

Software Requirements Specification Template

Software Engineering

The following annotated template shall be used to complete the Software Requirements Specification (SRS) assignment.

Template Usage:

Text contained within angle brackets ('<', '>') shall be replaced by your project-specific information and/or details. For example, <Project Name> will be replaced with either 'Smart Home' or 'Sensor Network'.

Italicized text is included to briefly annotate the purpose of each section within this template. This text should not appear in the final version of your submitted SRS.

This cover page is not a part of the final template and should be removed before your SRS is submitted.

<Movie Theater Ticketing System>

Software Requirements Specification

<Version 1>

<7/15/2025>

<No group>
<Sebastian Curran>

Prepared for
CS 250- Introduction to Software Systems
Instructor: Gus Hanna, Ph.D.
Fall 2023

<Project Name>

Revision History

Date	Description	Author	Comments
<date>	<Version 1>	<Sebastian Curran>	<First Revision>

Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Signature	Printed Name	Title	Date
	<Sebastian Curran>	Software Eng.	
	Dr. Gus Hanna	Instructor, CS 250	

Table of Contents

REVISION HISTORY	II
DOCUMENT APPROVAL	II
1. INTRODUCTION.....	1
1.1 PURPOSE.....	1
1.2 SCOPE.....	1
1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS.....	1
1.4 REFERENCES.....	1
1.5 OVERVIEW	2
2. GENERAL DESCRIPTION	2
2.1 PRODUCT PERSPECTIVE.....	2
2.2 PRODUCT FUNCTIONS	2
2.3 USER CHARACTERISTICS.....	2
2.4 GENERAL CONSTRAINTS.....	3
2.5 ASSUMPTIONS AND DEPENDENCIES.....	3
3. SPECIFIC REQUIREMENTS.....	3
3.1 EXTERNAL INTERFACE REQUIREMENTS.....	4
3.1.1 <i>User Interfaces</i>	4
3.1.2 <i>Hardware Interfaces</i>	4
3.1.3 <i>Software Interfaces</i>	4
3.1.4 <i>Communications Interfaces</i>	4
3.2 FUNCTIONAL REQUIREMENTS.....	4
3.2.1 < <i>Functional Requirement or Feature #1</i> >	4
3.2.2 < <i>Functional Requirement or Feature #2</i> >	5
3.3 USE CASES.....	5
3.3.1 <i>Use Case #1</i>	5
3.3.2 <i>Use Case #2</i>	6
3.4 CLASSES / OBJECTS.....	6
3.4.1 < <i>Class / Object #1</i> >	6
3.4.2 < <i>Class / Object #2</i> >	6
3.5 NON-FUNCTIONAL REQUIREMENTS.....	7
3.5.1 <i>Performance</i>	7
3.5.2 <i>Reliability</i>	7
3.5.3 <i>Availability</i>	7
3.5.4 <i>Security</i>	7
3.5.5 <i>Maintainability</i>	7
3.5.6 <i>Portability</i>	7
3.6 INVERSE REQUIREMENTS.....	8
3.7 DESIGN CONSTRAINTS	8
3.8 LOGICAL DATABASE REQUIREMENTS.....	8
3.9 OTHER REQUIREMENTS	8
4. ANALYSIS MODELS	8
4.1 SEQUENCE DIAGRAMS	8
4.3 DATA FLOW DIAGRAMS (DFD).....	8
4.2 STATE-TRANSITION DIAGRAMS (STD)	9
5. CHANGE MANAGEMENT PROCESS	10
A. APPENDICES	10
A.1 APPENDIX 1.....	10
A.2 APPENDIX 2.....	10

<Project Name>

1. Introduction

The introduction to the Software Requirement Specification (SRS) document should provide an overview of the complete SRS document. While writing this document please remember that this document should contain all of the information needed by a software engineer to adequately design and implement the software product described by the requirements listed in this document. (Note: the following subsection annotations are largely taken from the IEEE Guide to SRS).

1.1 Purpose

The purpose of this Software Requirement Specification is to determine and define the functional and non-functional requirements needed for a web-based movie theater ticketing system. The intent of this document is to serve as a guide for the development team, investors, and the testing team to ensure they all have an understanding of the project's goals.

