизайн и типография	28
3.2.2. Основна конструкция.....	29
3.2.3. Стилизация на полетата за въвеждане на информация	31
3.2.4. Навигация и потребителски панел	34
3.2.5. Панел за търсене.....	37
3.2.6. Модул Активност	39
3.3. Разработване на модул Бизнес.....	41
3.3.1. Заявка за Бизнес версия	41
3.3.2. Добавяне на елементи в модул Меню	43
3.3.3. Модул за промяна на външния вид на места	46
3.4. Реализация на модул Регистрация	49
ЧЕТВЪРТА ГЛАВА.....	51
РЪКОВОДСТВО НА ПОТРЕБИТЕЛЯ	51
4.1. Изисквания на програмния продукт.....	51
4.2. Инсталация на програмния продукт.....	51
4.3. Използване на социалната мрежа	52
4.3.1. Регистрация.....	52
4.3.2. Вход	53
4.3.3. Редакция на профил	53
4.3.4. Търсене на места	53
4.3.5. Проследяване на потребител.....	55
4.3.6. Използване на стената на началната страница	55
4.3.7. Добавяне на място	56
4.3.8. Добавяне на събитие	58
4.3.9. Идентификация на собственици на места.....	59
4.3.10. Бизнес модул Меню	59
4.3.11. Бизнес модул Визия	60
ЗАКЛЮЧЕНИЕ.....	62
Използвана литература	64