Software Requirements Specification

for

Detonating Baby Felines

**Version 1.0**

**Prepared by**

**Group Name: United Nations Unite**

|  |  |  |
| --- | --- | --- |
| **Adam Newell** | **axn180025** | **adam.newell@utdallas.edu** |
| **Andres Blanco** | **axb180036** | **Axb180036@utdallas.edu** |
| **Jacob Bias** | **jxb180001** | **jxb180001@utdallas.edu** |
| **Kevin Lie** | **kxl180022** | **kevin.lie@utdallas.edu** |
| **Tim McGuire** | **tsm080020** | **tsm080020@utdallas.edu** |
| **Osato Aigbomian** | **oea140130** | **oea140130@utdallas.eud** |

|  |  |
| --- | --- |
| **Instructor:** | **Neeraj Gupta** |
| **Course:** | **SE 6387.MH1** |
| **Date:** | **March 7, 2020** |

**Contents**

[Contents 1](#_Toc34507870)

[1 Introduction 1](#_Toc34507872)

[1.1 Document Purpose 1](#_Toc34507873)

[1.2 Product Scope 1](#_Toc34507874)

[1.3 Intended Audience and Document Overview 1](#_Toc34507875)

[1.4 Definitions, Acronyms and Abbreviations 1](#_Toc34507876)

[1.5 Document Conventions 1](#_Toc34507877)

[1.6 References and Acknowledgments 1](#_Toc34507878)

[2 Overall Description 2](#_Toc34507879)

[2.1 Product Overview 2](#_Toc34507880)

[2.2 Product Functionality 2](#_Toc34507881)

[2.3 Design and Implementation Constraints 2](#_Toc34507882)

[2.4 Assumptions and Dependencies 2](#_Toc34507883)

[3 Specific Requirements 3](#_Toc34507884)

[3.1 External Interface Requirements 3](#_Toc34507885)

[3.1.1 User Interfaces 3](#_Toc34507886)

[3.1.1.1 Login Screen 3](#_Toc34507887)

[3.1.1.2 Main Screen 3](#_Toc34507888)

[3.1.1.3 Gameplay Screen 3](#_Toc34507889)

[3.1.2 Hardware Interfaces 3](#_Toc34507890)

[3.1.3 Software Interfaces 3](#_Toc34507891)

[3.2 Functional Requirements 4](#_Toc34507892)

[**3.2.1** **F1**: The system shall allow users to load the lobby 4](#_Toc34507893)

[**3.2.2** **F2**: The lobby shall display list of games available to join 4](#_Toc34507894)

[**3.2.3** **F3**: The system shall allow user to create a game with 2-5 players with a name 4](#_Toc34507895)

[**3.2.4** **F4:** The user shall be able to join a game 4](#_Toc34507896)

[**3.2.5** **F5**: The user shall be able to quit any game they have joined 4](#_Toc34507897)

[**3.2.6** **F6**: The system shall maintain game state cross-client in real time 4](#_Toc34507898)

[**3.2.7** **F7**: The system shall maintain game state in database 4](#_Toc34507899)

[**3.2.8** **F8**: The system shall notify user when it is their turn 4](#_Toc34507900)

[**3.2.9** **F9**: The user shall be able to draw a card 4](#_Toc34507901)

[**3.2.10** **F10**: The user shall be able to play any number of valid cards from their hand 4](#_Toc34507902)

[**3.2.11** **F11**: The system shall not let players see contents of deck, unless specific card allows for it 4](#_Toc34507903)

[**3.2.12** **F12**: The system shall provide feedback messages to the users, such as warnings of game status change events. 4](#_Toc34507904)

[**3.2.13** **F13**: The user shall be able to have active status disabled upon death condition 4](#_Toc34507905)

[3.3 Use Case Model 5](#_Toc34507906)

[3.3.1 U1: Load Web App 6](#_Toc34507907)

[3.3.2 U2: Join Game 7](#_Toc34507908)

[3.3.3 U3: Create Game 8](#_Toc34507909)

[3.3.4 U4: Quit Game 9](#_Toc34507910)

[3.3.5 U5: Play Action Card 10](#_Toc34507911)

[3.3.6 U6: Draw Card 11](#_Toc34507912)

[3.3.7 U7: Player Death 12](#_Toc34507913)

[3.4 Card Play Mechanics 13](#_Toc34507914)

[3.4.1 Non-action Card 13](#_Toc34507915)

[3.4.2 Detonating Baby Feline 13](#_Toc34507916)

[3.4.3 Defuse 13](#_Toc34507917)

[3.4.4 Nope 13](#_Toc34507918)

[3.4.5 Attack 13](#_Toc34507919)

[3.4.6 Skip 13](#_Toc34507920)

