

EECE 4520 – Software Engineering

RFID Arcade Final Report

Team BCON:

Connal West

Nicholas Sullo

Ben Prisby

Omar Tuffaha

Table of Contents

1. Problem Statement	5
1.1 Objective	5
1.2 Rationale (value proposition)	5
1.3 Existing Systems	5
1.4 Proposed Systems	5
2. Team Details and Plan of Action	6
2.1 Team Roles	6
2.2 Action Plans	6
3. Requirements Analysis	6
3.1. Introduction	6
3.1.1 Purpose of this Section	6
3.1.2 Scope of the Development Project	7
3.1.3 Definitions, Acronyms, and Abbreviations	7
3.1.4 References	7
3.1.5 Overview of Section	7
3.2 General Description	7
3.2.1 User Characteristics	7
3.2.2 Product Perspective	8
3.2.3 Overview of Functional Requirements	8
3.2.4 Overview of Data Requirements	8
3.2.5 General Constraints, Assumptions, Dependencies, Guidelines	9
3.2.6 User View of Product Use	9
3.3 Specific Requirements	9
3.3.1 External Interface Requirements	9
3.3.1.1 User Interfaces	9
3.3.1.2 Hardware Interfaces	10
3.3.1.3 Communication Interfaces	10
3.3.2 Detailed Description of Functional Requirements	10
3.3.2.1 Template for describing functional requirements	10
3.3.2.2 Functional requirements from 2.3	11
3.3.2.2.1 Functional Requirement 2.3.1	11
3.3.2.2.2 Functional requirement 2.3.2	11
3.3.2.2.3 Functional requirement 2.3.3	11
3.3.2.2.4 Functional requirement 2.3.4	12
3.3.2.2.5 Functional requirement 2.3.5	12
3.3.2.2.6 Functional requirement 2.3.6	12
3.3.2.2.7 Functional requirement 2.3.7	13

3.3.2.2.8 Functional requirement 2.3.8	13
3.3.2.2.9 Functional requirement 2.3.9	13
3.3.3 Performance Requirements	13
3.3.4 Quality Attributes	14
3.4 Other requirements	14
3.5 Changes to Requirements During Development	14
4. System Design	14
4.1 Introduction	14
4.1.1 Purpose of this Section	14
4.1.2 Scope of the Development Project	15
4.1.3 Definitions, Acronyms, and Abbreviations	15
4.1.4 References	15
4.1.5 Major Software Requirements	15
4.1.6 Design Constraints, Limitations	15
4.1.6.1 Design Constraints	15
4.1.6.2 Limitations	16
4.1.7 Changes to requirements	16
4.1.8 Overview of Section	16
4.2 Data Design	16
4.2.1 Game Data	17
4.2.2 Player Data	17
4.2.2.1 Game Statistics	17
4.2.3 Prize Data	18
4.3 System Architecture Description	19
4.3.1 Overview of Modules / Components	19
4.3.1.1 Playing Subsystem	19
4.3.1.2 Kiosk Subsystem	19
4.3.1.3 Backend Subsystem	20
4.3.2 Structure and Relationships	20
4.4 Detailed Description of Components	20
4.4.1 Component Template Description	20
4.4.2 Description of RFID Card	21
4.4.3 Description of RFID Reader	21
4.4.4 Description of Arcade Game	23
4.4.5 Description of Rewards Center	24
4.4.6 Description of General Kiosk	24
4.4.7 Description of Server (Backend)	25
4.4.8 Description of Display	26

4.5 Interface Design	26
4.6 Software Design Goals	27
4.7 Changes to Design During Development	27
5. Test Document	27
6. User Manual	27
6.1 Staff User Manual	28
6.2 Player User Manual	28
7. Glossary	29
8. References	30
9. Appendices	31
9.1 Appendix A: Use Case Diagram	31
9.2 Appendix B: Class Diagram	32
9.3 Appendix C: Sequence Diagram	33
9.4 Appendix D: RFID Reader, Game, and Kiosk State Diagrams	34
9.5 Appendix E: Scenario	35
9.6 Appendix F: System Architecture Diagram (Subsystem interaction level)	35
9.7 Appendix G: System Architecture Diagram (Component level)	35
9.8 Appendix H: Test Plan Diagram	36
9.9 Appendix I: Completed Gantt Chart	36

1. Problem Statement

This section details the objective and rationale for designing the RFID Arcade. Additionally, the specifics of the system, both in terms of design and requirements, are detailed and the differences between this system and other available systems are addressed.

1.1 Objective

The objective of this project is to add more functionality to an already existing idea: the combination of traditional arcades, and the accessibility of RFID chips. This was accomplished through the use and integration of a sophisticated backend which keeps track of unique client IDs, to which it attaches a ticket value, a token value, high scores, and a record of games played.

1.2 Rationale (value proposition)

The real value in the RFID Arcade project is in the data collected and stored in the backend based upon how the customers in the arcade play the games. This usage data can then be examined by the staff and management of the arcade to determine which games are being played most often, and which games may need to be phased out based on low usage. The usage statistics can also be used for quantitatively determining peak times at the arcade for each day of the week. Obviously some of this data could be collected in a traditional arcade by monitoring the number of tokens bought on any given day, but the RFID arcade system, and the backend attached to it, provide a single location for this data to be accessed.

1.3 Existing Systems

There are systems currently available which allow owners of arcades to retrofit a traditional coin operated arcade to a card based payment system, whether that be a card with a magnetic strip, a barcode, or RFID technology [4]. These systems include a device for integrating with already existing arcade games which may take coins or tokens to interface with the card technology, and also include a system for adding new funds to the card. There are no existing systems though which aggregate all of this player data in a useful way, or any systems which allow the player to choose their prizes at a kiosk by tapping their card and viewing redeemable prizes. These two areas of interest are the true advantages and unique aspects of the system outlined below.

1.4 Proposed Systems

The proposed systems are a series of networked machines communicating with a backend that will facilitate the playing of arcade games, ticket generation, and prize redemption. Arcade owners will be able to view statistics of most arcade games played as well as the most popular days for each arcade game. Players will be issued an RFID card using an available kiosk by exchanging currency for tokens, which are necessary to play arcade games. The RFID card

serves as a unique identifier for the player within the system. Players may elect to participate in games which publish statistics to the backend. Once the player has finished playing, they may redeem their tickets for prizes, which are selected at a special Rewards Kiosk. General Kiosks will allow the player to view their total number of tokens, tickets available for redemption, and interesting gameplay statistics as customized by the client.

2. Team Details and Plan of Action

This section details the members of the team and their roles over the course of the project. This section also outlines the timeline and process of development for the project.

2.1 Team Roles

After the original planning, the team roles were distributed as follows: Ben would create a flexible backend and facilitate the integration of the kiosk and arcade game with the backend, Omar would focus more on the arcade game itself and its connection to the backend, Nick would primarily research how to integrate the RFID reader with all of the applications, and I was tasked with programming the kiosk. However, over time the roles changed and Ben facilitated most of the programming, while Connal and Nick transitioned over to writing documentation, reports, etc. Omar's role was mostly unchanged.

2.2 Action Plans

As can be seen in the completed Gantt Chart below in Appendix H, Ben and Omar eventually were assigned to most of the tasks. Connal and Nick were delegated to support roles such as documentation writing and some research into how the RFID Readers worked. It's an unfortunate truth that there were many unexpected challenges that popped up that hindered the progress of the development. As such, our action plan was modified to reflect this. In particular, less people were assigned to software development tasks to reduce the amount of communication needed to progress with the project. This is important, as this project had many software components acting independently of one another, so reducing the communication load was key to our eventual success. Additionally, with such a complex project, more documentation to describe and detail it was deemed necessary. Additionally, with less people programming, that meant that we didn't need as many RFID readers for testing purposes, removing the need to share our 2 RFID readers (since we only had 2). Overall, our action plan was adjusted so that we would stay on schedule.

3. Requirements Analysis

3.1. Introduction

3.1.1 Purpose of this Section

The purpose of this section is to describe the function and performance specification requirements for an RFID Arcade, as well as to highlight the purpose of this system in greater

detail. This section is intended to be used by a business owner who wishes to develop their preexisting arcade into an RFID arcade.

3.1.2 Scope of the Development Project

The RFID Arcade is a series of machines connected with a backend that will facilitate the playing of arcade games, ticket generation, as well as prize redemption. Arcade owners will be able to view statistics of most arcade games played as well as the most popular days for each arcade game. Players will be able to “load” an RFID Arcade card using an available kiosk by exchanging currency for “Tokens,” which are necessary to play arcade games. Players will be able to use their RFID card both to play arcade games and to record the number of “Tickets” they won on the backend. Additionally, players will be able to redeem their tickets for prizes, which can be selected at a “Rewards Center”. Kiosks will be able to display a player’s total number of tokens, tickets available for redemption, and most popular games.

3.1.3 Definitions, Acronyms, and Abbreviations

See Section 7 below for a full glossary of terms used in this report.

3.1.4 References

See Section 8 below for a full list of references used in the creation of this report.

