



M. S. Ramaiah University of Applied Sciences

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

Software Development Laboratory

B.Tech. IV Semester

Name : Tanishq R Porwar

Roll Number : 18ETCS002131

Department : Computer Science and Engineering

Faculty of Engineering & Technology

M. S. Ramaiah University of Applied Sciences



Faculty	Engineering & Technology
Programme	B. Tech. in Computer Science and Engineering
Year/Semester	2018/4 th 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
8. Entity-Relationship Diagram
9. Data Flow Diagram
10. Implementation and Testing

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.

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	Activity Diagram				
8	Entity-Relationship Diagram				
9	Data Flow Diagram				
10	Implementation and Testing				
	Total Marks				

Component 1 (Lab Internal Marks) =**Signature of the Staff In-charge**

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 analyzing 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. Calculations/Computations/Algorithms

Functional requirements:

- FR1:** The system should allow registration for new user.
- FR2:** The system should allow login for registered user.
- FR3:** The system should allow the user to browse the movies.
- FR4:** The system should filter the movies on basis of language/show time and location.
- FR5:** The system should show the upcoming movies.
- FR6:** The system should show details of a particular movie upon selection.
- FR7:** The system should show option to select seats.
- FR8:** The system should show payment options.
- FR9:** The system should show the order summary and provide the e –ticket.
- FR10:** The system should allow the user to logout.

NON FUNTIONAL REQUIREMENT:

- NFR1: Maintainability:** Flutter should be used since it is object oriented and the app is built with widgets so that the maintainability is simple.
- NFR2: Security:** The system should have a firebase authentication in all transactions that include any confidential customer information.
- NFR3: Reliability:** The system should have a good server support.
- NFR4: User Friendly:** The system should be user-friendly with a one tap login using Google or Facebook.
- NFR5: Portability:** The system should work in multiple platforms and so flutter should be used.

5. Presentation of Results

6. Analysis and Discussions

The basic knowledge of process models and its list of requirements to enlist the customer data as a systematic point in a high-level representation.

7. Comments

a. Limitations of Experiments

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

b. Learning happened

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

c. Recommendations

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

Laboratory 2

Title of the Laboratory Exercise: Requirements Analysis - II

7. Introduction and Purpose of Experiment

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

8. 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

9. 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

10. Calculations/Computations/Algorithms

Item	Detail
Requirement Tag	FR1
Requirement Statement	The system should allow registration for new user.
Dependent on Requirements	-
Stake Holder Owning the Requirement	End User
Example of user/system interaction for this requirement	<p>A user should be able to register from anywhere.</p> <p>End user 'A' wants to register as a user to the app.</p>

Item	Detail
Requirement Tag	FR2
Requirement Statement	The system should allow login for registered user.
Dependent on Requirements	FR1
Stake Holder Owning the Requirement	End User
Example of user/system interaction for this requirement	<p>A user should be able to login from anywhere.</p> <p>End user 'A' is already a registered user who wants to login into the app.</p>

Requirement Tag	FR3
Requirement Statement	The system should allow the user to browse the movies.
Dependent on Requirements	-
Stake Holder Owning the Requirement	End User
Example of user/system interaction for this requirement	End user 'A' should be able to browse the movies currently on screen and the upcoming movies.

Requirement Tag	FR4
Requirement Statement	The system should filter the movies on basis of language/show time and location.
Dependent on Requirements	FR3
Stake Holder Owning the Requirement	End User
Example of user/system interaction for this requirement	End user 'A' should be able to filter the movies according to location near him, language of choice, and show timings.

Requirement Tag	FR5
Requirement Statement	The system should show the upcoming movies.
Dependent on Requirements	-
Stake Holder Owning the Requirement	End User
Example of user/system interaction for this requirement	End user 'A' should be able to browse the upcoming movies and their details.

Requirement Tag	FR6
Requirement Statement	The system should show details of a particular movie upon selection.
Dependent on Requirements	-
Stake Holder Owning the Requirement	End User
Example of user/system interaction for this requirement	End user 'A' should be able to browse the details of the selected movie.

Requirement Tag	FR7
Requirement Statement	The system should show option to select seats.
Dependent on Requirements	FR6
Stake Holder Owning the Requirement	End User
Example of user/system interaction for this requirement	End user 'A' should be able to select the seats of the selected movie.

Requirement Tag	FR8
Requirement Statement	The system should show payment options.
Dependent on Requirements	FR7, FR2
Stake Holder Owning the Requirement	End User
Example of user/system interaction for this requirement	End user 'A' who is logged in should be able to pay for the selected movie and seat.

Requirement Tag	FR9
Requirement Statement	The system should show the order summary and provide the e –ticket.
Dependent on Requirements	FR8, FR7, FR6, FR2
Stake Holder Owning the Requirement	End User
Example of user/system interaction for this requirement	End user 'A' who has completed the payment should receive an e- ticket.

Requirement Tag	FR10
Requirement Statement	The system should allow the user to logout.
Dependent on Requirements	FR2
Stake Holder Owning the Requirement	End User
Example of user/system interaction for this requirement	End user 'A' who is logged in the app should be able to logout.

5. Presentation of Results

6. Analysis and Discussions

The requirements of the model are distinguished with reference to skeleton of the session one to elaborate and use of the data in more lengthy way.

7. Comments

1. Limitations of Experiments

It requires a systematic and more linear scale to jot down the requirements.

2. Limitations of Results

After each step, it is making sure the previous is sustainable.

3. Learning happened

With the help of the requirements of the lab session one and its initial requirements we make the tables.

4. Recommendations

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

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. Calculations/Computations/Algorithms

