

**ТЕХНОЛОГИЧНО УЧИЛИЩЕ “ЕЛЕКТРОННИ СИСТЕМИ”
към ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - СОФИЯ**

ДИПЛОМНА РАБОТА

Тема: Приложение за споделяне на снимки, в което целта на потребителите
ще е да пресъздават дадени изображения на различни теми

Дипломант:
Виктория Радкова

Научен ръководител:
инж. Марио Бенов

С О Ф И Я
2 0 2 0

Увод

Социалните медии са уеб-базирани среди за общуване, които биват създавани и използвани за комуникация, споделяне на информация, новини, снимки, файлове и т.н. Те се използват както от отделни индивиди, така и от социални групи за споделяне на мнение, опит, виждания и всякаква друга информация. Едни от най-известните социални медии са: Facebook, Instagram, Twitter, Tumbler.

Социалните медии имат множество различни форми. В някои случаи комуникацията е директен диалог (в чат приложения, като Messenger), в други тя е посредствена – чрез статии.

Все по известно става в социалните медии да се интегрират игри (Social-network games). Тези игри обикновено поддържат multiplayer, което позволява на потребителите да играят с приятелите си или дори да правят нови запознанства онлайн.

Целта на настоящата дипломна работа е да се реализира мобилно приложение за споделяне на снимки (социална медия), включващо игра, в която целта на потребителите ще е да пресъздават дадени изображения на различни теми, като по този начин ще трупат точки, ще качват нива и ще получават награди.

Глава 1

Въведение

1.1 Социалните медии

За да достигнат социалните медии мащаба, който имат днес, те извървяват дълъг път. Стартирали като канал за трансфер на информация, те се превръщат бързо в средишна точка на социалното взаимодействие, предоставяйки възможности, които никой друг информационен канал не може да предложи.

Социалните медии са познати от преди стотици години, когато главният начин за комуникация бил посредством ръчно написани писма. В последствие през 19-ти век се появяват първите онлайн чат приложения (като например Talkomatic)[1].

В днешно време социалните медии са навсякъде и се използват от почти всеки. Те ни свързват и ни предоставят множество възможности за развитие и комуникация. Чрез тях могат лесно и удобно да общуват хора по целия свят.

Едни от най-известните социални медии днес са започнали съществуването си като прост училищен проект. Пример за това е Facebook, социална медия която всеки ден става все по-мощна и по-мощна, добавяйки нови функционалности в приложението си, като например социални онлайн игри.

1.2 Социалните онлайн игри

Социалните игри не са нещо ново – те са просто игри които играем с други хора. Появили са се преди хиляди години. Най-древните игри били открити в археологически разкопки. С времето социалните игри станали толкова много, че вече дори не могат да се преброят.

Социалните онлайн игри (Social-network games) първоначално били имплементирани като браузър игри, а в последствие и като мобилни приложения. Много наподобяват обикновените видео игри, но са проектирани за забавление с приятели в свободното време и обикновено не изискват постоянство. Тъй като не са и толкова предизвикателни, те обикновено разчитат на различни техники за привличане на потребителите.

Социалните онлайн игри са базирани на client-server (клиент-сървър) комуникация. Клиентът в мрежата се реализира посредством различни уеб технологии, като Flash, HTML5, PHP, JavaScript. Когато игрите се преместили на мобилни устройства за реализацията им било нужно използването и на други технологии, като Java, Objective-C, Swift, C++[2].

Социални онлайн игри се срещат и в социалните медии. Пример за това са игрите във Facebook, използващи client-server технологията (Uno, 8 Ball Pool, не се сърди човече, шах и т.н.). При тях участието на други потребители е задължително, за да започне играта.

1.3 Обзор на различните технологии за разработка на мобилни приложения

Има много технологии за разработка на мобилни приложения, които се използват за определени или кръстосани платформи. Петте най-разпространени езика за разработка на мобилни приложения са следните[3]:

1.3.1 Swift

Swift предлага много функции и е лесен за работа и поддръжка. Може да се използва за разработка на Apple приложения.

1.3.2 C++

C++ е основата на много програмни езици. Прост, ефективен, универсален инструмент, който може да се използва за множество платформи.

1.3.3 Java

Java е обектно ориентиран език за програмиране, приет за основен език за разработка на мобилни приложения. Има много готови библиотеки и е лесен за употреба.

1.3.4 HTML5

HTML5 е един от най-удобните езици за разработка на уеб-базирани мобилни приложения.

1.3.5 PHP

PHP е обектно ориентиран език на програмиране, подходящ за разработка на динамични мобилни приложения, използващи база данни.

Съществуват множество инструменти за създаване на приложения „от нулата“. Едни от най-популярните са[3]:

1.3.6 PhoneGap

PhoneGap, още познат като Apache Cordova, използва CSS3, HTML5 и JavaScript за да създава приложения за различни платформи, като Android, Windows и iOS.

1.3.7 Appcelerator

Appcelerator е отличен избор за създаване на приложения, използващи хардуер. Използва CSS3, HTML5, JavaScript и Ruby за изработване на приложения за различни платформи (Android, Windows, iOS).

1.3.8 RhoMobile

RhoMobile използва CSS3, HTML5, JavaScript и Ruby за лесна и удобна разработка на мобилни приложения за най-мащабните платформи.

1.3.9 WidgetPad

WidgetPad е една от най-добрите среди за разработка на мобилни приложения. Използва езици като HTML5 и JavaScript, предоставяйки удобно редактиране и разпространение на код за Android, Windows и iOS платформи.

1.3.10 MoSync

MoSync поддържа езици като JavaScript, PHP, Ruby, and Python. Интегриран е с базирано на Eclipse IDE и позволява разработка на приложения за различни платформи, използвайки C/C++.

MySQL е сред най-използваните и популярни технологии за бази данни, така че са създадени много инструменти, за да се направят процесите на проектиране, създаване и администриране на бази данни по-лесни и удобни. Въпреки това, с всички възможни опции, може да бъде трудно да изберете един инструмент, който да отговаря най-добре на вашите изисквания. Най-известните MySQL GUI (Graphical user interface - Графичен потребителски интерфейс) инструменти са[4]:

1.3.11 MySQL Workbench

MySQL Workbench е унифициран визуален инструмент за архитекти на бази данни, разработчици и DBA. Той предоставя различни функции за проектиране, разработване и администриране на бази данни. Той е достъпен в 3 издания - Community, Standard и Enterprise. Community е безплатно издание с отворен код, предлагащо основен набор от функции, а останалите две са търговски издания с разширена функционалност.

1.3.12 PhpMyAdmin

PhpMyAdmin е уеб-базиран инструмент за администриране на MySQL, написан на PHP. Това е един от най-популярните по рода си инструменти, главно благодарение на това колко е лесен за използване и на факта, че е напълно безплатен. Може да липсват някои разширени функции, които други инструменти предоставят, но основната му функционалност е достатъчна, за да ви помогне да създадете функциониращи уебсайтове без напреднали познания за MySQL.

1.3.13 HeidiSQL

HeidiSQL е лек безплатен инструмент за работа с бази данни, било то сърфиране или редактиране на данни, създаване и промяна на таблици, управление на потребителски привилегии или други задачи.

1.3.14 Toad Edge

Toad Edge за MySQL предоставя набор от инструменти за разработка и администриране на бази данни с функции като сравняване и синхронизация на схеми, SQL монитор на заявки, стабилен импорт и експортиране на данни, създаване на моментни снимки на базата данни, разширен редактор на JSON и други. Toad Edge е наличен за Windows и Mac OS X.

За по лесно тестване на код може да се използва уеб сървър. Някои от актуалните приложения за хостване на уеб сървър са:

1.3.15 XAMPP

XAMPP е софтуер с отворен код, разработен от Apache friends. Този XAMPP софтуер дава подходяща среда за тестване на проекти, като MYSQL, PHP, Apache и Perl на локалния компютър.

1.3.16 WampServer

WampServer се отнася до софтуерен стак за операционната система Microsoft Windows, създаден от Romain Bourdon и състоящ се от уеб сървъра Apache, OpenSSL за поддръжка на SSL, MySQL база данни и програмният език PHP.

