

# **Software Requirements Specification**

for

## **Grabble**

Prepared by:  
Mantas Serapinas (1431261)

University of Edinburgh

2016 10 11

# Table of Contents

<b>Table of Contents</b>	<b>2</b>
<b>Introduction</b>	<b>3</b>
Description	3
Definitions	3
Operating Environment	3
References	3
<b>Bonus Features</b>	<b>4</b>
API and remote database	4
Authorization and Authentication	4
Show scoreboard	4
Achievements	4
<b>System Actions Requirements</b>	<b>6</b>
Show map	6
Collect letter	6
Collect word	6
Show scoreboard (bonus feature)	6
Authorization and Authentication (bonus feature)	6
Achievements (bonus feature)	7
<b>Non-functional Requirements</b>	<b>8</b>
Performance Requirements	8
Security Requirements	8

# Introduction

## Description

Grabble is an Android application which allows users to make words by collecting letters which are distributed around the University of Edinburgh's Central Area. The object of the game is to make seven-letter words out of the letters which have been collected.

## Definitions

- ❖ **API** - a back-end of the application developed using Ruby on Rails framework. Android application will communicate with API by JSON resources.
- ❖ **Dictionary** - a list of seven-letter words considered to be a word in Grabble application.
- ❖ **Letter Map** - the list of letters and their locations in the map. Each day of the week has its own Letter Map.
- ❖ **Table of Letter Points** - the point of value of each letter. (**Table 1**)

A	B	C	D	E	F	G	H	I	J	K	L	M
3	20	13	10	1	15	18	9	5	25	22	11	14

  

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
6	4	19	24	8	7	2	12	21	17	23	16	26

**Table 1**

## Operating Environment

Android 7.0 Nougat  
Nexus 6

## References

<http://www.inf.ed.ac.uk/teaching/courses/selp/coursework/grabble/selp-coursework.pdf>  
<http://www.inf.ed.ac.uk/teaching/courses/selp/coursework/grabble.txt>

# Bonus Features

## API and remote database

The application will communicate with Ruby on Rails API that consumes and produces JSON. The API and the remote PostgreSQL database is stored in Heroku Cloud. The database should contain two tables: users and achievements, which are connected by foreign keys. The API will provide such features as authorization, authentication, scoreboard calculation and achievements storing.

## Authorization and Authentication

Multiple users can play on one phone. User has credentials: email and password, using these he or she can login with his or her account. User has the following information:

- Email;
- Password;
- Nickname;
- Place;
- Points;
- Authentication token.

All user places are recalculated if any of the users collects a new word. User place depends on the points she or he has from collecting words.

To authenticate user requests, user will also have the authentication token. After the user logins, the API sets and sends the authentication token as a response and then every successive request has to add authentication token to the header of the request. If the authentication token set on the header does not match the one stored in the database with user info, the API will return unauthorized status code and the error message in the response. If the user log outs, the authentication token is reset.

## Show scoreboard

As user information is stored in remote database, it is possible to sort users by their points and show user scoreboard. The scoreboard is recalculated if any of the users collects a new word. The user would be able to see the changes on the scoreboard screen. The scoreboard doesn't show sensitive user information, such as email or password.

## Achievements

All users can unlock achievements by simply playing the game. User achievements has a separate screen, which as the user registers should contain a number of locked achievement icons. If the user does the action which unlocks the achievement, the corresponding achievement is added to user information. The achievement icon is then

being visible in the achievements screen. On the database, achievements are stored in a separate table with user id as the foreign key. Some examples of possible achievements:

- Find rarest letter ('Z');
- Collect a word with 20+ points;
- Get to the top 10 of the scoreboard;
- Collect 10 different letters in a day;
- Collect 10 words in a week;
- Collect the same word twice.

Hopefully, this will gamify Grabble application as users can compete not only by the points but also by achievements.

# System Actions Requirements

## 1. Show map

- 1.1. Should show Google Maps
- 1.2. Map should not be zoomable
- 1.3. Map should show all letter markers in a visible region
- 1.4. Map should show current user location
- 1.5. Map should delete the marker when user gets in proximity of the letter marker (REQ. 2.1) and confirms the picking of the letter.

## 2. Collect letter

- 2.1. User should be in the range of 25 meters from the letter to grab one.
- 2.2. User cannot grab a letter from the same spot more than once in a day.
- 2.3. When the application is started the **Letter Map** should show the letter map corresponding to the current day of the week.
- 2.4. The point value of each letter should correspond to the values in **Table of Letter Points**.
- 2.5. When user collects a letter, it should become visible in user portfolio.

## 3. Collect word

- 3.1. A word must be a seven-letter sequence of characters.
- 3.2. A word must appear in Official Grabble Dictionary.
- 3.3. When a user collects word, the sum of the letter points from the word are added to user points.
- 3.4. User can collect a word, when having the exact letters of the word in his or her portfolio.
- 3.5. User should be able to press a button 'Collect word' in order to collect the word.

## 4. Show scoreboard (bonus feature)

- 4.1. Should show user rating
- 4.2. Should show 9 other user ratings
- 4.3. Should not show users email

## 5. Authorization and Authentication (bonus feature)

- 5.1. After starting the application, the login screen is showed.
- 5.2. User must login with valid email and password.

- 5.3. If user doesn't have an account yet, he must register in order to use the application.
- 5.4. If the credentials are invalid, user should be notified.
- 5.5. If user log outs, the login screen is showed.
- 5.6. If user logged out and closed the application, next time she has to login before playing.

## 6. Achievements (bonus feature)

- 6.1. All users start with 0 achievements.
- 6.2. If the user does the action which unlocks the achievement, the corresponding achievement is added to user information.
- 6.3. User achievements are visible in user Achievements screen.
- 6.4. Achievements screen consists of number of achievement icons.
- 6.5. If achievement is locked, the default icon is shown.
- 6.6. If achievement is unlocked, the corresponding achievement icon is shown.

# Non-functional Requirements

## Performance Requirements

- The application should run smoothly
- The application should prioritize speed over the precision when trying to get user location.

## Security Requirements

- Unauthenticated user should not be able to start the game
- Unauthorized actions should not change any data
- Use principle of least privilege to all users