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.

<Theater Ticketing System>

Software Requirements Specification

<Version 4>

<3-27-2024>

<Group #19>
<Emilio Cecena, Matthew Washington, Nanci Cardenas Martinez>

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

Revision History

Date	Description	Author	Comments
<1/31/24 -2/14/24 >	<Prototype 1>	<Emilio Cecena,Matthew Washington, Nanci Cardenas Martinez >	<First Revision>
<2/28/24 >	<Prototype 2>	<Emillio Cecena, Matthew Washington, Nanci Cardenas Martinez>	<Version 2, in this one we fixed some previous mistakes and updated it with our diagrams and descriptions>
<3/13/24 >	<Prototype 2>	<Emillio Cecena, Matthew Washington, Nanci Cardenas Martinez>	<Version 3, >
<3/27/24 >	<Prototype 2>	<Emillio Cecena, Matthew Washington, Nanci Cardenas Martinez>	<Version 4, updated diagrams and included a data management strategy>

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

This is a Theater Ticketing system that is web based. It will be able to handle up to one thousand users simultaneously, and be able to block bots from buying multiple tickets at once. It will have security that will protect users information as well as have a login service for recurring users.

1.1 Purpose

The purpose of this document is to create an online ticketing system that will allow users to purchase tickets to watch movies and protect their information.

1.2 Scope

1. The software products will be a Theater Ticketing System called EncoreTicket, a customer service line, and an AI chat feature called TicketGPT.
2. EncoreTicket will be the hub where users can login to buy tickets for a movie in a theater near them, the customer service line will be for people that need immediate assistance, TicketGPT is an AI chat system for people who have general questions that can be answered by the chat.
3.
    (a) The objective of EncoreTicket is to provide a safe website where people can purchase tickets for nearby theaters and for certain showings. The search will be by movie name, time frame, and location if specified by the user. The customer service line is meant for specific customer problems such as a malfunction in payment or overcharge. TicketGPT is meant for more general questions such as refunds or password recovery.
    (b) For system requirements:
        i. Prices will be uniform with the theater's price
        ii. Payment will be behind a login screen and an encrypted website, it will also only accept payment from the country the user is currently in
        iii. The languages it supports is English, Spanish, Swedish, along with 20+ languages
        iv. The website will automatically convert currency based off of the country the user is currently in
        v. Users will only be able to purchase up to 10 tickets per account.
        vi. Only one device can be logged into at a time, and will require two factor authentication
        vii. EncoreTicket will have a history tab logging all purchases with dates going from newest to oldest.
        viii. EncoreTicket will have a queue for individual showings and will update constantly and display what place in the queue the user is
        ix. EncoreTicket allows users to register for loyalty membership that stores user information, payment, and loyalty points. Users can purchase pre orders

x. Payment will only be accepted by banks in the country the user is currently logged in or has been logged in before and will also accept PayPal and Visa anywhere in the world.
xi. Users will be able to choose where they want to sit and in what theater in which country they are currently in, they can also preorder tickets for movies that have yet to show.
xii. Tickets will be able to be returned or exchanged for a different showing of equal value up to 15 minutes before the original showing.
xiii. EncoreTicket will be able to handle up to 15 million users simultaneously.
xiv. The Website will be encoded and each account will require two factor authentication, and each ticket will have a unique hexadecimal code that will make it so it can only be scanned once.

1.3 Definitions, Acronyms, and Abbreviations

DB - database
ET - EncoreTickets
TGPT - TicketGPT

1.4 References

1) chatGPT
2) FanDango
3) AMC

1.5 Overview

1. EncoreTickets allows a user to purchase tickets online quickly and easily. The website also allows users to access easy helplines through a customer service line and an AI assisted TicketGPT.
2. Encore tickets will have a home page displaying the different movies available for purchase as well as the times available. The home page will also have a search bar that allows for users to search for desired movies. On the home page there will also be a drop down tab that allows for users to swap to their spoken language.There will be a clickable tab called Help that will allow the user to access the two different assistance options that we have in place, the customer hotline and the ticketGPT which will have automated responses for common questions. There will also be another tab called account which will allow the user to access numerous other tabs that allow the user to view purchase history and other important account information. There will also be a cart tab that will allow users to view items ready to purchase.

2. General Description

2.1 Product Perspective

A ticket selling service that performs the following functions:

1. Movie details: it will hold all the movies being shown at different theaters and the showings that are available to watch and purchase tickets for.
2. Customer information: encore tickets will hold the customer information such as payment information and login information
3. Ticket details: The software will also hold each ticket purchased by the customer and the tickets unique code to get into the movie after purchase. It will also hold information on all the tickets purchased by the customer.

