Software Development Laboratory

B.Tech. IV Semester

Name : Subhendu Maji

Roll Number : 18ETCS002121

Department : Computer Science and Engineering

Faculty of Engineering & Technology

M. S. Ramaiah University of Applied Sciences

Name: Subhendu Maji Registration Number: 18ETCS002121

M. S. Ramaiah University of Applied Sciences

Private University Established in Karnataka State by Act No. 15 of 2013

Faculty	Engineering & Technology
Programme	B. Tech. in Computer Science and Engineering
Year/Semester	2018/4th Semester
Name of the Laboratory	Software Development Laboratory
Laboratory Code	19CSL216A

List of Experiments

1. Requirements Analysis - I
2. Requirements Analysis - II
3. Use Case Diagram
4. Sequence Diagram
5. UML Modelling: Class Diagrams
6. UML Modelling: State Chart Diagrams
7. Activity Diagram and Data Flow Diagram
8. Entity-Relationship Diagram
9. Implementation of Software Design
10. Software Testing with Test Cases

Scenario for all labs:

Various scenarios will be given to students in the lab. Work in groups of 3 and develop the software solution. The Course leader is the customer. Contact the Course leader for any clarifications.

Name: Subhendu Maji Registration Number: 18ETCS002121

Index Sheet

No.	Lab Experiment	Viva (6)	Results (7)	Documentation (7)	Total Marks (20)
1	Requirements Analysis - I				
2	Requirements Analysis - II				
3	UML Modelling: Use Case Diagrams				
4	UML Modelling: Sequence Diagram				
5	UML Modelling: Class Diagram				
6	State Chart Diagram				
7	Entity-Relationship Diagram				
8	Activity Diagram and Data Flow Diagram				
9	Implementation and Testing				
10	Lab Internal Test conducted along the lines of SEE and valued for 50 Marks and reduced for 20 Marks				
	Total Marks				

Component 1 (Lab Internal Marks) =

Signature of the Staff In-charge

Name: Subhendu Maji Registration Number: 18ETCS002121

Laboratory 1

Title of the Laboratory Exercise: Requirements Analysis - I

1. Introduction and Purpose of Experiment

   Students get familiar with the documentation and scenario specified for all the lab exercises while analysing the requirements of the scenario

2. Aim and Objectives

   Aim

   - To develop formal software requirements in a standard format for a given engineering problem

   Objectives

   At the end of this lab, the student will be able to

   – Identify software requirements from problem statement
   – Identify type of a software requirement
   – Create an unambiguous list of software requirements based on interaction with a client

3. Experimental Procedure

- Work in teams of 3 students
- Each team should read the problem statement and identify requirements as a group
- Each team will then confirm the requirements and document the requirements in an SRS document
- Each individual will then write their lab manual, documenting their observations

4. Presentation of Results

Functional Requirements:

FR1: The system should allow users to browse movies available to watch and display the movie description.

FR2: The system should allow new users to register and existing users to login.

FR3: The system should allow users to select movies by applying filters

FR4: The system should allow user to view vacant and booked seats separately and select and book only the vacant seats.

FR5: The system should allow users to book multiple seats at a time.

FR6: The software should support multiple payment options.

FR7: The system should send booking details to the theater.

FR8: The system should send a confirmation email and e-ticket to the user.

FR9: The system should display customer care details / help.

FR10: The system should allow the user to logout.

Non-functional requirements:

NFR1: the software should have good maintainability.

The maintenance team should update the software periodically. The bugs should be fixed as soon as discovered. The team should update the software with the trending UI. New features should be added periodically.

NFR2: the system should have a secure payment gateway.

The system should be secure and clean. The gateway should be taken from a reputative bank server. The system should not be easy to hack.

NFR3: the system should have a good server support.

The system should not get crashed frequently. The system should have a backup server support when doing maintenance. The system should not be down for too long.

NFR4: The system should be user-friendly.

The system should have clear navigation.

NFR5: the system should be responsive in multiple platforms.

The system should be supportive in mobile, tablet and desktop. The software should be supportive in multiple operating system like Linux, Mac and Windows. The system should be working on Android as well as on iOS.

5. Analysis and Discussions

   In this lab we discussed about the requirements needed for the development of an app for booking movie tickets. There are two types of requirements: Functional and Non-functional requirements. Functional requirements describe what a system must do and what the user expects from the system like user login and registration, displaying movies, displaying the seats, booking of the tickets and payment options, while non-functional requirements describe aspects of the system that don't relate to its execution like the system's maintainability, compatibility, security and modifiability.

6. Conclusions

   We realized the requirements needed for the development of an app for booking movie tickets.

7. Comments

   1. Limitations of Experiments

The requirements are pre-defined and cannot be changed(frozen).

   2. Learning happened

Using any process model, we find skeleton of the customers' requirements there by making further improvements in future design.

Component	Max Marks	Marks Obtained
Viva	6	
Results	7	
Documentation	7	
Total	20	

Name: Subhendu Maji Registration Number: 18ETCS002121

Laboratory 2

Title of the Laboratory Exercise: Requirements Analysis - II

1. Introduction and Purpose of Experiment

Students will formally document the identified requirements in an SRS document for the scenario

2. Aim and Objectives

Aim

- To develop formal SRS document in a standard format for a given engineering problem

Objectives

At the end of this lab, the student will be able to

- Identify dependencies of a software requirement
- Create SRS document in a standard format

3. Experimental Procedure

- Work in teams of 3 students
- Each team should read the problem statement and identify requirements as a group
- Each team will then confirm the requirements and document the requirements in an SRS document
- Each individual will then write their lab manual, documenting their observations

4. Presentation of Results

Item	**Detail**
Requirement tag	FR1
Requirement statement	The System should allow users to browse movies available to watch and display movie description.
System requirement Addressed	
Dependent on requirements	--
Response time constraint	The desired response time for users interface is 10ms based on survey.
Stake holder owning the requirement	End User, Admin
Example of user/system interaction for this requirement	A User should to be able to view different movies available to watch and the theatres it is being screened at, should be able to view the updates of new movies that are to be released soon and check out the movie description and ratings. End user 'A' wants to view the movies available at 6:00pm at the nearest theatres. End user 'B' wants to view the list of new movies that are going to be released the next week. End user 'C' wants to view the description and ratings of the movie. The admin should be able to update the movies and show timings

Item	**Detail**
Requirement tag	FR2
Requirement statement	The system should allow new users to register and existing user to login.
System requirement Addressed	
Dependent on requirements	----
Response time constraint	The desired response time for users interface is 10ms based on survey.
Stake holder owning the requirement	End User, Admin

Example of user/system interaction for this requirement	The new user should be able to register by entering details such as name, user id, password, phone number and email id. A existing user should be able login by entering user id and password. End user 'A' is a new user and can register by entering his name, phone number, email id, new user id and a new password. End user 'B' is an existing user and login by entering her user id and password. Admin 'X' should validate the credentials entered by the user

Item	**Detail**
Requirement tag	FR3
Requirement Statement	The system should allow users to select movies by applying filters
System Requirement Addressed	
Dependent on Requirements	FR1
Response Time Constraint	The desired response time for users interface is 10ms based on survey.
Stake Holder Owning the Requirement	End user
Example of user/system interaction for this requirement	A user should be able to add filters as per his preference regarding the location of the theatre, language of the movie, video display, timing of the movie and type of seats and select the movie. End user 'A' wants to watch a movie in English 3d display. End user 'B' wants to book a recliner seat in the nearest theatre at 9:30 pm.

Item	**Detail**
Requirement tag	FR4
Requirement Statement	The system should allow user to view vacant and booked seats separately and select and book only the vacant seats.
System Requirement Addressed	
Dependent on Requirements	FR1,FR2,FR3

Response Time Constraint	The desired response time for users interface is 10ms based on survey.
Stake Holder Owning the Requirement	End user, Admin
Example of user/system interaction for this requirement	A user must be able to view the seats which are available to book and those which are already booked by other users separately and be able to only select seats which are available and is directed to the payment page after confirmation. End user 'B' can book the seat number 13D if it is available. End user 'A' cannot book the seat number 13D if it is already booked by another user. Admin 'X' must update the booked seats.

Item	**Detail**
Requirement tag	FR5
Requirement Statement	The system should allow users to book multiple tickets at a time.
System Requirement Addressed	
Dependent on Requirements	FR1, FR2,F3,F4
Response Time Constraint	The desired response time for users interface is 10ms based on survey.
Stake Holder Owning the Requirement	End User, Administrator
Example of user/system interaction for this requirement	A user should be able to book as many tickets as required as per availability. End user 'A' wants to book 10 tickets. Admin 'X' should update the booked seats.

