Software Requirements Specification

for

Lottery Purchase System

Version 1.0

Prepared by

Group Name: Team 14

Rory Campbell	001	rorycampb@ttu.edu
Tyler Kuchta	001	tkuchta@ttu.edu
Samuel Riggan	001	samuel.riggan@ttu.edu
Simon Swopes	001	siswopes@ttu.edu

Date: November 04, 2023

Contents

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
0.1	Simon Swopes	The Introduction of the document has been written, updates will be required as subsequent sections are written.	10/31/23
0.2	Rory Campbell	The Overall Description of the document has been written, updates will be required as subsequent sections are written	11/02/23
0.3	Tyler Kuchta	The Sketch for the graphic	
0.4	Samuel Riggan	Listed potential performance and safety/security requirements that could arise in this system using information provided.	11/03/23

1 Introduction

1.1 Document Purpose

The Lottery Purchase System is a system designed for the Texas Lottery Commission. The system will be a mobile based application capable of tracking the purchase, redemption and management of lottery tickets. This document is intended to describe the system, its various functionalities and the external and internal requirements of the system.

1.2 Product Scope

The purpose of the Lottery Purchase System is to ease the process of purchasing and managing tickets for both users and the Texas Lottery Commission. The system is based on a relational database with its ticket ordering, redemption, and management functions. We will have a database server supporting thousands of users, including administrators, and ticket orders. There are 4 ticket types to choose from, but administrators shall have the ability to add, remove, or change the prices on existing tickets.

The system shall be able to communicate with the existing lottery system to update prize amounts, access the winning numbers for a drawing, and retailers for redemption. The system shall have the ability to email users if a winning ticket was purchased, and permit the user to redeem prizes valued at or under $599. The system shall support Paypal, Credit, or Debit cards for purchases and allow users to cash prizes to corresponding accounts. The system must adhere to the Americans with Disabilities Act and Federal Communication Commission government regulations. Lastly Users shall be able to display or print their purchased tickets for redemption at a lottery claim center.

1.3 Intended Audience

This is a project prototype for the Lottery Purchase system and it is restricted within the college premises. This has been implemented under the guidance of college professors. This project is useful for the Texas Lottery Commision, purchasers of lottery tickets, retailers who allow for redemption of these tickets, and Government agencies who handle the tax of lottery sales and redemption. This document will be useful to the developers, project managers, testers, and documentation writers of the Lottery Purchase System.

1.4 Definitions, Acronyms, and Abbreviations

Americans with Disabilities Act henceforth shall be referred to as ADA.

Federal Communications Commission henceforth shall be referred to as FCC.

Institute of Electrical and Electronic Engineers henceforth shall be referred to as IEEE.

Lottery Purchase System henceforth shall be referred to as LPS or as “The System.”

The Professor shall henceforth be referred to as the stakeholder

1.5 Document Conventions

This document shall follow IEEE formatting requirements. All sections shall be clearly defined at the top of their corresponding section and shall start on their own page. Section numbers will start at one and each subsequent section shall be one more than the previous section. All subsections are required to be in bold text and their numbers shall begin with the main section number followed by a decimal and the subsection number starting at one.

All text shall be Arial font. Section headings shall be size 18 bold in white text with a light grey banner around them. All subsection headings will be size 14 bold in black text, and all plain text shall be size 11 with black text. Comments shall be italicized and the document should maintain a consistent single spacing with one inch margins. All pages subsequent to the title page shall start with the name of the project in the top left of the header and the page number in the top right. Any revisions made to the document must be documented in the page following the title page. Any changes that affect page numbers must be updated in the table of contents. Lastly all external sources are required to be cited and follow IEEE conventions.

1.6 References and Acknowledgments

I. Sommerville, Software Engineering, 10th ed. Hallbergmoos/Germany: Pearson, 2018.

2 Overall Description

2.1 Product Overview

In this section, it is crucial that you will be creative and provide as much information as possible.

TO DO:
The LPS is a software system users shall access via smartphone to purchase and manage lottery tickets. Users will be able to view their ticket history as well as the winning numbers for each lottery. The system will primarily store/retrieve information from a private database in order to keep user credentials as well as their purchase history and winning numbers from each lottery. Other databases and software systems may be used to do things like connect to PayPal and record relevant security data within LPS.

Provide at least one paragraph describing the product perspective. Provide a ***context diagram*** *that will illustrate how your product interacts with the environment and in what context it is being used. This is not a formal diagram, but rather something that is used to illustrate the product at a high level.>*
Refer to in-class lectures and exercises we did in class. When drawing a context diagram, you can use PowerPoint, Word, or any other software to draw this diagram.

