

ПЛОВДИВСКИ УНИВЕРСИТЕТ „ПАИСИЙ ХИЛЕНДАРСКИ“
ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА
КАТЕДРА „КОМПЮТЪРНА ИНФОРМАТИКА“

Тодор Минков Рачовски

ПЛОВДИВСКИ МОБИЛЕН УНИВЕРСИТЕТ

АВТОРЕФЕРАТ

на дисертационен труд за присъждане
на образователна и научна степен „доктор“

област на висшето образование 4. Природни науки, математика и информатика
професионално направление 4.6. Информатика и компютърни науки
докторска програма по Информатика

Научен ръководител:
проф. д.м.н. Георги Тотков

Рецензенти:
проф. д-р Нели Манева
доц. д-р Николай Янев

Пловдив
2018

Дисертационният труд е обсъден и насочен за защита пред научно жури, на заседание на катедра „Компютърна информатика“ при Факултета по математика и информатика на Пловдивския университет „Паисий Хилендарски“ на 16.2.2018 г.

Дисертационният труд „Пловдивски мобилен университет“ съдържа 146 страници. Списъкът на използваната литература съдържа 102 заглавия, от които 9 на кирилица и 93 на латиница. От тях Интернет-источниците са 1 на кирилица и 47 на латиница.

Списъкът на авторските публикации по темата се състои от **6 заглавия**, като резултатите са докладвани на **5 научни конференции**.

Защитата на дисертационния труд ще се състои на 14.5.2016 г. от 12:30 часа в Заседателна зала (Нова сграда на Пловдивския университет, бул. „България“ № 236).

Материалите по защитата са на разположение на интересувашите се в Деканата на Факултета по математика и информатика, Нова сграда на ПУ „Паисий Хилендарски“ (бул. „България“ №236), всеки работен ден от 8.30 до 17 часа.

Автор: *Тодор Минков Рачовски*

Заглавие: *„Пловдивски мобилен университет“*

Университетско издателство „Паисий Хилендарски“

Тираж: *100 бр.*

Пловдив, 2018 г.

Съдържание

УВОД	4
ГЛАВА 1. СЪСТОЯНИЕ НА ИЗСЛЕДВАНИЯТА	6
1.1. Адаптивен и отзивчив УЕБ ДИЗАЙН	6
1.2. Подходи за създаване на мобилен дизайн	7
1.3. Анализ на съществуващи мобилни УЕБ УНИВЕРСИТЕТСКИ СИСТЕМИ	9
1.4. Анализ и оценка на УЕБ САЙТОВЕ ОТ ГЛЕДНА ТОЧКА „МОБИЛНА ВИЗУАЛИЗАЦИЯ“	9
1.5. АВТОМАТИЗИРАНО ГЕНЕРИРАНЕ И МОДЕЛИРАНЕ НА МОБИЛНИ ВЕРСИИ НА УЕБ САЙТОВЕ	10
ГЛАВА 2. МОДЕЛ НА СОФТУЕРНО ПРИЛОЖЕНИЕ ЗА ГЕНЕРИРАНЕ НА МОБИЛНИ ВЕРСИИ НА УЕБ САЙТОВЕ	11
2.1. Основни изисквания и функционалности.....	11
2.2. Подходи за създаване на мобилен дизайн	11
2.3. АРХИТЕКТУРА	13
2.4. МОДЕЛ НА КЕШИРАНЕ НА ГЕНЕРИРАНИ МОБИЛНИ УЕБ СТРАНИЦИ	15
2.5. МОДУЛ ЗА СИГУРНОСТ	16
2.6. ЗАКЛЮЧЕНИЕ	16
ГЛАВА 3. СОФТУЕРЕН ПРОТОТИП MOBILES ONLINE	17
3.1. АРХИТЕКТУРА	17
3.2. ИНТЕРПРЕТАТИВЕН АЛГОРИТЪМ.....	18
3.3. МЕТОДИКА ЗА СЪЗДАВАНЕ НА МОБИЛНИ ПРИЛОЖЕНИЯ С ПРЕПРОЕКТИРАНЕ	20
3.4. МЕТОДИКА ЗА СЪЗДАВАНЕ МОБИЛНИ ПРИЛОЖЕНИЯ ОТ ДИНАМИЧНИ СТРАНИЦИ	21
ГЛАВА 4. ЕКСПЕРИМЕНТИ	21
4.1. АВТОМАТИЗИРАНО ГЕНЕРИРАНЕ НА „ПЛОВДИВСКИ МОБИЛЕН УНИВЕРСИТЕТ“	22
4.1.1. Създаване с препроектиране.....	22
4.1.2. Създаване от динамични страници	23
ЗАКЛЮЧЕНИЕ	25
БИБЛИОГРАФИЯ	28
ПУБЛИКАЦИИ ПО ТЕМАТА НА ДИСЕРТАЦИЯТА	31

Увод

През последните години се наблюдава устойчива тенденция за увеличаване на достъпа до уеб сайтове през мобилни устройства. Поводите за това са разнообразни – преглеждане на информация, пазаруване, обучение и други бизнес дейности. Според независимата компания за уеб анализи StatCounter, през м. октомври 2016 г. за първи път в световен мащаб използването на Интернет през мобилни устройства и таблети надвишава трафика, генериран от десктоп компютри [StatCounter].

Цел и задачи на дисертационния труд

Предмет на изследване на дисертационния труд са възможностите за автоматизирано генериране на мобилни версии на уеб страници, подходящи за разглеждане през различни мобилни устройства, с различна разделителна способност.

Основна цел на изследването *е разработване на технология за автоматизирано изграждане на мобилни уеб страници и апробирането ѝ чрез динамично конструиране на приложение „Пловдивски мобилен университет“.*

За постигане на основната цел са поставени и решени следните задачи:

Задача 1. Анализ на съвременни технологии и средства за изграждане на уеб сайтове;

Задача 2. Създаване на софтуерен модел на приложение за генериране на мобилни версии на уеб страници;

Задача 3. Изграждане на софтуерен прототип за автоматизирано генериране на мобилни уеб страници, базиран на разработения софтуерен модел;

Задача 4. Създаване на методика за изграждането на мобилни уеб страници;

Задача 5. Разработване и апробиране на технология за динамично изграждане на Пловдивски мобилен университет.

Задача 6. Провеждане на експерименти и усъвършенстване на прототипа и софтуерния модел.

Структура на дисертацията

Представеният дисертационен труд се състои от увод, четири глави и заключение, списък на авторските публикации по темата, декларация за оригиналност на резултати и приноси, списък на използваната литература и две приложения.

В първа глава е направен обзор на софтуерни инструменти за автоматизиран анализ и генериране на мобилни версии на уеб сайтове. Разгледани са основните подходи за създаване на мобилен дизайн на уеб стра-

ници. Проучени са приложения за анализ и оценка на уеб сайтове за визуализация от мобилни устройства, а също така и приложения за автоматизирано генериране и моделиране на мобилни версии на уеб сайтове.

Във втора глава е представен софтуерен модел на приложение за генериране на мобилни версии на уеб страници. Последователно са описани поставените функционални и нефункционални изисквания, разработения концептуален модел, модел за кеширане с цел ускорена работа на приложенията и модел на сигурност.

В трета глава е описан създаденият софтуерен прототип. Накратко са представени използваните технологии. Детайлно са разгледани архитектурата и базата данни на приложението, алгоритъма на работа на приложението и основните модули.

Четвърта глава съдържа информация за апробирането на технологията за динамично изграждане на мобилни версии на сайтове. Включва информация за проектиране на дизайн и динамично изграждане на Пловдивския мобилен университет, основните елементи за CSS стилизация, информация за проведените тестове и експерименти с различни бази данни, с и без използване на механизми за кеширане.

В заключението е направена равностметка за извършената работа и са очертани перспективите за бъдещо развитие на разработените модели и софтуерни инструменти.

