TITLE OF THE PROJECT

LION CINEMA

Submitted by

Dominic Cleary Gomes

(ID: 191-16-405)

Submitted to

Abdullah Bin Kashem Bhuiyan

Lecturer of CIS Department (DIU)

Submission Date: 14 January 2023

Course code: CIS499

Department of Computing and Information System (CIS)

Daffodil International University

Fall – 2022

APPROVAL

This Project titled "Lion Cinema", Submitted by Dominic Cleary Gomes, ID No:191-16-405 to the Department of Computing & Information Systems, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computing & Information Systems and approved as to its style and contents. The presentation has been held on- 14-01-2023.

BOARD OF EXAMINERS

Mr. Md Sarwar Hossain Mollah — **Chairman**
Associate Professor and Head
Department of Computing & Information Systems
Faculty of Science & Information Technology
Daffodil International University

Mr. Md. Mehedi Hassan — **Internal Examiner**
Lecturer
Department of Computing & Information Systems
Faculty of Science & Information Technology
Daffodil International University

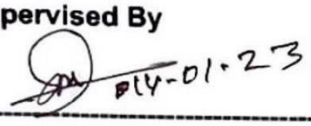
Mr. Syed Tangim Pasha — **Internal Examiner**
Lecturer
Department of Computing & Information Systems
Faculty of Science & Information Technology
Daffodil International University

Dr. Saifuddin Md. Tareeq — **External Examiner**
Professor & Chairman
Department of Computer Science and Engineering
University of Dhaka, Dhaka

DECLARATION

I hereby declare that; this project has been done by me under the supervision of **Abdullah Bin Kashem Bhuiyan, Lecturer,** Department of Computing and Information System (CIS) of Daffodil International University. I am also declaring that this project or any part of there has never been submitted anywhere else for the award of any educational degree like, B.Sc., M.Sc., Diploma or other qualifications.

Supervised By



Abdullah Bin Kashem Bhuiyan
Former Lecturer
Department of CIS
Daffodil International University

Submitted By

Dominic Cleary Gomes
ID: 191-16-405
Department of CIS
Daffodil International University

ACKNOWLEDGEMENT

First and foremost, I want to thank and thank God for giving me the ability to finish the project and documentation. Furthermore, my sincere appreciation to my supervisor Abdullah Bin Kashem Bhuiyan for all of his assistance with the project. Your insightful counsel and recommendations were quite beneficial to me as I finished the project and documentation in time. I will always be grateful to you for this.

To wrap things up, I might offer my thanks to Daffodil International University's faculty members and my friends who supported me to finish the project.

DEDICATION

I ought to express a special appreciation to my parents because without their blessings, encouragement, and motivation I would be nothing more than an empty shell. I will always be grateful to them.

EXECUTIVE SUMMARY

Lion Cinema is my first internship project. It is basically for both Android and iOS mobile devices, Lion Cinema is a flutter-based dart-based mobile application. By downloading this app, users can effortlessly plan their schedules to meet their favorite hours, view upcoming movies, and get a list of the movies now showing in theaters. Additionally, readers can read about the movie's genre, title, trailer, showtimes, and other relevant details. Using a debit/credit card, mobile banking, or net banking is simple for the user to make payments.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER1: INTRODUCTION

1.1 Introduction

We are in a scientific era. On this technologically advanced planet, everything has become an Internet-based program. It has got many changes in our daily lives. Science has given us many ways of recreation. Such as ticket booking to watch cinemas in theaters. Nowadays people can buy cinema tickets online. Those who love to watch movies in theaters but are busy with office work throughout the week definitely won't like to go to cinema halls and stand in long queues to buy tickets so booking tickets using their mobile devices will be much easier for them. This project aims to give users the option to reserve movie tickets by using their mobile devices, allowing them to do so whenever and wherever they want. Lion Cinema is primarily designed to offer clients a 24/7 service for reserving a seat in the theater and gathering online movie information. The consumer may simply learn about the movies that have been released and then make a decision. For people who live far away from the theater, this project will make it an easy way to book movie tickets using mobile devices instead of standing in queues to buy tickets and it will also save them time and money. So, it offers an immense multiplex online ticket booking solution.

1.2 Report Contents

This documentation will contain the following chapters:

Chapter 1: Introduction

An overview of the project is presented in this section.

Chapter 2: Initial Study

This section includes a thorough explanation of the subject field, approaches, an assessment of existing strategies, and proposals.

Chapter 3: Literature Review

This section includes a thorough explanation of the issue field, approaches, an assessment of current solutions, and recommendations.

Chapter 4: Methodology

This section covers aspects of the methodology that has been used, why it is used, the methodology itself, and its execution.

Chapter 5: Planning

The purpose of this segment is to discuss the undertaking planning process for this project.

Chapter 6: Feasibility

In this part, a specific perspective on all kinds of feasibility and an overview of cost-benefit analysis are provided

Chapter 7: Foundation

This section includes a list of the project's requirements as well as the problem area and other potential solutions.

Chapter 8: Exploration

This section includes some necessary diagrams and prototypes of the project

Chapter 9: Engineering

This section covers the conceptual and functional architecture of the suggested approach.

Chapter 10: Deployment

This section covers some necessary coding samples and prioritization of the listed requirements for the project.

Chapter 11: Testing

During this segment, different types of testing have been conducted and results are recorded.

Chapter 12: Implementation

Here, the execution plan, training model, and associated topics are covered.

Chapter 13: Critical Evaluation

This section shows the project objectives that are covered and those that are not covered

Chapter 14: Lessons Learned

This section shows the learning outcomes from the project and the problems encountered during the project

Chapter 15: Conclusion

Here, a short summary has been given from the project

CHAPTER 2: INITIAL STUDY

2.1 Project Proposal

The project proposal section of the documentation is used to provide background and a more detailed description of the proposed study. The title of the report should give an accurate and succinct summary of the purpose of the report. In addition, it indicates the background study of the project, the significance of the project, findings, and conclusions.

2.2 Background Study

In our nation, there are many movie theaters, but not all of them offer an online system for purchasing tickets. Many well-known movie theaters across the nation, including **Star Cineplex** and **Blockbuster Cinemas**, offer web-based online ticket booking. Though **Star Cineplex** also has an android app, however, there are many negative user reviews for it. The fact that it is exclusively accessible through the Play Store indicates that android users are primarily their target market. Because it is not available in App Store, iOS users cannot locate it there. Therefore, there is no decent app available for purchasing tickets for both android and IOS devices. These days mobile devices are easy and convenient to carry out such online transactions. Purchasing tickets from an app is more convenient than purchasing from the web. So, this project will simplify the procedure, increase its dependability and efficiency, and guarantee the security of the online transaction. Thus, there is a big market for this kind of app.

2.2.1 Description of the Work Project

Lion Cinema is a mobile application built in flutter using dart language for both android devices and IOS devices that helps users to purchase movie tickets whenever they want. By using this application users can find out the list of movies that are currently showing in the theater and will also be able to view upcoming movies and can easily plan their schedule according to their convenient time. Additionally, people can view movie details like genre, trailer, description, available show time, and date. Moreover, people can select seat the type (eg regular or premium) and they can also view the location of the theater seats through the app.

On the other hand, they can purchase tickets in various ways. They can purchase tickets through debit/credit cards if anyone doesn't have one, they can buy using **Bkash**, **Nagad**, **Rocket**, **Okwallet**, **Mycash**, **Dmoney**, **Tap,** or **Upay**. Internet banking channel is also available in this regard.

2.2.2 Feasibility Analysis

- **Operational Feasibility**

  It is a parameter used to determine how effectively a proposed system covers the functionalities and complies with requirements found during the requirements analysis stage of project development. **Lion Cinema** has a proper verification system to take user credentials, an optimized and smooth layout for different modules, a simple ticket reserving option that makes the app pretty user-friendly for ticket booking, and various online payment options for purchasing tickets. Admin can maintain the whole system from the backend by simply logging in with valid inputs.

- **Technical Feasibility**

  Previously ticket purchasing system was being used manually. It was quite difficult to manage and had taken up a lot of time in the manual process. Purchasing tickets by using an app is way more comfortable and secure for users too. The company wanted the application to be useful for both android and IOS. Therefore, the app will be developed in android and IOS based approach using **Flutter**. The app will also have a backend and admin panel.

  The app will be compatible to run in any android and IOS device with a low configuration because it will be created and developed utilizing the most recent and well-liked mobile technologies. Platform independence, the cost-effectiveness of the development platform, and the resources which will be used will make the application technically feasible.

- **Economic feasibility**

  Economic feasibility refers to the idea that a system must be profitable for the Organization in order to be created and utilized upon installation. According to the stage in which they occur, we divided the costs of the project. We already know that system development expenditures are often one-time expenses that

vanish after the project is over. We looked at specific cost categories to determine the development costs.

Mobile Application Cost:

Equipment	Cost per unit	Cost
Desktop pc (core i7 10th gen, 2.90 GHz / AMD 5 5600 G processor, 8 GB DDR4 RAM, 256 SSD)	৳ 65,000	৳ 65,000
Web, email, file servers	৳ 5,000	৳ 5,000
	Total	৳ 70,000

Table 1: Cost assessment for mobile based application has been conducted

Every device just needs to install the app to get services on the other hand it's quite simple and less costly.

Web Application Cost:

Equipment	Cost per unit	Cost
Desktop pc (core i7 10th gen, 2.90 GHz / AMD 5 5600 G processor, 8 GB DDR4 RAM, 256 SSD)	৳ 65,000	৳ 65,000
Web, email, file servers	৳ 5,000	৳ 5,000
Extranet network with VPN	৳ 2,000	৳ 2,000
	Total	৳ 72,000

Table 2: Cost assessment for web based application has been conducted

In this case, there is no requirement for installation because the complete application and data are stored on a server and are accessible at any time via the internet by using a browser. But a bit expensive.