[3.4.7 Favor 13](#_Toc34507921)

[3.4.8 Shuffle 14](#_Toc34507922)

[3.4.9 See the Future 14](#_Toc34507923)

[4 Other Non-functional Requirements 15](#_Toc34507924)

[4.1 Performance Requirements 15](#_Toc34507925)

[4.1.1 P1: The system shall work on all browsers 15](#_Toc34507926)

[4.1.2 P2: The system shall work with X users at the same time with no impact on performance 15](#_Toc34507927)

[4.1.3 P3: Average page load (From a user perspective) should be less than 1 second 15](#_Toc34507928)

[4.2 Software Quality Attributes 15](#_Toc34507929)

[4.2.1 Availability 15](#_Toc34507930)

[4.2.2 Usability 15](#_Toc34507931)

[4.2.3 Reliability 15](#_Toc34507932)

[4.2.4 Integrity 15](#_Toc34507933)

**Revisions**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Primary Author(s)** | **Description of Version** | **Date Completed** |
| 1.0 | Timothy McGuire, Andres Blanco, Adam Newell, Jacob Bias, Kevin Lie, Osato | Initial draft of requirements document | 03/07/2020 |

# Introduction

*The goal of this project is to design and implement a web-based multiplayer card game called Detonating Baby Felines. In this section we will go delve into the details of the project and what is required of the system for the game to work.*

## Document Purpose

This document’s purpose is to list and describe the requirements needed to design and implement the web-based multiplayer card game, Detonating Baby Felines. This document will not go into details about specific technologies to achieve requirements.

## Product Scope

*The scope of the project is the design and implementation of the web-based multiplayer card game, which includes any technologies and databases needed to achieve the final product.*

## Intended Audience and Document Overview

*This document is intended to be viewed by developers, project managers, testers, stake-holders, and users to clarify the purpose of the project and what is expected from the application.*

## Definitions, Acronyms and Abbreviations

*User Interface (UI) – The user facing part of the application. This is where the user can view or play their cards*

## Document Conventions

*Follows standard document conventions.*

## References and Acknowledgments

*This requirement document follows the IEEE SRS format.*

# Overall Description

## Product Overview

*Detonating Baby Felines is a turn-based card game in which the goal is to knock the other players out of the game. This product will move this card game from the physical space to the digital space so that users can enjoy this game with other players around the world.*

## Product Functionality

*The product shall have the following functionality:*

* *Allow users to create User Accounts*
* *Allow users to create games*
* *Allow users to join games*
* *Allow users to leave games*
* *User interface that allows the user to view and play their cards on their turn*
* *A back-end system that will manage the state that each game is in*

## Design and Implementation Constraints

*There are no requirements specifically related to the design and implementation of the final product. The developers have free reign to implement the project in the best way they see fit.*

*The only constraint is that it must be completed and functional by the end of the Spring 2020 semester.*

## Assumptions and Dependencies

* *The card game is turn based and can operate in a stateful way*
* *Users will need to have internet access*
* *Project is web-based and should work on all browsers*

# Specific Requirements

## External Interface Requirements

### User Interfaces

#### Login Screen

This is the user interface that shall allow a user to log into their account or create an account.

#### Main Screen

*This is the user interface that shall list all active games the user is a part of. Clicking on an active game should send them to the Gameplay Screen.*

*This screen also allows the user to create or join a new game.*

#### Gameplay Screen

*This is the user interface that will allow a user to interact in the game.*

*When it is not a user’s turn, the user shall be able to view cards played and their hand.*

*When it is a user’s turn, the user shall be allowed to play a valid card in addition to viewing cards played and their hand.*

*There shall also be a button that lets a user leave the current game.*

*There shall be a button that allows a user to go back to the Main Screen.*

*Selecting a card on the Gameplay Screen should display the details of the card in a user-friendly way without needing another screen.*

### Hardware Interfaces

*There are no hardware interface requirements. This product should work on all desktop computers with an internet connection and web browser.*