Списъкът на използваната литература съдържа 102 заглавия, от които 9 на кирилица и 93 на латиница. От тях Интернет-източниците са 1 на кирилица и 47 на латиница.

По темата на дисертацията има **6 публикации**, като резултатите са докладвани на **5 научни конференции**. Една от публикациите е в списание, индексирано в Web of Science и SCOPUS.

Получените по време на изследването резултати са използвани в **проект НИ13-ФИФ009, Многофункционален дигитален университетски архив: моделиране, създаване и споделяне**, ФНИ при ПУ „Паисий Хилендарски“, 2013 – 2014.

Благодарности

Бих желал да благодаря на проф. Георги Тотков за поставянето на амбициозната тема за дисертационното изследване, за насоките и препоръките, и за включването ми в три научно-изследователски проекта, които осигуриха финансова подкрепа за провеждане на дисертационното изследване и публикуване на научни публикации.

Глава 1. Състояние на изследванията

Необходимостта от технология за адаптивно визуализиране на сайтовете, в зависимост от използваното мобилно устройство се поражда още с навлизането на първите по-известни браузъри Netscape (1994 г.) и Internet Explorer (1995 г.) [Wiki-Web-browser]. Те поддържат собствени характеристики за форматиране на определени стандартни HTML-елементи, което в някои случаи води до разлики в представянията на страниците.

Усъвършенстването на стандартите HTML и CSS (съответно. за задаване структурата на уеб страниците и за стилового им представяне) и стремежа за спазването им от съвременните браузъри значително улесни работата на разработчиците на сайтове. Решенията са „фиксиран“, „еластичен“, „течен“ дизайн и др., които понякога предполагат съществуването на минимални размери за определени HTML елементи [Kim'2013], [Mohorovičić'2013].

1.1. Адаптивен и отзивчив уеб дизайн

Терминът *отзивчив уеб дизайн* (Responsive Web Design) е въведен от Ethan Marcotte [Marcotte'2010]. Визира уеб сайтове, които без промяна на изграждащия ги код се визуализират еднакво добре на различни устройства, с различна разделителна способност – мобилни телефони, таблети, персонални компютри, лаптопи и др. Въпреки че понятието отзивчив дизайн придобива популярност относително скоро – след 2012 година, сега – няколко години по-късно, изграждането на нови уеб сайтове, които не притежават подобни възможности, се счита за голям пропуск [Рачовски'2014].

Адаптивен дизайн (Adaptive Web Design) – се отнася за създаване на различни уеб страници, които се визуализират в съответствие със специфичния размер на извикващото ги мобилно устройство. Т.е. въз основа на различна резолюция на екрана ще се визуализират различни уеб страници [Berry'2017].

Общото между двата вида дизайн е, че както отзивчивият, така и адаптивният, са основно за създаване на уеб дизайн, които е оптимизирани за размера на екрана или типа на устройството, което се използва за тяхното преглеждане. Основната концепция е, да се създаде уебсайт, който работи на няколко различни размера на екрани. Най-често е необходимо да работи добре на поне три различни размера на екрана – малък мобилен екран, екран с размер на таблет и по-голям монитор за настолен компютър.

Основните различия между двата вида дизайн [Warner'2013] са следните:

- **Отзивчивите дизайни** отговарят на промените в ширината на прозореца на браузъра чрез плавно регулиране на разположението на елементите в уеб страницата, така че да съответстват най-добре на наличното пространство. По този начин, докато потребителят плъзга страниците на браузъра, за да го направи по-голям или по-малък, ще види промените в дизайна в реално време.
- **Адаптивните дизайни** използват сложни скриптове за автоматично откриване на уеб сървъра, за да идентифицират всяко устройство, което посети сайта и след това да достави най-добрата версия на сайта въз основа на размера и възможностите на всяко устройство. Когато клиентът разглежда такъв сайт, и промени размера на прозореца на браузъра, сайтът няма да се промени. **Предимствата от използването на отзивчив дизайн** [Wiki-RD'2017] са много:
- по-голяма достъпност от различни мобилни устройства и десктоп компютри;
- изработва се дизайн, наподобяващ основния (десктоп дизайн), но за устройства с различни резолюции;
- по-добро индексирание от Google (SEO оптимизация);
- разработва се само един сайт, а не няколко различни, обслужващи различни видове устройства;
- удобно и добре изглеждащо съдържание, и др.

Основните недостатъци от използването на отзивчив дизайн са следните:

- скоростта на зареждане може да е проблем при сайтове с големи размери и такива с много видеоматериали и изображения;
- висок бюджет – за имплементиране на отзивчив дизайн е нужно пълно пренаписване на кода, което обикновено излиза скъпо;
- бавен мобилен интернет или слабо устройство може да доведат до забавяне при зареждането на сайта.

1.2. Подходи за създаване на мобилен дизайн

Дизайнът на огромна част от сайтовете, разработени преди „мобилната ера“ не е адаптиран за употреба чрез мобилни устройства. Причините за това са много и разнообразни – липса на финансови средства, недостатъчна информираност за предимствата от създаването на мобилна версия, нисък приоритет при използването на Интернет като канал за комуника-

ция и маркетинг и др. За да привлекат по-голяма аудитория, собствениците на такива сайтове трябва да вземат решение – дали да изградят сайта си наново, което често изисква много времеви и финансови ресурси, или да създадат отделна мобилна версия на сайта.

Проблемът с визуализацията на уеб сайтове в мобилни устройства може да бъде решен по различни начини:

- осъвременяване на дизайна, с цел преминаване към по-семпъл, отзивчив дизайн;
- изграждане на самостоятелна мобилна версия на уеб сайта;
- използване на софтуерен инструмент, който при необходимост създава мобилни версии на десктоп сайтовете, и ги визуализира по подходящ начин, в зависимост от използваното мобилно устройство;

Осъвременяването на дизайна на уеб сайт, с цел използване на семпъл, отзивчив дизайн е естествен избор за фирми, които разчитат основно на мобилни клиенти. В общия случай подобни решения са евтини за реализация и поддръжка.

Изграждането на нова, самостоятелна мобилна версия на уеб сайт си има множество предимства. Това включва избор на определено съдържание, което да бъде включено в мобилната версия, възможност за прилагане на различни стилове, за различни мобилни устройства (с различна разделителна способност и големина на екраните), премахване на рекламни блокове, изключване на специфични компоненти, визуализацията на които изисква инсталиране на плъгини или друг специализиран софтуер, вкл. с отворен код (Java, JavaScript, и пр.) и др. Основния недостатък на това решение е необходимостта от допълнителна поддръжка и актуализация на мобилната версия. Към настоящия момент има софтуерни инструменти, които подпомагат процеса по изграждане мобилна версия на сайта. Те работят по следния начин: анализират уеб сайт и предлагат мобилен дизайн на сайта, който след това може да бъде модифициран допълнително от дизайнера – визуално или програмно.

Друго възможно решение е използването на специализиран софтуерен инструмент – „обвивка“, който генерира мобилна версия на десктоп сайт и го визуализира по най-подходящия начин за конкретното потребителско мобилно устройство. Изграждането на такова приложение обаче е свързано с множество трудности, основните от които са:

- има голямо разнообразие от уеб сайтове, с различни схеми (layout), вкл. и нестандартни, което затруднява създаването на единен алгоритъм за анализ на съдържанието на уеб сайт и изграждане на мобилна версия;

- много сайтове са изградени автоматично, с помощта на софтуерни инструменти, при което съдържат голямо количество „излишен“ код, който усложнява процеса на парсиране на уеб сайта;
- за постигане на по-добра визуализация на сайтовете, за улеснение при въвеждане и за проверка на въвежданите данни, програмистите често използват софтуерни компоненти с отворен код и JavaScript компоненти, които усложняват автоматизирания анализ и обработка на сайта;
- някои сайтове съдържат голям обем информация, която трудно може да бъде визуализирана по подходящ начин на малък екран, и др.