1.2 Scope

The software product we are intending to design is a Movie Theater Ticketing System. This application will be web-based and allows users to browse, select, and purchase movie tickets and serves multiple locations. This system will support the creation of accounts, seat selections, digital ticket delivery, and discount handling. This system will operate using a web browser of the users choosing and will have no mobile app support for now. This app is not intended to support internal management such as staffing, operations, or concession sales. The goal of this app is to streamline the ticket purchasing process for users while also allowing staff to manage pricing, showtimes, and transactions. This system will also provide access for administrators to override transactions, resolve ticket issues, and generate reports.

1.3 Definitions, Acronyms, and Abbreviations

SRS: Software Requirements Specification. This document which outlines the requirements of this system.

UI: User Interface. This is the visual part which users interact with.

QR Code: Quick Response Code. A machine readable code which is used for digital tickets.

NFT: Non-fungible Token. Unique digital asset which protects against duplication and guarantees ticket authenticity

HTTP: HyperText Transfer Protocol. Used for secure communication between the browser and web server.

IEEE: Institute of Electrical and Electronics Engineers

API: Application Programming Interface

1.4 References

1. IEEE Std 830-1998 – IEEE Recommended Practice for Software Requirements Specifications. Institute of Electrical and Electronics Engineers (IEEE), 1998.
Available at: <https://ieeexplore.ieee.org/document/720574>
2. CS 250 Lecture Slides – Lecture 1–3: Introduction, Life Cycle Models, Requirements Gathering.

San Diego State University, Dr. Gus Hanna, Summer 2025.

Provided through Canvas course site: <https://sdsu.instructure.com>

3. Cornell University Use Case Lecture Notes

William Y. Arms, CS 5150 Software Engineering.

Used to guide use case structure and format. Provided in PDF format: "lecture-use-cases.pdf"

1.5 Overview

The remaining parts of this Software Requirements Specification outline the requirements and structure of the Movie Theater Ticketing System. Section 2 outlines provides a general overview of the system including its main features, user demographics, and functionality. Section 3 contains the specific requirements of the system including the functional requirements and nonfunctional requirements. It also outlines thorough use cases and business rules. The document follows the IEEE SRS format.

2. General Description

This section of the SRS should describe the general factors that affect 'the product and its requirements. It should be made clear that this section does not state specific requirements; it only makes those requirements easier to understand.

2.1 Product Perspective

The Movie Theater Ticketing System is a standalone web based application which is intended to replace traditional in person ticket purchases. Previous manual box office system and desktop apps are inconvenient. This system will provide real time constantly updated ticket availability, seat selection, and automatic digital ticket delivery. This system will also be able to handle payments in foreign currency. This application will also integrate with third party services to process payments, user authentication, and movie metadata. This system runs independently but may share data with other existing theater management platforms.

2.2 Product Functions

The Movie Theater Ticketing System will have the following functions,

- Allow users to search for movies, showtimes, and theater locations
- Support for account creation and users to manage their profile
- Enable users to select seats and ticket quantities
- Process ticket purchases with support for credit card, PayPal, Bitcoin or Apple Pay payments
- Deliver unique tickets via email and store in user accounts
- Apply applicable discounts for students, military, etc....
- Allow administrators to manage theaters, showtimes, pricing, and perform transaction overrides
- Integrate with third party sources to display ratings and reviews of movies
- Enforce timeout periods, ticket limits, and employ anti-bot measures to combat abuse.

2.3 User Characteristics

The Movie Theater Ticketing System will be used by a wide variety of users whose technical expertise vary. General users will use this system either through the web or at an in person kiosk at a theater to browse showtimes and purchase tickets. These users may create accounts or continue as guests. Returning users are users who created accounts and want to manage payment

information, view purchases, use loyalty points, and apply discounts. Theater staff will also use the system to resolve ticket issues and help customers in person with their issues. Administrators are users with the most knowledge of the system and have elevated privileges. They will manage theater settings, showtimes, pricing, and access system reports. Since the technological expertise of users vary, its crucial to use a user friendly UI.

2.4 General Constraints

The Movie Theater Ticketing System will be a web based application only, with no mobile application support. This application must support up to 10 million concurrent users and have a maximum of 20 tickets per purchase. Users have a window of 10 minutes to complete their purchase. The system must support English, Spanish, and Swedish with support for other languages in future versions. Processing for payments is limited to PayPal, Apple Pay, Bitcoin and credit or debit cards.

2.5 Assumptions and Dependencies

This Movie Theater Ticketing system makes the assumption that users have access to a web browser and a stable internet connection. The system also depends on third party services for payment processing, email ticket delivery, and data from movie reviews. The system also assumes that all theaters using this application will follow a standard format for displaying movie showtime and pricing data. The systems behavior and requirements may need to be updated if these services become unavailable or change.

