Project Summary

We are implementing a classic 4 player game, mahjong. This is a Chinese Game, a user will be able to play against 3 Artificial Players and display the rules. We will use a basic set of rules and scoring that we identify in the application. You will also be able to have a log in so that you can save your high scores

Project Requirements

1. Must be able to play 1 game of Mahjong
   a. Must be able to play 1 round
      i. Must be able to play 1 hand
2. Must allow User to Log in
3. Must allow User to Save and Access High Score
4. Minimum SDK version is 8
5. User has access to Rules on all screens

Use Cases

As a player so that I can keep my high score I want to log in and have a scoreboard attached to my log in.

As a player so that I can choose how long I want to play for I want to have 3 game type options - 1 hand, 1 round, 1 game.

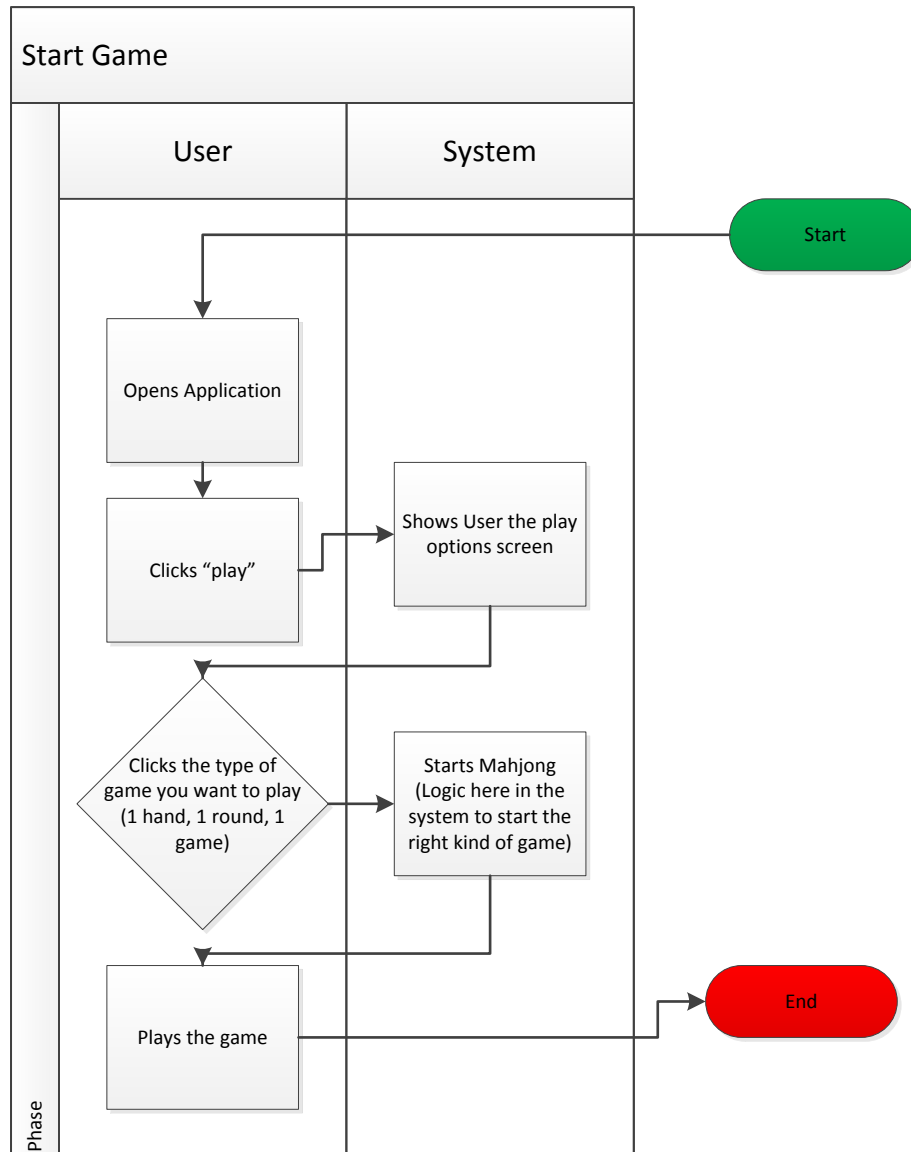As a player so that I can play I want easy to read tiles and a nice backdrop.