Item	**Detail**
Requirement Tag	FR6
Requirement Statement	The software should support multiple payment options. .
System Requirement Addressed	

Dependent on Requirements	FR2,FR4,FR5
Response Time Constraint	The desired response time for users interface is 10ms based on survey.
Stake Holder Owning the Requirement	End User, Bank
Example of user/system interaction for this requirement	The software should allow user to pay through multiple payment options like credit card, debit card, UPI, and net banking after confirming the ticket. After clicking on one of the payments options, the users should be redirected to its respective payment portal. End user 'A' wants to pay for the ticket using his debit card, he can do so by entering his card number and CVV. End user 'B' wants to pay for the ticket using net banking. After selecting the option he is redirected to the respective bank's server where the payment is done.

Item	**Detail**
Requirement tag	FR7
Requirement Statement	The system should send the booking details to the theatre.
System Requirement Addressed	
Dependent on Requirements	FR2, FR4,F6
Response Time Constraint	The system should allow users to select movies by applying filters
Stake Holder Owning the Requirement	Administrator
Example of user/system interaction for this requirement	After the confirmation of the ticket from the user the software must send the ticket details such as the user's name, seat number, ticket number to the administrator. Admin 'X' gets the details of end user 'A' such as his name, seat number, and ticket number after confirmation.

Item	**Detail**
Requirement Tag	FR8
Requirement Statement	The system should send a confirmation email and bill summary to the user.
System Requirement Addressed	
Dependent on Requirements	FR2,F4, FR6
Response Time Constraint	--
Stake Holder Owning the Requirement	End User, administrator
Example of user/system interaction for this requirement	The user should get a confirmation mail and an SMS which should contain his seat number and ticket summary. End user 'A' gets a confirmation mail containing his seat number and ticket number on his registered email id and an SMS to his registered phone number. Admin 'X' should send a confirmation mail and SMS to the user.

Item	**Detail**
Requirement tag	FR9
Requirement statement	The system should display customer care details/help.
System requirement Addressed	
Dependent on requirements	-----
Response time constraint	
Stake holder owning the requirement	End User, admin
Example of user/system interaction for this requirement	The user should be able to contact the customer care/ view help document i.e., stored in the corresponding website. If the end user 'A' is facing any problems regarding payments, he/she can contact the customer care for help. If end user 'B' has any issues navigating through the software, he can look into the help document.

Item	**Detail**
Requirement tag	FR10
Requirement statement	The system must allow the logged in user to logout
System requirement Addressed	The system should allow users to select movies by applying filters
Dependent on requirements	FR2
Response time constraint	The desired response time for users interface is 10ms based on survey.
Stake holder owning the requirement	End User
Example of user/system interaction for this requirement	The user should be able to log out, once he has logged in End user 'A' wants to log out of the website.

8. Analysis and Discussions

In this lab we discussed about the requirements needed for the development of an app for booking movie tickets. There are two types of requirements: Functional and Non-functional requirements. Functional requirements describe what a system must do and what the user expects from the system like user login and registration, displaying movies, displaying the seats, booking of the tickets and payment options, while non-functional requirements describe aspects of the system that don't relate to its execution like the system's maintainability, compatibility, security and modifiability.

9. Conclusions

We developed a formal SRS document in a standard format using the requirements elicited from the user.

Component	Max Marks	Marks Obtained
Viva	6	
Results	7	
Documentation	7	
Total	20	

Name: Subhendu Maji Registration Number: 18ETCS002121

Laboratory 3

Title of the Laboratory Exercise: Data flow modelling with CASE tools – High Level Design

1. Introduction and Purpose of Experiment

   Students will apply data flow modelling to develop the high level design for given scenario

2. Aim and Objectives

   Aim

   - To develop software architecture for a given requirements specification using Structured analysis and Design Technique

   Objectives

   At the end of this lab, the student will be able to

   – Identify context of the software
   – Identify Inputs, Outputs and Data Stores for a given software
   – Identify modules in a software and their dependencies
   – Create design document for a given SRS

3. Experimental Procedure

- Work in teams of 7 students
- Each team should read the problem statement and identify requirements as a group
- Each team will then confirm the requirements and document the requirements in an high level design document
- Each individual will then write their lab manual, documenting their observations

4. Presentation of Results

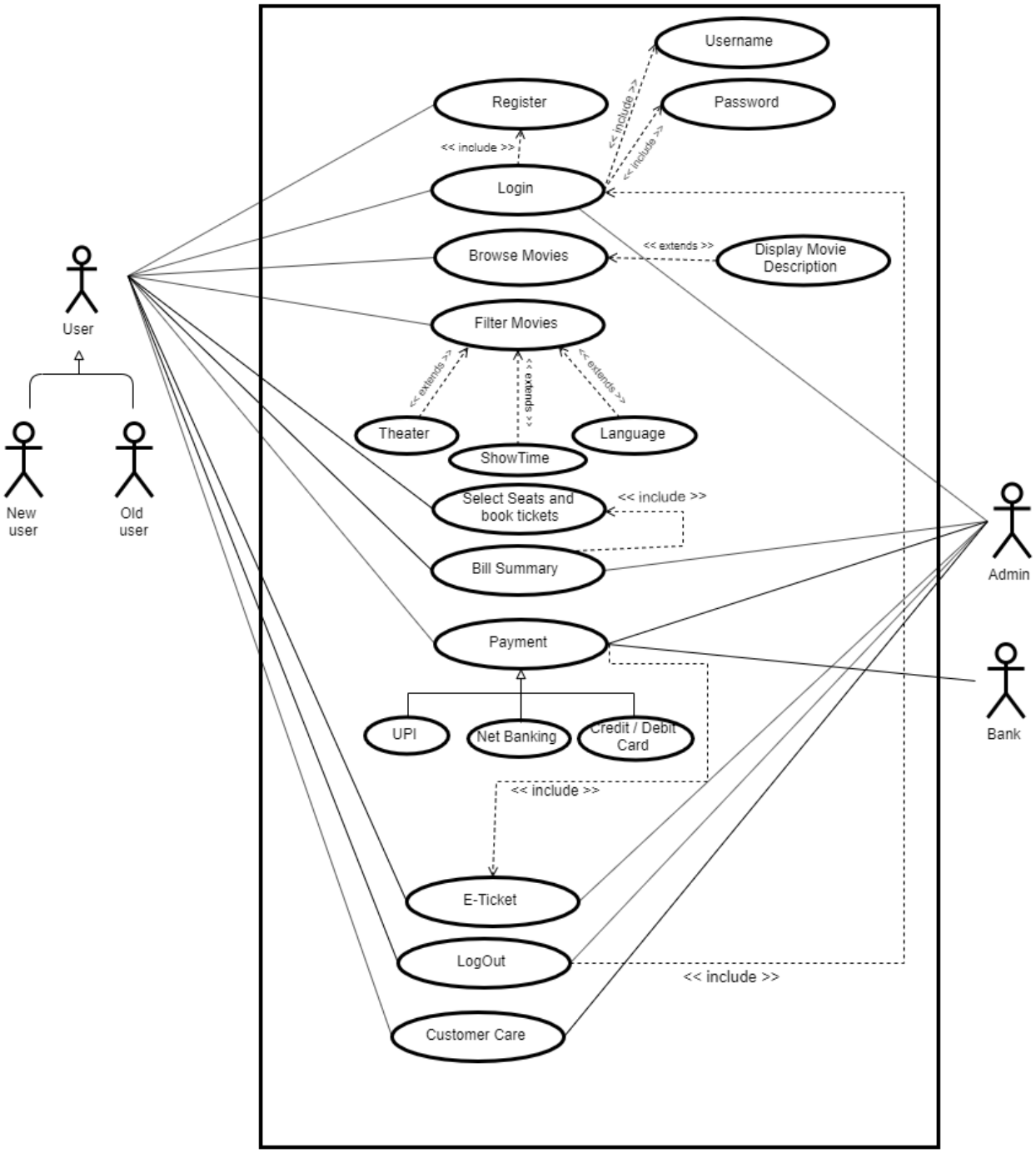

Figure 3.1 use case diagram

5. Analysis and Discussions

   In this lab we learnt about the use case diagram and the its various components.

   We discussed about the relationship between the various functional requirements (generalization and dependency) and have analysed the dependency of requirements (include and extends). We discussed about the actors involved and their relationship with the use case (association) and among themselves(generalization). We used the DIA software to generate the use case diagram.

