CS307 -- Software Engineering Design Document

Team 17 - PlayDeck

Greg Yogan
Kyle Fischer
Christian Nitz
Mark Pundmann
Alex Hamm

Table of Contents

1. **Purpose**

People like to play card games with their friends but often find themselves without a deck of cards. The idea of this app is to provide people with a mobile platform on which they can play a variety of card games with friends while on the go. There are currently no applications that can provide this multiplayer functionality.

The following items are major areas of functionality that our team will be focusing on:

1. **Main game functionality:**
   These features are fundamental in our games operation. They represent the users main game playing experience, which is this project's main focus.

    a. As a user, I want to join a game with others over wifi or bluetooth
    b. As a user, I would like to be able to host a game to which others can join either over wifi or bluetooth
    c. As a user, I would like to ensure that I control who joins the game
    d. As a user, I would like to see a list of players in my game
    e. As a user, I would like to have sufficient functionality to allow me to play my card game of choice.
    f. As a user, I would like to remove certain cards from the deck in order to play certain games.
    g. As a user, I would like to choose how the cards are dealt
    h. As a user, I would like to choose how the dealer is selected
    i. As a user, I would like to be able to shake my device to shuffle the cards
    j. As a user, I would like to know when each hand is complete, so the next one can begin promptly.
    k. As a user, I would like to be able to choose if there is a center deck onto which cards are played
    l. As a user, I would like to be able to save a set of rules so that I can efficiently reuse them
    m. As a user, I would like to see the cards in my hand separately from community cards

2. **The User Interface:**
   The user interface is also critical to the games function. This is what allows for a smooth user experience when playing the game. The UI allows the user to painlessly access all of the above features.

    a. As a user, I would like the user interface to be uncluttered and easy to navigate.
    b. As a user, I would like to send feedback to the developers

c. As a user, I would like to join and leave games at will,without significant down-time
d. As a user, I would like to see if anyone else nearby is playing
e. As a user, I would like the cards and graphics to be colorful and distinct
f. As a user, I would like to see suggestions of popular card games to play
g. As a user, I want to be able to quickly start a game from the time of opening the app

3. Features for the developers:

In order to continually improve our product, a mechanism for feedback and iteration must be established. For this, systems must be in place to allow users to give feedback, and the developers to update the game.

a. As a developer, I would like to see user feedback
b. As a developer, I would like to easily update my software
c. As a developer, I would like to be able to easily add new functionality
d. As a user, I would like to send feedback to the developers

General Priorities:

1. Usability

Our main priority in this project is to create a game space that people can easily interact with their friends in. With this goal in mind, our top priorities should involve the ease of use of our product. Ensuring the user's satisfaction in this area will allow the other areas of our product to shine.

2. Performance

The user should be able to play cards and have their decisions propagated to all players quickly (<250ms) so that gameplay is seamless.

3. Security

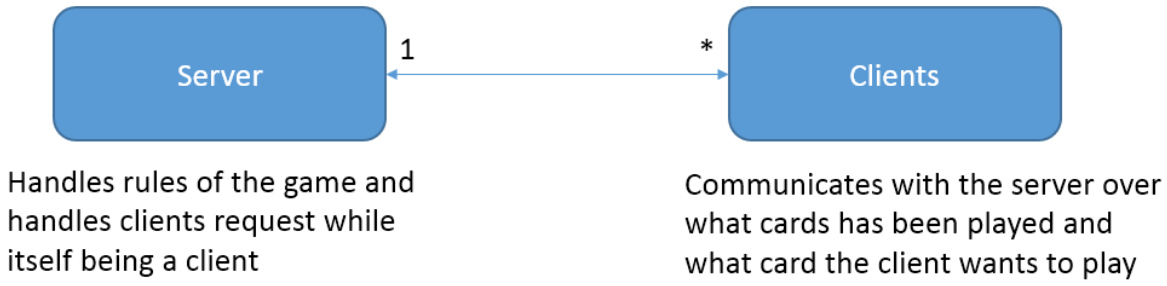
Because this game is played over wifi and bluetooth, malicious attacks could impact our users and their data. Because of this, its is our responsibility as developers to prevent such malicious attacks, and to update and secure our game if any vulnerabilities are found.

