

Megarama Theaters

Software Requirements Specification

Version 1

02/14/2024

Group #3

Kenzie Kerrigan, Luis Alvarez, Mohamed
Boughou

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

Revision History

Date	Description	Author	Comments
2/7/24	Version 1	Group 3	First Revision
2/28/24	Version 2	Group 3	Second Revision

Document Approval

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

Signature	Printed Name	Title	Date
	<Your Name>	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.....	1
2. GENERAL DESCRIPTION.....	2
2.1 PRODUCT PERSPECTIVE.....	2
2.2 PRODUCT FUNCTIONS.....	2
2.3 USER CHARACTERISTICS.....	2
2.4 GENERAL CONSTRAINTS.....	2
2.5 ASSUMPTIONS AND DEPENDENCIES.....	2
3. SPECIFIC REQUIREMENTS.....	2
3.1 EXTERNAL INTERFACE REQUIREMENTS.....	3
3.1.1 <i>User Interfaces</i>	3
3.1.2 <i>Hardware Interfaces</i>	3
3.1.3 <i>Software Interfaces</i>	3
3.1.4 <i>Communications Interfaces</i>	3
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>	4
3.2.3 <i><Functional Requirement or Feature #3></i>	4
3.3 USE CASES.....	5
3.3.1 <i>Use Case #1</i>	3
3.3.2 <i>Use Case #2</i>	3
3.4 CLASSES / OBJECTS.....	3
3.4.1 <i><Class / Object #1></i>	3
3.4.2 <i><Class / Object #2></i>	3
3.4.3 <i><Class / Object #3></i>	3
3.4.4 <i><Class / Object #4></i>	3
3.5 NON-FUNCTIONAL REQUIREMENTS.....	4
3.5.1 <i>Performance</i>	4
3.5.2 <i>Reliability</i>	4
3.5.3 <i>Availability</i>	4
3.5.4 <i>Security</i>	4
3.5.5 <i>Maintainability</i>	4
3.5.6 <i>Portability</i>	4
3.6 INVERSE REQUIREMENTS.....	4
3.7 DESIGN CONSTRAINTS.....	4
3.8 LOGICAL DATABASE REQUIREMENTS.....	4
3.9 OTHER REQUIREMENTS.....	4
4. ANALYSIS MODELS.....	4
4.1 SEQUENCE DIAGRAMS.....	5
4.3 DATA FLOW DIAGRAMS (DFD).....	5
4.2 STATE-TRANSITION DIAGRAMS (STD).....	5
5. CHANGE MANAGEMENT PROCESS.....	5
A. APPENDICES.....	5

A.1 APPENDIX 1..... 5

A.2 APPENDIX 2..... 5

1. Introduction

The introduction to the Software Requirement Specification (SRS) document provides an overview of the complete SRS document, including the purpose, scope, definitions, acronyms, abbreviations, references, and an overview. The detailed requirements of the Megarama Theaters are provided in this document.

1.1 Purpose

The purpose of this document is to provide a detailed description of the software and its specifications and parameters. This document will contain ideas including how the system is supposed to work and goals that the client has about the system. It will also include how the user interacts with the system and ideas that are both used and discarded when developing and testing the system of how it can be more efficient and easy to use for the audience.

1.2 Scope

Principally, this scope is to be applicable to the features of the development of "Megarama" project of a Ticketing System with Live Data Manager. It emphasizes primarily on versatility, stability and live data updates for management. Ticketing system will provide a convenient web based interface easy-to-use for theater users and staff to be able to purchase and handle tickets for available films on specific times thru various locations. Alongside, will have a datacenter serving to provide live data on purchase products, tendencies and transactions system for management. This software will facilitate the clients ticket purchase experience to be quick and accurate, also to staff to be able to assist clients easily in any transaction and management to have live data form a datacenter given in simple readable reports provided when requested in supporting direction of business interests. The purpose of this software is to provide a uniform ticketing system to provide structured services to clients and assist with future business plan management. This SRS goes along with the specified requirements given to provide the developed software with the main purpose of the use in mind, also to have a model to follow for the specific standard use in conjunction with assessment of future development of this software.

1.3 Definitions, Acronyms, and Abbreviations

<i>TS</i>	<i>Ticketing System</i>
<i>LDF</i>	<i>Live Data Feed</i>
<i>DBMS</i>	<i>Database Management System</i>
<i>CRM</i>	<i>Customer Relationship Management</i>
<i>GUI</i>	<i>Graphical User Interface</i>

1.4 References

Fundamentals of Data Center

https://lsi.vc.ehu.eus/pablogn/docencia/AS/Act1%20Sysadmin%20y%20%20CPD/Act1B%20CPD/Fundamentals_of_Data_Centre_Part_1.pdf

Cinema Reservation System Essentials

<https://filmgrail.com/blog/cinema-reservation-system-essentials/>

1.5 Overview

The rest of this document provides a general description of the product and its requirements.

