2022

Online Ticket System

NOUR MOHAMED GABER 6392
YARA TAREK 6445
SARAH AWNI 6475

Use Case Description

Actors: Customer, Admin

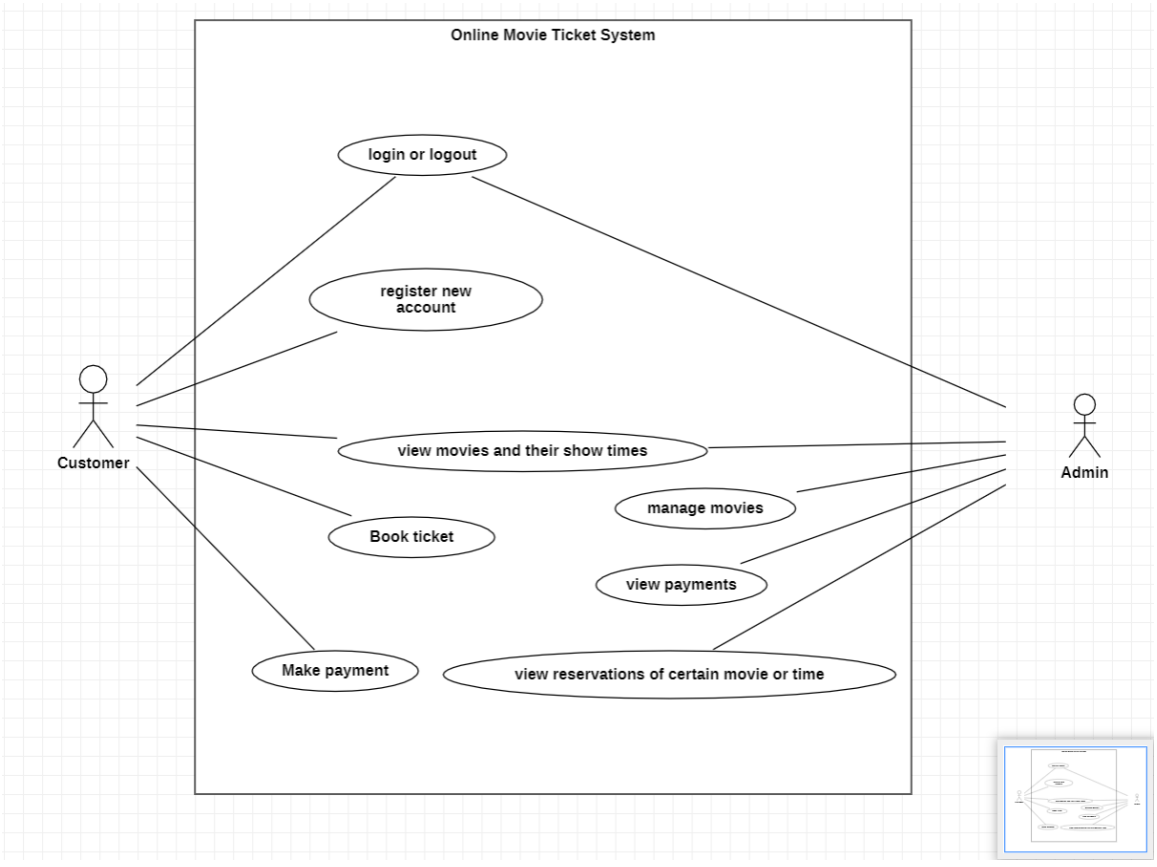
Description: The website will display various movies each in the specified time at which it will be displayed. The customer will be able to choose movie, time, number of seats, and purchase a ticket accordingly. The admin will be able to manage movies, view payments, view reservations report for a certain movie or time.

