

System Requirement Specifications (SRS) for Drawdiculous

Bian Hengwei	U1923732B
He Yinan	U1922693C
Jin Han	U1922733A
Lee Yu Jie Melvin	U1922533G
Renganathan Ramasamy	U1922494L
Sannabhadti Shipra Deepak	U1822459L

Submitted to—
School of Computer Science and Engineering, Nanyang Technological University

1 Table of Contents

2 Problem Statement	4
3 Overview	4
3.1 Background	4
3.2 Overall Description	4
4 Investigation & Analysis Methodology	4
4.1 System Investigation	4
4.2 Analysis Methodology	5
4.2.1 Feasibility study and requirements elicitation	5
4.2.2 System analysis and requirements specification	5
4.2.3 Object-oriented design using UML	6
4.2.4 Prototyping	6
5 Constraints	7
5.1 Scalability	7
5.2 Platform	7
5.3 Project Schedule	7
6 Operational Requirements	7
6.1 System User Support	7
6.2 Application Services and Technical Support	7
6.3 System Hardware and Data	7
6.4 Audit Trail	8
7 Functional Requirements	8
7.1 Registration	8
7.2 Login	8
7.3 Edit Profile	8
7.4 Forget Password	8
7.5 Play Game	9
7.6 Join Game Room	9
7.7 Create Game Room	9
7.8 Remove Player	9
8 Input Requirements	10
8.1 Room Type	10
8.2 Room ID	10
8.3 Language	10
8.4 Game Input	10
9 Process Requirements	10

9.1 DB transaction	10
9.2 Data Confidentiality	10
9.3 Data Integrity	11
9.4 Data Availability	11
10 Output Requirements	11
10.1 Game summary and Results	11
10.2 Leaderboard Display	11
10.3 Correct guess output	11
11 Hardware Requirements	11
11.1 Client Devices	11
11.2 Developer Devices	11
11.3 System for Team to communicate	11
12. Software Requirements	12
12.1 Client Operating Systems	12
12.2 Client Applications	12
12.3 Network System	12
12.4 Mainframe Architecture	12
12.5 Licenses	12
13. Deployment Requirements	13

2 Problem Statement

As people age, people are more likely to suffer from age-related diseases such as dementia. Currently, there are around 50 million people worldwide suffering from dementia. Due to the aging population, this number is expected to increase to 82 million by 2030. Dementia typically affects the elderly over 65 years old, however, research shows that dementia is now on the rise among younger people. In Singapore, the National Neuroscience Institute diagnosed four times as many people under 65 with dementia in 2017 than 2011. Research shows that cognitive training can remarkably slow down the cognitive decline associated with aging.

3 Overview

3.1 Background

As technology develops, more and more elderly are owning and using mobile phones. Mobile games are the most accessible games now as it requires nothing other than a functioning mobile phone. There exist many games that target people with dementia, but these games often involve in person interaction and are not playable if players do not meet physically. These games are less applicable now with Covid-19 pandemic that discourages face to face interaction. There also exist many mobile games that are generally friendly to elderly, however, the majority of these games target children, which makes the design less appealing to elderly.

Considering the current situation of aging population and Covid-19 pandemic, a mobile game that is accessible, elderly friendly, and stimulates thinking is much needed to alleviate the problem the world is facing now.

3.2 Overall Description

The mobile game will be developed using Android Studio for the frontend and Java for the backend. The mobile game targets elderly by providing elderly friendly user interface and stimulates thinking while being interesting.

The mobile game will allow players to form groups and play draw and guess games. One user in the group will be provided with a prompt to draw, other players will attempt to guess the word that the drawer is trying to draw. Multiple languages will be supported in the game to encourage elderly across races to interact.

4 Investigation & Analysis Methodology

4.1 System Investigation

Drawdiculous pushes any local changes, including account changes, game room status, drawing status, and guessing status to the server. Server listening to port 3002 will receive all

updates from the user and store the updates in a MySQL database. Some crucial data, such as game room structures and drawing status, will be processed in an active object in the program. Upon user requests, information is pulled by the users and the UI is subsequently updated. Specifically, all data transmissions will be in a json format to help simplify the program.

4.2 Analysis Methodology

4.2.1 Feasibility study and requirements elicitation

Surveys on potential users about opinions on Draw and Guess type games will be conducted. Small scale testing of the working protocol on these users will be done before the application is released. Regular meetings with sponsors and all team members will be held regularly at a fortnightly frequency. Small meetings with most team members present will be held while needed to discuss project schedule, requirement details and other questions.