The second section discusses general information about the product, the functions that the software will perform, characteristics of the eventual users of the product, constraints, and assumptions and dependencies that affect the requirements of the SRS. The next section includes the specific functional and non-functional requirements, use cases, classes/objects, design constraints, and logical database requirements. Section 4 is for analysis models.

Finally, section 5 covers the process of updating the SRS.

2. General Description

This section of the SRS describes the general factors that affect the product and its requirements. It includes brief descriptions of the product, its functions, and the characteristics of the users that will interact with it in the future. It also describes the general constraints, assumptions, and dependencies of the product.

2.1 Product Perspective

In this section we will look at the product from the market point of view, the whole business aspect of the product. We will also look at the product through the customers and look at what needs to be improved in the updates for example why customers do not do not result in the purchase, and finally we will give all the information on the product, for example the number of tickets sold, the turnover and the profits after having accounted for all the expenses

2.2 Product Functions

The system will allow users to browse different movies and their showtimes at the 20 different Megarama theaters located in the San Diego area. They can see what room each movie is playing in for each theater. They can purchase movie tickets online on any browser. The user will be able to purchase up to 20 tickets at a time, and they must choose the seat(s), both in the regular and deluxe theaters (deluxe theaters have reclining chairs and tickets are more expensive). If the user leaves tickets in their basket for 10 minutes without checking out, the user will have to go and reselect their seats. Users will get their tickets online through an email that they enter. They will be able to get a refund on tickets purchased before the movie starts online or in person. If the theater is sold out of tickets, the user will get options for other show times or other theaters nearby that have available seats. They also have options for memberships to the theaters that

give them discounts on ticket and food prices. This software system will be quicker and easier for users to navigate.

2.3 User Characteristics

This represents almost all the people who will use and have access to your ticketing system. Typically this will be the role of each person working in the ticketing system (staff, manager, etc.) and any additional information they seek to add. This part is important for the design of the ticketing system and is user-friendly for everyone who uses it.

2.4 General Constraints

This is all the constraints that the user faces: This means the SKUs in the system, debugging, or just people trying to break into the system. This remains a large part where only the system developers have access to.

2.5 Assumptions and Dependencies

This system is dependent on any operating system that can run web based applications using java script, also running on systems capable to run a web browser without interruption that will be used at kiosk in theaters. The assumption that 50,000 live users will be exchanging data with the server at the same time.

3. Specific Requirements

This section covers the specific requirements of the system. It contains both the functional and non-functional requirements. There are also use cases that describe how users will be able to interact with the system. This will help guide the project's software design, implementation, and testing. The requirements in this section are correct, traceable, unambiguous, verifiable, prioritized, complete, consistent, and uniquely identifiable.

3.1 External Interface Requirements

3.1.1 User Interfaces

The user interface should be compatible with any Internet browser such as Safari or Google Chrome.

The user interface shall be a GUI that allows the user to click on icons and menus to interact with the system.

3.1.2 Hardware Interfaces

Hardware interfaces of the system include hardware that connects to the Internet, such as printers, WiFi routers, etc.

3.1.3 Software Interfaces

1. The system shall be developed with Java and use the Linux operating system.

2. The system shall communicate with the content management system to update the website with current promotions.
3. The system shall communicate with Stripe Payments, a secure payment processing platform to accept online payments.
4. The system shall communicate with the CRM system Oracle NetSuite to enhance the user's experience.

3.1.4 Communications Interfaces

The system shall use HTTP protocol for data transfer over the web. This includes the transfer of e-tickets through e-mail.

3.2 Functional Requirements

The specific features on this software system are the use of databases to be updated with the latest films, digital media imagery for viewing on webpage, and real time ticket purchase transactions. Secondly, this system will be able to collect data upon request to show tendencies and be able to make management decisions. The system as well has the feature of having direct calling to bank databases to make transactions reliable and be able to revert the process if needed.

3.2.1 <Functional Requirement or Feature #1>

3.2.1.1 Introduction

- A GUI specific for users will be shown in the main page of the web page, in which will be able to choose to log in or proceed as a guest, with live view of featured films playing, displaying imagery of films and being able to purchase a viewing ticket on the page.

3.2.1.2 Inputs

- Inputs will be login information, selection of film with time stamp, quantity of tickets to purchase and for transaction inputs are full information of users card number, date expiration, name on account card and CVV code on the card.

3.2.1.3 Processing

- Webpage will call DBMS for loading website packages to display media, add queue to ticket purchase in the TS and find available seats and will have data transfer to banks database for purchase efficiency.

3.2.1.4 Outputs

- Information given to the user will be purchased ticket information and purchase confirmation.

3.2.1.5 Error Handling

- In case of a bad display the website will auto refresh and send a report to DBMS and in case of a failed transaction bank will be notified. Users can call tech support for help on purchase or trained on site staff.