Desktop Application Cost:

Equipment	Cost per unit	Cost
Desktop pc (core i7 10th gen, 2.90 GHz / AMD 5 5600 G processor, 8 GB DDR4 RAM, 256 SSD)	৳ 65,000	৳ 65,000
Web, email, file servers	৳ 5,000	৳ 5,000
	Total	৳ 70,000

Table 3: Cost assessment for desktop application has been conducted

Here, every desktop computer must have the software installed, which makes it inflexible. Additionally, for storing data the app requires a remote server. However, it is even more costly.

The company wanted the project to be mobile application based as it is less expensive and more useful. So as per their requirement, the project will be a mobile application based with a back-end and admin panel.

2.2.3 Market Research Analysis

Online movie ticket service is the practice of reserving seats and tickets for various entertainment events like movies through internet portals and without a doubt it's quite flexible for customers to buy tickets by using mobile applications or websites to watch movies in the theater. And research shows that customers find mobile applications preferable to web applications. Customers can reserve advanced tickets as well of their choice. Bookings can be made on mobile devices, and payments can be done as well. Moreover, it is far better than buying tickets in person. Aside from that, it has a ton of other benefits, including a huge variety of movies to choose from and time and money efficiency. It is predicted that this would increase market valuation and, as a result, will give market participants some respectable growth opportunities. Numerous factors, such as technical developments and the use of technology for convenience. So, Lion Cinema will be an efficient platform for the people of the country.

2.2.4 Project Goals

The project aims to develop innovative solutions that meet end-user requirements, thereby improving the accuracy, speed, and digitization of the application. To achieve this:

- Offer customers service at any time, wherever they may need it.
- Offer practical solutions to the easy accessibility of cinema tickets.
- Offer a simpler approach for us to select our seat and viewing spot for a movie
- To reduce the number of employees working at the ticket booth.
- To advertise online movie content.
- To offer suggestions and recommendations for the greatest theater in the area.
- To offer a better user experience that shows movies in transition, users can quickly view current movie details (eg genre, trailer, movie description) and make quick decisions
- Make every effort to increase profit.

2.2.5 Project Objectives

- We offer a complete solution for theater ticket booking through this project. The owner of theaters can sell tickets online with the help of this efficient mobile application.
- The “Lion Cinema” app’s primary goal is to provide an improved, mobile-based, accurate, user-friendly, secure transaction system and low-maintenance application.

The required high-level features for attaining the objectives and goals:

- Login approval for users to view the reserved tickets
- Secure administrative access to private management information.
- Manage upcoming and currently showing movies.
- Showing available movies.
- Showing schedules (show time and date) for movies
- Showing seat types for customers
- View purchased tickets
- Mobile number verification for purchasing tickets
- Login with email

- Showing movie details (genre, IMBD rating, trailer)
- Showing theater information such as hall number and movie type (3D or 3D)
- Showing seat availability for the particular movie

MoSCoW prioritization

SL	Requirements	Priority
01	Facility for login	Must have
02	Manage upcoming movies information	Must have
03	Manage now showing movies information	Must have
04	Manage seat and movie schedule	Must have
05	Phone number verification	Must have
06	Manage theater information	Should have
07	Email verification	Should have
08	Refund policy	Could have
09	Email verification	Could have

Table 4: Requirements using MosCow prioritization

Following the MoSCoW prioritization technique, the final features are:

- Login facility
- Ticket purchasing and viewing facility
- Available and upcoming movies list
- Theater seat selection facility
- Movie schedule
- OTP confirmation
- Refund policy

2.2.6 Key Tasks Toward Achieving the Goal

Task	Description
Initial study phase of the project	The project is being developed for providing customers an easy approach for purchasing movie tickets
Feasibility study	The project's operational, technical, and economic feasibility are evaluated to see if it is feasible enough to move on and meet its usefulness.
System requirements analysis	The types of equipment required to develop the project are examined in this section.
Data requirements	This section analyzes the types of information that must be kept in the database.
Project interface design	This section will analyze how the user interface will function. It has examined how simple it is for users to interact with the system.
Project development	In this section the entire project has been going through a systemic development process
Testing	This section will use every test that this project may require.
Documentation	This section will have the entire project report

Table 5: Tasks for achieving goals

2.2.7 Time Boxing

Task	Starting date	Ending date	Duration
Initial study phase of the project	11/09/2022	18/09/2022	7 days
Feasibility study	19/09/2022	26/09/2022	7 days
System requirements analysis	27/09/2022	01/10/2022	5 days
Data requirements analysis	02/10/2022	08/10/2022	6 days
Project interface design	09/10/2022	25/10/2022	16 days
Project development	26/10/2022	29/11/2022	35 days
Testing	30/11/2022	09/12/2022	10 days
Documentation	10/12/2022	20/12/2022	10 days
		Total	96 days

Table 6: Time box table

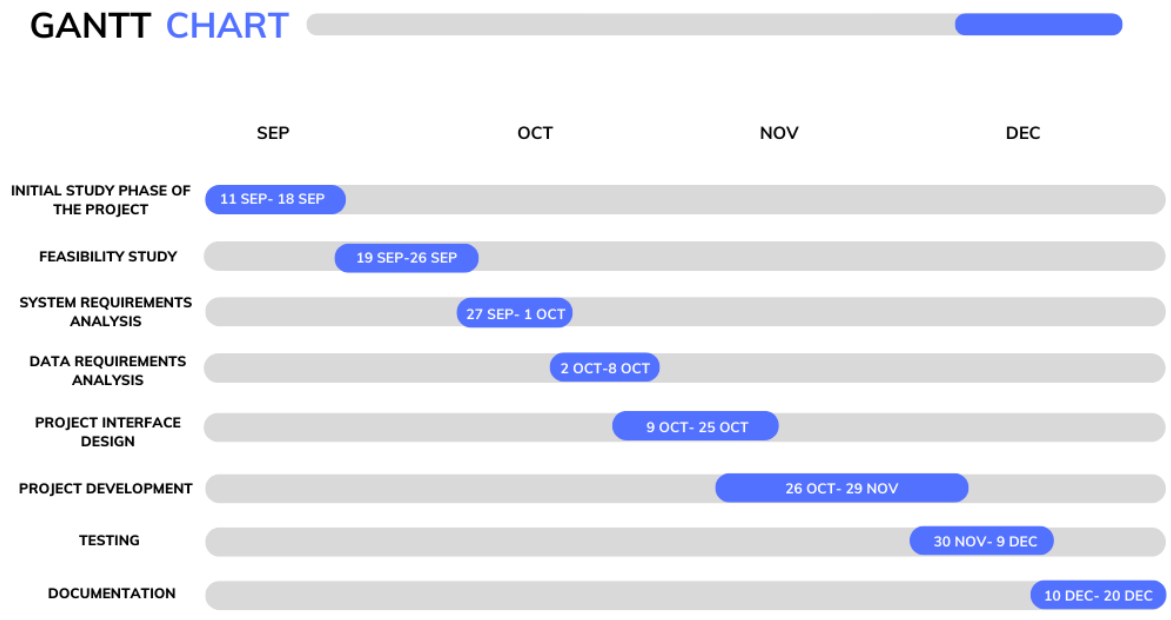


Figure 1: Project schedule in Gantt chart

2.2.8 Deployment

The recommended solution will be put into action during this phase. For the proposed approach, a mobile application will be made which will be accessible from both android and IOS mobile devices. For this reason, the flutter framework will be used to do so. The back end will also be developed using the Yii framework. MySQL has been chosen for implementing the database.

2.2.9 Conclusion

From the explanation above, it is clear that the "Lion Cinema" will benefit the public greatly and will offer customers a 24/7 service for reserving a seat in the theater and gathering online movie information. The consumer may simply learn about the movies that have been released and then make a decision. Hopefully, this approach will have a significant impact on the entertainment purpose for the people.

2.3 Problem Areas

Through the old traditional way, the consumer needs to go to the specific theater where the desired movie is screening, stand in line, and purchase the ticket for the movie via credit/debit card or cash, which will make it more difficult for a busy person to solve this issue. Because not everyone has this much free time to do so. The process is very complicated, and customers encounter lots of difficulties.

Another thing is that technology is improving day by day and so are all countries. Our neighboring country India is more technologically advanced than us. They have a bunch of movie ticket booking apps. But we have barely one in our country which is definitely a big drawback for us. (*34 Pros & Cons Of Starting A Movie Ticket Booking App Business (2023)*, n.d.) Even if we have one there remain optimization issues and server issues. Furthermore, the app does not have a refund policy or any guidelines for it.

2.4 Possible Solutions

Since theaters have been primarily used to sell tickets to audiences for a certain movie or specific performance. Therefore, this solution needs to save the customer's time and energy because traditionally, theaters are busy, wasting the time of the average person. The solution must have no server issues, especially at the time of payment, it needs to have a refund policy or instruction for it, and customers should be able to view their purchased tickets

CHAPTER 3: LITERATURE REVIEW

An assessment of published sources or academic sources (such as books, journal articles, and theses) that are relevant to a given subject or research question is known as a literature review. It is frequently included in a thesis, dissertation, or research paper to contextualize your work in light of the body of knowledge. It is meant to assist audiences to understand the current research and discussions that are relevant to a particular subject or field of study and to communicate that understanding in the form of a written report. This section contributes to creating a comparison between studies in the same field of study and offers the potential for a conversation about the research, which lays the path for a fruitful initiative.

3.1 Discussion on Problem Domain

The problem of a decent movie ticket booking app remains in our country. Although some well-known companies have online ticket booking still there is no user-friendly and optimized movie ticket booking app in our country. Star Cineplex which is one of the highest-grossing cineplexes in our country, though they have an app for booking tickets it lacks a lot. Launched in February 2011, BEFTN was the country's first paperless electronic interbank fund transfer system. (*Bangladesh Electronic Fund Transfer Network*, n.d.) It facilitates both credit and debit transactions, as a lead-over check-clearing system. After that, the system started becoming more efficient day after day. But still, we don't have a complete cinema ticket booking app where payment can be done via mobile phone without server issues. Some of the common problems are:

- Server issue
- The payment channel is not well developed for the customers
- Instruction of refund policy
- No proper terms and conditions

3.2 Discussion on Problem Solution

Nowadays, as it is the age of technology and modernization, different companies from different nations have already started developing mobile-based or web-based solutions to serve their interest. Everything from the bed to the office has started to be digitized. The entertainment sector is not out of there. Every day, technology is changing this industry. The world is becoming more digital and data-driven.