1.3. Анализ на съществуващи мобилни уеб университетски системи

Предвид съществуването на огромният брой университети, както при нас така и в целият свят са разработели редица уеб мобилни приложения за улесняването на работата от страна на студентите, така и на преподавателите [Рачовски'2011]. Едни от най-известните са:

M-Harvard (<https://m.harvard.edu>) – в него се наблюдава една доста обширна мобилна университетска система. В началната визуализация потребителят има достъп до почти всичко необходимо, както лесен и достъпен начин за навигиране из менютата. Тук може да се акцентира за опция на клиентски настройки на страницата, където потребителят може да избира вида на визуализация спрямо мобилното устройство и шрифта.

Stanford Mobile (<https://uit.stanford.edu/mobile>) – тук е наблегнато на изграждане на страница с адаптивен дизайн, но за сметка на това има изградено и мобилно приложение. Има поддръжка за операционни системи **Android** и **iOS**.

Като извод от горе посочените може да се твърди за добра практика е изграждането на отделни мобилни университетски системи, както и мобилни приложения за достъп от потребителите [Рачовски'2012].

1.4. Анализ и оценка на уеб сайтове от гледна точка „мобилна визуализация“

В Интернет има множество инструменти за анализ и оценка на това, доколко една уеб страница е удобна за мобилни устройства.

Един от най-известните и безплатни инструменти за анализ на уеб страници – **Mobile-Friendly Test Tool**, е разработен от Google. Той проверява дали дадена уеб страница е с отзивчив дизайн, маркира слабостите на дизайна и посочва ресурси с информация за оптимизирането му [Google-TestTool]. Основните критерии, които се тестват са:

- **използване на Flash.** Повечето мобилни браузъри не поддържат Flash. Съдържанието, базирано на Flash не се визуализира, затова не се препоръчва използването на такива компоненти;
- **конфигуриране на изглед (Viewport).** Всяка страница трябва има конфигуриран изглед, чрез HTML мета тага viewport.
- **фиксирана ширина на изгледа.** Някои страници са с фиксирана ширина, което е в разрез с изискванията за отзивчив дизайн;
- **оразмеряване на съдържанието според изгледа.** Несъобразяването с това изискване води до необходимост от хоризонтално скролиране, за да се визуализират всички думи или картинки от уеб страницата;
- **малък размер на шрифта,** който затруднява четенето;
- **разположение на контролите за управление и навигация.** Близкото разположение на контролите за навигация (бутони, менюта, линкове и др.) може да затрудни докосването им с пръст от потребителя или потребителят да докосне няколко елемента, разположени близо един до друг;

Google са разработили API за тестване на отзивчивост на дизайна, наречено **Mobile-friendly test API** [Google-API]. То може да бъде използвано за разработване на алтернативни инструменти за валидиране и анализ на уеб сайтове. Успешен пример в тази насока е **Mobile-Friendly Bulk Testing Tool** [Bulk-Testing-Tool].

WompMobile [WompMobile] е друг инструмент, предоставящ опции за анализ на мобилна страница, както и изграждане на мобилна страница от вече съществуваща.

1.5. Автоматизирано генериране и моделиране на мобилни версии на уеб сайтове

Има разработени редица **софтуерни инструменти, които напълно или частично автоматизират процеса по създаване на мобилна версия на даден сайт.** Такива приложения са Duda [Duda], bMobilized [bMobilized], Snapmobi [Snapmobi], Mobify [Mobify], goMobi [goMobi] и др. Те изграждат автоматизирана мобилна версия на обикновена уеб страница на няколко стъпки: разпознаване на менюта; автоматичен избор на подходящ мобилен изглед; създаване на мобилни стилове; добавяне на плъгини. На следващ етап потребителят може самостоятелно, използвайки визуални инструменти, да укаже множество допълнителни настройки: да посочи блоковете с менюта и основното съдържание; да избере друг мобилен изглед; да зададе стилове; да добави различни бутони в сайта, реализирани като плъгини – телефон, e-mail, карта за местонахождение, контакти, връзки към социални мрежи и споделяне и мн. др.

Глава 2. Модел на софтуерно приложение за генериране на мобилни версии на уеб сайтове

В настоящата глава е представен концептуален софтуерен модел за автоматизирано генериране на мобилни версии на уеб страници. Описани са основните изисквания към приложението, неговия модел, модел за кеширане и модел за сигурност.

Представеният модел описва приложение, което работи като „обвивка“ на десктоп сайтовете, като генерира мобилни версии, подходящи за визуализация в конкретното потребителско мобилно устройство.

2.1. Основни изисквания и функционалности

В контекста на непрекъснато променящите се изисквания и стандарти за уеб, наложени както от потребителите, така и от търсещите машини, са поставени следните *основни изисквания към създаваната технология*:

- *да поддържа отзивчив дизайн;*
- *да позволява добавяне на стилове за нови мобилни устройства, вкл. с нестандартни размери и резолюции;*
- *да е с интуитивен потребителски интерфейс;*
- *да съхранява изградените мобилни версии на уеб страниците*
- *да събира информация за грешки в процеса на конвертиране;*
- *да е отворено за бъдещи разширения и промени.*

Основни типове потребители на приложението са: администратор, регистриран потребител и нерегистриран потребител. **Нерегистрираният потребител** има права да въвежда URL адреси на уеб страници и да преглежда генерираните за мобилно устройство визуализации. В допълнение на това, **регистрираният потребител** може да преглежда генерираните от него мобилни страници.

2.2. Подходи за създаване на мобилен дизайн

Дизайнът на огромна част от сайтовете, разработени преди „мобилната ера“ не е адаптиран за употреба чрез мобилни устройства. Причините за това са много и разнообразни – липса на финансови средства, недостатъчна информираност за предимствата от създаването на мобилна версия, нисък приоритет при използването на Интернет като канал за комуникация и маркетинг и др. За да привлекат по-голяма аудитория, собствениците на такива сайтове трябва да вземат решение – дали да изградят сайта си наново, което често изисква много времеви и финансови ресурси, или да създадат отделна мобилна версия на сайта.

Проблемът с визуализацията на уеб сайтове в мобилни устройства може да бъде решен по различни начини:

- осъвременяване на дизайна, с цел преминаване към по-семпъл, отзивчив дизайн;
- изграждане на самостоятелна мобилна версия на уеб сайта;
- използване на софтуерен инструмент, който при необходимост създава мобилни версии на десктоп сайтовете, и ги визуализира по подходящ начин, в зависимост от използваното мобилно устройство;

Осъвременяването на дизайна на уеб сайт, с цел използване на семпъл, отзивчив дизайн е естествен избор за фирми, които разчитат основно на мобилни клиенти. В общия случай подобни решения са евтини за реализация и поддръжка.

Изграждането на нова, самостоятелна мобилна версия на уеб сайт си има множество предимства. Това включва избор на определено съдържание, което да бъде включено в мобилната версия, възможност за прилагане на различни стилове, за различни мобилни устройства (с различна разделителна способност и големина на екраните), премахване на рекламни блокове, изключване на специфични компоненти, визуализацията на които изисква инсталиране на плъгини или друг специализиран софтуер, вкл. с отворен код (Java, JavaScript и пр.) и др. Основния недостатък на това решение е необходимостта от допълнителна поддръжка и актуализация на мобилната версия. Към настоящия момент има **софтуерни инструменти, които подпомагат процеса по изграждане мобилна версия на сайта.** Те работят по следния начин: анализират уеб сайт и предлагат мобилен дизайн на сайта, който след това може да бъде модифициран допълнително от дизайнера – визуално или програмно.