5. Presentation of Results

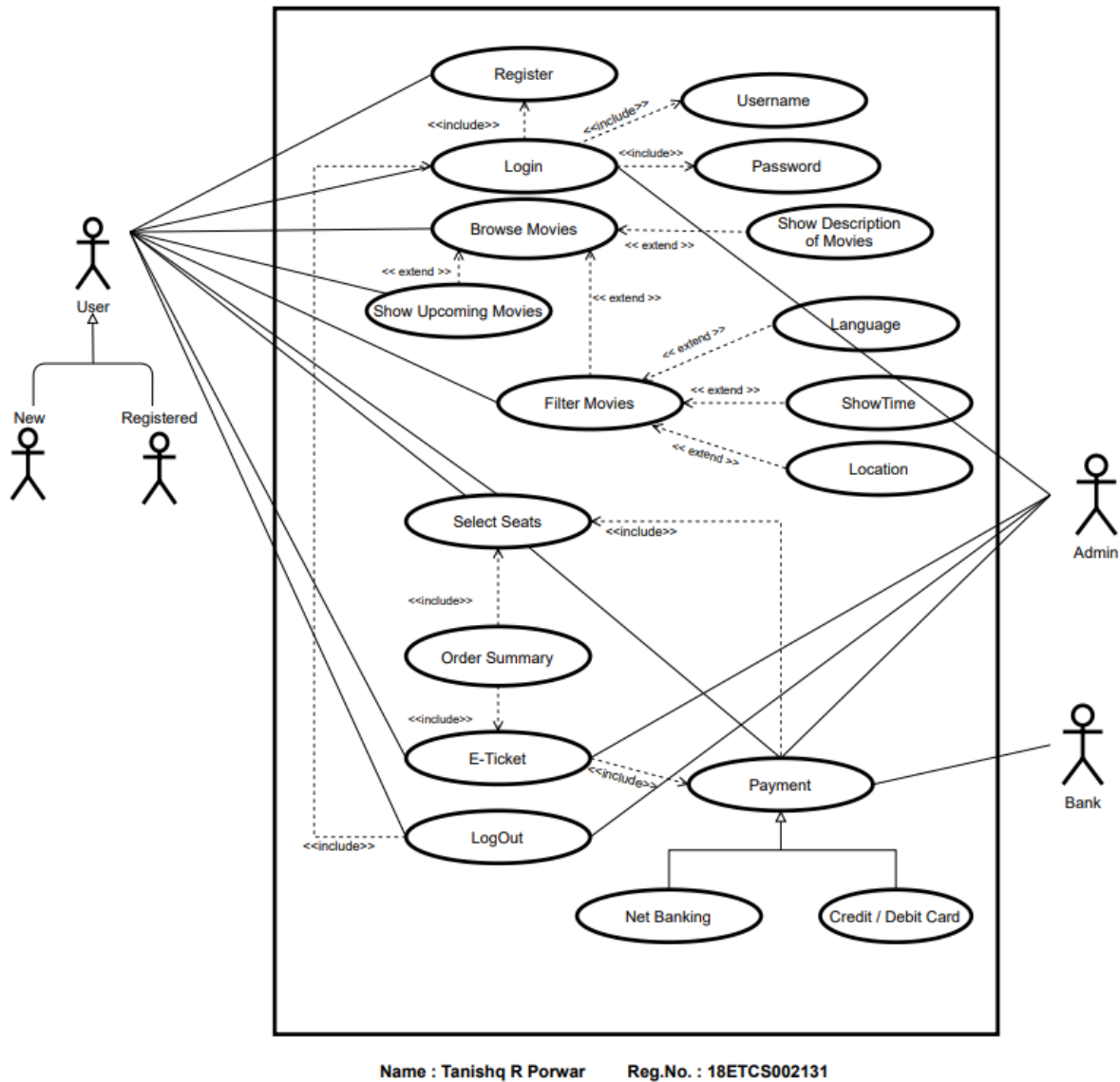


Fig. 3.1 represents Use Case Diagram

6. Analysis and Discussions

We discussed about the relationship between the various functional requirements (generalization and dependency) and have analyzed the dependency of requirements

(include and extends). We have 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.

7. Conclusions

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

8. Comments

1. Limitations of Experiments

The requirements of this model are frozen and cannot be changed.

2. Limitations of Results

If the customer changes any requirements, the process will be reverted from first that makes the process a lengthy and time consuming.

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

Laboratory 4

Title of the Laboratory Exercise: Sequence diagram

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

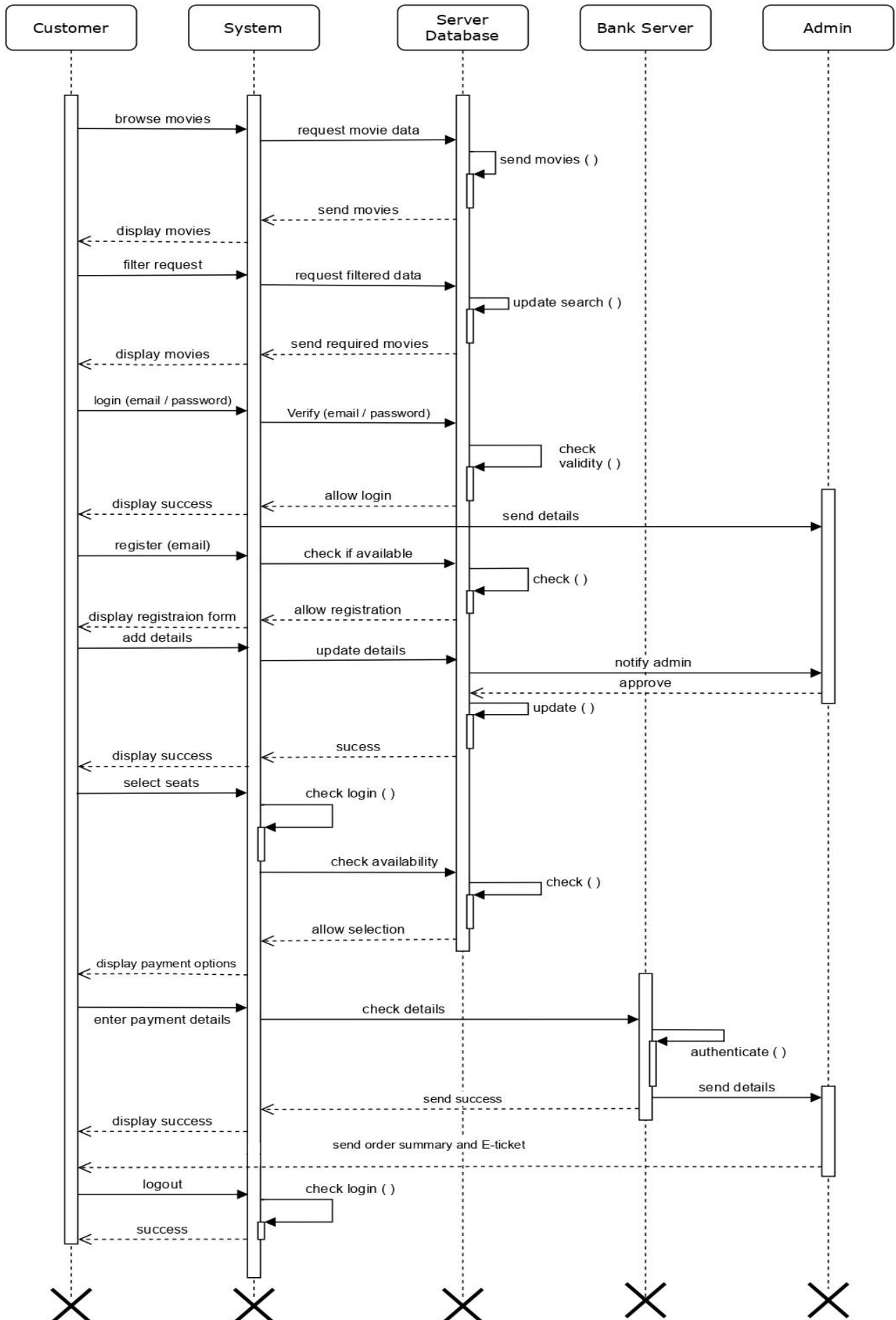
3. Experimental Procedure

- 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

Required use case diagram shows in presentation of results.