4. Maintainability

This app should have rules that are easily modifiable so that future updates could increase the number of games that can possibly be played.

2. Design Outline

High Level Overview



Our program will use peer to peer connectivity through wifi or bluetooth. Users will create games and become the host, allowing other users to join their game and become a client of that host.

A Basic User Story:

The typical user will initially open the app, either want to host a game, or join one. If the user would like to host a game, the user will click that button on the initial menu and be brought to a settings page where they can adjust certain settings before starting the game. If the user wishes to join a pre existing game, they will select the find game button, where they will be shown a list of games they can join. The user will then play their game, and exit the app when they are finished.

Design Issues

Functional Issues:

1. Game alignment / aspect ratio
    a. Landscape
    b. Portrait

    Our choice: Landscape, it allows you to more easily see a full hand of cards due to the wider horizontal field of view. A portrait mode may be implemented later.
2. Target platform
    a. Android
    b. iOS
    c. Desktop (Windows/Mac/Linux)

    Our choice: Android. Android Studio allows for a much easier and streamlined development process. It uses Java and has a fully fledged UI designing tool. iOS may come later, time allowing.
3. How host will communicate with clients
    a. wifi
    b. bluetooth
    c. server/client

    Our choice: we chose to use wifi and bluetooth as that allows users to use wifi when they are near router, or bluetooth when one does not exist. We chose not to use server/client because that will add a lot more scalability issues.
4. How to handle the case when a host disconnects from the game
    a. Disband the lobby and have the users manually remake the game if desired.
    b. Migrate the host to a different person, sending all needed data to the newly selected host

    Our choice: Disband the lobby. The resources required for sending the entire state of play to a newly selected person is very expensive on mobile platforms. Because this game is intended to be played locally, Individuals leaving should not occur very often. Disbanding the lobby is a choice that is simple to understand, and is not computationally expensive.
5. Which rules the host can set
    a. Which cards are in the deck
    b. Who starts as the dealer and if it rotates
    c. How many cards are dealt to each player
    d. Is there a main deck through which cards can be drawn from
    e. Is there a discard pile
    f. If there is a discard pile, when is it cleared and where do those cards go

g. Is there a pile onto which users play cards, and how many cards do they play at a time
h. Can the user set a rule for what cards are allowed to be played onto the table
i. Reshuffle deck after each turn.
j. Can you stack certain cards on the play area
k. Can users trade cards, and with whom
l. If the main pile is face up or down
m. Is there a set of cards played onto the table from the deck, how many are played, and when are they played
n. Can users see other people's cards

Our choice: (As of our current understanding) We chose to focus our rules on a few games such as poker and war. If time allows, we will add more rules to allow the user to play a larger variety of games. Thus our initial rules the user can set will consist of a,b,c,d,e,f,g,h,i,j,l, and m.