Друго възможно решение е използването на **специализиран софтуерен инструмент – „обвивка“, който генерира мобилна версия на десктоп сайт и го визуализира по най-подходящия начин за конкретното потребителско мобилно устройство.** Изграждането на такова приложение обаче е свързано с множество трудности, основните от които са:

- има голямо разнообразие от уеб сайтове, с различни схеми (layout), вкл. и нестандартни, което затруднява създаването на единен алгоритъм за анализ на съдържанието на уеб сайт и изграждане на мобилна версия;
- много сайтове са изградени автоматично, с помощта на софтуерни инструменти, при което съдържат голямо количество „излишен“ код, който усложнява процеса на парсиране на уеб сайта;
- за постигане на по-добра визуализация на сайтовете, за улеснение при въвеждане и за проверка на въвежданите данни, програмистите

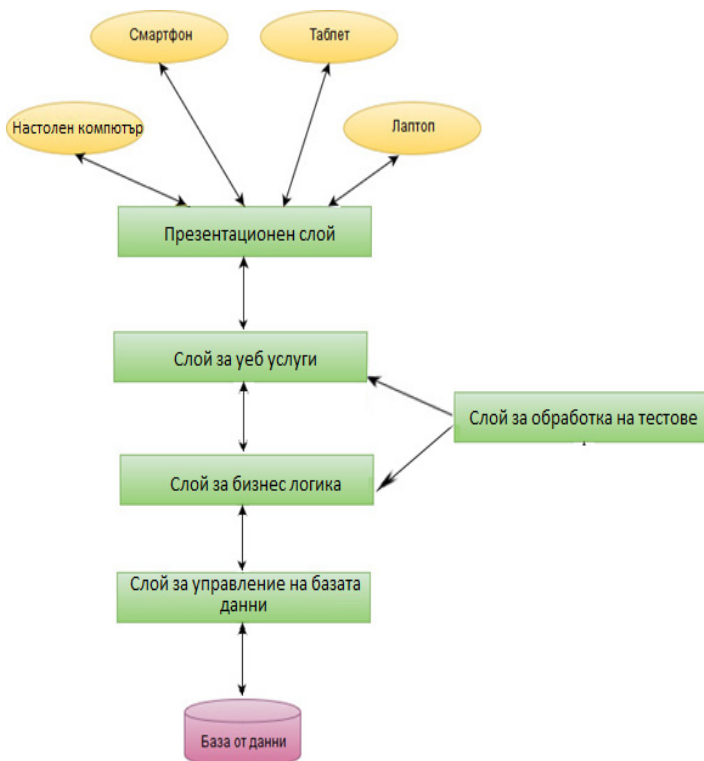
често използват софтуерни компоненти с отворен код и JavaScript компоненти, които усложняват автоматизирания анализ и обработка на сайта;

- някои сайтове съдържат голям обем информация, която трудно може да бъде визуализирана по подходящ начин на малък екран, и др.

2.3. Архитектура

Предлаганата архитектура е многослойна и включва: База данни, слоеве за управление на базата данни, бизнес логика, уеб услуги и Презентационен слой (Фигура 1) [Rachovski'2017a].

В **Базата данни** се съхраняват всички данни за потребителите на приложението, генерираните мобилни страници, журналът за грешки и някои статистически данни. За да се гарантира по-голяма гъвкавост и скалируемост, и възможност за запис на данни без предварително дефинирана схема, е подходящо използването на нерелационни бази данни (NoSQL).



Фигура 1. Модел на приложението

Слоят за управление на базата данни е базиран на концепцията на ORM технологията (Object-relational mapping) за конвертиране на данни чрез използване на обектно-ориентирани езици за програмиране. Той съдържа информация за осъществяване на достъп на приложението до базата данни, представяния на информация от базата данни във вид на класове и обекти и др.

Слоят за бизнес логика съдържа компонентите, които извършват същинската работа по изграждане на мобилната страница. Основен компонент в него е HTML парсерът, който извършва анализ на оригиналната уеб страница, създава дървовиден обектен модел, оптимизира обектите на уеб страницата и изгражда междинна, нестилизирана версия на уеб страницата.

Слоят за уеб услуги получава от слоя за бизнес логика обработената информация, преобразува я в JSON формат и я предава за последваща обработка на презентационния слой.

Презентационният слой е отговорен за избора и използването на подходящ CSS стил за изгражданата мобилна страница, в зависимост от използваното мобилно устройство.

Слоят за обработка на тестове в него се намират всички тестове свързани с валидацията на функциите от слоя за уеб услуги и слоя за бизнес логика.

Предимство на предложената архитектура е че се отличава с високо ниво на абстрактност, като при това е отворена, лесно разширяема и гъвкава.

Създаденият алгоритъм за конвертиране на обикновена уеб страница в мобилна включва следните основни стъпки:

Стъпка 1. Прочитане на уеб страницата – потребителят въвежда URL адрес на уеб страница, приложението валидира адреса и извлича уеб страницата;

Стъпка 2.Парсиране на уеб страницата, при което с помощта на HTML парсер се извършва анализ и обработка на информацията и се изгражда йерархичен обектен модел на елементите на уеб страницата;

Стъпка 3. Генериране на модел, подходящ за стилизиране. На този етап всички елементи на йерархичния обектен модел се подлагат на допълнителна обработка. Обектите се разпознават и записват в таблица на съответствията, съдържаща информация за основните структурни компоненти (заглавие на страницата, език, мета данни, линкове, основно съдържание и др.);

Стъпка 4. Визуализация. Върху уеб страницата се прилага стил, съобразен с параметрите на извикващото я мобилно устройство и тя се визуализира в браузъра на клиента.

2.4. Модел на кеширане на генерирани мобилни уеб страници

Интернет потребителите са изключително чувствителни към темата за скорост на зареждане и визуализиране на уеб страниците в браузъра им. За ускоряване на тези процеси е удачно да се извършва кеширане на уеб съдържанието. Редица доставчици на ИТ технологии предлагат различни решения за кеширане. Основната цел е да се редуцира броят на заявките към уеб сървъра, и тяхната обработка. Т.е. при заявка за конкретна уеб страница от потребител, ако нейното съдържание не е променяно, и тя е кеширана, на потребителя се предоставя нейното кеширано копие, и не се изпълнява повторна заявка за уеб страницата към страната на сървъра.

В модела могат да се дефинират **2 (два) типа кеширане на данни:** кеширане на сесийни данни и кеширане на вече съществуващи генерирани страници.

Кеширанто на сесийни данни включва 3 (три) основни компонента.

Компонент 1. Браузър кеширане. Основна роля при това кеширане играе уеб браузърът, като неговата функция е да запамята всякаква информация за уеб страницата, вкл. изображения, текст и др. Това се извършва при първоначалното зареждане на уеб страницата. При повторно зареждане на същата страница, тя се зарежда от кеша на браузъра. По този начин самото предоставяне на информацията е много по-бързо, защото браузърът не изпраща заявки и не изисква ресурси от страната на сървъра.

Компонент 2. Технология за кеширане на ниво уеб сървър. При това кеширане основна роля играе така нареченото Reverse Proxy (обратно прокси). То е от тип прокси сървър (посредник между уеб клиент и уеб сървър) и една от основните функции, които може да изпълнява, е кеширането. Прокси сървърът обикновено се поставя пред уеб сървъра, като по този начин скрива уеб сървъра от потребителя. При това, потребителят се свързва с прокси сървъра, а не директно с уеб сървъра. Предимствата от използването му са следните:

- намаляване на натоварването на уеб сървъра чрез извършването от прокси сървъра кеширане на информацията;
- защита на уеб сървъра;
- намаляване времето за достъп;

- лансиране на натоварването (ако има повече от един уеб сървър след проксито), и др.

Компонент 3. Технология за ниво приложение. Тук кеширането се осъществява на сървъра, на който се намира и самото приложение. Това кеширане обикновено се комбинира с някои от другите видове кеширане. Съществуват различни системи и технологии за кеширане на информация на ниво приложение. Сред най-използваните са: Memcache, Redis, APC, eAccelerator и др.

Кеширането на вече съществуващи генерирани страници включва локално съхранение на данните на всяка генерирана мобилна страница, и извикването им преди повторна обработка.

В контекста на спецификата на разработвания модел, е необходимо да се извършва специфично кеширане на данни. Данните, обект на кеширане са HTML низове (стрингове).