Data: Customer's or admin's email and password and generated report

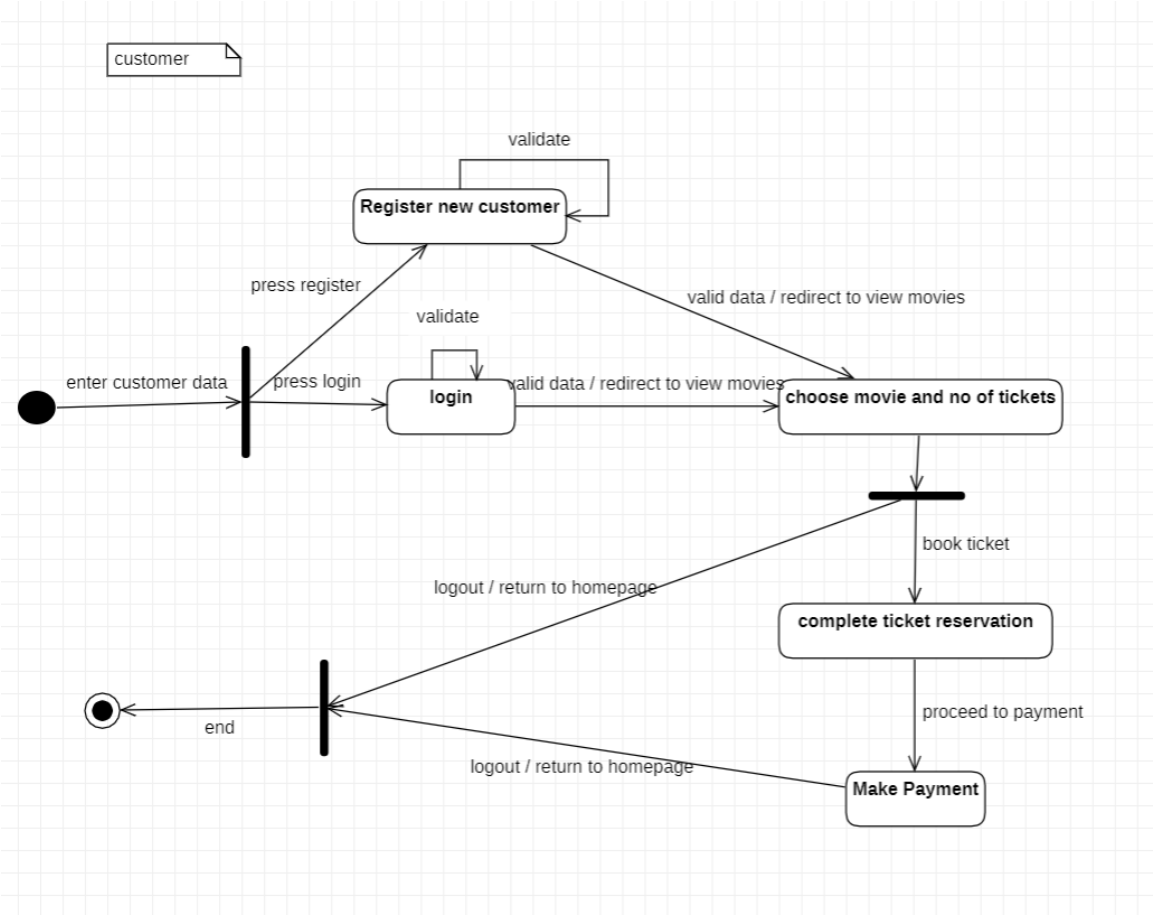
Stimulus: Action taken by customer for ticket reservation or admin issuing system management commands.

Response: Reservation and payment Confirmation for customer and generated reports for admin

Comment: The admin must have the admin's email and password to be able to view the cinema's information.



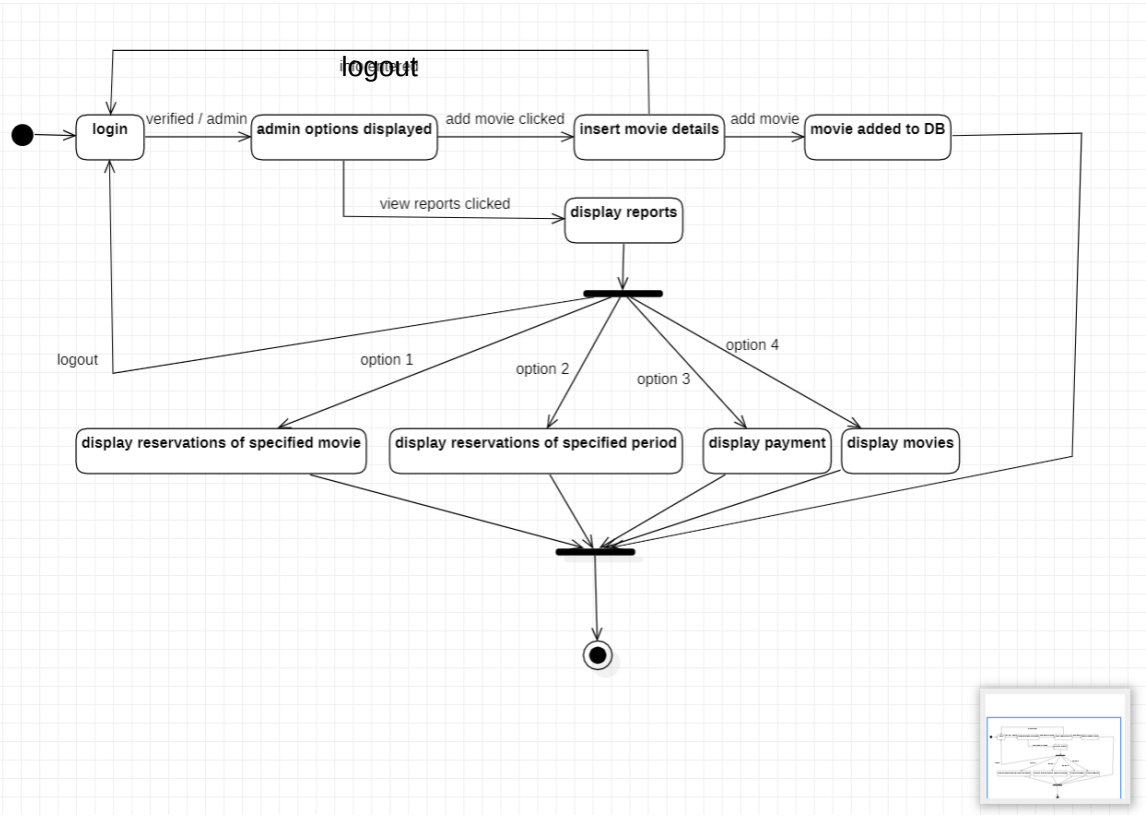
Customer's State Machine Description



State	Description
login	Customer or admin enters email and password and the validation starts.
Register new customer	Customer can create a new account and validation for the email and password will start.
Choose movie and no of tickets	Customer can view and choose movie and number of seats.
Complete ticket reservation	Reservation done by customer will be confirmed
Make payment	Customer will choose to pay using visa or cash to finish booking.