1.4 Алгоритми за сходство между изображения

Две снимки са сходни ако на тях има сходни обекти. Обикновено „пиксел по пиксел“ сравнението не е особено ефективно. Първо трябва да се разпознаят и запазят определени обекти от снимките и после те да бъдат сравнени.

Един от най-добрите алгоритми, които могат да бъдат използвани е ConvNet (convolutional neural network) - спираловидна невронна мрежа, точно както при ImageNet. Друг вариант за генериране на функции е използването на стандартни методи за извличане на информация от изображения, като SIFT (scale-invariant feature transform - мащабно-инвариантно преобразуване на характеристиките) и SURF (speeded up robust features – ускорени стабилни функции), но тъй като тези методи извличат общи функции, ефектът няма да бъде толкова голям както при ConvNet.

1.4.1 ConvNet

ConvNet, още позната като CNN, е спираловидна невронна мрежа, която е клас на дълбоките невронни мрежи, често използвани за анализиране на изображения. Тя има приложение в разпознаването на снимки и видеа, системи за предсказване и класификация на изображения.

Този вид невронни мрежи са узаконените версии на многослойните перцептрони. Многослойните перцептрони обикновено означават напълно свързани мрежи, тоест всеки неврон в един слой е свързан с всички неврони в следващия слой. Връзката на тези мрежи ги прави склонни към прецизиране на

данните. CNN използват различен подход към регуларизация, те се възползват от йерархичния модел в данните и събират по-сложни модели, използвайки по-малки и по-прости. Следователно, в мащаба на свързаност и сложност, CNN са в долната крайност[5].

1.4.2 SIFT

SIFT е алгоритъм за откриване и описание на локални характеристики в изображенията. Приложенията, използващи SIFT, включват разпознаване на обекти, роботизирано картографиране и навигация, 3D моделиране, разпознаване на жестове, проследяване на видео, индивидуална идентификация на дивата природа и придвижване на мачове.

SIFT ключовите точки на обектите първо се извличат от набор от референтни изображения и се съхраняват в база данни. Обект в ново изображение се разпознава чрез индивидуално сравняване на всяка характеристика в него с базата данни и намиране на характеристики за съвпадение на кандидати, въз основа на евклидово разстояние от техните вектори. От пълния набор от съвпадения са идентифицирани подмножества от ключови точки, които се поставят върху обекта и неговото местоположение, мащаб и ориентация в новото изображение, за да се филтрират добрите съвпадения. Определянето на последователни групи се извършва бързо чрез използване на ефективна хеш-таблица за изпълнение на обобщената трансформация на Хъф. Всяка група от 3 или повече функции, които се

съгласяват върху даден обект и позицията му, след това подлежи на допълнителна подробна проверка на модела и впоследствие остатъците се изхвърлят. Накрая вероятността, че определен набор от характеристики показва наличието на обект, се изчислява, като се има предвид точността на напасване и броя на вероятните неверни съвпадения. Съвпаденията на обектите, които преминават всички тези тестове, могат да бъдат идентифицирани като правилни[6].

1.4.3 SURF

В компютърната визуализация SURF (ускорени стабилни функции) са патентован локален детектор и дескриптор на функции. Може да се използва за задачи като разпознаване на обекти, регистрация на изображения, класификация или 3D реконструкция. Отчасти е вдъхновен от дескриптора на мащабно-инвариантното преобразуване на характеристики (SIFT). Стандартната версия на SURF е няколко пъти по-бърза от SIFT и е по-устойчива срещу различни трансформации на изображения от SIFT.

За откриване на точки на интерес, SURF използва цялостно сближаване, което може да се изчисли с 3 цели операции, като се използва предварително изчислено интегрално изображение. Дескрипторът на характеристиките му се основава на сумата от вълновия отговор на Хаар около точката на интерес. Те могат да бъдат изчислени и с помощта на интегралното изображение[7].

1.5 Съществуващи подобни продукти

1.5.1 Pinterest

Pinterest е една от най-известните социални медии. Там хората споделят вдъхновяващи снимки и видеа на най-различни теми, като всеки може да ги запазва в своя профил. Под всяко изображение хората могат да споделят свои снимки или видеа наподобяващи оригинала.

1.5.2 GuruShots

GuruShots е приложение за споделяне на оригинални фотографии. В него всеки може да изразява себе си чрез снимки, а други хора могат да оценяват творбите им. Предлага ежедневни предизвикателства на различни теми. Ранка ти в класацията се определя от това колко хора са харесали снимките ти.

1.5.3 Facebook

Facebook, може би най-мощната социална медия, използвана от милиарди потребители, поддържа множество социални онлайн игри. Позволява изпращане на покани за съвместна игра. При повечето игри има ежедневни награди, които да поддържат интереса на потребителите.

Глава 2

Проектиране

2.1 Изисквания към програмния продукт

Целта на дипломната работа е да се реализира социална медия с интегрирана игра. Когато потребител се регистрира и влезе в профила си, той ще може да разглежда множество снимки на различни теми. Ще бъдат организирани турнири с най-различни награди (баджове, стикери, темплейти за самото приложение и т.н.). Всеки ще има възможността да избере турнир, в който да се включи и да се опита да пресъздаде определена снимка, на предварително обявена тема. Отделно от играта потребителите ще могат да качват свои авторски фотографии. Колкото повече снимки качват, толкова по-нагоре в класацията ще се изкачват.

Основни изисквания:

- ✓ Регистрация на потребители
- ✓ Търсене на потребители
- ✓ Качване на изображения
- ✓ Търсене на изображения
- ✓ Списък с последователи
- ✓ Турнири

2.2 Софтуерни средства

2.2.1 Apache Cordova

Най-подходящо за реализация на дипломната работа е приложението Apache Cordova, чрез което могат да се правят мобилни приложения посредством HTML5, CSS3 и JavaScript.

Apache Cordova, създадена от Nitobi, е едно от най-известните приложения за разработка на мобилни приложения. Adobe Systems закупуват Nitobi през 2011 г., преобразуват го във PhoneGap и по-късно пускат общодостъпна версия, наречена Apache Cordova. Apache Cordova дава възможност на софтуерните програмисти да изграждат приложения за мобилни устройства, използващи CSS3, HTML5 и JavaScript, вместо да разчитат на специфични за платформата интерфейси, като тези в Android, iOS или Windows Phone. Получените приложения са хибридни, което означава, че те не са нито истинско мобилно приложение (тъй като цялото визуализиране на оформлението се осъществява чрез уеб изгледи), нито чисто уеб базирано (защото те не са само уеб приложения, а са проектирани като приложения за разпространение). Смесването на чисти и хибридни кодови фрагменти е възможно от версия 1.9[8].

2.2.1 MySQL

MySQL е сред най-използваните и популярни технологии за бази данни, така че са създадени много инструменти, за да се направят процесите на проектиране, създаване и администриране на бази данни по-лесни и удобни. Има много предимства при работата с MySQL[10]:

2.2.1.1 Сигурност

MySQL е световно известен с това, че е най-сигурната и надеждна система за управление на бази данни, използвана в популярни уеб приложения като WordPress, Drupal, Joomla, Facebook и Twitter. Сигурността на данните и поддръжката за транзакционна обработка, придружаваща неотдавнашната версия на MySQL, може да бъде от голяма полза за всеки бизнес, особено ако става въпрос за електронна търговия, която включва чести парични преводи.

2.2.1.2 Висока производителност

Независимо дали става въпрос за уебсайт за електронна търговия, който получава милион заявки всеки ден или високоскоростна система за обработка на транзакции, MySQL е проектиран да отговаря дори на най-взискателните приложения, като същевременно гарантира оптимална скорост, текстови индекси и уникални кешове за памет за подобрена производителност.

2.2.1.3 Мащабируемост по поръчка

MySQL предлага несравнима мащабируемост, за да улесни управлението на дълбоко вградените приложения, използвайки по-малък отпечатък дори в

масивни складове, които подреждат терабайти от данни. Гъвкавостта при поискване е звездната характеристика на MySQL. Този източник с отворен код позволява цялостно персонализиране на бизнеса в електронната търговия с уникални изисквания за сървър на база данни.