3.1.5 Overview of Section

The remainder of this section contains three sections: the General Description section, the Specific Requirements section, and the Other Requirements section. The General Description section will describe the RFID arcade user requirements and constraints. The Specific Requirements section will detail the performance and functional requirements for this system. The Other Requirements section will elaborate on any extraneous details not covered in the Specific Requirements section.

3.2 General Description

This section will give a client-oriented description of the system.

3.2.1 User Characteristics

There are two types of people that will interact with this system: players and arcade staff. These two types of people interact with the system differently.

Players use a kiosk both to obtain an RFID card and to put tickets on their card. Additionally, players can view statistics on the most played games along with the total number of tickets tied to the card’s player ID. Players spend tokens that are stored on their card to play games in the arcade in the hopes of winning tickets. Players may redeem their tickets at a Rewards Center for prizes, which the arcade staff give to the players.

The arcade staff will be able to view different statistics about the usage of the arcade on a monitor connected to the backend. This is also where they will receive updates that a customer needs to receive a prize at the Rewards Center.

3.2.2 Product Perspective

This system consists of five parts: the arcade games, at least one kiosk, the Rewards Center, the server backend, and the RFID cards. The arcade games, Kiosk, and Rewards Center are all connected to the backend at all times, while the RFID card would be standalone.

The RFID cards will be distributed to the players using a Kiosk. After which, the card is assigned a player ID which is stored on the card and an entry in the database is also made for that player ID. The entry in the database tracks the games a customer plays as well as keeping track of the total number of tickets a player has accumulated. A player can put tokens on the card by using the Kiosk again. In order to play an arcade game, a player must spend tokens by tapping the RFID card at the arcade game and accepting the payment of the tokens. After a player plays an arcade game, the arcade game will update the total ticket value stored under the customer's ID in the backend. At any point a player can tap their RFID card at a Kiosk to see information such as total number of tickets their account has, and the number of tokens currently on the card. Players can tap their card at the Rewards Center to view potential prizes, the ticket cost of each prize, and their current ticket balance. The Rewards Center is able to access and update all of this information on the backend.

3.2.3 Overview of Functional Requirements

- Kiosks should be able to distribute RFID Cards with matching player ID's in the backend
- Kiosks should be able to add tokens to RFID Cards
- Kiosks should be able to display the last played game, total number of tickets, and total number of tokens
- Arcade games should be able to decrease the number of tokens on a card
- Arcade games should be able to increase the total number of tickets a player has based upon their performance in the game
- The Rewards Center should be able to decrease the total number of tickets a player has
- The Rewards Center should be able to display available prizes with their matching ticket prices
- The Rewards Center should be able to alert a staff member that a player has redeemed tickets for a specific prize
- The staff members should be able to view statistics that the backend provides such as most played game, most popular days, and most popular games for a given date and time

3.2.4 Overview of Data Requirements

Only two components in the system store any information: the player's RFID card and the backend. The RFID card stores a player ID that corresponds to metrics in the backend, and locally stores the total number of tokens in the player's account. The backend has a record of

each customer, their played games, the games in which they won the most tickets, the total number of tickets a player has, and which prizes the customer has purchased.

The kiosk can display information from the backend such as total number of tickets available for use. The kiosk can also read the number of tokens stored on the RFID card and display this back to the customer. A kiosk can also increase the number of tokens on an RFID card when the customer purchases them.

An arcade game can decrease the number of tokens on an RFID card, as well as increment the total number of tickets a customer has stored in their backend entry. Additionally an arcade game can display how many tokens a customer currently has on their card, and after a game will display how many tickets the player won for that game and how many total tickets they have.

A Rewards Center can display how many tickets a player has by querying the database, as well as how many tickets each available prize costs. All of this information is stored in the backend. A Rewards Center can also decrease the number of tickets a customer has in the database, once the player decides to redeem their tickets for a prize.

3.2.5 General Constraints, Assumptions, Dependencies, Guidelines

This system necessitates an arcade with at least one game, and is really designed to be retrofitted onto an already existing arcade. Additionally, at least one staff member is required at the arcade to distribute prizes. At least one kiosk is also required, which is stocked with at least one RFID card which can be dispensed to players. At least one Rewards Center is required for customers to choose prizes they wish to redeem accrued tickets for. Additionally, a maximum RFID card recognition time of three seconds is required at the arcade games, the kiosks, and the Rewards Center. The Rewards Center, kiosks, and arcade games need to always be connected to the backend. There needs to be at least one prize for redemption at all times. The backend needs to take less than three seconds to update a single entry.

3.2.6 User View of Product Use

The players will use blank white RFID Cards. They will scan these on an RFID reader for each game. Each game will have a different appearance depending on what the game is. The user will be able to use a Kiosk to check their remaining tokens and total number of tickets. For example, a user might tap their RFID card on a reader and see the following output below:

Remaining Tokens: 12

Total Tickets: 235

3.3 Specific Requirements

3.3.1 External Interface Requirements

3.3.1.1 User Interfaces

When a player touches a Kiosk they should see the following options: “Dispense New RFID Card”, “Add more Tokens to Card”, and “Display Card Information”. When the “Dispense New RFID Card” option has been selected the player will be asked to type in a player name using the touchscreen keypad. After which a new RFID card will be dispensed, with the accompanying user entry in the server backend. When the “Add more Tokens to Card” option has been selected the player will be prompted to put in money, and display the equivalent token amount. Once the player has finalized the number of tokens they want to put on their card, they are prompted again to tap the Kiosk with their card. Lastly, when the “Display Card Information” option has been selected, players are prompted to tap the Kiosk with their card. After which, player information will be displayed.

When a player touches an arcade game, they are informed of how many tokens the game costs to play. The arcade game also prompts the player to tap their card if they would like to play. If a player taps their card again, they are asked if they would like to play the game. If a player accepts they are prompted to tap their card one last time in order to decrease the number of tokens on the card. After the game is over, the game displays how many tickets they won for that game and their new ticket value as well as their new token value after the price of the game has been paid.

When a player touches a Reward Center, the Reward Center displays all available prizes and their corresponding ticket costs. If a player picks a prize the Reward Center asks if the player would like to redeem the prize. If the player accepts, the Reward Center prompts the player to tap their card in order to validate they have enough tickets on their player account, and the staff are alerted to who won which prize.

3.3.1.2 Hardware Interfaces

The Kiosks, arcade games, and Reward Centers will all have touch screens for interfacing with the players. The staff backend will be a more traditional software interface using a keyboard and mouse setup. Additionally, the Kiosks will have hardware for accepting and dispensing money.

3.3.1.3 Communication Interfaces

All machines will be connected via an ethernet cable to the server backend. Additionally, this connectivity is what allows the Rewards Center to alert the staff machines of prize winnings.

3.3.2 Detailed Description of Functional Requirements

3.3.2.1 Template for describing functional requirements

Component Name:

- *Purpose:*
- *Inputs to the Component:*
- *Processing:*

- *Outputs:*

3.3.2.2 Functional requirements from 2.3

- Kiosks should be able to distribute RFID Cards with matching played ID's in the backend
- Kiosks should be able to add tokens to RFID Cards
- Kiosks should be able to display last played game, total number of tickets, and total number of tokens
- Arcade games should be able to decrease the number of tokens on a card
- Arcade games should be able to update the total number of tickets a player has
- Reward Centers should be able to decrease the total number of tickets a player has
- Reward Centers should be able to display available prizes with their matching ticket prices
- Reward Centers should be able to alert staff that a player wants a specific prize
- Staff should be able to view statistics that the backend provides such as most played game, and most popular days

3.3.2.2.1 Functional Requirement 2.3.1

Dispense RFID Cards:

- Purpose: A kiosk should be able to give a player an RFID card with no tokens on it that is tied to an entry in the backend.
- Inputs: A player pressing the "Dispense New RFID Card" button on the kiosk. The kiosk will also ask for a player name for display purposes.
- Processing: A random player ID is generated and a new entry with this player ID and player name is created in the backend. Lastly, the card will be wiped and start with 0 Tokens.
- Outputs: A functional RFID card with a starting token balance of 0

3.3.2.2.2 Functional requirement 2.3.2

Add tokens to card:

- Purpose: A kiosk should be able to add tokens to an RFID card.
- Inputs: A player pressing the "Add more Tokens to Card" button, and providing a payment method with sufficient funds. The player will also define the number of tokens they wish to add to their card.
- Processing: The kiosk will determine if the provided payment method has sufficient funds for the number of tokens the player has defined. If the funds are sufficient the kiosk will add this number of tokens to the card. If paper money is input into the kiosk, the sum of which is higher than the required amount to purchase the tokens, change will be dispensed from the kiosk.
- Outputs: The output result is a player's RFID card with the total number of tokens incremented by their transaction.

3.3.2.2.3 Functional requirement 2.3.3

Kiosk display info:

- Purpose: A kiosk should be able to display the total number of tokens on an RFID card, the total number of tickets the player has accrued, and the number of tickets the player has won playing their last game.
- Inputs: A player tapping their card and selecting “Display Card Information”.
- Processing: The kiosk will query the backend to find ticket and game information. The kiosk is able to read the number of tokens that are stored locally on the player’s card.
- Outputs: A display screen with ticket, token, and game information for the player.

