



INDIANA UNIVERSITY

SCHOOL OF INFORMATICS

Bloomington

I494 CAPSTONE

TEAM#71

Full Requirements Documentation

Team Name: 666

Team member:

Shiqi Shen

Jinchuan Zhanghe

Hongze Wu

Qianyue Ma

Zhicheng Zhu



Table of Contents

1. Introduction.....	3
1.1 Purpose	3
1.2 Document Conventions	3
1.3 Project Vision and Product Scope	3
1.3.1 Vision Statement	3
1.3.2 Product Scope.....	3
1.4 References	3
2. Overall Description.....	3
2.1 User Classes and Their Use Cases	3
2.1.1 Visitor.....	3
2.1.2 Player.....	4
2.1.3 Submitter	4
2.1.4 Administrator.....	4
2.2 Operating Environment	4
2.3 Design and Implementation Constraints	4
2.4 User Documentation.....	5
2.5 Assumptions and Dependencies	5
3. Functional Requirements	5
3.1 Use Cases for Visitors	5
UC-V-1: Get Information about WeePlay	5
UC-V-2: Get Information about the Author	6
UC-V-3: Read the Post Article.....	6
3.2 Use Cases for Player.....	6
UC-P-1: Register for an account	6
UC-P-2: Fill the personal profile	7
UC-P-3: Get information about the game.....	8
UC-P-4: Match players with the same interests	8
UC-P-5: Create group.....	8
UC-P-6: View profile of players	9
UC-P-7: Add players as friends and chat with players.....	9
UC-P-8: Unregister for an account.....	10
3.3 Use Cases for Submitters	10
UC-S-1: Submit a Game Article/Video	10
UC-S-2: Withdraw or Modify a Post Articles/Video.....	12
3.4 Use Cases for Administrators	13
UC-A-1: Modify Site Contents.....	13
UC-A-2: Manage Post Article Entries.....	13
UC-A-3: Manage Feature List.....	14
UC-A-4: Manage Players' data (database manager)	14
4. External Interface Requirements	15
4.1 User Interfaces.....	15
4.2 Software Interfaces	15
4.3 Communications Interfaces	16
5. Other Nonfunctional Requirements	16
5.1 Performance Requirements	16
5.2 Security.....	16
5.3 Extensibility.....	16
6. Business Rules	16
Appendi: App Flow. Low-fidelity mockups (Screens)	17

1. Introduction

1.1 Purpose

This document describes the requirements for the Weeplay app, which provides supporting information for helping to build this app, clarify the responsibilities for each role.

1.2 Document Conventions

The functional requirements for this project are organized by use case within user class.

1.3 Project Vision and Product Scope

1.3.1 Vision Statement

For people who love PC games and intend to find and play with other players, Weeplay is an app that will provide a platform to help players meet and communicate with others, and share their questions, moments on the forum and allow other players to answer and comment. Unlike game center nor steam, Weeplay will contain all PC online games from different companies, for example, Riot, Blizzard, Steam & etc. Weeplay will also keep updating new PC games when they are published.

1.3.2 Product Scope

Weeplay a phone app on Android platform. It will provide a platform for players to find and communicate with others who share same interest with them, provide a platform for visitors to view the topics in forums for different games.

1.4 References

Existing App used as models for ideas regarding the functionality documented here include:

<http://www.xiaoheihe.cn/>

<https://play.google.com/store/apps/details?id=com.tencent.mm&hl=en>

2. Overall Description

2.1 User Classes and Their Use Cases

2.1.1 Visitor

A Visitor is someone who downloads the WeePlay but still not registers an account on our application. they might want to know what the application's usage and attracted by the content published in our forum. Use cases:

- UC-V-1: Get Information about *WeePlay*
- UC-V-2: Get Information about the Author
- UC-V-3: Read the Post Article

2.1.2 Player

A Player is someone who downloads the WeePlay and already registers an account on our application. They can access most of the features of the application and they can easily contact with other users. Use cases:

- UC-P-1: Register for an account
- UC-P-2: Fill the personal profile
- UC-P-3: Get information about the game
- UC-P-4: Match players with the same interests
- UC-P-5: Create group
- UC-P-6: View profile of players
- UC-P-7: Add players as friends and chat with players
- UC-P-8: Unregister for an account

2.1.3 Submitter

A Submitter is some Player who submits an article to the Platform and has full access permission to the content which they posted. Use cases:

- UC-S-1: Submit a Game Article/Video
- UC-S-2: Withdraw or Modify a Post Articles/Video

2.1.4 Administrator

An Administrator is responsible for setting up, updating, modifying, and keep the daily operation of functional modules, they have full access permission for most features of our application and have the right to ban player to access some modules.

