Index

[1 Introduction 3](#_Toc368482145)

[1.1 Purpose 3](#_Toc368482146)

[1.2 Stakeholders 3](#_Toc368482147)

[1.3 Definitions 3](#_Toc368482148)

[1.4 Overview of Document 3](#_Toc368482149)

[1.5 References 3](#_Toc368482150)

[2 Overall description 4](#_Toc368482151)

[2.1 Use cases 4](#_Toc368482152)

[2.1.1 0. Play Game 4](#_Toc368482153)

[2.1.2 1. Tutorial 4](#_Toc368482154)

[2.1.3 2. View Stats 5](#_Toc368482155)

[2.1.4 3. Replay Board 6](#_Toc368482156)

[2.1.5 4. Set up social function 7](#_Toc368482157)

[2.1.6 5. Add Friend 8](#_Toc368482158)

[2.1.7 6. Remove Friend 9](#_Toc368482159)

[2.1.8 7. View Dictionary 9](#_Toc368482160)

[2.1.9 8. Add and Edit Dictionary 10](#_Toc368482161)

[2.1.10 9. Remove Dictionary 11](#_Toc368482162)

[2.1.11 10. Play Friend's board 12](#_Toc368482163)

[2.2 Actor characteristics 12](#_Toc368482164)

[3 Specific Requirements 13](#_Toc368482165)

[3.1 Functional Requirements 13](#_Toc368482166)

[3.1.1 Use case 0. Play Game 13](#_Toc368482167)

[3.1.2 Use case 1. Tutorial 13](#_Toc368482168)

[3.1.3 Use case 2. View stats 13](#_Toc368482169)

[3.1.4 Use case 3. Replay board 13](#_Toc368482170)

[3.1.5 Use case 4. Set up social function 13](#_Toc368482171)

[3.1.6 Use case 5. Add friend 14](#_Toc368482172)

[3.1.7 Use case 6. Remove friend 14](#_Toc368482173)

[3.1.8 Use case 7. View dictionary 14](#_Toc368482174)

[3.1.9 Use case 8. Add and edit dictionary 14](#_Toc368482175)

[3.1.10 Use case 9. Remove Dictionary 14](#_Toc368482176)

[3.1.11 Use case 10. Play friends board 14](#_Toc368482177)

[3.2 Nonfunctional Requirements 15](#_Toc368482178)

[3.2.1 Use case 0. Play Game 15](#_Toc368482179)

[3.2.2 Use case 1. Tutorial 15](#_Toc368482180)

[3.2.3 Use case 2. Show stats 15](#_Toc368482181)

[3.2.4 Use case 3. Replay board 15](#_Toc368482182)

[3.2.5 Use case 4. Set up social function 15](#_Toc368482183)

[3.2.6 Use case 5. Add friend 15](#_Toc368482184)

[3.2.7 Use case 6. Remove friend 16](#_Toc368482185)

[3.2.8 Use case 7. View dictionary 16](#_Toc368482186)

[3.2.9 Use case 8. Add and edit dictionary 16](#_Toc368482187)

[3.2.10 Use case 9. Remove dictionary 16](#_Toc368482188)

[3.2.11 Use case 10. Play friends board 16](#_Toc368482189)

[3.3 General 16](#_Toc368482190)

# Introduction

## Purpose

The reason this document is written, is for the stakeholders and the developers to define an appropriate definition of the Android application “Wordfinder”. It should declare the application’s purpose and features, including the use cases and requirement specifications.

## Stakeholders

* User
* Client
* Developer

## Definitions

|  |  |
| --- | --- |
| Board / Matrix | Area where the player has to play in, the place where he has to find the words. |
| Dictionary |  |
| Friend | Another player who was added as “Friend” in the social section of the game. A friend allows to play other modi of the game |
| Highscore | Amount of all points, that a player collected in his playtime. |
| Language | Defines the language in which the game is played (language of the words). |
| Player | The person who uses the application. |
| Time, -r | A visual countdown that is used to determine the length of the game. |
| Word | Predefined chain of letters |
| Wordfinder | Name of the Android-application |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## Overview of Document

The next chapter, the Overall description, contains a diagram that present an overview and summary of all the use cases. Therefor the Overall description describes the application in the sight of an user, and simulates every action he should be able to perform.

The focus of the third chapter is primary written for the developers and describes in technical terms and the details of the functionalities of the application.

## References

None atm.