3.3.2.2.4 Functional requirement 2.3.4

Arcade Game Payment:

- Purpose: An arcade game should be able to decrement the number of tokens on a card by the cost of gameplay
- Inputs: A player tapping an arcade game and selecting “Yes” when the game asks if the player would like to play the game at a given token price.
- Processing: The arcade game will update the entry, categorized by player ID, in the backend stating that this player has played this game, and will decrease the number of tokens on the card.
- Outputs: The player will be allowed to play an arcade game.

3.3.2.2.5 Functional requirement 2.3.5

Win Tickets:

- Purpose: An arcade game should provide tickets upon completion of an arcade game
- Inputs: A player who has successfully finished an arcade game without walking away
- Processing: The arcade game should query the backend to find out the current number of tickets, and add at least one ticket for playing the game.
- Outputs: The arcade game adds the tickets to the player’s account, and informs the player of how many tickets they won, and their new current ticket total.

3.3.2.2.6 Functional requirement 2.3.6

Reward Center Cost:

- Purpose: Reward Centers should be capable of decreasing the number of tickets a player has based upon the prize the player selects to purchase.
- Inputs: A player selecting a prize at the Reward Center.
- Processing: A Rewards Center should query the backend to see if the player has enough tickets for the prize they selected. If the player has enough tickets, then the Rewards Center should decrease the player’s total ticket count by that amount, and display the new total ticket amount to the player.
- Outputs: The player has less tickets in their account, and have been informed how many tickets they have left.

3.3.2.2.7 Functional requirement 2.3.7

Reward Center Display Prizes:

- Purpose: A Rewards Center should be able to display all available prizes with their corresponding ticket prices.
- Inputs: A player tapping their RFID card at a Rewards Center.
- Processing: The Rewards Center should query the backend to get a list of all prizes that are available with their respective prices and display this information to the player.
- Outputs: A display screen showing all available prizes with their corresponding ticket costs.

3.3.2.2.8 Functional requirement 2.3.8

Reward Center Alert Staff:

- Purpose: A Rewards Center should be able to alert the staff when a player redeems their tickets for a prize.
- Inputs: A player selecting a prize to spend their accrued tickets on.
- Processing: A Rewards Center should send a popup message to the staff's monitor detailing the prize requested, the player making the request, and the Rewards Center this request originated from.
- Outputs: A staff member should be informed of which Rewards Center sent the popup, which prize was won, and which player purchased the prize.

3.3.2.2.9 Functional requirement 2.3.9

View Staff Metrics:

- Purpose: The backend should let staff view certain metrics such as most popular game, and which games generate the most tickets.
- Inputs: A staff member clicking on "View Stats" on their backend computer.
- Processing: The backend will compile information for the staff member. Some metrics it will pay attention to is most popular games played, and total number of tickets awarded by an arcade game during its lifespan.
- Outputs: Arcade staff will be more aware of which arcade games are most popular, and which arcade games distribute the most amount of tickets.

3.3.3 Performance Requirements

- All machines, when dinged with an RFID Card, need to respond within three seconds.
- All Kiosk machines needs to have at least one RFID Card on hand, just in case a player wants a new card.
- At least one staff should be on hand at all times in order to maintain order in the arcade, as well as award prizes from the Reward Center
- There should always be at least one working Kiosk, Arcade game, and Rewards Center
- There should always be at least one available prize for redemption
- The backend should take three or less seconds to query

3.3.4 Quality Attributes

Kiosk machines must be 100% accurate when adding funds to an RFID card. RFID readers need to be 100% accurate when scanning an RFID card. The data stored on an RFID card should be entirely secure in the sense that a user cannot access another user's data without having their card. The backend should be able to maintain over a hundred cards at once without losing any data or modifying data under an incorrect player ID.

3.4 Other requirements

N/A

3.5 Changes to Requirements During Development

In an effort to focus on the user facing aspects of the system, some of the requirements originally laid out needed to be relaxed. Notably, Functional Requirement 2.3.8: Reward Center Alert Staff and Functional Requirement 2.3.9: View Staff Metrics needed to be relaxed. The focus was instead placed on the development of the elements of the system the players in the arcade will interact with, including the General Kiosk, the Rewards Center, and the Arcade Game interfaces. There were some developments completed on the two requirements which needed to be relaxed, shown through the Arcade Game Statistics display which is populated with information from the backend. This display allows the staff a view of the players in the arcade, the games these players have played, and the scores and tickets received at each game. Future development is intended to expand this functionality to better encompass the functionality desired by the arcade staff, and to display more completed data for the owners and managers of the arcade.

4. System Design

4.1 Introduction

4.1.1 Purpose of this Section

The objective of this section is to detail the design of the RFID Arcade that BCON is planning. This design specification will attempt to unify the development efforts towards one coherent design, and instruct the testers how to verify that the product behaves appropriately. This section will address how a game in the arcade should react to external events, the expected behavior of an RFID card, the many different ways that a kiosk is used, the functionality of the rewards center, and the purpose of the backend server.

4.1.2 Scope of the Development Project

The RFID Arcade is a series of networked machines communicating with a backend that will facilitate the playing of arcade games, ticket generation, and prize redemption. Arcade owners will be able to view statistics of most arcade games played as well as the most popular days for

each arcade game. Players will be issued an RFID card using an available kiosk by exchanging currency for tokens, which are necessary to play arcade games. The RFID card serves as a unique identifier for the player within the system. Players may elect to participate in games which publish statistics to the backend. Additionally, players will be able to redeem their tickets for prizes, which can be selected at a special kiosk. Kiosks will be able to display a player's total number of tokens, tickets available for redemption, and interesting gameplay statistics as customized by the client.

4.1.3 Definitions, Acronyms, and Abbreviations

See Section 7 below for a full glossary of terms used in this report.

4.1.4 References

See Section 8 below for a full list of references used in the creation of this report.

4.1.5 Major Software Requirements

- Kiosks shall be able to distribute new RFID cards to players with a new player ID requested from the backend and (if applicable) token balance.
- Kiosks shall be able to add tokens to RFID cards.
- Kiosks shall be able to display last played game, total number of tickets, and total number of tokens to a player.
- Arcade games shall be able to debit the number of tokens on a card by the amount required for a single round of gameplay.
- Arcade games shall be able to publish gameplay statistics to the backend.
- Rewards Centers shall be able to display a browsable prize catalog to the player.
- Rewards Centers shall be able to redeem a prize by debiting a player's ticket balance by the prize amount and request retrieval of the prize.
- Staff and players shall be able to see visually-friendly statistics such as top scorers and ticket earners.

4.1.6 Design Constraints, Limitations

4.1.6.1 Design Constraints

Due to the physically distributed nature of the components deployed throughout the arcade, each must be networked. Thus, a robust internal network is required within the arcade to ensure all components can securely and reliably communicate with one another. Ideally, this network should not have online access and be on a separate VLAN from other arcade components such that an intrusion could be controlled. As the backend serves as the central communication hub for the system, it should also be physically secured to prevent unauthorized access.

4.1.6.2 Limitations

The team is limited by the resources and time available for this project. Given the April 2019 deadline, the developers must prioritize basic functionality over lower-priority functionality to meet MVP. Due to the time limitation, there will likely not be a lot of quality assurance going into the product. There will be a brief testing period after the project is finished, but development is the major concern of the project. The team is also limited by budget. Due to the project being academic-related, the team will not be able to construct a real arcade with actual arcade games and offer legitimate prizes. Instead, there will be a simulation of an arcade for purposes of demoing the overall architecture.

4.1.7 Changes to requirements

There are no changes to any requirements at this time. (Some requirements have been relaxed since the SDS document was completed. A full listing of the relaxed requirements and modified design approach can be seen in the 3.5 Changes to Requirements During Development and 4.7 Changes to Design During Development sections, respectively.)

4.1.8 Overview of Section

The succeeding sections shall traverse the technical details of the system. The Data Design section will detail the specific data that are tracked within the system by category along with information about how each is represented in the backend. The System Architecture Description section will discuss more thoroughly each component within the system at an integration level. Each of these components will then have its individual implementation discussed in Detailed Description of Components. The Interface Design section will discuss more on the specific user interactions that will occur with each machine in the arcade, building on the previously-discussed technical details. The Software design goals will be outlined next, which corresponds to the non-functional requirements outlined in the SRS. Finally, the Appendices found below contain various diagrams modeling the system and events that may occur within.

4.2 Data Design

The backend is the centralized hub for storing data within the system. As such, it defines and enforces schemas to provide structure and context for data passed throughout the system. For speed and flexibility, MongoDB serves as the DBMS managed by the backend. It is scalable and robust in that clients who may wish to track more or less information will be supported, as adjusting schemas can be done with little code impact.

The data are grouped into the following categories, each of which has an individual schema which governs storage within its collection in the database. Types are one of the choices supported by the MongoDB specification.