Use cases:

- UC-A-1: Modify Site Contents
- UC-A-2: Manage Post Article Entries
- UC-A-3: Manage feature List
- UC-A-4: Manage Players' data (database manager)

2.2 Operating Environment

WeePlay shall be compatible with various versions of Android system on different smartphone platforms.

2.3 Design and Implementation Constraints

CO-1: The Administrator should have basic database knowledge, highly recommended Mysql, and any SQL related languages.

CO-2: Player shall be able to maintain and modify the content they posted without requiring any specialized technical knowledge.

CO-3: The register page should have enough guide information to help user complete the registration.

2.4 User Documentation

The developer shall provide necessary documentation to enable an Administrator to perform the use cases listed in section 2.1.4 Administrator.

2.5 Assumptions and Dependencies

There are no known assumptions or dependencies for this app.

3. Functional Requirements

3.1 Use Cases for Visitors

UC-V-1: Get Information about *Weeplay*

Description: A Visitor shall be able to get information about the *Weeplay in the main game menu*.

Main.game.info:	The home page of the App shall contain a button that goes to a page that provides the popular main games menu. If a Visitor clicks on the game button, visitor can go back to the home page directly.	(Priority=H)
Main.game.select:	The display of main games menu shall contain a button link that invites the Visitor to see more information about any game. If a Visitor clicks on each selected game button, then see Main.game.view	(Priority=H)
Main.game.view:	The App shall provide the Visitor entered into the game community with the selected game button, then the visitor can see the post list about any share story of games which is published by players.If a visitor clicks on each selected post from the list, then see Main.game.viewdetails	(Priority=H)

Main.game.viewdetails: A Visitor shall be able to enter the details view page when each game story post has been selected. Then a visitor will see the content of the post article, the author of the post and the date. (Priority=H)

Main.game.back: If a Visitor requests to back to the previous page to select another interested game, the back image button is available. (Priority=M)

UC-V-2: Get Information about the Author

Description: A Visitor shall be able to get information about the author of *Post Articles about games from the App*.

author.info: The game view details page shall provide a link to a page with biographical information about the author. (Priority=H)

author.contact: The page with author information shall contain a link to the author contact form (see UC-V-4). (Priority=H)

UC-V-3: Read the Post Article

Description: A Visitor shall be able to view the Post Article on the game view page.

post.article.list The home page shall provide a link to the game post articles list. The article list view page provides visitor to choose the interested topic. (Priority=H)

post.article.view: The list view page shall provide a link to the post details view. A Visitor shall be able to read any current contents of the posted game articles. (Priority=H)

3.2 Use Cases for Players

UC-P-1: Registration for an account

Description: A player shall be able to register an account of his/her in order to get the full experience of Weeplay.

Player.registration.information:	Weeplay already provided a registration page for each person who wants to join in. The user has to fill out the registration form. A verified personal Email is required while registering by while the user change the password, unregister the account or deal with other problems about security. Weeplay would lead the user directly to profile page after finishing registering.	(Priority=H)
player.registration.edition:	After finish registering, a homepage would be provided for each one of the user in which he/she could edit their personal information and choose what types of PC game he/she always engages. The user could also skip filling out the profile form, which means the user could edit this area anytime	(Priority=H)
player.registration.error:	If a user enters an invalid type of the information in the given place such as typing an Email in the place a First name is needed, Weeplay shall display an error message, showing that the type is wrong. Also, an error would also be displayed when the username has already be existed.	(Priority=H)
Player.registeration.page:	When profile editing is finished or skipped, Weeplay leads user to view their favorite type of the game (or the one with a particular name which the user might have already written in the profile page). If the user has not filled out the profile yet, the game page would display the popular recent PC games	(Priority=H)

UC-P-2: Fill the personal profile

Description: The app shall enable a Player to fill in his/her personal profile after registration.

player.profile.fill: If a player already register the account, player (Priority=H)
will be send to the page to fill the profile, Either
player finishes filling the profile or skips this
step. Player will be sent to
player.registration.page.

player.profile.edit: Player is able to edit their profile at any time (Priority=H)
after they finish filling the profile.

UC-P-3: Get information about the game

Description: Player login their account and enter to the main page for the game
community. They will get the same use case with the visitors (3.1). Also, when player
read the post articles, they can give any comments.

player.comment: Users are allowed to comment each article posted. (Priority=M)
By touching the title of the article, users may
view the whole article. Users' comments could be
left at the end of the page.