4.2.2 System analysis and requirements specification

4.2.2.1 Perform an analysis of the problem using object-oriented techniques

An external view of the product including game rooms, g drawing games, activities such as guessing answers and checking leaderboard will be developed using Unified Modeling Language (UML). This System Requirement Specifications document is a part of the second round deliverable of the project. Our system will contain some features including:

- Create and manage an account
- Login to an account
- Create / join a game room
- Draw on the canvas
- View others' drawing real-time and make guesses
- Check real-time leaderboard

4.2.2.2 Scope and Limitations

Analysis will include business analysis, functional and nonfunctional requirement analysis, architecture analysis, process analysis and code analysis.

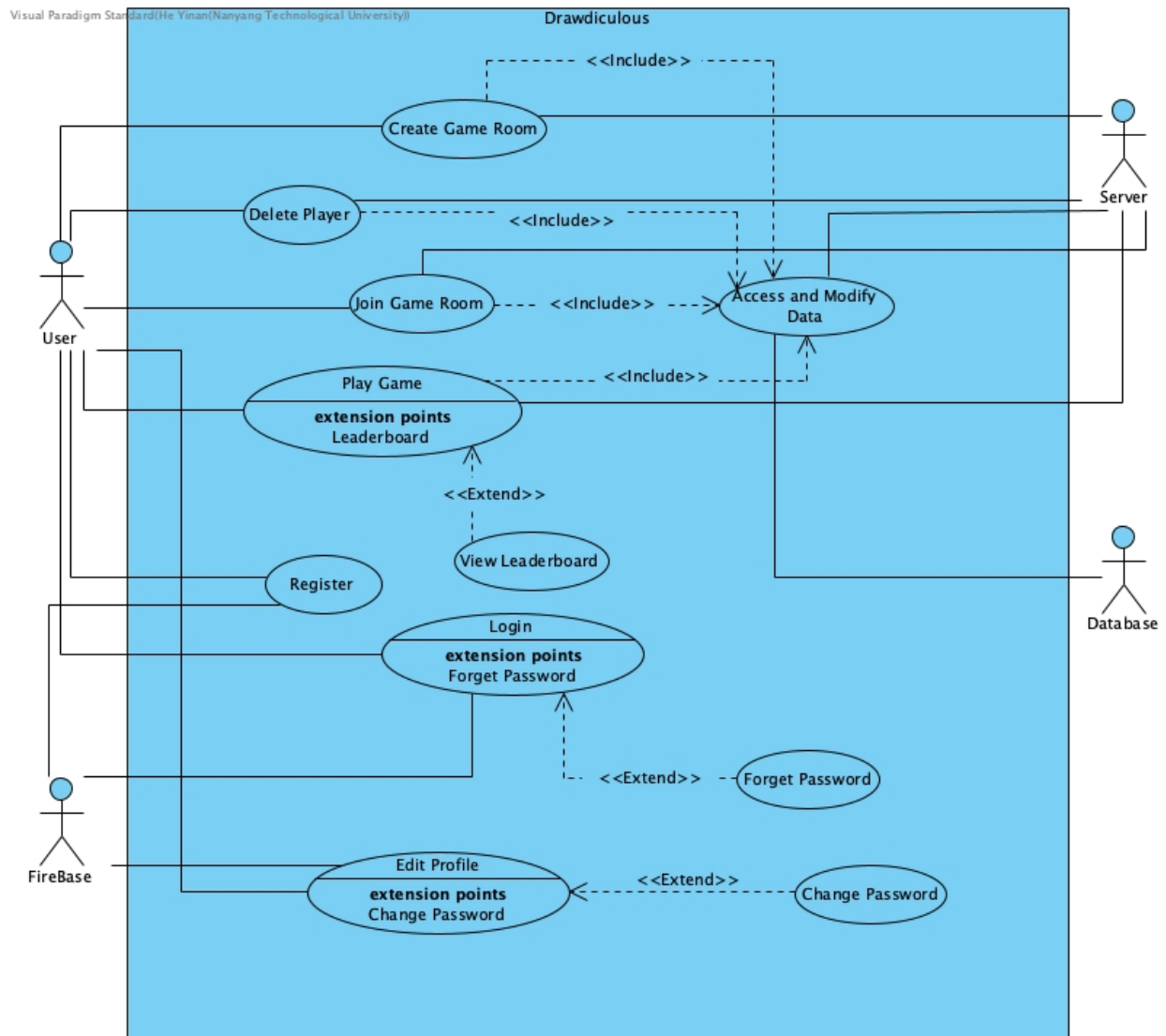
- Business analysis: business rules, business partners, sponsorship, budget management
- Requirement analysis: functional and nonfunctional requirements, sponsors' requirements
- Architecture analysis: app layered architecture, server implementation
- Process analysis: data flow analysis, client-server connection analysis
- Code analysis: code style, code length

There is an unavoidable time limitation on our project. Though analysis is done with care, the time limitation decides that we are not able to extend more.