2.2 Product Functions

1. *Display the movie theaters and movies showings along with the location of theater*
2. *Display the dates and times of those movies*
3. *Display the price of the tickets*
4. *Display the seats available for the movies*
5. *Keep users' card information safe*
6. *Convert currency based on location*
7. *Update location whenever website is logged into*
8. *Allow users to make a login to keep information in one space*
9. *Allows for customers to receive a refund if they decide to cancel their reservation*

2.3 User Characteristics

Customers should be able to do the following functions:

1. *Buy tickets for certain times*
2. *View purchase history*
3. *Get refunds for movies they want to cancel*
4. *View movie times*

2.4 General Constraints

1. *Time and Location*
2. *Supported Languages*
3. *Security*
4. *Human error*
5. *Theaters that aren't supported*

2.5 Assumptions and Dependencies

1. *Browser compatibility*
2. *Mobile compatibility*
3. *Country User is in*
4. *Theater cooperation*
5. *Age Limit*

3. Specific Requirements

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)*

3.1 External Interface Requirements

3.1.1 User Interfaces

1. Home page
2. Checkout page
3. Help page
4. AI chat
5. Login screen
6. Settings page
7. Search page
8. Profile
9. Ticket page
10. Seating page
11. Theater page
12. Graphical
13. Java

3.1.2 Hardware Interfaces

1. Connect to printer
2. Internet connection
3. Ticket Download

3.1.3 Software Interfaces

1. Browser compatibility
2. Information Database
3. Currently active users
4. Total user signups
5. Number of times webpage has been frequented
6. Amount of tickets bought off of website for a specific movie
7. Built for windows machines

3.1.4 Communications Interfaces

1. Wifi
2. Printer

3. Hard drive

3.2 Functional Requirements

3.2.1 <AI chat feature>

3.2.1.1 Introduction: AI chat that is meant to help users with questions.
3.2.1.2 Input: User needs to be clear when talking to AI inorder to get the best answers
3.2.1.3 Processing: The AI will try its best to answer and provide help to users
3.2.1.4 Outputs: An AI answer that will try and satisfy the user
3.2.1.5 Error Handling: Have the AI people to take more broad questions, and if it can not answer a question send users to help line by providing the phone number

3.2.2 <Login page #2>

3.2.1.1 introduction: Allows the user to login to EncoreTickets
3.2.1.2 Inputs: The user will put in their username and password
3.2.1.3 Processing: The inputted username and password will be checked using our information database to check if it is correct or not.
3.2.1.4 Outputs: If the information inputted is correct the user will be successfully logged into the account
3.2.1.5 Error Handling: If the information inputted is incorrect then the user will not be logged in and they will receive an error message “Wrong username or password”. There will also be a password reset button that will allow the user to reset their password through the email associated with logging in

3.2.3 <Security>

3.2.1.1 Introduction: The system secures all confidential customer information
3.2.1.2 Inputs: The user inputs credit card information, email address, name, password when logging in
3.2.1.3 Processing: secures confidential information of customers
3.2.1.4 Outputs: Keeps customer information hidden and only allows customer to see their information
3.2.1.5 Error Handling: Let’s user know if credit card information displayed is incorrect or if the email address can not be found.

3.3 Use Cases

3.3.1 Use Case #1

The user logs into the page to access their account and movie showings

3.3.2 Use Case #2

User purchases tickets for desired movie getting the ticket code and payment confirmation

3.3.3 Use Case #3

User refunds a ticket receiving their money back and getting confirmation

3.3.4 Use Case #4

The user asks for assistance through our ai chat bot receiving assistance

3.3.5 Use Case #5

The user gets assistance through a human operated phone line

3.3.6 Use Case #6

It will not allow the user to purchase more than ten tickets at once

3.3.7 Use Case #7

Will allow the user to recover password or username if forgotten by sending a link via text message or email to user

3.3.8 Use Case #8

Will allow user to login into only two devices at a time

3.3.9 Use Case #9

In order to swap device the user needs to insert a code that is either sent via text or email to user, and can only swap every 4 hours

3.3.10 Use Case #10

If the internet goes out during checkout the process is automatically stopped and payment will not go through.