Проверката за това, дали данните са вече обработени, се прави на две стъпки.

Стъпка 1.Проверява се дали URL адреса, който е зададен в полето, е записан в базата данни;

Стъпка 2. Проверява се дали взетия HTML стринг е идентичен с този в базата данни. За проверка на съдържанието се използва хеш функцията MD5.Тази функция се използва за проверка цялостта на данните. 128-битовите MD5 хешове обикновено се представят като низ от 32 шестнадесетични цифри. Сравняват се изтеглената информация и информацията в низа. При наличие на дори и малка промяна в HTML съдържанието на уеб страницата, то генерираните низове няма да съвпадат.

2.5. Модул за сигурност

Една от най-нашумелите теми в общественото пространство е Кибер сигурността. Редица са проблемите, породени от Кибер атаки, източването на поверителна информация, както и поверителността на личните данни. В повечето случаи свободното използване на едно приложение без реализирани защити за сигурността на потребителските данни, води до негативни последици. Като основни проблеми в уеб сигурността може да се посочат: Cross Site Scripting (XSS), SQL Injection, General Script Injection [Huang'2004].

2.6. Заключение

Разработения модел на приложение за разглеждане на десктоп сайтове през мобилни устройства е с многослойна архитектура. Той включва база данни, слоеве за управление на базата данни, бизнес логика, уеб услуги и

презентационен слой. В базата данни се съхраняват всички данни, необходими за работата на приложението. Слой за управление на базата данни използва ORM модел за конвертиране на релационните данни в обектно-ориентиран формат и управлява достъпа до тях. Слой за бизнес логика съдържа компоненти, които извършват същинската работа по изграждане на мобилна версия на страницата, вкл. парсиране, изграждане на йерархичен модел на уеб страницата, и редица преобразувания. Слой за уеб услуги осигурява на външни приложения достъп до функционалности на по-ниско ниво, а презентационният слой е отговорен за избора и използването на подходящ CSS .

Конвертирането на десктоп уеб страница в мобилна се извършва по следните стъпки:

Стъпка 1 – Потребителят въвежда URL адрес на уеб страница, след което приложението валидира адреса и извлича съдържанието ѝ;

Стъпка 2 – HTML парсер извършва анализ и обработка на информацията и изгражда йерархичен обектен модел на елементите на уеб страницата;

Стъпка 3 – Генерира се модел, подходящ за стилизиране. Всички елементи на йерархичния обектен модел се подлагат на допълнителна обработка. Генерира се представяне на уеб страницата в JSON формат, подходящо за визуализиране чрез различни стилове;

Стъпка 4 – Върху уеб страницата се прилага стил, съобразен с параметрите на извикващото я мобилно устройство и тя се визуализира в браузъра на потребителя.

Глава 3. Софтуерен прототип Mobiles Online

3.1. Архитектура

Архитектурата на създадения прототип Mobiles Online е многослойна и е изградена от вече описаната в модела на софтуерното приложение. Тя включва: база данни, слоеве за управление на базата данни, бизнес логика, уеб услуги и презентационен слой.

В **базата данни** се съдържат таблиците описващи вече генерираните мобилни страници, история на грешки при генерирането им и статистически данни за най-често посещаваните адреси на страници. За да се гарантира по-голяма гъвкавост и скалируемост, се дава възможност за запис на данни без предварително дефинирана схема, като е подходящо използването на нерелационни бази данни. Това гарантира стабилност на приложението и при бързо нарастване на обема на съхраняваните данни.

Слоят за **управление на базата данни** служи за комуникацията между базата данни и приложението, като основната му функционалност е да прави и интерпретира модела на базата данни. Създава се виртуален модел, който може лесно да бъде манипулирана чрез средствата на конкретен език за програмиране. Предвид използването на ORM технологията са реализирани класове с имената на таблиците и колоните:

- Content – тук е връзката между таблиците MobileContent и RealContent;
- MobileContent – съхранява се вече преработената информация;
- RealContent – съхранява се първоначалната информация;
- RequestError – грешките, които са възникнали (пример: грешен отговор при направяна заявка към сайта);
- Role – роли на потребителите;
- Users – данните на потребителите;
- Users_Roles – връзката между отделните потребители и техните роли.

Като функционалност на слоя за **бизнес логика** или още Core слой е извличането на информацията за уеб страницата и преобразуването ѝ от обикновена в мобилна. За текущото реализиране на системата е създаден HTML анализатор, който служи за обработката на целия документ и след обработка, подава на различни функции част от този HTML, които го преработват и преобразуват по съответен начин [Рачовски'2017b].

Стъпките, през които се преминава при цялостната обработка на документа са следните:

- Изграждане на анализатор на HTML документи;
- Извличане на информация от мета тагове;
- Извличане на навигационното меню;
- Извличане на footer частта;
- Извличане на title частта;
- Извличане на основната информация за уеб страницата – body частта.

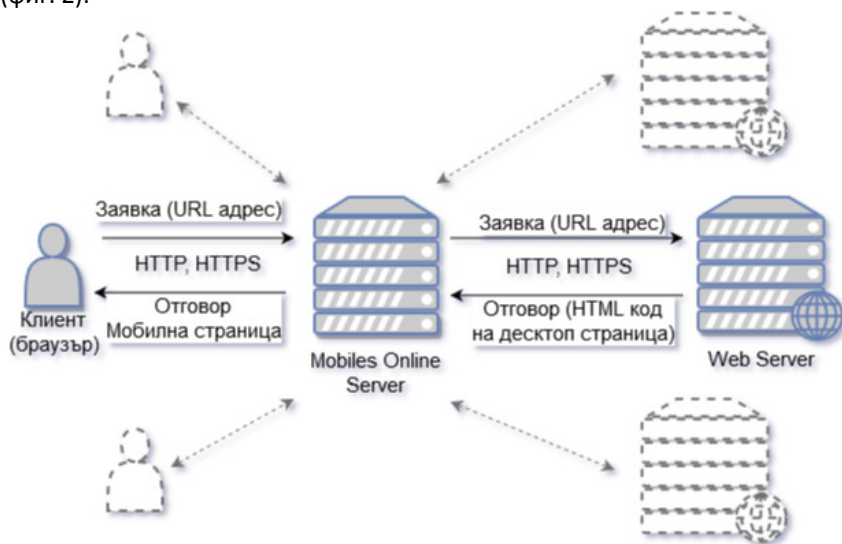
В **презентационният слой** се съдържа визуализацията на създадения прототип, като служи за връзката между външният свят и различните шаблони за онагледяване на вече генерираните страници. В него също е реализирана частта с уеб услуги - Web API Controller, които всъщност управляват и правят връзката между генерирана визията (View в MVC), както и позволява външни приложения за достъп до системата.

3.2. Интерпретативен алгоритъм

Mobiles Online има за цел улесни потребителите на мобилни устройства, като генерира мобилни версии на уеб страниците, в зависимост от

параметрите на конкретното използвано мобилно устройство. То служи като посредник между клиентския агент (desktop или мобилен уеб браузър) и реалния сървър, на който се хоства целевата уеб страница. Потребителят задава заявка за уеб страница, като изписва адреса ѝ в Mobiles Online. Заявката се препредава към целевия уеб сървър, който връща оригиналната HTML страница. Mobiles Online обработва получените данни и генерира мобилна версия на страницата, която предоставя на клиента. Mobiles Online работи като прокси сървър, който със знанието на клиента променя начина на визуализация на желаното от него съдържание.

6 (шест) са основните стъпки в процеса на работа на приложението (фиг. 2).