3.2.2 <Functional Requirement or Feature #2>

3.2.2.1 Introduction

- A GUI specific for management will be shown in the webpage as they first enter their credentials in the log in section in the webpage will they will be able to request live data on purchase and tendencies.

3.2.2.2 Inputs

- Inputs are login credentials then will be in a selection menu the selection of tendencies or purchases to view. On what time frame data will be displayed day, week, month and year. In which format the information will be displayed, excel, pdf. or graph.

3.2.2.3 Processing

- Upon data request the webpage will call DBMS for live data gathering at that time requested and will process to send selected file format to be decrypted by the website to be able to download the file in a secured way.

3.2.2.4 Outputs

- Given data will be displayed on the requested time frame in a graph, text pdf. file or an excel spreadsheet.

3.2.2.5 Error Handling

- Maintenance on the DBMS will prevent it from returning errors on information when requested and the live check of information before sending it to the management user.

3.2.3 <Functional Requirement or Feature #3>

3.2.3.1 Introduction

- On the GUI users will be able to make transactions for ticket purchases with the information given from users of their bank accounts and communication with banks databases will revert transactions if the user wants to revert for another selection on the webpage and secure payment.

3.2.3.2 Inputs

- For transaction inputs are the selection of film viewing with timestamp, full information of users card number, date expiration, name on account card and CVV code on the card and if requested, a reverse of the transaction call to bank if requested by the user.

3.2.3.3 Processing

- Selection will be added to the ticketing queue in the TS, then transaction information is sent to bank databases to confirm purchase and email with parched ticket will be sent to the user. In case of waiting to change the selection will be given 10 min after the timestamp which film is due for being displayed, a request for reverse transaction to the bank database will be sent.

3.2.3.4 Outputs

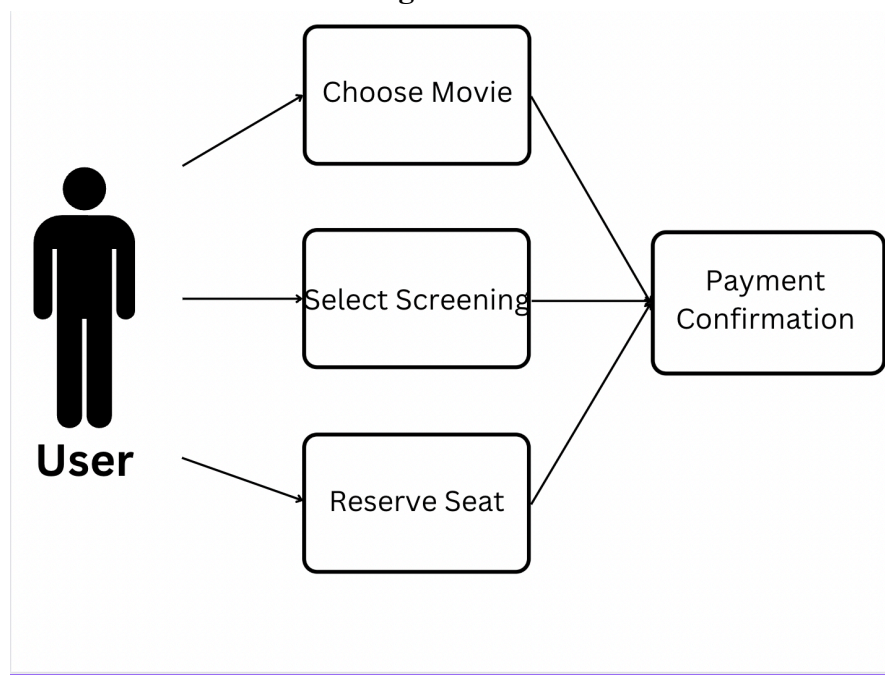
- The information given back will be an email of a ticket bought with a timestamp to display at the theater, purchase confirmation number and in case reverse transaction funds are given back with confirmation number.

3.2.3.5 Error Handling

- Constant maintenance to the DBMS will be scheduled for the prevention of errors during data transfer to bank databases as well as constant communication with these bank entities to maintain healthy and secure exchange of information.

3.3 Use Cases

3.3.1 Use Case #1: Purchasing ticket



-User: A customer who is buying tickets.