5. Presentation of Results



6. Analysis and Discussions

We have discussed about the sequence diagram and its components. We have analyzed 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	

Laboratory 5

Title of the Laboratory Exercise Class diagram

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 study the given requirements and develop use case diagrams and Sequence diagrams

Objectives

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

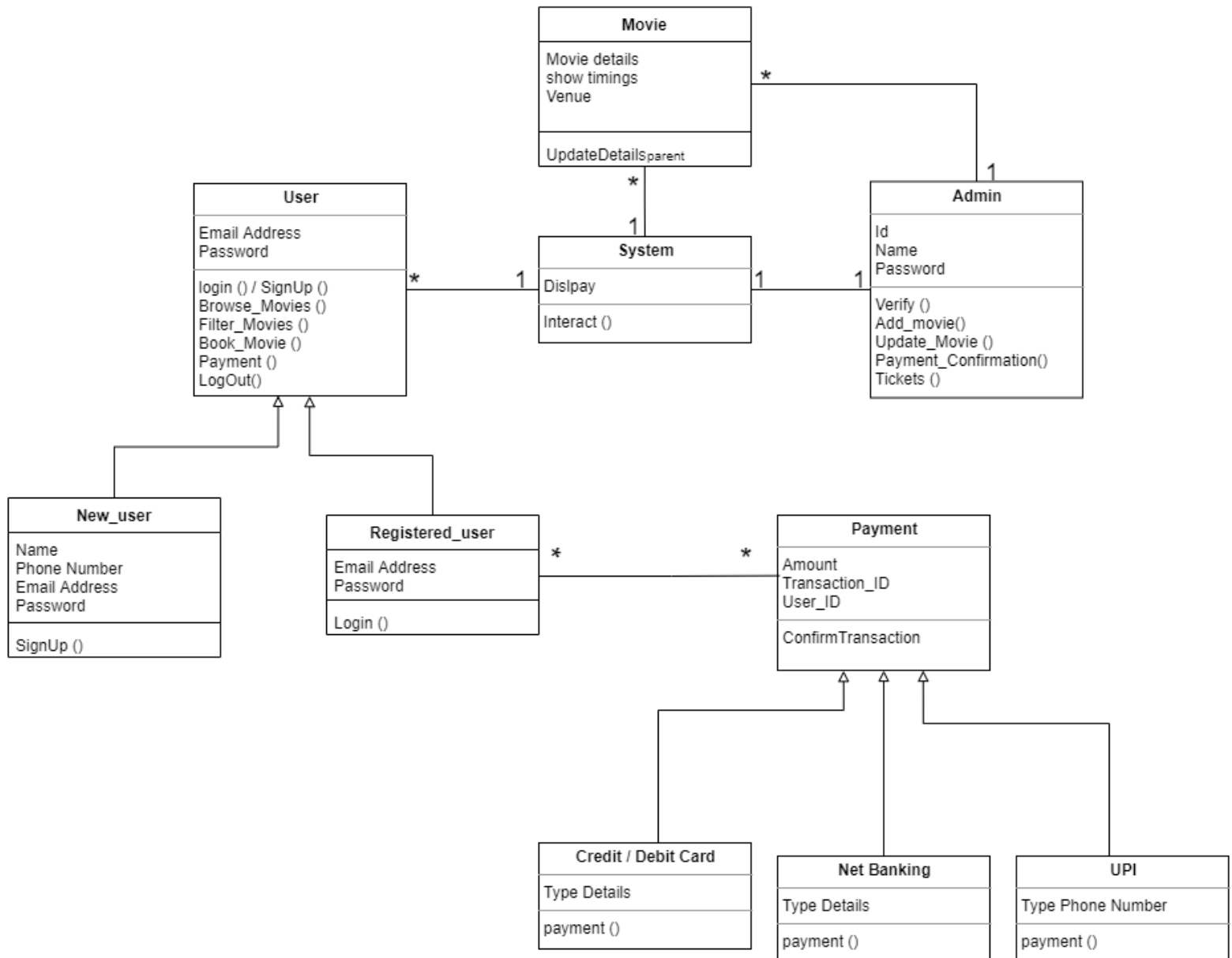
- Explain the purpose of the sequence diagram
- Identify the logical sequence of activities undergoing in a system, and represent them pictorially
- Design and model a given use case using UML-sequence diagrams

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. Calculations/Computations/Algorithms

5. Presentation of Results



6. 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

7. Conclusions

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

8. 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	

Laboratory 6

Title of the Laboratory Exercise: State chart Diagrams

Introduction and Purpose of Experiment

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

1. 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

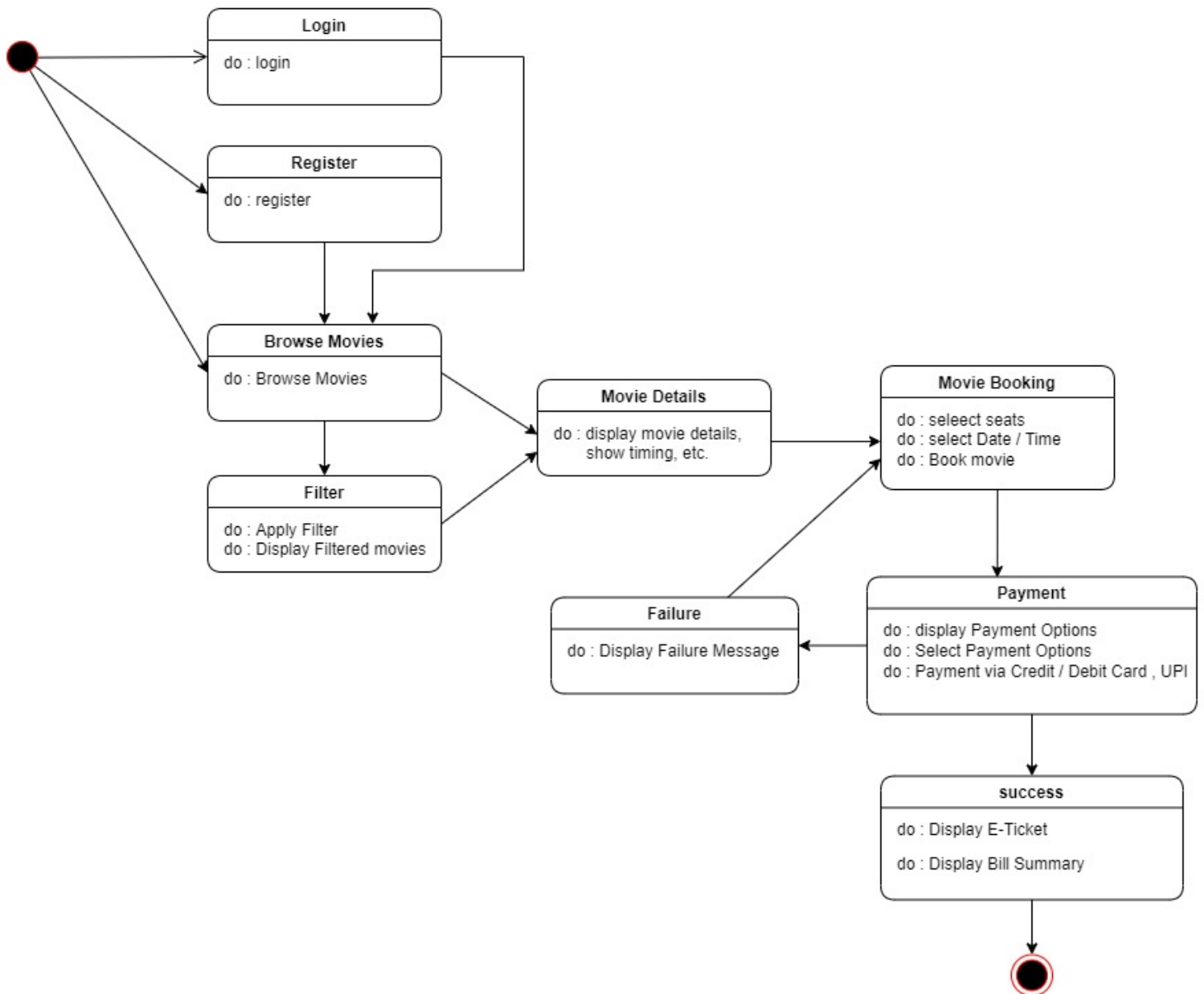
- Identify the main members of the family
- Identify how they are related to each other
- Find the characteristics of each family member
- Determine relations among family members
- Decide the inheritance of personal traits and characters

