Movie Theater Ticketing System

Software Requirements Specification

Version 1.2

02/14/2024

Group 22
Fernando Quintana

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

Revision History

Date	Description	Author	Comments
<date>	<Version 1>	<Your Name>	<First Revision>
2/14/24	Version 1	Fernando Quintana	000
2/14/24	Version 1	Luiza Rocha	000
2/14/24	Version 1	Alex Kondan	000

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 User Interfaces *3*
3.1.2 Hardware Interfaces *3*
3.1.3 Software Interfaces *3*
3.1.4 Communications Interfaces *3*
3.2 Functional Requirements 3
3.2.1 <Functional Requirement or Feature #1> *3*
3.2.2 <Functional Requirement or Feature #2> *3*
3.3 Use Cases 3
3.3.1 Use Case #1 *3*
3.3.2 Use Case #2 *3*
3.4 Classes / Objects 3
3.4.1 <Class / Object #1> *3*
3.4.2 <Class / Object #2> *3*
3.5 Non-Functional Requirements 4
3.5.1 Performance *4*
3.5.2 Reliability *4*
3.5.3 Availability *4*
3.5.4 Security *4*
3.5.5 Maintainability *4*
3.5.6 Portability *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 following document will display the required information on how a Movie Theater Ticketing System will function and give full insights on what is required while keeping an insightful look on how the system itself operates.

1.1 Purpose

The purpose of this project is to develop a functional movie theater system that allows consumers to purchase and view movies in available theaters with what tickets are available while explaining the systems put in place in order for such a system to function.

1.2 Scope

A movie theater ticketing system that will have access to movie theaters systems that will show the availability of select movies within various theaters, their movie tickets available with their corresponding seat and the different formats of each movie. This system will be an application that will be accessible upon online websites or mobile applications. All of this will be accomplished by showing listings of movie theaters that will then show each movie in their catalog leading to each time that is available and then finalizing with seat location.

1.3 Definitions, Acronyms, and Abbreviations

TERMS	DEFINITION
SRS	Software Requirements Specification - A comprehensive description of the intended purpose and environment for software under development.
WPA3	Wi-Fi Protected Access 3 - The third iteration of a security certification standard developed by the Wi-Fi Alliance
UI	User Interface - The means by which the user and a computer system interact, in particular the use of input devices and software.
Compatible	A computer that can use software designed for another make or type.
DBMS	Database management system - an interface between an end-user and a database, allowing users to create, read, update, and delete data in the database.

1.4 References

This Movie Theater Ticketing System has been prepared on the basis of discussion with Team members, faculty members and also taken information from following books & website.

1.4.1. Websites:

1.4.1.1. www.google.com

1.4.1.2. www.wikipedia.org

1.4.2. Books:

1.4.2.1. Reservations and Ticketing with Sabre by Dennis Foster

1.4.2.2. Computers as Theatre (2nd Edition) by Brenda Laurel

1.4.2.3. Computer Systems: A Programmer's Perspective (3rd Edition) by Randal E. Bryant, David R. O'Hallaron

1.4.2.4. Schaum's Series, "Software Engineering"

1.5 Overview

Section 2 of this SRS will review the applicational purpose on what a movie theater can accomplish while showing who the ideal consumers would be for the product. Section 3 of this SRS describes the internal details of the software that allows for the theater to show available movies. It will also cover the software specifications on what is required. Section 4 will show models on how this movie theater will function. Section 5 will cover who may modify this document that may cover future improvements within the software/hardware and who will approve of these modifications.

2. General Description

The movie ticketing system is a user friendly software designed for practicality and convenience that will allow users to have access and purchase movies that are currently on movie theaters so they will be able to watch it in the comfort of their home without having to wait for it to be available on other platforms or having to go to a movie theater in person.