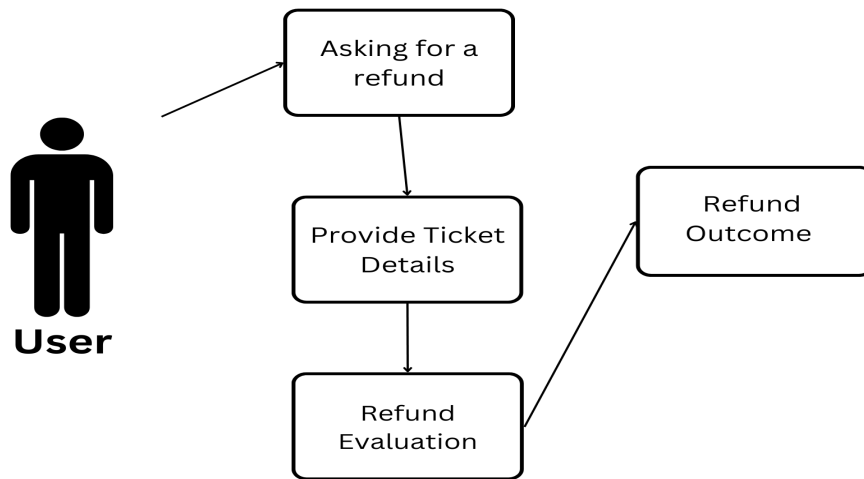
-Choose Movie: A step where the user chooses the movie he wants to see.

-Select Screen: The user chooses a screening time.

-Reserve Seat: The user reserves a seat in the viewing room of the theater where the movie is being shown.

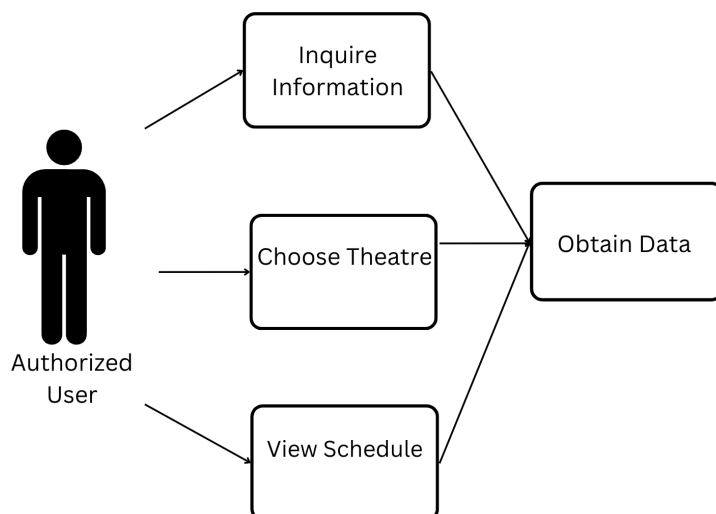
-Payment Confirmation: After entering payment details, this is the last step where the action is done and tickets are received.

3.3.2 Use Case #2: Getting a refund



- User: It's an icon of a person requesting the process, representing the individual who is requesting the refund.
- Asking for a refund: The step where the user starts the request for a refund.
- Provide Ticket Details: The user is asked to provide the details for the ticket, including the time.
- Refund Evaluation: If the movie has started or has already played, the user is ineligible for a refund. If the movie has not started yet, they are eligible for a full refund.
- Refund Outcome: It is the final part where the guest is refunded and the operation is done.

3.3.3 Use Case #3: Obtaining movie theater data



-
- User: Authorized user (i.e. employee) represented by an icon, showing the individual who asks for the information.

- Inquire Information: The first action where the authorized user is asking for the information.
- Choose Theater: The user chooses a theater he wants to have information about.
- View Schedule: After selecting a theater, he is provided with the schedule of the movies he chooses.
- Obtain Data: The final process where the users obtain the data he is inquiring about.

3.4 Classes / Objects

3.4.1 <Class / Object #1>

- Tickets

3.4.1.1 Attributes

Tickets attributes contains :

- The theater in which the film is played.
- Which movie room within the selected theaters.
- Persons allowed using purchased tickets.
- The film to be viewed.

3.4.1.2 Functions

Tickets are the main object in the system, and are present in functional requirement 3.2.1, 3.2.2 and 3.2.3

3.4.2 <Class / Object #2>

- Theater

3.4.2.1 Attributes

Theater attributes will include:

- The location in the city where the physical location on the theater is.
- Capacity of viewing rooms.
- Full capacity of people in theater
- Films being published to view in the theater.

3.4.2.2 Functions

This is the physical place where the software will be implemented and is present in functional requirement 3.2.1, 3.2.2 and 3.2.3

3.4.3 <Class / Object #3>

- Viewing room

3.4.3.1 Attributes

Main attributes of viewing rooms will have:

- Films that will be played in that specific room.
- Full capacity of users plus premium seats.
- Schedule in which films will be played on a certain day.

3.4.3.2 Functions

Viewing room essentially will be where the film will be viewed and will work as queue of full capacity to measure users to view a film in a specific day also implemented and is present in functional requirement 3.2.1, 3.2.2 and 3.2.3

3.4.4 <Class / Object #4>

- Film

3.4.4.1 Attributes

Film attributes will contain:

The title of the film being displayed.

Any media for the film advertising to display on the webpage.

Overall rating of film.

Age rating.

Genre of the film.

Time to be displayed and location.

