Megarama Theaters

Software Requirements Specification

Version 4

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
3/13/24	Version 3	Group 3	Third Revision
3/27/24	Version 4	Group 3	Fourth 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

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 developement of “Megarama" project of a Ticketing System with Live Data Manager. It emphasizes primarly 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

TS	*Ticketing System*
LDF	*Live Data Feed*
DBMS	*Database Management System*
CRM	*Customer Relationship Management*
GUI	*Graphical User Interface*

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 caplable to run a web browser without interuption that will 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 quest, with live view of futured 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 parched 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 slection 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

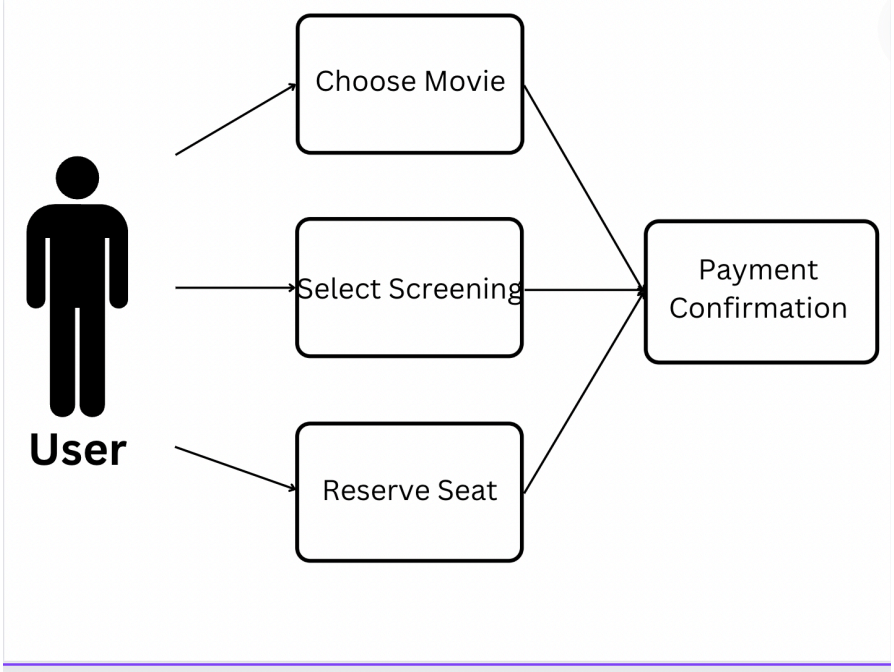
- The information given back will be an email of a ticket bought with a timestamp to display at the theater, purchase confirmation number and in case reverse transaction funds are given back with confirmation number.

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.

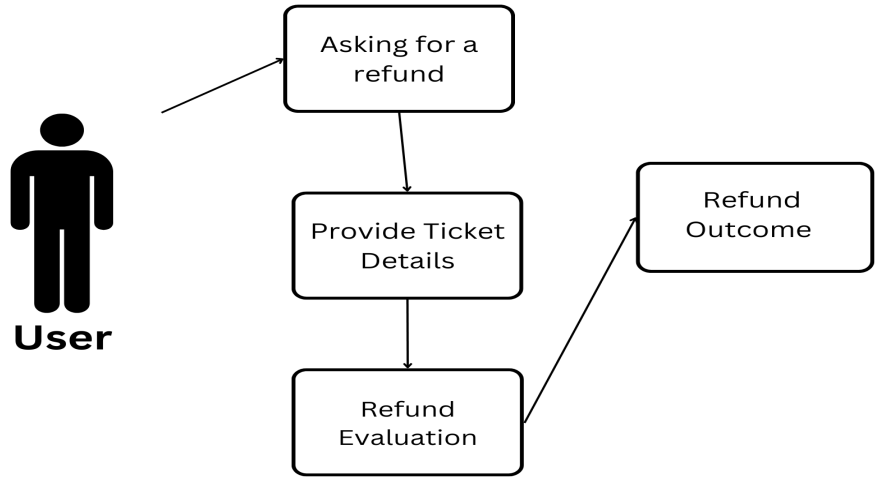
3.3 Use Cases

3.3.1 Use Case #1: Purchasing ticket



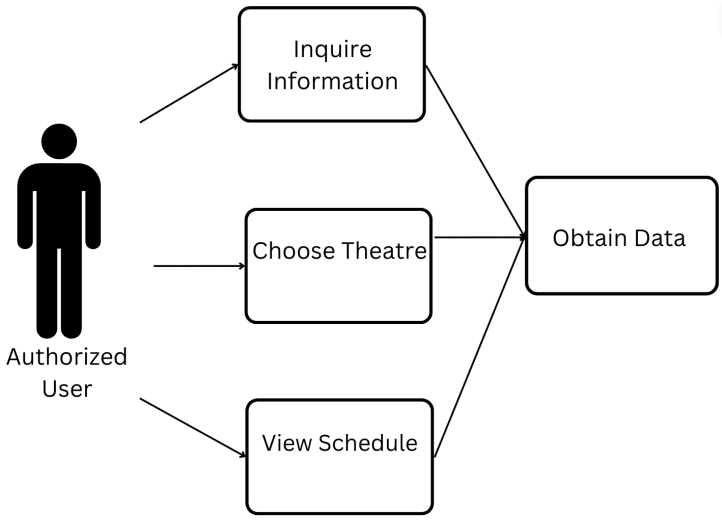
-User: A customer who is buying tickets.
-Choose Movie: A step where the user chooses the movie he wants to see.
-Select Screen: The user chooses a screening time.
-Reserve Seat: The user reserves a seat in the viewing room of the theater where the movie is being shown.
-Payment Confirmation: After entering 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 atribbutes 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 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-email 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.