The solution should be taken into account considering these topics:

- **Security:** Since the application must store sensitive data, including personal data, data security should be the main concern for the solution. The app needs to have advanced security measures.
- **Refund Policy:** The company has sole authority over its return and refund policies thus it is not required by law to accept returns or refunds**.** But it is a necessary action to gain customers' trust.
- **Trust:** Trustworthiness is their biggest issue. They have trust issues because they are often exposed to such deception. One of the main reasons why users feel that shopping through online platforms is unsafe. That's why users should have complete access to the app.
- **Server Availability:** It is important to take into account that the application is always accessible from everywhere in the nation. Since there is no substitute, the application must be accessible 24/7.
- **Accurate User Identification:** The user purchasing the ticket must have valid identification and authentication. There are various methods of identification like email verification or mobile number verification which can easily prevent fraud in this area.

3.3 Comparison Among Popular Movie Ticket Booking Apps

As days pass, technology also makes daily advancements. As a result, comparable types of products, websites, and applications have been developed to accomplish the same task in various ways. The main goal of inventing is to figure out how to make anything more user-friendly, time-consuming, inexpensive, etc. Each application has distinct qualities that make it more reliable and distinguishable from others.

Here, I'll compare the top three widely used apps for purchasing cinema tickets and discuss their benefits and drawbacks. These are:

- **BookMyShow (India)**
- **Start Cineplex (Bangladesh)**
- **PVR Cinemas (India)**

BookMyShow

One of the most widely used apps for purchasing online movie tickets is BookMyShow. India's top entertainment website, BookMyShow, has more than 60 million users who have downloaded the app as of this writing(*5 Best Apps For Booking Movie Tickets Online in India*, n.d.). You can reserve tickets using this app for a variety of events taking place in your location, including sporting events, concerts, and movies. Through this app, you can also check movie times, movie reviews, and upcoming movies.

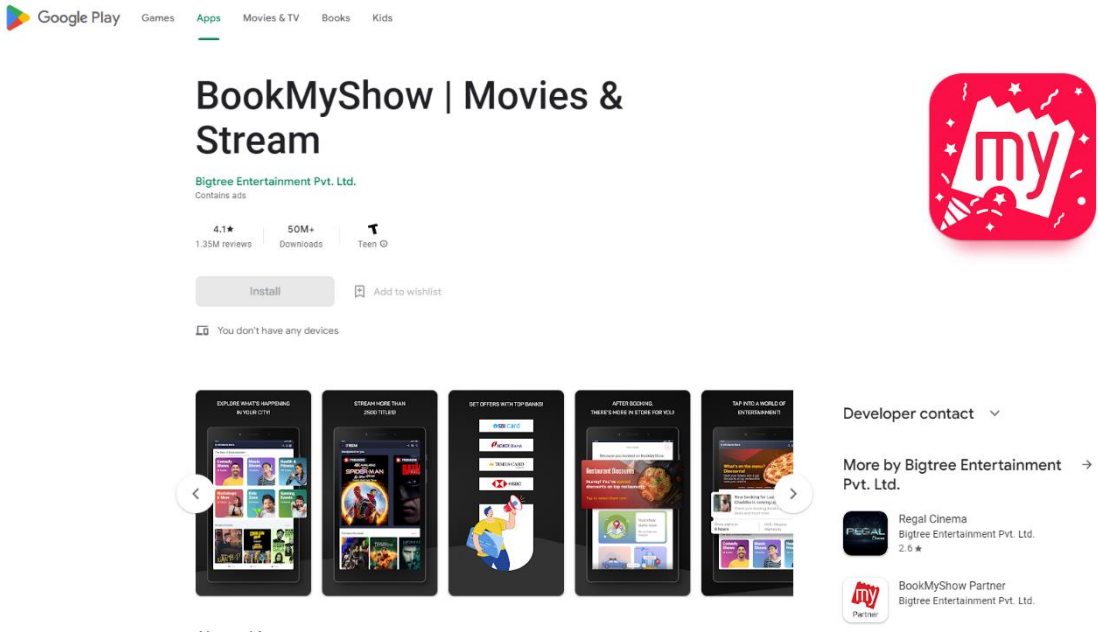


Figure 2: Overview of the BookMyShow app

Pros of BookMyShow app

- Purchase tickets for movies, plays, sports, concerts, and more with a single app.
- Excellent deals and cashback
- Buy tickets for any theater with a single app (PVR, Cinepolis, Inox, and Many More)
- Payment alternatives
- Optional reserve seat booking

Cons of BookMyShow

- No Support for Cancellation
- higher than average Internet handling fees

Star Cineplex

Star Cineplex is the main and first multiplex cinema in Bangladesh. Show Motion Limited merged on the nineteenth of December 2002 with the Star Cineplex brand(*Cineplex Web*, n.d.). Since then, it has led the progressive multiplex cinema sector here. The company started its journey with an adaptable, well-built application that is running smoothly till 2020(*Cineplex Web*, n.d.).

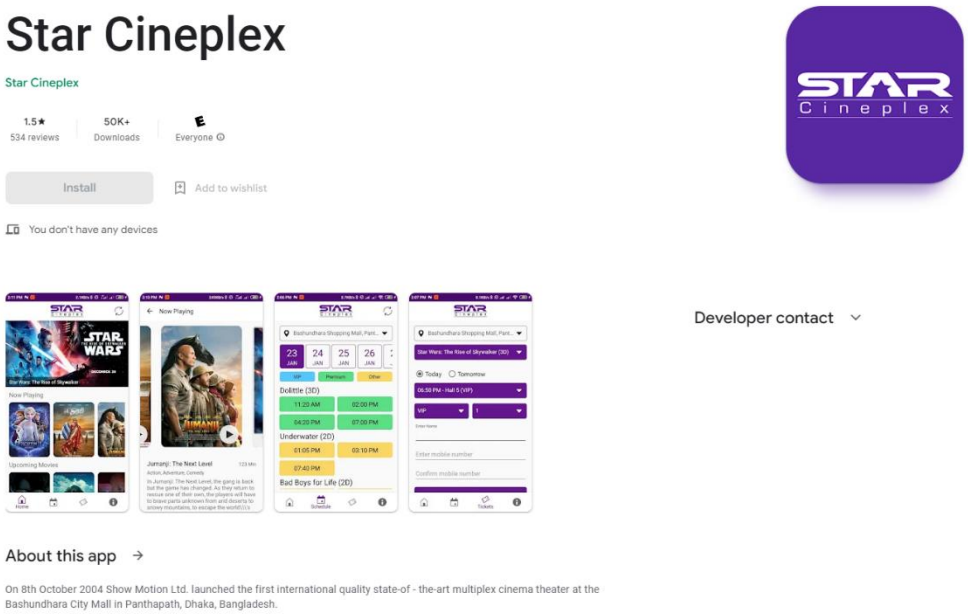


Figure 3: Overview of Star Cineplex app

Pros of Star Cineplex app

- Simple to use
- Partnerships with most of the Cinema Groups
- Light-weighted application

Cons of Star Cineplex app

- Often server goes down
- Fewer deals and offers
- Not public friendly at all
- No refund policy

PVR Cinemas

One of the most well-known movie theater chains in India is PVR. Any PVR Cinema is simple to locate in your location, and the movie-going experience there is comparable to that of the majority of the country's upscale theaters. (*5 Best Apps For Booking Movie Tickets Online in India*, n.d.) The PVR Group is the official provider of the PVR Cinema Movie Ticket app. Through this app, you may reserve movie tickets for any PVR Cinema.

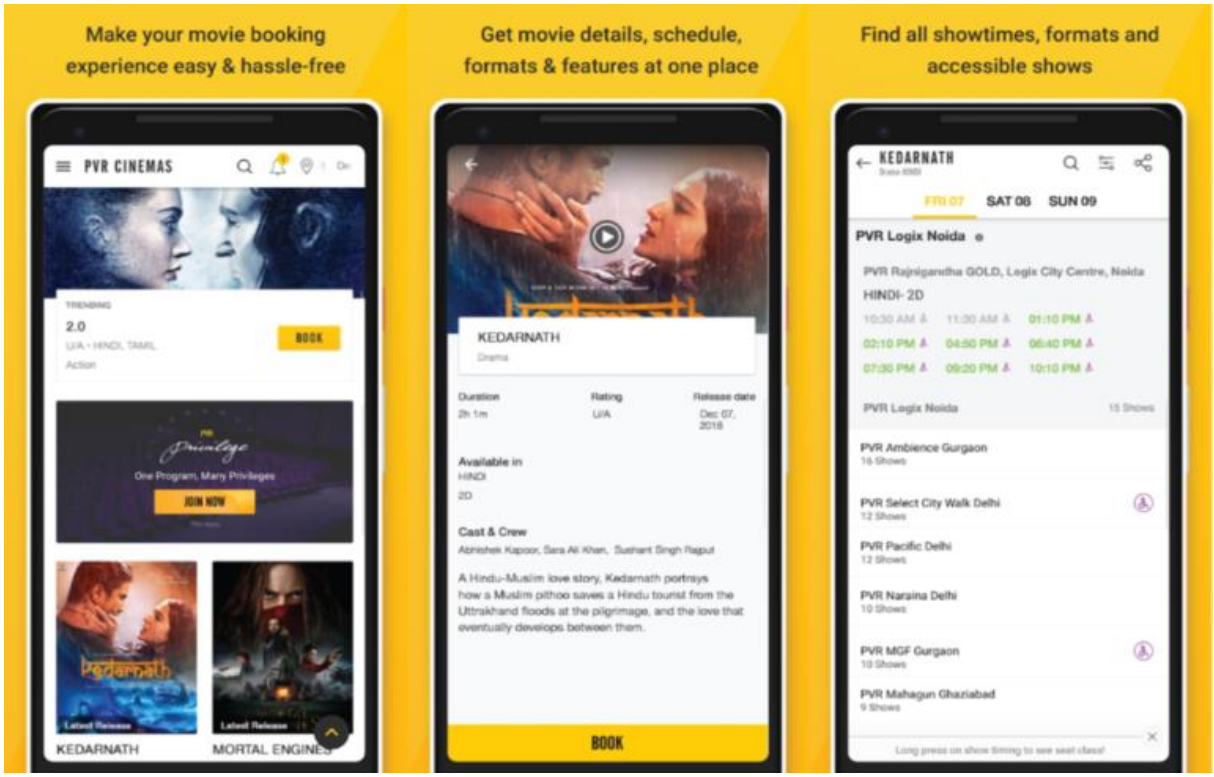


Figure 4: Overview of PVR Cinemas app

Pros of PVR Cinemas app

- Updates on current traffic
- Rent a car for your cinema excursion using the same app
- Purchase both tickets to a movie and food.
- No cancellation fees for tickets (on canceling before 20 Mins of show time)
- Options for payment