6. Conclusions

   We were able to draw UML diagrams and successfully designed for the applications discussed in the previous lab.

Component	Max Marks	Marks Obtained
Viva	7	
Results	6	
Documentation	6	
Total	20	

Name: Subhendu Maji Registration Number: 18ETCS002121

Laboratory 4

Title of the Laboratory Exercise: Data flow modelling with CASE tools – Low Level Design

1. Introduction and Purpose of Experiment

   Students will apply data flow modelling to develop the low level design for given scenario

2. Aim and Objectives

   Aim

   - To develop low level software design for a given requirements specification using Structured analysis and Design Technique

   Objectives

   At the end of this lab, the student will be able to

   – Identify functions in modules
   – Identify Inputs, Outputs and Data dependencies for functions
   – Create low level design document for a given SRS

3. Experimental Procedure

- Work in teams of 7 students
- Each team should read the problem statement and identify requirements as a group
- Each team will then confirm the requirements and document the requirements in an low level design document
- Each individual will then write their lab manual, documenting their observations

4. Calculations/Computations/Algorithms

   Sequence diagrams is a behavioural diagram shows interaction between objects. It shows the involved objects and classes and the messages exchanged between two object to carry out some functionality of the particular given scenario. The components are Objects, lifeline, activation box, messages, response.

5. Presentation of Results

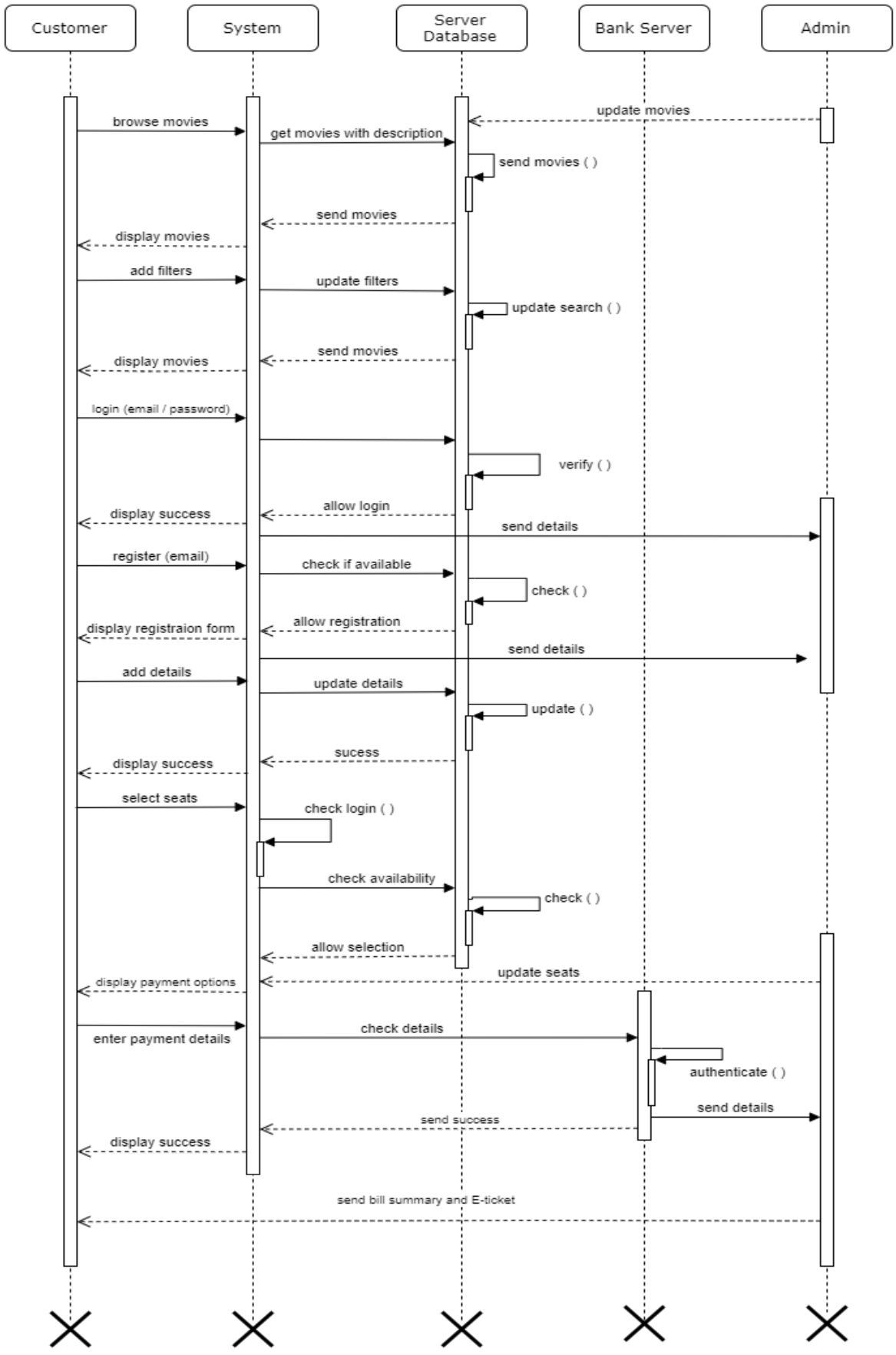

Figure 4.1 Sequence diagram

6. Analysis and Discussions

We have discussed about the sequence diagram and its components. We have analysed how the messages are passed between two objects in a sequential order and have used the sequence diagram to represent them. We have discussed about the messages (call and recursion) and responses by the objects in the given scenario of online booking application.

7. Conclusions

We were able to draw sequence diagram for the functional requirements (of the online booking application) discussed in previous lab sessions

Component	Max Marks	Marks Obtained
Viva	6	
Results	7	
Documentation	7	
Total	20	

Name: Subhendu Maji Registration Number: 18ETCS002121

Laboratory 5

Title of the Laboratory Exercise Class Diagrams

1. Introduction and Purpose of Experiment

   Students will apply object oriented analysis and design for the given scenario for analysis of requirements and possible interactions

2. Aim and Objectives

   Aim

   - To construct a UML class diagram for a given system and identify the class members and determine their relationships

   Objectives

   At the end of this lab, the student will be able to

   – Explain the purpose of the sequence diagram
   – Identify the logical sequence of activities undergoing in a system, and represent them pictorially
   – Design and model a given use case using UML-sequence diagrams

3. Experimental Procedure

- Work in teams of 7 students
- Each team should read the problem statement and discuss the requirements as a group
- Each team will then create and confirm the design and document the design in an software architecture specifications document
- Each individual will then write their lab manual, documenting their observations