3. Specific Requirements

This will be the largest and most important section of the SRS. The customer requirements will be embodied within Section 2, but this section will give the D-requirements that are used to guide the project's software design, implementation, and testing.

Each requirement in this section should be:

- *Correct*
- *Traceable (both forward and backward to prior/future artifacts)*
- *Unambiguous*
- *Verifiable (i.e., testable)*
- *Prioritized (with respect to importance and/or stability)*
- *Complete*
- *Consistent*
- *Uniquely identifiable (usually via numbering like 3.4.5.6)*

Attention should be paid to the carefully organize the requirements presented in this section so that they may easily accessed and understood. Furthermore, this SRS is not the software design document, therefore one should avoid the tendency to over-constrain (and therefore design) the software project within this SRS.

3.1 External Interface Requirements

3.1.1 User Interfaces

This application is intended to be used through a web browser or an in person kiosk. The user interface will allow users to browse movies, select seats, buy tickets, and also manage their account information. Its important that the UI is simple and easy to follow due to the varying technological knowledge of our users. The kiosk will be a limited scaled down version intended to streamline the in person ticket purchasing process. Both the kiosk and web version will support multiple languages and be accessible.

3.1.2 Hardware Interfaces

This system is intended to connect with physical devices such as QR code scanners, ticket printers, and in person kiosks. These devices will be plugged into local computers and kiosk systems using either standard USB for in person uses, or wireless connections.

3.1.3 Software Interfaces

This system will use external services to handle payments such as Bitcoin, PayPal, Apple Pay, or credit/debit cards. It will also connect to services like IMDb or Rotten Tomatoes for movie review data. The system will also generate and deliver each digital ticket as an NFT in order to ensure authenticity and prevent duplication. It will also be connected to a database that will store user data, purchase history, showtimes, along with other necessary information.

3.1.4 Communications Interfaces

Communication between the users browser, server, and any other third party services will use HTTPS for secure use. This system will rely on the use of APIs for communication with payment services, movie data sites, and email services for digital tickets and other communications.

3.2 Functional Requirements

This section describes specific features of the software project. If desired, some requirements may be specified in the use-case format and listed in the Use Cases Section.

3.2.1 <Purchase Tickets>

3.2.1.1 Introduction

The goal of this feature is to allow users to purchase one to twenty movie tickets through their web browser or in person kiosk. This process includes the user selecting a movie, choosing their preferred showtime, selecting seats and quantity, and a payment through the users preferred method.

3.2.1.2 Inputs

To begin their purchase, the user must make a movie selection, select theater, and showtime. If the theater supports assigned seating, then the user will be able to select seats as well. For all other purchases they can select the quantity. They also need to input their preferred payment method such as Bitcoin, PayPal, Apple Pay, or a credit/debit card. A user may also select whether they want to login to their account or continue as a guest. Either way they must enter an email for ticket delivery.

3.2.1.3 Processing

After the user enters their ticket selections and payment information, the system will check seat availability and place a hold on the seats until the payment goes through. The payment will be processed through a third party provider. Once the payment is approved, the system secures the

seats and tickets and will generate a NFT based digital ticket and the user will get an email confirmation along with the ticket.

3.2.1.4 Outputs

Once the payment goes through and tickets are secured, the system will display a confirmation page with the details of the purchase as well as sending the user a QR code ticket. If the user has an existing account and purchased their ticket while logged in, the ticket will be stored in their account for future use.

3.2.1.5 Error Handling

If payment fails to go through, the user is notified that their payment was unsuccessful and will have a chance to retry. If the seats become unavailable during checkout, the user will be prompted to select different seats. If the hold on the seats runs out before the purchase can be completed, the seats are released and the user will have to restart from the beginning. Any invalid inputs during checkout will trigger an error message to the user explaining the cause of the issue.

3.2.2 < Display Movie Listings and Showtimes >

3.2.2.1 Introduction

This application feature will allow users to view a complete list of available movies and their showtimes for all supported theater locations. They will also be able to narrow it down to see showtimes and movies for a specific theater instead of all locations. This will be one of the first interactions users will have with our system.

3.2.2.2 Inputs

The user has the option to filter the movie selection by location, date, time, or specific movie. If none of these filters are applied, then the system displays all movies for all locations supported. The user will also be able to sort by alphabetical order or by popularity of movie.