Cons of PVR Cinemas app

- Fewer cashback and offers
- Fewer payment options

3.4 Recommended Approach

After reviewing those movie ticket booking apps, it became obvious what kind of features the suggested system should include and how it should communicate with users. The suggested system should incorporate the following feature:

- Well-maintained server
- A variety of payment options
- User-friendly and eye-catching interface design
- Phone number verification
- Category of currently showing and upcoming movies
- Refund policy
- View purchased tickets option

CHAPTER 4: METHODOLOGY

The steps to be followed or procedures used to find, gather, process, and analyze information about a topic is known as methodology. Each methodology has its pros and cons, and each one ends for a different reason. The approach that is used for development objectives commonly determines the success rate of the project. This chapter will describe the reason for selecting an appropriate methodology.

4.1 What to Use

A development process is organized, planned, and managed using a software development methodology. Whether a developer selects waterfall, iterative, agile, or other methodology, how strictly they follow the SDM can determine whether a project and/or company succeeds or fails. In this section, three methodologies will be analyzed along with their merits and demerits.

Waterfall Model

The waterfall model is a software development model that assumes development projects can be completed in a series of stages(*SDLC - Waterfall Model*, n.d.). Each stage has its own set of requirements, and the project moves through these steps until it is finally complete. As such, each phase must proceed in order from top to bottom.

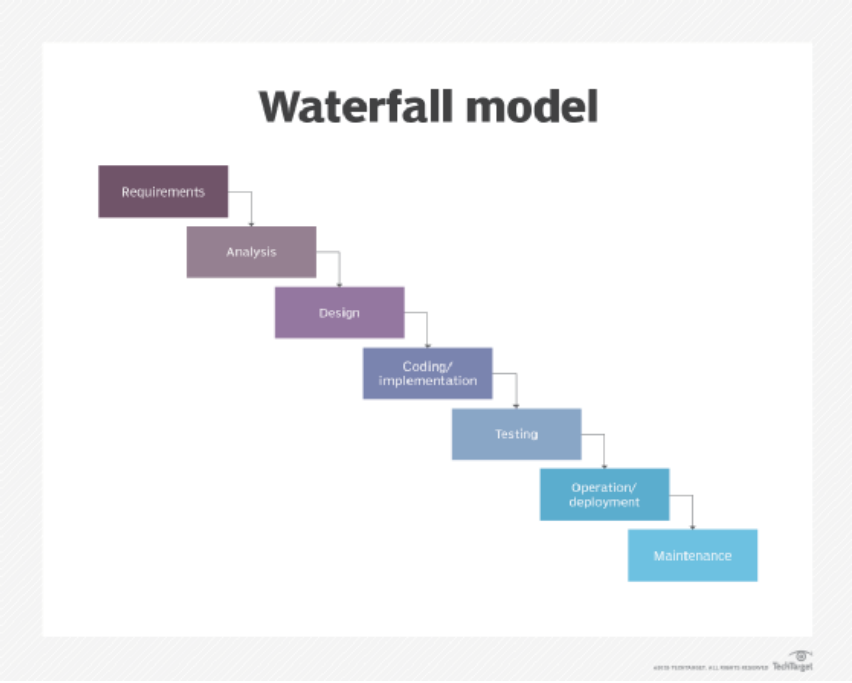


Figure 5: Waterfall model

Merits of the Waterfall model:

- It provides transparency and traceability so that it can be audited easily.
- The model's adaptability makes it simple to manage. At each stage, there are specific outputs and a review procedure.
- It begins with a well-defined base activity and processes it through several stages until it becomes a deliverable product or system.
- Works effectively for smaller projects with clearly defined criteria.

Demerits of the Waterfall model:

- It takes a lot of time for a project to start and end, as it involves several steps.
- The model also has a high risk of delaying the completion date.
- Not appropriate for use as a model in tangled and object-oriented designs.
- Lack of integration with agile software development

Dynamic System Development Model (DSDM)

The DSDM framework is meant to be more than just a way to build software development packages gradually. It is a long-term strategy that has applications outside of software development projects. (*Dynamic Systems Development Method (DSDM) - GeeksforGeeks*, n.d.) It contains the necessary instructions for bringing a product through the complete project, including the releases.

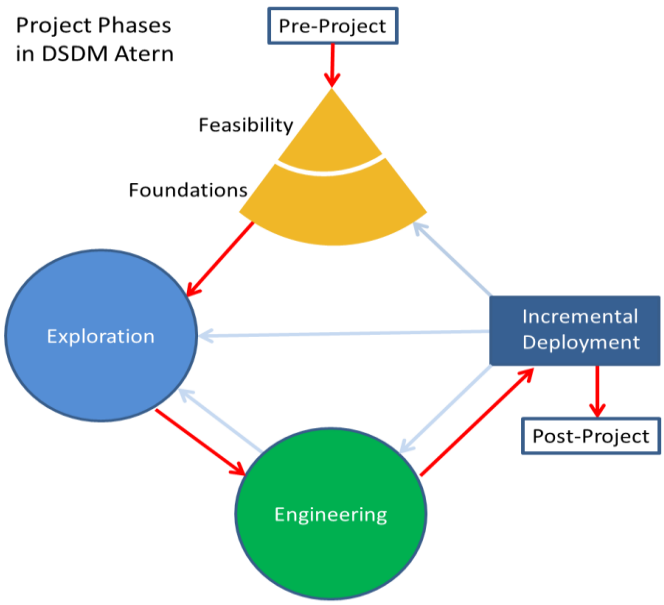


Figure 6: DSDM model

Merits of the DSDM model:

- high user participation.
- Basic functionalities are given more quickly and frequently.
- Projects are completed on schedule and within budget.
- gives developers access to end users.
- Improvisations can be made easily

Demerits of DSDM model:

- Not appropriate for startups or one-off initiatives.
- It is less well-known and simpler to understand because it is a more recent model compared to more established ones like the waterfall.
- Compared to other agile software development methodologies, DSDM can be restrictive and challenging to work with due to its strictness and eight principles.

Prototype Model

Prototyping is a technique used in the design process to test an idea, system, or service in a low-fidelity or interactive prototype. Prototypes are usually built quickly, usually with only a few people involved. Those allow us to test ideas and evaluate how they might function in practice before investing time and resources into developing full-scale applications (*Software Engineering | Prototyping Model - GeeksforGeeks*, n.d.)

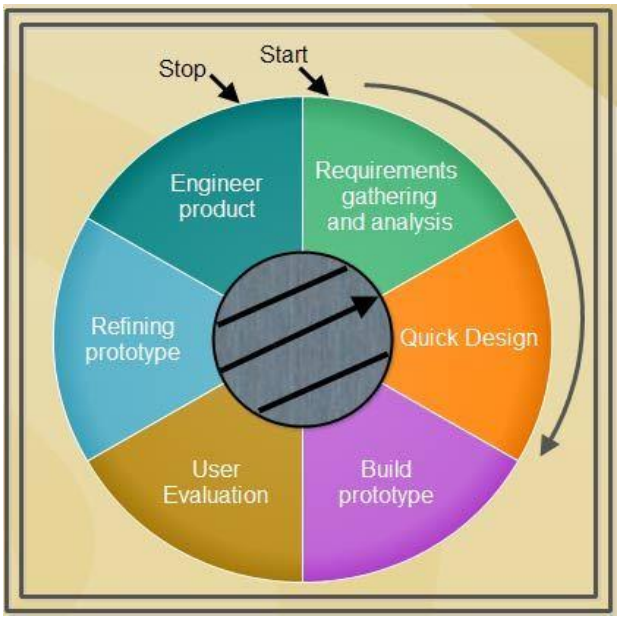


Figure 7: Prototype model

Merits of Prototype model:

- The design of this model is customizable.
- It is simple to spot faults.
- Missing functionality is simple to find.
- Because there is space for improvement, it is possible to readily meet new requirements.
- It provides higher levels of comfort and user satisfaction.
- It works well with an online system.

Demerits of Prototype model:

- This model is expensive.
- Because customer expectations are always changing, the documentation may be unsuitable sometimes.
- There can be too many different criteria.

4.1.1 Chosen Method

By considering the requirements I have decided to use **DSDM Method** for the **Lion Cinema** application. It will be the best option for the app as it is simple to spot bugs and errors. Missing functionality is simple to find. Additionally, it facilitates changes by iterative development, with regular evaluations by users to ensure that what is being developed is what the company needs.

4.2 Why to Use

The undertaking should follow a methodology that meets the requirements for the predefined time and financial plan. A methodology that allows changes if it is required. A methodology that places a focus on cooperative teamwork and flexible design methods.(Sheikh Parvez Mahamud, 2019) Finding the perfect method might be very difficult, but doing so can greatly assist in completing the project and producing the desired output.

4.3 Segments of Methodology

Pre-project: This segment works with the project's timeline, budget, and initial requirements. This phase also considers related issues and potential solutions.

Feasibility study: The project's operational, technological, and economic feasibility is evaluated in this segment.