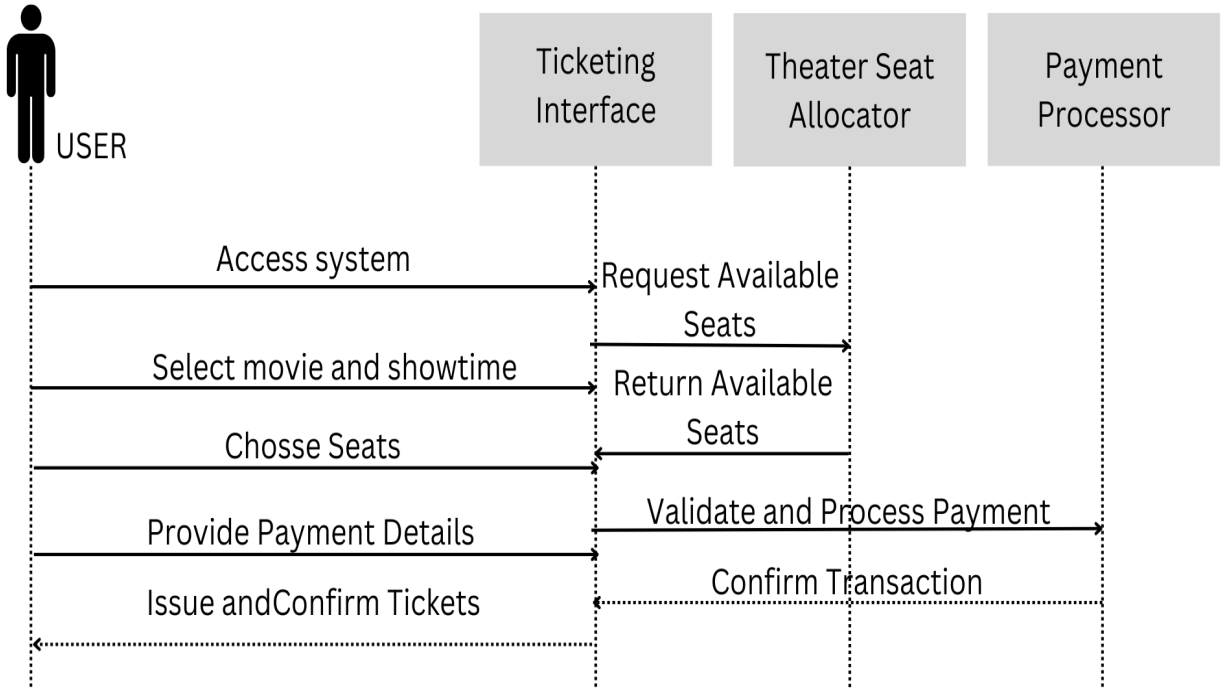
3.9 Other Requirements

Catchall section for any additional requirements.

4. Analysis Models

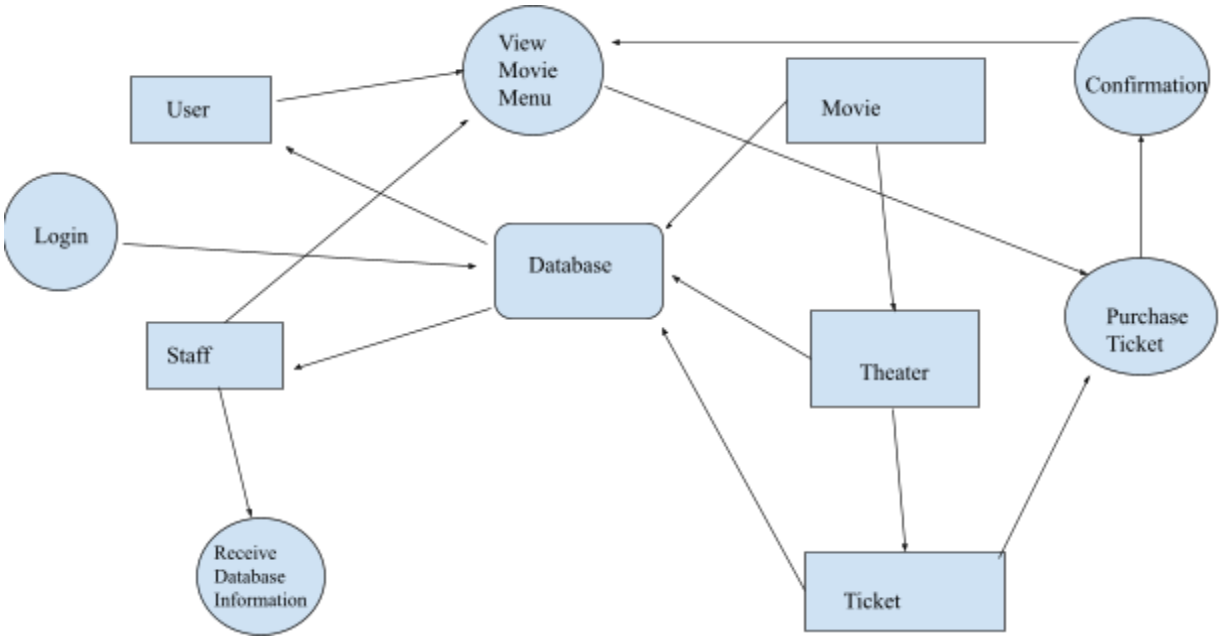
List all analysis models used in developing specific requirements previously given in this SRS. Each model should include an introduction and a narrative description. Furthermore, each model should be traceable to the SRS's requirements.