2.2 Product Functionality

TO DO:
1. Ability to buy and print tickets with as few clicks as possible
2. Ability to register or login to the Texas Lottery Commission system
3. Ability to show users their purchase history as well as any past winnings.
4. Ability to show users the winning ticket for the lottery they participated in.
5. Ability to update the lottery winnings in real time
6. Ability to allow admin users to track ticket sales and change ticket prices
7. Ability to allow admins to add or remove registered tickets

2.3 Operating Environment

The LPS will be primarily operating on iOS and Android operating systems. Because the software is dependent on using the Texas Lottery Commission database, it must also be able to communicate with external servers that will store user credentials as well as registered tickets. In order for users to purchase tickets through the LPS, the system must be able to store credit/debit card information and/or connect directly to PayPal. Modern smartphones can locally store card information, however, connecting to PayPal will require a built in API that will automatically send access tokens (usually in the form of JSON files) whenever the previous access token is timed out.

2.4 Design and Implementation Constraints

Some of you may have issues using certain databases, servers, banking systems, etc..

The LPS is an e-commerce system at its core. One of the main problems in managing such a system is making sure that the ticket purchase system is not being exploited by bots (*computer automated users*). This problem can be monitored by creating a personalized security system that records user activity on all parts of the LPS. This security system can use embedded JavaScript to place data such as user clicks, their IP address, email, country of origin, whether they solved their captcha challenge, etc. The database being used can be the main database used to store credentials, ticket purchases, etc., or with a database provided by a third-party security vendor. These vendors often provide their own proprietary software and data systems to monitor and enforce security.

Another requirement that may be needed is age verification. Users obviously need to be 18 in order to use the LPS, so a system to verify their legal age is absolutely necessary. The simplest verification would be to use the date of birth entered upon account registration. However, if a more valid authentication is required, the LPS may need to validate the user's government ID. Validating a user's ID would require using services like Plaid, a company that provides background checks and ID authentication. Services like this often do not need much programming from the customer's side, but the implementation of an API and/or subscribing to one of the company's payment plans may be necessary.

2.5 Assumptions and Dependencies

TO DO: Provide a short list of some major assumptions that might significantly affect your design.>

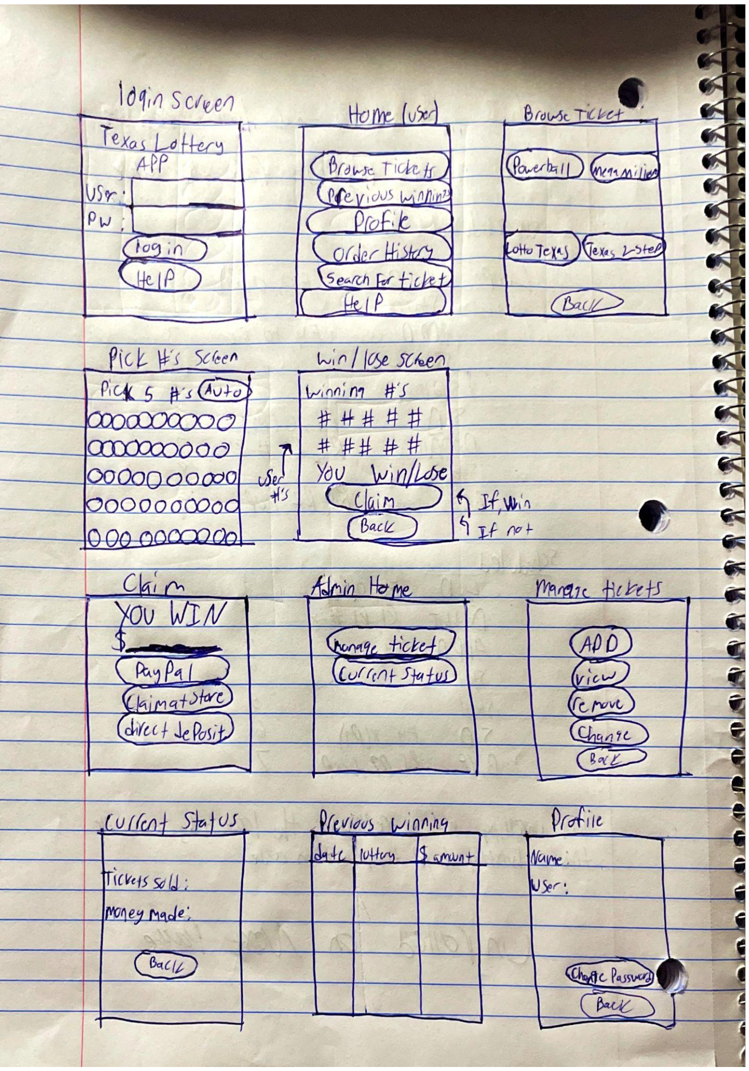
1. Assume that the LPS uses a pre existing database from the Texas Lottery Commission that can hold user credentials, their history of ticket purchases, the winning ticket of each lottery, and any other relevant information recorded in the LPS
2. Assume that there is not yet an e-commerce security system to enforce proper access of the LPS *(assuming the need to find a third-party to make one independently)*
3. Assume that the LPS shall be accessed by users on exclusively iOS or android devices *(there will be no website made for LPS)*
4. Assume that a web based system shall be required for admin users to perform tasks like adding/removing tickets or changing ticket prices.
5. Assume that there shall be only one winner for each lottery *(a system is needed to decide one winner in the case that there are multiple)*
6. *Assume that for every lottery ticket, there are only five numbers (there is no powerplay or special conditions)*