3.2.2.3 Processing

The system will pull from its internal database to display all currently showing movies, showtimes, theaters, and available seats left. If filters are selected by the user, the system will narrow it down.

3.2.2.4 Outputs

The intended output is a list of movies currently in theaters, with showtimes displayed, theater locations, and seats remaining which the user can browse through. This list will be displayed on the homepage by default and also accessible through the search interface. This 1

3.2.2.5 Error Handling

If the system fails to display movie data, an error message is displayed on the screen and users will be prompted to refresh the screen or try again later. If no movies are available for the users selected filters, then a message indicating that no results were found will be displayed.

...

3.3 Use Cases

3.3.1 Use Case #1: Purchase Tickets

Actor: Customer

Entry Conditions:

Customer has access to internet or access to a kiosk

Seats are available for selected movie and showtime

Flow of Events:

1. Customer accesses system through the website or kiosk
2. Customer selects preferred movie theater location, movie, and showtime

<Project Name>

3. System displays seats available and ticket quantity options
4. Customer chooses seat or number of tickets being purchased
5. Reservation timer initiated to hold selected seats
6. Customer enters payment information and email for digital tickets
7. Payment sent to third party for processing
8. When payment goes through, a unique NFT based ticket is created
9. Ticket is then sent to user via email and also stored in account if there is one made
10. System logs purchase for record keeping

Exit Conditions:

Ticket is issued to user

Internal database updates with seat assignments

3.3.2 Use Case #2: Display Movie Listings and Showtimes

Actor: Customer

Entry Conditions:

Customer has access to internet or access to a kiosk

Movie showtime data is available

Flow of Events:

1. Customer opens system through website or kiosk
2. System displays movies currently in theaters
3. Customer may apply filters if they wish
4. Movie selection refreshes based on applied filter
5. For each movie, the system displays title, poster, rating, showtimes, and ticket availability.
6. Customer will select a movie for purchase or to view more details

Exit Conditions:

Movies and showtimes are displayed

Customer selects movie and starts purchase process

...

3.4 Classes / Objects

3.4.1 <Ticket>

3.4.1.1 Attributes

ticketID

showtimeID

seatNumber

customerEmail

price

QRCode

isUsed

3.4.1.2 Functions

generateQRCode()

generateNFT()

3.4.2 <Movie >

3.4.2.1 Attributes

movieName
genre
rating
runTime
ShowTime
3.4.2.2 Functions
getRating()
getShowtimes()
getGenre()

...

3.5 Non-Functional Requirements

Non-functional requirements may exist for the following attributes. Often these requirements must be achieved at a system-wide level rather than at a unit level. State the requirements in the following sections in measurable terms (e.g., 95% of transaction shall be processed in less than a second, system downtime may not exceed 1 minute per day, > 30 day MTBF value, etc).

3.5.1 Performance

The Movie Theater Ticketing System must be able to handle 10 million users at the same time without any significant delays. The system should ensure that 95% of all purchases are able to be completed in under 2 seconds or less. Seat and movie listings must be displayed to user in 1 second.

3.5.2 Reliability

The ticketing system must maintain 99% uptime between the hours of 7AM-12AM daily. No more than 1 out of every 1000 ticket purchases may fail due to an error in the system. Any failures in the system should be recorded and followed up on.

3.5.3 Availability

This system shall be available to users 24/7 with the only exception being scheduled maintenance. Total downtime should not exceed an hour a month. Users will be notified of any scheduled downtime when they access the site.

3.5.4 Security

Any data sent between the client and the server must be encrypted. Users passwords must be hashed and salted using industry standard algorithms. Digital tickets will be issued as unique NFTs in order to prevent duplication or fraud. The system will also use CAPTCHA or rate-limiting to prevent automated abuse during checkout.

3.5.5 Maintainability

The code database must be well documented and modular so software updates can be implemented in a timely manner. Records of all errors will be retained for 90 days to support review and debugging.

3.5.6 Portability

The system must be able to run on any web browser such as Chrome, Microsoft Edge, Safari and also operate on Windows, macOS, and Linux operating systems. It must also be able to support touch screens for in person kiosks.

3.6 Inverse Requirements

State any *useful* inverse requirements.

3.7 Design Constraints

Specify design constraints imposed by other standards, company policies, hardware limitation, etc. that will impact this software project.

3.8 Logical Database Requirements

Will a database be used? If so, what logical requirements exist for data formats, storage capabilities, data retention, data integrity, etc.