4. Presentation of Results

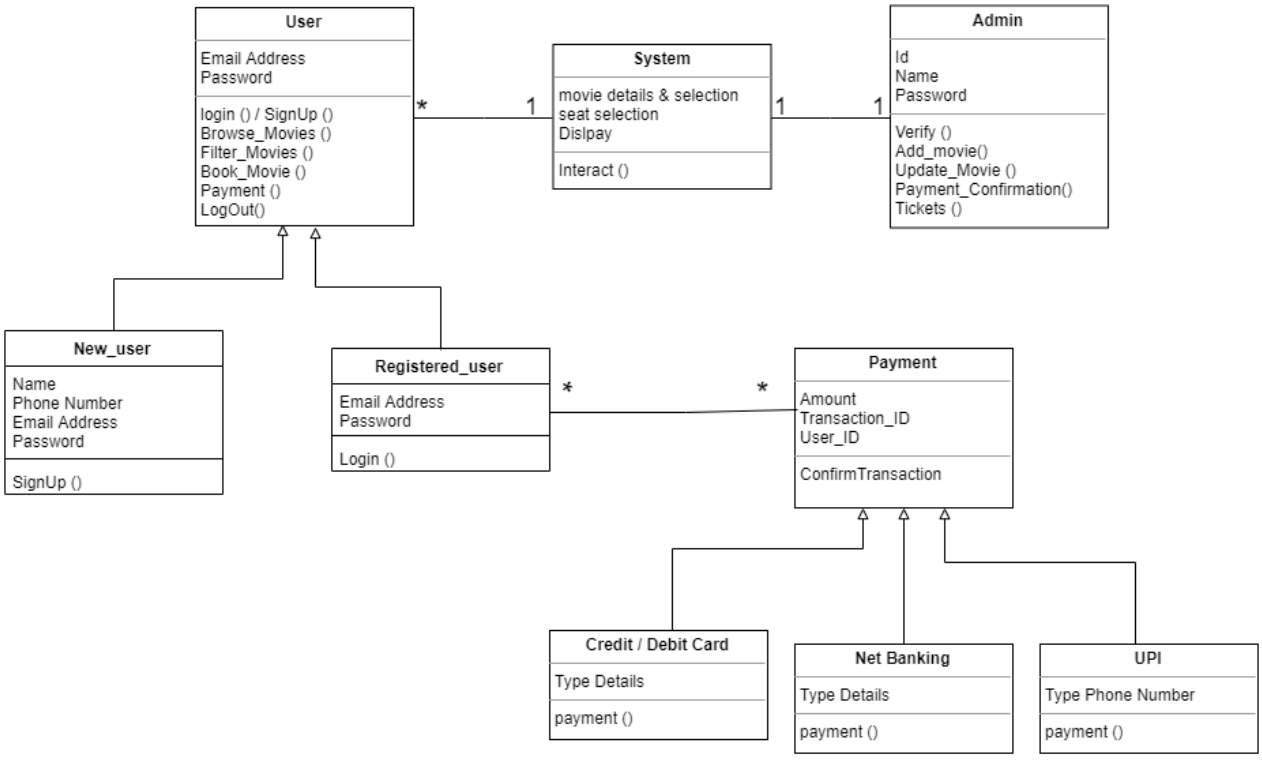

Figure 5.1 Class diagram

5. Analysis and Discussions

Class diagrams are structural models which are most widely used as it helps understand the requirements of the problem domain and to identify its components.

In this lab we developed a class diagram containing various class members which has attributes and functions. We used class members like user, existing user, new user, payment, Paytm, credit, debit, and logout

6. Conclusions

- We constructed a class diagram for an online movie ticket booking application and identified the class members and determined their relationships.

7. Comments

1. Limitations of Experiments

- In class diagram, as only class members are the elements, we cannot include all the functional requirements. We can include only limited functional requirements which are function of only particular class members.

Component	Max Marks	Marks Obtained
Viva	6	
Results	7	
Documentation	7	
Total	20	

Name: Subhendu Maji Registration Number: 18ETCS002121

Laboratory 6

Title of the Laboratory Exercise: State chart diagrams

1. Introduction and Purpose of Experiment

   Students will apply object-oriented analysis and design for the given scenario for low level design of classes

2. Aim and Objectives

   Aim

   - To develop low level software design for a given class diagram using state chart diagrams

   Objectives

   At the end of this lab, the student will be able to

   – Identify states of each object
   – Identify triggers and messages for each object
   – Understand the behavior of a class, given its state chart diagram

3. Experimental Procedure

- Work in teams of 7 students
- Each team should read the class diagram and identify objects, interactions and states of objects
- Each team will then design state transitions and simulate the same. They will then document the design in a low-level design specification document
- Each individual will then write their lab manual, documenting their observations

4. Presentation of Results

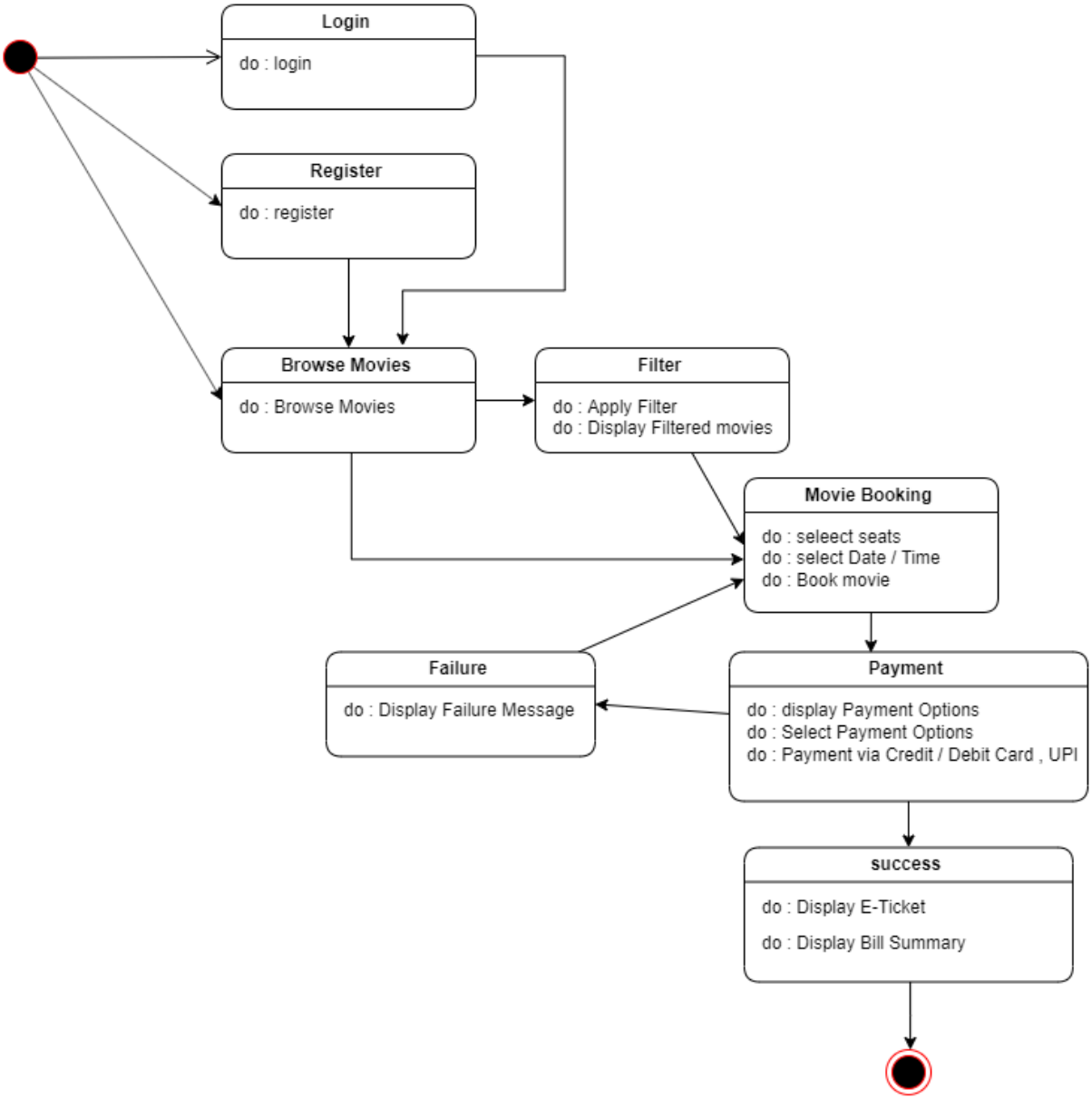

Figure 6.1 state chart diagram

5. Analysis and Discussions

In this lab we worked on the state chart diagram, which is a behavioural model.

The state chart diagram is a model which is used to describe the various states of the different objects in its life cycle. Here, the emphasis is laid on the state changes upon some internal and external events. In this lab we created the various states for the online movie booking website, like login, register, browse movies, filter, movie booking, payment and success and identified the triggers for the same which leads to the next state.

6. Conclusions

We implemented the state chart diagram.

State chart gives the control decisions and state each mode of operation foe the object which behaves differently depending on its state.