4.1 Sequence Diagrams



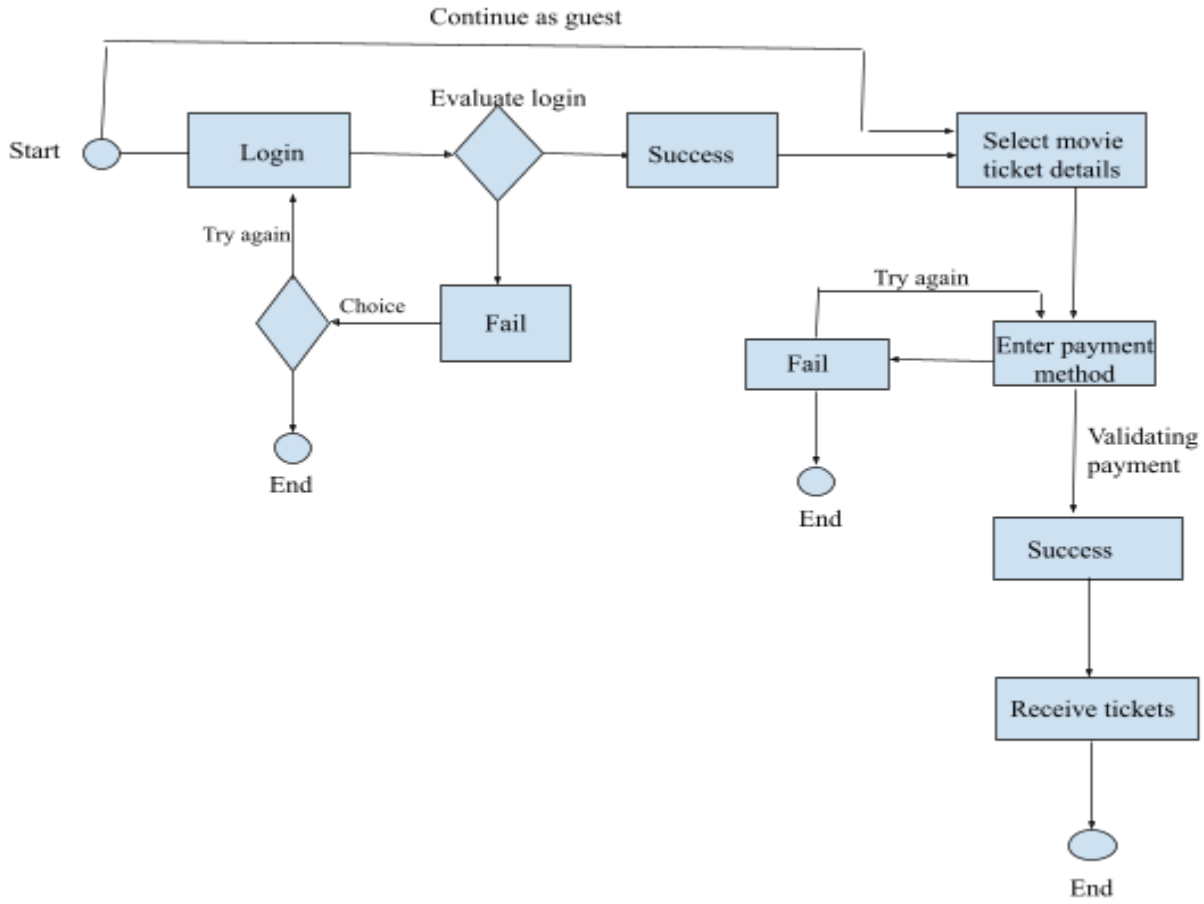
Overview: This sequence diagram describes the steps realized by a user and the whole process of what is happening when he is trying to purchase a ticket. Messages are sent to the Ticketing Interface, which then communicates with both the Payment Processor and the Theater Seat Allocator to finalize the 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)