# Overall description

## Use cases

### 0. Play Game

Actors

* Player (Bob)

Description

* Bob wants to play a local game

Trigger

* Presses the 'Start Game' button in the main screen

Pre-Conditions

Post-Conditions

1. New entry in personal recordings (Score, Board, ...)
2. Push personal recordings to server, if possible

Main scenario

1. Bob starts the 'word finder' app
2. Bob presses on 'start game' in the main screen
3. The Language selection screen is displayed
4. Bob selects the language / dictionary in / with which he wants to play
5. The chosen language / dictionary is loaded
6. The System generates a new game
7. Bob plays and finished the game
8. Score is displayed and saved in the personal recordings
9. The system tries to synchronize the personal recordings with the server, if possible
10. returning to the main screen

Alternative scenarios

1. Game is interrupted (e.g. incoming call) or paused manually
2. Pause the game
3. Hide the board, show 'resume' button
4. Wait until 'resume' is pressed
5. resume the game and continue normal execution
6. Game is quit manually

* End game and return to main screen

Special Requirements

* For synchronization with the server, a standing internet connection is required.

Notes

* The game can only be used to it's full extend when an internet connection is present, but local games can be played without an internet connection.

### 1. Tutorial

Actors

* Player (Bob)

Description

* Bob wants to make the tutorial

Trigger

* Presses the 'Tutorial' button in the main screen

Pre-Conditions

-

Post-Conditions

-

Main scenario

1. Bob starts the 'word finder' app
2. Bob presses on 'Tutorial' in the main screen
3. A preprogrammed board is loaded
4. Bob works through the instructions of the Tutorial. He can't perform any other action than the one, the tutorial currently asks from him.
5. Bob finishes the Tutorial
6. The system returns to the start screen

Alternative scenarios

1. App is interrupted (e.g. incoming call)
   * No special action is taken, other than those to guaranty that the tutorial will be continued after the interrupt.

Special Requirements

-

Notes

-

### 2. View Stats

Actors

* Player (Bob)

Description

* Bob want's to view his or a friend's score.

Trigger

* Bob selects 'Scores' in the main screen

Pre-Conditions

* Bob has started the app

Post-Conditions

-

Main scenario

1. Bob opens the app
2. Bob selects 'Scores'
3. The app tries to synchronize with the server, if possible
4. Bob is presented with an option to view his ('personal') or the friend's ('friends') scores
5. Bob selects an option:

a) 'personal'

i) The personal scores are shown for the top 5 and the last 20 boards. (?)

ii) For boards that a / some friend(s) has / have played, the friend's score(s) is / are displayed too.

iii) a) Bob selects a board and is asked if he wants to replay it ---> continue in 3

b) Bob returns to the window presented in 2.6.2 ---> continue in 2.6.2

b) 'friends'

i) The friends scores are displayed in one single - segmented - list. (?)

ii) For each board, the other player's and the personal scores are displayed to, provided that the person in question has played the same board before.

iii a) Bob selects a board and is asked if he wants to play the board ---> continue in 10

b) Bob returns to the window presented in 2.6.2 ---> continue in 2.6.2

3 b) Bob returns to the main screen

Alternative scenarios

1. App is interrupted

No actions taken other than storing the current state.

Special Requirements

* For synchronization with the server, a standing internet connection is required.

Notes

1. The scores represent the ones known at the last synchronization
2. The details of the points marked with (?) are depending on the final implementation. The provided version is meant as a visualization and not as a precise description of the final product.

### 3. Replay Board

Actors

* Player (Bob)

Description

* Bob wants to play a local game he already played before, and that is visible in his scores.

Trigger

* Bob selects 'Replay' on one of his scored boards.

Pre-Conditions

* Bob is viewing his scores and has selected a board.

Post-Conditions

1. New entry in personal recordings (Score, Board, ...)
2. Push personal recordings to server, if possible

Main scenario

1. Bob selects 'replay'
2. The board is loaded
3. The board's dictionary is loaded
4. The game is generated. The loaded board is used instead of a newly generated one
5. Bob plays and finished the game
6. Score is displayed and saved in the personal recordings
7. The system tries to synchronize the personal recordings with the server, if possible
8. returning to the main screen

Alternative scenarios

1. Game is interrupted (e.g. incoming call) or paused manually
2. Pause the game
3. Hide the board, show 'resume' button
4. Wait until 'resume' is pressed
5. resume the game and continue normal execution
6. Game is quit manually
7. End game and return to main screen

Special Requirements

* For synchronization with the server, a standing internet connection is required.

Notes

* The game can only be used to it's full extend when an internet connection is present, but local games can be played without an internet connection.

### 4. Set up social function

Actors

* Player (Bob)

Description

* Bob wants to share his boards with his friends and play friends boards
* Trigger
* Bob starts App

Pre-Conditions

* Bob hasn't set up the sharing functionality yet.