Component	Max Marks	Marks Obtained
Viva	6	
Results	7	
Documentation	7	
Total	20	

Name: Subhendu Maji Registration Number: 18ETCS002121

Laboratory 7

Title of the Laboratory Exercise: Activity Diagrams

1. Introduction and Purpose of Experiment

   Students will apply object oriented analysis and design for the given scenario for object decomposition

2. Aim and Objectives

   Aim

   - To construct a UML class diagram for a given system and identify the class members and determine their relationships

   Objectives

   At the end of this lab, the student will be able to

   - Identify the main members of the family
   - Identify how they are related to each other
   - Find the characteristics of each family member
   - Determine relations among family members
   - Decide the inheritance of personal traits and characters

3. Experimental Procedure

- Work in teams of 7 students
- Each team should read the problem statement and discuss the requirements as a group
- Each team will then create and confirm the design and document the design in an software design specifications document
- Each individual will then write their lab manual, documenting their observations

4. Presentation of Results

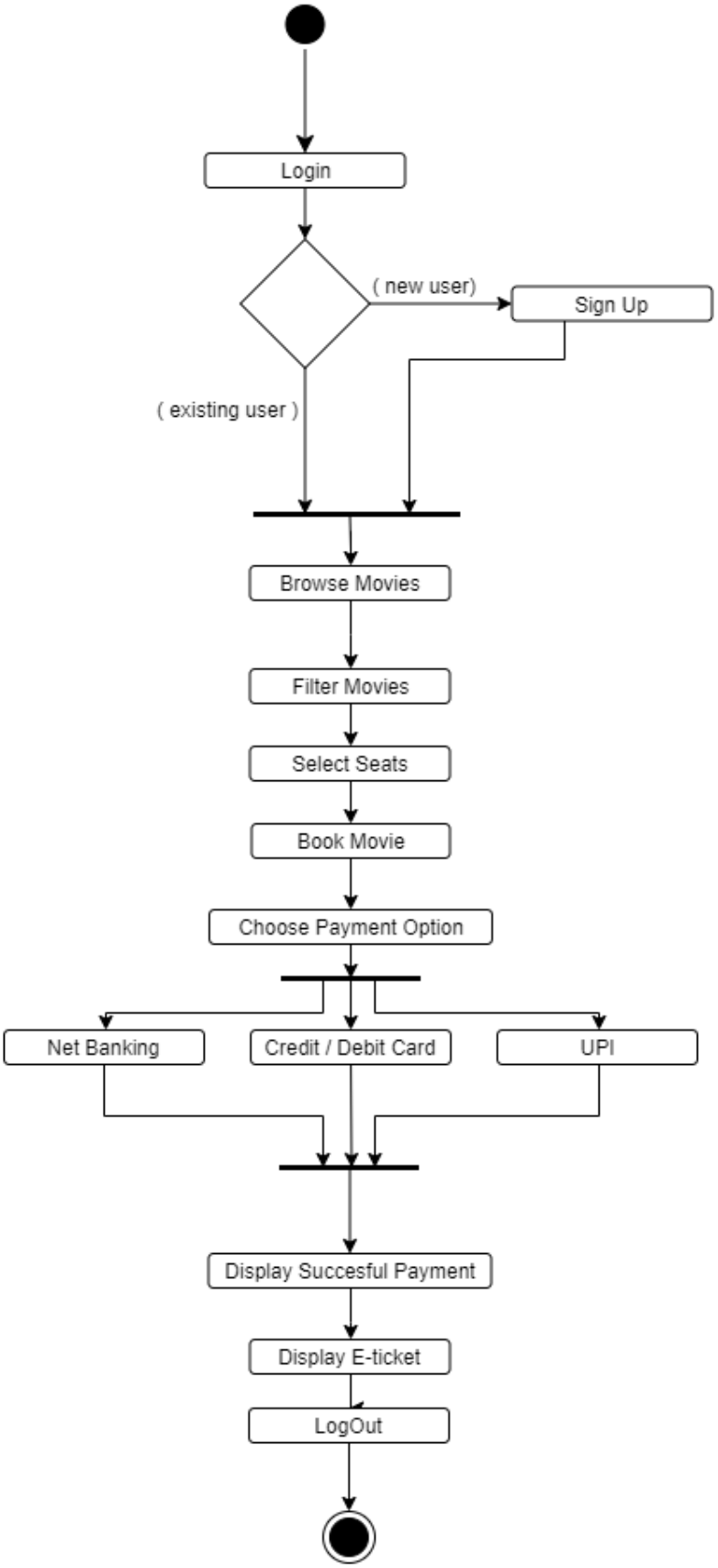

Figure 7.1 Activity diagram

5. Analysis and Discussions

In this lab we worked on the activity diagram, which is a behavioural models, for an online movie booking website. The activity diagram lays emphasis on the flow of the activity, we specified the activities like login, filter movies, select seats, book tickets and logout, we also determined the flow of the activities.

6. Conclusions

In this lab we created the state activity diagram using the DIA software for a movie booking system.

7. Comments

1. Limitations of Experiments

Activity Diagrams can be branched out into state diagrams with some additional notations and transitions.

2. Limitations of Results

Activity diagrams may stand alone to visualize, specify and document the dynamics of a society of objects or they may be used to model the flow of control of an operation.

3. Learning happened

Through Visualization of the requirements of the customer.

Component	Max Marks	Marks Obtained
Viva	6	
Results	7	
Documentation	7	
Total	20	

Name: Subhendu Maji Registration Number: 18ETCS002121

Laboratory 8

Title of the Laboratory Exercise: Entity-Relation Diagram

1. Introduction and Purpose of Experiment

   Students will apply object oriented analysis and design for the given scenario for low level design of classes

2. Aim and Objectives

   Aim

   - To develop low level software design for a given class diagram using entity-relation diagram.

Objectives

At the end of this lab, the student will be able to

– Apply industry standard coding standards

– Use automatic documentation tools

– Create maintainable code

3. Experimental Procedure

- Work in teams of 7 students
- Each team should read the class diagram and identify objects, interactions and states of objects
- Each team will then split workload and develop classes individually.
- Each individual will then write their lab manual, documenting their observations

4. Calculations/Computations/Algorithms

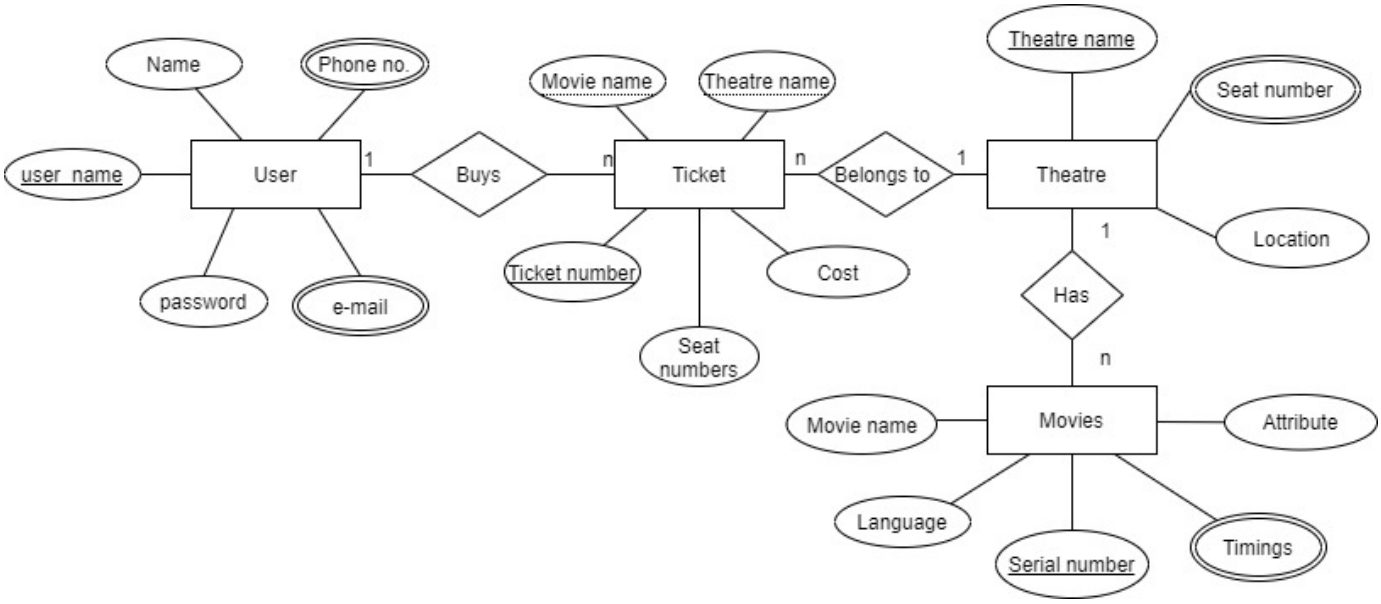

Figure 8.1 Entity relation diagram

6. Analysis and Discussions

The Entity Relationship diagram are a visual tool which is helpful to represent the ER model. They display the relationship of entity set stored in a database. It is based on three concepts i.e. Entities, Attributes and relationships. It is a visual representation of data that describe how data is related to each other.

7. Comments

1. Limitations of Experiments

Each entity only appears a single time in the ER diagram, the relationships shouldn't be connected to each other.

2. Limitations of Results

Entities and relationship should be properly labelled, eliminating any redundant entities or relationships

3. Learning happened

Through translatable relational tables which allows to build database regarding the entities, attributes and relationships.