Stimulus	Description
Enter customer data	The user entered email and password either as a customer or as an admin.
Press login	The user pressed on the login button.
Press register	The user pressed on the register button.
Validate	The data of the customer or the admin will be validated in order to login.
Validate	The email and password will be validated to register a new account.
Valid data/ redirect to view movies	When the email and password are validated, the customer will be able to view movies.
Logout/return to home page	The customer could logout from the view movies page without completing the booking going back to the home page.
Book ticket	The user will choose a movie at a certain time and the number of seats.
Proceed to pay	The user will choose the method of payment(cash/visa) so the process of booking a ticket is completed.
Logout/return to home page	After booking the tickets, the customer will logout and will go back to the homepage.

Admin's State Machine Description

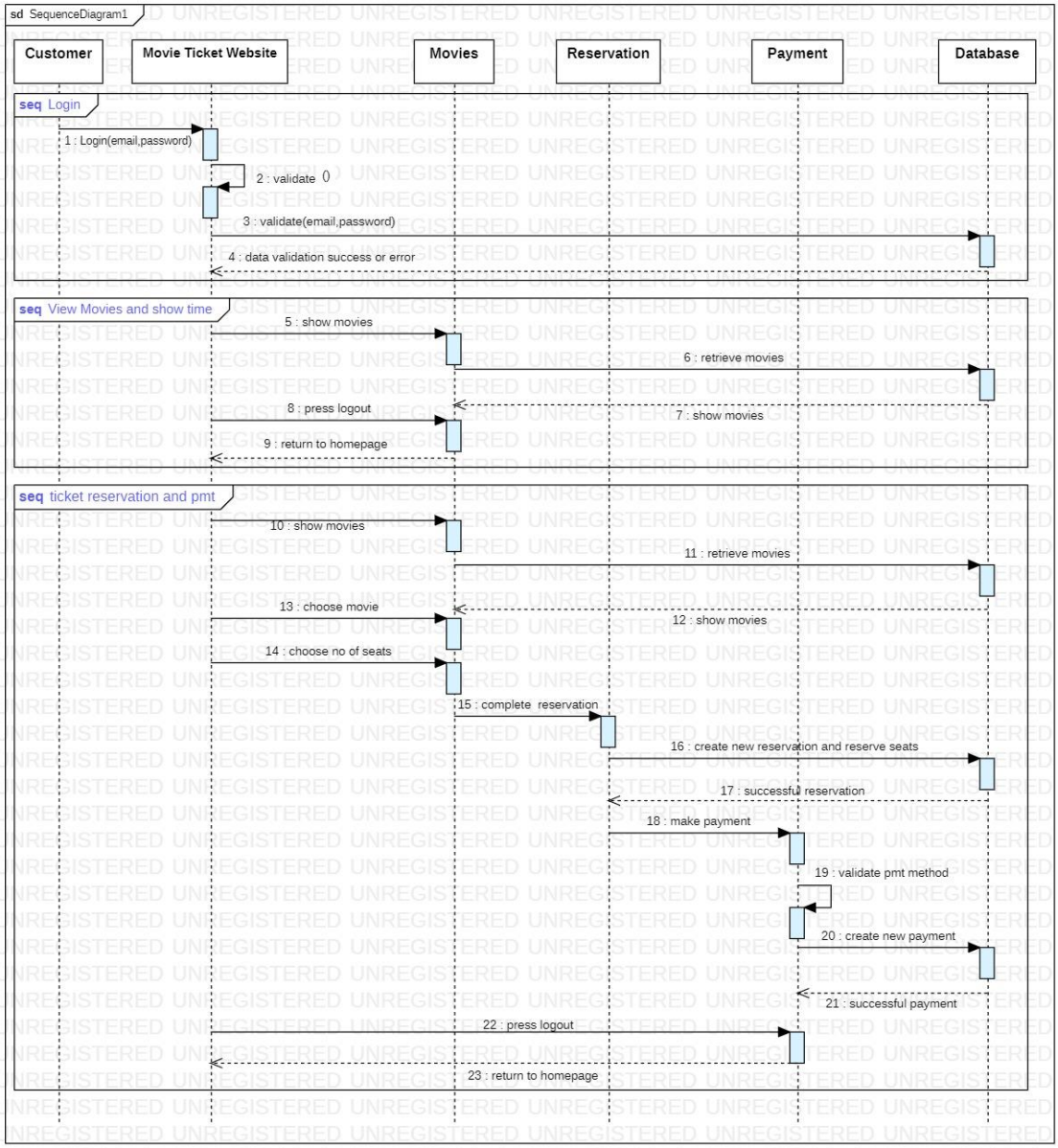


State	Description
Login	admin enters email and password and the validation starts.
Admin options displayed	System lists all functions provided to the admin.
Insert movie details	Admin insert information about a new movie that will be added to the database.
Movie added to the data base	The new movie will be added to the database.
Display reports	Admin should be to view reports.
Display reservation of specified movie	Admin will be able to view a report that shows the reservations of a specific movie.