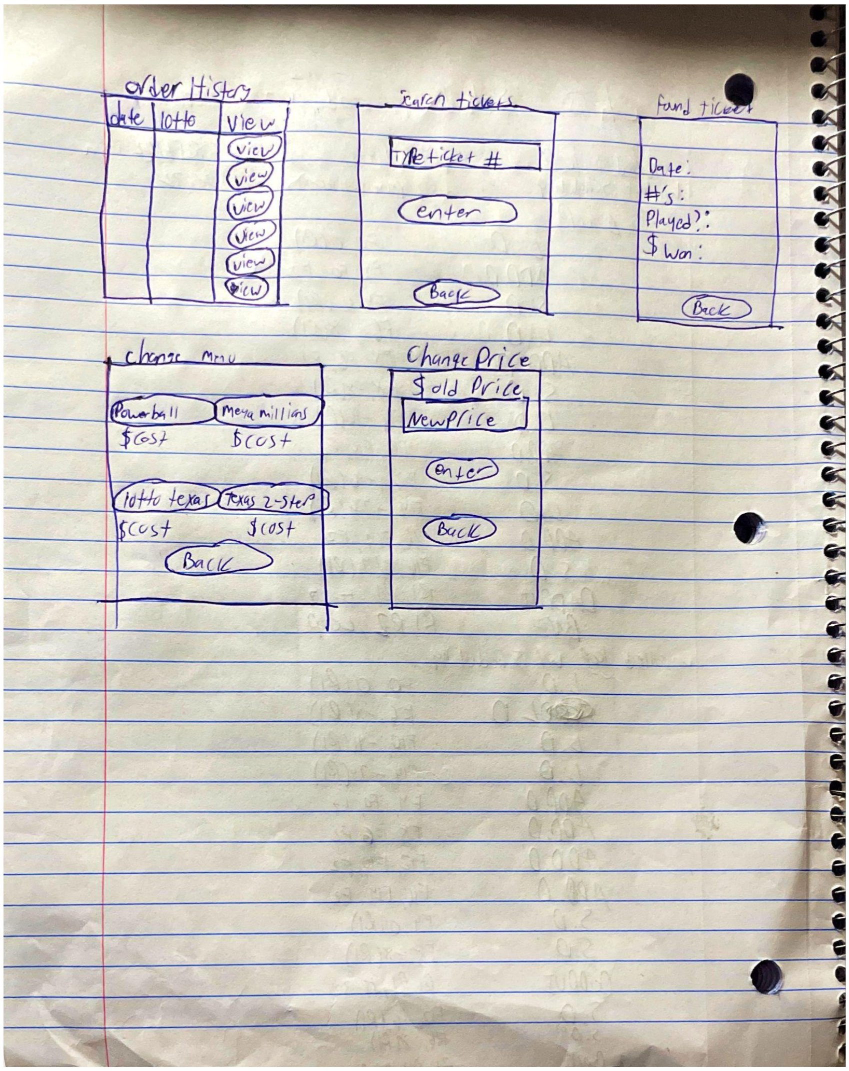
3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

TO DO: Provide the graphic for the SMS interface and provide a basic description as to how users will interact (e.g. tough screen, menus, etc.).>





The boxes above represent the screens you would see when using the app, the labels are above the boxes as a rough identifier of the boxes. The circles are touchable buttons that the user will be able to use to navigate through the app as they need, and the words you see are labels or placeholders to aid in instructing the user.