3.4 Classes / Objects

3.4.1 <User Profile #1>

3.4.1.1 Attributes: User name, Password, reset password, get help, User payment, Language, Location
3.4.1.2 Functions: setPassword, getPassword, setUserName, getUserName, passwordMatch, usernameMatch, setLocation, getLocation, setLanguage, getLanguage, setPayment, getPayment, changePassword, changeUsername, changeLocation, changeLanguage, changePayment, displayHelpPage
<Reference to functional requirements and/or use cases>

3.4.2 <Data Transfer #2>

3.4.1.1 Attributes: The system uses secure sockets in transactions that include confidential customer information
3.4.1.2 Functions: The system confirms transactions with customers, automatically logs out users if there is inactivity

3.5 Non-Functional Requirements

3.5.1 Performance

Something that will be done in order to optimize the performance is to make it only work on one search engine software such as chrome or firefox, or if they are running it on a non-optimized browser we will tell them in the corner of it that it will work better on a separate browser.

3.5.2 Reliability

We will make it so that it can handle over 100,000 active users in a region at a time, we will also have it so that the card information is handled separately from the main website

3.5.3 Availability

It will be available in almost every country with more than 40+ languages being supported. It will be available on every known browser and most used browser in the world.

3.5.4 Security

The credit card information will be held on a separate website and to login to an account in the first place there will be two factor authentication as well.

3.5.5 Maintainability

It will be a java program that will be divided into multiple classes that will be able to change and alter for the more basic things such as changing the homepage color but for things that are to be more secure inorder to make a change the code there will need to be a code to access that, in which only a higher up will have access to.

3.5.6 Portability

The website will be available and optimized for mobile devices such as smartphones, and tablets.

3.6 Inverse Requirements

The website shall not use red color when asking for user input from customers
We will not be including movie trailers
cant purchase more than 10 tickets per account
Only one account per email

3.7 Design Constraints

Specify design constraints imposed by other standards, company policies, hardware limitations, 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.