3.9 Other Requirements

Catchall section for any additional requirements.

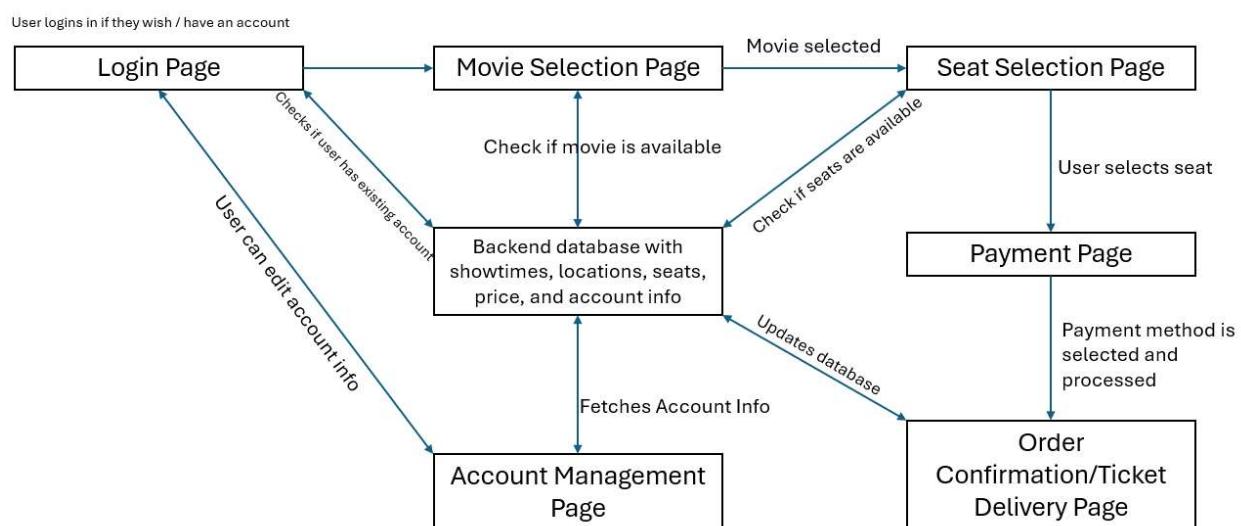
4. Analysis Models

List all analysis models used in developing specific requirements previously given in this SRS. Each model should include an introduction and a narrative description. Furthermore, each model should be traceable the SRS's requirements.

4.1 Software Architecture Overview

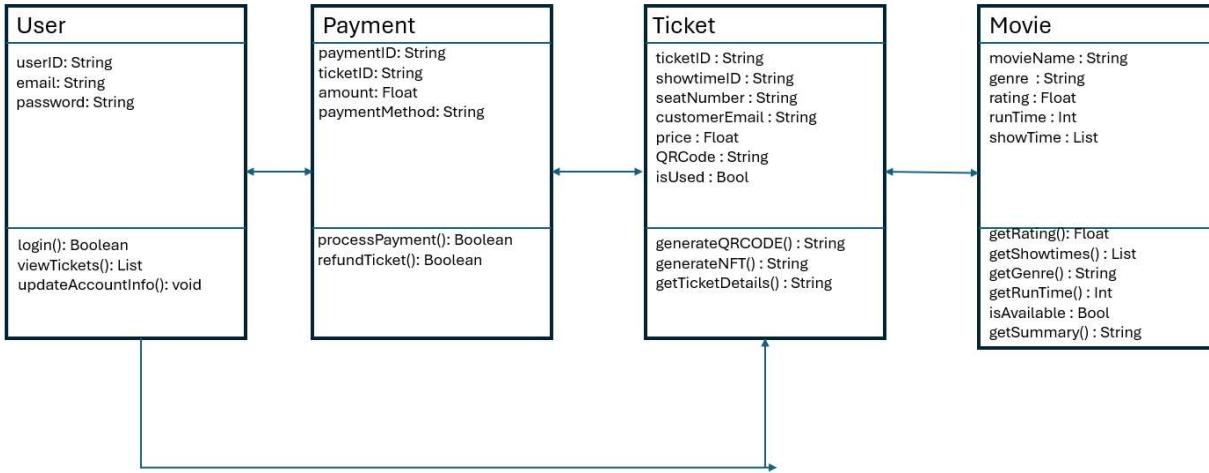
This subsection lays out the basic architecture of the Movie Theater Ticketing System.

