Group 4

Functional Requirements Document

Team Member 1 Project Manager
Christian Pacheco

Team Member 2 & 3 Software Developer
Keith Rodriguez & Brian Villar Contreras

Team Member 4 Graphic Artist
Olajumoke Olasimbo

Team Member 5 Quality Assurance
Reynaldo Santos

Team Member 6
Researcher/Documentation
Nathalia Salinas

Version	Description of Change	Author	Date
0.0	• Document Created	Christian Pacheco	10/11/2027
0.1	• Introduction (1) • Defined the Purpose (1.1) • Defined the Scope (1.2) • Added Background Info (1.3) • Added References (1.4) • Stated Assumptions and Constraints (1.5) • Defined the Assumptions (1.5.1) • Defined the Constraints (1.5.2) • Wrote the document overview (1.6)	Nathalia Salinas	10/12/2017
0.2	• Other Requirements Added (4)	Nathalia Salinas	10/13/2017
0.3	• Started changelog. • Started detailing functional. Requirements. (3.5.1) • Added more references. (1.4)	Brian Villar	10/15/2017
0.4	• Finished functional requirements. (3.5.1)	Brian Villar	10/16/2017
0.5	• Added Section (2) • Added (4.1.2) • Added (4.1.3) • Added (4.2)	Keith Rodriguez	10/16/2017
0.6	• Interface Requirements (4.1)	Nathalia Salinas	10/16/2017
0.7	• Data Flow Diagrams (3.3) Revision • Logical Data Model/Data Dictionary (3.4) Revision	Christian Pacheco	10/17/2017
0.8	• Added 4.1.1. • Revised 4.2	Olajumoke Olasimbo	10/17/2017

CONTENTS

1 INTRODUCTION..... 4

1.1 Purpose..... 4

1.2 Scope..... 4

1.3 Background..... 4

1.4 References..... 4

1.5 Assumptions and Constraints..... 4

1.6 Document Overview..... 5

2 METHODOLOGY..... 5

3 FUNCTIONAL REQUIREMENTS..... 5

3.1 Context..... 5

3.2 User Requirements..... 5

3.3 Data Flow Diagrams.....6

3.4 Logical Data Model/Data Dictionary..... 6

3.5 Functional Requirements.....6

4 OTHER REQUIREMENTS..... 6

4.1 Interface Requirements..... 6

4.2 Data Conversion Requirements..... 7

4.3 Hardware/Software Requirements..... 7

1 INTRODUCTION

Euchre is a trick card game most commonly played between four people divided into two teams. The game's roots can be traced back to the early German settlers in Pennsylvania. More currently, Euchre is theorized to have derived from an eighteenth-century game Juckerspiel. At times, the game is referred to as Knock Euchre to differentiate it from Bid Euchre. The main game objective involves a Trump. When declaring a suit as trump, a player and his or her partner intends to win most tricks. The hand is scored after all the five tricks have been played. The deal continues to the player on the left until a partnership scores ten points, winning the game.

1.1 Purpose

This Functional Requirements Document focuses on creating a digital version of the Euchre card game. Users will have a quick and simple installation of the game to any device (i.e. mobile, computer, tablet).

1.2 Scope

The scope of the document aims to create an efficient digital version of the Euchre card game. Following the rules of the game, the digital version must contain 4 players. If the game does not have enough users participating, the option of competing against a computer is available. The UI of the game must be simple, efficient, and similar to the physical version while navigating.

1.3 Background

A group of six students in the course CAP 3104 are instructed to create an effective Functional Requirements Document. The document has been divided amongst the six members.

1.4 References

- Group meetings held at the UCF campus and through the Discord messaging application.
- https://en.wikipedia.org/wiki/Euchre
- http://www.dummies.com/games/card-games/euchre/the-basics-of-playing-euchre/
- https://www.youtube.com/watch?v=w-BmY6uhR6k

1.5 Assumptions and Constraints

- Network Connection

- Patent to use Euchre license
- Device capable of playing the application
- Allocated memory to install the software

1.5.1 Assumptions

We assume the spectators can efficiently download the application and have adequate memory available in their devices. The Euchre game should run effortlessly preceding the installation procedure. Additionally, we assume the user has access to a network connection.