Component	Max Marks	Marks Obtained
Viva	6	
Results	7	
Documentation	7	
Total	20	

Name: Subhendu Maji Registration Number: 18ETCS002121

Laboratory 9

Title of the Laboratory Exercise: Data Flow Diagram

1. Introduction and Purpose of Experiment

   Students will apply object oriented analysis and design for the given scenario for low level design of classes

2. Aim and Objectives

   Aim

   - To develop low level software design for a given class diagram using data flow diagram.

Objectives

At the end of this lab, the student will be able to

- Apply industry standard coding standards
- Use automatic documentation tools
- Create maintainable code

3. Experimental Procedure

- Work in teams of 7 students
- Each team should read the class diagram and identify objects, interactions and states of objects
- Each team will then split workload and develop classes individually.
- Each individual will then write their lab manual, documenting their observations

4. Calculations/Computations/Algorithms

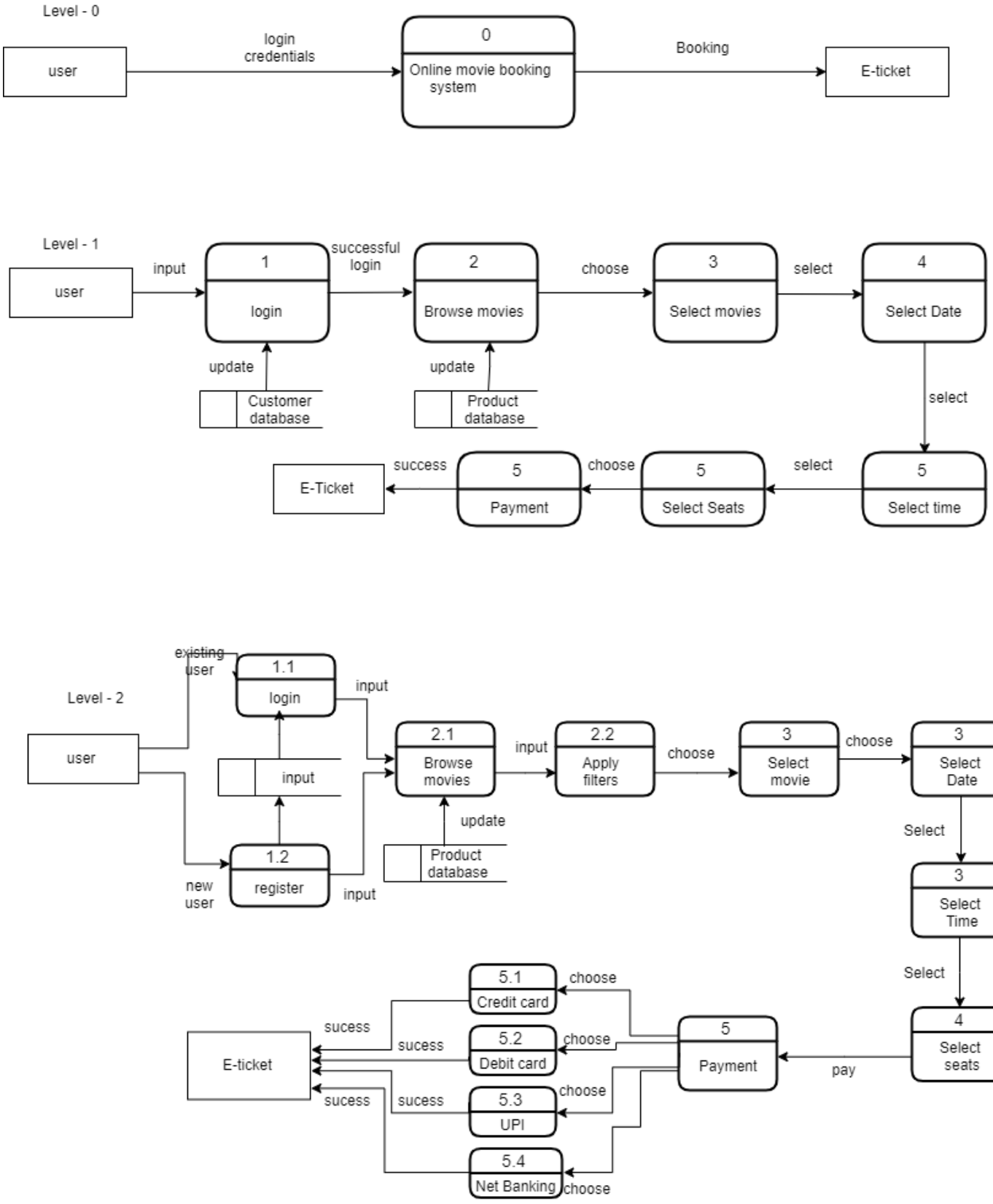

Figure 9.1 Data flow diagram

5. Analysis and Discussions

A data flow diagram is a way of representing a flow of data of a process or a system. The DFD also provides information about the outputs and inputs of each entity and the process itself.

Comments

- Limitations of Experiments

  The disadvantage of DFD is that if there are any changes in the requirements in the later stages there will be pattern change complexity.

  And it is also sophisticated.

- Learning happened

  Data Flow Diagram (DFD) is a hierarchical graphical model of a system. It is also known as bubble chart. The DFD takes an input-process-output view of a system.

  Level 0/ Context diagram depicts the environment, I/O conditions of the software.

- Recommendations

  I recommend OBJECT ORIENTED DIAGRAMS over data flow diagrams.

Component	Max Marks	Marks Obtained
Viva	6	
Results	7	
Documentation	7	
Total	20	

Name: Subhendu Maji Registration Number: 18ETCS002121

Laboratory 10

Title of the Laboratory Exercise: Implementation and Unit testing

Implementation:

The movie booking web-app I built is named “The Cinemasy”. The source code of the frontend of the web-app can be found at https://github.com/subhendu17620/Cinemasy .
The backend of Cinemasy can be found at https://github.com/subhendu17620/Cinemasy-backend .

This web-app is built with React which is JavaScript library for building UI’s. This web-app is a MERN - stack website.
The term MERN stack means it is built using 4 technologies MongoDB, Express, React and NodeJS.

Technologies Used:

- For frontend
  - ReactJS
  - JavaScript
  - CSS /SASS

- For backend
  - MongoDB
    - Mongoose - for simplification.
  - Express
    - As a middleware to connect backend with frontend.
  - Passport
    - As an authentication middleware for NodeJS.

The system is using online database of MongoDB I.e. Atlas.

