### Simple Social Card Collection Battling Game

Design Document

### Table of Content

1. Introduction……………………………………………………………………………...2
2. Design Considerations………………………………………………………………….2

2.1 Assumptions………………………………………………………………………...2

2.2 Design Principles…………………………………………………………………...2

2.3 Development Methods……………………………………………………………..2

3. Architecture Representation…………………………………………………………….2

4. System Architecture……………………………………………………………………..3

5. Policies and Tactics……………………………………………………………………….3

6. Detailed System Design………………………………………………………………….4

6.1 Login………………………………………………………………………………….4

6.2 Panel…………………………………………………………………………………4

6.3 View My Cards - Embedded in Panel…………………………………………...4

6.4 View My Friends - Embedded in Panel………………………………………...5

6.5 Quick Match Waiting - Embedded in Panel…………………………………….5

6.6 In Match…………………………………………………………………………….5

6.7 Chat Window……………………………………………………………………….5

7. Glossary…………………………………………………………………………………..6

### Introduction

This document is specifying the particular design choices being made for the software: Simple Card Collection and Battling Game.

### Design Considerations

##### 2.1 Assumptions

This design assumes that user has a standard mouse and keyboard in a PC platform, with an operating system compatible running Java programs.

Relatively stable internet connection is also assumed.

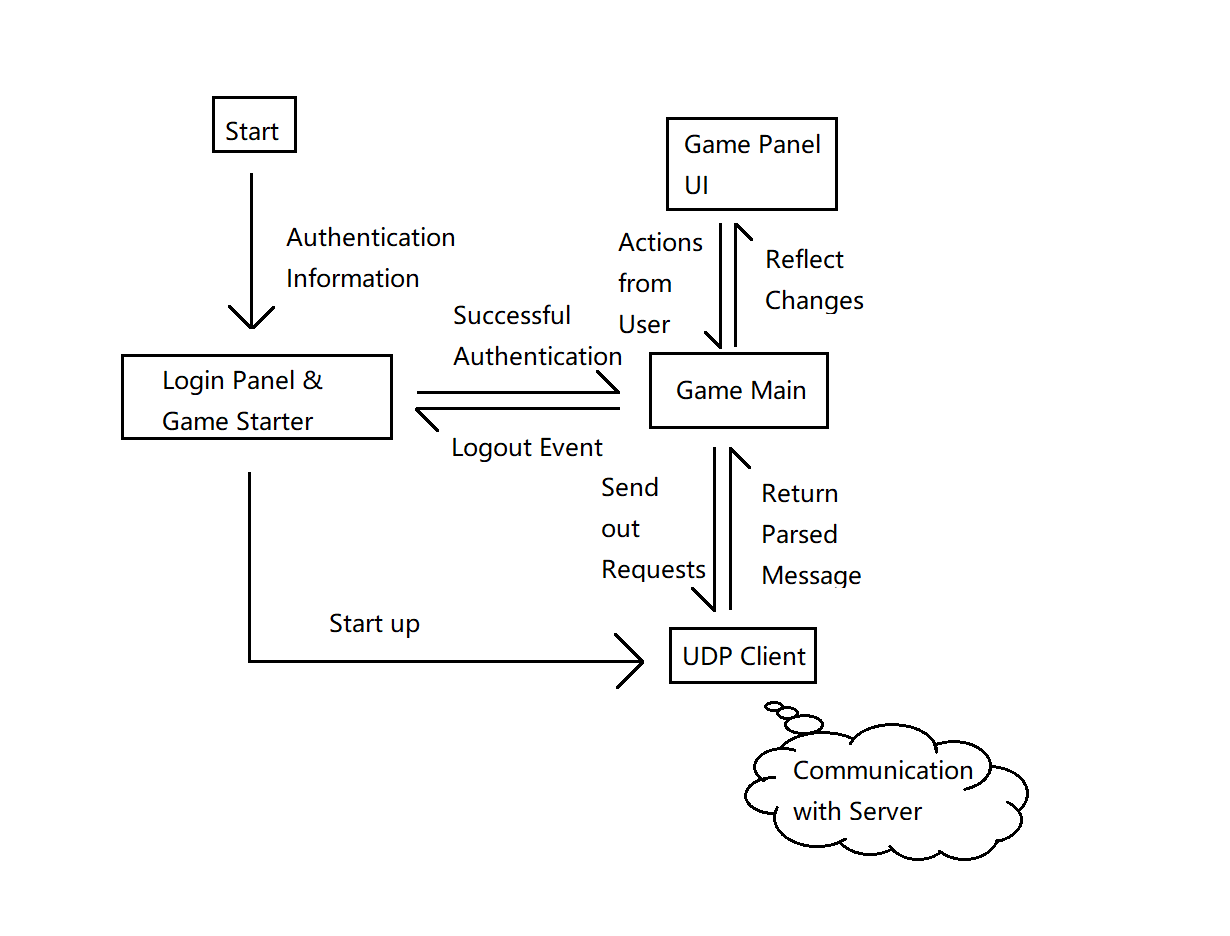
##### 2.2 Design Principles

The Design principle is kept simple, with minimal safety and privacy feature. GUI is kept as straightforward as possible and mostly limiting interactive items to buttons.