2.1 Product Perspective

The movie ticketing system is a user-friendly software that aims for practicality and convenience, by allowing consumers to purchase tickets for local movie theaters while letting the consumer be informed on the time and location for their movie.

a) user perspective: the system will allow users to have access to movies currently in theaters through a website or mobile application showing the listing of movies available and their corresponding availability for ticket seats.
b) theater perspective: the system will allow movie theaters to sell their available seat tickets for the users of the software as they normally would, however the user will be consuming the constant from their home instead of personally going to the movie theater.
c) Movie distributors perspective: the movie distributor will provide information on the movie through the software including trailers and date of release so the users will be able to directly have an interaction with the distributors to have access to the latest movie releases.

2.2 Product Functions

The functions this software will perform include the user browsing through movies currently available on movie theaters, by buying available seats on theaters, users creating an account and having their information saved for future use, users being able to have access to information regarding the movies on display including movie trailer, genera, movie duration, movie age rate, and rating. The system should also be multi platform allowing the user to have access to the system through an application or website. The system should also have a section for user support including the information needed to help them browse the software and contact information to find support if needed.

2.3 User Characteristics

- User characteristics: Users of this software should be able to access the movie ticketing system through a user-friendly mobile application or website including viewing the available movies, creating an account and paying to watch movies currently in theaters.
- This software can be used by individuals living alone as well as families that want to have access to movies currently in movie theaters without having to leave the comfort of their homes.
- Security protocols: the software should incorporate extensive security measures for a safe user environment including WPA3 encryption, security authentication, privacy features, a secure mobile application and data storage encryption.
- language: the system should have subtitles available for users with hearing problems but also users that speak different languages to reach a broader audience and ensure that every user will have accessibility to the software.

2.4 General Constraints

This project's general constraints include

a) regulatory policies: the software must be in agreement with privacy law regulations regarding personal data and follow law regulations in regards to permits to be able to display the movies in the software.
b) collaboration: the availability of the movies displayed on the software will depend on the collaboration with movie theaters and movie distributors to get authorization and have them available on the software.
c) payment: there should be a payment system in the software that will be connected to the movie ticketing system and the movie theater available seats.
d) system quality: the software should be able to display movies in high demand without having quality problems or delays.

2.5 Assumptions and Dependencies

Assuming that the user and the software have each of these requirements, we will have designed a working system for the movie ticketing system

a) internet connection since the software requires connectivity to have access to the movies.
b) reliable devices: the user should have a reliable device to access the internet or the application to have access to the software.
c) it should also be assumed that there will be productive cooperation between movie theaters and movie distributors with the software system in order to have them available in the movie ticketing system.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The user interface for the software is compatible with any browser which can access the system. There will be three UI for this software, one is for the employee, the customer and the other will be for the administrator. A login is required for both employees and administrators. Customers will be required to create an account to login accessing this software if not present the theater. After login authentication, the employee, customer or administrator user interface will be shown.

User Access Levels

User	**Permission**
Administrator	○ Add new users ○ Creating user accounts for employees ○ Update profile information ○ Close user accounts ○ View and create reports ○ Manages theater information
Employee	○ Manages theater information ○ Update customer profile information ○ Update their profile information ○ Manages customer ticket purchases

User	○ Manages ticket purchases ○ Change settings on account ○ View seat rates ○ Update their profile information

3.1.2 Hardware Interfaces

The application requires:

- core i3-i5 equivalent
- 8 gigs ram
- 20 gigs storage
- gigabit ethernet

User application requires:

- gigabit ethernet

3.1.3 Software Interfaces

- **Login Page**

Every user must login in order to use the functions of the system.

- **Administrator User Interface:**
  - Register
    - Add new user
    - View Existing User
    - Close Accounts
  - Profile
    - Edit Profile
    - Change Password
  - Theater system
    - View seat
    - Compare seat prices
    - Create / view/ edit rate
    - Create / view / edit concession rates
    - Change Name
    - Change Time
    - Change movie showings
    - Change Theater
    - Change movie times
    - Alert users
  - Settings
    - Update
    - Reboot

        - Logout
    - Archive
        - Monthly Report of seats filled
        - Monthly Report of sales
        - Monthly Reports of concession sales
        - Create Report