4.2.3 Object-oriented design using UML

An android application designed based on OOP principles and MVC pattern will be developed. The design will be detailedly documented using UML diagrams.

4.2.3.1 Use Case



4.2.4 Prototyping

We will be using the Object Oriented Rapid Prototyping (OORP) method to implement a basic working prototype of the product. The prototype will be used for demonstration purposes to our sponsors. The prototype will include a basic UI of the application and simplified communication protocols between the client and the server.

5 Constraints

5.1 Scalability

According to our current design, too many connections to the server would overload the server and result in either connection error or very bad response speed. Our product is not scalable due to the server constraint. In order to make the product more scalable, a distributed game server system would be needed instead of a single server. The server program needs to be improved as well.

5.2 Platform

We are developing the application using Android Studio and the product would only be available on the Android platform. An Android device is required to run the application. To make the application more portable, we need to rewrite the program in Swift / C# etc.

5.3 Project Schedule

We have a three-month timeframe to implement the Drawdiculous application. This project should begin from the Lab 1 session of CZ3002 and finish by Lab 5.

6 Operational Requirements

6.1 System User Support

Users must have access to the “Help” section of the application. This section includes a brief overview of how to play the game, with instructions for how to create a game room, how to key in your answer, et cetera. There will also be a list of Frequently Asked Questions (FAQ) to address and provide answers for common questions or concerns.

In case the user is unable to find the solution to their problem in the FAQ, the “Help” section will also include a telephone and email address where users can call or email for further support. Users will have 24x7 access to email and telephone assistance for any technical questions that may arise, such as trouble resetting their password, application errors and so on.

6.2 Application Services and Technical Support

Programmers and application developers will have access to source code to address bugs or system enhancements as deemed necessary. Any bugs that are discovered will be recorded and a ticket will be created for the bug fix.

Network Administrator and DBA support is also required to maintain a 24x7 system uptime.

6.3 System Hardware and Data

A computer operations center will be created. It will handle system hardware tasks such as carrying out scheduled back-ups to prevent data loss as well as system patches when a bug is discovered and fixed. Every month, hardware maintenance and scheduled system maintenance must be carried out to ensure quality remains up to standard. A failover must also be set up in anticipation of system shut-downs or servicings.

6.4 Audit Trail

System audit trails are an inherent part of Drawdiculous. Among others, all transaction records will capture what action was taken, when (time-stamp) the transaction occurred and who made the transaction. The audit trail must also be verified on a monthly basis.

7 Functional Requirements

Drawdiculous is a real-time, multiplayer mobile game that caters to the social and cognitive needs of the elderly.

7.1 Registration

7.1.1 New user must be able to create a new account using the following information:

7.1.1.1 Username

7.1.1.2 Email address

7.1.1.3 Password

7.1.1.4 Confirm password

7.1.2 System must check that all the fields are valid

7.1.3 System must send an email that contains a verification link to the user. Account is verified when the user clicks on the link.

7.2 Login

7.2.1 User must enter their email address and password

7.2.2 System must check that the credentials are valid

7.2.3 System must log the user into their account and redirect them to the main page

7.3 Edit Profile

7.3.1 User must be able to modify their email address

7.3.1.1 System must send an email that contains a verification link to the new email address

7.3.1.2 Upon verification, system must update the new email address to database

7.3.2 User must be able to modify their username

7.3.2.1 Upon confirmation, system must update the new username to database

7.3.3 User must be able to modify their password

7.3.3.1 User must key in both new password and confirm password

7.3.3.2 Upon verification, system must update the new password to database

7.3.3.3 System must log user out of account and prompt them to login again

7.4 Forget Password

7.4.1 User must enter their email address

7.4.2 Firebase must check validity of email

- 7.4.3 Firebase must send a link to change the password to the email
- 7.4.4 User must enter new password
- 7.4.5 System must update new password to database