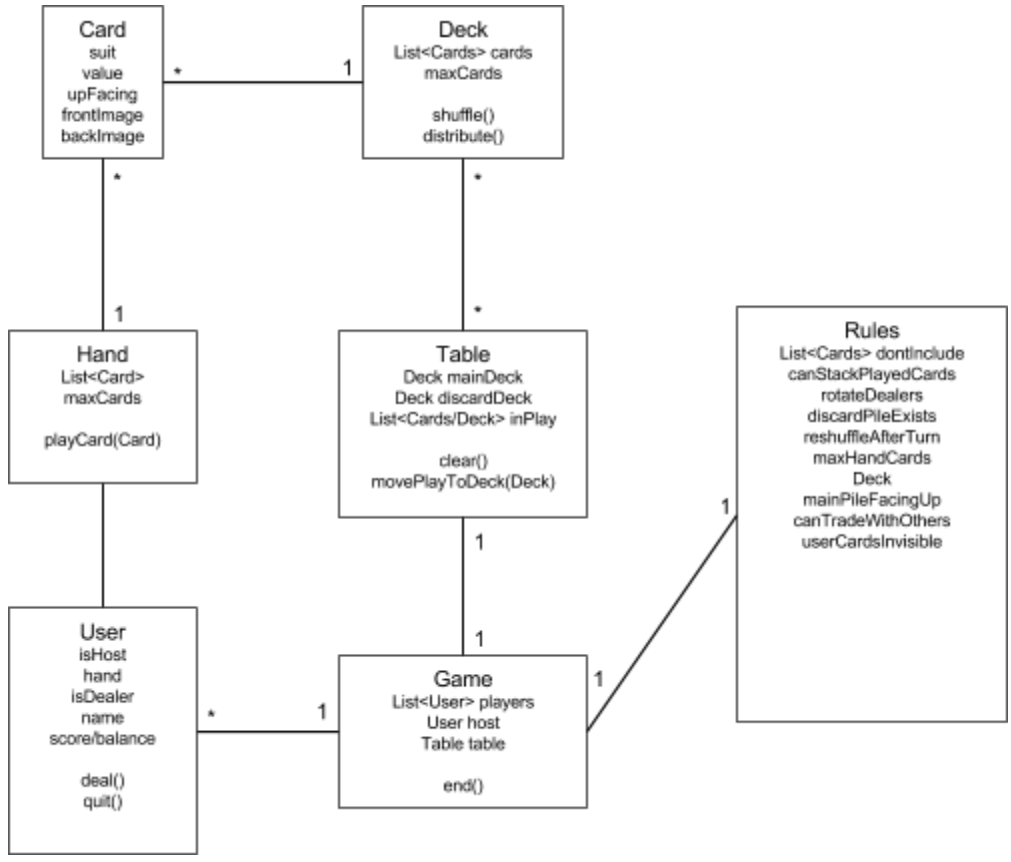
Nonfunctional Issues:

1. How does the app save custom games?
    a. Parseable text file (saved locally)
    b. Saved to an external server
    c. Don't save them and have them decided on the "Host Game" screen

    Our choice: a. For simplicity's sake, a simple method of saving the games locally is preferable. Because of this class's short design time frame, simple, easy to understand implementations allow us to tackle more of our product backlog without spending large amounts of time on this feature.

Design Details:

Class Level Design



Card:

The card class holds information of an individual card, including its suit, value, whether it's facing up or down, and what image each side of the card has.

Deck:

The deck class represents a stack of cards on the table. This could be the main deck in a game of poker or a pile of cards face up, such as in uno. This class will be held by the host, and any information needed will be sent to the players as needed. This will also prevent synchronizations.

Hand:

The hand class contains the information relating to the cards owned by a user. It is their personal collection of card classes that can be used to play/discard on the table. More can be obtained from by drawing from the deck.

Table:

The Table class will contain a main deck, a discard deck, a list to have all the cards that are currently in play and need to be displayed a clear function and a movePlayToDeck function. The main deck will contain all the cards that are yet to be played. The discard deck will contain the cards that are no longer in play. The clear function will clear the table of all cards that have been played. The movePlayToDeck function will move the current cards the the deck specified.

User:

The User class represents an individual device and user connected to a game. This class contains information about the user, such as if it is the host, and also is responsible for leaving the game in a graceful manner.

Game:

The Game class will contain a list of players, it will specify which user is the host. and contain a table. The List of players will be used to show all the user that are in the game. The host user will be used to know what user is the one in charge of the game. The table class will be used to hold the actual game.