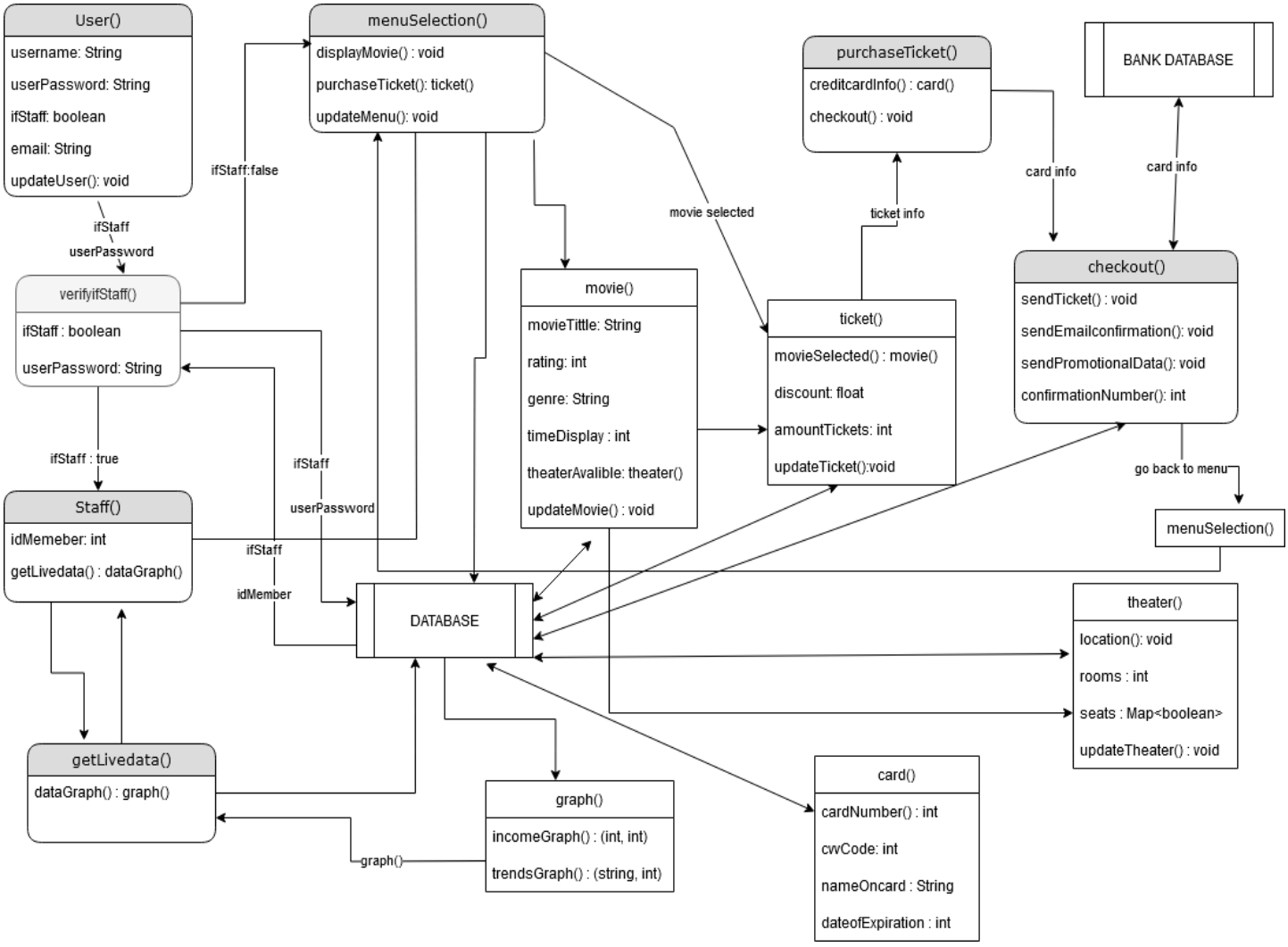
Overview: The data flow diagram represents the flow of data through the movie ticket system and the main flow of process of the software. The database contains user information and verifies whether the user logging in is a customer or staff member. It also has movie information, including the name of the movie, show time, and theater location. The user and staff can see the movie menu. Once they select the movie and tickets, they can purchase tickets. Once their payment is verified, their tickets are confirmed. After getting confirmation of tickets, the user is taken back to the main page with the movie menu.

