



Игра Hangman

Божидар Димитров и Васил Николов
81192, 81184

Проект по Мобилни приложения
Факултет по математика и информатика
Софийски университет „Св.Климент Охридски“, 2017г.

June 2017

Contents

1	Описание на проекта	3
2	Изисквания на проекта	3
3	Функционален обзор	3
3.1	Мобилно приложение	3
3.1.1	Main activity	3
3.1.2	Login activity	4
3.1.3	Register activity	4
3.1.4	Word handler	4
3.1.5	Lobby activity	4
3.1.6	Network handler	4
3.2	Сървър	4
4	Технологичен обзор	5
5	Бъдещо развитие на проекта	5

1 Описание на проекта

Проектът представлява просто мобилно приложение, което симулира играта на Обесване- познаване на дума със скрити букви като играчът има определен брой опити преди да бъде "обесен". Платформата, за която е разработено е Android.

Играта има прост потребителски интерфейс: в началото се отваря екран за избор между регистрация или логване. В зависимост потребителят трябва да въведе своите данни за да бъде верифициран/регистриран. След това потребителят бива пренасочен към самата игра: генерира се случайна дума със скрити букви, графично се изобразява бесило- репрезентация на броя оставащи опити, като на всеки несполучлив опит за познаване на буква се рисува нова част от тялото. Съответно ако играчът познае буква всичките нейни срещания се показват в думата. Играта свършва когато цялата дума бъде позната или се покажат всичките части на тялото на човечето на бесилото: опитите са свършили.

Проектът също така включва имплементиран сървър на Node.js, както и база данни на SQL, чиято основна употреба е да се грижи за логиката за регистрацията и верифицирането. При регистрация нов потребител бива добавен в базата данни, като неговата парола бива конкатенирана със сол и след това полученият низ- хеширан с SHA256. При верификация се проверява дали потребителят съществува в базата данни. Сървърът също така има endpoint за получаване на случайно генерирана дума, като в базата се съдържа таблица с възможни думи.

2 Изисквания на проекта

Android приложение, което позволява на потребител да се регистрира или верифицира, ако вече е преминал през стъпката на регистрация, след което да играе Hangman по правилата на играта. Графично изобразяване на играта. Имплементация на сървър и база данни за пазенето на входните данни на потребителите, както и съхраняването на възможните думи.

3 Функционален обзор

3.1 Мобилно приложение

3.1.1 Main activity

Имплементация на класа, който се грижи за началния изглед на приложението и предефинира задължителния onCreate() метод. Чрез loginButton и reg-Button се изобразяват бутоните съответно за верификация и регистриране. Към тях се закачат ClickListener за кликане, които препращат към форми за съответната функционалност.

3.1.2 Login activity

Грижи се за логин формата. Тя представлява две текстови полета съответно за потребителско име и парола и бутон със закаче за него ClickListener. Когато бутонът бива натиснат чрез NetworkHandler се прави HttpPost заявка към endpoint на сървъра, който очаква JSON обект съдържащ потребителското име и парола. Връща се AsyncTask, който изчаква отговор от сървъра дали верификацията е била успешна.

3.1.3 Register activity

Грижи се за регистрационната форма. Тя представлява две текстови полета съответно за потребителско име и парола и бутон със закаче за него ClickListener. Когато бутонът бива натиснат чрез NetworkHandler се прави HttpPost заявка към endpoint на сървъра, който очаква JSON обект съдържащ потребителското име и парола. Връща се AsyncTask, който изчаква отговор от сървъра дали регистрацията е била успешна.

3.1.4 Word handler

Чрез HttpURLConnection прави GET заявка до endpoint на сървъра, който връща случайно изтеглена от базата данни дума.

3.1.5 Lobby activity

Представлява самата игра. На екрана се изобразяват бесилото, скритите букви на думата и 26 бутона за цялата английска азбука, с които се играе играта. Чрез натискане на бутоните се прави опит да се познае буква, съдържаща се в думата. Ако опитът е успешен, т.е. в думата има такава буква, се изобразяват всички тези срещания на буквата. Ако не е успешен, се появява още една част от човечето на бесилото (броят на оставащи опити се намалява с един). Lobby activity също така се грижи за победа и загуба. Ако играчът не познае всички букви в рамките на позволените опити, той губи играта- появява се pop-up прозорец, гласящ: "You lose! The answer was:" думата, която е трябвало да познае. Ако победи се появява подобен прозорец със съобщение: "You win! The answer was:" думата, която е познал.

3.1.6 Network handler

Клас, който се грижи за POST заявките за регистрация и верификация. Използва HttpURLConnection клас, чрез който се изпращат JSON обекти с потребителско име и парола на потребителя.

3.2 Сървър

Сървърът е написан на Node.js, като използва express.js API за създаване на http server. Endpoints на сървъра биват:

- `/hangman/register` - очаква JSON обект с данните на потребителя, които да регистрира
- `/hangman/authorize` - също очаква JSON с данните, които проверява дали се съдържат в базата
- `/hangman/getWord` - взима случайно генерирана дума от базата данни и я връща като `response`

В базата данни има 2 таблици. Таблицата `Users` има колони: `Username` - съдържа потребителското име на играч, `PasswordSalt` - GUID, генериран за конкатенация с паролата, и `PasswordHash` - хешът на паролата с долепеният `PasswordSalt`. Когато се верифицира потребител от `request` тялото се взима неговата парола, долепя се солта, извлечена от базата данни, хешира се и се проверява дали съвпада с `PasswordHash`-ът. Така в базата не се съхраняват паролите на потребителите и това увеличава сигурността на данните. Използван е `SHA256` като алгоритъм за хеширане.

4 Технологичен обзор

Приложението е разработено за Андроид телефони с минимална `SDK Version 19` и `target SDK VERSION 24`. Разработвано е с `Android Studio 2.3`. Сървърът е имплементиран на `Node.js v6.11.0`, като използва `express.js`, `body-parser.js`, `edge.js`, `edge-sql.js` като зависимости, съответно за `http server`, парсър на тялото на `request` и връзка с `SQL` базата данни.

5 Бъдещо развитие на проекта

Предвидено е да се имплементира `multiplayer`: приложението да предоставя възможност играта да се играе от 2 играча, като единият ще определя дума, която другият да познае, редувайки се. Ще може всеки играч да се свързва към `lobby`, където да вижда онлайн играчи, както и създадени игри. На свой ред потребител ще може да създаде игра, в която друг потребител да влезе. Тази цел и мотивира създаването на сървър, който да се грижи за данните на потребителите.