As a player so that we can play the game I want to have a rules guide accessible in game and on the main menu.

As a player so that we can play the game without internet I want to have the option to play as a guest.
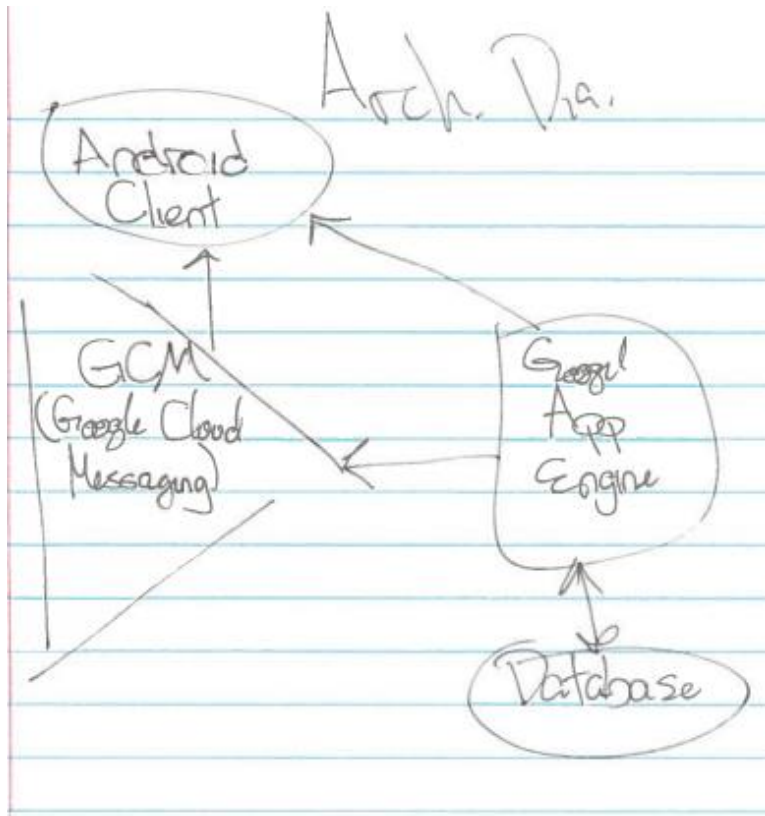
As a player so that I can discard I want to have an easy way to push tiles away.

Activity Diagrams

I drew out the "Start Game" diagram. Most every step a user will take will follow a very similar flow.

Start Game

User	System

Start

Opens Application

Clicks "play"

Shows User the play options screen

Clicks the type of game you want to play (1 hand, 1 round, 1 game)