2.2.1.4 Пълен контрол на работния процес

С средното време за изтегляне и инсталиране е по-малко от 30 минути, MySQL означава използваемост от първия ден. Независимо дали вашата платформа е Linux, Microsoft, Macintosh или UNIX, MySQL е цялостно решение с функции за самоуправление, които автоматизират всичко - от разширяване и конфигуриране на пространството до проектиране на данни и администриране на база данни.

2.2.1.5 Гъвкавост на отворения код

Всички страхове и притеснения, които възникват при решение с отворен код, могат да бъдат прекратени с денонощната поддръжка на MySQL и обезщетението на предприятието. Защитената обработка и надежден софтуер на MySQL се комбинират, за да осигурят ефективни транзакции за проекти с голям обем. Това прави поддръжката, отстраняването на грешки и надстройките бързо и лесно, като същевременно подобрява изживяването на крайния потребител.

2.2.1.6 Намалени общи разходи за собственост

Чрез миграция на текущите приложения за бази данни към MySQL предприятията се радват на значителни икономии на разходи за нови проекти.

Надеждността и лекотата на управление, които придружават MySQL, спестяват времето ви за отстраняване на неизправности, което иначе се губи при коригиране на проблеми с престоя и проблеми с производителността.

2.2.2 PhpMyAdmin

PhpMyAdmin е уеб-базиран инструмент за администриране на MySQL, написан на PHP. Това е един от най-популярните по рода си инструменти, главно благодарение на това колко е лесен за използване и на факта, че е напълно безплатен, точно заради това е подходящ за настоящата дипломна работа. Може да липсват някои разширени функции, които други инструменти предоставят, но основната му функционалност е достатъчна, за да ви помогне да създадете функциониращи уебсайтове без напреднали познания за MySQL.

2.2.3 XAMPP

XAMPP е софтуер с отворен код, разработен от Apache friends. Този XAMPP софтуер дава подходяща среда за тестване на проекти, като MYSQL, PHP, Apache и Perl на локалния компютър. Инсталирането е много лесно и съдържа всички необходими инструменти, а също така осигурява GUI базиран контролен панел за стартиране и спиране на модули / услуги и актуализиране на конфигурационни файлове.

Глава 3

Реализация

3.1 Регистрация на потребители

3.1.1 Схема на потребителите

В схемата на потребителите се описва начина, по който изглежда информацията в базата данни при регистрация. За всеки потребител се пази:

- 1.) Идентификационен номер
- 2.) Потребителско име
- 3.) Парола
- 4.) Имейл
- 5.) Профилна снимка по подразбиране
- 6.) Снимка на корицата по подразбиране

```
CREATE TABLE `registration` (  
  `id` int(11) NOT NULL,  
  `username` varchar(20) NOT NULL,  
  `password` varchar(20) NOT NULL,  
  `email` varchar(40) NOT NULL,  
  `profile` longblob NOT NULL,  
  `cover` longblob NOT NULL
```

Фигура 3.1 Схема на потребителите

3.1.2 Връзка с базата данни

```
<?php
$username = $_POST['username'];
$password = $_POST['password'];
$email = $_POST['email'];

if (!empty($username) || !empty($password) || !empty($email)) {
    $host = "localhost";
    $dbUsername = "root";
    $dbPassword = "";
    $dbname = "victorious_shots";
    //create connection
    $conn = new mysqli($host, $dbUsername, $dbPassword, $dbname);

    if (mysqli_connect_error()) {
        die('Connect Error('. mysqli_connect_errno().')'. mysqli_connect_error());
    } else {

        $SELECT = "SELECT username From registration Where username = ? Limit 1";
        $INSERT = "INSERT Into registration (username, password, email) values(?, ?, ?)";
        //Prepare statement
        $stmt = $conn->prepare($SELECT);
        $stmt->bind_param("s", $username);
        $stmt->execute();
        $stmt->bind_result($username);
        $stmt->store_result();
        $rnum = $stmt->num_rows;

        if ($rnum==0) {
            $stmt->close();
            $stmt = $conn->prepare($INSERT);
            $stmt->bind_param("sss", $username, $password, $email);
            $stmt->execute();

            $message = "Account created successfully!";
            echo "<script type='text/javascript'>alert('$message');</script>";

            $myfile = fopen("home.php", "a+") or die("Unable to open file!");
            echo fread($myfile,filesize("home.php"));
            fclose($myfile);
        }
        else {
            $message1 = "Username is already taken!";
            echo "<script type='text/javascript'>alert('$message1');</script>";

            $myfileagain = fopen("sign up.html", "a+") or die("Unable to open file!");
            echo fread($myfileagain,filesize("sign up.html"));
            fclose($myfileagain);
        }
        $stmt->close();
        $conn->close();
    }
} else {
    echo "All fields are required";
    die();
}
?>
```

Фигура 3.2 Връзка с базата данни

3.2 Качване и търсене на изображения

При качване на снимка, освен файл в jpg формат, се подават и тагове, разделени с „#“. Снимките се изпращат в една таблица, а таговете в друга, като в таблицата с таговете запазваме и идентификационния номер на снимката.

3.2.1 Информация за изображенията

```
CREATE TABLE `images` (  
  `id` int(11) NOT NULL,  
  `name` varchar(40) NOT NULL,  
  `image` longblob NOT NULL,  
  `username` varchar(40) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Фигура 3.3 Информация за изображенията

3.2.2 Информация за тагове към изображенията

```
CREATE TABLE `tags` (  
  `id` int(11) NOT NULL,  
  `tag` varchar(100) NOT NULL,  
  `img_id` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Фигура 3.4 Информация за тагове към изображенията

3.2.3 Запазване на изображенията и таговете към тях

```
function saveimage($name, $image, $username, $tags)
{
    $con = mysqli_connect("localhost", "root", "");
    mysqli_select_db($con, "victorious_shots");
    $qry = "insert into images(name, image, username) values ('$name', '$image', '$username')";
    $result = mysqli_query($con, $qry);
    $imgid = mysqli_insert_id($con);
    $tagsexplode = explode(" ", $tags);
    for($i = 0; $i < count($tagsexplode); $i++){
        $qrytags = "insert into tags(img_id, tag) values ('$imgid', '$tagsexplode[$i]')";
        $resulttags = mysqli_query($con, $qrytags);
    }
    if($result)
    {
        $message1 = "Image uploaded.";
        echo "<script type='text/javascript'>alert('$message1');</script>";
    }
    else
    {
        $message2 = "Image not uploaded.";
        echo "<script type='text/javascript'>alert('$message2');</script>";
    }
}
```

Фигура 3.5 Запазване на изображенията и таговете към тях

3.2.4 Връзка с базата данни

```
<?php
if(isset($_POST['submit']))
{
    $info = @getimagesize($_FILES['image']['tmp_name']);
    if($info == FALSE)
    {
        $message = "Please select an image.";
        echo "<script type='text/javascript'>alert('$message');</script>";
    }
    else
    {
        $image = addslashes($_FILES['image']['tmp_name']);
        $name = addslashes($_FILES['image']['name']);
        $image = file_get_contents($image);
        $image = base64_encode($image);
        $username = $_REQUEST['username'];
        $tags = $_REQUEST['tags'];
        saveimage($name, $image, $username, $tags);
    }
}
displayimage($queries['username']);

function saveimage($name, $image, $username, $tags)
{
    $con = mysqli_connect("localhost", "root", "");
    mysqli_select_db($con, "victorious_shots");
    $qry = "insert into images(name, image, username) values ('$name', '$image', '$username')";
    $result = mysqli_query($con, $qry);
    $imgid = mysqli_insert_id($con);
    $tagsexplode = explode(" ", $tags);
    for($i = 0; $i < count($tagsexplode); $i++){
        $qrytags = "insert into tags(img_id, tag) values ('$imgid', '$tagsexplode[$i]')";
        $resulttags = mysqli_query($con, $qrytags);
    }
    if($result)
    {
        $message1 = "Image uploaded.";
        echo "<script type='text/javascript'>alert('$message1');</script>";
    }
    else
    {
        $message2 = "Image not uploaded.";
        echo "<script type='text/javascript'>alert('$message2');</script>";
    }
}
```