- **Employee User Interface**
    - Theater
        - View Seats
        - Compare seat prices
        - View rates
        - View movie showings
        - View theater number
        - View theater seat availability
        - Alert users
    - Settings
        - Update
        - Reboot
        - Logout
    - Archive
        - Monthly Reports of seats filled
        - Monthly Reports of sales
        - Create Report
    - Profile
        - Edit Profile
        - Change Password

Customer User Interface

- Theater
    - View Seats availability
    - Compare seat prices
    - View rates
    - View movie showings
- Settings
    - Update
    - Reboot
    - Logout
- Profile
    - Edit Profile
    - Change Password

3.1.4 Communications Interfaces

This project supports all types of web browsers. We are using simple electronic forms for reservation forms, ticket booking etc.

3.2 Functional Requirements

3.2.1 <Functional Requirement or Feature #1>

3.2.1.1 Introduction
Purchase of ticket
3.2.1.2 Inputs
Input by customer or employee
3.2.1.3 Processing
Customer purchases ticket
Block bots of purchasing tickets

3.2.1.4 Outputs
Outputs ticket information
3.2.1.5 Error Handling
Allows multiple users to access system
Prevents bots from purchasing multiple tickets
Prevents multiple customers from purchasing the same ticket

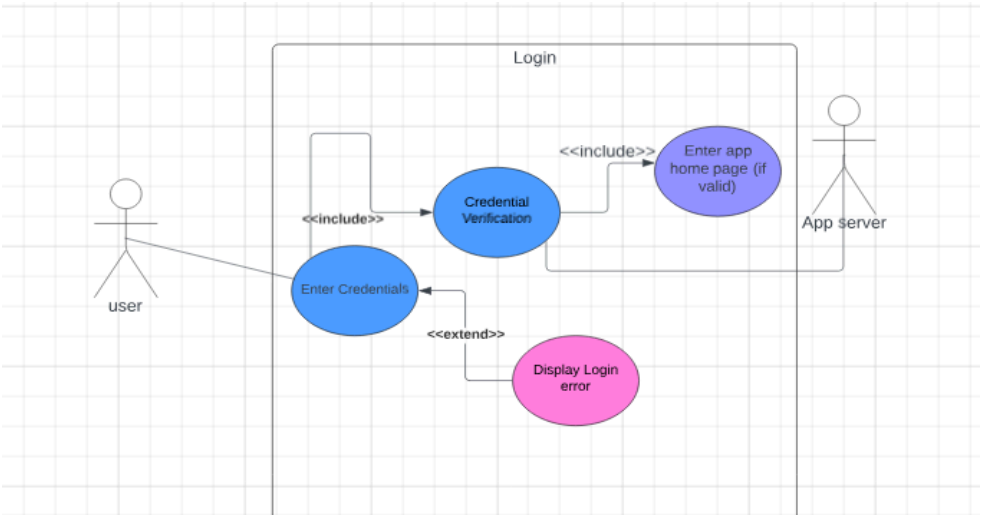
3.2.2 <Functional Requirement or Feature #2>