1.5.2 Constraints

A large constraint involves using the “Euchre” name for the application to avoid copyright infringement. The game will currently be available to play on a browser using a computer. However, the application will be available on mobile devices in the near future.

1.6 Document Overview

This document provides an overview of the functional requirements to make a digital version of the Euchre card game. First, a general introduction of the game is given. This includes the overall scope, purpose, background, assumptions, and constraints. The discussion then shifts to the overall methodology and description of the functional requirements. These requirements are then displayed with graphic diagrams. All other requirements for the application will additionally be discussed.

2 METHODOLOGY

The Agile Methodology will be used in order to maintain working, tested, and deployable software on a stage by stage basis. This will provide visibility and adaptability in the earlier stages of the project to reduce any risk.

3 FUNCTIONAL REQUIREMENTS

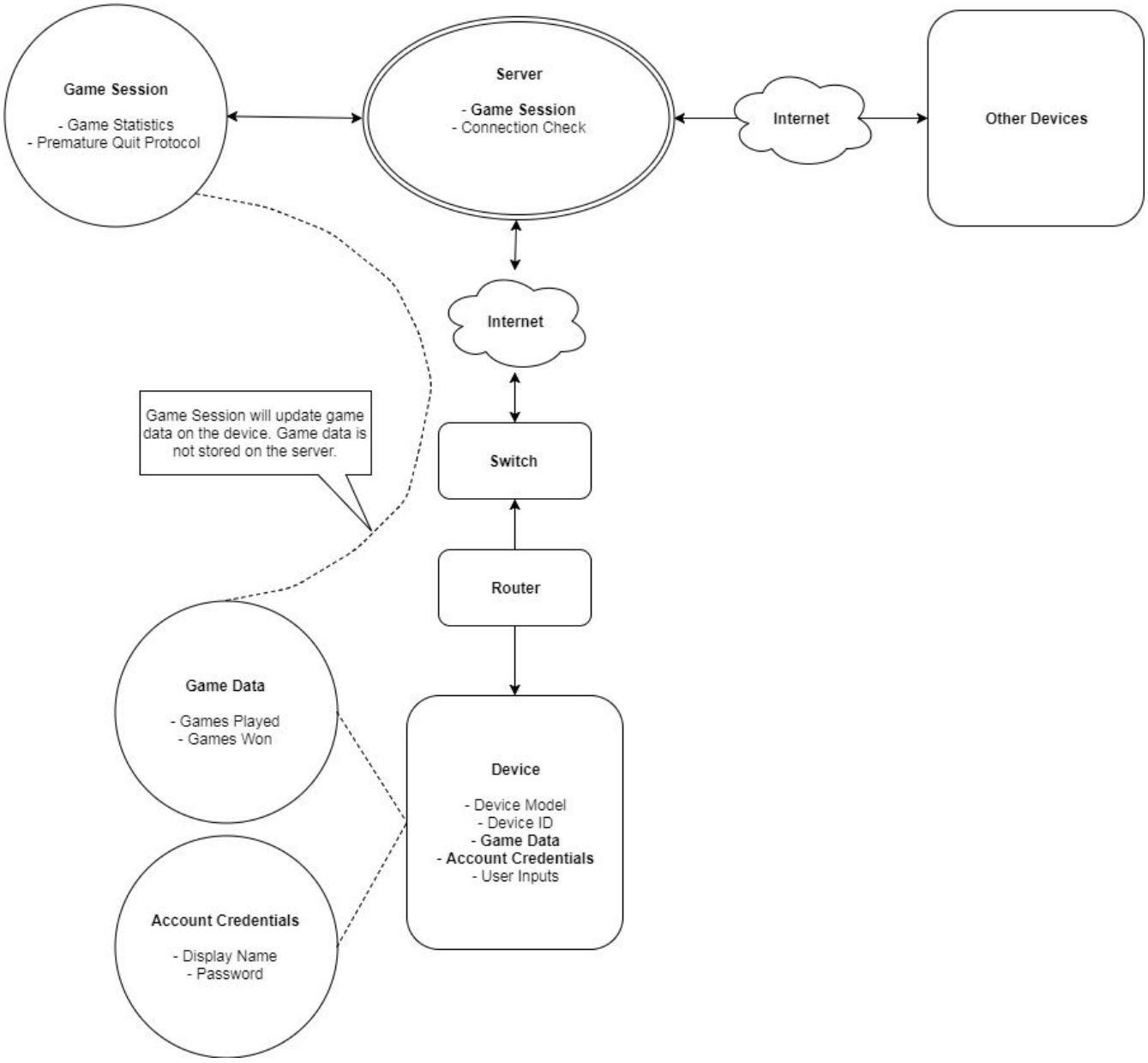
3.1 Context

Application will receive user ID from their device of choice. Application will send user ID to server and server will send back game ID. Game ID is then given back to user.
Exhibit 2 - Generic Context Diagram

3.2 User Requirements

- U - User will need smart devices in order to play.
- U - User will need four people to play game.
- U - User will need to have two groups of two for game.
- U - Users will need to pick the trump card suit.
- Secure network that will keep users from disconnecting from the game.
- Application must maintain consistent appearance.
- Application should be as close as to the card game.
- Application should show each trick points gained after each suit.
- Application should know if you're not following suit for your hand and notify you are not following rules of the game.
- Application will tally up score and first group to 10 wins the game.

3.3 Data Flow Diagram



3.4 Logical Data Model/Data Dictionary

Device: The device will link the user and the system. The device will connect the game application to a server through the internet. Each player's device will create their own data independent of the server and all other players. The application is initialized by the user on the device's provided system. Provides device model, device ID, account credentials, and game

data. Sends user inputs and available system resources. Inputs can include mouse clicks, screen taps, keystrokes, and other commonly utilized forms of input.