Фигура 3.6 Връзка с базата данни

3.2.5 Визуализация на изображенията

```
function displayimage($login_session)
{
    $con = mysqli_connect("localhost", "root", "");
    mysqli_select_db($con, "victorious_shots");
    $qry = "select * from images where username='".$login_session.'" order by id desc";
    $result = mysqli_query($con, $qry);
    echo '<div class="cont">';

    while($row = mysqli_fetch_array($result))
    {
        echo '<div class="idiv">';
        echo '';
        $qrytag = "select * from tags where img_id='".$row[0].'" ";
        $res = mysqli_query($con, $qrytag);
        echo '</div>';
    }
}
```

Фигура 3.7 Визуализация на изображенията

3.2.6 Визуализация на профилната снимка

```
<?php
function displayimagepro($login_session)
{
    $con = mysqli_connect("localhost", "root", "");
    mysqli_select_db($con, "victorious_shots");
    $qry = "select profile from registration where username='".$login_session.'" ";
    $result = mysqli_query($con, $qry);
    // echo '<div class="cont">';
    $row = mysqli_fetch_array($result);
    echo '';

    // echo '</div>';
    mysqli_close($con);
}
displayimagepro($queries['username']);
?>
```

Фигура 3.8 Визуализация на профилната снимка

3.2.7 Търсене на изображения и потребителски имена

При търсене на изображения потребителят трябва да въведе „#“ и думата за която търси, защото в противен случай (ако не въведе „#“) ще търси потребителско име.

```
<?php
$con = mysqli_connect("localhost","root","") or die ("could not connect");
mysqli_select_db($con, "victorious_shots") or die ("could not find db");
$output = '';

if(isset($_POST['searchVal'])){
    $searchq = $_POST['searchVal'];
    if(strpos($searchq, '#') === 0){
        $query = mysqli_query($con, "SELECT image FROM images WHERE id in (SELECT img_id FROM tags
            WHERE tag='$searchq')") or die ("could not search");
        $count = mysqli_num_rows($query);
        if($count==0){
            $output = 'No results found.';
        }else{
            while($row = mysqli_fetch_array($query)){
                $img = $row['image'];
                $output .= '<div class="divimgsearch"><a href="profile.php?username=\'.'.$uname.'\'>\'.'.$uname.'\'</a></div>';
            }
        }
    }
}
print("$output");
?>
```

Фигура 3.9 Търсене на изображения и потребителски имена

3.2.8 Визуализация на търсените изображения

```
<script type="text/javascript">
    function searchq(){
        var searchText = $("input[name='search']").val();
        if(searchText.length == 0){
            $("#output").html('');
        }
        else{
            $.post("search.php", {searchVal: searchText}, function(output){
                $("#output").html(output);
            });
        }
    }
</script>
```

...

```
<?php
function displayimage()
{
    $con = mysqli_connect("localhost", "root", "");
    mysqli_select_db($con, "victorious_shots");
    $qry = "select * from images order by id desc";
    $result = mysqli_query($con, $qry);
    $i = 1;
    while($row = mysqli_fetch_array($result))
    {
        if($i%2!=0){
            echo '<div>';
        }
        echo '';
        if($i%2==0){
            echo '</div>';
        }

        $i++;
    }
    mysqli_close($con);
}
displayimage();
?>
```

Фигура 3.10 Визуализация на търсените изображения

3.3 Списък с последователи

Всеки потребител може да потърси друг, посредством потребителско име и да го последва.

3.3.1 Информация за последователите

```
CREATE TABLE `followers` (  
  `id` int(11) NOT NULL,  
  `follower` varchar(40) NOT NULL,  
  `followed` varchar(40) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Фигура 3.11 Информация за последователите

3.3.2 Връзка с базата данни

```
<?php  
$con = mysqli_connect("localhost","root","") or die ("could not connect");  
mysqli_select_db($con, "victorious_shots") or die ("could not find db");  
  
$followeruname = $_POST['follower'];  
$followeduname = $_POST['followed'];  
  
$query = mysqli_query($con, "insert into followers(follower, followed) values ('$  
  followeruname', '$followeduname')");  
$count = mysqli_num_rows($query);  
  
exit();  
?>
```

Фигура 3.12 Връзка с базата данни

3.4 Известия

Всеки потребител получава известия, с информация, за това кой го е последвал.

3.4.1 Информация за известията

```
CREATE TABLE `notifications` (  
  `id` int(11) NOT NULL,  
  `username` varchar(40) NOT NULL,  
  `notification` varchar(100) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Фигура 3.13 Информация за известията

3.4.2 Връзка с базата данни

```
<?php  
$con = mysqli_connect("localhost","root","") or die ("could not connect");  
mysqli_select_db($con, "victorious_shots") or die ("could not find db");  
  
$followeruname = $_POST['followed'];  
$followeduname = $_POST['follower'];  
$notification = " Hey! ".$followeruname." just followed you!";  
  
$query = mysqli_query($con, "insert into followers(followed, follower)  
  values ('$followeduname', '$followeruname')");  
$count = mysqli_num_rows($query);  
  
$querynotifications = mysqli_query($con, "insert into notifications(  
  username, notification) values ('$followeduname', '$notification')")  
  ;  
  
exit();  
?>
```

Фигура 3.14 Връзка с базата данни

3.4.3 Визуализация на известията

```
<?php
function displaynotification($login_session)
{
    $con = mysqli_connect("localhost", "root", "");
    mysqli_select_db($con, "victorious_shots");
    $qry = "select * from notifications where username='".$login_session.'" order by id
        desc";
    $result = mysqli_query($con, $qry);

    while($row = mysqli_fetch_array($result))
    {
        echo '<tr class="tr"><td><i class="fa fa-bell">'.$row['notification'].'</i></td></tr>
        ';
    }
    mysqli_close($con);
}
displaynotification($login_session);
?>
```

Фигура 3.15 Визуализация на известията

3.5 Начална страница

Когато потребител качи снимка в профила си, тя автоматично отива в началната страница със снимки. Така всички потребители могат да видят авторската му фотография и да се вдъхновят от нея.

3.5.1 Визуализация на снимките, качени от потребители

```
<?php
function displayimage()
{
    $con = mysqli_connect("localhost", "root", "");
    mysqli_select_db($con, "victorious_shots");
    $qry = "select * from images order by id desc";
    $result = mysqli_query($con, $qry);
    $i = 1;
    while($row = mysqli_fetch_array($result))
    {
        if($i%2!=0){
            echo '<div>';
        }
        echo '';
        if($i%2==0){
            echo '</div>';
        }

        $i++;
    }
    mysqli_close($con);
}
displayimage();
?>
```

Фигура 3.16 Визуализация на снимките, качени от потребители

3.6 Турнири

3.6.1 Информация за турнирите

```
CREATE TABLE `challenges` (
  `id` int(11) NOT NULL,
  `username` varchar(40) NOT NULL,
  `challenge` varchar(40) NOT NULL,
  `image` longblob NOT NULL,
  `result` double NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Фигура 3.17 Информация за турнирите

3.6.2 Връзка с базата данни