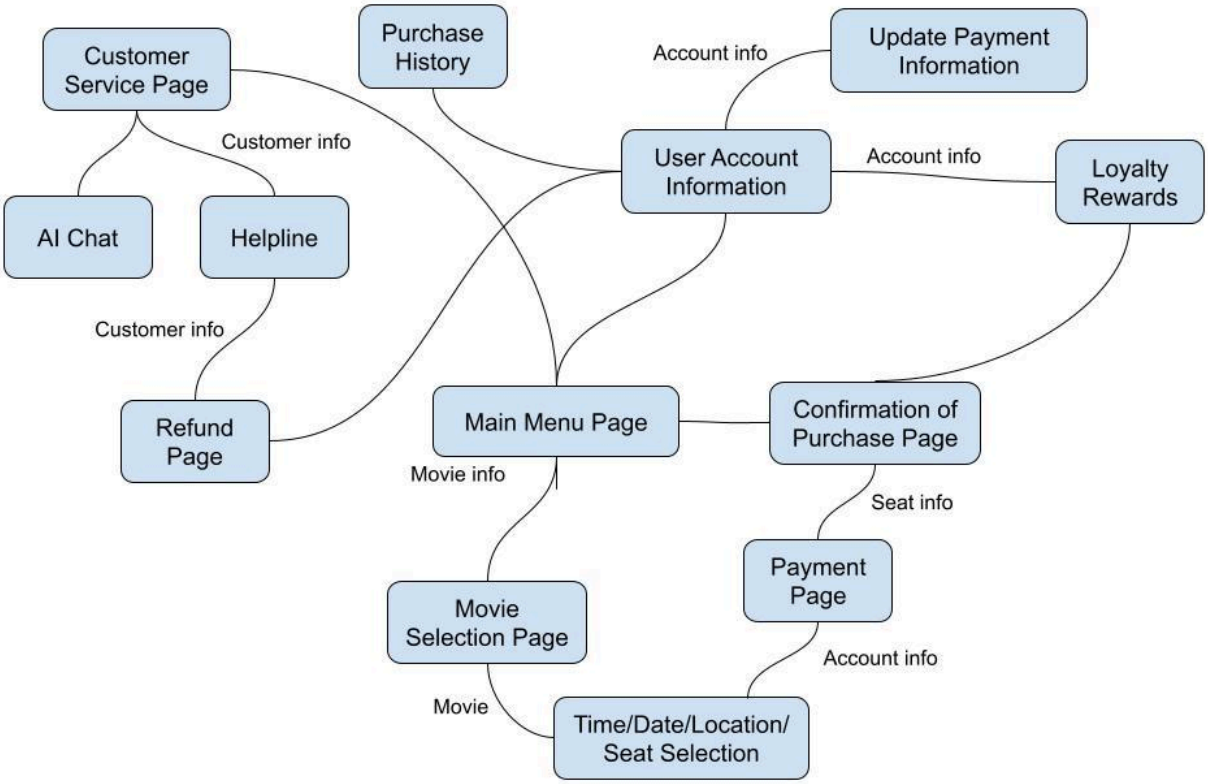
4. Software Design Specifications

4.1 Software Title: Encore Tickets

4.2 System Description:

EncoreTickets is a ticketing system that is meant to help users purchase tickets online. Users will be able to search for specific movies and movie theaters using our search bar; when the user finds a movie they'd like to see, they can click on it and it will come up with showtimes at theaters near them. With this, the user will be able to select the showtime they want at which theater they want, then it will take them to a page where they can select which seat they'd like. From this page the user will be able to select their seat, and any food they want so it will be ready for pickup beforehand. Then for the user to be able to finalize their purchase they will need to either sign up or login to their current account. This is because the user's details are kept safe behind multiple security measures. Now as the user logs in they will need to verify it is them by using a two-step verification process. Then they are able to purchase whatever they need. Some features of this application are that it is supported on all platforms, it will have over 30+ languages translated, and will have two ways to help users. One will be a call line where a user can talk to an operator who can help them with more complex questions, or they can use our AI help chat that will be able to answer more basic questions but will be faster at responding.