System Architecture



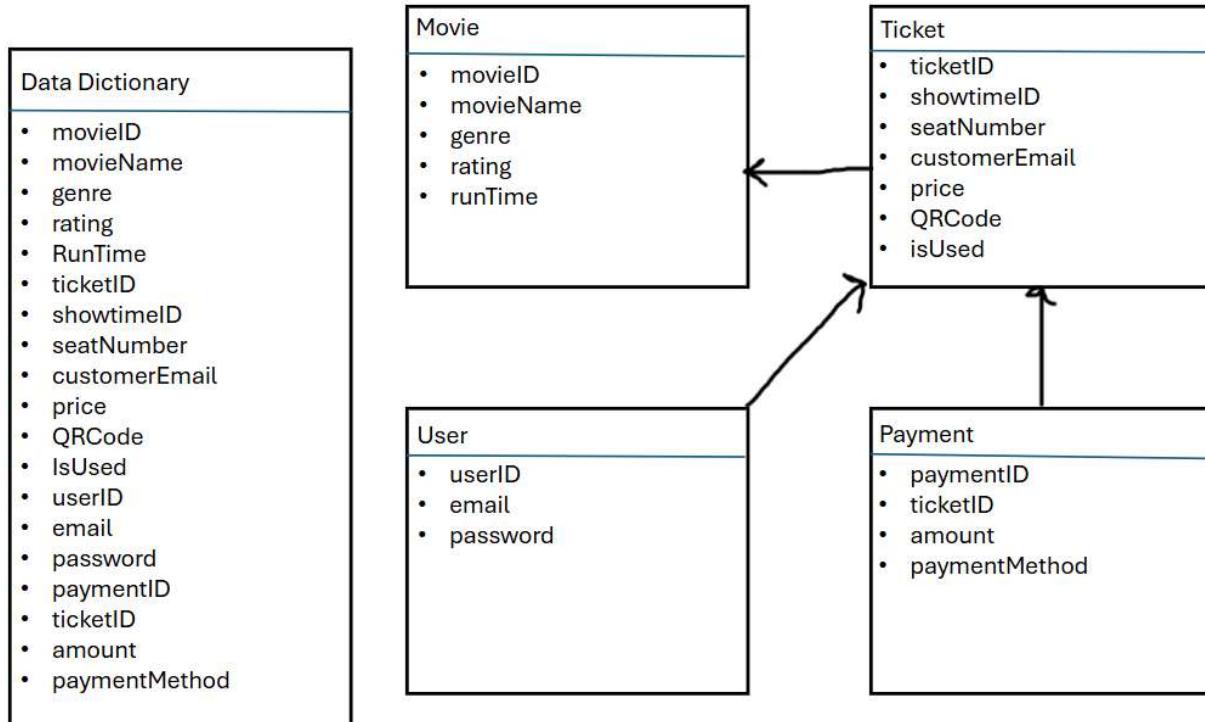
4.2 UML Class Diagram and Class Descriptions

This subsection will provide visual diagrams of four major system classes including User, Ticket, Payment, and Movie. Each class contains the necessary functions and attributes to carry out needed tasks.



4.3 Entity Relationship Diagram

This subsection will show an Entity Relationship diagram of basic but important entities of the Movie Theater Ticketing system and how they connect to each other.



This diagram shows the relationship between the foundation entities of this system. Each user can purchase one or more movie tickets, then the transaction is tied to a payment and movie.

4.4 Development Plan and Timeline

This Movie Theater Ticketing System will be completed incrementally with each increment meant to design a piece of the system. As an individual developer, every task will be handled by me personally but will take a modular and testable approach. The Requirement Gathering stage was laid out in Sections 1-3 of the SRS and was done in 2 weeks. It describes the systems goals, use cases, and requirements. The Design Phase is laid out in Section 4 and was done in 1 week.

In this phase I created the architecture diagram, UML diagram, and entity relationship diagram. Week 4 and beyond will be the testing and implementation stages of designing this system.

5. Change Management Process

Identify and describe the process that will be used to update the SRS, as needed, when project scope or requirements change. Who can submit changes and by what means, and how will these changes be approved.

A. Appendices

Appendices may be used to provide additional (and hopefully helpful) information. If present, the SRS should explicitly state whether the information contained within an appendix is to be considered as a part of the SRS's overall set of requirements.

Example Appendices could include (initial) conceptual documents for the software project, marketing materials, minutes of meetings with the customer(s), etc.

A.1 Appendix 1

A.2 Appendix 2