Фигура 2. *Процес на работа на Mobiles Online*

Стъпка 1. Потребителят стартира в браузъра уеб приложението Mobiles Online;

Стъпка 2. Потребителят въвежда URL адрес на уеб страница в поле „Търсене“;

Стъпка 3. Указаният адрес се валидира и ако е невалиден, на клиента се връща съобщение за грешка;

Стъпка 4. Извлича се съдържанието на изисквания ресурс. Ако съдържанието е невалиден HTML документ, на клиента се връща съобщение за грешка;

Стъпка 5. HTML парсер извършва анализ и обработка на информацията и изгражда йерархичен обектен модел на елементите на уеб страницата;

Стъпка 6. Генерира се модел на документа, подходящ за мобилно стилизация, при което:

- всички елементи на йерархичния обектен модел се подлагат на специфична обработка – премахват се стилове и елементи, заменят се елементи, добавят се нови „мобилни“ елементи и др.;
- генерира се представяне на уеб страницата в JSON формат, подходящо за визуализиране чрез различни стилове;
- хипервръзките се пренаписват така, че изпълнението им да се извършва отново от приложението Mobiles Online.

Стъпка 7 – Върху създаденият модел на уеб страницата се прилага стил, съобразен с параметрите на извикващото я мобилно устройство;

Стъпка 8. Към брауъра на потребителя се връща генерираната мобилна версия.

3.3. Методика за създаване на мобилни приложения с препроектиране

Както вече става ясно от модела на приложението представено в глава 2 и 3, както и спазвайки архитектурата от (фиг. 1) по много лесен начин даден потребител, използвайки алгоритъма на работа, може да създаде собствена страница за текущо съществуващо приложение. Няколко са основните неща които трябва да бъдат съобразени: изграждане на адаптивен дизайн, конфигуриране работата му с приложение за автоматично генериране на мобилни страници, изграждане на адаптивен дизайн от вече съществуващата информация. Методиката ще се представи по Стъпки, които да се следват в Глава 4. при създаване на Пловдивския мобилен университет за изграждането на мобилен дизайн.

Стъпка 1. Изграждане на адаптивен дизайн – при проектирането на дизайна могат да се използват редица стандарти за визуализация на мобилни страници. Потребителят може да проектира собствен изглед по негов вкус и в зависимост от приложението;

Стъпка 2. Конфигуриране работата на дизайна с приложение за автоматично генериране на мобилни страници – тук необходимото нещо на страната на потребителя е механизъм за достъпване до системата. Това се извършва чрез WEB API. Предвид това, дали се генерира клиентска (административна) част;

Стъпка 3. Изграждане на адаптивен дизайн от вече съществуващата информация – тук може да се акцентира над различна визуализация, съобразена спрямо изискванията на потребителя.

3.4. Методика за създаване мобилни приложения от динамични страници

Както в предишната глава, така и тук основната цел е създаването на методика при динамични мобилни страници. Използват се същите 3 (три) основни стъпки както при статични уеб страници.

Стъпка 1. Изграждане на адаптивен дизайн – съществена разлика се получава при изграждането на адаптивния дизайн. Препоръчително е използването на всички добри практики при изграждане на страници от;

Стъпка 2 . Конфигуриране работата на дизайна с приложение за автоматично генериране на мобилни страници – при използването на динамични мобилни приложения, е необходимо всеки път извличането и генерирането на нововъведената страница;

Стъпка 3. Изграждане на адаптивен дизайн от вече съществуващата информация – както вече дефинираната в методика, така и тук може да се акцентира над различна визуализация, съобразена спрямо изискванията на потребителя.

Глава 4. Експерименти

Направени са два основни теста за кеширане на информация и обработка на информация от базата данни и са представени два експеримента за автоматизирано създаване на мобилни приложения („Пловдивски мобилен университет“) с използване на софтуерния прототип и методика за създаване на статични мобилни приложения от Глава 3.10 и втори експеримент и динамична мобилна уеб страница с няколко дизайна следвайки методиката за създаване на динамични мобилни приложения от раздел 3.11. В тази част от работата са представени експерименти, свързани със създаването на мобилен интерфейс на дадена страница от страна по потребителя [Рачовски’2017с].

Предпоставки за провеждането на подобен тип експеримент са много. Като основен аспект може да бъде посочена необходимостта за разглеждане на вече съществуващи страници, които не предоставят удобството на потребителя за лесен и бърз достъп до информация. С използването на системата много лесно може да се генерира нова мобилна версия за дадена интернет страница. Използвайки текущото приложение на системата, може да се намалят разходите относно оптимизацията на дадена страница.

4.1. Автоматизирано генериране на „Пловдивски мобилен университет“

В текущият раздел ще се спрем над създаването прототипи за Пловдивски Мобилен Университет по методиките описани в Глава 3. За реализацията им са разделени на: създаване с препроектиране и създаване от динамични страници.

4.1.1. Създаване с препроектиране

За текущия експеримент (създаване и генериране на мобилна страница) е използвана методиката за създаване на мобилни приложения с препроектиране, описана в раздел 3.10 за страницата на Пловдивския университет „Паисий Хилендарски“ (<http://uni-plovdiv.bg>).

Стъпка 1. Изграждане на адаптивен дизайн

При изграждането на дизайна е много важно да бъдат взети предвид съвременните изисквания за мобилен дизайн. Има редица готови софтуерни рамки, които са удобни за употреба и предлагат готови реализирани системи, с лесни за използване функционалности за оформление на страниците. Едни от най-масово използваните са Twitter Bootstrap, jQuery mobile и др.

Създаването на адаптивен дизайн изисква и разпознаване и зачитане разделителната способност на екрана на мобилното устройство. Като главен аспект тук може да се погледне дефинирането на няколко „медийни екрани“ или „медийни заявки“, съобразени с тези изисквания. За текущият експеримент са дефинирани следните устройства с максимална ширина на екрана 1920 пиксела, 1600 пиксела, 1440 пиксела, 1280 пиксела, 1050 пиксела, 1024 пиксела, 991 пиксела, 768 пиксела.

С дефинирането на тези екрани се обхващат почти всички масово използвани мобилни устройства. Практиката показва, че почти винаги се прави комбинация от медийни екрани и оформления, за използване на CSS медийни заявки. В доста от случаите потребителите превключват от персонален компютър към друго устройство, така че уеб сайтът трябва автоматично да променя разделителната способност, като избира най-подходящата за използваното устройство.

Стъпка 2. Конфигуриране на собствено приложение за работа с модел за генериране на мобилни страници

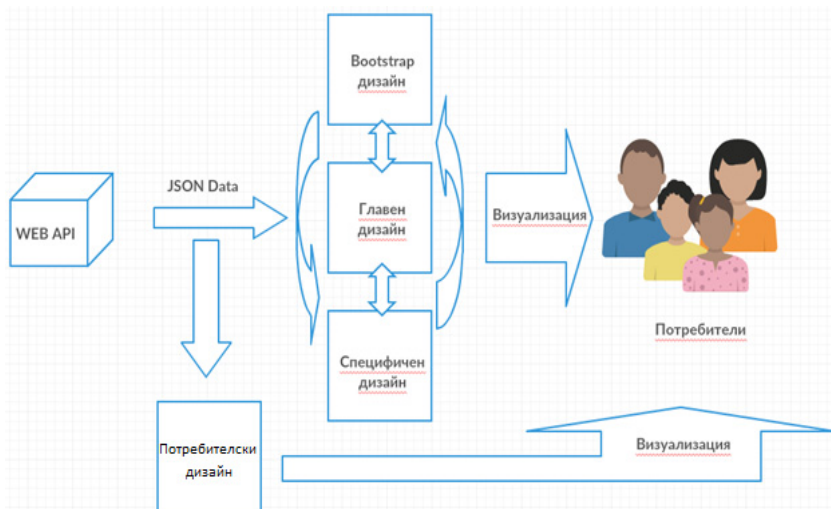
На тази стъпка се използва уеб езика за програмиране JavaScript. Той е скриптов език, с помощта на който се обработват данните от страната на клиента, т.е. в уеб браузъра на потребителя. В текущият модел, адреса на страни-

цата за обработка трябва да бъде статичен, който се подава като параметър при обработката. Прави се заявка към приложението и след връщането на вече парсираната информация се стилизира за текущата страница.