```
<?php
function saveimage($username, $challenge, $image){
    $con = mysqli_connect("localhost", "root", "");
    mysqli_select_db($con, "victorious_shots");
    $qry = "insert into challenges(username, challenge, image) values
    ('$username','$challenge', '$image')";
    $result = mysqli_query($con, $qry);
    $imgid = mysqli_insert_id($con);
    if($result)
    {
        $message1 = "Image uploaded.";
        echo "<script type='text/javascript'>alert('$message1');</script>";
    }
    else
    {
        $message2 = "Image not uploaded.";
        echo "<script type='text/javascript'>alert('$message2');</script>";
    }
}
if(isset($_POST['submit']))
{
    $info = @getimagesize($_FILES['image']['tmp_name']);
    if($info == FALSE)
    {
        $message = "Please select an image.";
        echo "<script type='text/javascript'>alert('$message');</script>";
    }
    else
    {
        $image = addslashes($_FILES['image']['tmp_name']);
        $name = addslashes($_FILES['image']['name']);
        $image = file_get_contents($image);
        $image = base64_encode($image);
        $username = $_REQUEST['username'];
        $challenge = $_REQUEST['challenge'];
        saveimage($username, $challenge, $image);
    }
}
?>
```

Фигура 3.18 Връзка с базата данни

Глава 4

Ръководство за потребителя

4.1 Изисквания

Минималната памет която се изисква, за да може да бъде свалено приложението са 80MB. Приложението не може да бъде пуснато на Android версия по-стара от 8.0.0.

4.2 Инсталация

За да се отвори приложението трябва да бъдат инсталирани:

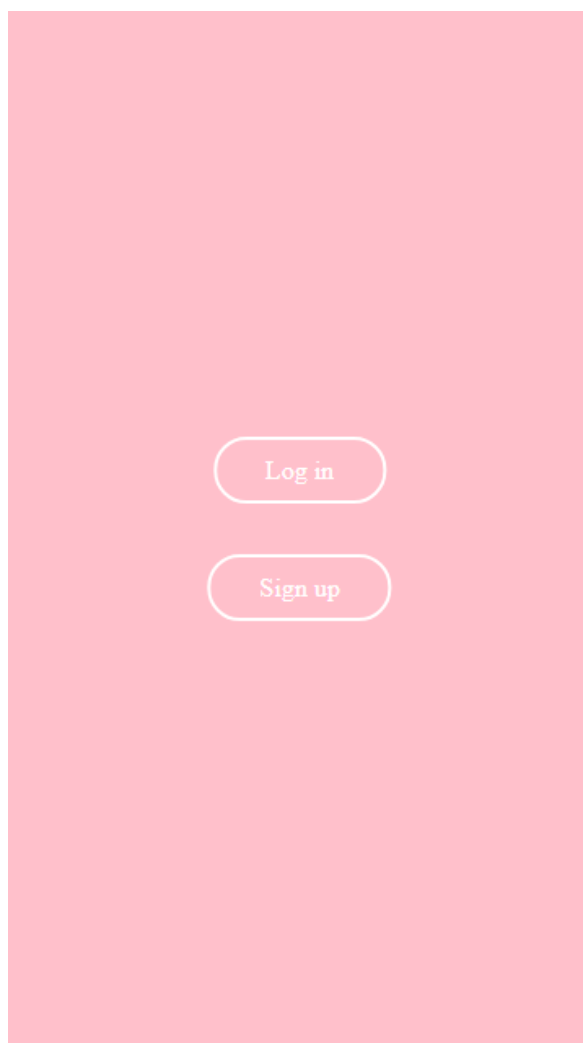
- ✓ XAMPP
- ✓ PHP
- ✓ PhpMyAdmin
- ✓ MySQL

Файловете, които се намират в папка App->www се поставят в папка хampp->htdocs, като предварително се премахват файловете от нея. Пуска се *xampp-control*, стартират се *Apache* и *MySQL*. В брауъра се пише localhost. За да бъде изгледа като на мобилно приложение се натиска F12.

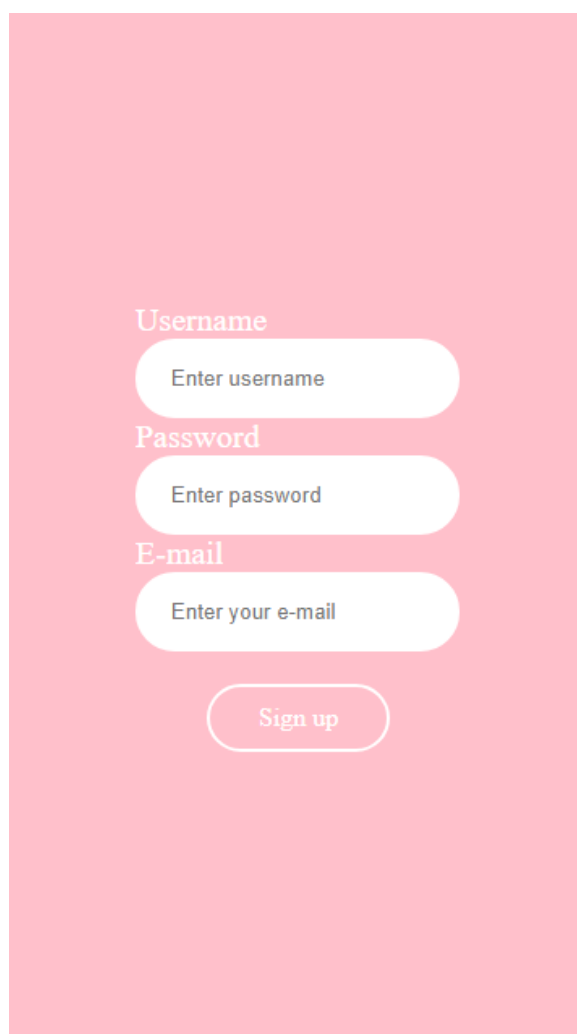
4.3 Ръководство за потребителя

4.3.1 Първи стъпки

В случай че потребителят е нов и няма регистрация, може да си направи такава чрез *Sign Up* бутона. Ако вече има налична регистрация, може да си влезе в профила чрез бутона *Log In*.



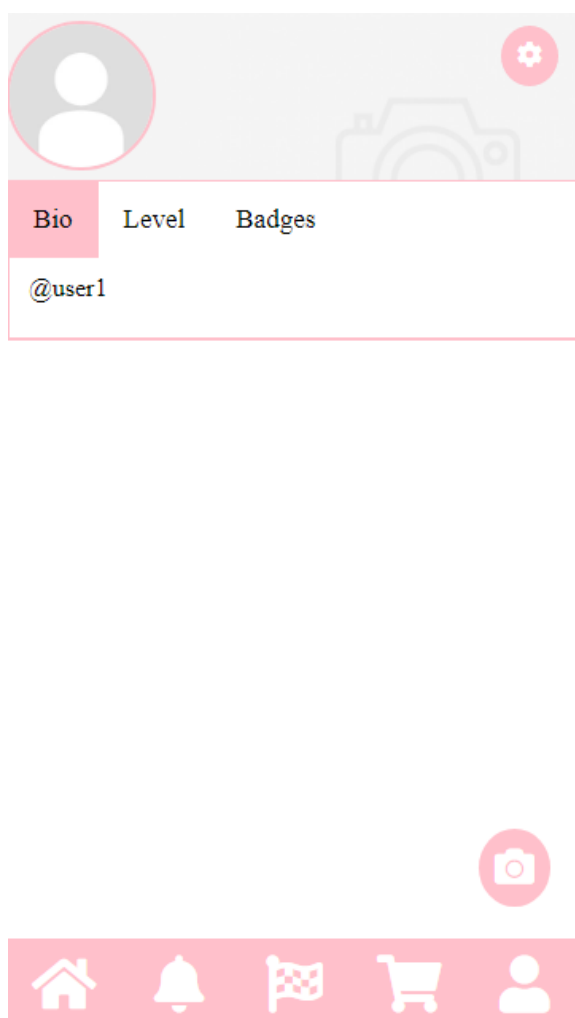
Фигура 4.1 Начало



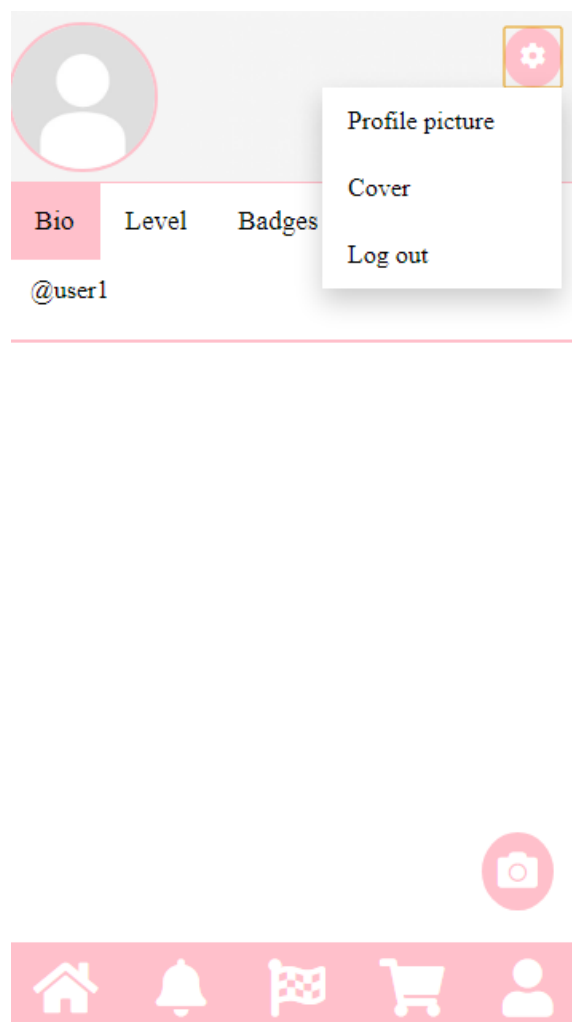