4.2.1 Game Data

Property	Type	Required	Unique	Description
----------	------	----------	--------	-------------

_id	ObjectId	Yes	Yes	Generated ID from the database.
name	String	Yes	No	Display name within the arcade.
tokenCost	Number	Yes	No	Tokens required for a single game.
topPlayer	ObjectId	No	Yes	The ID of the top-scoring player.

4.2.2 Player Data

Property	Type	Required	Unique	Description
_id	ObjectId	Yes	Yes	Generated ID from the database.
firstName	String	Yes	No	First name of the player.
lastName	String	Yes	No	Last name of the player.
screenName	String	Yes	Yes	Screen name selected by the player for display within the arcade.
tickets	Number	Yes	No	Total ticket balance earned by the player.
gameStats	Array	No	No	Gameplay statistics (see below).

4.2.2.1 Game Statistics

The *gameStats* player property is an array of JSON objects published by an arcade game once a game has finished that contains the following:

Property	Type	Description
----------	------	-------------

gameId	ObjectId	ID of the game.
ticketsEarned	String	Total tickets earned by the player on this game.
gamesPlayed	String	Total number of games played by the player on this game.
highScore	String	High score of the player on this game.

There shall only be a single entry per game in this player property. The server will check to see if properties need to be updated once games publish data (i.e. an improved high score).

4.2.3 Prize Data

Property	Type	Required	Unique	Description
_id	ObjectId	Yes	Yes	Generated ID from the database.
name	String	Yes	No	Display name within the arcade.
description	String	No	No	Brief user-facing prize description.
ticketCost	Number	Yes	No	Tickets required for a single unit.
availableQuantity	Number	Yes	No	Number of available units.
image	Object	No	No	Prize image, consisting of a data buffer and content type.

4.3 System Architecture Description

4.3.1 Overview of Modules / Components

The Arcade system consists of 7 components that are interconnected to each other within the local network. The components are classified into 3 subsystems based on functionality that allows for high cohesion with low coupling. At least one instance of each component is required to realize the full functionality of the system, though there would likely be more. The System Architecture Diagram of Appendix F shows the following subsystems.

4.3.1.1 Playing Subsystem

The first component is the RFID card, which stores the unique player ID and token amount. This card is required to interact with games and kiosks throughout the arcade as a means of reliably identifying the player to each machine. RFID cards can be reused as long as a new player ID is issued to reinitialize it.

The second component is the RFID reader, a device responsible for reading and writing data to RFID cards. These shall be integrated into each game and kiosk to support user identification and (in the case of games) token reduction. The reader features an API that abstracts managing the data fields stored on the card.

The third component is the arcade game itself, which is the primary way the player will spend purchased tickets and collect tickets (if supported by the game). Games can take a wide range of shapes and sizes; this system is designed to support retrofitting onto any game with some small modifications.

Together, these three components represent the core functionality of the system: playing games at the arcade. The typical player workflow represents purchasing tokens and thus receiving an RFID card before proceeding to play any number of games.

4.3.1.2 Kiosk Subsystem

The fourth component is the Rewards Center, a specific type of kiosk that handles prize redemption functionality. It provides a user interface for browsing available prize inventory as requested from the backend. As it also has an integrated RFID reader, it supports players selecting prizes to use tickets on and as such notifying staff that a prize needs to be retrieved while also updating the information in the backend.

The fifth component is the general kiosk which serves all player needs for managing their RFID card and checking data. Namely, this includes issuing a new card, adding tokens to a card, checking token and ticket balances, and viewing selected gameplay statistics associated with the user. These kiosks are present with the mindset of eliminating staff as a human bottleneck within the system, such that players can use general kiosks at their convenience to complete these tasks.

As these two components represent the two specific types of kiosks, they comprise the kiosk subsystem.

4.3.1.3 Backend Subsystem

The sixth component is the server (or backend) that is the central data hub for the system. It manages the database used to store player, game, and prize information based on the defined data schemas and provides a set of supported requests for machines to use in order to create, read, update, and delete information as needed. All games within the arcade need to be manually registered and programmed with the backend to ensure the data is processed accurately.

The seventh component is the display which serves as a frontend for the server. It is most catered toward the needs of the arcade staff to display useful public-facing information. For example, it can be configured to display top-scoring players for the day or list any out-of-stock prizes in the catalog. This also inherently provides a platform for clients to run promotions based on the displayed data.

4.3.2 Structure and Relationships

Appendix B contains the Class Diagram intended to clearly illustrate the above definitions and connections among components. For each, attributes, operations, and multiplicities are also given to provide a visualization of the preceding descriptions.

4.4 Detailed Description of Components

4.4.1 Component Template Description

The following format shall be used to describe in detail each of the components.

Component Name: **Name of Component**

- *Subsystem:* Subsystem that the component is part of, as shown in the System Architecture diagram in Appendix F.
- *Purpose:* The specific requirements within the Software Requirements Specification [1] which this component satisfies. These are listed in the format FR X.X.X, and as outlined in the Software Requirements Specification [1].
- *Function:* How the component functions to achieve the purpose outlined in the previous field.
- *Subordinates / Modules used:* An outline of the various parts which make up the component.
- *Dependencies:* Which other components this component depends upon.
- *Resources:* The hardware and software resources this component needs to complete the purpose outlined in the Purpose field.

- *Processing*: Full descriptions of the supported functionality from a technical perspective, spanning applicable algorithms and logic.
- *Data*: Data the component is concerned with, as described in section 2.

4.4.2 Description of RFID Card

Component Name: **RFID Card**

- *Subsystem*: Playing Subsystem
- *Purpose*: FR 2.3.1: Dispense RFID Cards, FR 2.3.2: Add Tokens to Card, FR 2.3.3: Kiosk Display Info, FR 2.3.4: Arcade Game Payment
- *Function*:
 - FR 2.3.1: Dispense RFID Cards: The RFID card is able to store a certain amount of data in various fields. This data can be read by an RFID reader or overwritten by an RFID writer. The data on the arcade RFID cards will be split into the player ID field and the token balance field. In order to dispense a new blank RFID card from the kiosk, the RFID card must support the functionality to have its contents overwritten, setting the player ID field to a unique player ID and the token balance back to 0.
 - FR 2.3.2: Add Tokens to Card: Tokens are then added to the RFID card by incrementing the amount in the token balance field by the amount of tokens being added to the card at the kiosk.
 - FR 2.3.3: Kiosk Display Info: Once the data in the player ID and token balance fields is written, it is stored on the RFID card. Because of this, the data may be read from the card at a General Kiosk in order to display the token balance stored on the card, and the information tied to the player ID stored on the card.
 - FR 2.3.4: Arcade Game Payment: Tokens can also be removed from the RFID card by decrementing the amount in the token balance field by the amount of tokens being spent on playing an arcade game.
- *Subordinates / Modules used*: The RFID card component is made up of a single off-the-shelf RFID readable and writable RFID card.
- *Dependencies*: RFID Reader/Writer
- *Resources*: None
- *Processing*: When participating in read/write operations, the card must return/store the commanded bits on the card.
- *Data*: Stored data include the player ID, which is returned from the backend as a result of a player creation request and is persistent until deleted, and the token balance, an integer with an initial value of 0 that can be incremented by kiosks and decremented by games.

4.4.3 Description of RFID Reader

Component Name: **RFID Reader**

- *Subsystem*: Playing Subsystem

- *Purpose:* FR 2.3.1: Dispense RFID Cards, FR 2.3.2: Add Tokens to Card, FR 2.3.3: Kiosk Display Info, FR 2.3.4: Arcade Game Payment, FR 2.3.5: Win Tickets, FR 2.3.6: Rewards Center Cost
- *Function:*
 - FR 2.3.1: Dispense RFID Cards: The RFID writer is used to overwrite any data which may already be present on an RFID card, providing the user with an RFID card with a uniquely populated player ID field and a token balance field equal to 0.
 - FR 2.3.2: Add Tokens to Card: Whenever a user purchases additional tokens at a General Kiosk, the token balance field on the RFID card is overwritten by the RFID writer to contain the previous balance + the newly purchased tokens.
 - FR 2.3.3: Kiosk Display Info: The user may also have their card read at a General Kiosk in order to check the token balance on the card. In this case, the RFID reader will read the player ID and token balance fields on the card once the card is placed within 50 mm of the reader [2].
 - FR 2.3.4: Arcade Game Payment: In a similar fashion, once the user places their card within 50 mm of an RFID reader on an Arcade Game, the token balance field on the card will be read to determine if the card has enough tokens to play the game. If it does, the token balance field on the card will be decremented by the cost to play the game, by the RFID writer, and the game will begin.
 - FR 2.3.5: Win Tickets: At the beginning of the interaction, the user will have tapped their card at the Arcade Game, allowing the RFID reader on the game to read the player ID field on the card. The game can then use this player ID to update the number of tickets that player has won.
 - FR 2.3.6: Rewards Center Cost: Once a user taps their card at the Rewards Center, the RFID reader on the Rewards Center will read the player ID field on the card, and use this to query the backend for the number of tickets the player has won.
- *Subordinates / Modules used:* The RFID Reader/Writer component is made up of the hardware reader/writer itself and the API for interfacing with the reader/writer. The reader/writer is an off-the-shelf component with a USB interface. The API will be custom built to allow easy integration of the reader/writer to the General Kiosk, Arcade Game, and Rewards Center software.
- *Dependencies:* RFID Card, Kiosk/Game (to control it)
- *Resources:* Software libraries will be used to ease the development of the API, such as a library for interfacing with the USB device. As the system shall run under Linux, libNFC provides a set of functionalities for interacting with reader/writers like this model. There shall also be an enabled system service that supports these interactions at boot time.
- *Processing:* Read/write interactions with the card happen asynchronously, using interrupt-style event-driven behavior. When a card is brought within range of the reader, it comes out of its idle state and notifies the higher-level application that an card is detected. The program will then either request to read or write data on the card, which results in the respective API call to carry out the operation. Enumerated return values defined in a header file indicate return status of the operation. After completing an operation, the driver places the reader back into its idle state to await further events.