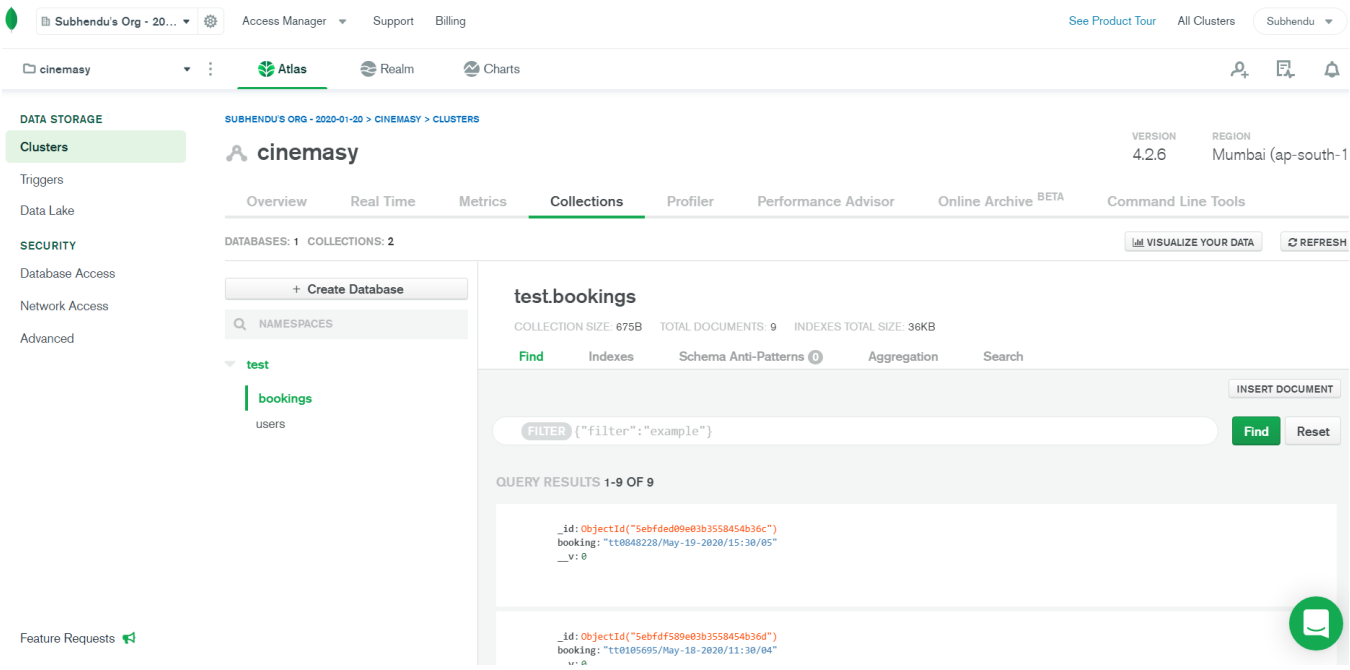

Figure 10.1 represents the online database of Cinemasy

All the movie bookings are stored in database after successful payment in format ***"IMDB id/ date/ time/ seat number".***


```
_id: ObjectId("5ebfded09e03b3558454b36c")
booking: "tt0848228/May-19-2020/15:30/05"
__v: 0

_id: ObjectId("5ebfdf589e03b3558454b36d")
booking: "tt0105695/May-18-2020/11:30/04"
__v: 0

_id: ObjectId("5ebfdf749e03b3558454b36e")
booking: "tt1375666/May-21-2020/18:00/06"
__v: 0

_id: ObjectId("5ebfe2d49e03b3558454b36f")
booking: "tt0800369/May-19-2020/11:30/02"
__v: 0
```


Figure 10.2 Successful movie bookings

All user details are also stored in database after Sign-up. The password is encoded as salt and hash by Passport-JWT.


```
_id: ObjectId("5ebfde8e9e03b3558454b36a")
fullName: "test name"
username: "test@gmail.com"
contactNumber: "9876543210"
salt: "8477bc00ffc7a4fcebf60180727d75658ea3957d2ed925a0d24cd5574f5ba73f"
hash: "08988a01ad0c0d23c21bed5769e02003bb1afcb1ce2d4da0283dae026bf632f696502a..."
__v: 0
```

```
_id: ObjectId("5ee239d65ba708457c614c67")
fullName: "Subhendu Maji"
username: "subhendu@gmail.com"
contactNumber: "1234567890"
salt: "7134e5e03b1fdd12a8f78a6f611458d1006bc19e7ff577dc6ae2420d17d41dee"
hash: "222752d921b5fd143d2acac4e405d4f41bc96e0db9f4be1961806219c98de4a6c74acd..."
__v: 0
```


Figure 10.3 User details after Sign-up

Features:

- Interactive Home page (Landing Page)

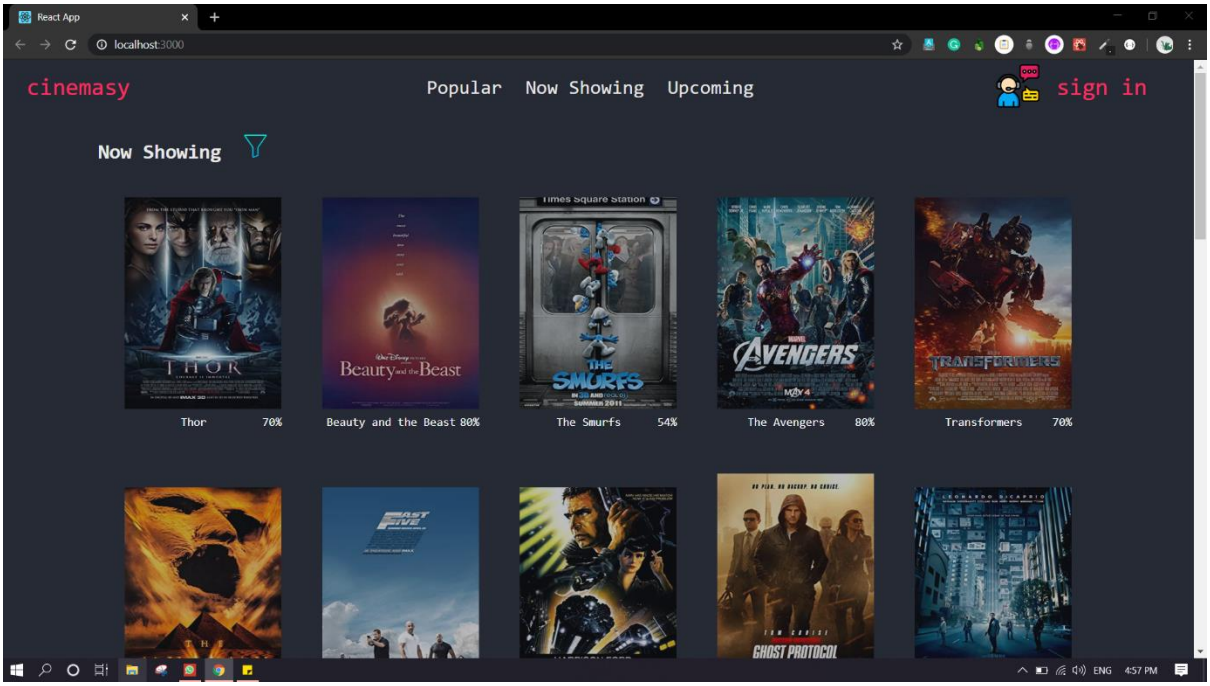

Figure 10.4 Home page of Cinemasy

- Three categories to Filter movies through i.e. Now Showing, Popular and Upcoming.