4.3 Architecture Diagram

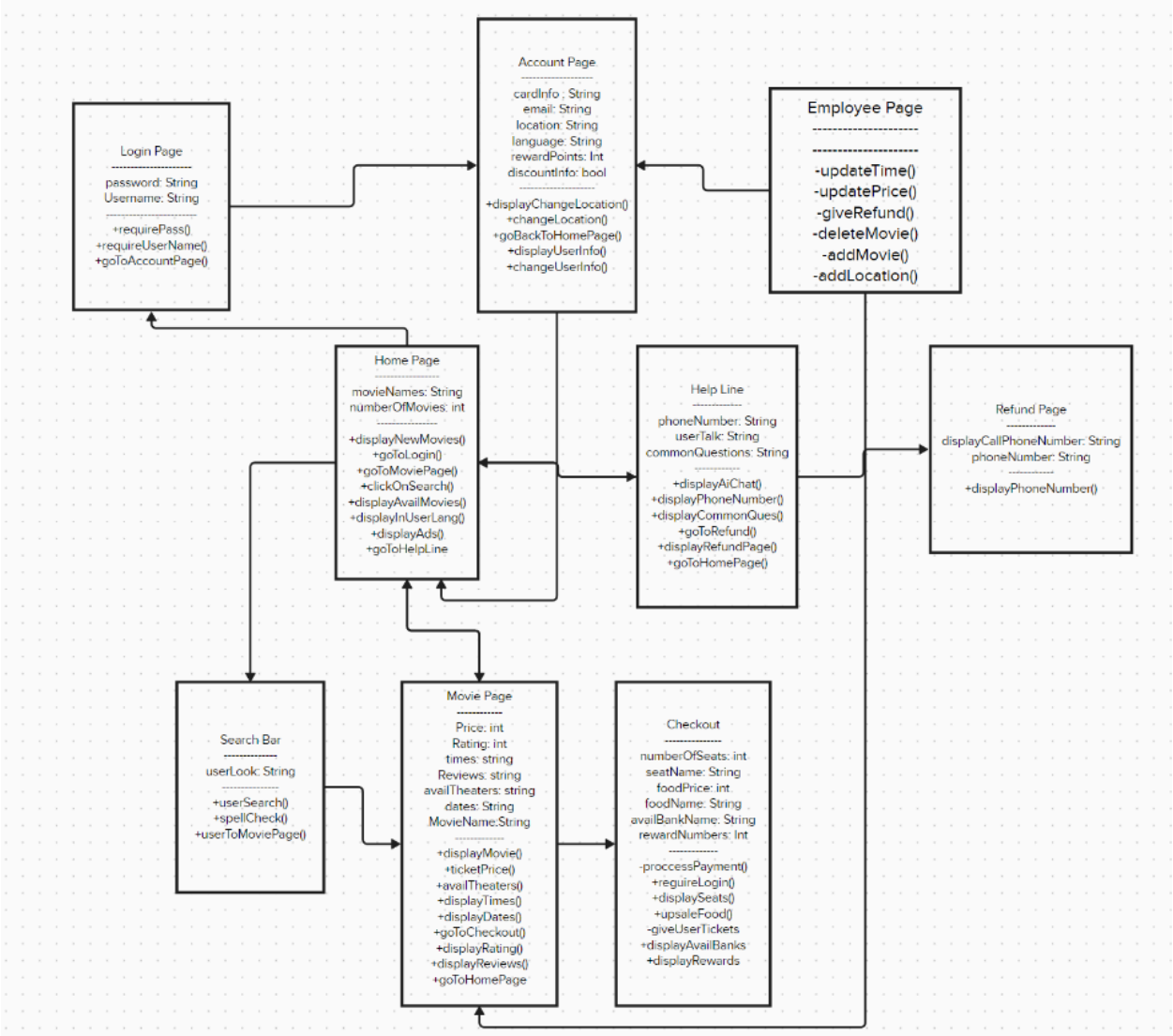


<account info, seat info, movie, movie info, customer info>

The components of the system include the main menu page, movie selection page, time and date selection page, seat selection, payment page, confirmation of purchase page, user account information, update payment information, loyalty rewards, purchase history, customer service page, AI chat, helpline, and refund page. The main menu page is able to go to the customer service page, user account information page and the movie selection page. One of the main components include the movie selection page where customers will go to select a movie and then continue to make a payment. Once a movie is selected from the movie selection page it will take them to the next page where they can select the time, date, and location of the movie. Once all these selections are made it goes to the next page which allows the customer to choose their seat. After all customer selections have been made the customer is taken to the payment page to make their payment and once payment has been made they will be taken to the confirmation purchase page. From the confirmation page customers can either go back to the main menu page or go to their loyalty rewards page to see their rewards or points they earned from their purchase. From the main menu page customers can access their account information which allows them to see their purchase history, loyalty rewards, and also allows them to make updates to their payment if needed. Lastly, from the main menu page customers can go to a customer service page which allows them to either use the generated AI chat or use the helpline to talk to a service agent. The

helpline component is able to go to the refund page to give a refund to a customer. Helpline also has access to user account information to help with any necessary problems.

4.4 UML Diagram



4. 5 Description of Classes:

Our UML diagram starts with the **login page** which has two methods requireUserName and requirePassword which will take in the username and password as input and then send the user to the account page through the goToAccountPage() method. Login page consists of two strings password and username holding the values of the username and password typed.

Moving on we also have a section in our diagram called the **account page** which has strings and ints that hold information such as card information, email address location of the user and reward points. It will also have a boolean that will check whether the user has a discount. It will consist of different methods to change their location changeLocation() return to the homepage

goBackToHomePage() display their user information DisplayUserInfo() and the ability to change their information changeUserInfo.

We will also have a class called **employee page** consisting of different methods allowing for an employee to alter movies that are going to be shown some of these methods include the ability to change the time on a movie updateTime() they will be able to change the prices on movies as well as provide refunds for users updatePrice() and giveRefund(). It will include removing movies and adding them addMovie() deleteMovie(). And lastly it is able to add a location to the movie addLocation()

We will also have an class referring to user refunds called **Refund page** it has strings called displayPhoneNumber and phoneNumber with a method titles displayPhoneNumber() it will be a page that tells the user to call the phone number displayed through the function displayPhoneNumber() for a refund.

We have a class in the diagram titled **help line** where the user will be able to get most of their assistance from. We will have three strings phoneNumber userTalk and commonQuestions along with the methods displayAiChat() displayPhoneNumber() displayCommonQuestions() goToRefund() displayRefundPage() and lastly goToHomePage(). These methods will display the ai chat in order for the user to interact with it display a phone number so the user can call for assistance. The method displayCommonQuestions() will show the user typically asked questions that can be solved with simple instructions. And the last three methods send the user to different pages such as displaying the refund page and sending the user to the homepage

There will also be a **Home Page** class that has a string called movieNames that stores the names of the different movies that are going to be displayed on the movie page it will also hold an int called numberOfMovies. We will have a method that can display the new movies coming to theaters. A method called goToLogin() that sends the user back to the login page. A method called goToMoviePage() which sends the user to the movie page. A method called clickOnSearch() which opens up the search bar for the user to type in movie names. A method called displayAvailMoveis() which will show the movies that are currently being shown in theaters. A method called displayInUserLang() which will change the text being displayed on our website to the users preferred language. ANd lastly a method that sends the user to the help line class and a method to displayAds() so our website can bring in more revenue.

Moving on from the home page we have a class called **Search Bar** that has a string called userLook to hold user input of what they are trying to search for. The class will only have 3 different methods userSearch() which will take in the string userLook and search for the movies available. We will also have a method that will check the user's spelling spellCheck(). And then a final method userToMoviePage() which sends the user to the movie page.