2. 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

3. Calculations/Computations/Algorithms

State Chart Diagram:



4. Analysis and Discussions

5. Comments

- Limitations of Experiments
- Learning happened
- Recommendations

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

Laboratory 7

Title of the Laboratory Exercise: Activity Diagrams

8. Introduction and Purpose of Experiment

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

9. Aim and Objectives

Aim

- To construct an activity diagram for a given system

Objectives

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

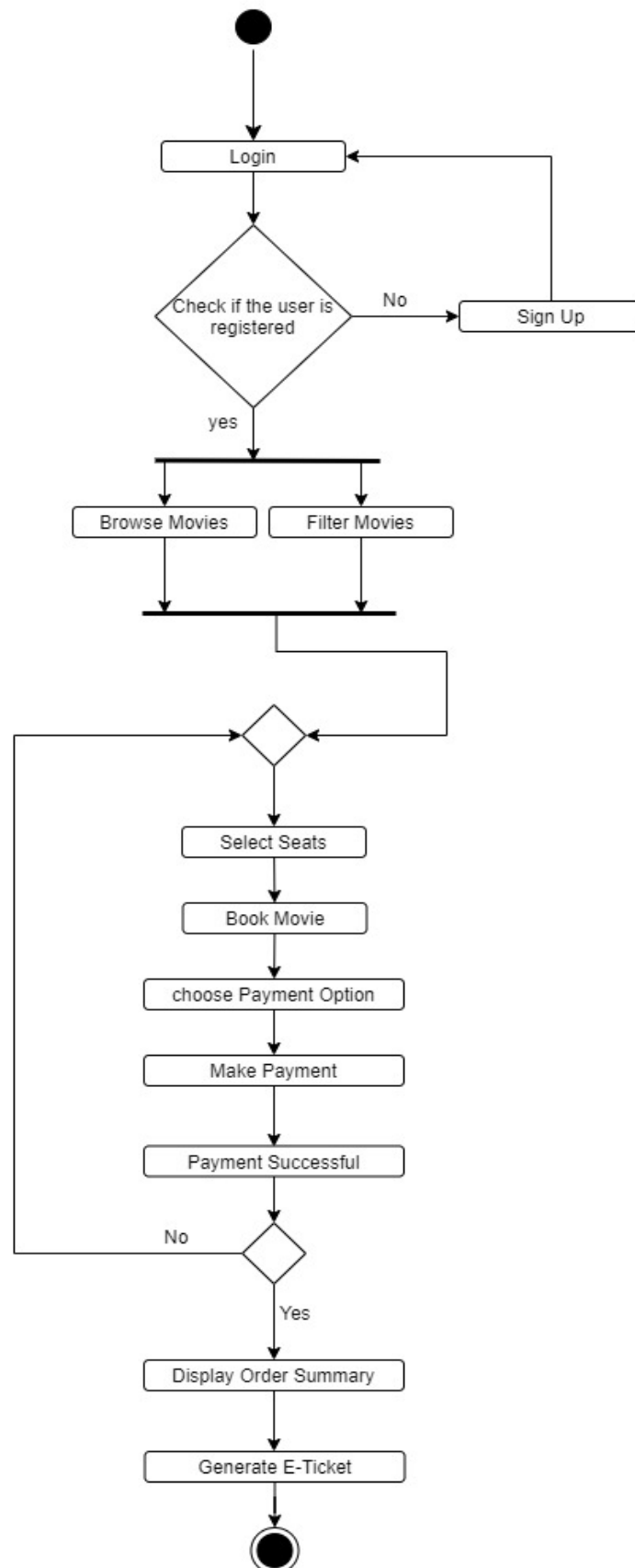
- Identify the main members of the family
- Identify how they are related to each other
- Find the characteristics of each family member
- Determine relations among family members
- Decide the inheritance of personal traits and characters

10. 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

11. Calculations/Computations/Algorithms

12. Presentation of Results



1. Analysis and Discussions

In this lab we worked on the activity diagram and the state chart diagram, which are behavioral 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 determined the flow of the activities. 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.

2. Conclusions

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

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

Laboratory 8

Title of the Laboratory Exercise: Entity-Relationship diagram

1. Introduction and Purpose of Experiment

Students will use JavaDoc and similar standards to ensure good documentation and maintainability of code

2. Aim and Objectives

Aim

- To construct an entity for a given system

Objectives

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

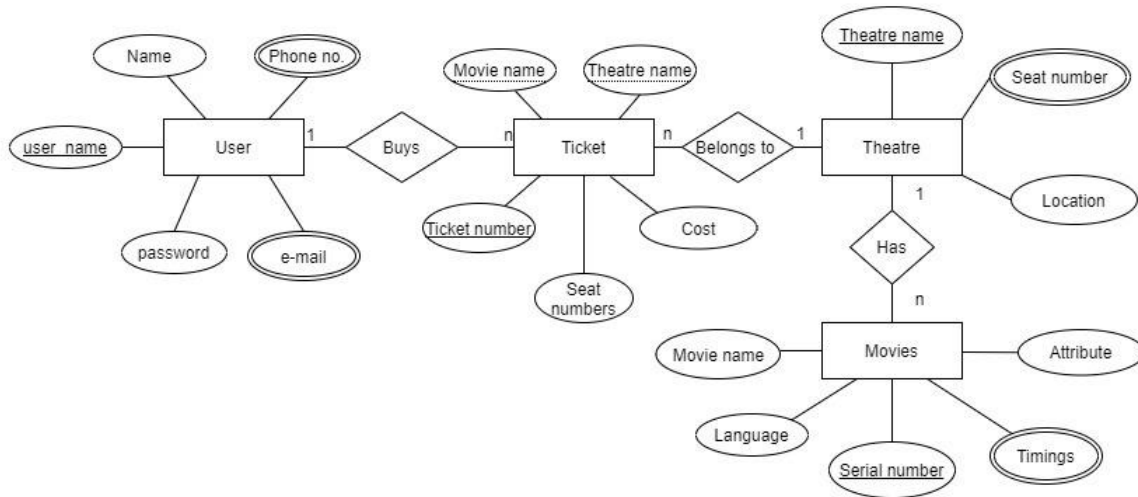
- Apply industry standard coding standards
- Use automatic documentation tools
- Create maintainable code

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

ER diagram:



13. Analysis and Discussions

14. Comments

- Limitations of Experiments
- Learning happened

4. Recommendations

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

Laboratory 9

Title of the Laboratory Exercise: Data Flow Diagrams

15. Introduction and Purpose of Experiment

Students will use JavaDoc and similar standards to ensure good documentation and maintainability of code

16. Aim and Objectives

Aim

- To construct a data flow diagram for a given system

Objectives

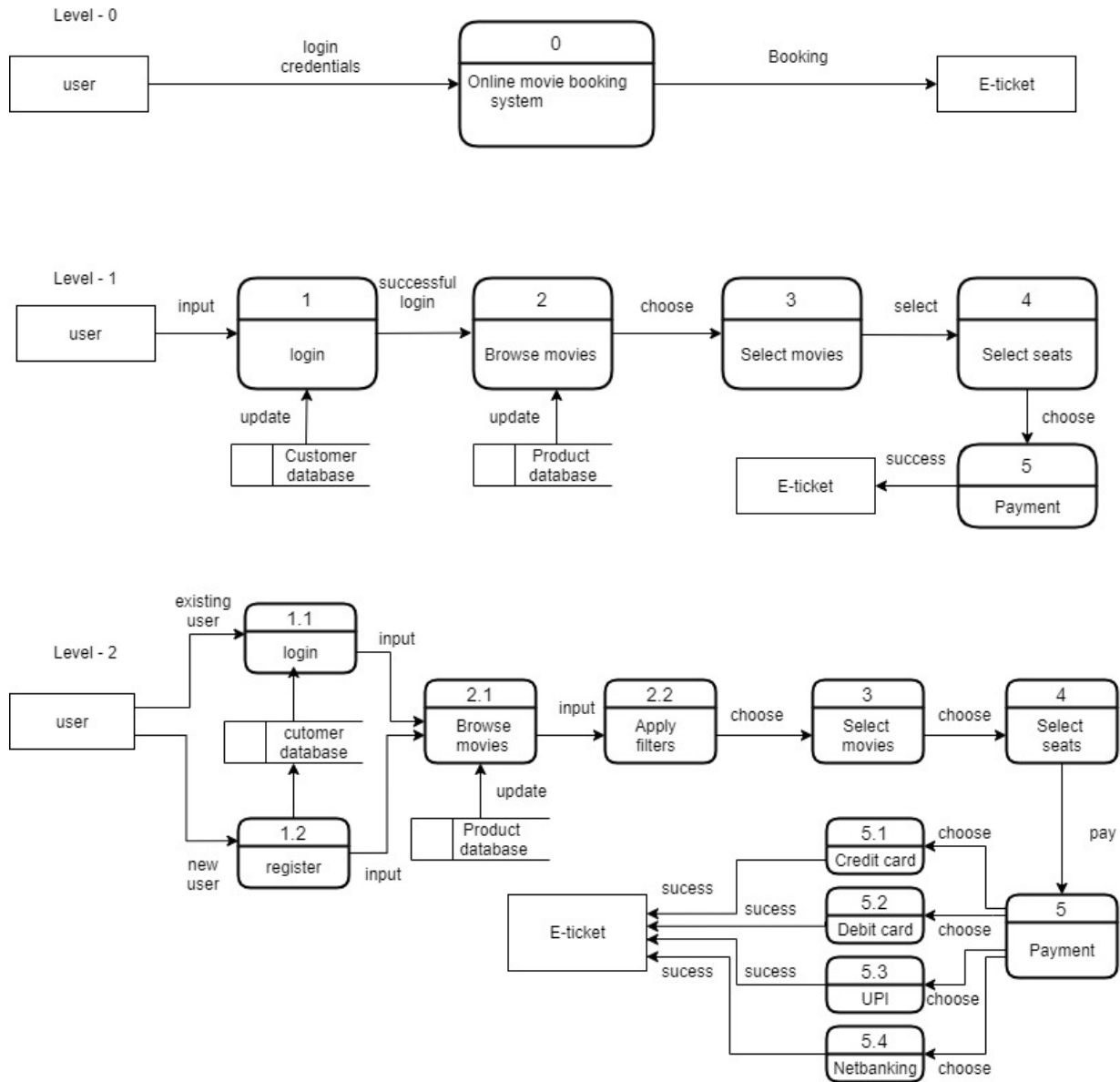
At the end of this lab, the student will be able to

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

17. 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

18. Calculations/Computations/Algorithms



19. 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.

20. Comments

- Limitations of Experiments

The disadvantage of DFD is that if there is 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.

4. Recommendations

I recommend OBJECT ORIENTED DIAGRAMS over data flow diagrams.

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

Laboratory 10

Title of the Laboratory Exercise: Implementation and Coding Standards

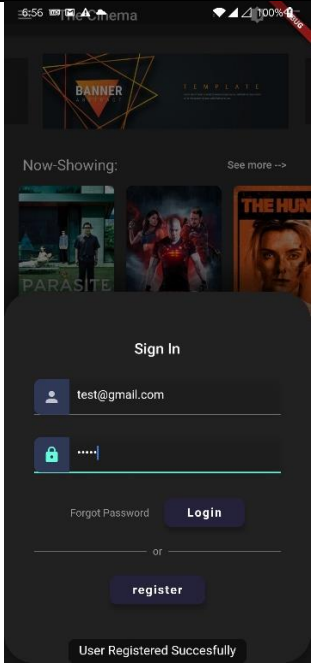
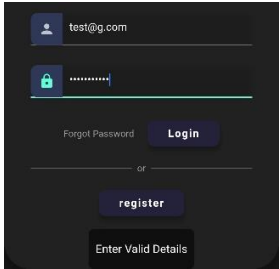
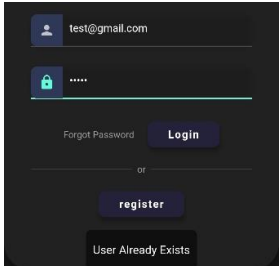
Implementation

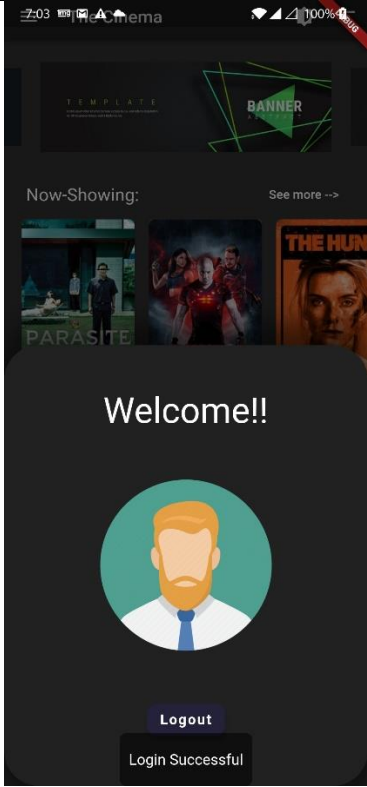
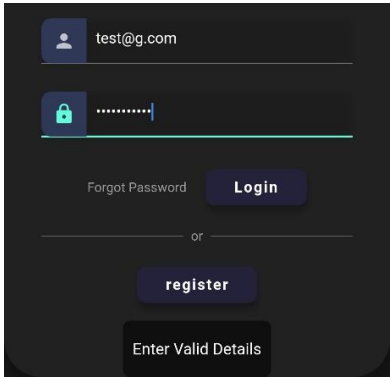
Code: Full code [here](#), see the full implementation and recording of working model [here](#)