…

3.3 Use Cases

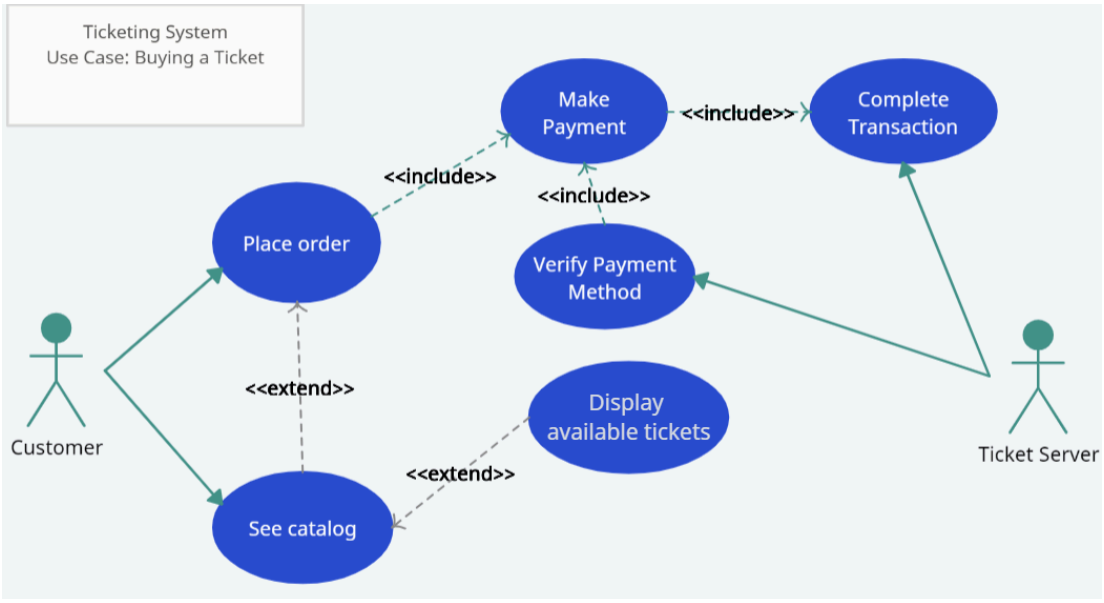
3.3.1 Use Case #1

User logs in:



3.3.2 Use Case #2

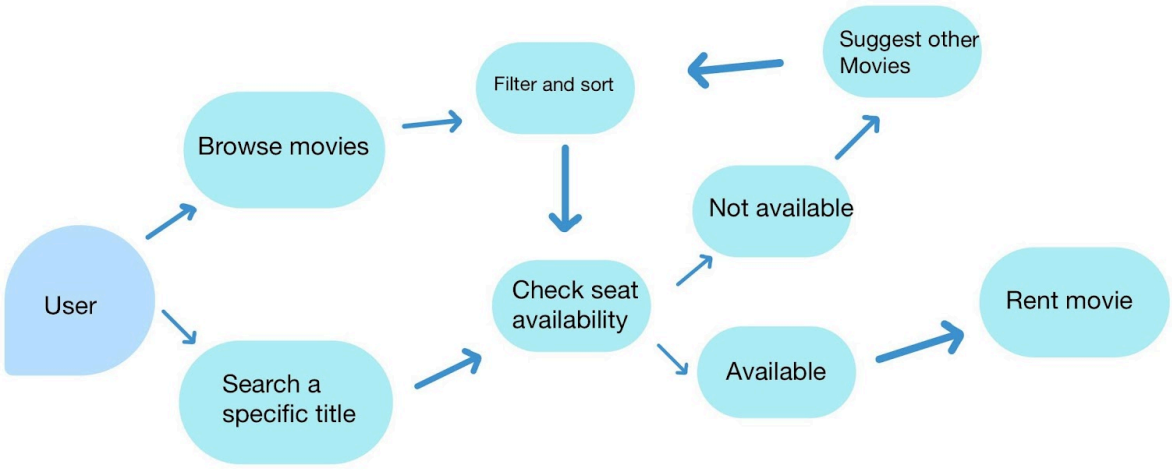
Customer places order:



-

3.3.3 Use Case #3

- Selecting a movie:



3.4 Classes / Objects

3.4.1 <Class / Object #1>

3.4.1.1 Attributes
name:String
Ticket num: int = 15
Seat num: int =2
Price: int =3.50
3.4.1.2 Functions

getPrice()
getSeat()
getTicket()
getName()

3.4.2 <Class / Object #2>

Attributes
customerName: String
customerNum: int
ticketNum: int
customerPassword: String

Functions
getCustomerName()
setCustomerName()
getTicketNum()
getCustomerPassword()
setCustomerPassword()

3.5 Non-Functional Requirements

3.5.1 Performance

The steps involved to perform the implementation of the movie theater system database are listed below.

3.5.2 Reliability

The movie theater system should have anti-malware to prevent security breaches and have multiple storage locations, and up to date software and hardware to ensure reliability.