Фигура 4.2 Регистрация

4.3.2 Персонализация на профила

След като потребителя влезе в своя акаунт, той може да промени своята профилна снимка и тази на корицата си, чрез достъпване на профил страницата (иконката с човек долу вдясно). За промяна на профилната снимка се използва бутонът *Profile picture*, а за промяна на корицата бутонът – *Cover*. За да се отвори менюто с тези опции се натиска бутонът горе вдясно.

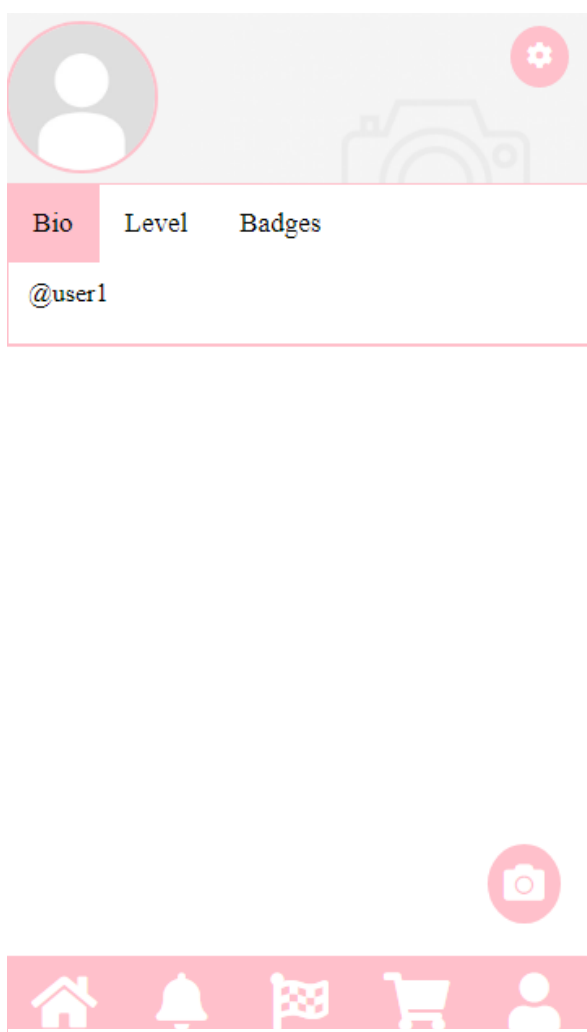


Фигура 4.3 Профил

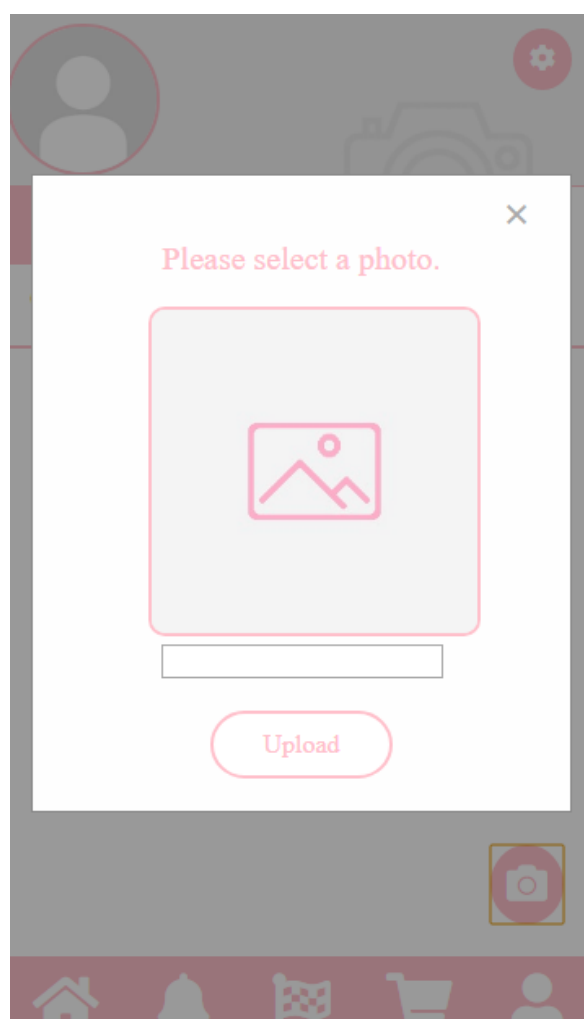


Фигура 4.4 Настройки

Всеки потребител може да качва снимки в профила си, които други потребители ще могат да виждат. Това става чрез бутона долу вдясно, с иконка на камера. След натискане на бутона излиза форма за качване на снимка. За да избере снимка за качване потребителят трябва да натисне полето за снимка. След това трябва да въведе тагове отнасящи се за снимката, като ги разделя с „#“. Накрая за да качи снимката трябва да натисне бутонът *Upload*.

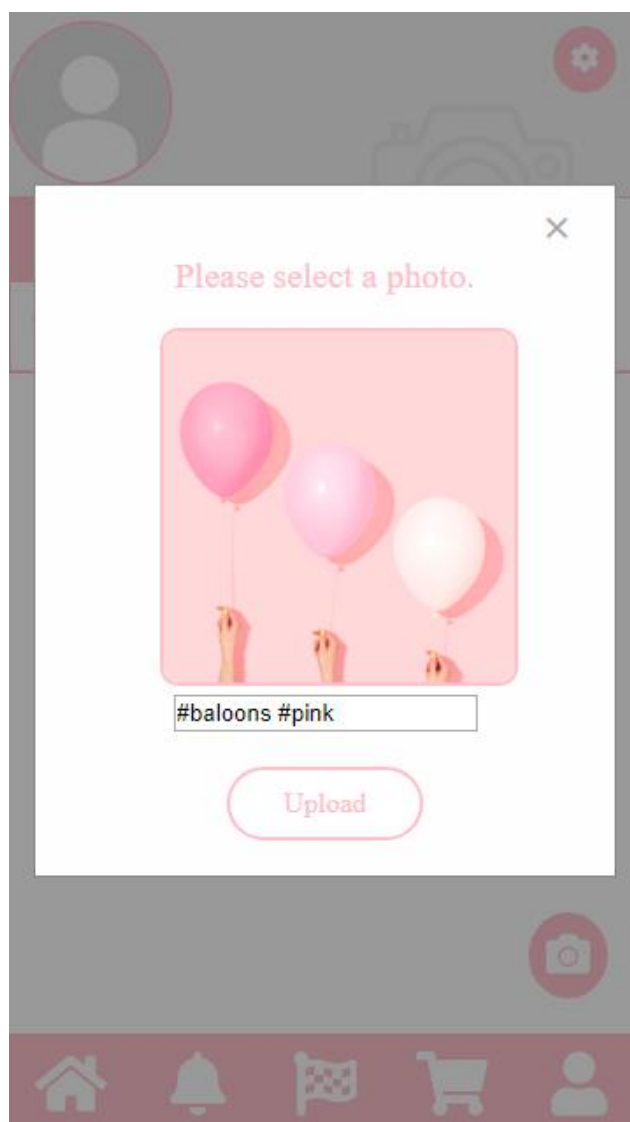


Фигура 4.5 Профил

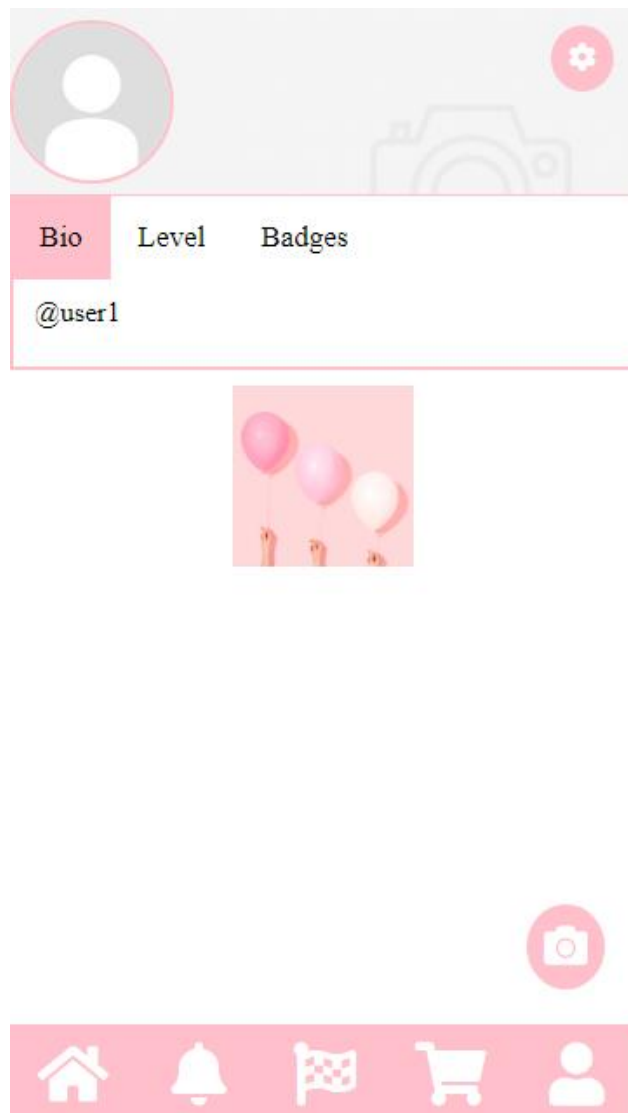


Фигура 4.6 Качване на снимка

След качване на снимка от потребител, тя ще се показва в неговият профил.



Фигура 4.7 Качване на снимка



Фигура 4.8 Профил със снимка

За да търси други потребители, даденият потребител трябва да кликне върху полето за търсене и да напише името на търсеният потребител.



Search...



Bu

BubbuleGum12

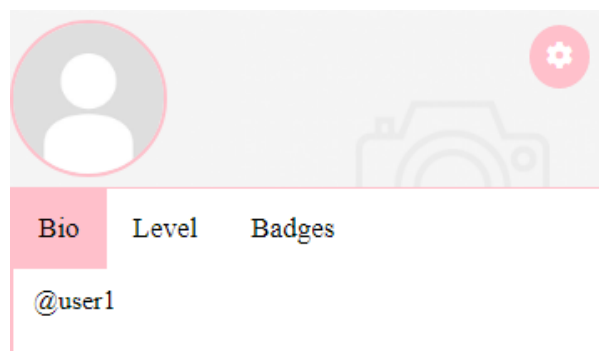


Фигура 4.9 Търсене



Фигура 4.10 Търсене на профил

За да отвори даден профил, потребителят трябва да натисне върху резултата от търсенето.

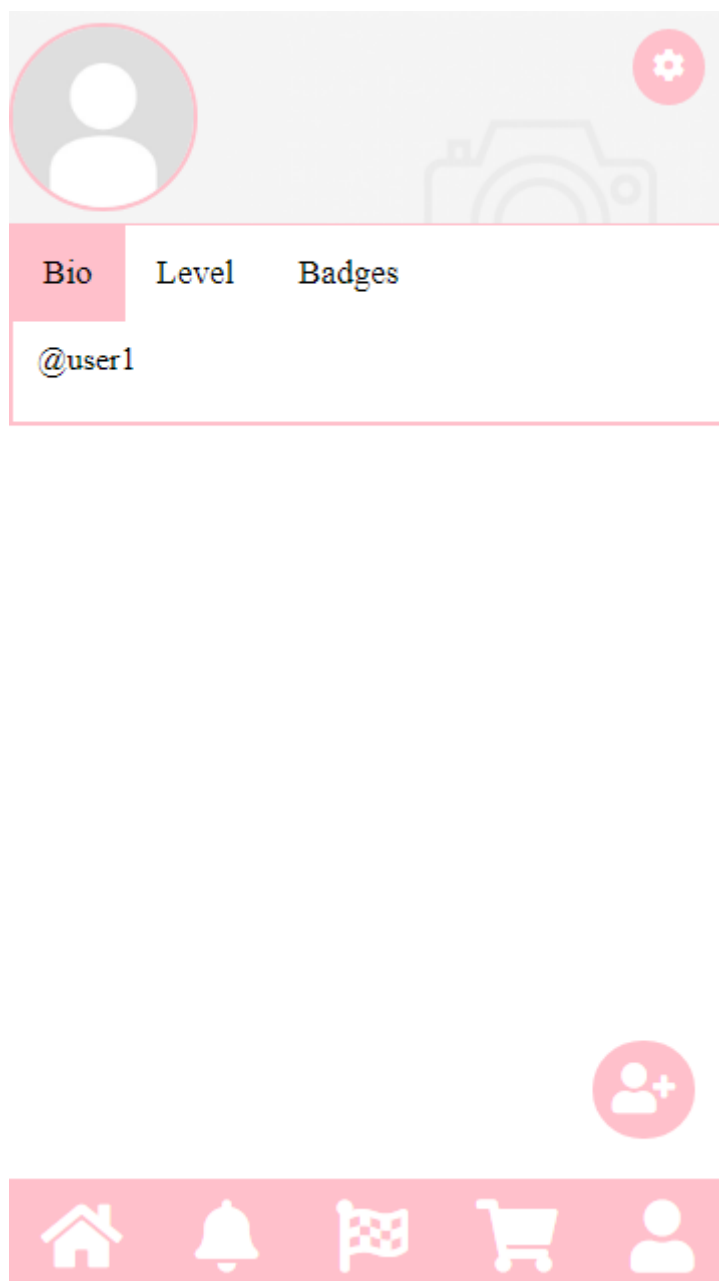


Фигура 4.11 Търсене на профил



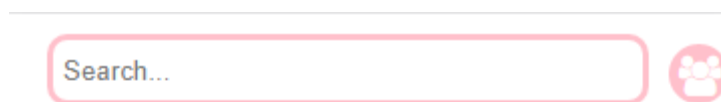
Фигура 4.12 Отваряне на потребителски профил

За да бъде последван даден потребител трябва да се натисне бутонът долу вдясно (с иконка на човек и „+“).



Фигура 4.13 Следване на потребител

За да види списък от хората които следва даден потребител и списък от хората които следват него, той трябва да натисне бутона вдясно от полето за търсене.



Фигура 4.14 Бутон за преглед на следванията



Фигура 4.15 Последователи



Фигура 4.16 Следвания

За да търси снимки, даденият потребител трябва да кликне върху полето за търсене и да напише тагът който търси, слагайки „#“ пред него.

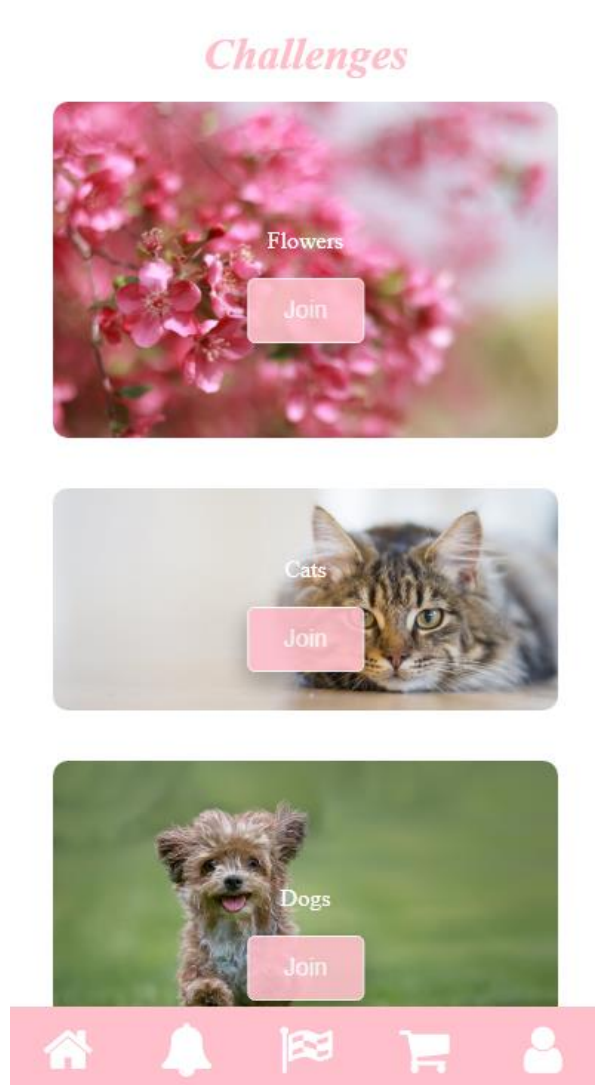