In the early stages, this could be something you would draw. A sketch.

3.1.2 Hardware Interfaces

The only devices that will be involved with this program are devices that run on the IOS or Android operating system and have access to the app store or android play store.

3.1.3 Software Interfaces

Ex: Following are the software used for the flight management online application. <<*Include the software details as per your project* >>

Software used	**Description**
Operating system	The lottery app can be supported by mobile IOS and Andriod operating systems.
Database	To save the account records of those who sign up, as well as to keep track of ticket records.

3.2 Functional Requirements

< Functional requirements capture the intended behavior of the system. This behavior may be expressed as services, tasks, or functions the system is required to perform. This section is the direct continuation of section 2.2 where you have specified the general functional requirements. Here, you should list in detail the different product functions.

3.2.1 F1: The program should ask the user to input their login info when the app is opened

F2: The program should present the user with a variety of options to choose from at the main menu including "Browse tickets, previous winnings, profile, order history, and search for tickets"

F3: The Browse tickets button should take you to a screen where you can pick which lottery you want to play, then pick your 5 numbers, check to see if you have winning numbers, and then collect your winnings.

F4: The previous winnings button should take the user to a page that shows them the date played, lottery they played, and amount won.

F5: The Profile button should take the user to a page that shows their basic information and they can change their password

F6: The Order History tab should take the user to a page that shows them all of their ticket purchases in the past with date. the lottery that was played and an option to view more details

F7: The search for ticket function should take you to a screen that allows you to put in a specific ticket number and search for it at anytime, showing you the date it was purchased, the numbers chosen on it, if its been played or not, and if it won how much $ the ticket won

F8: The Program should present a different home screen when an admin logs into the app. The different home screen should have "manage tickets and current status" as options to choose from.

F9: The manage tickets should bring the admin to a screen that has "Add, View, Remove" which allows them to do those actions on tickets.
F10: The current status option should bring the admin to a page that shows him the tickets sold and money made in real time

…

3.3 Use Case Model

3.3.1 Use Case #1 (use case name and a unique identifier – e.g. U1)

Use Case Name – User win

Summary – The user logs into the app, buys a ticket, wins, and collects their winnings

Actors – The actor is a human holding an iphone or android

Preconditions - The user is not logged in and has not bought a ticket

Main sequence – The user opens the app, logs in using their credentials, selects browse tickets, selects the lottery of their liking, picks the 5 numbers or auto generates them, goes to the win or lose screen to see if they won, if they won go on to the you win screen and select the method of collection you choose

Alternatives – The only alternative path would be selecting a different lottery game, the program would function the exact same regardless of the game your selected.

Postconditions - The user has logged in, purchased a ticket, picked their numbers, checked against the winning numbers, and has collected their winnings.

Author – Tyler Kuchta

3.3.2 **Use Case Name –** User loss

Summary – The user logs into the app, buys a ticket and loses

Actors – The actor is a human holding an iphone or android

Preconditions - The user is not logged in and has not bought a ticket

Main sequence – The user opens the app, logs in using their credentials, selects browse tickets, selects the lottery of their liking, picks the 5 numbers or auto generates them, goes to the win or lose screen to see if they won, if they lost then they can use the back button and return to the menu

Alternatives – The only alternative path would be selecting a different lottery game, the program would function the exact same regardless of the game your selected.

Postconditions - The user has logged in, purchased a ticket, picked their numbers, checked against the winning numbers, and has lost and returned to title screen

Author – Tyler Kuchta

…

3.3.3 **Use Case Name –** User wants to check previous winnings

Summary – The user logs into the app and selects previous winnings from the main menu

Actors – The actor is a human holding an iphone or android

Preconditions - The user is not logged in

Main sequence – The user opens the app, logs in using their credentials, selects previous winnings from the menu screen

Alternatives – There is no alternative path

Postconditions - The user has logged in and selected previous winnings option from the menu

Author – Tyler Kuchta

3.3.4 **Use Case Name –** User wants to check their profile

Summary – The user logs into the app and selects profile from the menu screen

Actors – The actor is a human holding an iphone or android

Preconditions - The user is not logged in

Main sequence – The user opens the app, logs in using their credentials, selects profile on the menu screen

Alternatives – There is no alternative path

Postconditions - The user has logged in and selected profile on the main screen and can now see the profile data