3.5.3 Availability

The movie theater system should be available to customers for advance booking and purchasing of theater seats.

3.5.4 Security

Such that a security breach happens. Movie Theater Ticketing systems have multiple backup storage units.

3.5.5 Maintainability

Movie Theater ticketing systems require regular updates to fix and maintain current with web technology, bugs, and security patches.

3.5.6 Portability

Movie Theater ticketing system will ensure the system is self descriptive. System must have sufficient comments to explain the implementation of the system.

3.6 Inverse Requirements

The movie theater system does not sell vip tickets or provide discount options.

3.7 Design Constraints

The movie theater system must have a well maintained firewall as it is a tool for transactions and will have valuable information about the consumer's bank accounts. There will also need to be a system set for returning money incase of refunds or other issues within the physical theater occur. The last thing is to be able to have a manageable bandwidth for when popular movies come out so that the software does not crash.

3.8 Logical Database Requirements

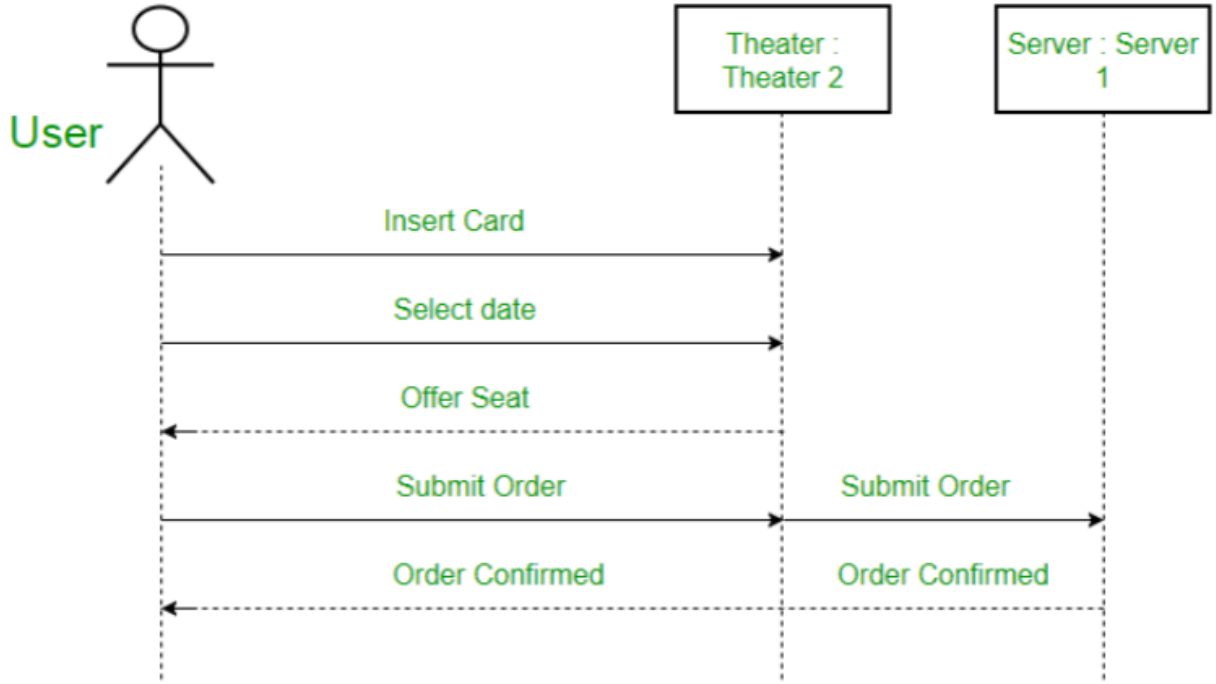
The movie theater system will use a database that will be controlled by a DBMS. Yet the physical server to handle this Database will need around 32 GB of storage minimum but can be upgraded to 64 or more if necessary.