### Software Interfaces

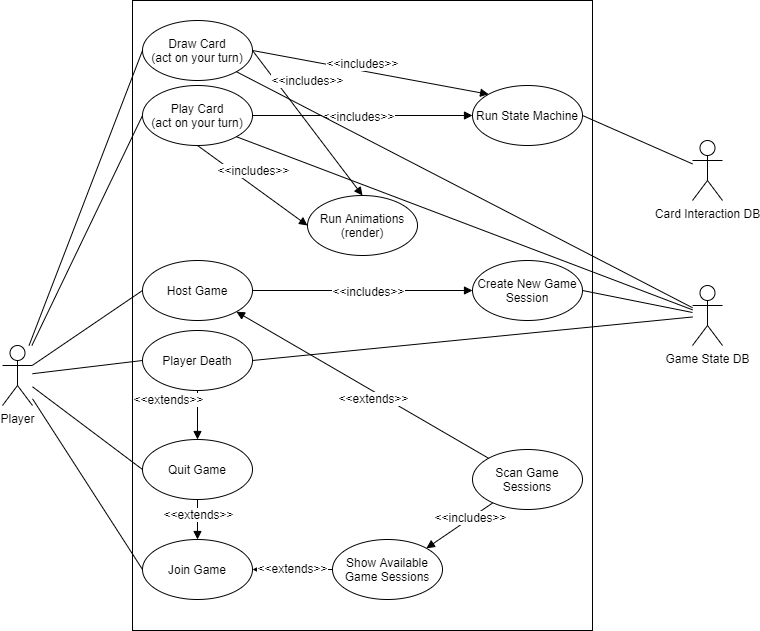
*The client-side software shall connect and communicate to the server-side software.*

*The server-side software shall handle the processing of the game, the client-side software is for communicating to the server and displaying the game only.*

## Functional Requirements

* + 1. **F1**: The system shall allow users to load the lobby
    2. **F2**: The lobby shall display list of games available to join
    3. **F3**: The system shall allow user to create a game with 2-5 players with a name
    4. **F4:** The user shall be able to join a game
    5. **F5**: The user shall be able to quit any game they have joined
    6. **F6**: The system shall maintain game state cross-client in real time
    7. **F7**: The system shall maintain game state in database
    8. **F8**: The system shall notify user when it is their turn
    9. **F9**: The user shall be able to draw a card
    10. **F10**: The user shall be able to play any number of valid cards from their hand
    11. **F11**: The system shall not let players see contents of deck, unless specific card allows for it
    12. **F12**: The system shall provide feedback messages to the users, such as warnings of game status change events.
    13. **F13**: The user shall be able to have active status disabled upon death condition

## Use Case Model



**Figure 1: Use Case Diagram for Game System**

### U1: Load Web App

**Author –** Andres Blanco

**Purpose** – Player is presented with primary screen for creating and joining games

**Requirements Traceability –** F1, F2, F3

**Priority** - High.

**Preconditions** – None

**Post conditions** – The player can successfully create or join a game

**Actors** – Player, Game State DB

**Extends –** None

**Flow of Events**

1. Basic Flow - flow of events normally executed in the use-case
   1. A player launches the web application
   2. They are automatically placed into a lobby which displays open games
2. Alternative Flow - a secondary flow of events due to infrequent conditions
   1. None
3. Exceptions - Exceptions that may happen during the execution of the use case
   1. The server hosting the game is down

**Includes** **–** N/A

**Notes/Issues** – N/A

### U2: Join Game

**Author –** Andres Blanco

**Purpose** – Join an existing game session

**Requirements Traceability –** F4, F6, F7

**Priority** - High.

**Preconditions** – Player must be be shown the Lobby screen

**Post conditions** – The player successfully joins a game session and has provided a display name

**Actors** – Player, Game State DB

**Extends –** Show Available Game Sessions

**Flow of Events**

1. Basic Flow - flow of events normally executed in the use-case
   1. A list of open games is listed for the player to choose from
   2. The player selects a game
   3. The player joins the selected game
2. Alternative Flow - a secondary flow of events due to infrequent conditions
   1. A list of open games is listed for the player to choose from
   2. The player selects a game
   3. The selected game is no longer a valid option
   4. The user is returned to the main screen to try again
3. Exceptions - Exceptions that may happen during the execution of the use case
   1. There are no open games

**Includes –** N/A

**Notes/Issues** – N/A

### U3: Create Game

**Author –** Andres Blanco

**Purpose** – A player creates a game session for other players to join

**Requirements Traceability –** F3, F6, F7

**Priority** - High.

**Preconditions** – Player must be shown the Lobby screen

**Post conditions** – The player has created a named game with 2-5 players

**Actors** – Player, User Account DB

**Extends –** Scan Game Sessions

**Flow of Events**

1. Basic Flow - flow of events normally executed in the use-case
   1. A player clicks the Start New Game button
   2. The player it added to a new open game that waits for more players
   3. Once the number of players is reached the game starts
2. Alternative Flow - a secondary flow of events due to infrequent conditions
   1. None
3. Exceptions - Exceptions that may happen during the execution of the use case
   1. Not enough other players join within the set timeframe