Post-Conditions

* Bobs boards are visible to others, and he can add friends to play their boards

Main scenario

1) Bob starts the 'word finder' app

2) Bob presses 'friends' in the main screen

3) Bob is asked to enter his email address and a password for creating an account.

4) Bob now has the social feature set up.

Alternative scenarios

1. The Device is not connected to the internet
   * When clicking on friends, Bob gets a message which states that Bob has to be connected to the internet to interact with his friends

Special Requirements

* Phone has to be connected to the internet

Notes

* When the social function is set up, anybody with Bobs email address can follow his plays

### 5. Add Friend

Actors

* Player (Bob), Friend (Susie)

Description

* Bob wants to add Susie to his friend list.

Trigger

* Bob starts App

Pre-Conditions

* Bob has the sharing function set up.

Post-Conditions

* New friend added to the friends list

Main scenario

1. Bob starts the 'word finder' app
2. Bob presses on 'friends' in the main screen
3. Bob presses follow a friend
4. Bob now enters Susie's email address.
5. Bob is now following Susie's game history

Alternative scenarios

1. The Device is not connected to the internet
   * When clicking on friends, Bob gets a message which states that Bob has to be connected to the internet to interact with his friends
2. Susie does not play Wordfinder
   * After entering the email address, a "friend not found" message is shown
   * Bob can enter a new email address and try again

Special Requirements

* Phone has to be connected to the internet

Notes

### 6. Remove Friend

Actors

* Player (Bob), Friend (Susie)

Description

* Bob wants to remove Susie from his friend-list.

Trigger

* Bob starts App

Pre-Conditions

* Bob has the sharing function set up.
* Bob has Susie in his friend-list.

Post-Conditions

* Susie is removed from the friends list.

Main scenario

1. Bob starts the 'word finder' app
2. Bob presses 'friends' in the main screen
3. Bob presses 'show' friends
4. Bob now sees a list of his friends and sees an edit button in the to right corner.
5. Bob presses the edit button
6. Bob can now select a friend he wants to delete (Susie in our case).
7. Bob confirms the deletion
8. Bob no longer follows Susie's games.

Alternative scenarios

1. The Device is not connected to the internet
   * When clicking on friends, Bob gets a message which states that Bob has to be connected to the internet to interact with his friends

Special Requirements

* Phone has to be connected to the internet

Notes

-

### 7. View Dictionary

Actors

* Player (Bob)

Description

* Bob wants to view the German dictionary

Trigger

* Bob starts App

Pre-Conditions

-

Post-Conditions

-

Main scenario

1. Bob starts the 'word finder' app
2. Bob presses languages
3. Bob chooses an existing language (German in our case)
4. Bob can now scroll through all the words in this language or enter a search term to filter all the entries for it.

Alternative scenarios

-

Special Requirements

-

Notes

-

### 8. Add and Edit Dictionary

Actors

* Player (Bob)

Description

* Bob wants to add a new dictionary and add words to it.

Trigger

* Bob starts App

Pre-Conditions

-

Post-Conditions

* A new dictionary with some words is created and can be used for a game.

8.6 Main scenario

1) Bob starts the 'word finder' app

2) Bob presses languages

3) Bob presses 'new language' and enters a name.

4) Bob presses this new language's name

5) Bob presses add word and enters the new word. A new word is created.

6) Bob presses 'import' to import words from another language

7) Bob chooses the language.

8) Bob can choose multiple words from this language to import.

9) Bob confirms and gets back to the dictionary overview

10) Bob presses done. Now he can use his new dictionary to play a new game.

Alternative scenarios

1. Bob doesn't define any words for his dictionary
   1. The dictionary cant’ be used to play games until Bob adds words to it.

Special Requirements

-

Notes

-

### 9. Remove Dictionary

Actors

- Player (Bob)

Description

- Bob wants to remove a dictionary

Trigger

- Bob starts App

Pre-Conditions

-

Post-Conditions

- A dictionary has been removed

Main scenario

1. Bob starts the 'word finder' app
2. Bob presses languages
3. Bob presses the 'edit' button
4. Bob selects the dictionary to delete and confirms
5. The dictionary is deleted and bob gets back to the languages menu

Alternative scenarios

-

Special Requirements

-

Notes

-

### 10. Play Friend's board

Actors

- Player (Bob)

Description

- Bob wants to play a game a friend has played before, and that is visible the friend's scores.

Trigger

- Bob selects 'Play' on one of his friend's boards.