3.9 Other Requirements

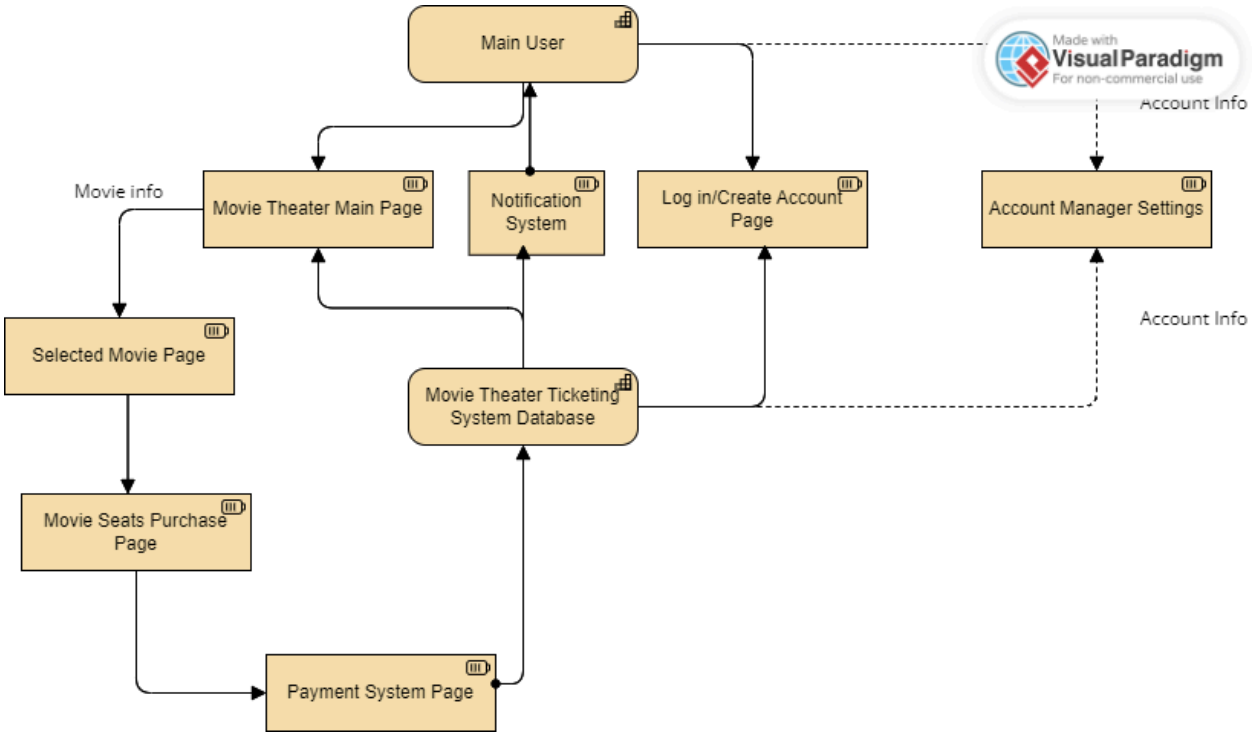
The movie theater system will also require a review system that can range from giving a movie a select amount of stars from 1-5 and/or a post board where the users can give detailed description on what they thought about the movie.