- *Data:* No data are stored on the reader/writer itself, but the API supports the scheme detailed for the RFID Card.

4.4.4 Description of Arcade Game

Component Name: **Arcade Game**

- *Subsystem:* Playing Subsystem
- *Purpose:* FR 2.3.4: Arcade Game Payment, FR 2.3.5: Win Tickets
- *Function:*
 - FR 2.3.4: Arcade Game Payment: Once the user places their RFID card within 50 mm of the RFID reader, the reader will recognize and read the contents of the card, and the RFID reader/writer API will send the player ID and the token balance contained on the card to the Arcade Game software. The Arcade Game software will first check if the token balance on the card is above the token cost to play the game. If it is, then the player ID will be stored for updating the backend later. Then the game will begin, and then user will play the game. If the token balance is insufficient, a message saying such will appear on the Arcade Game.
 - FR 2.3.5: Win Tickets: Once the user has finished playing the game, they will have received a number of tickets based on their performance in the game. This ticket amount will be displayed to the user. Then, using the stored player ID, this ticket amount, the player ID, and the game ID for this specific game will be sent to the backend for storing and updating the user's profile which corresponds to the player ID.
- *Subordinates / Modules used:* While the design of the game itself will vary depending on the specific type of game, this system will consist of a USB-connected RFID Reader/Writer, network card for handling the connection to the local network, and some user interface for guiding the user through starting a game and displaying statistics after.
- *Dependencies:* RFID Reader/Writer, Server (Backend)
- *Resources:* The hardware requirements of the game itself depend on the nature of its purpose. This should be separated from this system's implementation, however. The Game in the context of this system requires a USB port to connect the RFID Reader/Writer, a NIC to connect to the network (Ethernet preferred for stability), and a physical mounting location for the RFID Reader/Writer. At a minimum, the software requires a single-core CPU with 500MB of RAM to provide comfortable resource allocation space to carry out the above functionality.
- *Processing:* When a card is brought near the reader and it thus wakes from its idle state, the game UI welcomes the player. Using the API for the reader, the token balance is read from the card and compared to ensure that the player can afford the round. Simultaneously, the game requests player statistics from the backend with the read player ID from the card to display any relevant "welcome back" messages. After the player confirms and finishes gameplay, the game publishes statistics to the backend using the temporarily-stored player ID. After the request is served by the backend, the player ID is deleted from the game.

- *Data:* Concerned data for the game is described in section 2.1.

4.4.5 Description of Rewards Center

Component Name: **Rewards Center**

- *Subsystem:* Kiosk Subsystem
- *Purpose:* FR 2.3.6: Rewards Center Cost, FR 2.3.7: Rewards Center Display Prizes, FR 2.3.8: Rewards Center Alert Staff
- *Function:*
 - FR 2.3.6: Rewards Center Cost:
 - FR 2.3.7: Rewards Center Display Prizes:
 - FR 2.3.8: Rewards Center Alert Staff:
- *Subordinates / Modules used:* All common kiosk components are included (user interface, USB-connected RFID Reader/Writer, and network card for interconnection within the system). The accessory features offered are all software-related and consist namely of the prize catalog.
- *Dependencies:* RFID Reader/Writer, Server (Backend)
- *Resources:* All kiosks require a USB connection to the RFID Reader/Writer and a NIC for connecting to the network (Ethernet preferred). The user interface is preferably touchscreen for easier browsing of the prize catalog. At a minimum, the software requires a single-core CPU with 1GB of RAM to provide comfortable resource allocation space to carry out the above functionality.
- *Processing:* Similar to games, interaction with the system is initiated when a player brings a card in readable proximity of the reader. This triggers an event to be raised within the system, where the application software will call the reader's API function to read the player ID. Once retrieved, a backend request will be constructed for the player's profile as well as the available prize catalog. If applicable, images for the available prizes will be converted from their packed binary format in the backend to be displayed to the player. The player will see their ticket balance at the top of the screen. When a prize is selected, a request is made to the backend to redeem a prize, which decrements the stock of the prize by 1 and also the ticket balance of the player.
- *Data:* Concerned data for the Rewards Center is described in section 2.3, in addition to the retrieved player information from the backend as described in section 2.2.

4.4.6 Description of General Kiosk

Component Name: **General Kiosk**

- *Subsystem:* Kiosk Subsystem
- *Purpose:* FR 2.3.1: Dispense RFID Cards, FR 2.3.2: Add Tokens to Card, FR 2.3.3: Kiosk Display Info
- *Function:*
 - FR 2.3.1: Dispense RFID Cards
 - FR 2.3.2: Add Tokens to Card
 - FR 2.3.3: Kiosk Display Info

- *Subordinates / Modules used:* All common kiosk components are included (user interface, USB-connected RFID Reader/Writer, and network card for interconnection within the system). The general kiosk also features a payment terminal for accepting credit card and optionally cash payments to exchange for arcade tokens. This payment terminal is accompanied by a corresponding driver and API that allows ease of transaction processing.
- *Dependencies:* RFID Reader/Writer, Server (Backend)
- *Resources:* Same as Rewards Center.
- *Processing:* Similar to games, interaction with the system is initiated when a player brings a card in readable proximity of the reader. This triggers an event to be raised within the system, where the application software will call the reader's API function to read the player ID. Once retrieved, a backend request will be constructed for the player's profile. The menu of the user interface will then present the user with operations. Each will guide the user through the flow of events detailed by the Sequence Diagram found in Appendix C below, generally following the pattern of the Rewards Center. The General Kiosk makes similar backend requests to update player information based on the requested operation.
- *Data:* Concerned data for the General Kiosk is described in section 2.2.

4.4.7 Description of Server (Backend)

Component Name: **Server (Backend)**

- *Subsystem:* Backend Subsystem
- *Purpose:* FR 2.3.5: Win Tickets, FR 2.3.6: Rewards Center Cost, FR 2.3.7: Rewards Center Display Prizes, FR 2.3.9: View Staff Metrics
- *Function:*
 - FR 2.3.5: Win Tickets
 - FR 2.3.6: Rewards Center Cost
 - FR 2.3.7: Rewards Center Display Prizes
 - FR 2.3.9: View Staff Metrics
- *Subordinates / Modules used:* The server software is built on the Node.js framework, which must be installed on the target system. Furthermore, the server should be higher power as compared to kiosks to support scalable traffic loads in peak times of demand within the arcade. For example, a dual-core CPU with at least 2GB would be desirable. If the database is to be stored locally on the machine, MongoDB will also need to be installed. The technical documentation for the server designates how to deploy it on a machine.
- *Dependencies:* Arcade Game, Rewards Center, General Kiosk
- *Resources:* The server is the simplest from a hardware perspective. It needs nothing more than a basic runtime environment (see above, Linux preferred) with decent system specs. These are largely governed by the size of the arcade and the number of connected games and kiosks within the system, which would be planned at time of installing.
- *Processing:* The backend supports a variety of tasks and is designed to respond asynchronously to them. When idle, it consumes very few resources until another

machine issues a request or a periodic status message is pushed out. All requests take the form of standardized HTTP requests as detailed in the guide for using the backend. JSON is the language used across all requests to transfer data over the wire, as it is portable across platforms and the data are self-describing. Error messages that may result from a request will also comply to the JSON specification in order to provide simple context to the requester.

- *Data:* All supported data schemas that the backend manages are described in section 2.