Moving down the diagram we have the class **movie page** which will be the main area that the users will be able to view the movies price, the amount of seats available and the dates that the movies will be available. We have two ints holding the price and rating and 5 strings called times, reviews, availTheaters, and dates. The methods that will be in the class include displayMovie() which show the movies available, ticketPrice() which displays the cost of a ticket for the specified movie, availTheaters() which show the different theaters showing the movies, displayTimes() which shows the times of the movies, displayDates() which shows the dates of the showings, goToCheckout() which sens the user to the checkout page, displayRating() which shows the rating of the movies, displayReviews() which shows the audience reviews on the movies. And lastly a method called goToHomePage() which sends the user to the home page.

Using the previous method called goToCheckout() it will send the user from the home page to the checkout page. The **Checkout** class will contain 3 ints numberOfSeats, foodPrice, and rewardNumbers. The class will also contain 3 strings seatName, foodName, and availBankName. The checkout class will also have many different methods that help the user pay for their items. processPayment() which will check the users payment information and accept payment, requireLogin() which will make the user login before paying for their cart, displaySeats which will show the remaining seats before purchase, upsaleFood() which is trying to advertise food to the user, giveUserTicekts() which will send the user their tickets electronically after their purchase has been confirmed, DisplayAvailBanks() which will display the banks that are accepted by EncoreTickets, and lastly displayRewards() which will show the users who are loyal to encoreTickets their reward points for previous purchases made.

4.6 Description of attributes

Attributes of Login Page

1. Password which is a string of characters, that must be checked with what is saved in the database to make sure that they are equal
2. Username which is also a string of characters that is checked with what is saved in the database to make sure they are equal

Attributes of Account Page

1. User's selected language which will alter the characters shown
2. Their user information consists of the card number, email, current location, reward points, and discount information

Attributes of Employee Page

1. It will consist of operations that will allow the employee to alter the movie information and the website directly such as adding or removing a movie

Attributes of Home Page

1. Firstly it will show all of the new movies and movies in theaters currently
2. It will display the amount of movies available for viewing
3. It will allow users to access the search bar

Attributes of Help Line

1. It will display the help number and frequently asked questions
2. It will also display the AI chat feature and what it should be used for

Attributes of Refund Page

1. It will display a message that will tell the user that they need to call the help number and it will display said help number

Attributes of Search Bar

1. There is a variable that will allow the user to input a movie they are interested in and will take them to that page, with spell check to help

Attributes of Movie Page:

1. It will display the Prices of the tickets for that movie at a specific time and theater
2. It will also display the ratings and reviews of a movie and allow users to do their ratings and reviews
3. It will also display the time and dates for the movie, the movie itself, and the movie name

Attributes of Checkout

1. It will allow users to purchase tickets

2. Choose their seats, buy food to be ready by the time they get there, and display food price and quantity
3. Have the user rewards info available, and show which banks are accepted to use

4.7 Description of operations:

Home Page:

1. It will display the new movies, other available movies, display ads on the side of the screen, and display in the user language all on the home page
2. On the home page users are allowed to login
3. Users are allowed to click on a movie they want to purchase that is displayed on the home page and will take them to the movie page
4. Users are allowed to use the search bar to lookup movie titles

Movie Page:

1. The movie page displays all information about the movie. It displays the movie name along with the ticket price, available times, theaters and dates.
2. The movie page also displays reviews made on the movie and it's overall rating
3. Once user is done with choosing time, date, and theater they are taken to the checkout page
4. If user decides not to purchase any tickets then they can go back to home page

Search Bar:

1. Users are allowed to search any movie titles on the search bar
2. Spell check is enabled to suggest possible movie titles if user entered data is spelled incorrectly
3. If user searches up a movie then it will take them to the movie page

Login:

1. This page requires for user to enter their password and username
2. Once user has entered their account information it will take them to their account page

Checkout:

1. Checkout page processes the customer payment and requires for the user to be logged in to make a purchase, it also displays available payments that are allowed
2. Checkout page displays current reward points and allows users to use their reward points if able to
3. It displays the available seats in the theater and allows for the user to choose their seats
4. Checkout page also displays deals on food and allows for users to preorder food so that it will be ready for them once they are at the theater
5. Once user has completed their purchase they will displayed with their tickets which is available for download to print, and is also sent to their email address for confirmation of purchase

Account Page:

1. On account page it is displayed with locations near the user which allows them to change to whichever location they prefer
2. Account page displays users information
3. Allows for users to change their user information if there are any changes to their account
4. Allows users to return to home page

Help Line:

1. Help line displays the AI chat which helps customers with general questions, the customer service phone number, and common questions
2. Help line also has a refund page which allows users to go to the refund page and request a refund for a movie before showtime has started
3. Allows users to go back to home page

Refund Page:

1. Displays a message that tells the user what to tell the operator when they call in order for them to get their refund
2. Displays helpline phone number
3. Customers will receive a receipt in their email once refund has been completed

Employee:

1. Allows for the employees to change the time displayed on movies
2. Allows the employee to change the price of the movie tickets
3. Allows the employee to add and remove movies
4. Allows the employee to give refunds manually when requested by the user
5. Allows the employee to change the given location on the movies

4.8 Development plan and Timeline:

Nanci Cardenas Martinez: Architecture diagram and Description of Operations
https://github.com/nancicardenas/CS250.git
Emillio Anthony Cecena: UML Diagram, Title Page, and System Description
https://github.com/Emillio04/CS250-Assignment-2.git
Matthew Washington: UML Diagram description / Test Cases
https://github.com/mwashington7220/cs-250-assignment-2.git

Timeline:

|———————————|———————————|———————————|

Start	UML	Testing/Adding features	Release
Architecture Development	Implementation	We debug and implement features based of off testing	

5. SDS Test Plan

excel reference sheet

Test 1. In this first test, we are testing how well the movie search bar works. During this test, we are not only testing if it will take you to the movie page but also ensuring that the spell check works how we will know if the spell check is working if it is showing movies with similar spelling in the drop-down “suggested” menu.

Test 2. During this test, we are making sure that firstly user info is being displayed and secondly it is displaying the correct user info. How we will test this is by having the user login and go onto the user info page. By doing this we will be able to see what is being displayed and ensure that the right user information is being displayed properly.

Test 3. This test is meant to make sure that movies are being displayed correctly. So we need them to display correctly on the home page under the different tabs that sort by genre, or if the movie is trending. The movie also needs to display correctly on the correct movie page, an example of this is that you wouldn't want "Kung Fu Panda 4" to display on "Godzilla Minus One"'s movie page.

Test 4. This test is to make sure the drop-down bars and menus need to be working better. Drop-down bars and menus are very common in our service, such as in the search bar, login page, or at checkout page when selecting the amount of tickets you want. How we will test this is by testing everything that uses a drop-down bar or menu and making sure it is displaying properly and is working as well.

Test 5. This test checks to make sure that you can delete an item from your cart that you may have accidentally added, or just no longer wanted. We will check this by adding a random assortment of items into our cart, such as tickets for different movies at different theaters, or food items such as candy or popcorn. Once we do this we will try to remove them, we will check this by clicking on delete for an item and then refreshing our cart page to make sure they are actually deleted. We will also be checking the payment afterwards in the testing process to make sure that no one gets charged before rolling out this feature.

Test 6. This test is to check that add to the cart function is implemented correctly. This requires for a user to have an account and also for them to have already selected a movie along with the information of the date, time, location, and seat. We will test this by having the user select a movie, then fill in information that is needed to add to cart, and then select "add to cart". We will do this for multiple movies to see if it updates correctly into the shopping cart . We expect for the correct movie to be added to the cart and update as the user adds more movies to their cart.

Test 7. This test checks that user login is a success. The test requires the user to already have an account created. We will test this by having the user enter their username in the "username" section and then password in the "password" section, the user will then click login. We expect for the user to successfully login and be prompted to the two-factor authentication where they will then need to perform those further steps.

Test 8. This test checks that logout is correctly implemented. The test requires the user to already be logged into their account. We will test this by having the user select the dropdown in the far right of the application and select "logout". We expect for the user to be taken to the login menu page.