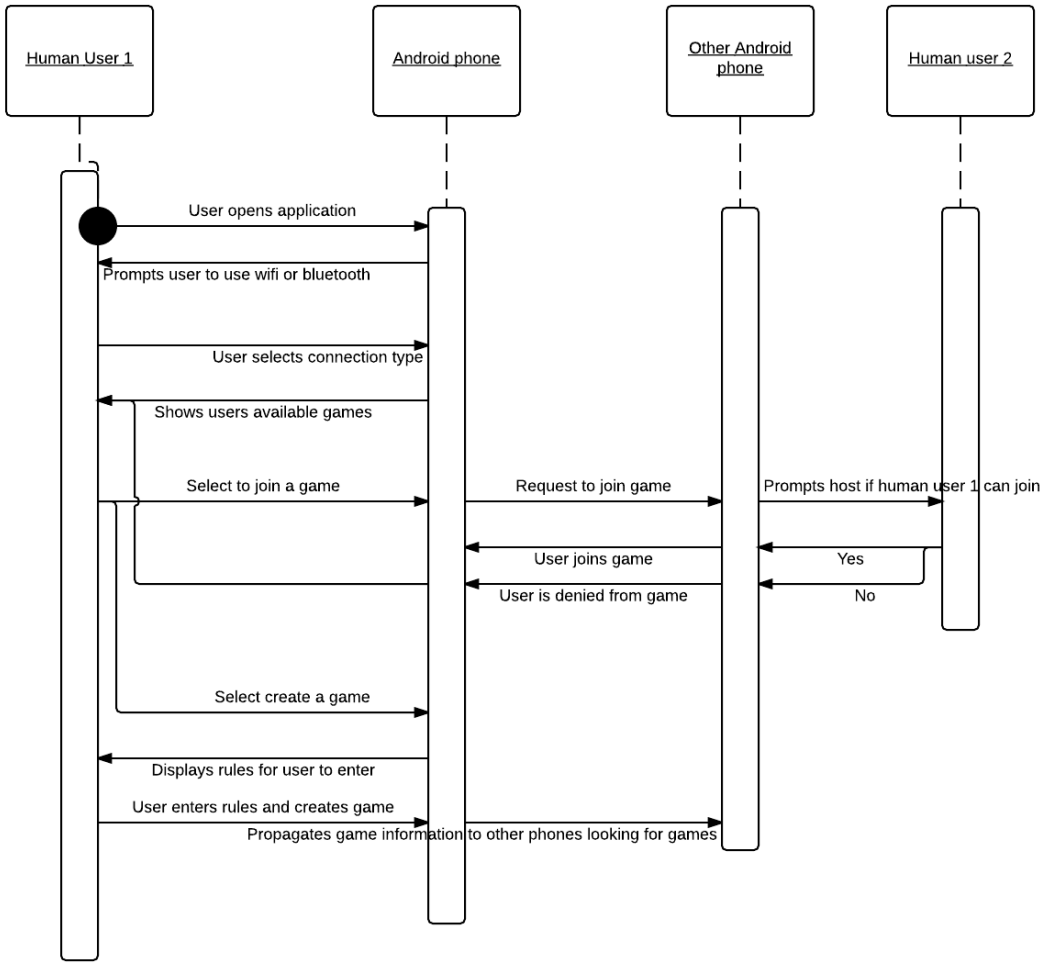
Rules:

The rules class is tied to the game class and will hold information about the game such as where piles are, trading cards with others, being able to pick cards up off the table, and etcetera. These parameters will be sent to each user when they join the game.