Pre-Conditions

- Bob is viewing his friend's scores

Post-Conditions

1. New entry in personal recordings (Score, Board, ...)
2. Push personal recordings to server, if possible

Main scenario

1. Bob selects ‘Play’
2. The board is loaded
3. The board's dictionary is loaded if stored locally
   1. if the dictionary is not stored, Bob is asked to download it, which then is done for him.
4. The dictionary is displayed to Bob
5. The game is generated. The loaded board is used instead of a newly generated one
6. Bob plays and finished the game
7. Score is displayed and saved in the personal recordings
8. system tries to synchronize the personal recordings with the server, if possible
9. to the main screen

Alternative scenarios

1. Game is interrupted (e.g. incoming call) or paused manually
   1. Pause the game
   2. Hide the board, show ‘resume’ button
   3. Wait until ‘resume’ is pressed
   4. resume the game and continue normal execution
2. Game is quit manually
   1. End game and return to main screen

## Actor characteristics

Main actor of our use cases in Bob.

Bob is a informatics student at the university of Bern.

So Bob is experienced using software, and acquiring new skills.

Bob has owned Android phones since the first Android came out.

Bob so has a lot of experience in using Android software.

Second actor is Susie.

Susie has already installed and used our app.

# Specific Requirements

## Functional Requirements

### Use case 0. Play Game

* Makes playing Word Finder game.
* Home-screen has a start game button to start game.
* By selecting the language other standard dictionaries are excluded from game, until next chose.
* After every game user has possibility to change language of dictionaries in a menu.
* User makes his profile at first opening, user database is created.
* Play game initializes a new 7x7 word game grid, sets time and score to initial value. (minimal word-count of 10 words and minimal word-length of 3 letters are assured).
* Initial value of score is 0.
* Initial value of time is 0s.
* Internet connection for synchronization.
* Pause button stops the game (time halts) and shows pause screen.
* Pause screen shows a resume button.
* At end of game time and score is showed to player, and achievement of new hight score is shown to player.

### Use case 1. Tutorial

* Tutorial is shown to user.
* Tutorial button available in main screen.
* Existance of a tutorial board.
* Abbort tutorial button.

### Use case 2. View stats

* Shows own accieved scores.
* Shows friends best five scores.
* Score button in main screen.
* Diffrent scores selectable in score menue: friend and own.

### Use case 3. Replay board

* Play another time on a known board.
* Replay button beside the scores of played boards.
* A replay don’t overwrites the last game it adds a new entry to the last 20 games.

### Use case 4. Set up social function

* Set up sharing functionality means uploading profile to server.
* Server is needed for distribution and maintenance of user data.
* Friend button available in home screen.
* Login window : 1 Username , 2 Password.
* App can establish a internet connection to a determined server.
* Capability of sending and receiving data to/from internet.
* Capability of local storage of snips (5-10) of friend list consisting in last loaded view of it.

### Use case 5. Add friend

* App allows to add friends to friend list.
* Pressing on friend in no- friend list shows add as friend button.
* Pressing on add friend button send friend request to other user.
* Accepting friend request from other user or other user accepts own request causes the friend to be added in friend list.
* Identify user according to selected e-mail addresses.
* Existence of a list of friends and a list of all users on server side.

### Use case 6. Remove friend

* Edit-Friends button available under the friend menu.
* Delete-Friends button is shown when Edit-Friends button is pressed, by selecting it, game asks if really wanted to perform friend deletion if yes removes it from friend list.
* Deleted friend doesn`t becomes a notification when being deleted.
* Deleted friend appears in normal person list of the server.

### Use case 7. View dictionary

* App includes dictionaries of provided languages (here only German).
* Possibility to select own word of provided dictionary for use in own dictionary.
* Possibility to delete own word out of own dictionary.
* Create a new dictionary with own name.

### Use case 8. Add and edit dictionary

* Creates a new file containing the new custom dictionary.
* Dictionaries are saved with a new name called the language of the dictionary.
* It’s possible to importing words from other dictionaries.
* Dictionary upload to server.

### Use case 9. Remove Dictionary

* Removes dictionary from the game and the phone.

### Use case 10. Play friends board

* Play game a friend already played.
* Play button on score list of friends.
* Dictionary load if not on phone

## Nonfunctional Requirements

### Use case 0. Play Game

* <Performance requirement> : <System start (home screen) under 10s.>
* <Portability> : <System portability has to be specified before installation and download of the App.>
* <Space requirement> : <Installation space have to be specified before download and installation. (kbyte)>
* <Space requirement> : <Minimal RAM space during use has to be specified before installation. And checked for availability before usage.>
* <Space requirement> : <Space used for offline recorded scores/times/Board/ is specified and checked before playing game.>
* <Performance requirement> : <System has to stop immediately when pushing pause, the screen has to be covered in an instant.>
* <Legislative requirement> : <Copyright and terms of use has to be shown to the User before first launch of the App (during download or installation).>
* <Legislative requirement> : <User has possibility to modify or correct his own user record database.>
* <Efficiency requirement> : <Internet connection is checked when opening the game. If no connection is established a offline game will be started and game history or statistics are saved in a external file for recovery when the connection is reestablished. Assures no continuous connection searching, and performance lost by this.>
* <Performance requirement> <Opening a new game should not take more than 10s.>

### Use case 1. Tutorial

* <Variability> : <Requirement of different languages or only symbols or animation>.