3.4.4.2 Functions

Films are the main product to be sold, users will purchase to view these films and these are present in functional requirement 3.2.1, 3.2.2 and 3.2.3

3.5 Non-Functional Requirements

This section of the document describes the non-functional requirements of the system. These are broken down into performance, reliability, availability, security, maintainability, portability and inverse requirements. They describe how the system must operate.

3.5.1 Performance

The system must host 50,000 users at once without compromising the performance. The performance will depend on the user's connection to the internet. A strong connection to the internet should guarantee the site to load very quickly (new messages/ pages should load in under 1 second). The website should be easy to navigate.

3.5.2 Reliability

The website should be available for all those connected to the Internet. It should be available 24 hours a day, 7 days a week, unless there is scheduled site maintenance that shall take place at the least busy time of the site.

3.5.3 Availability

The website must perform consistently across different devices and browsers 99.98% of the time every month during all hours.

3.5.4 Security

3.5.4.1 Data Transfer

The transactions must be secure to protect user information. The system should log out users after 10 minutes of inactivity and users must re-enter their password to resume shopping. The system should securely transfer tickets through e-mail to the purchaser.

3.5.4.1 Data Storage

The system must have a secure database of customer information, including their account and payment information. The system should only display the last four digits of their credit or debit card that is on file. The system should not display their password anywhere, including when typing it in. Instead, there should be dot placeholders to represent how many characters have been typed in. The system's back end servers need to be secure and only allow access to authorized personnel. The system's back end databases need to be encrypted.

3.5.5 Maintainability

There are going to be periodic monthly and weekly maintenance schedules. Website maintenance includes tasks like monitoring web traffic, updating content, and ensuring that the website is secure while ensuring that the DBMS is in optimal state.

3.5.6 Portability

The website is easy to get running on any compatible server(s). There is a completely separate test version of the website to prevent a new feature from breaking other parts of the application. Website will know if it is being displayed on a mobile device, desktop environment or is being viewed on the in theater kiosk.

3.6 Inverse Requirements

Users should not be able to request refunds for tickets of movies that have already been shown.
Users should not be able to get discounted tickets unless there is a current promotion.
Users should not be able to obtain data about the theaters and the ticket sales.

3.7 Design Constraints

1. The system needs to be in compliance with industry security standards to maintain secure transactions.
2. The system must adhere to data privacy regulations and protect user information such as their payment information.
3. The system must ensure that the performance is not compromised when the site is experiencing a high volume of users.
4. The system must be user friendly. The site should be easy to navigate and be aesthetically pleasing to the user.
5. The system must be developed within the specified time, 9 months, and stay on budget.

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.

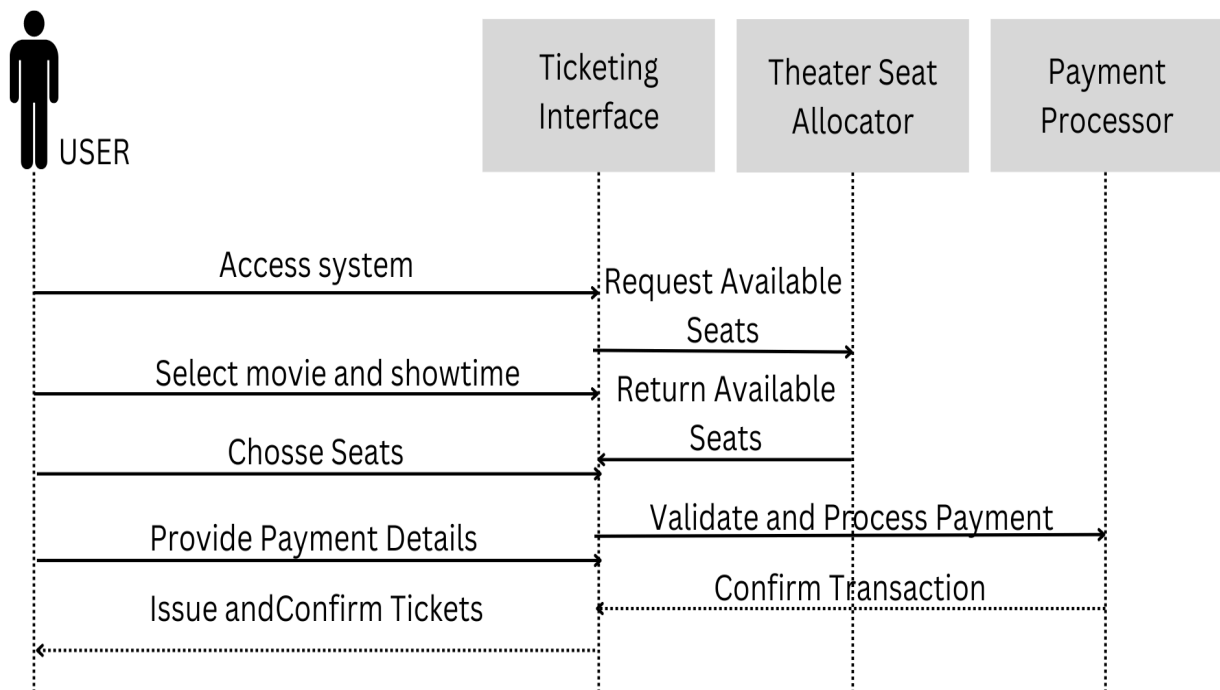
3.9 Other Requirements

Catchall section for any additional requirements.