4.3 State-Transition Diagrams (STD)



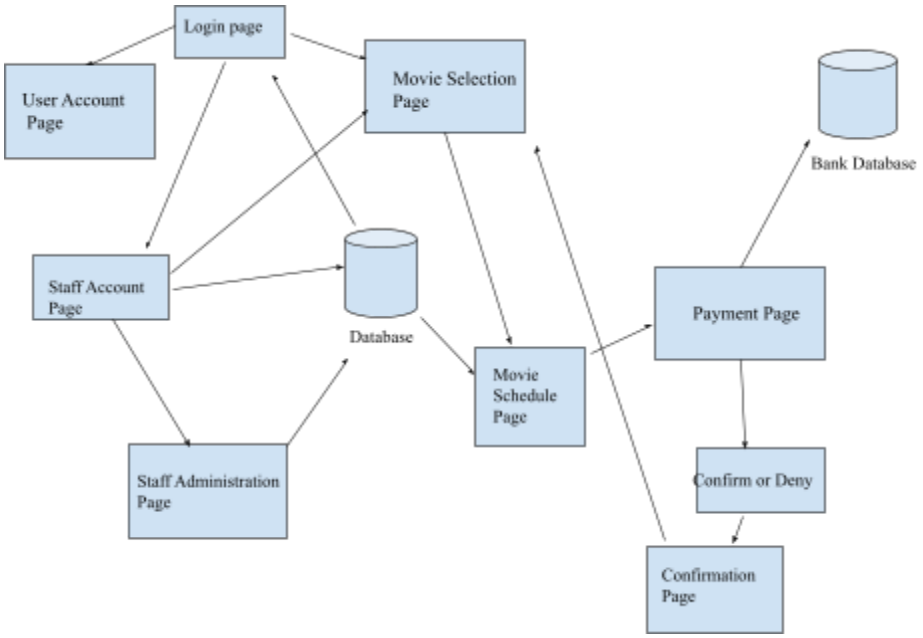
Overview: This diagram describes the behavior of the system, showing how a user interacts with the system and what state they are in at each step, as well as the transitions between each state. At the start, users can either login to the system or continue as a guest. If their login fails, they have the option to try again. They are then shown the movie options. Once they choose a movie, they will choose a time and theater location. Once these details are selected, they will be shown a map of the theater and choose an available seat(s). They then will enter their payment information. Guest users will have to enter their information, and users with an account will have to select the payment method they would like to use. The payment gets validated, and if it fails, they will have the option to enter a new payment method. Once the payment has been approved, the purchase will be a success and the user will receive the tickets. They will then be taken back to the main page.

4.4 UML



Overview: This system is a user logs in into the system database by the User() class verifies the login credentials in verifyStaff() method, then will add if this user is a staff admin or a normal user. Then first a normal user will be received with the main menu in menuSelection() method where they can select the movie() class to be viewed in an interactive and visual menu. Once the movie, seat() class and quantity of tickets which is an attribute of ticket() class is selected it will prompt to make a payment in the purchaseTicket() method, the system database will search for saved credit card credentials which are attributes if the card() class and add to make a payment and communicate with the bank database in the checkout() method. 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 which again is the menuSelection() method. The staff user which its own class Staff() can purchase tickets as well but also will have access to retrieve data from the database in the getLivedata() method which will give a graph of trends on purchases and transactions made in a lapse of selected time which is represented in a graph() class.

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 database has information about the type of user and what they can access. Both the user and the staff can access the movie selection page, but only the staff can get to the administration page where they can get data about the movies, theaters, and ticket sales. Once you select the tickets for the movie, you go to the cart and are taken to the payment page. After entering the payment details, the payment is pending verification, and once it is verified, your tickets are confirmed and you get taken back to the main movie selection page.

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.

4.6 Testing

4.6.1 Test 1

Test Case ID: SelectSeat_1
Test Case Description: This test verifies that the user can select only available seats and add them to cart. This will be done using integration testing. The individual modules will be combined and tested as a group to see if these steps can occur and work successfully. This will be done

using the integration approach. Modules will be tested as they are available. Higher level modules such as login can be tested first to increase efficiency.
Preconditions: The user must select the specific movie, time, and location. Then they must click on the seats they would like on the map of tickets and add them to cart.

Step Number	Step Description	Test Data	Expected Result
1.	Select movie, time and location	“Dune: Part Two” 5:45 PM La Jolla	Taken to map of theater
2.	View the seating and select available seats	Seat 4 C	User can select ticket and add to cart
3.	View the seating and try to select a non-available seat	Seat 5 E	User cannot select seat or add to cart
4.	Go to cart		Available ticket is in cart

Post Description: The tickets were successfully selected. Only available seats were added to the cart and the unavailable seats were not able to be clicked on or added to the cart.

4.6.2 Test 2
Test Case ID: UpdatePassword_1
Test Case Description: This test verifies that the user can change their password to their account. Once the password is changed, only the new password should allow the user to login, not the old one. This will be done using integration testing. The individual modules will be combined and tested as a group to see if these steps can occur and work successfully. This will also be tested using the incremental approach where higher level modules are tested first and modules are tested when available.
Preconditions: The user must log into their account and press the account icon. They then must press change password and enter their new password and confirm their new changes.

Step Number	Step Description	Test Data	Expected Result
1.	Login to account	UserID: EmmaT Password: Password1	Successfully login
2.	Click on account icon and select Change Password, confirm new password		Prompts user to enter new password and confirm changes
3.	Re-login with new password	UserID: EmmaT Password: Password2	Successfully login with new password
4.	Re-login with old test data	UserID: EmmaT	Password denied, try

		Password: Password1	again

Post Description: Password was successfully changed. Old password does not work for login, only new password

4.6.3 Test 3

Test Case ID: SeatType_1

Test Case Description: This test verifies that the correct pricing is assigned to the premium and regular seats. Users will be able to tell the difference between what seat they are choosing on the map with a key to distinguish regular versus premium seats. Integration testing will be used to test the different modules and the data transfer between them.