### Use case 2. Show stats

### Use case 3. Replay board

* <Space requirement> : <text file for the generation numbers of the last 20 boards and the top 5.>

### Use case 4. Set up social function

* <Legislative requirement> : <User agrees terms of social networking of Word Finder Game.>
* <Security requirement> : <Using of secure script for password and user data transfers to server.>
* <Privacy requirement> : <Data is only accessible to friends and the general permission to show data to others has been accepted from the user.>
* <Product requirement> : <The phone has possibility to connect to internet: 4G, 3G,(â€¦ internet standards of mobile phones) and Wi-Fi.>
* <Space requirement> : <Space allocation for user database on server side and for offline usage on cellphone.>
* <Portability requirement> : <check if the board is playable on other device should by installing conditions.>

### Use case 5. Add friend

* <Space requirement> : <Allocation of space for the list of friends and the list of all users on the server side ( Database ).>
* <Fault tolerance> : <A sent of a wrong friendship request cant be aborted, friendship has to be deleted once it was accepted.>
* <Privacy requirement> : <A friend can see public things.>
* <Privacy requirement> : <A other user is only a friend if friendship was accepted not yet if only a request is sent.>

### Use case 6. Remove friend

* <Legislative requirement> : <deletion of friendship don't requires double con sense.>
* <Fault tolerance> : <A sent of a wrong friendship deletion request can’t be aborted, friendship has to be recreated once it was deleted.>
* <Accessibility> : <The data of non- friends can’t be accessed.>

### Use case 7. View dictionary

* <Customizability> : <Own dictionary can be added to the game.>
* <Variability> : <Own dictionaries can variate, from user to user but standard dictionaries don't.>
* <Correctness> : <Dictionary has to contain only correct spelled words.>
* <Space requirement> : <Space for installation of dictionary databases needed.>

### Use case 8. Add and edit dictionary

* <redundancy> : <imported word can’t be imported if they already exists in the dictionary, duplicates are ignored and no failure is reported.>
* <privacy requirement> : <custom dictionaries can only bee seen from friends in friend list.>
* <Space requirement> : <Server side space for custom dictionary (database with views).>

### Use case 9. Remove dictionary

* <robustness> : <All game boards with the old dictionary are deleted and are no longer playable.>
* <robustness> : statistics referring to boards with deleted dictionaries are deleted.>
* <Failure tolerance> : <User has to confirm 2 times before deletion.>
* <Failure tolerance> : <Updates of deleted dictionaries take places after 24h. During this a player is capable to restore changes.>

### Use case 10. Play friends board

## General

* <Backup> : <No backup is done fore the game in execution, excepted the stats sent to the server.>
* <Backup> : <Github for version control an backups.>
* <Interoperability> : <Word Finder Game is designed to work alone, and for synchronization with a server.>
* <Price> : <Set by "Customer".>
* <Copyright> : <"Customer".>
* <Open Source> : <Code is not open source.>
* <Priority> : <For all interactions with friends a internet connection has to be established, in order to synchronies the action. If not a, action is saved in a file for later synchronization.>
* <Fault tolerance> : <Every action modifying the friends list has to be confirmed at least once.>
* <Response time> : <average time to respond non during gaming is 10s-15s during gaming 1s for selecting max 5s for verification.>
* <Delivery requirement> : <Product delivery according to scheduled time by the course ESE HS 2013.>
* <Ethical requirement> : <No specification about a word filter.>
* <Testability> : <All requirements has to be tested.>
* <Supportability requirement> : <Group 4 ESE HS 2013 is available for support questions and issues.>
* <Disaster recovery> : <System shuts down and no data is saved.>
* <Documentation> : <User manual or short tutorial is available.>
* <Extensibility> : < - >.
* <Usability> : <Specification comes with first prototype.>
* <Accessibility> : <App launched on Swiss stores for android apps, available to all without restrictions.>

Annotation: Tried not to write the same requirements more than one time, some points ar missing requirements but would be covered by redundancy of the other scenarios.