Author – Tyler Kuchta

3.3.5 **Use Case Name –** User wants to check order history

Summary – The user logs into the app and selects order history

Actors – The actor is a human holding an iphone or android

Preconditions - The user is not logged in

Main sequence – The user opens the app, logs in using their credentials, selects order history from the main menu

Alternatives – There is no alternative path

Postconditions - The user has logged in and selected the order history option on the main menu

Author – Tyler Kuchta

3.3.6 **Use Case Name –** User wants to search for a ticket

Summary – The user logs into the app and selects search for ticket from the main menu

Actors – The actor is a human holding an iphone or android

Preconditions - The user is not logged in

Main sequence – The user opens the app, logs in using their credentials, selects search for ticket at the main menu, input the ticket number into the search bar and hit enter, the next screen pulls up details about the ticket

Alternatives – There is no alternative path

Postconditions - The user has logged in and selected the search ticket option at the menu, and is now looking at the details of the ticket they were searching for

Author – Tyler Kuchta

3.3.7 **Use Case Name –** Admin wants to add a ticket

Summary – The admin logs into the app and selects manage tickets at the admin menu, then chooses which action he would like to perform upon the tickets

Actors – The actor is an admin holding an iphone or android

Preconditions - The admin is not logged in

Main sequence – The admin opens the app, logs in using their credentials, selects manage tickets at the menu, selects the add function at the manage ticket menu

Alternatives – There is no alternative path

Postconditions - The admin has logged in and selected manage tickets at the main menu and add at the managte tickets menu

Author – Tyler Kuchta

3.3.8 **Use Case Name –** Admin wants to add a ticket

Summary – The admin logs into the app and selects manage tickets at the admin menu, then chooses add at the manage tickets screen

Actors – The actor is an admin holding an iphone or android

Preconditions - The admin is not logged in

Main sequence – The admin opens the app, logs in using their credentials, selects manage tickets at the menu, selects the add function at the manage ticket menu

Alternatives – There is no alternative path

Postconditions - The admin has logged in and selected manage tickets at the main menu and add at the manage tickets menu

Author – Tyler Kuchta

3.3.9 **Use Case Name –** Admin wants to remove a ticket

Summary – The admin logs into the app and selects manage tickets at the admin menu, then chooses remove at the manage tickets screen

Actors – The actor is an admin holding an iphone or android

Preconditions - The admin is not logged in

Main sequence – The admin opens the app, logs in using their credentials, selects manage tickets at the menu, selects the remove function at the manage ticket menu

Alternatives – There is no alternative path

Postconditions - The admin has logged in and selected manage tickets at the main menu and remove at the manage tickets menu

Author – Tyler Kuchta

3.3.10 **Use Case Name –** Admin wants to view a ticket

Summary – The admin logs into the app and selects manage tickets at the admin menu, then chooses view at the manage tickets screen

Actors – The actor is an admin holding an iphone or android

Preconditions - The admin is not logged in

Main sequence – The admin opens the app, logs in using their credentials, selects manage tickets at the menu, selects the view function at the manage ticket menu

Alternatives – There is no alternative path

Postconditions - The admin has logged in and selected manage tickets at the main menu and view at the manage tickets menu

Author – Tyler Kuchta

3.3.11 **Use Case Name –** Admin wants to change the price of a ticket

Summary – The admin logs into the app and selects manage tickets at the admin menu, then chooses change at the manage tickets screen, selects the lottery ticket price he wishes to change, enters the amount desired

Actors – The actor is an admin holding an iphone or android

Preconditions - The admin is not logged in

Main sequence – The admin opens the app, logs in using their credentials, selects manage tickets at the menu, selects the change function at the manage ticket menu, he then chooses any of the lottery games, enters the new price of the ticket

Alternatives – The only alternative path would be selecting a different lottery game, the program would function the exact same regardless of the game your selected.

Postconditions - The admin has logged in and selected manage tickets at the main menu and change at the manage tickets menu, selected the ticket he wishes to change and his entered the desired amount.

Author – Tyler Kuchta

3.3.12 **Use Case Name –** Admin wants to view current status of the system

Summary – The admin logs into the app and selects current status at the menu

Actors – The actor is an admin holding an iphone or android

Preconditions - The admin is not logged in

Main sequence – The admin opens the app, logs in using their credentials, selects current status at the main menu, the page displayed will have the desired information.