4. Analysis Models

4.1 Sequence Diagrams

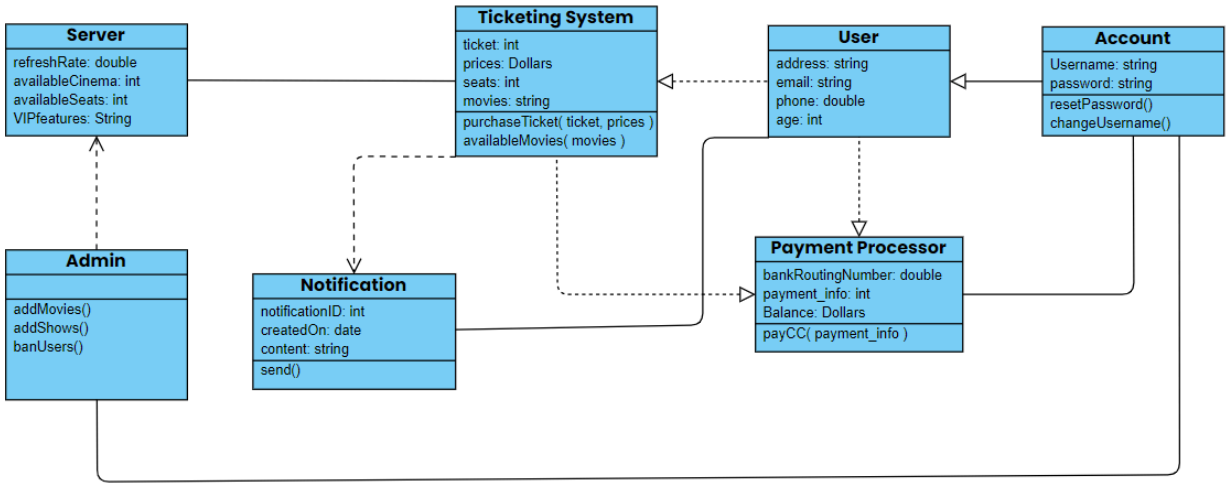


4.2 State-Transition Diagrams (STD)



4.3 Data Flow Diagrams (DFD)

4.4 Unified Modeling Language (UML)



4.4.1 Unified Modeling Language (UML) Explained

Class: Server
Description of Class: The main system that gives back information provided on films and other data.

Attributes:

- refreshRate: double - how fast data can be provided about specific topics.
- availableCinema: int- how many theaters will carry specific films.
- password: string - stores the password of the user's account.
- VIPfeatures: string - info on what premium status gives to the users.

Operations: None as most of the Attributes are called from here.

Class: Ticketing System
Description of Class: The system that manages the prices and location for the seats.

Attributes:

- ticket: int - How much each seat costs within a movie theater.
- Prices: Dollars - The cost of what is being purchased from seats to food.
  Seats: int - How many seats are available.
- Movies: string - The movies that are available within the movie theater.

Operations:

- purchaseTickets(tickets, prices) - obtains the prices of the chosen movie and how many.
- availableMovies (movies) - Show the user what movies are available to be shown and any additional info within the movie

Class: User
Description of Class: The person who is using the interface of the ticketing system
Attributes:

- address: int - Known location to provide info on local movie theaters.
- email: string - email address to give notifications on deals & for signing in to the website/app.
- phone: double - To give notifications & function for double verification when signing in.
- Age: int - To verify age limit & give deals on birthday

Operations: They do not give any additional operators besides keep information on themself

Class: Account
Description of Class: The main system of setting to review threw website settings and ID verify.
Attributes:

- Username: string - Secondary login username besides email & to customize the user
- Password: string - The main system to lock other users in order for security & to keep the creator of the account to access the system.

Operations:

- resetPassword() - Allows the user to change the password to a new one after completing a 2-Step verification.
- changeUsername() - Allows the user to change the username to a new one after an email verification.

Class: Admin
Description of Class:
Attributes: No operators available as Admin is for management in order to adjust movie information as needed
Operations:

- addMovies() - add films as they release.
- addShows() - add additional films as they release.
- banUsers() - operator to ban any problematic user that causes issues within a movie theater.

Class: Notification
Description of Class: System to notify the user of verified purchase, the time of the movie and advertisements
Attributes:

- notificationID: int - Formatting on notification if it on android, iphone, email or phone number message.
- createdOn: date - A system set to remind the user of when the movie they bought is on in the day.
- Content: string - The information of the notification

Operations:

- send() - push a notification to the user on important info through email and/or phone

Class: Payment Processor
Description of Class: called upon from the ticketing system to make the transaction function once a ticket is purchased.
Attributes:

- bankRoutingNumber: double - The location of the bank account from the user to withdraw the money to complete the transaction
- payment_info: int - what the payment is for in order to give to the bank
- Balance: Dollars - The total amount being withdrawn from the account

Operations:

- payCC(payment_info) - The operation done to transfer money once purchased.

4.5 Segmentation by Weighted Aggregation (SWA)

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