Starts Mahjong (Logic here in the system to start the right kind of game)
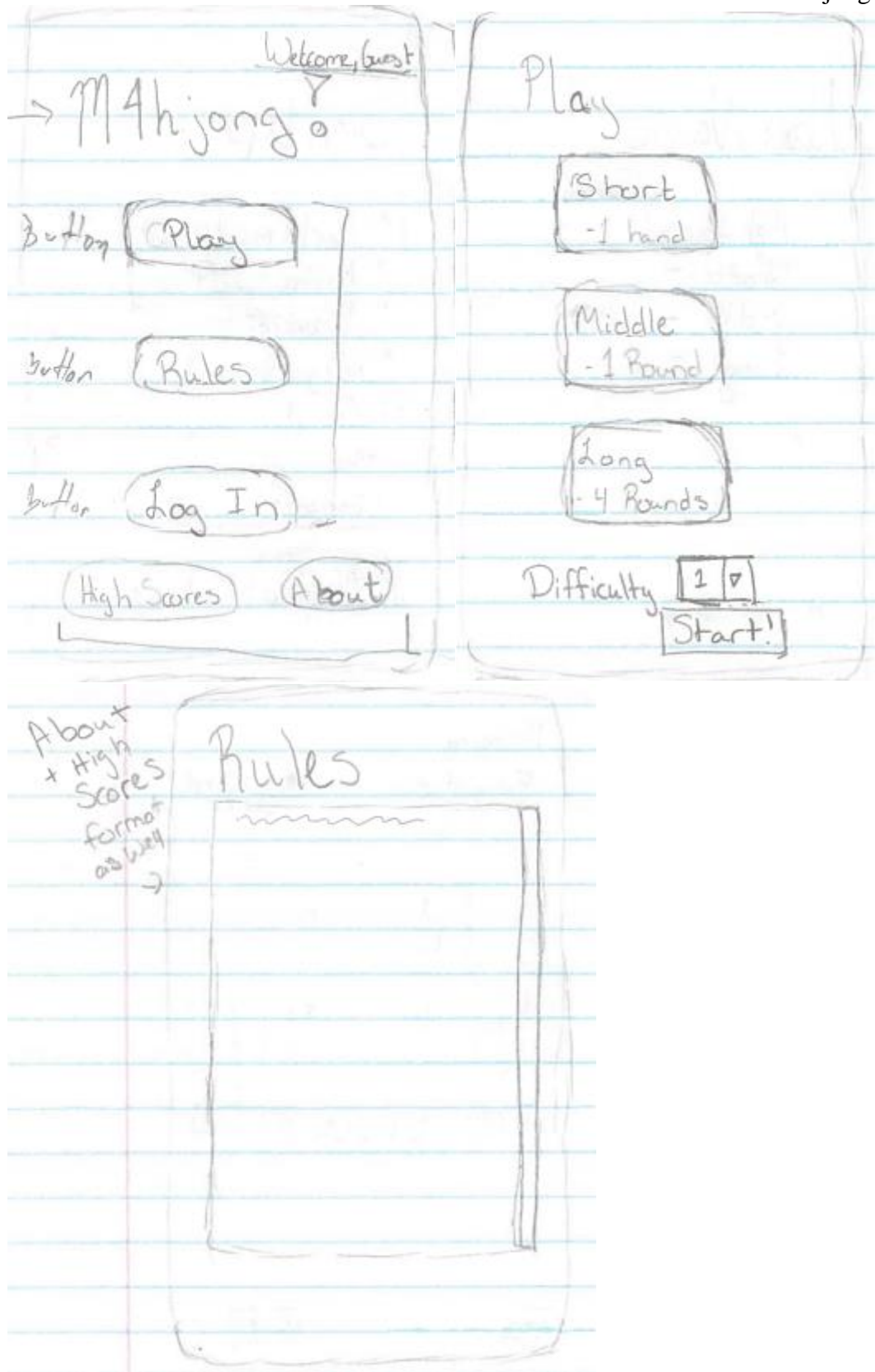
End

Plays the game

Phase

Architecture Diagram

Data Storage

We plan on using the Google API Engine Database with a SQL backend for storing usernames, passwords, and high scores.

UI Mockup

→ M4hjong?

Welcome, Guest

Button [Play]

Button (Rules)

Button (Log In)

(High Scores)   (About)

Play

[Short
  -1 hand]

[Middle
  -1 Round]

[Long
  - 4 Rounds]

Difficulty [1 | ▽]

[Start!]

About
+ High
Scores
format
as well
→

Rules

Log In

User Name: [   ]
Password: [   ]
[Sign in]
[Lost Password]

Don't have an
account? [Sign up!]