##### 2.3 Development Methods

We used Build & Fix, to keep it simple.

### Architectural Representation



### System Architecture

The system is build using object-oriented design. A center object, Game, manages nearly all possible attributes of this software, and GUI objects calls on method of the center object to update itself. The center object has access to all GUI objects to update the display.

The center object operates the communication with server, which packs message according to a defined protocol, and unpacks message from the server and updates itself accordingly.

### Policies and Tactics

This software uses standard JRE 1.8 and Eclipse SWT library for compile and running.

To make GUI not as ugly and code not as messy, SWT library is a better option than Javax -Swing.

Java provides straightforward object-oriented design support.

### Detailed System Design

#### 6.1 Login:

|  |
| --- |
| Label\_Username |
| Textfield\_Username |
| Label\_Password |
| Textfield\_Password |
| Button\_Submit |

Submit button calls a method in Game, packs the two fields into a packet and send to server, and be expecting a result indicating the authentication status.

With successful login, direct user to Panel.

With unsuccessful login, clears two textfields and ask the user to try again.

#### 6.2 Panel:

|  |
| --- |
| Label\_MyNickname |
| Button\_ViewMyCards |
| Button\_ViewMyFriends |
| Button\_QuickMatch |
| Button\_Logout |

Logout redirects user back to login (quitting is implement in closing window)

The other 3 buttons redirects user to their screens, while a request for information update is sent to the server.

#### 6.3 View My Cards – embedded in Panel:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Label\_CardStatus1 | Label\_CardStatus2 | … | … | … | Button\_AddSelected |
| Card 1 | Card 2 | … | … | … |  |
| Label\_CardStatusM | Label\_CardStatusN | … | … | … | Button\_RemoveSelected |
| Card M | Card N | … | … | … |  |
| Label\_DeckStatus | Label\_SelectedCard |  |  |  | Button\_Back |

Card Status label indicates how many cards of this kind are already in the deck.

Card M displays the card image and its description.

User can select cards by clicking on to them, and add and remove operations are only limited to the card selected. (Legal operations check)

Back brings user back to Panel.

#### 6.4 View My Friends – embedded in Panel:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Label\_FriendStatus1 | Label\_FriendStatus2 | … | … | … | Button\_StartChat |
| Friend 1 | Friend 2 | … | … | … | Button\_RequestBattle |
| Label\_FriendStatusM | Label\_FriendStatusN | … | … | … | Label\_SelectedFriend |
| Friend M | Friend N | … | … | … | Button\_Back |

Friend Status Label indicates whether they are online, and whether there are messages or battle requests pending.

Friend M displays their friend’s nickname.

Select a friend by clicking onto the nickname, and Start Chat and Request Battle functions are limited to selected friend.

Back brings user to Panel.

#### 6.5 Quick Match Waiting – embedded in Panel:

|  |
| --- |
| Label\_TimeWaited |
| Button\_CancelQueueing |

Each second queueing status is updated, also updating the time waited.

A successful matchmaking brings users directly to a match.

Cancel Queueing brings user back to Panel.

#### 6.6 In Match:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Enemy\_Slot1 | Enemy\_Slot2 | Enemy\_Slot3 | Enemy\_Slot4 | TextArea\_BattleLog |
| Enemy\_Slot5 | Enemy\_Slot6 | Enemy\_Slot7 | Enemy\_Slot8 | Button\_StartChat |
| My\_Slot1 | My\_Slot2 | My\_Slot3 | My\_Slot4 | Button\_EndTurn |
| My\_Slot5 | My\_Slot6 | My\_Slot7 | My\_Slot8 | Button\_Surrender |
| My\_Hand1 | My\_Hand2 | My\_Hand3 | My\_Hand4 | My\_Hand5 |

Match is separate from Panel, but there is a field prevent user from going to matchmaking or request battle to a friend when a match is active.

Cards are active on the field, and users can perform interactions with them as long as they are legal.

End turn tells the server it is user opponent turn to move.

Surrendering ends the game.

Closing the window also results in surrendering.

#### 6.7 Chat Window:

|  |
| --- |
| TextArea\_MessageLog |
| TextField\_InputMessage |
| Button\_Send |
| Button\_RequestBattle |

Chat window is also separated from panel, and multiple windows can be active at the same time.

All messages are stored in Game, Message Log is updated when Game receives an message or the user sends something out.

User enter message into the input textfield.

Send clears the input field, and sends the message to Game, where Game contacts with server.

Battle Requests can also be sent here.

### Glossary

This design document is made by Lee Kelvin lxk201 and Qianxiang Ma qxm28.