Account Credentials: Stores user information for use in any played games. Includes Display Name and password, both of which are required information for server communication. Account credentials are not stored on the server, but on their device so account statistics can be viewed at any time.

Game Data: Game data will be stored on the device and automatically be created and updated. Stored information includes games played, games won, and other relevant data. Each device will have its own game data that is updated only after a complete game and only by the game session.

Server: Main component of the application and enable other entities to communicate with each other. Will receive Also hosts each game for all players. Server will run regular connection checks to ensure each player is still connected to the game. A fail state should be run when a player loses connection to the game. The server will handle user inputs and insert them into the game as

Game Session: Creates a dedicated session for all players. Statistics generated in each game will be saved to the User ID and Device from the game sessions through the server. Players will start a new game once 4 players are able to connect. Should a player quit prematurely, the game should update the games won statistic on all remaining players. If the server initiates a fail state after a lost connection, a notification should be sent out notifying other players and will wait for the person to connect back in 2 minutes or game will automatically close with no updates to game data.

Other Devices: All players will communicate with each other through other devices. The application should be able to connect 4 players through the internet onto a server and into a single game session.

3.5 Functional Requirements

3.5.1 Functional Requirements Group 1

Section/ Requirement ID	Requirement Definition

FR 1.0.	The system shall simulate a traditional game of Euchre.
FR 1.1	The system shall make two teams.
FR 1.1.1	The system shall assign two players per team.
FR 1.1.2	The system shall decide the first dealer at random.
FR 1.1.3	The system shall allow the first dealer to be chosen to choose the point total needed to win the overall game.
FR 1.1.4	The system shall have a 10 point game minimum for every game. The system shall have a 50 point maximum to win the game.
FR 1.1.5	The system shall select new dealers in a clockwise manner.
FR 1.1.6	The system shall call each individual hand a trick.