User Name

High Scores!
Short —
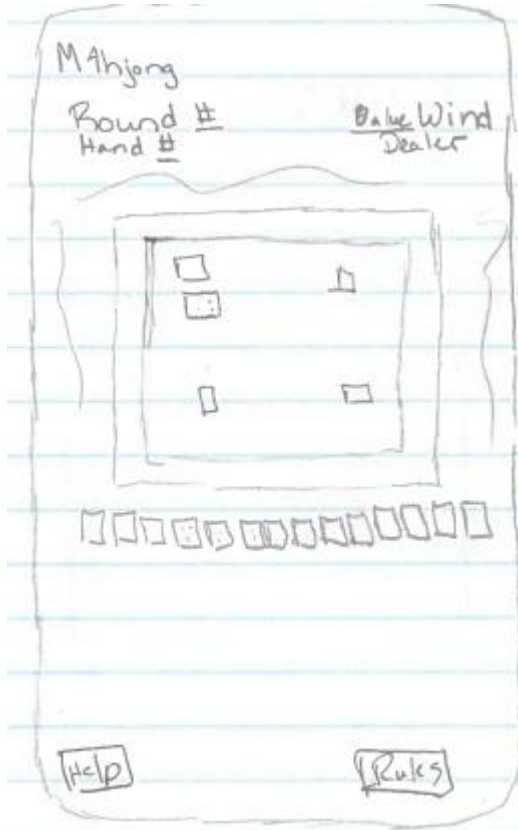Middle —
Long —

Sign Up .

* UserName: [   ]
* Name: [   ]
* Password: [   ]
* Retype Password: [   ]
* Email: [   ]
  We will not send you anything
*Required
Terms: _____

[Accept and
  Sign up]   [Reject]

Lost Password

UserName [   ]

[Send Email Reset]