7.5 Play Game

- 7.5.1 System must assign first player as drawer
 - 7.5.1.1 System must give user three prompts
 - 7.5.1.2 User must choose one of the prompts and draw its picture
 - 7.5.1.3 System must send the drawing in realtime to the guessers
- 7.5.2 System must assign all other players as guessers
 - 7.5.2.1 User must guess the drawer's picture and enter their answer
- 7.5.3 System must be able to assign points to each player
 - 7.5.3.1 For drawers, system must be able to assign points based on how many guessers got the correct answer
 - 7.5.3.2 For guessers, system must be able to assign points based on how fast they entered the correct answer
- 7.5.4 System must assign next player in line as drawer, and others as guessers when each round ends
 - 7.5.4.1 System must assign in a sequential manner until all players have been drawer once
 - 7.5.4.2 System must end the game and declare the winner when all players have finished being the drawer
- 7.5.5 System must be able to rank each player according to their cumulative points
 - 7.5.5.1 User must be able to view this ranking by clicking on "View Leaderboard"
 - 7.5.5.2 System must be able to declare the player with the most points as winner

7.6 Join Game Room

- 7.6.1 User must choose "Join Room"
- 7.6.2 User must choose "Random Room", "Public Room" or "Private Room"
 - 7.6.2.1 For public room, user must enter the room id of the room they wish to join
 - 7.6.2.1.1 For private room, user must enter the password as well
 - 7.6.2.2 System must validate the room id (and password, if applicable)
- 7.6.3 System must redirect user into their chosen game room

7.7 Create Game Room

- 7.7.1 User must choose "Create Room"
- 7.7.2 User must choose "Private Room" or "Public Room"
- 7.7.3 System must generate unique room id
 - 7.7.3.1 If private room is chosen, system must generate unique password as well
- 7.7.4 System must redirect user into their chosen game room

7.8 Remove Player

- 7.8.1 User must click on "Manage Room"
 - 7.8.1.1 User must be host player of the game room

- 7.8.2 User must click on the name of the player they wish to remove and then “X”
- 7.8.3 System must prompt user for confirmation
 - 7.8.3.1 User must choose “Yes” to confirm removal
- 7.8.4 System must remove the player from the game room

8 Input Requirements

8.1 Room Type

Players should be able to create a room mode (i.e. public or private). The system will generate a room ID upon creations.

8.2 Room ID

Players who choose to join an existing room should know the ID of the room and key in the corresponding ID to play with friends. For those private rooms, only guests of the creator should be allowed to enter the room.

8.3 Language

New and existing players should be able to choose the respective language of the choice, and the interface should automatically reflect such a change of menu.

8.4 Game Input

During the games, players will play in 2 different roles. For drawers, the players will need to draw the given words. The system should be able to recognise, save and transmit that information. The input bar should be disabled.

During the guessing phrase, all non-drawers should be able to guess the drawing through the input bar. However, the drawing interface should be disabled at all times for non-drawers.

9 Process Requirements

9.1 DB transaction

The system must be able to send, receive and trigger transactions to the Database system.

9.2 Data Confidentiality

To protect personal data or information, all data transmitted to and from the database should be encrypted. Only authenticated users could change their own personally setting preference in the respective database (i.e. B cannot change A preferences database)

9.3 Data Integrity

Data error from the user's end to the data-based end must be gracefully handled. Timeout transactions such as during login should be rolled back to prevent cheating (such as when playing). During processing, the data should not be changed to another value which may alter the result of the information.

9.4 Data Availability

The database must be available on a 24x7 basis. The performance should be there to prevent any in game lagging when playing.

10 Output Requirements

10.1 Game summary and Results

After each Drawdiculous guess is finished, there must be an increase in points for the specific player(will be displayed in the 'rankings' section). The database will be able to display all the accumulated points of the players.

10.2 Leaderboard Display

The game must be able to extract all the points data to display the points table of the players in the leaderboard section of the game, after each game ends.

10.3 Correct guess output

After each correct guess is made, the system must check with the actual word in the database and display the correct word in the screen (if the correct word is guessed already).

11 Hardware Requirements

11.1 Client Devices

Android Mobile Phones which can support at least Android 10

11.2 Developer Devices

Mac or Windows laptops running Android Studio

11.3 System for Team to communicate

Computer systems with Internet access to connect through a cloud server such as Google Drive.

12. Software Requirements

12.1 Client Operating Systems

1. Android OS 10

12.2 Client Applications

1. Android OS 10

12.3 Network System

1. Network software and protocols for system communication: TCP/IP, HTTP, HTTPS, SSH, etc

12.4 Mainframe Architecture

1. Windows machine running MySQL server version 14.14 & IIS v7.5.

12.5 Licenses

Valid licenses are required to run software from third-party vendors:

1. To use application development tools
2. To use application server and database software in development, testing and production

13. Deployment Requirements