Стъпка 3. Изграждане на адаптивен дизайн от вече съществуващата информация – основна роля тук играе стилизирането на информацията подходяща за различните мобилни устройства, спрямо разделителните способности. След върнатата парсирана информация подобно на другите случаи, се записва в определени елементи. Необходима е допълнителна преработка на информацията, за по-добра визуализация.

4.1.2. Създаване от динамични страници

При проектирането на дизайна на приложението са използвани редица стандарти за визуализация на мобилни страници. Както при направеният експеримент за страницата на Пловдивския университет, така и тук се спазва строго **методиката за създаване на мобилни приложения от динамични страници** (фиг. 3).



Фигура 1. Обща схема за изграждане на шаблон на страница

Стъпка 1. Изграждане на адаптивен дизайн – В този случай за улеснение на потребителите са създадени три основни дизайна (шаблона):

- **Основен дизайн** – проектиран е въз основа на съвременните стандарти за изграждане на мобилни интернет страници;

- *Bootstrap дизайн* – проектиран въз основа на bootstrap технологията;
- *Специфичен дизайн* – съобразен със спецификата на използваните стари стандарти за уеб (по-старите страници).

И трите дизайна са насочени към изграждане на мобилна версия на приложението.

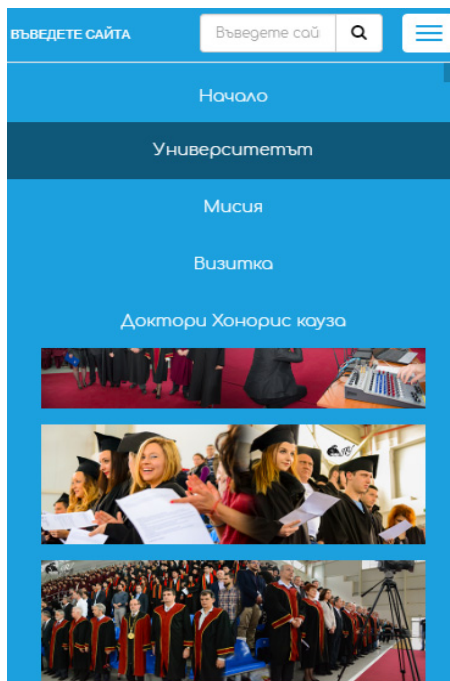
Данните, необходими за визуализацията, се вземат от интернет услуга (service), която се генерира в Web Core частта на приложението, която подава и обработва данните в JSON формат на самите страници, т.е. страницата извиква услугата, а услугата предоставя нужните данни.

Стъпка 2. Конфигуриране работата на дизайна с приложение за автоматично генериране на мобилни страници – след въвеждане на URL в текстовото поле и натискане на бутона, се визуализират основните DOM елементи. При първоначалното зареждане на уеб страницата, по подразбиране се използва основния дизайн. При следващите зареждания, се извършва проверка кой е последно използвания от потребителя стил, след което се изпълнява JavaScript функция за попълване на обработената информация за страницата.

След вземането на цялата информация от интернет услугата, отделните елементи се попълват в съответните секции, които са разделени на няколко елемента:

- *Заглавие на страницата;*
- *Меню за навигация;*
- *Тяло на документа;*
- *Списъчни блокове;*
- *Долна част на страницата.*

Стъпка 3. Изграждане на адаптивен дизайн от вече съществуващата информация – за сметка на предишният експеримент тук трябва да се съобрази визуализацията на различните уеб страници. Предвид различното естество на съдържание, отново са разделени на три основни шаблона.



Фигура 2. Изглед на Пловдивския мобилен университет

Заклучение

Адаптирането на създадените преди „мобилната ера“ сайтове към съвременните изисквания за отзивчив дизайн е едно от основните предизвикателства пред разработчиците на уеб сайтове. Най-общо задачата може да се реши по два начина – пълно пренаписване на уеб сайта с цел изграждане на отзивчив дизайн или създаване на нова, мобилна версия на сайта. Универсално решение за потребителите на такива сайтове е използването на приложения – обвивки, които по зададен адрес на уеб страница автоматизирано генерират мобилна версия на уеб страницата, подходяща за разглеждане през конкретното използвано устройство.

В дисертационното изследване е представен многослоен концептуален модел на приложение за автоматизирано генериране на мобилни страници. Той парсира дадена уеб страница, създава йерархичен обектен модел на страницата и я представя в JSON формат. JSON файловете са достъпни чрез уеб услуги и могат да бъдат използвани за форматиране от трети страни, чрез прилагане на CSS стилове. Той използва ORM технология, за да осигури гъвкаво представяне на данните от базата данни в обектен формат и лесното им манипулиране. Представеният модел е апробиран чрез изграждане на съответен прототип.

В рамките на дисертацията са поставени и **решени следните основни задачи:**

Задача 1. Проучване и анализ на съвременни технологии и средства за изграждане на уеб сайтове.

Задача 2. Създаване на софтуерен модел на приложение за генериране на мобилни версии на уеб страници.

Задача 3. Проектиране и изграждане на софтуерен прототип за автоматизирано генериране на мобилни уеб страници, базиран на разработения софтуерен модел.

Задача 4. Създаване на методика за изграждането на мобилни уеб страници;

Задача 5. Разработване и апробиране на технология за динамично изграждане на Пловдивски мобилен университет.

Задача 6. Провеждане на експерименти и усъвършенстване на прототипа и софтуерния модел.

Въз основа на получените при изследването резултати, основните приноси на дисертационния труд могат да се характеризират като научно-приложни и приложни, както следва:

Научен принос

1. Създаден е концептуален модел на софтуерно приложение за генериране на мобилни версии на уеб страници; модел на кеширане на генерирани мобилни уеб страници, и модел за сигурност на приложението.

Научно-приложен принос

1. Проектирана и имплементирана е софтуерна система, базирана на разработения концептуален модел;
2. Създаване на методика за изграждането на мобилни уеб страници.

Приложни приноси:

1. Извършена е практическа верификация на предложените модели чрез създаването на софтуерен прототип (достъпен на адрес <http://mobile.studentschool.eu/>);
2. Проведени са успешни експерименти с прототипа върху реални данни в реална среда, вкл. работа на прототипа с различни бази данни и модели на кеширане на данните. Изследвана е и доказана приложимостта на създадения прототип, а от там и на предложенния концептуален модел.

Таблица 1. Приноси на дисертационния труд

Приноси	Задача	Глава/Раздел	Публикация
Научен			
Н1	2	2.2	1,3
Научно-приложен			
НП1	3, 5	2.1, 2.2, 2,3	2,4
НП2	4	3.10	
Приложни			
П1	3	3, 3.3, 3.5, 3.6, 3.7	5,6
П2	4	4.1, 4.3	

Апробация

Предложеният модел е валидиран чрез разработване на софтуерен прототип. Същият е тестван с различни входни данни, вкл. за автоматизирано генериране на Пловдивски мобилен университет.

Получените по време на изследването резултати са използвани в **проект** НИ13-ФИФ009, *Многофункционален дигитален университетски ар-*

хив: моделиране, създаване и споделяне, ФНИ при ПУ „Паисий Хилендарски“, 2013 – 2014.

Резултатите са **представени на следните научни конференции:**

- Научна сесия „Дни на науката 2017“, 2 – 3 ноември 2017 г., гр. Пловдив.
- Международна научна конференция „Техника, технологии, образование“ ICTTE 2017, 19 – 20 октомври 2017, гр. Ямбол.
- Научна сесия „Дни на науката 2014“, 31 октомври – 1 ноември 2014 г., Пловдив.
- Научна сесия „Дни на науката 2012“, 25 – 26 октомври 2012, Пловдив.
- IV Национална Научна Конференция 2011, Федерация „Наука и Висше Образование“ при Пловдивски ВУЗ, СУБ – Пловдив.