UC-P-4: Match players with the same interests

Description: Player may match up with players who share same interests (they are
interested in same games) after player finishes filling in the profile. App will use
information in database to find players have same interests, and similar ages. After
matching, player can view players' profiles, chat with players, and add players as
friends.

UC-P-5: Create group

Description: Player shall be able to create groups with other players, players can chat
with each other in the group.

player.group.create: Player shall use create group function to create (Priority=H)
a group. Player can name the group when
group is created, player can also edit the name
after group is created.

player.group.invite: Player can invite their friends to the group they created, the friends player invites can also invite their friends to the group. (Priority=H)

player.group.kick: Player can choose the players in the group to kick, but only the founder of group has the right to kick players. (Priority=H)

player.group.disband: Player who creates the group can also disband the group, by the time, players will no longer be able to invite others to the group and send messages in the group, the group will disappear from chat list. (Priority=H)

UC-P-6: View profile of players

Description: Player can view other players' profiles at any moment, the games other players are interested in.

profile.view.author: Player shall be able to view the author's profile through the UC-V-2 post article page. (Priority=H)

profile.view.friends: Player shall be able to view the player's profile through the UC-P-6 the recommendation (match up) page and the friends contact list page. . (Priority=H)

UC-P-7: Add players as friends and chat with players

Description: Player can add players as friends in recommendation (match up) and the players in search results

player.friend.add: Player can add friends by clicking 'add friend' button, then a friend request will send to other players, and wait other players to accept the friend request. A friend can also be found by searching the username. (Priority=H)

player.friend.chat: Player shall be able to choose the friends they want to chat with. They need to find the friends in his/her friend list and touch “chat” button to start chatting. (Priority=H)

player.friend.delete: Player can delete the friends from his/her friend list, after deleting the friend, player can no longer chat with that person unless they become friends again, which means he/she needs to resend add-friend request to that person and wait for acceptance. (Priority=H)

UC-P-8: Registration for an account

Description: All players are able to unregister their accounts, after they unregister their account, they will not be able to get full experience in Weeplay, and their email addresses will be available to be used to register new accounts when they feel regretted and decided to come back. At the same time when they unregister their accounts, all information will be erased from database.

3.3 Use Cases for Submitters

UC-S-1: Submit a Game Article/Video

Description: A Submitter shall be able to submit a game article/video from the App.

article/video.submit: The Application shall provide a platform for Submitters to upload articles or videos mainly about the strategies for PC games. Each article/video submission shall be composed of: (Priority=H)

- Submitter's name (It is the player's register name)
- Statement of the article/video topic (required; 30 words maximum)
- Category of the Article/video (optional; select from game types, article/video type, Other)
- Description of the article/video category (required only if the Submitter selected "Other" for the Category)
- Content of the posted article/video (required; 250 words minimum, 1500 words maximum/30mins maximum for each video)

article/video.fields.requirement: The post article/video submission form app visually indicate which fields are required. (Priority=H)

article/video.valid.info:	If any of the required fields in the article/video submission are missing, the App shall display specific error messages that identify all of the missing inputs.	(Priority=H)
article/video.word_count	For fields that have minimum and/or maximum required words lengths, the app shall display messages that show how many more words are required, how many more words are permitted, or how many words must be removed to create acceptable field entries.	(Priority=H)
article/video.agree:	A Submitter shall be able to submit a article/video only if he has checked a checkbox on the post submission form that indicates the Submitter has read and agrees to the submission terms.	(Priority=H)
article/video.acknowledge:	When an article/video has been successfully submitted, the website shall display an acknowledgment to the Submitter. If an error was encountered during submission, the App shall display an explanatory message.	(Priority=H)

UC-S-2: Withdraw or Modify a Post Articles/Video

Description: A Submitter shall be able to withdraw a post article that he previously submitted.

article/video.withdraw:	The App shall provide a link on the article/video submission description page to allow a Submitter to request that a pearl of wisdom he previously submitted be withdrawn.	(Priority=M)
article/video.modify:	The App shall provide a link on the article/video submission description page to allow a Submitter to make modifications in the previously submitted.	(Priority=M)

article/video.list: A Submitter shall be able to view a list of all of the articles/video he has submitted. This list shall display the post topic, date submitted, and statement of the article/video (dependent upon UC-S-1). (Priority=L)

3.4 Use Cases for Administrators

UC-A-1: Modify Site Contents

Description: An Administrator shall be able to modify the contents of any page in the app.

page.modify: An Administrator should be able to modify the contents of any page in the app using the Android Studio or app editing tools. (Priority=H)

page.upload: An Administrator should be able to upload new pages and files directly to Weeplay App from his development PC. (Priority=H)