Preconditions: The user must select the specific movie, time, and location. Then they must click on the seats they would like on the map of tickets and add them to cart.

Step Number	Step Description	Test Data	Expected Result
1.	Select movie, time and location	"Dune: Part Two" 7:00 PM La Mesa	Taken to map of theater
2.	View the seating and select a premium seat	Seat 2 A	User selects the premium seat and adds to cart
3.	View the seating and select a regular seat	Seat 15 D	User selects the regular seat and adds to cart
4.	Go to cart		The Row 2 ticket (premium) is $18.00 and the Row 12 ticket (regular) is $12.00

Post Description: The pricing of the tickets were right.

4.6.4 Test 4

Test Case ID: UpdatePayment_1

Test Case Description: This test verifies that the user can update their payment information, adding or deleting a new payment method. This will be done using integration testing.

Preconditions: The user must login and select the user account icon. The user must select the option to update their payment method. They can replace their old payment method by deleting the one currently in their account, or add an additional one.

Step Number	Step Description	Test Data	Expected Result

1.	Login	UserID: JamesR Password: orangeCat9	Successfully login
2.	Select the user account icon		User is taken to their account information page
3.	Select Update Payment Method and confirm new changes	Add Card: James Robinson 4815 2234 4394 0398 Exp: 12/26 CVV: 123	New card is saved into account
4.	Select Update Payment Method and Delete Payment Method	Delete Card: James Robinson 4926 2252 2804 3782 Exp: 11/25 CVV: 202	The old payment method is deleted from the account

Post Description: The payment methods were able to be updated. New card was able to be added to the account and the old card was able to be deleted.

4.6.5 Test 5

Test Case ID: DatabaseCapacity_1

Test Case Description: This system test is to verify the number of live users that the database can have logged in and are active in sending and requesting data packaged to the server. This a test to be able to have a margin of the capacity of bandwidth of the database.

Preconditions: Live accounts must be login into database and are active receiving packages from database

Step Number	Step Description	Test Data	Expected Result
1.	Database environment is prepared, server and accounts are logged in.		Database in idle
2.	Metrics are added to the monitor network utilization.		Database in idle
3.	Visual Studio Load Test is launched and runs tests to the database.	51,000 users at same time	Database will struggle to send and receive packages to all users.

4.	Metrics are collected and tested again with less users	49,000 Users at the same time	Database will be successful to send and receive packages to all users.

Post Description: Test on database bandwidth was tested and the limit on the hardware is the total of approx. less or equal of 50,000 users prompting the database at the same time, further expansion of capacity can be implemented.

4.6.6 Test 6

Test Case ID: DatabaseTime_1
Test Case Description: This test is to track the user experience thru the website menus while retrieving the information form the database to display the website correctly and ensure in a quick manner the whole system process of browsing and purchase a ticket.
Preconditions: Users must be logged in and go through the process of purchasing a ticket and go back to the main menu.

Step Number	Step Description	Test Data	Expected Result
1.	Database environment is prepared, server and account are logged in.		Database is in idle
2.	Server website bandwidth speed monitor is been set		Speeds are at idle
3.	Test user will undergo through all the process of purchasing a ticket while a monitor will register speeds of up and down	Prompts form the user while going through menus	Speeds are consistent and data is displayed correctly
4.	Collected metrics and timed user purchasing cycle is recorded	User prompts database for aa ticket purchase	A successful purchase in a quick and timely manner.

Post Description: Test showed quick speeds on data sending to user when this was going through websites menu and was able to purchase a ticket in a quick and timely experience thru the website.

4.6.7 Test 7

Test Case ID: DatabaseGetData_1

Test Case Description: This test is to verify that data received when prompted by the upper lever user called staff through the website and retrieved data is on a consistent and accurate consistency.

Preconditions: Staff user must be logged in and go to the staff menu and retrieve live data from the server.

Step Number	Step Description	Test Data	Expected Result
1.	Database environment is prepared		Database at idle
2.	Staff user is logged in	Staff123 PaswordsStaff	Database recognize this log in as a Staff User
3.	Staff user will prompt database thru the website staff menu to retrieve live data	Website prompt to retrieve live data with the	Database receives prompt to retrieve data and starts the data gathering into file
4.	Staff user will store live saved data		Database will send encrypted file to be decrypted by website and be able to download .xlsx file
5.	Collected data will be analyzed for accuracy and inconsistencies		Data is accurate and correct data is displayed when opening file

Post Description: Test showed that the database is able to retrieve live data of purchase and tendencies into a file when prompted by an upper class user called staff in an efficient and accurate dynamic.

4.6.8 Test 8

Test Case ID: LoginSuccess_001

Test Case Description:This scenario checks that a person can sign in to the cinema ticket booking platform using their username and password. The aim is to confirm the login process making

sure that individuals can reach their accounts to see and handle their reservations. The platform must validate the login details. Permit entry only if they align with the stored data. A successful sign in will be signaled by the user reaching the page or their personalized dashboard.
Preconditions: User must use its correct login credentials as they register on the webpage and is still valid on the server.