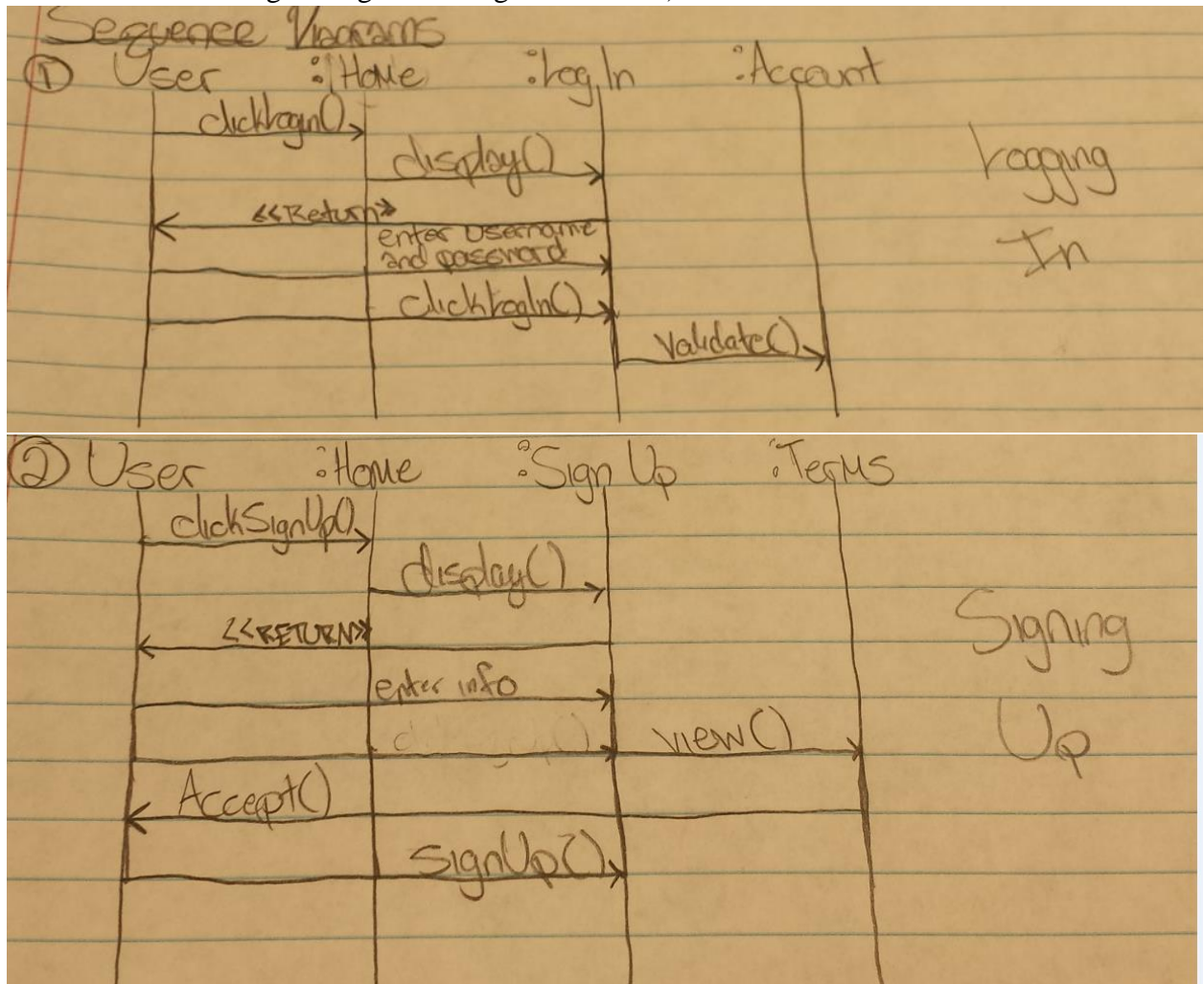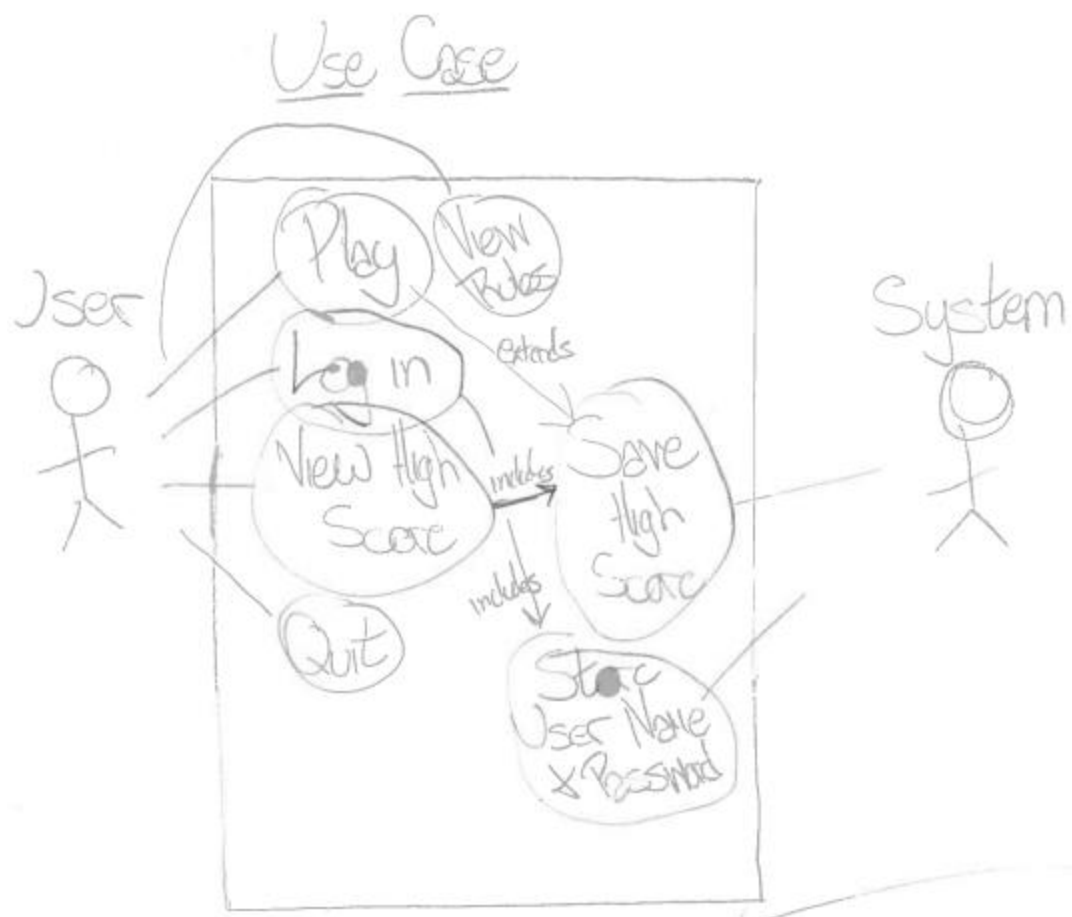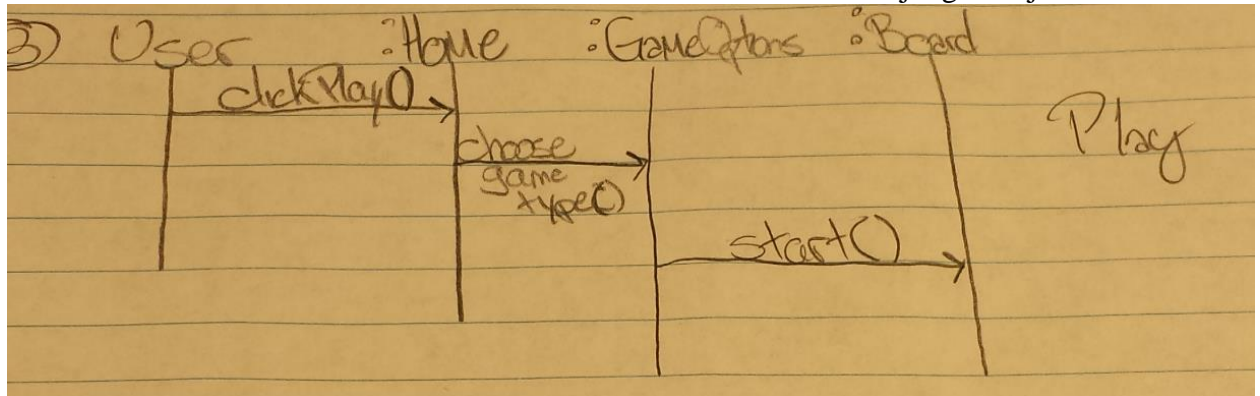### User Interaction