Requirements gathering: Gathering the functional and non-functional requirements is the objective of this phase. There are a few suggested systems, like questionnaires, to obtain needs. group discussion, interview, etc.

Requirements analysis and prioritization: Prioritizing is required after gathering the necessary information. Some criteria need to be developed initially because they are of high priority. Some requirements may come from several sources, and those requirements need to be sorted out and analyzed.

Exploration and engineering: The requirements are analyzed and examined at the exploration stage. The engineering stage is then notified to continue development. If the criteria need to be changed, the process starts over at the exploration segment.

4.4 Implementation Plans

The final phase of the project, during which the finished application is made available for use After identifying and resolving any issues related to this to make the project viable. Delivery methods, arrangements, and plans are chosen in this section. After that, if everything goes according to plan, the project is established.

CHAPTER 5: PLANNING

5.1 Project Plan

The project is planned out completely in this chapter so that it can be easily finished. The project is broken up into a few manageable portions, and those are completed within the specified amount of time. When a particular task will begin and be complete, In the end, all of those decisions are made in this stage.

5.1.1 Work Breakdown Structure

SL	Task Title	Start Date	End Date	Duration
1	Introduction	13/09/22	18/09/22	5 days
2	Initial Study	19/09/22	25/09/22	6 days
3	Literature Review	26/09/22	03/10/22	7 days
4	Methodology	04/10/22	09/10/22	5 days
5	Planning	10/10/22	14/10/22	4 days
6	Feasibility	15/10/22	19/10/22	4 days
7	Foundation	20/10/22	25/10/22	5 days
8	Exploration	26/10/22	29/10/22	3 days
9	Engineering	30/10/22	05/11/22	6 days
10	Deployment	06/11/22	24/11/22	18 days
11	Testing	25/11/22	01/12/22	6 days
12	Implementation	02/12/22	05/12/22	3 days
13	Critical Evaluation	06/12/22	09/12/22	2 days
14	Lessons Learned	10/12/22	11/12/22	1 days
15	Conclusion	11/12/22	12/12/22	1 days

Figure 8: Work breakdown structure for Lion Cinema

5.1.2 Resource Allocation

Resource allocation refers to the process of organizing and distributing all relevant assets and materials so that the proposed project can be carried out according to schedule. It is a crucial project planning task. The task must be accomplished with teamwork. The fundamental goal of resource allocation is to keep the work schedule in good shape so that the objectives and deadlines can be met.

The following resource allocation is being used for the **Lion Cinema** project to meet the pre-scheduled project delivery deadline:

SL	Task Title	Duration	Resource Name
1	Introduction	5 days	Analyst
2	Initial Study	6 days	Analyst
3	Literature Review	7 days	Analyst
4	Methodology	5 days	Analyst, User
5	Planning	4 days	Analyst, Team Leader
6	Feasibility	4 days	Analyst, User
7	Foundation	5 days	Analyst, Designer, Developer
8	Exploration	3 days	Analyst, Developer
9	Engineering	6 days	Analyst, Developer
10	Deployment	18 days	Developer, Designer
11	Testing	6 days	Developer, Tester, User
12	Implementation	3 days	Analyst, Developer, User
13	Critical Evaluation	2 days	Analyst
14	Lessons Learned	1 days	Analyst, Developer, Tester
15	Conclusion	1 days	Analyst,

Table 7: List of resource allocation

5.1.3 Tim Boxing

Time boxing is the technique used in the Dynamic Systems Development Method (DSDM) to organize, plan and manage a project's time. Because DSDM has a top-down approach, it uses timeboxing early in the project life cycle. It requires separating the system into smaller pieces and sequences of events. There are six-time boxes for this project, which are completed with the resources allotted.

Time Box	Task Title	Resource Name
TB-1	Introduction	Analyst
	Initial Study	Analyst
	Literature Review	Analyst
	Methodology	Analyst, User
TB-2	Planning	Analyst, Team Leader
	Feasibility	Analyst, User
	Foundation	Analyst, Designer, Developer
TB-3	Exploration	Analyst, Developer
	Engineering	Analyst, Developer
TB-4	Deployment	Developer, Designer
	Testing	Developer, Tester, User
TB-5	Implementation	Analyst, Developer, User
TB-6	Critical Evaluation	Analyst
	Lessons Learned	Analyst, Developer, Tester
	Conclusion	Analyst,

Figure 9: Time box

5.1.4 Gantt Chart

Start Date	End Date
13/09/22	18/09/22
19/09/22	25/09/22
26/09/22	03/10/22
04/10/22	09/10/22
10/10/22	14/10/22
15/10/22	19/10/22
20/10/22	25/10/22
26/10/22	29/10/22
30/10/22	05/11/22
06/11/22	24/11/22
25/11/22	01/12/22
02/12/22	05/12/22
06/12/22	09/12/22
10/12/22	11/12/22
11/12/22	12/12/22

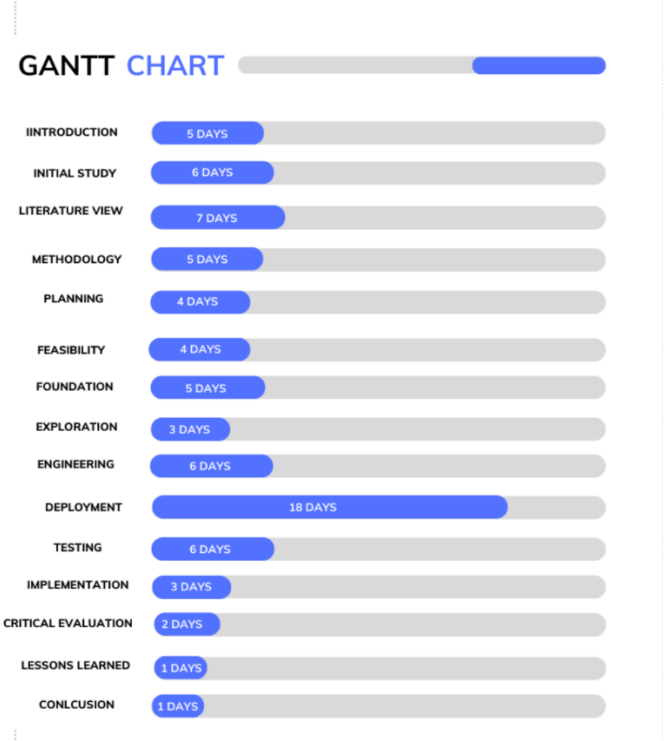


Figure 10: Gantt chart

5.2 Test Plan

Each time box has set tasks and times. So, after finishing each time box, those tasks must be tested. It helps to understand the results clearly. Below are the results of the testing against the time limits:

5.2.1 Testing Against the Time Boxes

Testing Against the Time Boxes is a style of requirement testing that ensures that the functionality of a software item is not restricted by the time box. It tests for correctness, functionality, and consistency with other requirements

Username	Example	Role	Example
Time Box ID			
Time Box Content			

Test Type	Test Steps	Expected Result	Actual Result	Comment
Unit test				
Integration test				
System test				
Acceptance test				
Security test				
Reliability test				
Usability test				

Figure 11: Testing example against time box

5.2.2 Required Tests

There are essentially two types of testing: one is referred to as functional testing, and the other is non-functional testing. A brief description of those is provided below:

Functional Testing:

Unit testing: It is the process of testing a component, class, or entire program unit separately from the rest of the system and without production code. Unit testing is a technique for ensuring code reliability, quality control, and high-quality software. Moreover, it focuses on individual units of code, so it can fail if the code is broken or contains errors. (*Unit Testing Tutorial – What Is, Types & Test Example*, n.d.)

Integration testing: Integration testing is a testing method used to verify the integrity of a program against multiple versions. The goal of integration testing is to ensure that the various pieces of your software communicate with each other, pass data back and forth, and behave consistently.

System testing: System testing is the process of testing a system at the system level and checking the operational integrity of these entities. This may include proving that all components are operating correctly, exchanging information, and even simulating an application's behavior to ensure it works as expected.

Acceptance testing: Acceptance testing is a phase of software testing that tests applications under conditions intended to match what your real users will experience. Acceptance testing focuses on determining whether the product is fit for use by prospective end users, rather than on correcting technical defects in the application.

Non-Functional Testing:

Security testing: It is the process of checking systems for product vulnerabilities, weak configuration, and other possible weaknesses. Security testing is an important part of software development and application security.

Usability testing: It is a process that involves observing a person as they use the application, to determine whether the interface or application is easy to use and understand, while also measuring how well the material performs.

Reliability testing: It refers to the ability of a component or system to perform its intended function over time. The reliability of a device may be assessed by either testing or monitoring, although testing is more common in practice.

5.2.3 Test Case

The test case is a description of the expected behavior of a test to verify that it accurately reflects what the intended user experience will be. Types of the tests, description of the tests, expected results, and the actual results are contained in the test case.

Test Case Name			
Test Type			
Test Description			
Test Steps	Expected Result	Actual Result	Comment

Table 8: Test Case sample

5.2.4 User Acceptance Test Plan

The user acceptance test is a test plan where the intended users are asked to evaluate the product or service after its use. This helps in identifying issues with any feature or function of the product. It is the final stage of testing

Test Case No			
Test Type			
Test Title			
Precondition of Testing			
User's Name			
Act As			
Test Steps	Expected Result	Actual Result	Comment

Table 9: An illustration of the test plan for user acceptance

5.3 Risk Management

Risk management is a process that identifies and monitors potential threats, such as financial loss or loss of business, and then takes steps to manage those risks. It involves identifying, assessing, and preventing the risk associated with the organization. (*Software Engineering | Risk Management - GeeksforGeeks*, n.d.) Effective risk management involves having a solid understanding of how risks impact an organization's critical business processes and decisions, as well as its broader environment. There are several steps of risk management such as:

- Risk identification
- Risk assessment
- Risk avoidance
- Risk reduction

5.3.1 Risk Identification

Risk identification is the process of identifying risks to be protected against in an application, which can be done through the use of risk criteria. It is the initial stage of risk management of a project. Several types of risks may occur at the time of developing the project such as business risk, methodological risk, technical risk, internal risk, and external risk. When identifying is completed the reason for the occurrence of each risk needs to be discovered. Certain steps need to be followed:

- Find out the significant application harm that has a huge chance to occur.
- Keep accurate records of the risks and root causes of this.
- List the risks that are likely to occur regularly when developing the application.