4.4.8 Description of Display

Component Name: **Display**

- *Subsystem:* Backend Subsystem
- *Purpose:* FR 2.3.8: Rewards Center Alert Staff, FR 2.3.9: View Staff Metrics
- *Function:*
 - FR 2.3.8: Rewards Center Alert Staff:
 - FR 2.3.9: View Staff Metrics:
- *Subordinates / Modules used:*
- *Dependencies:* Server (Backend)
- *Resources:* The display consists of a small network-connected computer and a connected display. The size of the display should be large enough to allow easy visibility within the arcade. The computer does not require intensive hardware, as it simply accepts messages from the backend and can occasionally make requests. Thus, a single-core CPU with 500MB is more than adequate for this component.
- *Processing:* The display is primarily driven by the backend. In most cases, it should not need to request data. Rather, each request that the backend responds to should also push new player, game, and prize data to the front end, such that the frontend has an up-to-date representation of each data model. When a new message arrives to the frontend, it will update the displayed content according to a configuration file set up at time of installation, though this can be modified at any time.
- *Data:* Concerned data varies depending on the configuration, but in the extreme case can include everything stored by the backend (though this may become unwieldy in a practical sense).

4.5 Interface Design

In total, there are 3 subsystems: Playing, Kiosk, and Backend. The user primarily interacts with the Playing subsystem since this consists of the RFID Card, RFID Reader, and Arcade Game components. However, in order to put tokens on the RFID Card, a player must interact with the Kiosk subsystem since this subsystem holds the General Kiosk component. Additionally, the player will eventually interact with the Rewards Center which is also part of the Kiosk subsystem to redeem prizes. Both the Kiosk and Playing subsystems will interact with Backend subsystem since the Playing subsystem pushes and pulls information to and from the Backend, while the Kiosk subsystem performs similar operations.

The Backend subsystem is the centralized interface for communication within the system. All components that request or publish data must comply to the commonized message specification, which in this case is the standard HTTP protocol with message bodies being JSON-formatted. While the player does not see this communication, it is important to emphasize with respect to the interaction among components as well as subsystems.

4.6 Software Design Goals

The Design Goals for the RFID Arcade are as follows:

- **Responsiveness:** the RFID Arcade should be responsive by taking less than 3 seconds to recognize a RFID Card being tapped. Also any query to the backend should be completed within 3 seconds.
- **Security:** the RFID Arcade should be secure as information is stored on either the card, or the backend.
- **Connected:** The three different subsystems of the RFID Arcade (General Kiosk, Rewards Center, and Arcade Game) should be connected through the backend so that all the data generated in each of these subsystems can be collected and catalogued.
- **Scalable:** The RFID Arcade should be scalable as any number of players can send data to the backend simultaneously thanks to the robust backend. Additionally, setting up a new Kiosk or Arcade Game is as simple as connecting it to the network.

4.7 Changes to Design During Development

The only notable modification to the design originally outlined in the SDS document is in the specific implementation of the RFID readers/writers. It proved difficult to write to the RFID cards using the readers purchased. This necessitated storing the user's token amount in the backend, and strictly reading the player ID from the RFID card. Moving forward with the design allowed for the core functionality of the system to be implemented, without sacrificing any of the outlined design goals.

5. Test Document

To test our systems we utilized the bottom-up testing approach. We started at the bottom with the backend and the RFID readers. Without these nothing else would function. Then we moved to testing the arcade game and general kiosk respectively since they both require the backend and RFID readers to operate. We finished the testing with testing the rewards center since that requires the backend, RFID readers, as well as won tickets from an arcade game to work. Lastly, we tested the entire system as a whole. This bottom-up approach works best with real time systems and systems with strict performance requirements, so it seemed like the best approach.

To further expand on the testing involved, we performed unit tests on each component, then performed a system test on all of component at once. Functional tests were performed to iron out any "bugs" that were found. Performance testing was used to make sure everything satisfies the strict non-functional requirements. Lastly, we have performed a test

demonstration to validate that the entire system works as advertised, and the demonstration on April 12th will function as an acceptance test (making sure our clients like the end product).

6. User Manual

There are two types of people that will interact with this system: players and arcade staff. These two types of people interact with the system differently, as such they would have different instructions. The staff needs to understand each component of the RFID, how it works, and how to integrate it into their current arcade configuration, while the players just need to know how to interact with the new machines to play their games.

6.1 Staff User Manual

This system consists of five integrated parts: the arcade games, at least one General Kiosk, the Rewards Center, the server backend, and the RFID cards. The arcade games, General Kiosk, and Rewards Center are all connected to the backend at all times, while the RFID card is a standalone component.

The arcade staff will be able to view different statistics about the usage of the arcade on a monitor connected to the backend. This is also where they will receive updates that a customer needs to receive a prize at the Rewards Center.

The RFID cards will be distributed to the players using the General Kiosk. After which, the built in identifier on the card is attached to a user defined player ID. Once this new player ID is created, this change is sent to the backend whereupon a new entry is created for that player ID. This entry in the database then tracks the games a customer plays as well as keeping track of the total number of tickets a player has accumulated and the total number of tokens remaining. A player can put additional tokens on their card by using the General Kiosk again. In order to play an arcade game, a player must spend tokens by tapping the RFID card at the arcade game and accepting the payment of the tokens. After a player plays an arcade game, the arcade game will update the total ticket value stored under the customer's ID in the backend. At any point a player can tap their RFID card at a General Kiosk to see their information such as total number of tickets on their account, the number of tokens currently on the card, and other information such as recent high scores achieved. Players can tap their card at the Rewards Center to view potential prizes (based upon the number of tickets they have in their account), the ticket cost of each prize, and their current ticket balance. The Rewards Center is able to access and update all of this information on the backend.

To transform your arcade into a RFID arcade you would connect all your Kiosks and Arcade Games to the backend (which would be setup by us).

6.2 Player User Manual

When a player touches a Kiosk they should see the following options: "Dispense New RFID Card", "Add more Tokens to Card", and "Display Card Information". When the "Dispense New

RFID Card” option has been selected the player will be asked to type in a player name using the touchscreen keypad. After which a new RFID card will be dispensed, with the accompanying user entry in the server backend. When the “Add more Tokens to Card” option has been selected the player will be prompted to put in money, and display the equivalent token amount. Once the player has finalized the number of tokens they want to put on their card, they are prompted again to tap the Kiosk with their card. Lastly, when the “Display Card Information” option has been selected, players are prompted to tap the Kiosk with their card. After which, player information will be displayed.

When a player taps their RFID card at an arcade game, they are informed of how many tokens the game costs to play. The arcade game also prompts the player to tap their card if they would like to play. If a player taps their card again, they are asked if they would like to play the game. If a player accepts they are prompted to tap their card one last time in order to decrease the number of tokens on the card. After the game is over, the game displays how many tickets they won for that game and their new ticket value as well as their new token value after the price of the game has been paid.

When a player touches a Reward Center, the Reward Center displays all available prizes and their corresponding ticket costs. If a player picks a prize the Reward Center asks if the player would like to redeem the prize. If the player accepts, the Reward Center prompts the player to tap their card in order to validate they have enough tickets on their player account, and the staff are alerted to who won which prize.

7. Glossary

Term	Definition
Arcade Game	Machine offering gameplay, which require a set number of tokens for a round and may reward tickets to the player depending upon the design of the game.
Arcade Staff	People who staff the arcade, responsible for machine maintenance and distributing redeemed prizes.
Backend	Server responsible for managing the database storing player, game, and prize information as well as supporting requests to read, write, update, and delete data.
Kiosk	Machine players use to put tokens on their RFID card, view statistics on games played, and view token and ticket balances.
Player	A customer of the arcade which participates in offered services.