1. User can click button to Log In
    a. User can type Username and Password on Log In screen and click Sign In
        i. Logs you in and brings you to Menu Screen as well as loads correct high scores for User if Username and Password match (Check this through Google API Engine Database)
        ii. If Username and Password don't match an error will be brought up with the option to reset Password
    b. User can click Lost Password if needed
        i. User will provide Username and click Send Email for Password Reset Button
            1. Email will be known through Google API Engine Database
    c. User can click Sign Up if they don't have an account
2. User can Sign Up button
    a. User can provide information for Sign Up fields: Username, Name, Password, Re-type Password, Email
    b. User can view Terms and Conditions and check box if they agree to them
        i. If User doesn't accept Terms and Conditions, they won't be able to Sign Up
    c. User can click Accept to Sign Up after providing their information
    d. User can click Cancel to be returned to Menu Screen without being Signed Up

3. User can click Play button
   a. User if provided with 3 gameplay options: Short Game (1 Hand), Medium Game (1 Round), Long Game (4 Rounds, Real Game)
      i. User will be taken to Game Board after selecting a Game Type and pressing the Start Button
4. User can click About to see information on the app and its developers
5. User can click High Score Button to view their high scores if they are logged in, (Log in will be verified through Google API Engine Database)

3) User   :Home   :GameOptions   :Board

clickPlay()

choose
game
type()

start()

Play

Use Case

User

Play   View
            Rules

Log in   extends

View High   includes   Save
Score   High
                         Score

Quit

includes

Store
User Name
& Password

System

Class Diagram

This is only the class diagram for once you make it past all the menus
(Due to android setups, this would be a lot of classes, and it would get messy and not help make anything more clear)
Game will be a singleton pattern
Player will be a strategy pattern
Game() is the drawing of everything that will exist that each player can see

Game

-instance : Game = null
-Wall : Tile

+getInstance() : Game
+Game() : void

1

4

1

*

Tile

-drawable
-nameOfTile : string

+getTile()

*

1

Player

+HandOfTiles : Tile
+DiscardedTiles : Tile
+PlayedTiles : Tile
+Score : double

+PlayTile()
+DrawTile() : Tile

AIPlayer

+HandOfTiles : Tile
+DiscardedTiles : Tile
+PlayedTiles : Tile
+Score : double

+PlayTile()
+DrawTile() : Tile

HumanPlayer

+HandOfTiles : Tile
+DiscardedTiles : Tile
+PlayedTiles : Tile
+Score : double

+PlayTile()
+DrawTile() : Tile