Step Number	Description	Test Data	Expected Result
1.	Navigate to the login page of the theater ticketing system	Url of the system	login page is displayed
2.	Enter valid username and password	Username UserTest Password: CorrectPassword	Fields are populated with the test data
3.	Click on the 'Login' button		User is logged in and redirected to the homepage of the ticketing system

Postcondition: The test passed as user utilized the same credentials created when registering on the website and stored user values on database were compared and verified, to be able confirmed a successful login and redirection to the movie menu selection

4.6.9 Test 9

Test case ID: PaymentSuccess_002
Test Case Description: This test scenario is designed to confirm the functionality of the payment gateway integration, within the theater ticketing system. The main goal is to guarantee that users are able to finalize a transaction using payment options. The procedure includes validating the accuracy of payment details like credit card information and ensuring that users receive a confirmation once the transaction is completed. The anticipated result includes ticket issuance, a decrease in seat numbers and an accurate update, in the sales records of the system.
Preconditions: The user is logged in and has added tickets to cart. The user goes to checkout and enters their payment information.

Step Number	Description	Test Data	Expected result
1.	Navigate to the login page of the theater ticketing system	Preselected Tickets	Checkout page is displayed
2.	Enter valid username and password	Credit Card Payment Methods Details	Payment fields are populated

3.	Click Buy Tickets		Payment is processed ,and a success message is displayed with a confirmation number
4.	With new tickets in cart, click Buy Tickets	Credit Card Payment Method Details (invalid)	Payment fails and user is asked for another payment method

Postcondition: The test passed and the valid payments went through and the invalid payment methods were denied.

4.6.10 Test 10

Test Case ID: MovieSelect_003

Test Case Description: This trial is set up to ensure that a person can smoothly navigate the theater ticket system to pick a movie and a preferred showtime. The features being checked cover showing the list of movies picking a movie selecting a showtime and moving on to choosing seats. It's important that the system updates in time to show the showtimes based on the chosen movie and prevents choosing times that have already passed. Successfully finishing this trial results in a map of seats where users can select their seats. Unit testing will be used for this test where the smallest units of code will be tested.

Preconditions: The user logs in or continues as a guest and is on the movie menu page. The user selects a movie.

Step number	Description	Test Data	Expected result
1.	Navigate to the movie selection page		List of Available movies is displayed
2.	Select a movie from the list	Movie Name: “Epic Adventure”	The selected movie details are displayed
3.	Proceed to select a showing time	Showing time :7 PM	User is taken to the seat selection map for the chosen time
4.	Select the movie seat	Selected seat 7 B	User is taken to the confirmation

Postcondition: The test passed. The correct movie information was shown when selecting a specific movie.

5. Data Management

5.1 Design Choices

The system will use a SQL database, which is advantageous compared to NOSQL. Data is structured using SQL schemas that define the tables in the database. There are relationships between the data among multiple tables and these relationships can be defined using keys. This is an effective way to organize and structure data. Additionally, SQL systems have very strong security systems, compared to NOSQL that can sometimes be weaker. The SQL systems have built in features for data encryption and user authentication. It is a scalable system that can handle large volumes of data. Large quantities of data can be retrieved and manipulated very quickly and efficiently. There are also backup and recovery tools that can help recover data in many different cases.

5.2 SQL Diagrams

User Accounts

idNumber	email	username	userPassword	isStaff	paymentInfo
000001	jroberts@gmail.com	justinroberts	Password1	yes	Visa ending in 0390
000002	chrisj10@gmail.com	chrisjones1	Password2	no	Visa ending in 8834
000003	emmasmith@yahoo.com	emmasmith22	Password3	no	Mastercard ending in 3927
000004	jackreese@icloud.com	jackreese	Password4	no	AMEX ending in 4882

Movies

movieTitle	rating	genre
Hunger Games	8.9	Dystopian Fiction
Avatar	6.7	Science Fiction
Titanic	7.2	Romance/ Drama
The Lion King	9.1	Animation/ Childrens Film

Theaters

theaterID	location	rooms	deluxeSeats
0001	El Cajon	12	no
0002	La Mesa	9	yes
0003	Pacific Beach	10	yes

Showtimes- Pacific Beach Location (example)