Display reservation of specified period	Admin will be able to view a report that shows the reservations of movies at a certain interval of time.
Display payment	Admin will be able to view a report that shows the payments .

Display movies	Admin will be able to view a report that shows the movies .

Stimulus	**Description**
Verified/admin	Email and password are verified and admin functionalities will be displayed.
Add movie clicked	Add movie button is pressed
Add movie	The movie is added to the data base.
Info entered	
View reports clicked	The selected button will display the specified report.
Logout	Admin could logout and go back to the home page.

Customer Sequence diagram Description

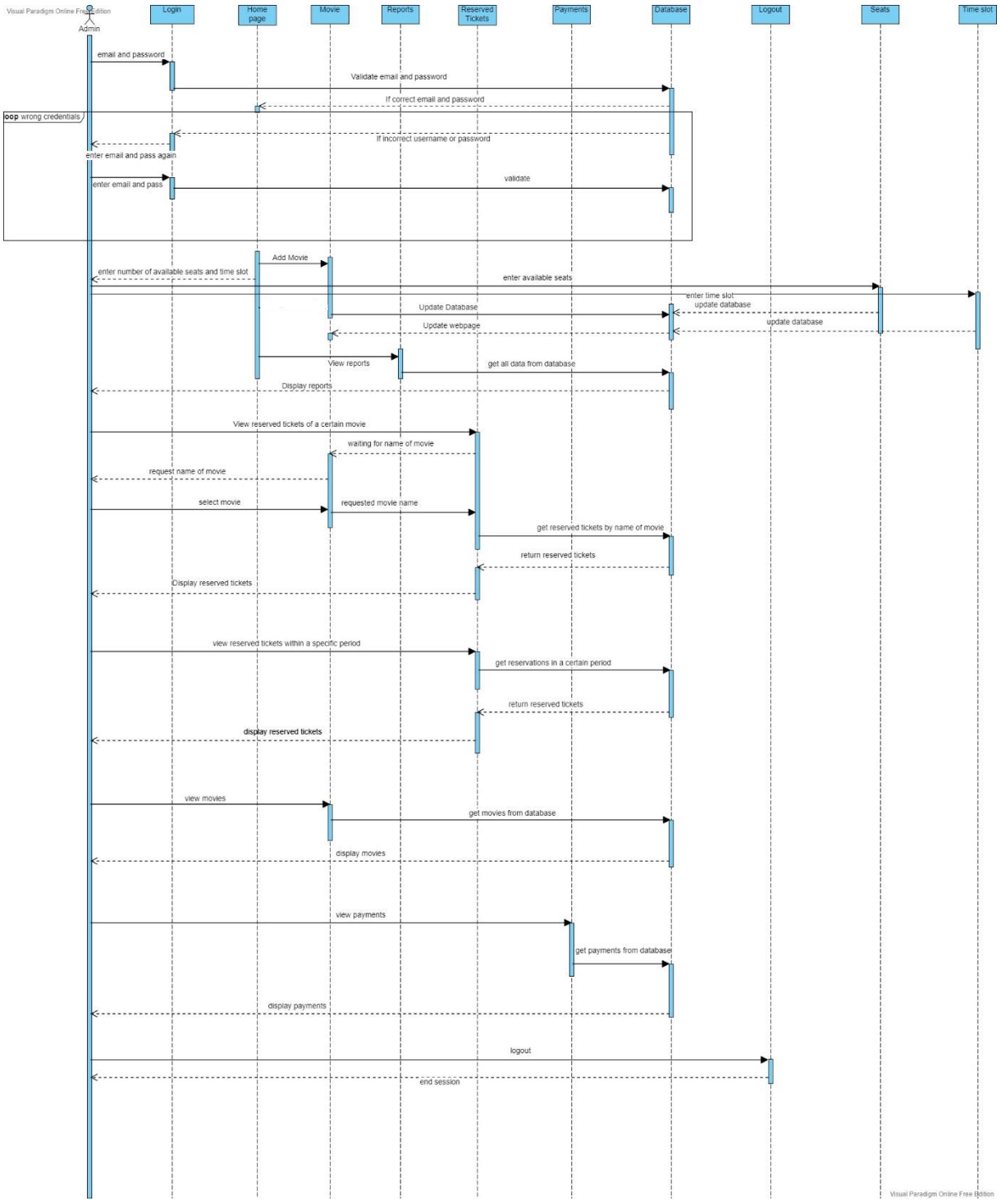


- Customer will enter the email and password which will be validated in the movie website to check syntax errors and then email and password will also be sent to database to be validated. Data validation successfully or error will be sent back to the movie ticket website.
- The website will show movies available which will be retrieved from the data base. If the user pressed logout he will return to the home page.
- Customer could choose movie and number of seats to complete reservation which will create a new reservation and stored in the data base and a confirmation will be sent back. Also a payment tuple will be added to the data

base in which the payment method will be validated first and a confirmation will be sent back from the data base.