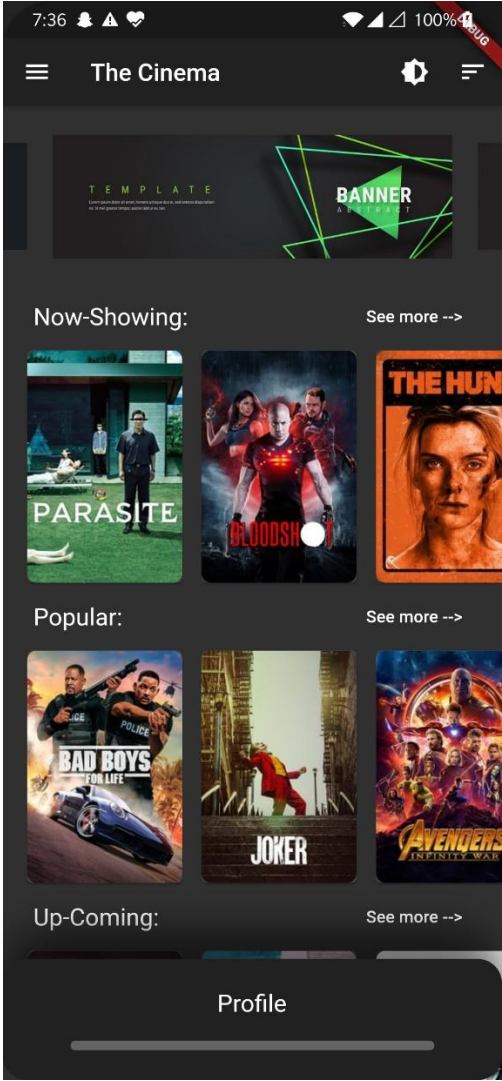
Features:

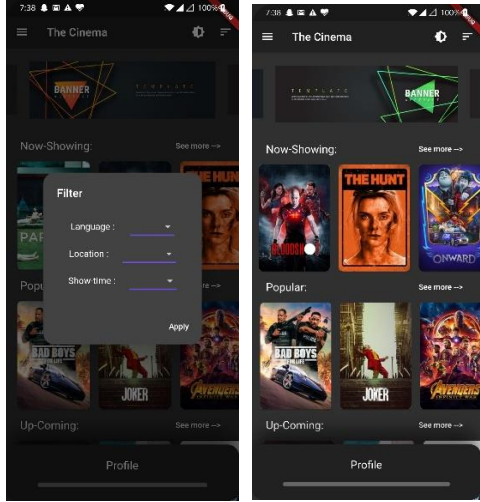
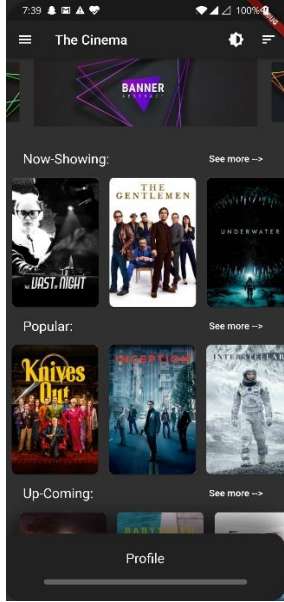
- Browse movies
 - Register new users and login existing users
 - Filter Movies based on location, language and show time
 - Show Upcoming movies
 - Show Movie Details page
 - Choose seats
 - Show different payment options
 - Generate e-ticket
 - Toggle between dark theme and light
-
- This project is built using flutter which is a Google's UI toolkit for building beautiful, natively compiled applications for mobile both IOS and android, web, and desktop from a single codebase.
 - The movie data is fetched from [themoviedb](#) using GET API request.
 - The state of the app is managed using the BLoC pattern in flutter, BLoC a.k.a **Business Logic Components**. The BLoC pattern uses Reactive Programming to handle the flow of data within an app. A BLoC stands as a middleman between a source of data in your app (e.g. an API response) and widgets that need the data. It receives streams of events/data from the source, handles any required business logic and publishes streams of data changes to widgets that are interested in them.
 - The payment is made using [razorpay's](#) payment portal, it allows to make payment through cards, UPI, e-wallets, net banking, etc

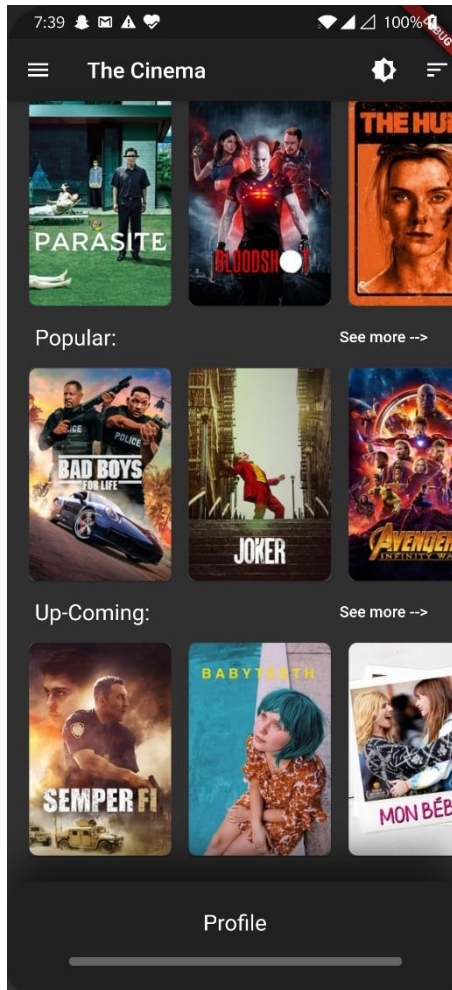
Testing



TEST CASE ID	FUNCTIONAL REQUIREMENT NUMBER	TEST CASE DESCRIPTION	TEST DATA	EXPECTED OUTPUT	ACTUAL OUTPUT	TEST RESULT
TC 1	FR 1	<p>The system should allow new users to register</p> <p>Test Steps: -</p> <p>Step 1. Launch the application and click slide-up the bottom panel</p> <p>Step 2. Fill the email and password</p> <p>Step 3. Click on the register button.</p>	<p>Valid Data +++++++ email: test@gmail.com Password: 12345</p> <p>Invalid Data +++++++ email: test@g.com Password: abc1</p> <p>email: test@gmail.com Password: 12345</p>	<p>“User Registered Successfully” message is displayed.</p> <p>“Enter valid details” message is displayed.</p> <p>“User already exists” message is displayed.</p>	  	<p>PASS</p> <p>PASS</p> <p>PASS</p>