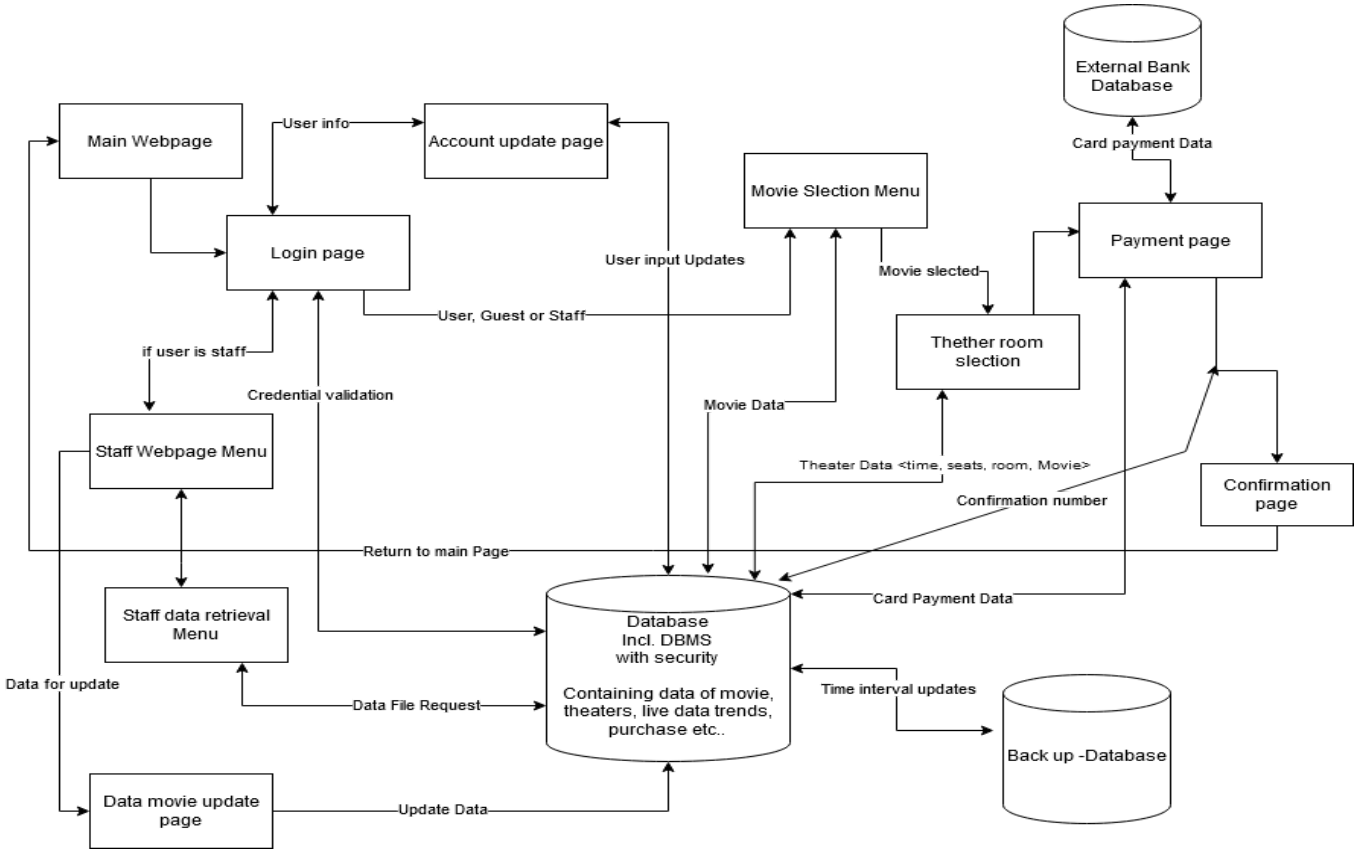
movieTitle	roomNumber	time	Runtime (minutes)	regularSeats	deluxeSeats
Hunger Games	1	4:30 PM	130	4	10
Hunger Games	3	7:30PM	130	7	12
Avatar	2	5:30 PM	117	16	14
Avatar	7	9:00 PM	117	11	13

5.3 Data Management Strategy

The Data stored will be managed by the DBMS in one location only and there will be another location for backup, this will result in a more efficient way to use the storage capacity. The application interacting with the DBMS will be the webpage of the movie theater. The DBMS will coordinate updates and data changes to the database, users and programmers can access this data upon request with hierarchy privileges and relationships. DBMS will ensure that the database will support both business and activities goals. This database will have stored the values of the data of the movie theater's information such as user credentials, movies that will be displayed, information on trends and purchases and live information on the available theaters capacity / rooms. The data manipulation language used will be SQL, which will be used in a standalone way for quicker access of data in tables as well for easy display when requesting this from the DBMS. Users will be able to interact with the database using the webpage calling the DBMS and Staff users will be able to request live data stored in the database through a web page that requests DBMS for data stored in the time requested. Multiple users will be allowed to access the same database using the application of SQL DBMS offers benefits such as scalability, high performance, data security, reliability and availability, and handle large amounts of data and traffic. Backup and recovery services will be available by the use of a backup database that will be updated in time intervals. There will be active security scans in every data request that the DBMS manipulates to ensure that certain rules are followed with regard to data in the database and any changes that are made in the data. There will also be a data dictionary with the movie theater terms to have a detailed description of the data used on the database and quick update. Webpage application when requesting data will follow a logical path presented in the graph below. This is the updated flow of data and the data management and architecture of the system.

Webpage will interact with the DBMS in user validation if it is a staff, regular user or a guest. Then will be sent depending on the user hierarchy to the specific menus intersecting with the database for information to be displayed on menus. Trusted staff will be able to update movie, teather, time, rooms and prices information to database thru DBMS in an automation way to ensure no corrupted data and secure information is updated correctly. Theater website will be interacting with the DBMS on movie selection, live theater seat selection and to the end of the

purchase, all this data will be stored in the server in a time interval for space management as well a backup server in case of contingencies can be switched and continue the data output and input.



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