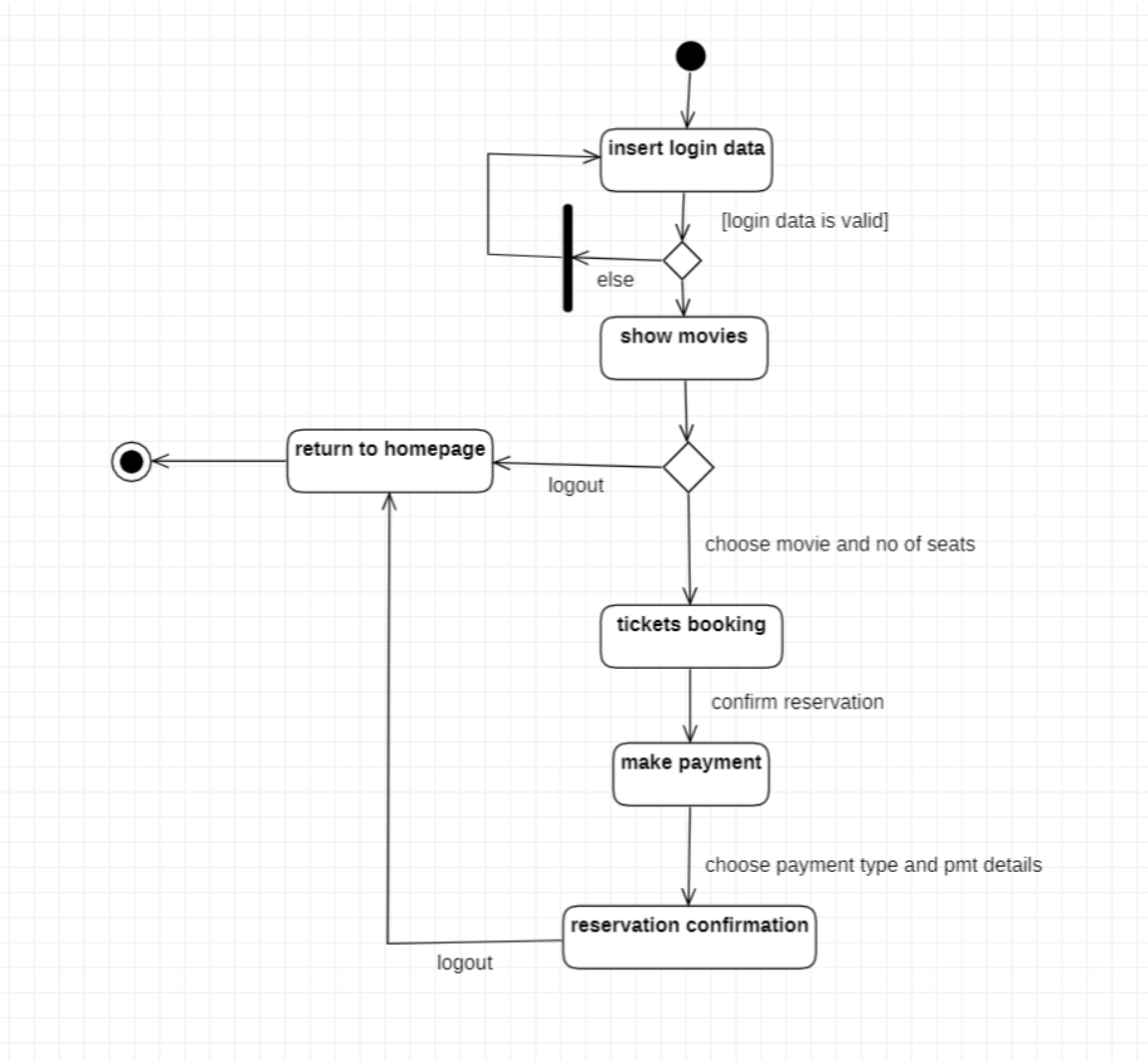
- Then the customer will logout and return to the home page.