Sequence Diagrams

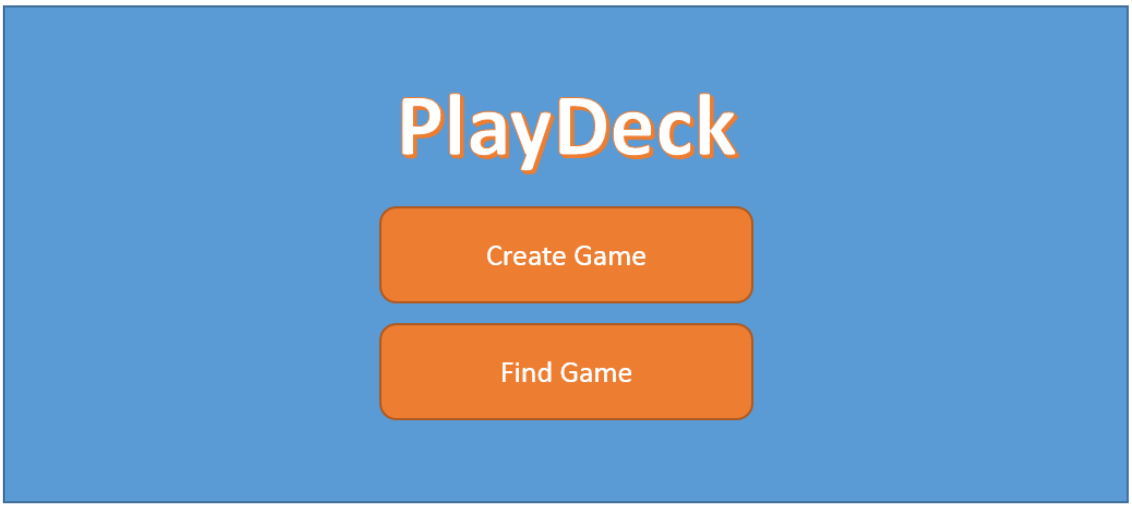
First sequence to user

The outline for what the user experiences when they first open the application and try to join or host a game.

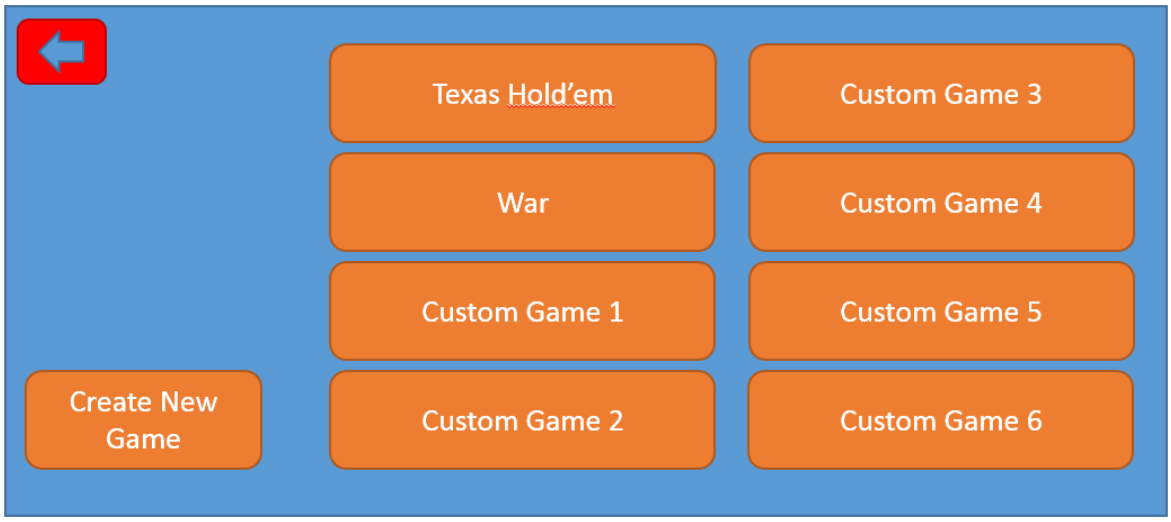


UI Mockups

This is a mockup of the user interface when they first launch that app. They will be presented with two options being whether they want to create a new game or find a game that someone else has made.



A mock up of the Create Game Page. This is where the host user will be able to decide what game they want to play from a list of already made games or ones that he has saved or if he wants to create a brand new game to play.



This is a mock up of the game screen. This is where the game will be played. In the given example we show features that may not be apart of the game being played but they will exist for the games that they do exist in.

This mockup is the screen that will presented to the user if they select find Game. They will be able to see what user is hosting which game. They will select the game they want to play, then it will highlight the one selected and then when you press the button it will open that game.