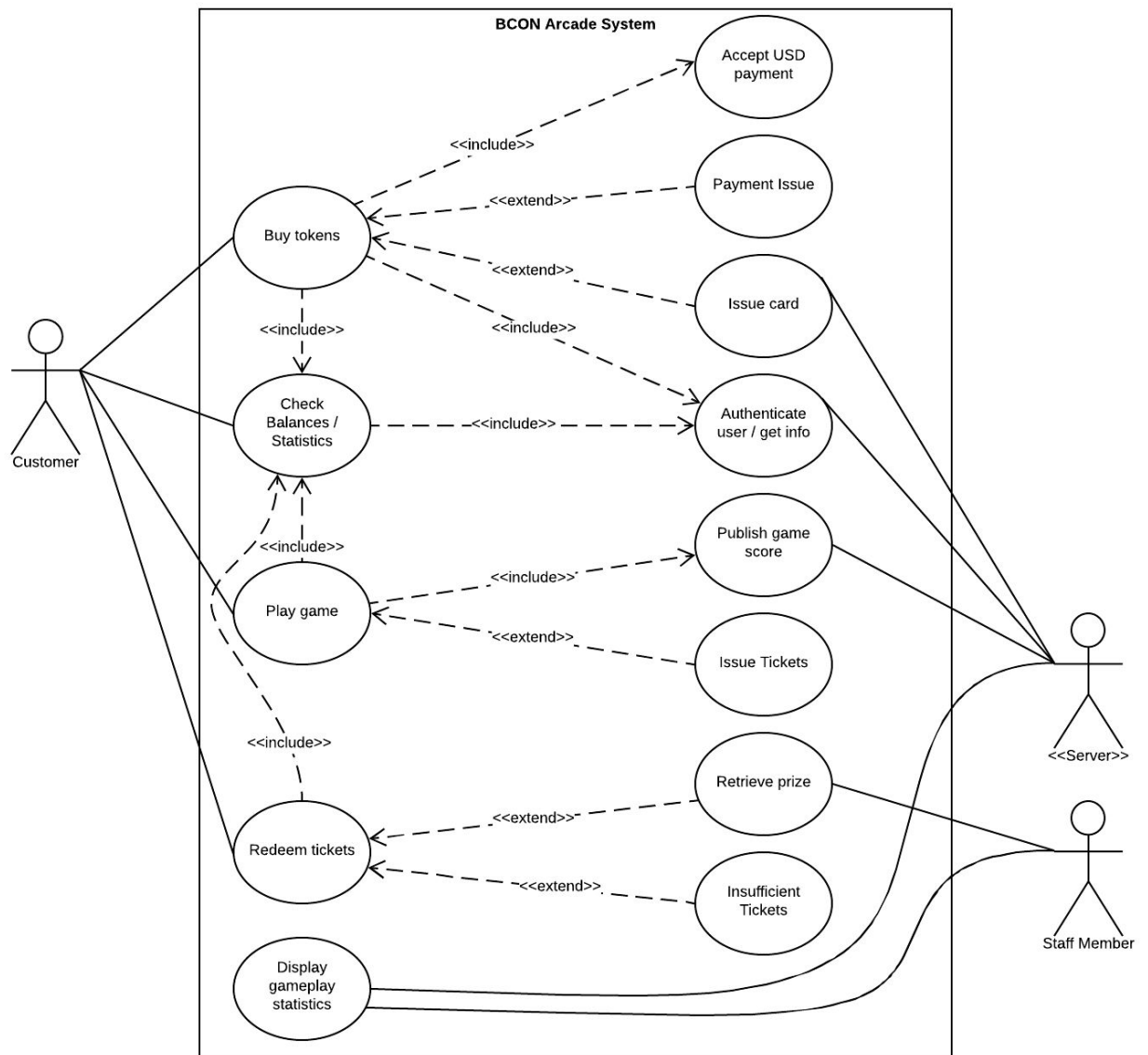
Player ID	The unique identifier for a player.
Prize	Products stocked by the arcade that can be redeemed by players in exchange for tickets.
Rewards Center	Special type of kiosk that allows a player to browse the prize catalog and redeem prizes.
RFID Card	Physical key identifying a player traveling throughout the arcade storing a player ID and token balance, used to interact with games and kiosks.
Tickets	Form of reward given to players by supported games used to enable the redemption of prizes at a Rewards Center.
Tokens	Internal arcade currency used to play games, purchased at a kiosk with standard money.

8. References

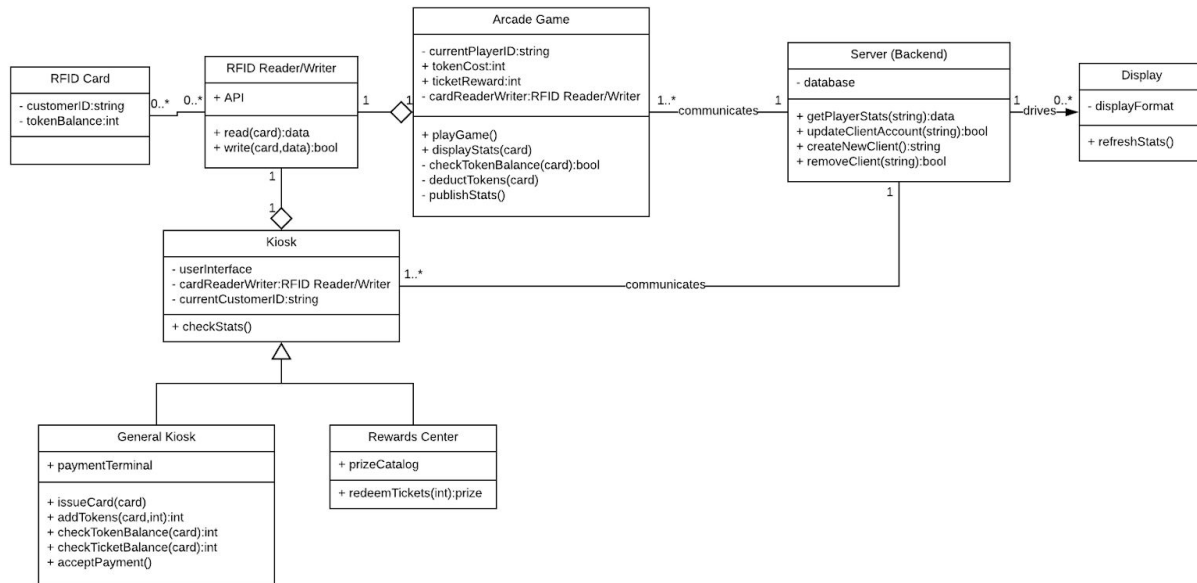
- [1] Prisby, Ben. Tuffaha, Omar. Sullo, Nicholas. & West, Connal. "Software Requirements Specification (SRS) RFID Arcade." BCON. 2/12/2019.
- [2] Prisby, Ben. Tuffaha, Omar. Sullo, Nicholas. & West, Connal. "Software Design Specification (SDS) RFID Arcade." BCON. 3/19/2019.
- [3] "ETEKJOY ACR122U NFC RFID 13.56MHz Contactless Smart Card Reader Writer w/USB Cable, SDK, 5X Writable IC Card." *Amazon*, Amazon, www.amazon.com/ETEKJOY-ACR122U-13-56MHz-Contactless-Writable/dp/B07FCLY4S9/ref=pd_ybh_a_1?_encoding=UTF8&psc=1&refRID=9AV7ZY0QPD5MCM4BF7PW.
- [4] "Cashless Card Arcade System Software." *Ideal Amusement Software*, Ideal Software Systems, idealamusementsoftware.com/cashless-card-arcade-system/.