Description of admin sequence diagram:

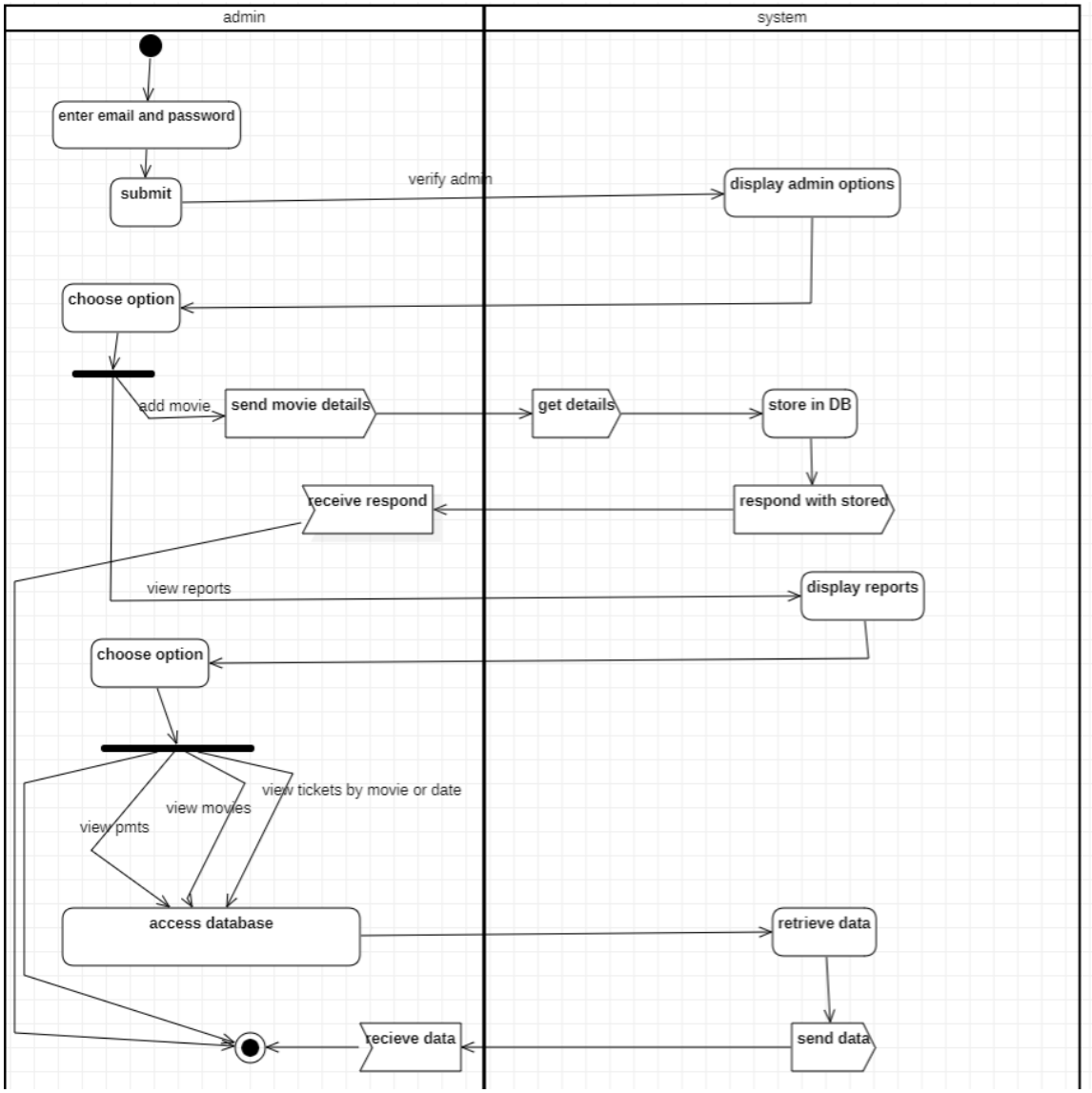


- Admin enters email and password to login
- Email and password will be checked from the database
- If they don't match the data saved in the database, then the admin will be requested to reenter the email and password until a match occurs
- When email and password are checked with the data in the database, and they're matched, then the home page will appear to the admin
- Admin could choose add movie option (option 1)
- This will allow him to add a new movie to the database
- When adding a new movie, the admin will be requested to enter some data about this movie like the number of seats available and the time slot
- Then the database will be updated
- The admin could choose the 2nd option, which is view reports
- Data will be retrieved from the database and a new webpage will appear to the admin
- Now, if the admin chooses the first option, which is view reserved tickets of a certain movie, then the admin will enter the movie name and all the reservations of this movie will be retrieved and displayed to the admin
- If the admin chooses the view the reserved tickets within a specific period, then he'll enter the period (start date and end date) and all the movies within this period will be retrieved from the database and displayed to the admin
- If he chooses to only view the movies, then all movies will be displayed to the admin
- Lastly, if he chooses to display the payments, all payments will be retrieved from the database and displayed to the admin
- Lastly, the admin could logout and the session will be ended