Перспективи

Проведеното изследване **може да бъде продължено в четири направления:**

- Предложената технология е подходяща за **изграждане на плъгин**, който да се интегрира в браузърите на мобилните устройства.
- **Създаване на мобилно приложение** – реализацията на допълнителен модул за мобилно приложение тип браузър за визуализацията на генерираните страници;
- **Обработка на документи** – предвид обработката на интернет страниците, може се създаде модул за обработка на документи с различни стилове;
- **Създаване на нови релативни шаблони**, които да са с по-добър дизайн по отношение на визуализацията на генерираните страници.

Библиография

- [**Рачовски'2011**] Рачовски Т., Г. Тотков, С. Енков, *Мобилно управление и администриране на е-Обучение в Moodle*, IV Национална Научна Конференция 2011, Федерация „Наука и Висше Образование“ при Пловдивски ВУЗ, СУБ-Пловдив, стр. 117 – 121, 30.04.2011, ISSN: 978-954-9449-44-0.
- [**Рачовски'2012**] Рачовски Т., Б. Ковачев, Г. Тотков, *Към мобилна университетска система*, Научни трудове на СУБ-Пловдив, Серия Б, Естествени и хуманитарни науки, Пловдив, 25 – 26 октомври 2012.
- [**Рачовски'2014**] Рачовски, Т., Г. Тотков, С. Енков, *Принципи при създаването на уеб страници за мобилно обучение*, Научни трудове на СУБ – Пловдив Серия В. Техника и технологии, том XII., Съюз на учените сесия 31 октомври – 1 ноември 2014 г. Пловдив.
- [**Рачовски'2017b**] Рачовски, Т., Е. Хаджиколев, Ст. Хаджиколева, *Софтуерно приложение за генериране на мобилни версии на уеб страници*, Научна сесия „Дни на науката 2017“, 2 – 3 ноември 2017 г., гр. Пловдив (в печат).
- [**Рачовски'2017c**] Рачовски, Т., Ст. Хаджиколева, Е. Хаджиколев, *Софтуерни инструменти за автоматизиран анализ и генериране на мобилни версии на уеб сайтове*, Международна научна конференция „Техника, технологии, образование“ ICTTE 2017, гр. Ямбол, 19 – 20 октомври 2017 (в печат).
- [**Berry'2017**] Berry, *RWD Is Not AWD, What Is the Difference Between Responsive and Adaptive Design?*, Mockplus, 2017, Available at: <https://www.mockplus.com/blog/post/difference-between-responsive-and-adaptive>, (accessed 08.01.2018).
- [**bMobilized**] *bMobilized*. Available at: <http://www.bmobilized.com>, (accessed 08.01.2018).
- [**Bootstrap**] *Bootstrap Documentation*, Available at: <https://getbootstrap.com/docs/4.0/getting-started/introduction/>, (accessed 08.01.2018).
- [**Bulk-Testing-Tool**] *Mobile-Friendly Bulk Testing Tool*. Available at: <https://technicalseo.com/seo-tools/mobile-friendly/>, (accessed 08.01.2018).
- [**Duda**] *Duda*. Available at: <https://www.duda.co>, (accessed 08.01.2018).
- [**goMobi**] *goMobi*. Available at: <https://www.gomobi.info>, (accessed 08.01.2018).

- [Google-API]** Google. *Mobile-Friendly Test API*. Available at: <https://developers.google.com/webmaster-tools/search-console-api/>, (accessed 08.01.2018).
- [Google-TestTool]** Google. *Mobile-Friendly Test Tool*. Available at: <https://search.google.com/test/mobile-friendly/>, (accessed 08.01.2018).
- [Huang'2004]** Y.-W. Huang, F. Yu, C. Hang, C.-H. Tsai, D. Lee, and S.-Y. Kuo. *Securing Web Application Code by Static Analysis and Runtime Protection*. In: Proceedings of the 13th International World Wide Web Conference (WWW 2004), May 2004.
- [Kim'2013]** Kim, B., (2013), *Responsive web design, discoverability, and mobile challenge*, <https://journals.ala.org/index.php/ltr/article/view/4507>, (accessed 08.01.2018).
- [Marcotte'2010]** Marcotte, E., (2010), *Ethan Marcotte Blog*, <https://ethanmarcotte.com/>, (accessed 08.01.2018).
- [Mobify]** Mobify. Available at: <https://www.mobify.com/>, (accessed 08.01.2018).
- [Mohorovičić'2013]** Mohorovičić, S., (2013), *Implementing responsive web design for enhanced web presence*, 36th International Convention on Information & Communication Technology Electronics & Microelectronics (MI-PRO), 20-24 May 2013, Opatija, Croatia.
- [Pastore'2012]** Pastore, S., *The Role of Open Web Standards for Website Development Adhering to the One Web Vision*, International Journal of Engineering and Technology, vol. 2 no. 11 2012 pp. 1824-1834.
- [Rachovski'2017a]** Rachovski, T., S. Hadzhikoleva, E. Hadzhikolev, *Conceptual Model of an Application for Automated Generation of Webpage Mobile Versions*, Tem Journal Vol.6, No.4, 2017, 877-882.
- [Snapmobl]** Snapmobl. Available at: <https://snapmobl.com/en/>, (accessed 08.01.2018).
- [StatCounter]** StatCounter. (2016). *Desktop vs Mobile vs Tablet Market Share Worldwide* (Aug 2016 – Aug 2017). Available at: <http://gs.statcounter.com/press/mobile-and-tablet-internet-usage-exceeds-desktop-for-first-time-worldwide>, (accessed 08.01.2018).
- [Warner'2013]** Janine Warner, *What's the difference between adaptive and responsive web design?*, Digital Family, 2013, Available at: <http://www.digitalfamily.com/tutorials/css-article/whats-the-difference-between-adaptive-and-responsive-web-design/>, (accessed 08.01.2018).

[Wiki-Web-browser] Wikipedia, (2017), *Web browser*,
https://en.wikipedia.org/wiki/Web_browser, (accessed 08.01.2018).

[WompMobile] *WompMobile*. Available at: <http://www.wompmobile.com>,
(accessed 08.01.2018).

Публикации по темата на дисертацията

Rachovski, T., S. Hadzhikoleva, E. Hadzhikolev, *Conceptual Model of an Application for Automated Generation of Webpage Mobile Versions*, Tem Journal Vol. 6, No. 4, 2017, 877 – 882 (индексирана в **Web of science** и **SCOPUS**).

Рачовски, Т., Е. Хаджиколев, Ст. Хаджиколева, *Софтуерно приложение за генериране на мобилни версии на уеб страници*, Сборник доклади от Научна сесия „Дни на науката 2017“, 2 – 3 ноември 2017 г., гр. Пловдив (под печат).

Рачовски, Т., Ст. Хаджиколева, Е. Хаджиколев, *Софтуерни инструменти за автоматизиран анализ и генериране на мобилни версии на уеб сайтове*, Сборник доклади от Международна научна конференция "Техника, технологии, образование" ICTTE 2017, гр. Ямбол, 19 – 20 октомври 2017, 206 – 2013, ISSN 1314-9474.

Рачовски, Т., Г. Тотков, С. Енков, *Принципи при създаването на уеб страници за мобилно обучение*, Научни трудове на СУБ – Пловдив Серия В. Техника и технологии, том XII., Съюз на учените сесия 31 октомври – 1 ноември 2014 г. Пловдив.

Рачовски Т., Б. Ковачев, Г. Тотков, *Към мобилна университетска система*, Научни трудове на СУБ – Пловдив, Серия Б, Естествени и хуманитарни науки, Пловдив, 25 – 26 октомври 2012.

Рачовски Т., Г. Тотков, С. Енков, *Мобилно управление и администриране на е-Обучение в Moodle*, Сборник доклади от IV Национална Научна Конференция 2011, Федерация „Наука и Висше Образование“ при Пловдивски ВУЗ, СУБ-Пловдив, 30.04.2011, стр. 117 – 121, ISSN: 978-954-9449-44-0.