**Includes** **–** Create New Game Session

**Notes/Issues** – N/A

### U4: Quit Game

**Author –** Andres Blanco

**Purpose** – A player leaves an active session

**Requirements Traceability –** F5, F6, F7

**Priority** - High.

**Preconditions** – Player must be one of the active players of an active game session

**Post conditions** – The player successfully leaves the game, the game is considered a loss and goes back to the Lobby

**Actors** – Player

**Extends –** Join Game

**Flow of Events**

1. Basic Flow - flow of events normally executed in the use-case
   1. The player clicks the menu button
   2. The player clicks the quit game button in the menu
   3. The player exits the game and it is treated as a loss
2. Alternative Flow - a secondary flow of events due to infrequent conditions
   1. None
3. Exceptions - Exceptions that may happen during the execution of the use case
   1. None

**Includes** **–** N/A

**Notes/Issues** – N/A

### U5: Play Action Card

**Author –** Tim McGuire

**Purpose** – A player plays a card during their turn

**Requirements Traceability –** F6, F7, F8, F10, F11

**Priority** - High.

**Preconditions** – It is the players turn

**Post conditions** – The card is checked to see if it is valid, if valid the game will process the card played

**Actors** – Player, Game State DB, Card Interaction DB

**Extends –** N/A

**Flow of Events**

1. Basic Flow - flow of events normally executed in the use-case
   1. The player drags an action card from their hand into play
   2. The card action is triggered
   3. The user returns to 1.1
2. Alternative Flow - a secondary flow of events due to infrequent conditions
   1. The player drags an action card from their hand into play
   2. The card action is triggered
   3. The card action causes the players turn to end
3. Exceptions - Exceptions that may happen during the execution of the use case
   1. The player has no cards to play

**Includes** Run State Machine, Run Animations

**Notes/Issues** – N/A

### U6: Draw Card

**Author –** Tim McGuire

**Purpose** – A player plays a card during their turn

**Requirements Traceability –** F6, F7, F8, F9, F11

**Priority** - High.

**Preconditions** – It is the players turn

**Post conditions** – The card is checked to see if it is valid, if valid the game will process the card played

**Actors** – Player, Game State DB, Card Interaction DB

**Extends –** N/A

**Flow of Events**

1. Basic Flow - flow of events normally executed in the use-case
   1. The player automatically draws a card
   2. The player moves to the next phase of their turn
2. Alternative Flow - a secondary flow of events due to infrequent conditions
   1. The player automatically draws a card
   2. The card’s on draw action is triggered
   3. The player moves to the next phase of their turn
3. Exceptions - Exceptions that may happen during the execution of the use case
   1. None

**Includes** Run State Machine, Run Animations

**Notes/Issues** – N/A

### U7: Player Death

**Author –** Tim McGuire

**Purpose** – A player dies and can no longer participate

**Requirements Traceability –** F6, F7, F9, F13

**Priority** - High.

**Preconditions** – Player draws a card

**Post conditions** – The player can no longer participate. May be able to remain as a spectator or return to Lobby

**Actors** – Player, Game State DB

**Extends –** Quit Game

**Flow of Events**

1. Basic Flow - flow of events normally executed in the use-case
   1. The player remains in the game as a spectator
   2. When the game is over it is treated as a loss for that player
2. Alternative Flow - a secondary flow of events due to infrequent conditions
   1. The player exits the game and it is treated as a loss
3. Exceptions - Exceptions that may happen during the execution of the use case
   1. None

**Includes** N/A

**Notes/Issues** – N/A

## Card Play Mechanics

* + 1. **Non-action Card**
       1. Two of a Kind

Transfer ownership of random card from selected player

* + - 1. Three of a Kind

Name a card, if selected player has it, transfer ownership

* + - 1. Five Different Cards

Take card from discard pile

* + 1. **Detonating Baby Feline**
       1. Must be played immediately after draw
       2. Unless player plays Defuse, player dies
    2. **Defuse**
       1. Only playable after drawing/playing Detonating Baby Feline
    3. **Nope**
       1. Stop any action
    4. **Attack**
       1. End turn without drawing
       2. Next player takes two turns
    5. **Skip**
       1. End turn without drawing a card
    6. **Favor**
       1. Force a player to transfer a card of their choosing
    7. **Shuffle**
       1. Shuffle deck
    8. **See the Future**
       1. Privately view the top 3 cards of the deck

# Other Non-functional Requirements

## Performance Requirements

### P1: The system shall work on all browsers

### P2: The system shall work with X users at the same time with no impact on performance

### P3: Average page load (From a user perspective) should be less than 1 second

## Software Quality Attributes

### Availability

*The backend services shall be available and accepting connections 95% of the time.*

### Usability

*The user interface shall be simple, intuitive, and easy to use.*

### Reliability

*The system should perform and store all game moves on the first try. The system should never get confused about whose turn it is, what cards have been played, etc.*

### Integrity

*The data from the user game session shall not cross-pollinate with another game session the user has active.*