UC-A-2: Manage Post Article Entries

Description: An Administrator shall be able to create new post article entries, modify existing post article entries, delete article entries or comments to post article entries, and archive a set of post article entries.

post.article.create: An Administrator shall be able to create post article entries. (Priority=H)

post.article.initial: Before the Weeplay App goes live, the Administrator should create at least one post article entry. (Priority=H)

post.article.modify: An Administrator should be able to modify an existing post article entry. (Priority=H)

post.article.delete: An Administrator should be able to delete an existing post article entry. (Priority=H)

post.article.reply: An Administrator should be able to post a reply to any comment entered on a post article entry. (Priority=H)

post.article.archive: An Administrator shall be able to archive a set of post article entries, and make that archive available in the Weeplay App. (Priority=H)

UC-A-3: Manage feature List

Description: An Administrator shall be able to create new features for the APP and modify or delete existing features.

feature.create: An Administrator should be able to create a new feature for the Weeplay user's community. (Priority=H)
Attributes of a feature are a name, description, image, and function.

feature.modify: An Administrator should be able to modify any of the attributes of an existing feature in the App through editing either in the Android Studio or an editing tool that supports the App. (Priority=H)

UC-A-4: Manage Players' data (database manager)

Description: An Administrator shall be able to create database store the Player's personal information.

Database.Create: An Administrator should be able to create a Database to store Player's Information. (Priority=H)

Database.Modify: An Administrator should be able to modify the Database, such as create new column select Player's information. Administrator has full access permission for every database they create before. (Priority=H)

Database.add: An Administrator shall be able to add new entries to the Database. (Priority=M)

Database.delete: An Administrator shall be able to delete specific entries from the Database. (Priority=M)

Auto-matching system.Create:	An Administrator should be able to Create the Auto-matching, Which can use the information entered by User and matching up. According to different Player's interest, helping them find the player with similar interests.	(Priority=H)
------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------

4. External Interface Requirements

4.1 User Interfaces

UI-1: The fields used in all forms shall be wide enough to accommodate 95% of expected entries without requiring either horizontal or vertical scrolling, except for content field, where vertical scroll bars shall become enabled if the user enters enough information to force a line break.

UI-2: For fields of known maximum length, such as the 10 characters of postal code (ZIP+4, plus hyphen), the fields shall not be wider than necessary to contain the expected entries.

UI-3: Each subpage in the Application shall have a return bar with links to the previous pages. If the currently displayed page appears in the menu bar, its link shall be shown in a different font color than the other menu options.

UI-4: The login page should hash the password which entered by Users.

UI-5: The register page should have enough guide information to help user complete the registration.

UI-6: When processing of a script is completed, the Application shall directly go to next page or pop up a descriptive error message if an error occurred during script processing.

4.2 Software Interfaces

External links will exist from Weeplay App to the following sites:

4.2.1 steam.com

4.2.2 spotify.com

Links are expected to come into Weeplay App from the following sites:

4.2.3 facebook.com

4.2.4 amazon.com

4.2.5 ebay.com

4.3 Communications Interfaces

CO-1: Weeplay creates dialogue boxes for communication among players which are the chats between players, the chats in group and the forum.(UC-P-7) Voice message, images or short videos could also be sent to players while chatting in the dialogue box.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

PE-1: The app must be run on Android system, and requires internet connection.

5.2 Security

SE-1: The app will request authentication from players, fail to authenticate will not be able to login, every time account is logged in a new phone will require email verification.