4. Analysis Models

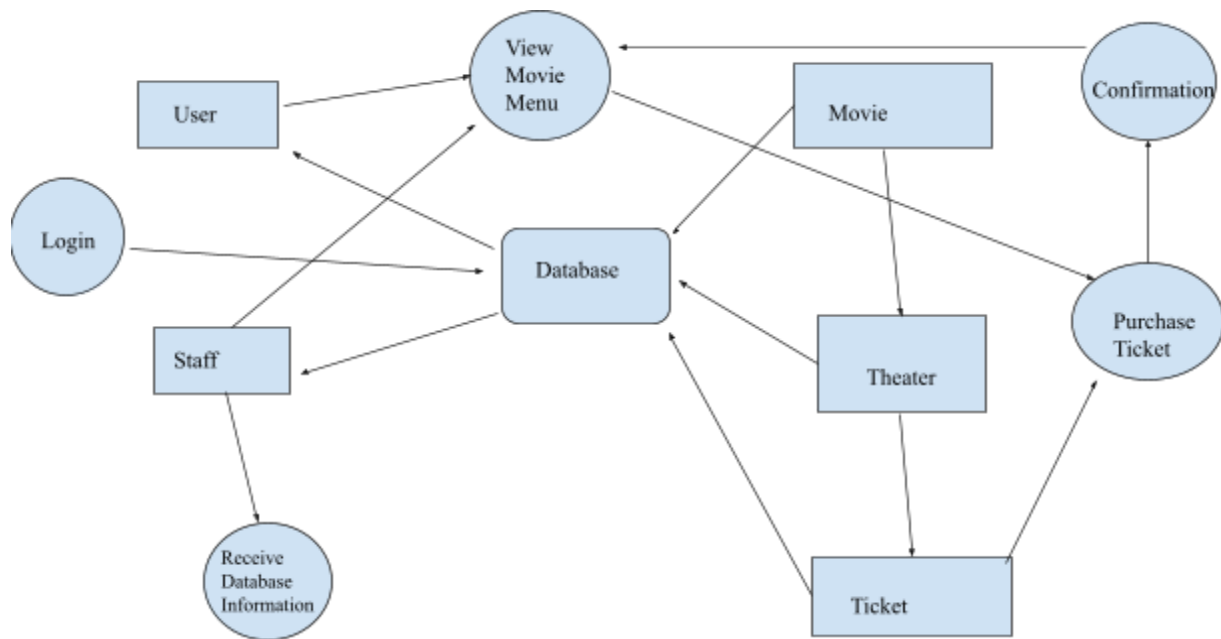
List all analysis models used in developing specific requirements previously given in this SRS. Each model should include an introduction and a narrative description. Furthermore, each model should be traceable to the SRS's requirements.