9. Appendices

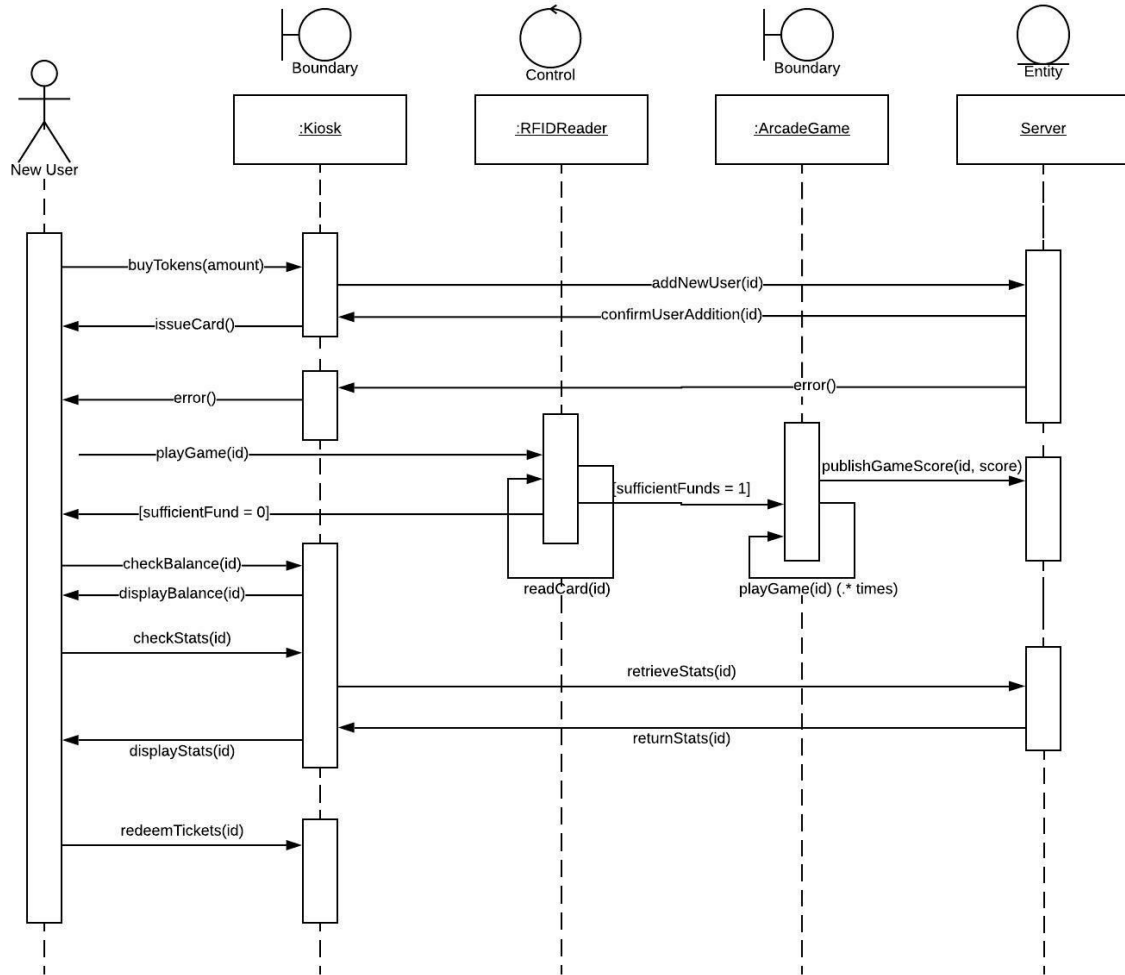
9.1 Appendix A: Use Case Diagram



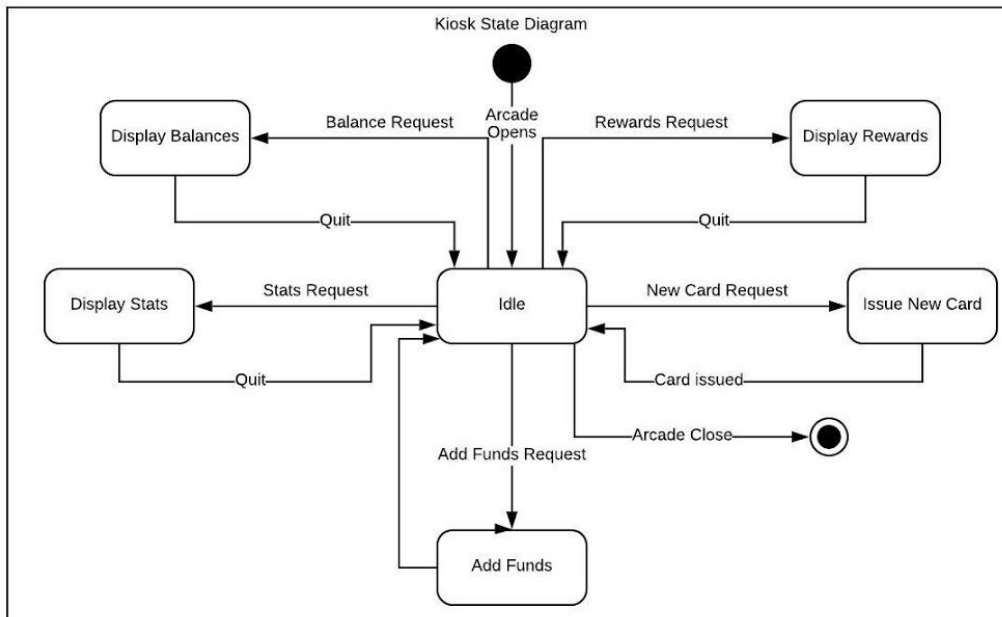
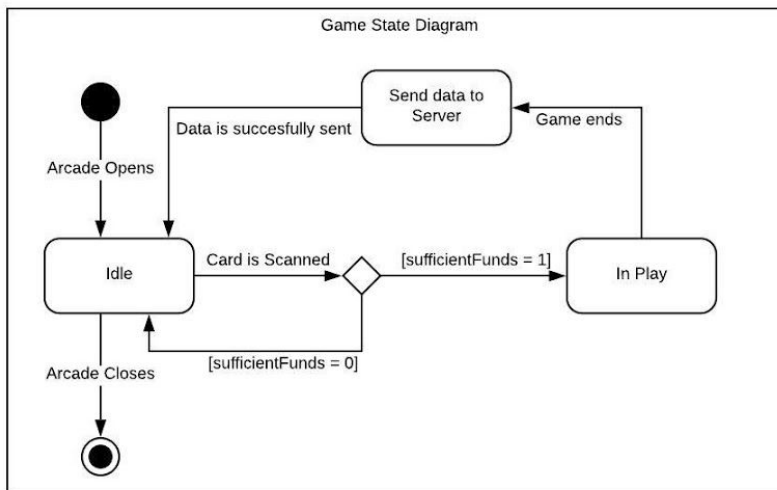
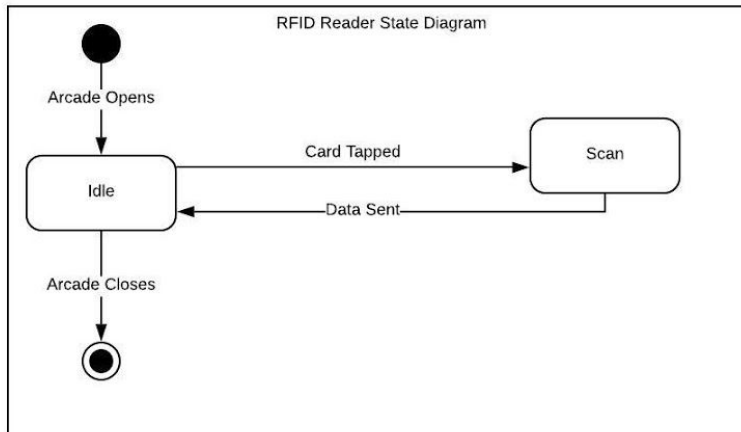
9.2 Appendix B: Class Diagram



9.3 Appendix C: Sequence Diagram



9.4 Appendix D: RFID Reader, Game, and Kiosk State Diagrams

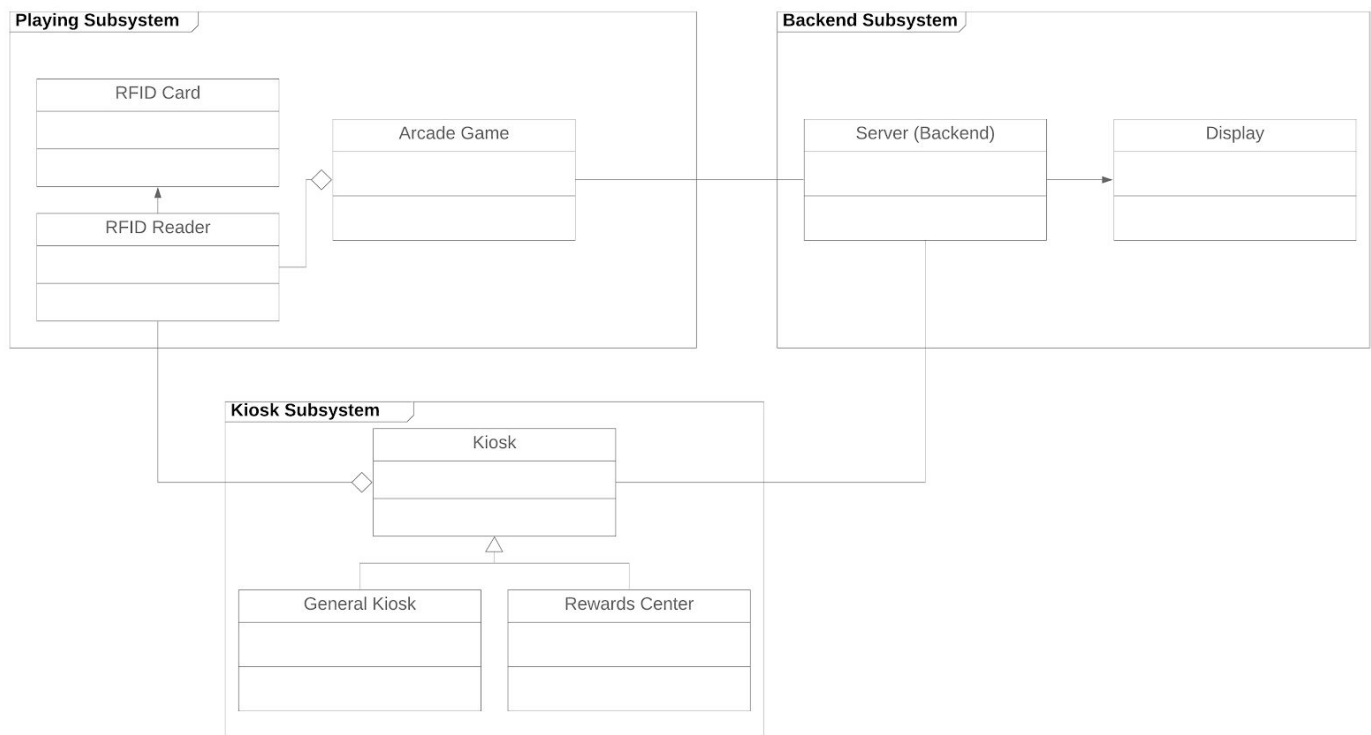


9.5 Appendix E: Scenario

Scenario

- A 12 year-old girl named Miley enters *Global Arcade* for the first time.
- She walks up to a Kiosk and registers herself for a player card, adding \$20 worth of tokens to her card.
- The first game that Miley plays is *skee-ball*, where she scans her card on the skee-ball card-reader. Here, she scores the highest score of the day!
- Next she plays *Deal or No Deal* and ended up with case number 14, containing 75 tickets!
- Miley forgets how many tokens she has left and returns to a kiosk where she scans her card and is shown her ticket and token balances. She has 0 tokens left.
- Miley ends her day at the arcade by redeeming her tickets for prizes.
- She also checks to see that she still has the highest *skee-ball* score of the day.

9.6 Appendix F: System Architecture Diagram (Subsystem interaction level)



9.7 Appendix G: System Architecture Diagram (Component level)