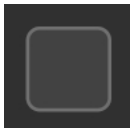

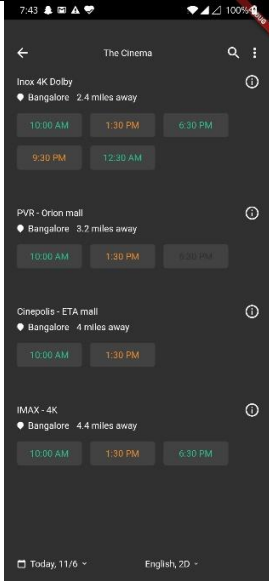
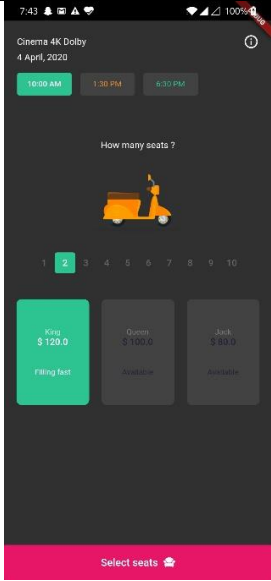
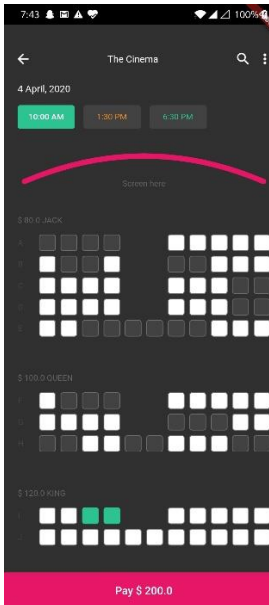
TEST CASE ID	FUNCTIONAL REQUIREMENT NUMBER	TEST CASE DESCRIPTION	TEST DATA	EXPECTED OUTPUT	ACTUAL OUTPUT	TEST RESULT
TC 2	FR 2	<p>The system should allow registered user to login</p> <p>Test Steps: -</p> <p>Step 1. Launch the application and click slide-up the bottom panel</p> <p>Step 2. Fill the email and password</p> <p>Step 3. Click on the login button.</p>	<p>Valid Data +++++++ email: test@gmail.com Password: 12345</p> <p>Invalid Data +++++++ email: test@gmail.com Password: 123456</p> <p>email: test@g.com Password: 12345</p>	<p>“User Logged In Successfully” message is displayed.</p> <p>“Enter valid details” message is displayed.</p>	 	<p>PASS</p> <p>PASS</p>

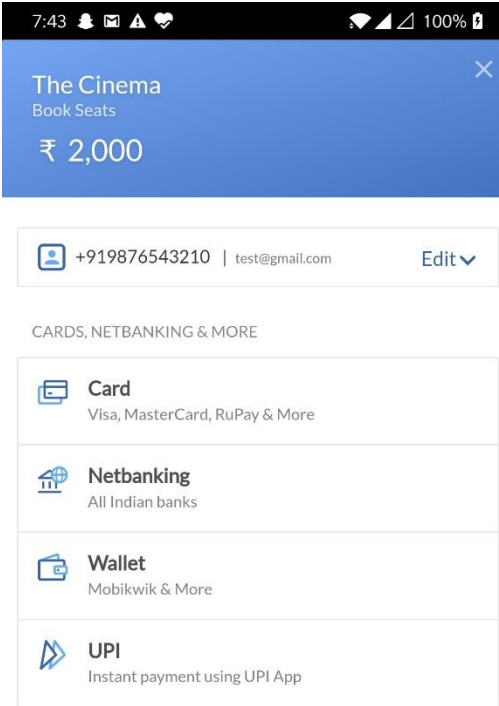
TEST CASE ID	FUNCTIONAL REQUIREMENT NUMBER	TEST CASE DESCRIPTION	TEST DATA	EXPECTED OUTPUT	ACTUAL OUTPUT	TEST RESULT
TC 1	FR 3	<p>The system should allow users to browse movies available to watch.</p> <p>Test Steps: - Step 1. Launch the application.</p> <p>(Movies can be browsed without Sign-In in the app)</p>		List of Now-showing movies to browse from	 <p>The screenshot shows the 'The Cinema' app interface. At the top, there's a header with the app name 'The Cinema' and navigation icons. Below the header is a banner area with a 'TEMPLATE' placeholder and a 'BANNER' placeholder. The main content is divided into three sections: 'Now-Showing:', 'Popular:', and 'Up-Coming:'. Each section has a 'See more -->' link. The 'Now-Showing' section displays three movie posters: 'PARASITE', 'BLOODSH', and 'THE HUN'. The 'Popular' section displays three movie posters: 'BAD BOYS', 'JOKER', and 'AVENGERS'. The 'Up-Coming' section is partially visible at the bottom. A 'Profile' button is located at the bottom of the screen.</p>	PASS

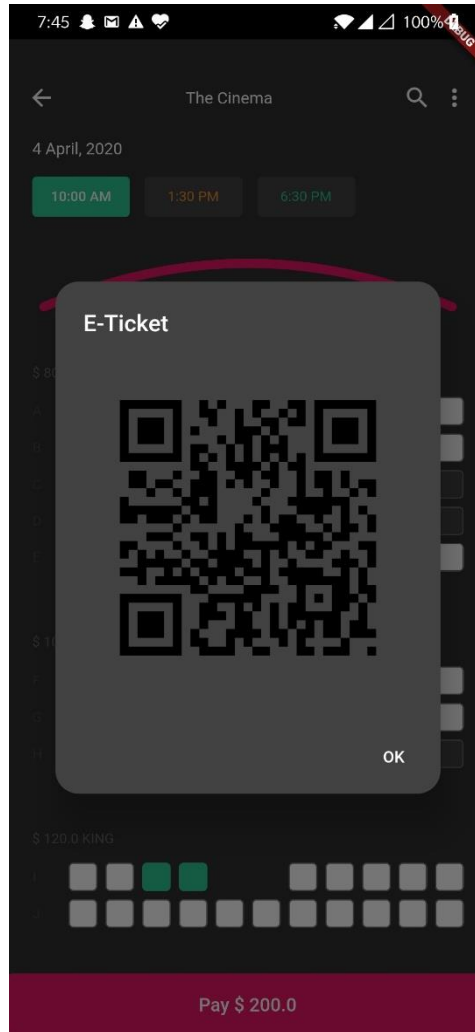
TEST CASE ID	FUNCTIONAL REQUIREMENT NUMBER	TEST CASE DESCRIPTION	TEST DATA	EXPECTED OUTPUT	ACTUAL OUTPUT	TEST RESULT
TC 3	FR 4	<p>The system should allow users to filter movies.</p> <p>Test Steps: -</p> <p>Step 1. Launch the application</p> <p>Step 2. Click on the filter button in the navbar</p> <p>Step 3. Select the language, location and show timing, then click on apply</p>	<p>Valid Data +++++++ Select language English</p> <p>Valid Data +++++++ Select location Whitefield</p>	<p>Only English Movies are displayed.</p> <p>Movies playing at Whitefield are displayed.</p>	 	<p>PASS</p> <p>PASS</p>