4.1 Sequence Diagrams



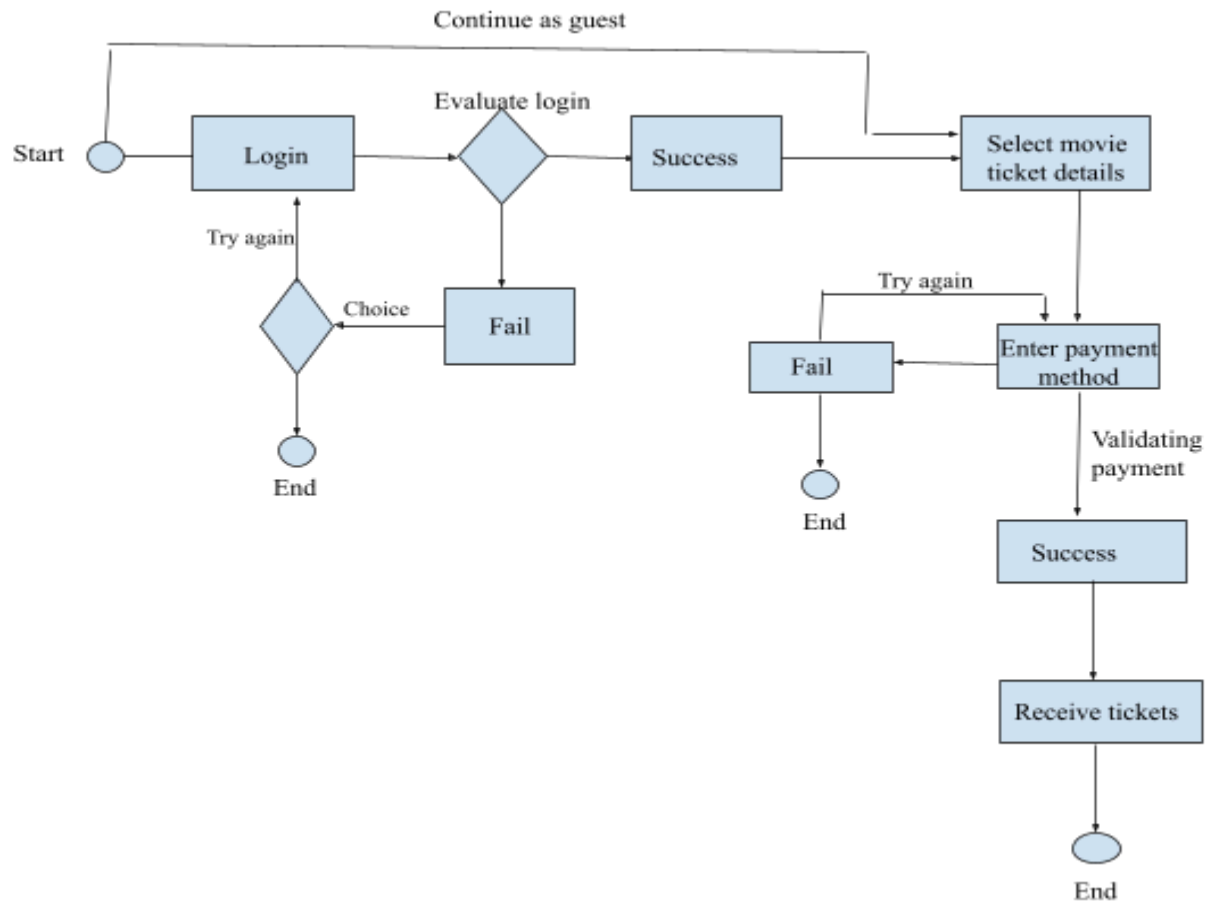
This sequence diagram describes the steps realized by a user and the whole process of what is happening when he is trying to purchase a ticket. Messages are sent to the Ticketing Interface, which then communicates with both the Payment Processor and the Theater Seat Allocator to finalize the procedure. Each entity depicted in the diagram is assigned a lifeline with messages illustrated as arrows to show the order of engagements. The Ticketing Interface acts as the core system that manages the operation akin to the "Server" shown in your diagram.

4.2 Data Flow Diagrams (DFD)



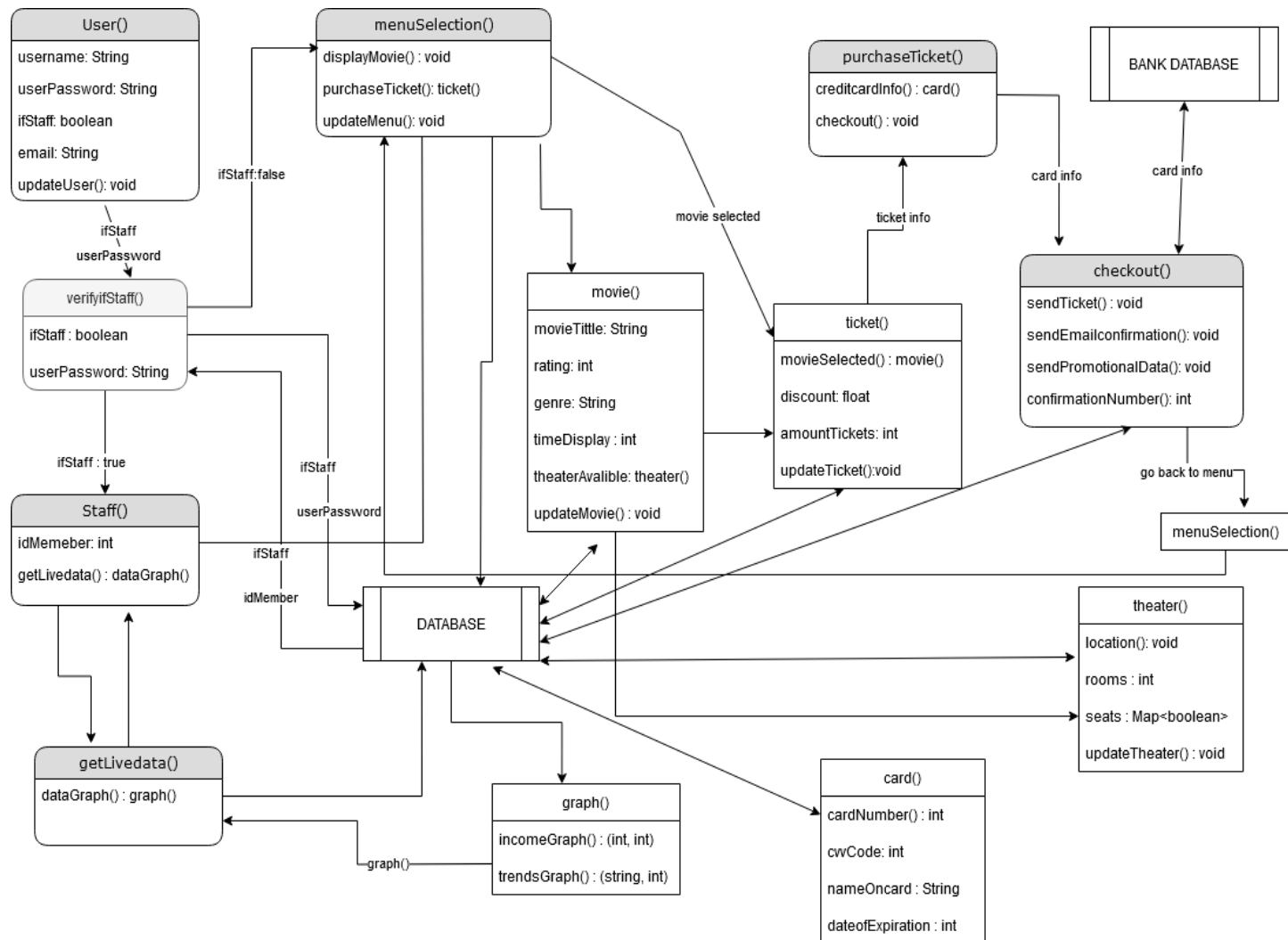
The data flow diagram represents the flow of data through the movie ticket system and the main flow of process of the software.