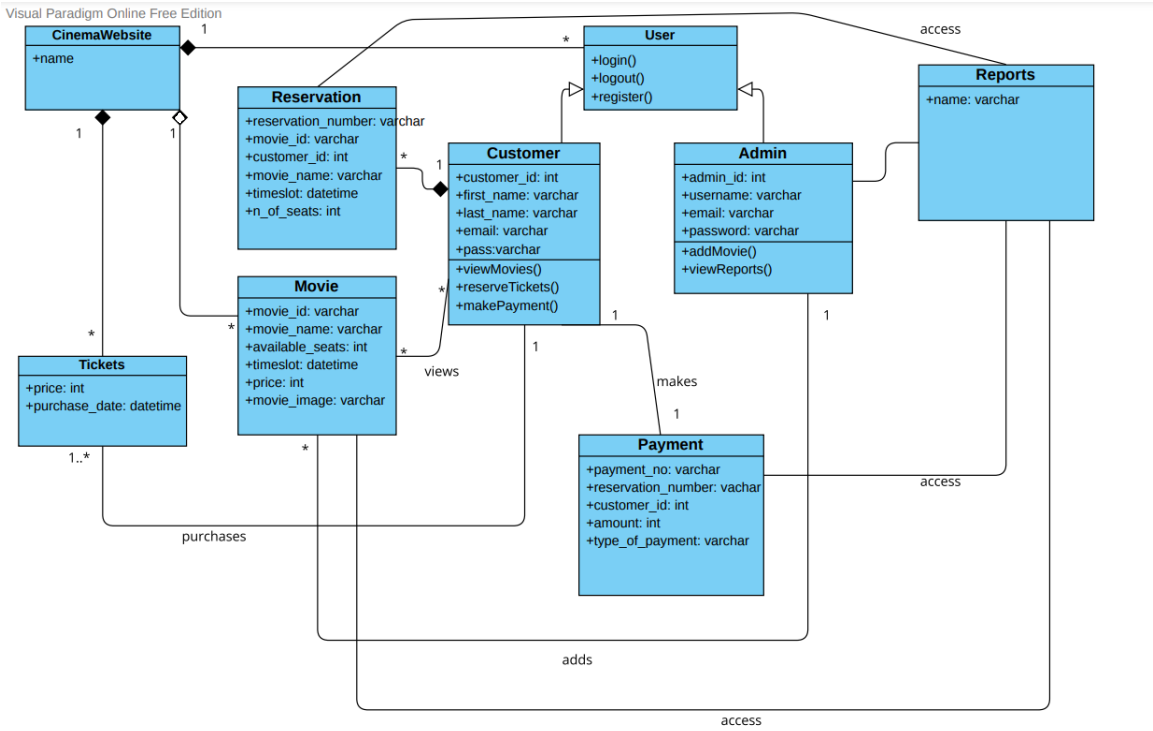
Customer Activity Diagram



Admin Activity Diagram



Class Diagram



The cinema website is used by either customers or an admin. The admin can either add a new movie (a movie can be added by one admin but an admin can add many movies) or view reports on reservations of a certain movie, reservations within a specified date, view payments made or view movie details. This requires accessing reservation, movie and payment tables (reports access these tables).
A customer can view all available movies on the website (a customer can view many movies and each movie can be viewed by many customers), a customer can make many reservations per session and each reservation can be made by only one customer. Each customer makes one payment per session and each payment is made by one customer.
The cinema website has many tickets and the tickets are all available in one website.

User Requirements

- The user shall be required to enter a correct email and password and should be approved as a valid user
- If the user is not registered, then the user must register to be able to book the tickets.
- The user shoud be able to view the movies without completing the booking process.

- User should be able to book movie tickets by selecting movie name, time and number of tickets.
- The user will choose the payment method to complete the booking
- The user should be able to logout and go back to the home page.

- The admin should be able to manage the movies.

- The admin should be able to view and generate payment and reservation reports.

System Requirements

- When the customer or admin enters email and password, the system should verify the email and password successfully or else an error alert will appear.
- When the customer registers a new account , email and password should be validated and added to the data base.
- When the customer chooses a movie ,time ,and number of seats a reservation should be done and add a tuple to the data base.
- The system shall send a notification that the reservation is confirmed with a reservation number.
- When the customer choses payment method, the booking should be completed .
- The system shall validate the payment method and if it is completed successfully , a tuple shall be added to the database.
- The system shall send a notification that the booking is confirmed.
- Reports shall be generated for the admin to view reservations and payments.
- Access to reports should be authorized only to admins.

System architecture:

Client-Server Architecture:

Client-server architecture is an important concept for system design. It's like the foundation of how the modern internet works. Nowadays, digital devices like computers, laptops, mobile devices are everywhere. Client-server architecture is the foundation of knowing how these computers talk to one another.

It's a distributed system model which shows how data and processing is distributed across a range of components. It can be implemented on a single computer. It's a set of stand-alone servers which provide specific services, a set of clients which call on these services, and a network which allows clients to access servers.

In a client-server architecture, the system is presented as a set of services, with each service delivered by a separate server. Clients are users of these services and access servers to make use of them.

Our system is purely a client server one, where the customer will be interacting through the web and sending requests to the server and expects some sort of response. It also allows multiple clients to access the website (internet), and all for them could sending requests and receiving responses.

Application Architecture

Transaction processing system

Example: reservation systems, E-commerce systems

Interactive transaction-based applications:

These are applications that execute on a remote computer and that are accessed by users from their own computers, phones, or tablets. Obviously, these include web applications such as e-commerce applications where you interact with a remote system to buy goods and services. This class of application also includes business systems, where a business provides access to its systems through a web browser or special-purpose client program and cloud-based services, such as mail and photo sharing. Interactive applications often incorporate a large data store that is accessed and updated in each transaction.