Фигура 4.17 Търсене

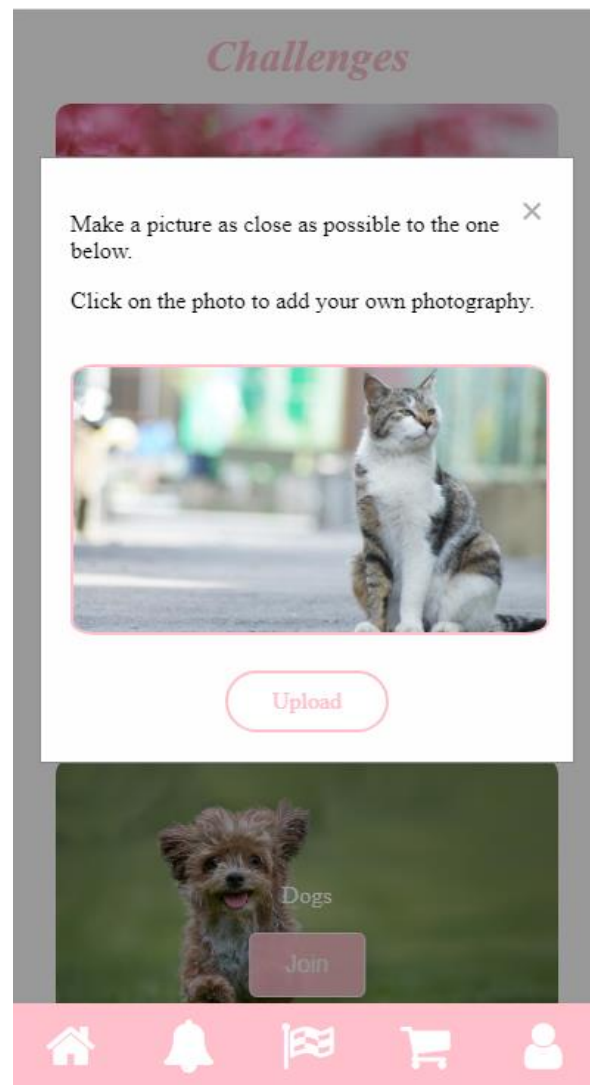


Фигура 4.18 Резултат от търсенето на снимки

За да вземе участие в турнир, потребителят трябва да кликне върху иконката със знаме, най-долу в средата на страницата. След това ще види актуалните турнири. За да участва в някой от тях, той трябва да кликне върху бутона с надпис *Join*, който се намира върху съответния турнир, а после да качи снимка към него.



Фигура 4.19 Турнири



Фигура 4.20 Качване на снимка към турнир

Заклучение

Всички задачи, поставени в заданието са изпълнени. Това не означава, че проектът е в напълно завършен вид.

Може да се работи още по стилизацията и функционалностите на проекта.

Разработено бе мобилно приложение, представляващо социална медия, в което всеки потребител може да си направи регистрация, да качва снимки към турнири или към собствения си профил, да търси снимки по даден таг, да търси други потребители по техните потребителски имена и да ги последва.

Бъдещото развитие би включвало чат система.

Библиография

- [1]Social medias - https://en.wikipedia.org/wiki/Social_media#History
- [2]Social-network games - https://en.wikipedia.org/wiki/Social-network_game
- [3]Technologies for mobile app development - <https://www.fingent.com/blog/top-technologies-used-to-develop-mobile-app>
- [4]MySQL GUI Tools - <https://codingsight.com/10-best-mysql-gui-tools/>
- [5]Convolutional neural network - https://en.wikipedia.org/wiki/Convolutional_neural_network
- [6]SIFT - https://en.wikipedia.org/wiki/Scale-invariant_feature_transform
- [7]SURF - https://en.wikipedia.org/wiki/Speeded_up_robust_features
- [8]Apache Cordova - https://en.wikipedia.org/wiki/Apache_Cordova
- [9]JavaServer pages - https://bg.wikipedia.org/wiki/Java_server_pages
- [10]MySQL - <https://www.datamation.com/storage/8-major-advantages-of-using-mysql.html>

Използвани съкращения

1. ConvNet - convolutional neural network - CNN - спираловидна невронна мрежа
2. GUI - graphical user interface - графичен потребителски интерфейс
3. SIFT - scale-invariant feature transform - мащабно-инвариантно преобразуване на характеристиките
4. SURF - speeded up robust features - ускорени стабилни функции

Сорс код + CD

Github хранилище: <https://github.com/vikir01/DR>

CD:

Съдържание

Увод.....	1
Глава 1.....	2
Въведение.....	2