4.3 State-Transition Diagrams (STD)



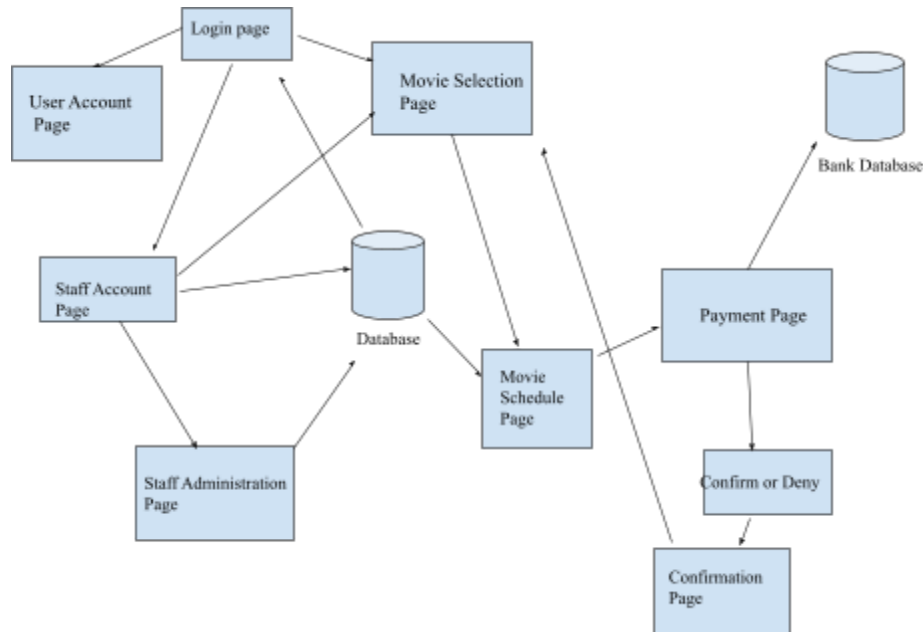
This diagram describes the behavior of the system, showing how a user interacts with the system and what state they are in at each step, as well as the transitions between each state.

4.4 UML



The overview of this system is a user logs in into the system database verifies the login credentials then will add if this user is a staff admin or a normal user. Then a normal user will be received with the main menu where they can select the movie to be viewed in an interactive and visual menu. Once the movie, seat and quantity of tickets is selected it will prompt to make a payment, the system database will search for saved credit card credentials and add to make a payment and communicate with the bank database. Once a transaction is processed it will send an email to the user with their ticket information and will be sent back to the main menu. The staff user can purchase tickets as well but also will have access to retrieve data from the database which will give a graph of trends on purchases and transactions made in a lapse of selected time.

4.5 Architecture



This diagram shows the background architecture of the system and describes the action of every page and what databases have access too.

The task is to develop this site that follows the previous diagrams. It should be a joint effort of front and back end developers to make the website function and be user friendly and aesthetically pleasing. The timeline for this system is to be developed within 6 months and tested and released within 9 months. It must stay within the allocated budget. The design team, development team, and testing team must work together to accomplish this.

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