3.5.2 Functional Requirements Group 2

Section/ Requirement ID	**Requirement Definition**
FR 2.0.	The system shall use a proper deck of cards of a traditional game of Euchre.
FR 2.1	The system shall only use cards designated 9, 10, Jack, Queen, King, and Ace of each suit.
FR 2.1.1	The system shall end up with a deck of 24 cards for the game.
FR 2.1.2	The system shall use the remaining 30 shuffled cards to determine the potential trump suit.
FR 2.1.3	The system shall deal five cards to each player
FR 2.1.4	The system shall deal in alternating sets of 2 and 3 cards.
FR 2.1.5	The system shall set aside the remaining 4 cards. **This is the Kitty.**

3.5.3 Functional Requirements Group 3

Section/ Requirement ID	Requirement Definition
FR 3.0	The system shall have a proper trump selection mechanism. Trump suit = strongest suit for current game.
FR 3.1	The system shall flip a card from the Kitty to determine the potential trump suit.
FR 3.1.1	The system shall ask the player to the dealer's left if they accept the trump suit. **1st round** If player refuses: Ask the next player. Note: All players may refuse the system's trump suit. **2nd round** If all players refuse the system's trump card: Flip another card from the Kitty and ask players (starting from the dealer's left) if they want that suit as trump. Scenario 1: player picks a suit, and trump is decided. Scenario 2: player refuses to pick and passes, the next player may pick the trump. Note: This keeps happening until a trump is picked or Kitty is exhausted.
FR 3.1.2	The system shall ensure that if a trump suit is chosen by the dealer, then the following must occur: ● The dealer must take the card from the Kitty into his hand and replace his weakest card with it. If the trump is selected by a player that's not the dealer, no replacement needs to take place.
FR 3.1.3	The system shall declare a misdeal if no player chooses a trump suit

	during the aforementioned second round.
FR 3.1.4	The system shall keep track of the team that chose the trump suit.
FR 3.1.5	The system system shall declare the ‘Jack’ of the trump suit as the strongest card. The system shall call this card the ‘Right Bower’
FR 3.1.6	The system shall declare the ‘Jack’ of the suit that’s the same color as the trump suit as the second strongest card. The system shall call this card the ‘Left Bower’.
FR 3.1.7	The system shall ensure that the ‘Left Bower’ is considered the same suit as the trump suit for the purposes of the current game.

3.5.4 Functional Requirements Group 4

Section/ Requirement ID	**Requirement Definition**
FR 4.0	The system shall enforce proper game play.
FR 4.1	The system shall not start game play until a trump suit is selected.
FR 4.1.1	The system shall ensure that the first card of game play is played by the player to the left of the dealer. The player may play any suit.
FR 4.1.2	The system shall remind (but NOT force) that all subsequent players ‘follow suit’ if possible. Following suit = playing a card of the suit that the current game is being played on..
FR 4.1.3	The system shall allow a player to play a different suit if so they choose. ● This suit can be a trump card if possible ● This suit can be any other suit if the player doesn’t have any other option.

FR 4.1.4	The system shall allow players to call renege to other players. ● Renege: If a player plays OFF suit when they had the proper suit available.
FR 4.1.5	The system shall award 4 points to the team that called a proper renege. The system shall penalize a team 4 points for a false ‘renege’ call.
FR 4.1.6	The system shall award the current trick to the player that plays the highest card.
FR 4.1.7	The system shall allow a trump card to beat a higher card of a non-trump suit.
FR 4.1.8	The system shall ensure that if more than one trump card is played, the highest trump card wins the trick.

3.5.5 Functional Requirements Group 5

Section/ Requirement ID	**Requirement Definition**
FR 5.0	The system shall have a proper scoring system.
FR 5.1.1	The system shall add the number of tricks won by each team after the end of every hand.
FR 5.1.2	The system shall award **1 point** to the team that chose the trump suit if they **win 3 or 4 tricks**.
FR 5.1.3	The system shall award **2 points** if the team that chose the trump suit **wins 5 tricks** (the entire hand).
FR 5.1.4	The system shall award **2 points** to the **opposite team** if the team that chose the trump suit doesn’t win at least 3 tricks. This means that the team that called the trump suit has been **Euchred!**
FR 5.1.5	The system shall declare the first team to reach or surpass the score

	chosen in the beginning of the game the winner.