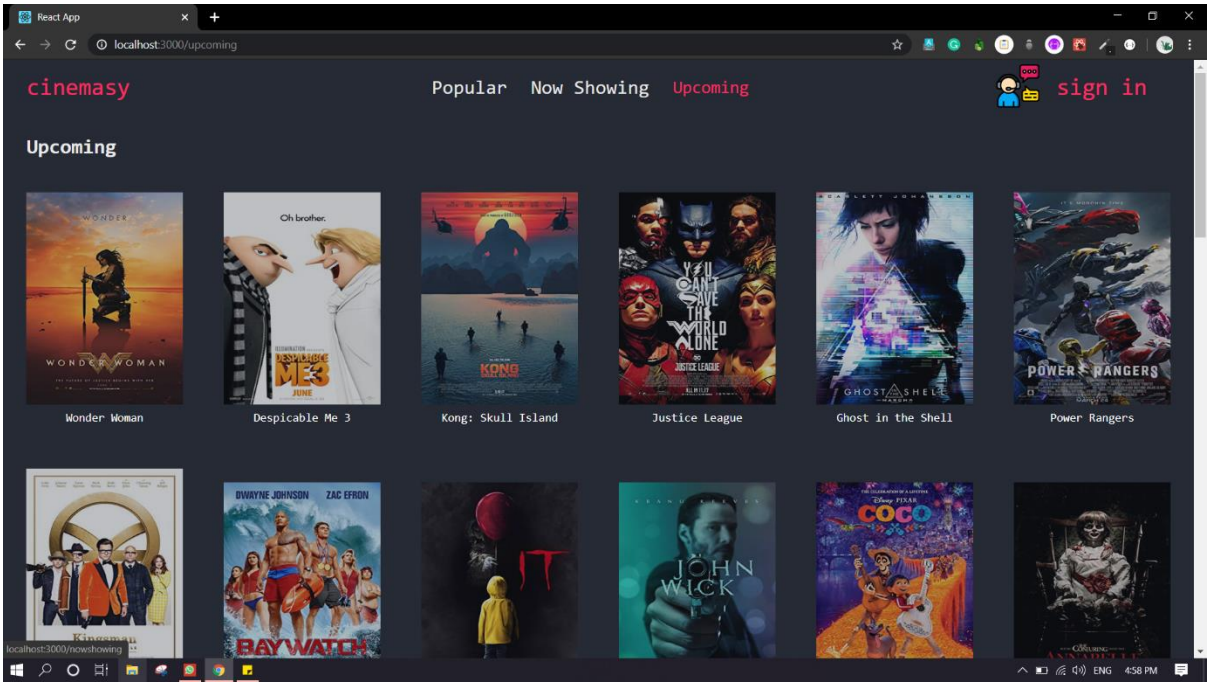

Figure 10.5 represents one of the three filter options - Upcoming Page

- Sign-In /Sign-Up page for new users to Sign-Up and existing user to Sign-In

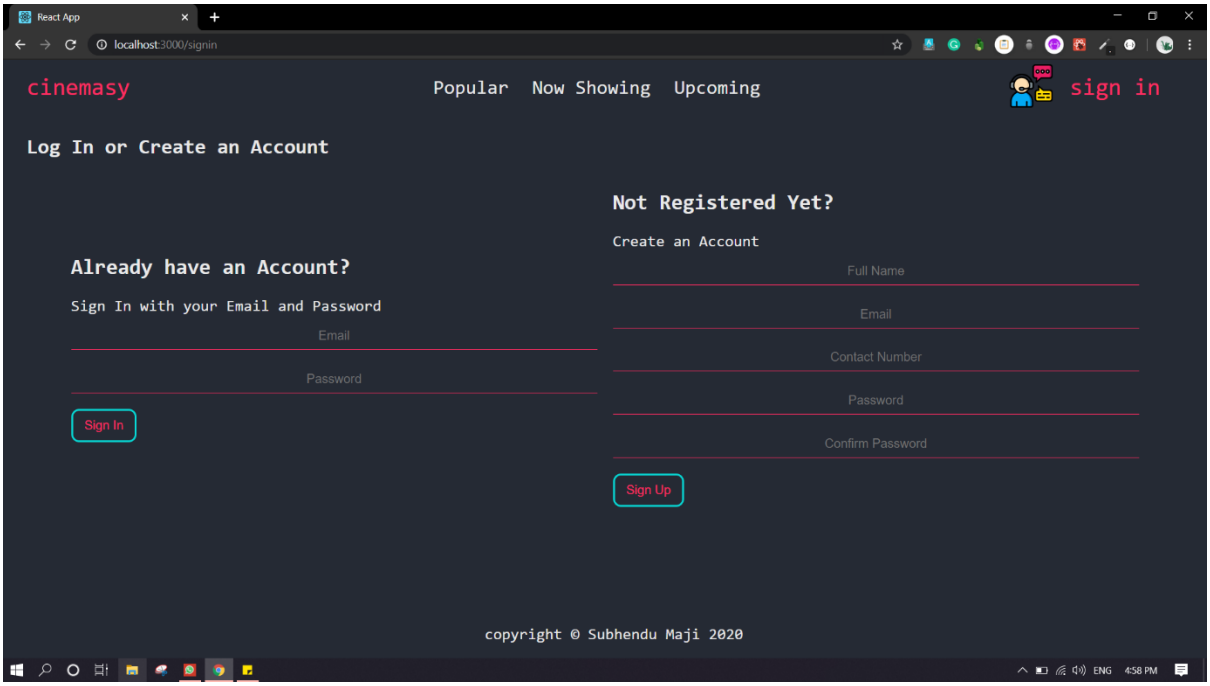

Figure 10.6 Sign-In / Sign-Up Page

- *Shows Movie Description of the movie by Clicking on its movie poster.*

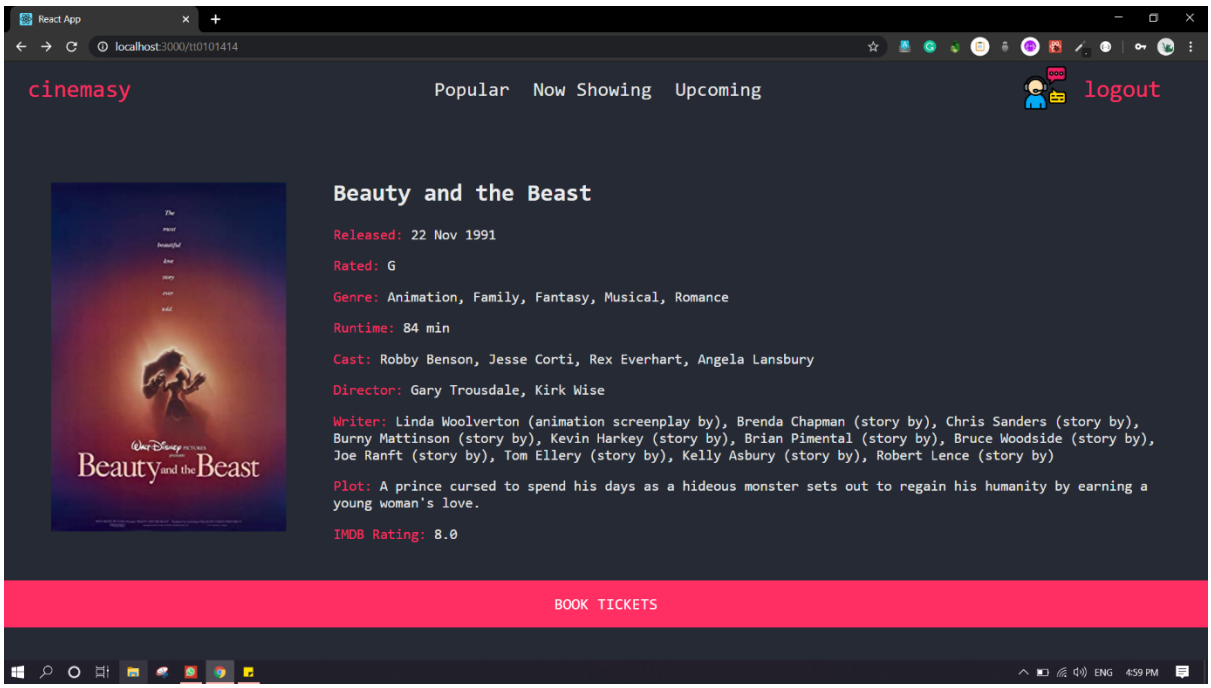

Figure 10.7 represents displaying of movie Description

- *After successful payment, an E-ticket (QR code) is generated for the user with all the booking details.*

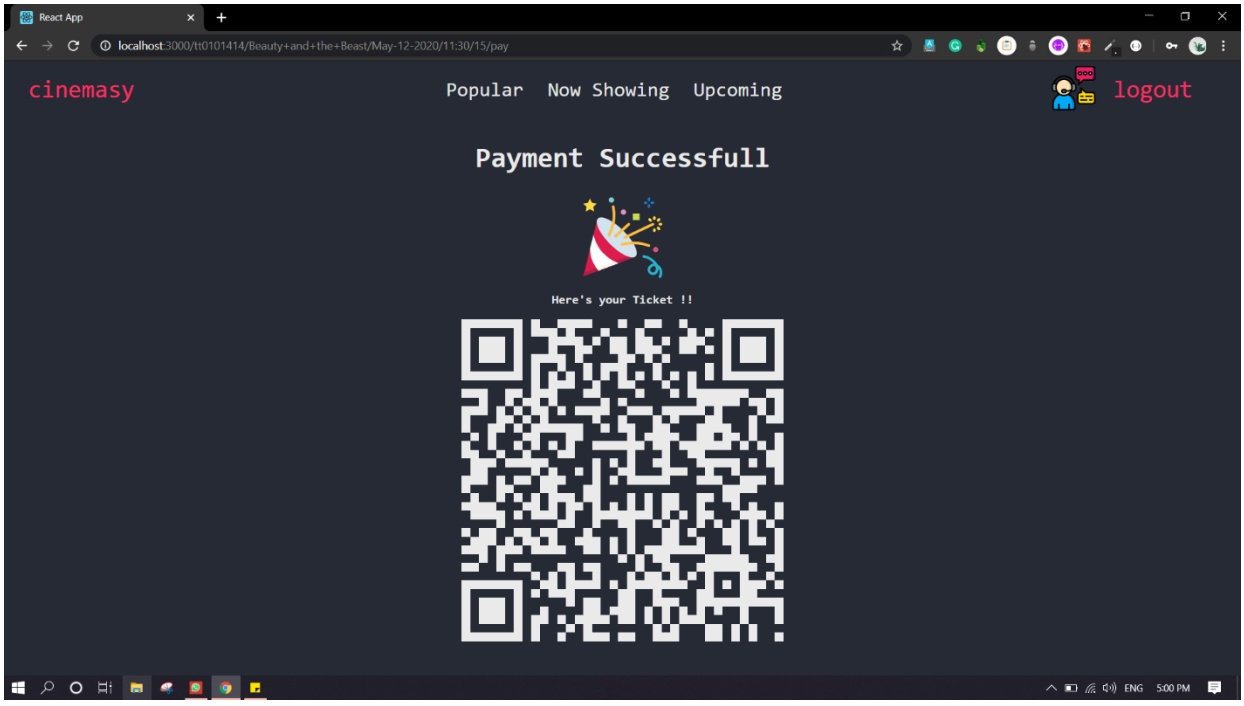

Figure 10.8 E-ticket (QR code) with all the booking details