5.3 Extensibility

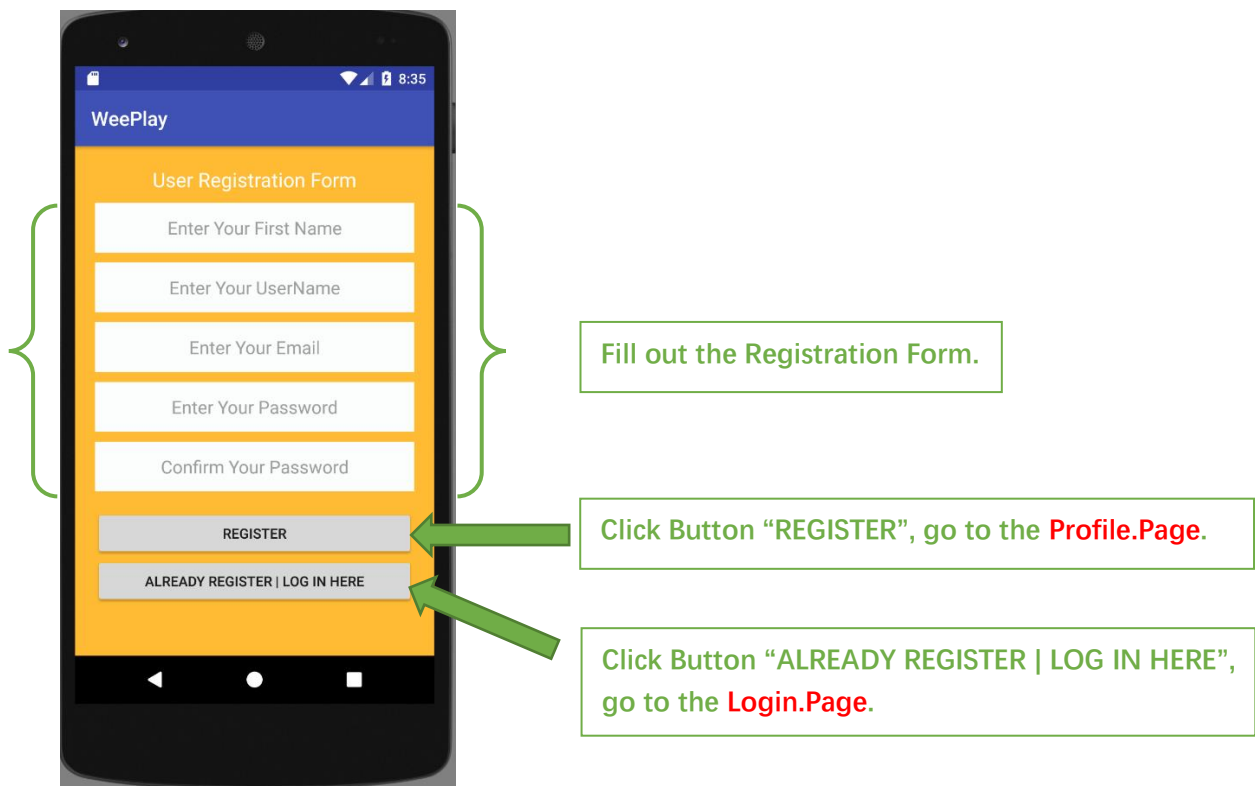
EX-1: The app shall be designed to be compatible on IOS platform.

6. Business Rules

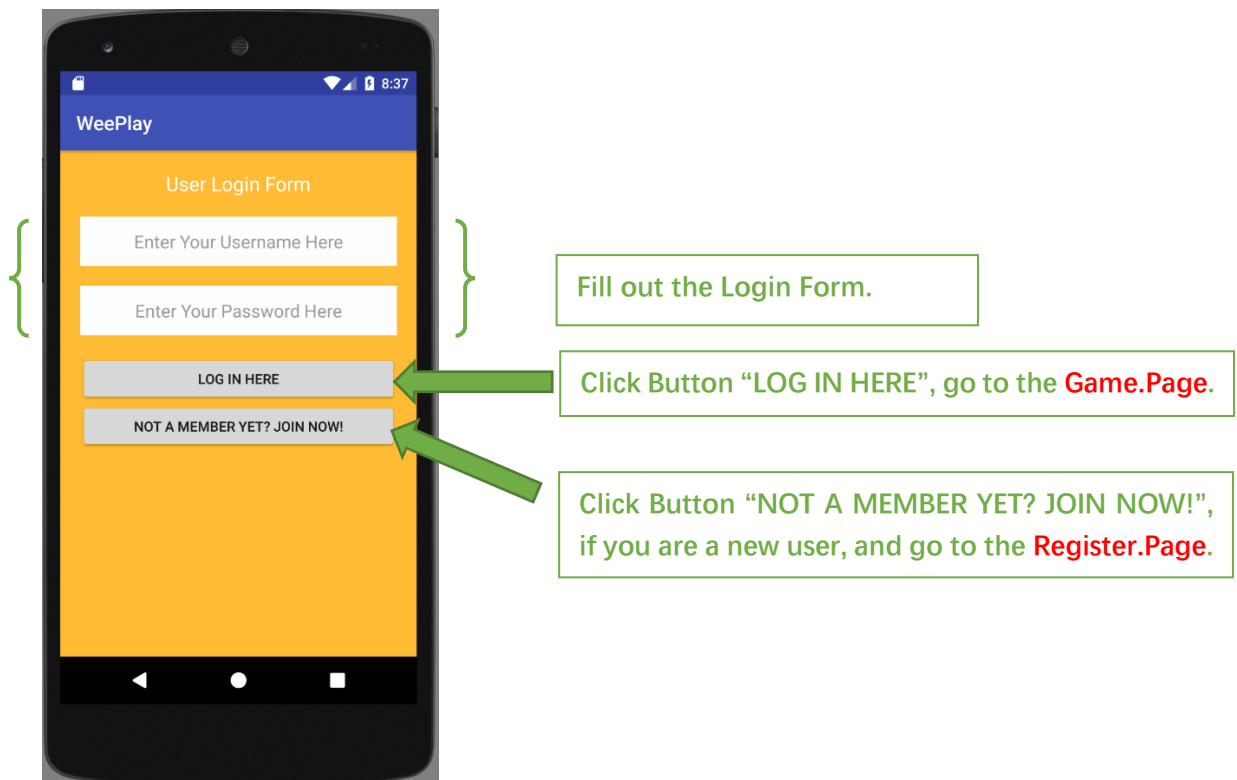
BR-1: The flow economics which means when we have a huge number of players in Weeplay, we can promote advertisements into our app. This is our final goal in revenue, because we make our App as a platform for many different gamer and will be related with different demand for e-sports hardware and software, includes keyboard, mouse, etc.