5.3.2 Risk Assessment

Risk assessment is a process of identifying and analyzing risks, considering the probability of those risks. It involves looking at all the critical factors surrounding the application, such as technology, people, process, and materials. For this project, I have identified some factors that are needed for risk assessment such as risk title, risk specification, likelihood, impact, and frequency.

The risk title will provide a concise description of the risk pattern. The likelihood will probability for the occurrence of the risk. The impact will represent the level of the risks and frequency will show the possibility of occurrence according to per year

Risk Title	Risk Specification	Likelihood	Impact	Frequency
Business risk	No need other development tool or existing system.	Unexpected	High	0-1 year
	Business requirements not matching	Expected	High	0-1 year
Methodological risk	Choosing an unnecessary methodology	Expected	Medium	0-1 year
Technical risk	Frequent change of requirements during development	Expected	High	1-2 years
	Data Inconsistency	Expected	Medium	1-2 years
	Server Issues	Expected	Medium	1-2 years
Internal risk	Unauthorized access	Rare	High	1-2 years
External risk	Refund policy changes	Rare	Medium	0-1 years

Table 10: A representation of risk assessment

5.3.3 Risk Avoidance Plan

After identifying possible risks some necessary actions are needed to avoid those risks. Those necessary actions are:

Risk Title	Risk Specification	Likelihood	Impact	Necessary action
Business risk	Business requirements not matching	Expected	High	Cooperative work with clients, suppliers and other business partners
Methodological risk	Choosing an unnecessary methodology	Expected	Medium	Use reliable data and analyzing properly
Technical risk	Frequent change of requirements during development	Expected	High	Implementing requirements properly and continuous monitoring and review procedures
	Data Inconsistency	Expected	Medium	
	Server Issues	Expected	Medium	
Internal risk	Unauthorized access	Rare	High	Proper planning and management.
External risk	Changes in terms and policy of app store/play store	Rare	Medium	Frequently reviewing terms and policy of app store/play store

Table 11: A representation of risk avoidance

5.3.4 Taken Actions for Possible Risk

Risk Title	Risk Specification	Likelihood	Impact	Actions taken
Business risk	Business requirements not matching	Expected	High	Analyzed the requirements properly and cooperated clients, suppliers.
Methodological risk	Choosing an unnecessary methodology	Expected	Medium	Following the methodological approach, all other decisions, including time scheduling and scope, will be made. Custom methodology will be used if the correct approach does not match the project.
Technical risk	Frequent change of requirements during development	Expected	High	Implemented requirements properly and continuous monitoring and review procedures has taken into action
	Data Inconsistency	Expected	Medium	
	Server Issues	Expected	Medium	
Internal risk	Unauthorized access	Rare	High	Proper authentication system has been implemented
External risk	Changes in terms and policy of app store/play store	Rare	Medium	Terms and policy of app store/play store is being regularly checked

Table 12: Sample of taken actions for avoiding risk

5.4 Change Management

5.4.1 Factors That Might Cause Change

Change management is the process of guiding your app development project through various stages of development, to completion. Several changes have been mentioned by the company owner. These are:

- Changes in application layout design
- Changes in functional requirements
- Changes in the payment module
- Scheduling phase changes

5.4.2 DSDM Atern Welcomes Change

During the development phase of a project, there may often be changes required by the organization. Therefore, a method that allows us to modify the scheme needs to be used in such projects. DSDM is a type of methodology that allows us to make changes as needed during the development phase. The way it works:

- It operates following the conclusion of modifications.
- It guarantees the security objectives.
- It guarantees scalability and reliability.
- When a module changes, it requires an assessment from several users in real-time.

5.4.3 Considering Business Priority

System modifications are followed by priority in the DSDM. Modifications that are necessary to be made are performed by using MoSCoW prioritization. At first, the most crucial requirements are taken into action for modification. After that, less important requirements. Eventually, the modifications also depend on the time-box, to check whether or not it is compatible with time. The less important requirements remain untouched.

5.4.4 Change Workshop

DSDM Change workshop provides participants with a detailed understanding of their challenge so that users and developers can help them with it and make their communication strong. This will help in achieving goals for the organization. Developers can organize an interview session or questionnaire by taking authorities' permission and support to get feedback from users directly and can analyze the changes that could be required.

5.4.5 Changes That are Allowed

Not all changes could be made but a resource, budget, time, quality, and risk are some common criteria that are open to change. Before making changes, the criteria are assessed from the business perspective.

5.4.6 Key Decision Takers to Make Change

In this project, there are several decision takers to allow changes. They are:

- Analyst
- Developer
- Tester

If necessary, the analyst can change the functional requirements, and the developer can decide whether to modify the relevant code.

5.5 Quality Management

Quality management is the process of monitoring, measuring, and improving products or services to ensure they meet user requirements. Quality management has three primary objectives, which are:

- To establish quality requirements and complete quality assurance planning
- To design and implement the organization's quality system to achieve its objectives
- To guide for implementing effective measures for meeting user needs.

5.5.1 Rules Applied to Maintain Quality

It is pretty common for features and requirements to vary from one system to another. Two guidelines have been followed to preserve the quality:

Quality Control: It is a process that controls the quality of products and services by making sure that they are produced in conforming with the quality standards expected by the user(*Quality Assurance vs Quality Control: Definitions & Differences* | *ASQ*, n.d.).

In this project, it has been maintained by taking a review of whether or not the users can make payments through the application.

Quality Assurance: It is a process used to assess, analyze and improve the quality of a project.

In this project, it has been maintained by evaluating whether or not the mobile verification works.

5.5.2 DSDM Atern Standard Quality Measures

The DSDM produces a set of standard quality measures that quantify the performance of an organization. It maintains two different rules:

Solution quality: It is the highest level of performance and functionality that meets customer specifications. It involves users understanding.

Two prioritization techniques have been used to maintain solution quality in this project. Those are **Time Boxing** and **MoScoW**.

Process quality: It is the quality of having a defined set of goals, processes, and plans to consistently produce high-quality products or services. Processes play an important role in measuring product or service quality.

In this project, **the DSDM** approach has been used to process solution quality

5.5.3 Quality Plan and Measuring Meter

A quality plan is a way to ensure that an organization meets all standards, policies, and processes of the destination. Several plans have been made such as:

- To achieve the goal appropriately.
- Resources and responsibilities are equally distributed.
- Appropriate testing at every phase.
- If any changes have been made, relevant documentation and guidelines have been recorded.

CHAPTER 6: FEASIBILITY

6.1 Possible Types of Feasibility for This Project

Operational Feasibility: It focuses on the overall usability of the application and how well it operates. The **Lion Cinema** application has a proper verification system to take user credentials, an optimized and smooth layout for different modules, a simple ticket reserving option that makes the app pretty user-friendly for ticket booking, and various online payment options for purchasing tickets. Admin can maintain the whole system from the backend by simply logging in with valid inputs.

Technical Feasibility: There are some crucial criteria of technical feasibility that have been maintained in this project.

The app has been designed and developed in android and IOS based approach using **Flutter**. The app also has a back-end and admin panel. The app is compatible to run in any android and IOS device with a low configuration because it has been created and developed utilizing the most recent and well-liked mobile technologies. Platform independence, the cost-effectiveness of the development platform, and resources have been used that makes the application technically feasible.

In this project technical aspects are:

Hardware:

- Desktop PC (core i7 10^{th} gen, 2.90 GHz, 8 GB DDR4 RAM, 256 SSD)
- Wifi Router

Software:

- Visual Studio Code
- Android Studio (for emulator only)
- Vysor
- Postman

Operating System:

- Windows 11

Economic feasibility: According to the stage in which they occur, we divided the costs of the project. We already know that system development expenditures are often one-time expenses that vanish after the project is over. We looked at specific cost categories to determine the development costs.

- **Mobile Application Cost:**

Equipment	Cost per unit	Cost
Desktop pc (core i7 10th gen, 2.90 GHz / AMD 5 5600 G processor, 8 GB DDR4 RAM, 256 SSD)	৳ 65,000	৳ 65,000
Web, email, and file servers	৳ 5,000	৳ 5,000
	Total	৳ 70,000

Every device just needs to install the app to get services on the other hand it's quite simple and less costly.

- **Web Application Cost:**

Equipment	Cost per unit	Cost
Desktop pc (core i7 10th gen, 2.90 GHz / AMD 5 5600 G processor, 8 GB DDR4 RAM, 256 SSD)	৳ 65,000	৳ 65,000
Web, email, and file servers	৳ 5,000	৳ 5,000
Extranet network with VPN	৳ 2,000	৳ 2,000
	Total	৳ 72,000

In this case, there is no requirement for installation because the complete application and data are stored on a server and are accessible at any time via the internet by using a browser. But it is a bit expensive.

- **Desktop Application Cost:**

Equipment	Cost per unit	Cost
Desktop pc (core i7 10th gen, 2.90 GHz / AMD 5 5600 G processor, 8 GB DDR4 RAM, 256 SSD)	৳ 65,000	৳ 65,000
Web, email, and file servers	৳ 5,000	৳ 5,000
	Total	৳ 70,000

Here, every desktop computer must have the software installed, which makes it inflexible. Additionally, for storing data the app requires a remote server. However, it is even more costly.

As was previously noted, the organization preferred a mobile application-based project because it is more practical and less expensive. Therefore, the project has been developed as a mobile application with a back end and an admin panel following client requirements.