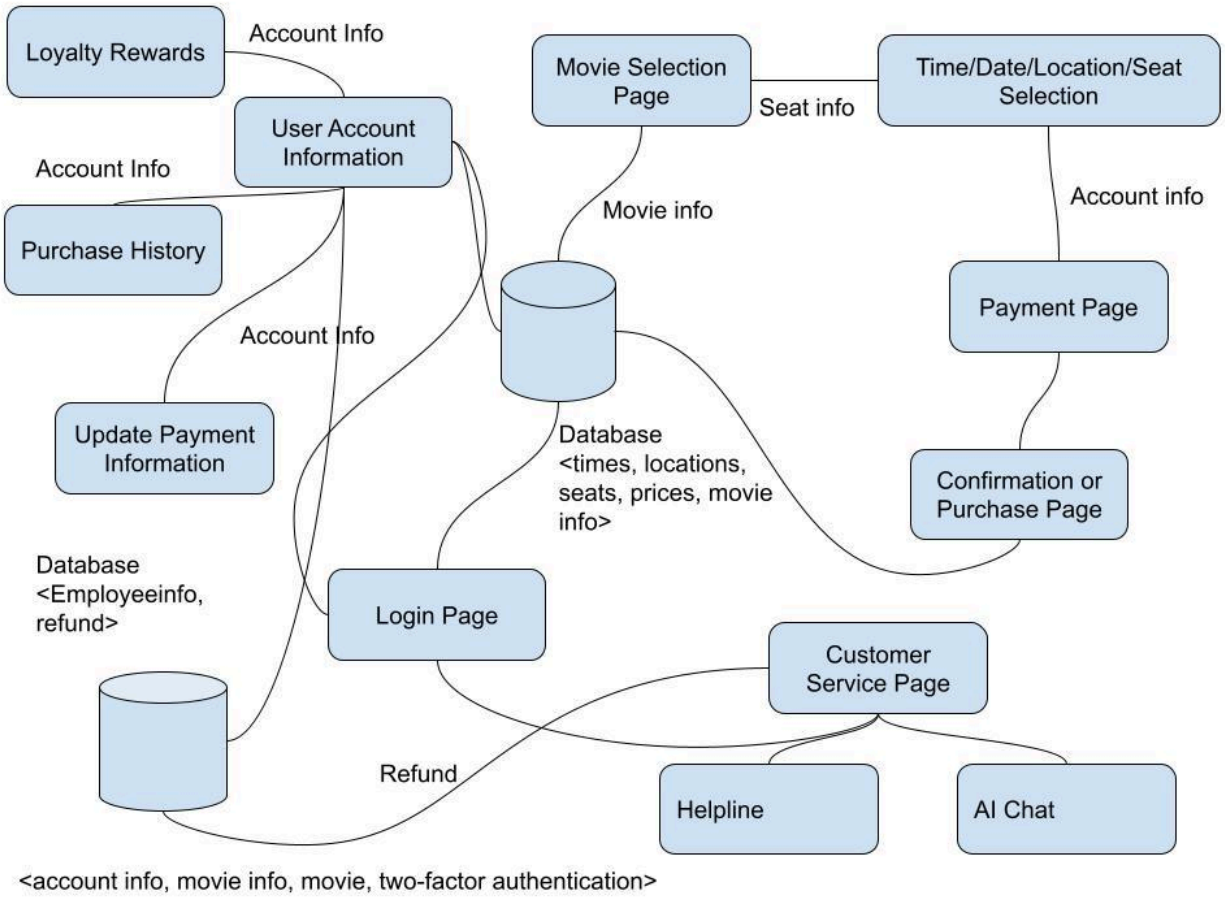
Test 9. This test checks if the user imputed language is correctly stored in the string variable. The user must have an account and have chosen a language. We will test this by having the user input

the language and test that the language is translated. We expect for the string variable to be correctly stored.

Test 10. This test makes sure that the integer variable that holds the rewards points is correctly incremented to the right amount based on the customer's purchase. We will test this by having a customer complete a purchase and then have their points awarded for the purchase on the receipt page. We expect the results of this test to show the customer the correct amount of points earned for the purchase and have it correctly incremented into their current reward points.

Estimated Test Development and Execution Time
Estimated Number of Tests: 200
Average Test Development time: 2.5 (hours/test)
Estimated Test Development time: 500 (hours)
Average Test Execution time: 2 (hours/test)
Estimated Execution time: 400 (hours)
Estimated Regression Testing (50%): 200 (hours)
Total Estimated Test Execution Time 600 (hours)

5.2 Data Management Strategy



We chose three databases for our program. The reason as to why we chose three specifically is because there are three main points of data in it. The first being user data the reason as to why it's SQL is because user data will not be rapidly changing or be unrelated at all. As to why we know this is because to even login and change one's user data they need to go through a two step verification process. Next for employee services that also will not be changing all that much so there is not a reason to use a more difficult language such as NoSQL. But for movie data we do need to use NoSQL because we are accounting for the variability in what goes on into a movie. Such as seat availability changing and new theaters or times. There is a lot that goes into just booking a movie so we found it best to do NoSQL because of the amount of varying data to display the correct information on the movie page.

6. 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