1.1 Социалните медии.....	2
1.2 Социалните онлайн игри.....	3
1.3 Обзор на различните технологии за разработка на мобилни приложения.....	4
1.3.1 Swift.....	4
1.3.2 C++.....	4
1.3.3 Java.....	4
1.3.4 HTML5.....	4
1.3.5 PHP.....	4
1.3.6 PhoneGap.....	5
1.3.7 Appcelerator.....	5
1.3.8 RhoMobile.....	5
1.3.9 WidgetPad.....	5
1.3.10 MoSync.....	5
1.3.11 MySQL Workbench.....	6
1.3.12 PhpMyAdmin.....	6
1.3.13 HeidiSQL.....	7
1.3.14 Toad Edge.....	7
1.3.15 XAMPP.....	7
1.3.16 WampServer.....	7
1.4 Алгоритми за сходство между изображения.....	8
1.4.1 ConvNet.....	8
1.4.2 SIFT.....	9
1.4.3 SURF.....	10
1.5 Съществуващи подобни продукти.....	11
1.5.1 Pinterest.....	11
1.5.2 GuruShots.....	11
1.5.3 Facebook.....	11
Глава 2.....	12
Проектиране.....	12
2.1 Изисквания към програмия продукт.....	12
2.2 Софтуерни средства.....	13
2.2.1 Apache Cordova.....	13
2.2.1 MySQL.....	14

2.2.1.1 Сигурност....	14
2.2.1.2 Висока производителност.	14
2.2.1.3 Мащабируемост по поръчка.....	14
2.2.1.4 Пълен контрол на работния процес.....	15
2.2.1.5 Гъвкавост на отворения код.....	15
2.2.1.6 Намалени общи разходи за собственост.....	15
2.2.2 PhpMyAdmin.....	16
2.2.3 ХАМРР.....	16
Глава 3...	17
Реализация..	17
3.1 Регистрация на потребители.....	17
3.1.1 Схема на потребителите.....	17
3.1.2 Връзка с базата данни... ..	18
3.2 Качване и търсене на изображения.....	19
3.2.1 Информация за изображения.... ..	19
3.2.2 Информация за тагове към изображенията.....	19
3.2.3 Запазване на изображенията и таговете към тях.....	20
3.2.4 Връзка с базата данни.....	21
3.2.5 Визуализация на изображенията.....	22
3.2.6 Визуализация на профилната снимка.....	22
3.2.7 Търсене на изображения и потребителски имена.....	23
3.2.8 Визуализация на търсените изображения.....	24
3.3 Списък с последователи.....	25
3.3.1 Информация за последователите.....	25
3.3.2 Връзка с базата данни.....	25
3.4 Известия.... ..	26
3.4.1 Информация за известията.....	26
3.4.2 Връзка с базата данни.....	26
3.4.3 Визуализация на известията.....	27
3.5 Начална страница.....	27
3.5.1 Визуализация на снимките, качени от потребители.....	28
3.6 Турнири.....	28
3.6.1 Информация за турнирите.....	28
3.6.2 Връзка с базата данни.....	29
Глава 4.....	30
Ръководство за потребителя.....	30
4.1 Изисквания.....	30
4.2 Инсталация.....	30
4.3 Ръководство за потребителя.....	31