Appendix: App Flow. Low-fidelity mockups

Register.Page



Login.Page



Profile.Page

WeePlay

Last Name Middle Name

First name

Age

Address

City State Zip

Enter the game you play

Enter the game you play (Optional)

Enter the game you play (Optional)

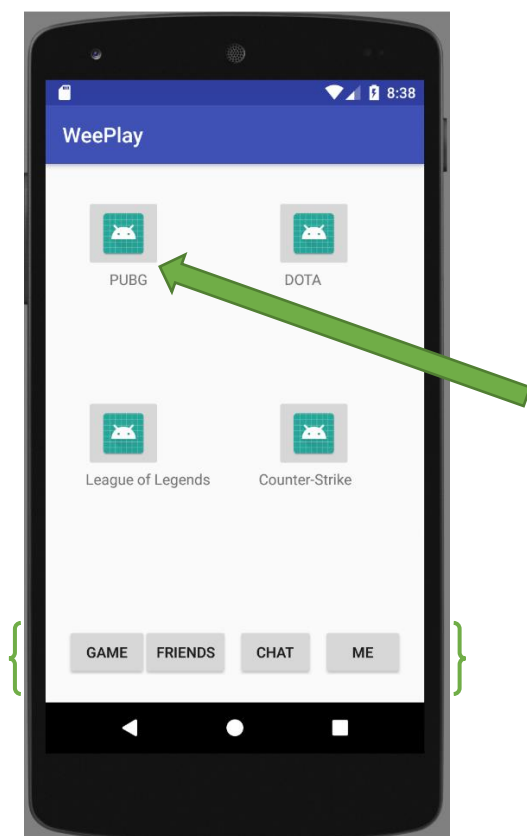
Type of game you play

START SKIP

Fill out the Profile Form.

Click Button “START”, if you fill out the profile form and ready to use Weeplay. Also, you can click button “Skip”, if you want fill out the profile later. Go to the **Game.Page**.

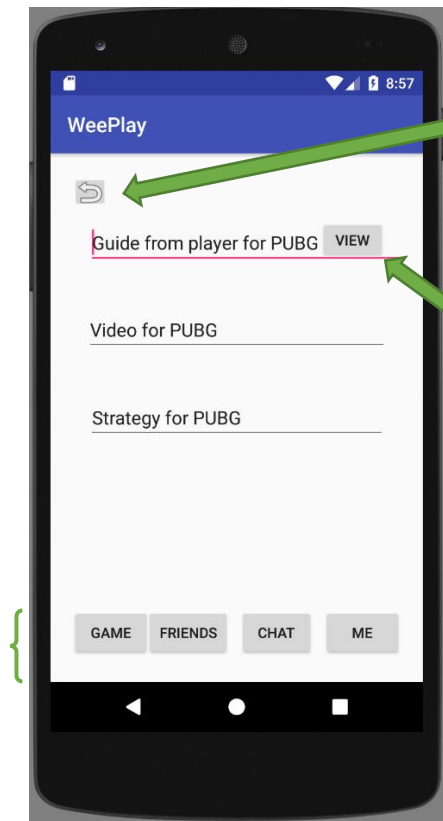
Game.Page



Each image button for different games, the game display is depending on user's prefer game which have written in user's profile. If user didn't fill out the profile form, the Game.Page will display the popular game recently. If Click Button for game “PUBG”, go to the **SelectGame.Page**.

The four main buttons connect to four main pages. Click each of them, it will take user to each different main page. Now in **Game.Page**.