Transaction processing applications:

Transaction processing applications are database-centered applications that process user requests for information and update the information in a database. These are the most common types of interactive business systems. They are organized in such a way that user actions can't interfere with each other and the integrity of the database is maintained. This class of system includes interactive banking systems, e-commerce systems, information systems, and booking systems.

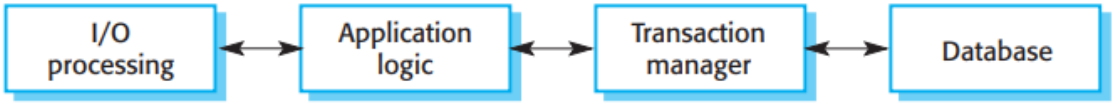
Transaction processing systems

Transaction processing systems are designed to process user requests for information from a database, or requests to update a database. Technically, a database transaction is part of a sequence of operations and is treated as a single unit (an atomic unit). All of the operations in a transaction have to be completed before the database changes are made permanent. This ensures that failure of operations within a transaction does not lead to inconsistencies in the database.

From a **user perspective**, a transaction is any coherent sequence of operations that satisfies a goal, such as "find the times of flights from London to Paris." If the user transaction does not require the database to be changed, then it may not be necessary to package this as a technical database transaction. An example of a database transaction is a customer request to withdraw money from a bank account using an ATM. This involves checking the customer account balance to see if sufficient funds are available, modifying the balance by the amount withdrawn and sending commands to the ATM to deliver the cash. Until all of these steps have been completed, the transaction is incomplete and the customer accounts database is not changed.

Transaction processing systems are usually interactive systems in which users make **asynchronous** requests for service which are then processed by a transaction manager.

Figure below illustrates the conceptual architectural structure of transaction processing applications. First, a user makes a request to the system through an I/O processing component. The request is processed by some application-specific logic. A transaction is created and passed to a transaction manager, which is usually embedded in the database management system. After the transaction manager has ensured that the transaction is properly completed, it signals to the application that processing has finished. Transaction processing systems may be organized as a "pipe and filter" architecture, with system components responsible for input, processing, and output.



Software Process

The development model in our online ticket system is Incremental development model as it will help break down the into several modules that can be incrementally constructed and delivered. The incremental build model is a method of software development where the product is designed, implemented and tested incrementally (a little more is added each time) until the product is finished. It involves both development and maintenance.

We chose to follow this model to be able to have functional phases of the project and after validating each increment we move further to the next increment. Furthermore, specification, development and validation are interleaved which helped us in the process development and develop the system in increments and evaluate each increment before proceeding to the development of the next increment.

We also used agile methods which is characterized for its are rapid delivery of functionality and responsiveness to changing customer requirements. In agile processes, planning is incremental and it is easier to change the process to reflect changing customer requirements.

Phases of incremental model:

- Requirement analysis: In the first phase of the incremental model, the product analysis expertise identifies the requirements. And the system functional requirements are understood by the requirement analysis team.

- Design & Development: In this phase of the Incremental model of SDLC, the design of the system functionality and the development method are finished with success. When software develops new practicality, the incremental model uses style and development phase.
- Testing: In the incremental model, the testing phase checks the performance of each existing function as well as additional functionality. In the testing phase, the various methods are used to test the behavior of each task.

Advantage of Incremental Model

- Errors are easy to be recognized.
- Easier to test and debug
- More flexible.
- Simple to manage risk because it handled during its iteration.
- The Client gets important functionality early.

FUNCTIONAL REQUIREMENTS

- The login email and password should be sent to the mentioned email address if a new account is created.
- The system should automatically calculate the fare for the movie and amount of money that will be paid for selected seats.
- Only registered users should be able to book tickets.
- Information about the credit card for the payment like card number and CVV number.
- Only admin email and password shall be able to generate the cinema's report.
- Confirmation messages will be sent at booking and reservation.
- Only movies with upcoming dates will be shown to the customer.

NONFUNCTIONAL REQUIREMENTS

- Availability: the system should always be available so the users could be able to view the information needed.

- Maintainability: the system should be able to be updated and modified
- Ease of use : the system should be easy to understand and use and be highly responsive.
- Security: only admins will be able to view the generated reports by using their email and password.
- Portability: the website should run on anyone's computer.
- Robustnes: the system downtime should not exceed five minutes.

Design Description

Our online ticket system was developed incremental development model so the design was divided into increments:

- Define system and user requirements
- Defining system users and their functionalities

We created increments based on the functionalities of each system user by prioritizing the requirements and started the specification, development and validation interleaving.

So for the admin user we started with the login/registration then added each functionality and tested each functionality before proceeding. Then we moved to the movies management then the report generation functionality.

For the customer, we started from viewing the available movies then moving forward to the reservation and then the payment process and testing each increment separately then moving forward to the next increment.

Note

We decided to show all available movie without filtering by the genre since this is more realistic for movie systems to show all movies as a marketing strategy.so if the genre is not available the customer will be able to see all options.