6.2 Cost Benefit Analysis

It is the process of analyzing the different costs and benefits of policies, projects, and social goals. The cost-benefit analysis of this project is given beneath:

Total Expense:

SL No	Sectors of Expenses	Year 1	Year 2	Year 3	Year 4	Year 5	Total
1	Mobile application cost	৳ 70,000	-	-	-	-	৳70,000
2	Web application cost	৳ 72,000	-	-	-	-	৳ 72,000
3	Domain & Hosting	৳ 1,850	৳ 1,850	৳ 1,850	৳ 1,850	৳ 1,850	৳ 9,250
4	Developers Salary	৳ 50,000	৳ 50,000	৳ 50,000	৳ 50,000	৳ 50,000	৳ 2,50,000
5	Other Costs	৳ 15,000	৳ 15,000	৳ 15,000	৳ 15,000	৳ 15,000	৳ 75,000
	Total	৳ 208,850	৳ 66,850	৳ 66,850	৳ 66,850	৳ 66,850	৳ 4,76,250

Table 13: Comprehensive analysis of the project's total expense

Total Income:

SL No	Sectors	Year 1	Year 2	Year 3	Year 4	Year 5	Total
1	Total Income	৳ 1,90,000	৳ 2,20,000	৳ 2,50,000	৳ 2,95,000	৳ 3,30,000	৳ 12,85,000
2	Total Expense	৳ 208,850	৳ 66,850	৳ 66,850	৳ 66,850	৳ 66,850	৳ 4,76,250
	Total revenue	৳ -18,850	৳ 153,150	৳ 183,150	৳ 228,150	৳ 263,150	৳ 808,750

Table 14: Comprehensive analysis of the project's total income

Total Revenue:

SL No	Sectors	Year 1	Year 2	Year 3	Year 4	Year 5	Total
1	Total Income	৳ 1,90,000	৳ 2,20,000	৳ 2,50,000	৳ 2,95,000	৳ 3,30,000	৳ 12,85,000
2	Total Expense	৳ 208,850	৳ 66,850	৳ 66,850	৳ 66,850	৳ 66,850	৳ 4,76,250
	Total revenue	৳ -18,850	৳ 153,150	৳ 183,150	৳ 228,150	৳ 263,150	৳ 808,750

Table 15: Comprehensive analysis of the project's total revenue

Therefore, it is obvious that the company will produce huge annual profits once this project is in place. Annual revenue growth will be increased on yearly basis. In the first five years, they will make approximately 808,750 taka in revenue. So, this application will be helpful for them.

6.3 DSDM - good or not for this project

The project needs to be developed iteratively because it may need to accept changes while being developed. DSDM ensures the iterative development process. Moreover, it simplifies mobile application development by providing an all-inclusive set of mobile service APIs. Its lightweight weight, rich compatibility with the native SDKs for iOS and Android, and straightforward usage model make DSDM ideal for this project.

CHAPTER 7: FOUNDATION

7.1 Problem Area Identification

Identification of problems is one of the most important steps because it will determine what the final product is to accomplish as well as it will improve performance. There are several ways to know problems from the user. Those are:

7.1.1 Interview

One of the best ways to learn about issues and other information from the users is through an interview. By conducting interviews, it is simple to learn about the needs of the user, the problems they encounter most frequently, and find out potential solutions. Therefore, for this project, the following user interview questions have been put in place:

- What kind of problems do they face while selecting movies and scheduling?
- What kind of payment methods do they prefer?
- What kind of movies do they prefer?
- Faced any problems while purchasing tickets?
- Faced any problems while viewing the purchased ticket list?

7.1.2 Observation

Another well-liked method for gathering information about user needs and issues is observation. The observation technique is used to visit the commercial sector and resolve issues so that users can be benefitted. The primary causes for the observation are listed below:

- The payment method of the application
- Check the mobile number verification method of the application
- Check the usability of the application
- Check the functionalities of the application

7.1.3 Questionnaires

Questionnaires are extremely beneficial to identify and define a problem and identify the areas of concern. It provides objective information about the current situation. Users are given a set of questions to which they need to respond. These questions have to be in MCQs or short question format. A set of questions is given below:

Questions for problem identification	
Name	Age:
General User	Gender:
Question-1	What kind of movies should exist in the application
Answer	
Question-2	What should be the payment and refund system?
Answer	
Question-3	What kind of movies should be in the upcoming list?
Answer	
Question-4	What kind of movies you prefer?
Answer	

Figure 12: Sample of questionnaires

7.2 Rich Picture

The rich picture portrays the system's overall business process. It helps in illustrating the conflicts, processes, and issues of the company that transform the entire company.

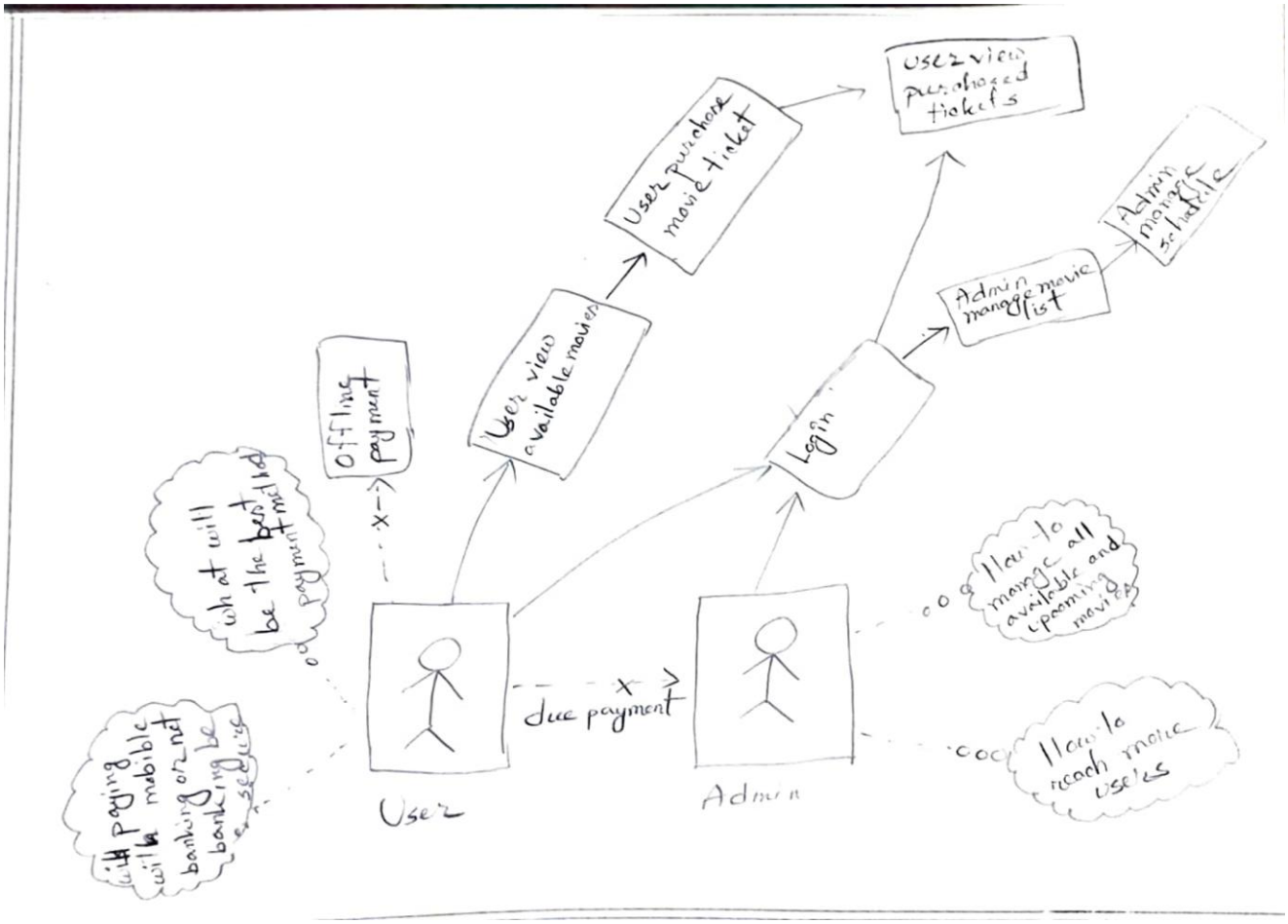


Figure 13: Rich picture

Elements of the rich picture

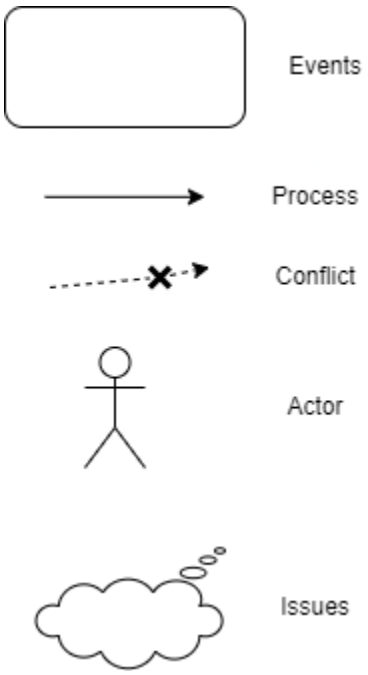


Figure 14: Elements from the rich picture

Key actor

There are two types of actors

- Admin
- User

Description of rich picture

The rich picture illustrates how an admin can keep track of every interaction and solve conflicts if it may arise. It also portrays how a user can purchase tickets, view purchased tickets, and view available movies. It also shows the issues of the admin as well as users. Moreover, it shows the conflict between the user and the admin.

7.3 Identification and Description of Specific Problem Area

Some significant issues have been identified via information-gathering techniques which need to be solved. By evaluating these some specific problems have been identified. Those are:

- No application for purchasing movie tickets available in the play store and app store.

- Lack of trust issues for purchasing movie tickets through an app
- Server issues have been a common problem
- No optimized application for purchasing movie tickets that shows correct information properly
- Payment issue
- No refund policy

7.4 Possible Solution

The problems that have been identified can be solved through this project. The solutions for those problems are given below:

- As there is no application available on the play store and app store for purchasing movie tickets through the **Lion Cinema** app this issue would be resolved.
- The optimized application should be developed
- A refund policy should be included
- A trusted application should be developed by providing them with authentic movie tickets and convincing them of secure transactions.