SelectGame.Page

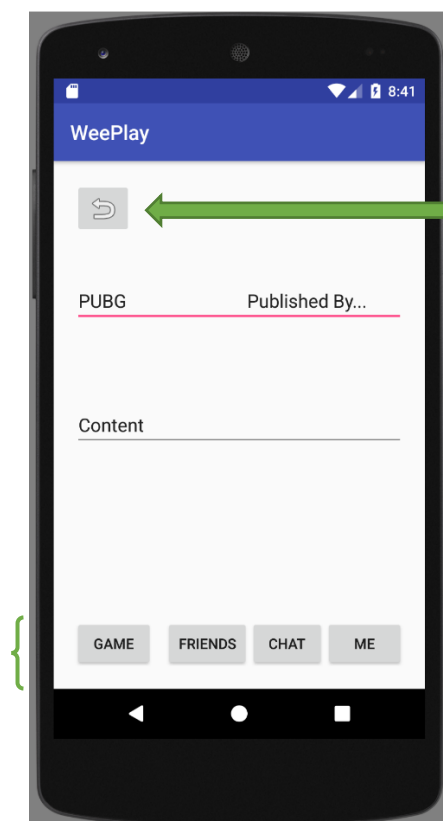


Click Image Button, go back to the **Game.Page**.

This page displays the forum and latest news of the game you selected in **Game.Page**. Click Button "VIEW", then go to the **GameContent.Page** to see the details of each post.

The four main buttons connect to four main pages. Click each of them, it will take user to each different main page. Now if click Button "GAME", it will take user to **Game.Page**.

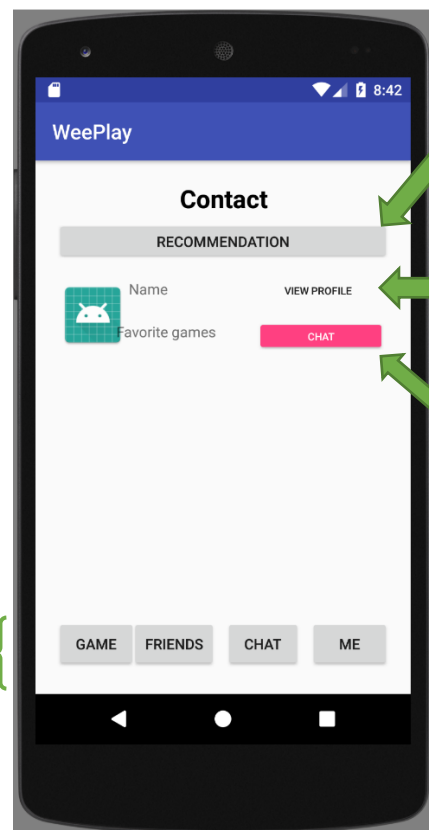
GameContent.Page



Click Image Button, go back to the **SelectGame.Page**.

The four main buttons connect to four main pages. Click each of them, it will take user to each different main page. Now if click Button "GAME", it will take user to **Game.Page**.

Friends.Page



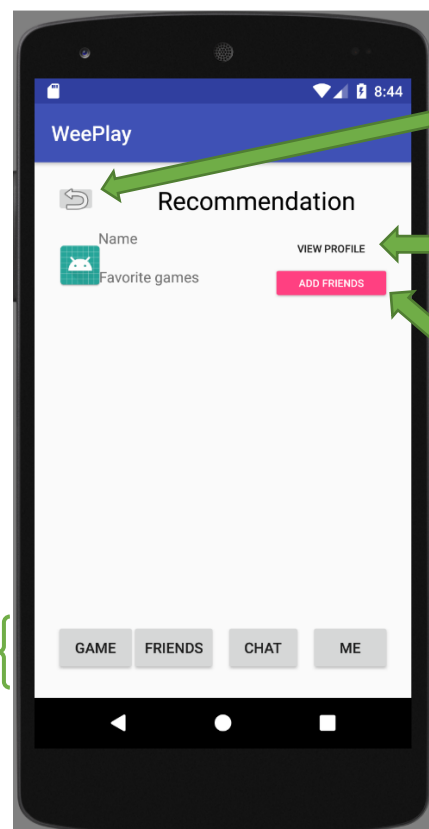
Click Button "RECOMMENDATION", go to **Recommendation.Page**. to see the recommended friends which is depending on the user's profile and the system will do auto-match in database.

Click Button "VIEW PROFILE", go to **ViewProfile.Page**. to see the profile of user's friends in the contact list.

Click Button "CHAT", go to **Chat.Page**. to start chatting.

The four main buttons connect to four main pages. Click each of them, it will take user to each different main page. Now in **Friends.Page**.