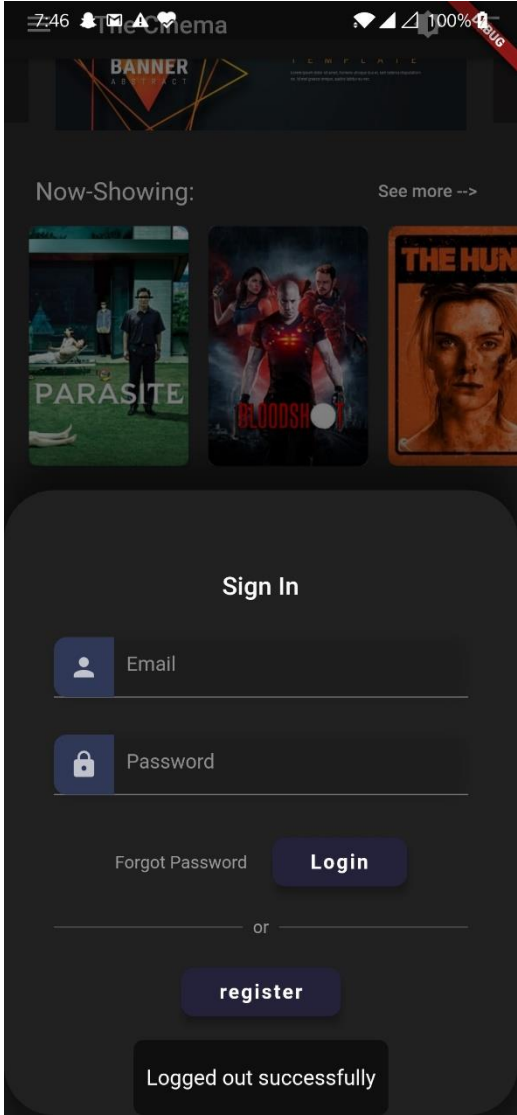
TEST CASE ID	FUNCTIONAL REQUIREMENT NUMBER	TEST CASE DESCRIPTION	TEST DATA	EXPECTED OUTPUT	ACTUAL OUTPUT	TEST RESULT
TC 1	FR 5	<p>The system should allow users to browse upcoming movies.</p> <p>Test Steps: - Step 1. Launch the application. Step 2. Scroll down to up-coming movies.</p> <p>(Movies can be browsed without Sign-In in the app)</p>		Movie Posters with their respective name and rating at bottom.	 <p>The screenshot shows the 'The Cinema' app interface. At the top, there's a header with the app name 'The Cinema', a settings gear icon, and a menu icon. Below the header, there are three rows of movie posters. The first row shows 'PARASITE', 'BLOODSHOT', and 'THE HUNT'. The second row is labeled 'Popular:' and shows 'BAD BOYS', 'JOKER', and 'AVENGERS: INFINITY WAR'. The third row is labeled 'Up-Coming:' and shows 'SEMPER FI', 'BABYTEETH', and 'MON BÉBÉ'. At the bottom, there's a 'Profile' button. The status bar at the top shows the time as 7:39, battery at 100%, and various notification icons.</p>	PASS

TEST CASE ID	FUNCTIONAL REQUIREMENT NUMBER	TEST CASE DESCRIPTION	TEST DATA	EXPECTED OUTPUT	ACTUAL OUTPUT	TEST RESULT
TC 4	FR 6	<p>The software should display movie description.</p> <p>Test Steps: - Step 1. Launch the application.</p> <p>Step 2. Click on the movie poster to view the description of the movie. (movie description can be viewed without signing-in.)</p>	<p>Valid Data +++++++ Click on <i>Bad Boys for life</i> Movie Poster.</p> <p>Valid Data +++++++ Click on <i>The Hunt</i> Movie Poster.</p>	<p>Description like Release date, Rating, Genre, Runtime, Cast, Director, IMDB rating etc. of <i>Bad Boys for life</i> movie is displayed.</p> <p>Description of <i>The Hunt</i> movie is displayed.</p>	 	<p>PASS</p> <p>PASS</p>

TEST CASE ID	FUNCTIONAL REQUIREMENT NUMBER	TEST CASE DESCRIPTION	TEST DATA	EXPECTED OUTPUT	ACTUAL OUTPUT	TEST RESULT
TC 5	FR 7	<p>The system should allow the user to select seats</p> <p>Test Steps: -</p> <p>Step 1. Launch the application and open the slide-up panel</p> <p>Step 2. Fill the form and Click on login.</p> <p>Step 3. After Successful login , click on any movie poster.</p> <p>Step 4. Click on Book Seat button</p> <p>Step 5. Select the desired time slot from the desired theatre</p> <p>Step 6. Select seat from available seats.</p> <p>Available seats and Reserved Seats are shown differently.</p> <p>Available Seats are shown using white color</p>  <p>Reserved seats are shown using grey color</p>  <p>Selected seats are shown using teal color</p> 		<p>Available Seats are shown using white color</p> <p>Reserved seats are shown using grey color</p> <p>Selected seats are shown using teal color</p>	  	PASS

TEST CASE ID	FUNCTIONAL REQUIREMENT NUMBER	TEST CASE DESCRIPTION	TEST DATA	EXPECTED OUTPUT	ACTUAL OUTPUT	TEST RESULT
TC 6	FR 8	<p>The software should support multiple payment options</p> <p>Test Steps: -</p> <p>After selecting seats for the desired movie</p> <p>Step 1. Click on pay button</p> <p>Step 2. Select the payment option and fill the details</p>		Multiple Payment options are displayed.	 <p>The screenshot shows a mobile app interface. At the top, there's a status bar with the time 7:43 and various icons. Below it, a blue header bar contains the text 'The Cinema' and 'Book Seats', with a close button (X) on the right. The main content area displays the amount '₹ 2,000'. Below this is a white box containing a phone number '+919876543210', an email 'test@gmail.com', and an 'Edit' button with a dropdown arrow. Underneath is a section titled 'CARDS, NETBANKING & MORE' which lists four payment options: 'Card' (with a card icon and subtext 'Visa, MasterCard, RuPay & More'), 'Netbanking' (with a bank icon and subtext 'All Indian banks'), 'Wallet' (with a wallet icon and subtext 'Mobikwik & More'), and 'UPI' (with a UPI icon and subtext 'Instant payment using UPI App').</p>	PASS

TEST CASE ID	FUNCTIONAL REQUIREMENT NUMBER	TEST CASE DESCRIPTION	TEST DATA	EXPECTED OUTPUT	ACTUAL OUTPUT	TEST RESULT
TC 9	FR 9	<p>The system should give order summary by generating e-ticket.</p> <p>Test Steps: -</p> <p>After the payment is successful, the e-ticket is generated.</p>		<p><i>"Payment Successful"</i> message is displayed with a QR code containing all the details of the Movie booking.</p> <p>If we scan the QR code, the message displayed is</p> <p><i>"Welcome to Cinemasy Movie: Beauty and the Beast Date: May-12-2020 Show Time: 11:30 Seat No.: 15 "</i></p>		PASS

TEST CASE ID	FUNCTIONAL REQUIREMENT NUMBER	TEST CASE DESCRIPTION	TEST DATA	EXPECTED OUTPUT	ACTUAL OUTPUT	TEST RESULT
TC 10	FR 10	<p>The system should allow user to logout</p> <p>Test Steps: - Step 1. Launch the application. Step 2. Open the slide-up panel Step3. Click on logout button</p>		User is logged-out		PASS

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