4 OTHER REQUIREMENTS

The interface should be able to store each player's score count to keep track of the number of tricks won by each player. Additionally, the system should keep track of the current trump and the right and left bowers. Internet is necessary as users will compete online. The system needs access to the cloud in order for users to store their player information.

4.1 Interface Requirements

As the user opens the software, a menu displays on the screen. Multiple menu options are available for the user, such as "new player", "log in", or "guest." The user then selects whether to play online or play against the computer. If the user selects online gameplay, the system stores this information and selects the other three players from the web at random. If the user wishes to compete against the AI, the application proceeds to ask the user the level of difficulty he or she wishes to battle and sets the user to play against three computer generated players.

The screen then prompts a "Ready to play?" queue, displaying information, such as the current dealer and the partnerships. The application asks the first assigned dealer an additional question concerning the total points needed to win the overall game. After each user selects the "Start Game" option, a new screen is displayed, the game begins, and the cards are sorted. The users are asked which suit they wish to make trump. Once the users decide on a trump suit, the game begins to the left of the dealer. The monitor displays the point of view of the user, showing the user holding the cards. The other players' cards are shown as well. However, their cards are presented slanted at angles not possible for the user to see. The system will monitor each player's game play to ensure proper points are rewarded. If the user selected the easy difficulty setting when competing with an AI, suggested cards will be highlighted to help the player.

While the user is in play mode, a button to call Renege on the other players becomes available. This occurs if a player does not follow suit when he or she is able to. Once a user calls Renege, all other users receive a prompt. The opposing team (the one that called Renege) receives two points. This button is displayed inside the top menu bar. The menu bar additionally contains the options "Game" and "Options."

The Game menu allows the user to easily change from competing against players online and competing against the AI.

The Options menu includes “game difficulty” settings (when competing against AI), “start new game,” “how to play,” “sound,” and a “my statistics” menu. “Start new game” allows users to leave the current game and find new partners for a new game. “How to play” guides users through a tutorial showing a step by step process of paying the Euchre card game, along with a list of all the technical terms and rules.

The sound menu allows users to control the level of volume or turn off/on the sound of the entire game.

The “my statistics” menu gives a summary of the current user’s overall wins and losses. A scorecard is displayed at the center of the top menu bar. This card displays the points for the competing partnerships. After the end of every hand, the number of tricks won by each team are added. Based on the number of tricks won, proper points are rewarded. The game continues, and the next player to the left of the current dealer becomes the new dealer. If a team has been Euchred, a prompt will display to notify all players. Once one partnership exceeds the winning score chosen from the first dealer, the winner is selected.

4.1.1 Hardware Interfaces

The hardware requirements for the game must be able to run on Android's running on the operating system 4.0.4, “Ice Cream Sandwich.” While, for iPhones the IOS system should be running on 7.1.2. Most smart-phones created after 2012 should be able to run the application.

4.1.2 Software Interfaces

The application should interface with Network interface in order to obtain access to the cloud server. The server will support the game by obtain user ID and sending back the user’s game ID which will be utilized when playing with other players via P2P sessions. The application will also need to interface with device storage so that single player gameplay can be done locally from the user’s device.

4.1.3 Communications Interfaces

The game should be able to play over the internet using any network connection the device has access to. User’s should also be able to connect locally without network utilizing local Wifi(hotspot), Wifi - Direct, or bluetooth.

4.2 Data Conversion Requirements

As the development of this application will be using modern mobile/portable devices there is no need for legacy data conversion.

4.3 Hardware/Software Requirements

A smart phone with internet connection that will allow for each player to communicate in the game. Android should have the latest OS and so should Apple phones. Each will need to be able to run application on their phones in order to play against each other in teams.