Recommendation.Page



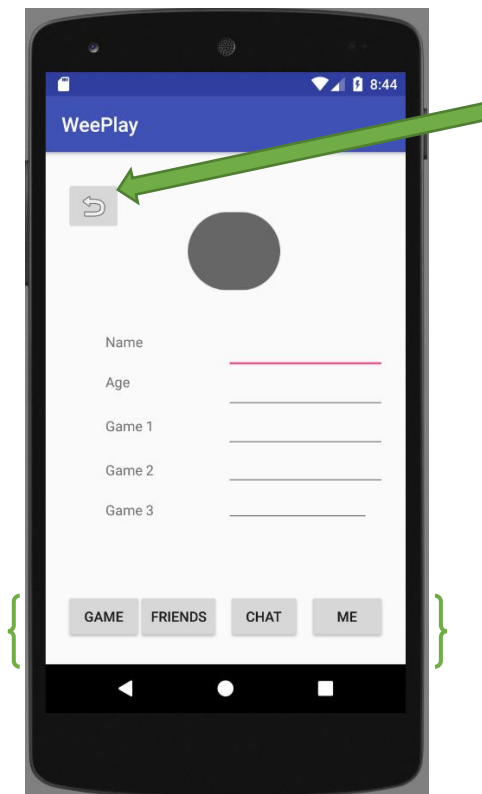
Click Image Button, go back to the **Friends.Page**.

Click Button "VIEW PROFILE", go to **ViewProfile.Page**. to see the profile of recommended friend.

Click Button "ADD FRIENDS", go to **Friends.Page**.

The four main buttons connect to four main pages. Click each of them, it will take user to each different main page. Now if click Button "FRIENDS", it will take user to **Firends.Page**.

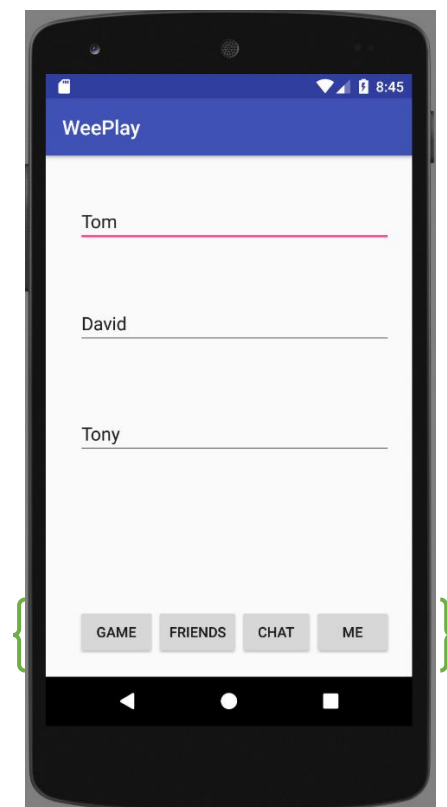
ViewProfile.Page



Click Image Button, go back to the **Friends.Page**.

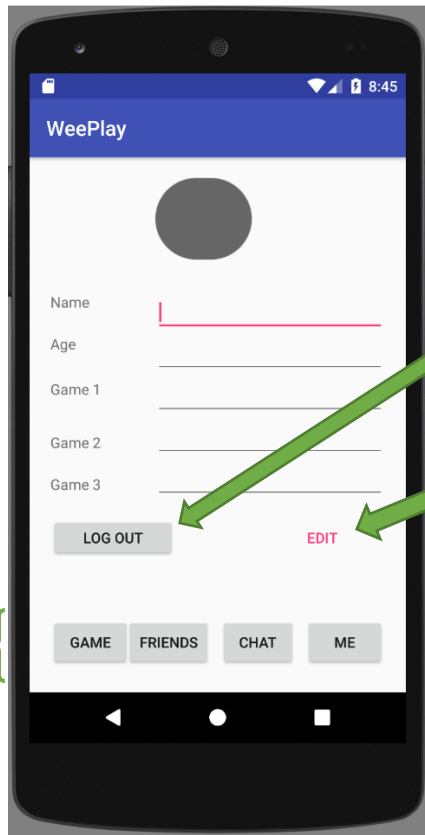
The four main buttons connect to four main pages. Click each of them, it will take user to each different main page. Now if click Button "FRIENDS", it will take user to **Firends.Page**.

Chat.Page



The four main buttons connect to four main pages. Click each of them, it will take user to each different main page. Now in **Chat.Page**.

Me.Page



Click Button "LOG OUT", go back to the **Login.Page**.

Click Button "EDIT", go back to the **Profile.Page**.

The four main buttons connect to four main pages. Click each of them, it will take user to each different main page. Now in **Me.Page**.