Alternatives – There is no alternative path

Postconditions - The admin has logged in and selected cuurent status

Author – Tyler Kuchta

4 Other Non-functional Requirements

4.1 Performance Requirements

Performance requirements:

- Page load times:
  - All pages must load within 2 seconds.
  - Rationale: Quick page loading is essential to provide a responsive user experience and prevent users from becoming frustrated due to slow performance.

- Ticket Purchase Process:
  - The ticket purchase process should take no more than 4 seconds to complete.
  - Rationale: Fast ticket purchasing is critical to retain user engagement and satisfaction. Users should not experience delays during the transaction.
- Search ticket response:
  - The search for a specific ticket should return results within 3 seconds.
  - Rationale: Fast search response times are necessary to provide users with quick access to the tickets they are interested in.
- Email Notifications:
  - Email notifications for winnings must be sent within 5 minutes of a drawing.
  - Rationale: Timely notifications are crucial to inform users of their winnings promptly, maintaining trust and excitement.
- Database query performance:
  - Database queries for user information, ticket details, and order history should execute within 1 second.
  - Rationale: Efficient database performance ensures quick access to user data and reduces page load times.
- Concurrent User handling:
  - The system should support at least 1,000 concurrent users without degrading performance.
  - Rationale: To handle peak usage periods, the system must be capable of accommodating a substantial number of users simultaneously without slowdowns.
- Payment Processing:
  - Payment processing should not exceed 5 seconds for credit card, debit card, and PayPal transactions.
  - Rationale: Efficient payment processing is essential to prevent users from experiencing delays or timeouts during transactions.
- Scalability:
  - The system should be easily scalable to handle increased loads during special events or promotions without significant degradation in performance.
  - Rationale: Scalability ensures that the system can adapt to changing usage patterns and accommodate surges in user activity.
- Real-Time system(drawing days):
  - During drawing days, the system must ensure that ticket purchases, number matching, and notification processing are completed in real-time, with no more than a 2-second delay.
  - Rationale: Real-time performance is critical on drawing days to maintain the integrity of the lottery and provide immediate feedback to users.
- Failover and Redundancy:
  - The system should have a failover and redundancy mechanism to ensure continuous operation even in the event of server failures.
  - Rationale: Redundancy is essential to maintain system availability and minimize downtime in case of unexpected server failures.

4.2 Safety and Security Requirements

Safety and Security Requirements:

- Payment Information Protection:

  - The system must ensure the protection of users' payment information, including credit card and debit card details, to prevent potential financial loss or fraud.
  - Safegaurd: Employ industry-standard encryption for secure transmission of payment data. Comply with the industry security standard for handling credit card information.
- User Authentication Security
  - User authentication mechanisms must be robust to prevent unauthorized access and protect user data.
  - Safegaurd: Implement secure password storage, enforce strong password policies, and provide multi-factor authentication options for added security.
- Data Privacy Compliance:
  - Ensure compliance with data privacy regulations to protect user privacy and prevent legal issues.
  - Safegaurd: implement data protection measures, allow users to manage their data, and provide clear privacy policies and consent mechanisms.
- User Identify Authentication:
  - Users must provide accurate identification during registration to prevent fraudulent activities.
  - Safegaurd: Verify user identity during registration through email confirmation or identity verification services, if required by regulations.
- Security Certifications:
  - The system must obtain necessary security certifications to ensure that it adheres to industry best practices and standards.
  - Safegaurd: Obtain certifications such as ISO 27001 or other relevant security certifications to demonstrate the system's commitment to security.
- Access Control:
  - Limit access to sensitive areas of the system (e.g., admin functions) to authorized personnel only to prevent misuse or unauthorized changes.
  - Safegaurd: Implement role-based access control (RBAC) and strong authentication for administrators. Restrict access to admin functionalities to authorized users.
- User Data Protection:
  - All user data, including personal information, must be securely stored, and access should be restricted to authorized personnel.
  - Safegaurd: Encrypt user data at rest and in transit. Implement access controls and audit logs to track data access.
- Consent for Data usage:
  - Users must provide clear consent for the collection and use of their data.
  - Safegaurd: Implement opt-in mechanisms for data collections, with easy to understand forms

5 Other Requirements

<This section is **Optional**. *Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.*>

Appendix A - Group Log

<Please include here all the minutes from your group meetings, your group activities, and any other relevant information that will assist in determining the effort put forth to produce this document. Everyone in the group needs to do an equal amount of work to receive an equal grade.>

(Note: Whenever you have a question about the project, bring your question to my attention after the class rather than emailing).