7.5 Overall Requirements List

Two types of requirements lists should be included. Functional and non-functional

Functional requirements:

- User login system
- Create admin panel
- Mobile number validation and verification
- Manage movie schedule and tickets
- Tickets generate
- Now showing and upcoming movie list

Non-Functional requirements:

- User-friendly interface
- User profile
- Refund policy
- Secure all data types appropriately

- Observe the effective color contrast

7.6 Technologies to be Implemented

Several innovative technologies can be used to implement the suggested framework. Selecting the best technology is crucial to develop a successful application and gaining desired output.

Client-server application

A client-server application is a kind of app that is used to connect with a server that provides information to the client(*Client-Server Application - OOSE*, n.d.). The main purpose of Client Server Architecture is to present a single interface to the user and access services offered by another computer or server. Some common features are:

- Installation required on a user's device
- Not being able to access too many users at once
- High expensive
- Powerful and quick to act

Web server application

Web server applications are used to handle requests and responses through a web server. Through a set domain name on the internet, the user can simply access the server(*Difference Between Web Server and Application Server - GeeksforGeeks*, n.d.). From anywhere in the world, they can connect to the system. No additional equipment needs to be installed. They utilize the internet and a browser to use the service. Some common features are:

- Installation is unnecessary.
- Accessed on the cloud via a web browser and the internet.
- Accessible at any time and from any location.

Cloud Integration

Cloud Integration makes it easy to use our services across your organization, even when you're in a different region. It allows you to add, remove and update sites on Cloud Sites and makes it easier for team members to collaborate when working on their projects(*Cloud-Based Integration - Wikipedia*, n.d.). Some common features are:

- Utilize the cloud's services to connect to it.
- Simple API scaling and up/down
- API refactoring
- Configuration takes a long period

7.7 Recommendation and Justification

The proposed solution will work well with a web server because its users are located across the nation. They must use several locations to access the system. The same kind of data is accessible to various users simultaneously. Therefore, client-server applications are not appropriate because this system is unable to support a big number of users. Therefore, it is recommended to utilize a web server for the application.

CHAPTER 8: FOUNDATION

8.1 Old System Use Case

Use Case diagram portrays the operation of a framework's tasks. I'll provide an outdated use case diagram here:

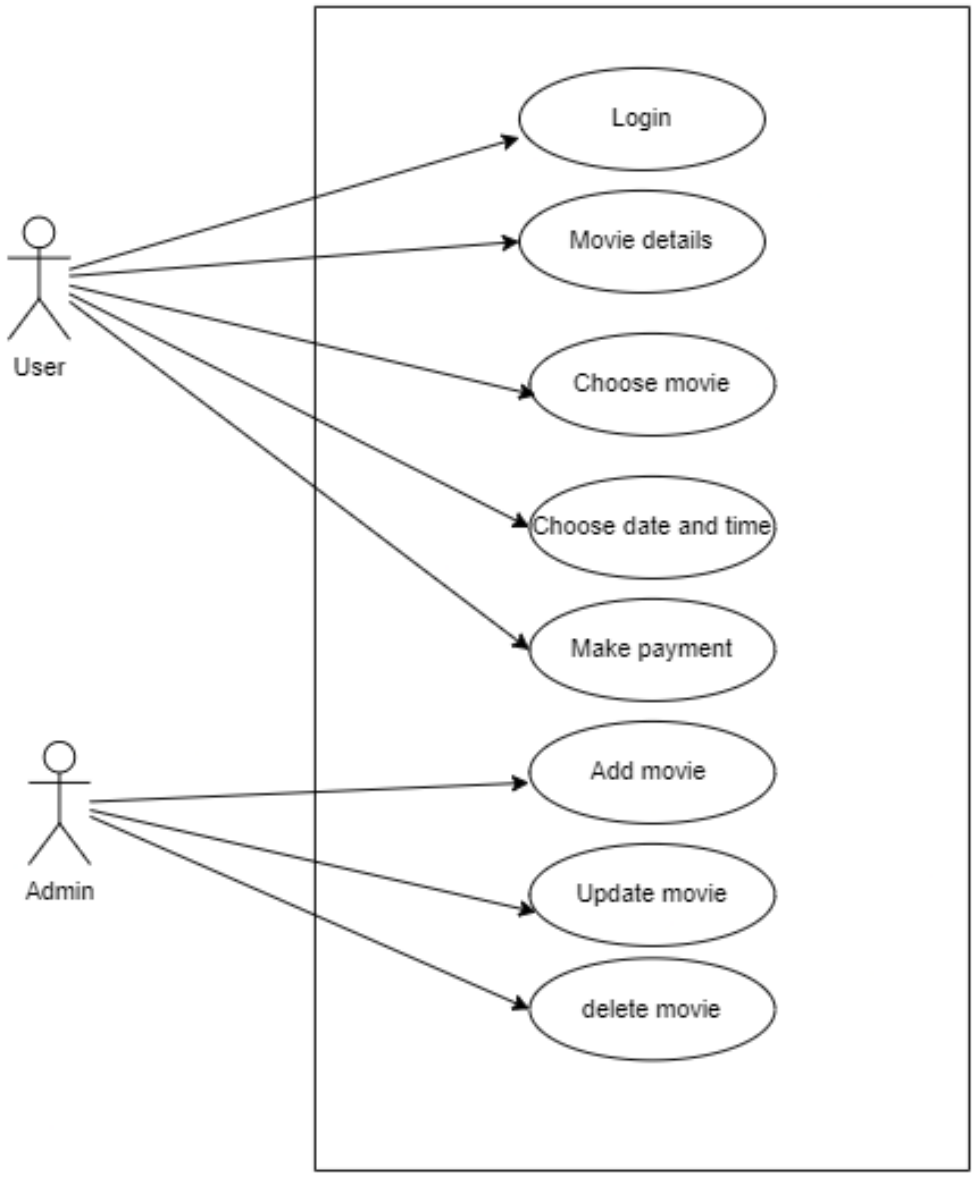


Figure 15: Old use case diagram

8.2 Full Application's Use Case Diagram

Figure 16: Full application's use case diagram

8.3 Full Application Activity Diagram

The Lion Cinema application has two types of actors. One is the admin and the other is the users. Both actor's activity diagrams are given beneath:

Admin's activity diagram:

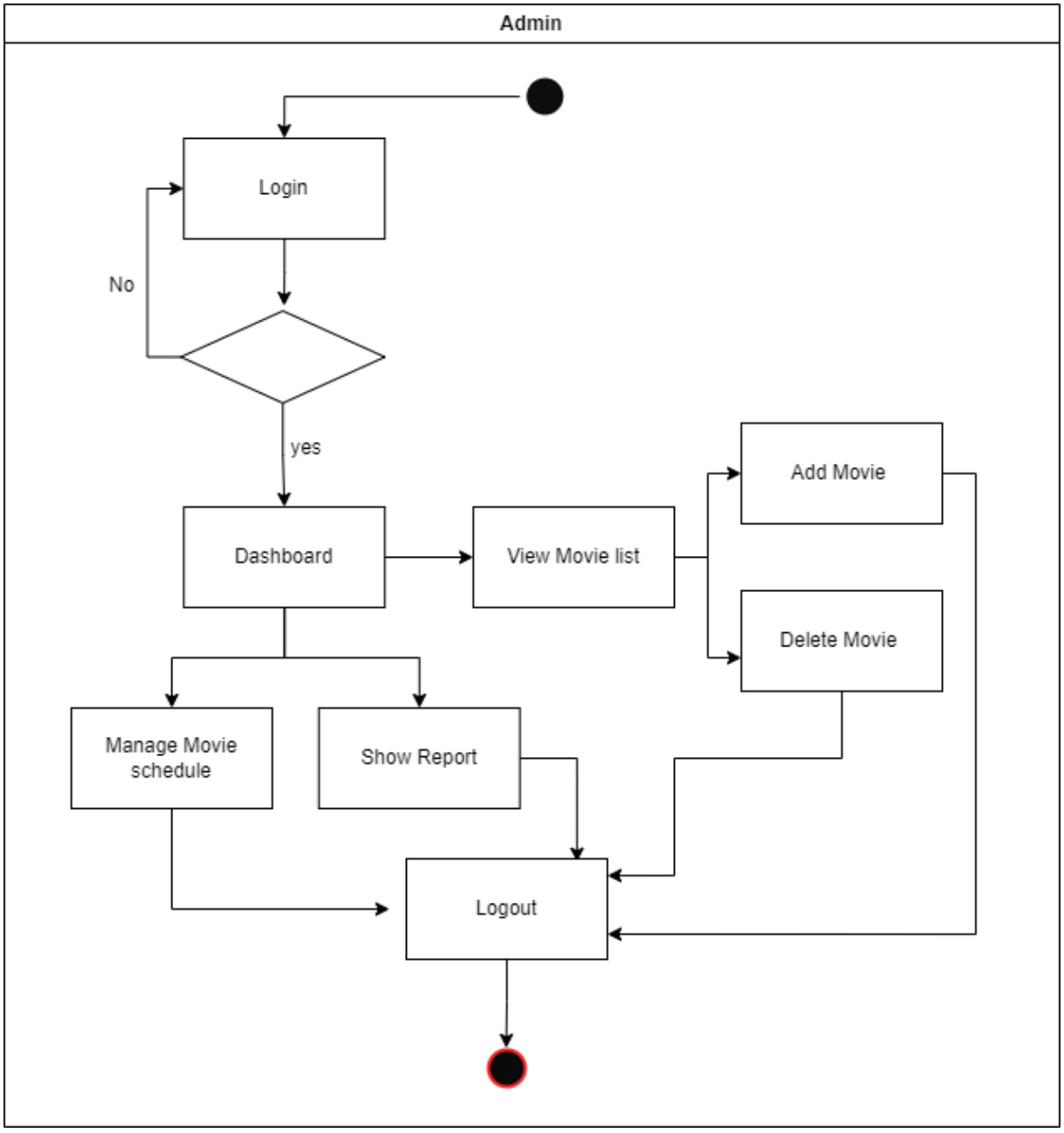


Figure 17: Admin's activity diagram

User's activity diagram:

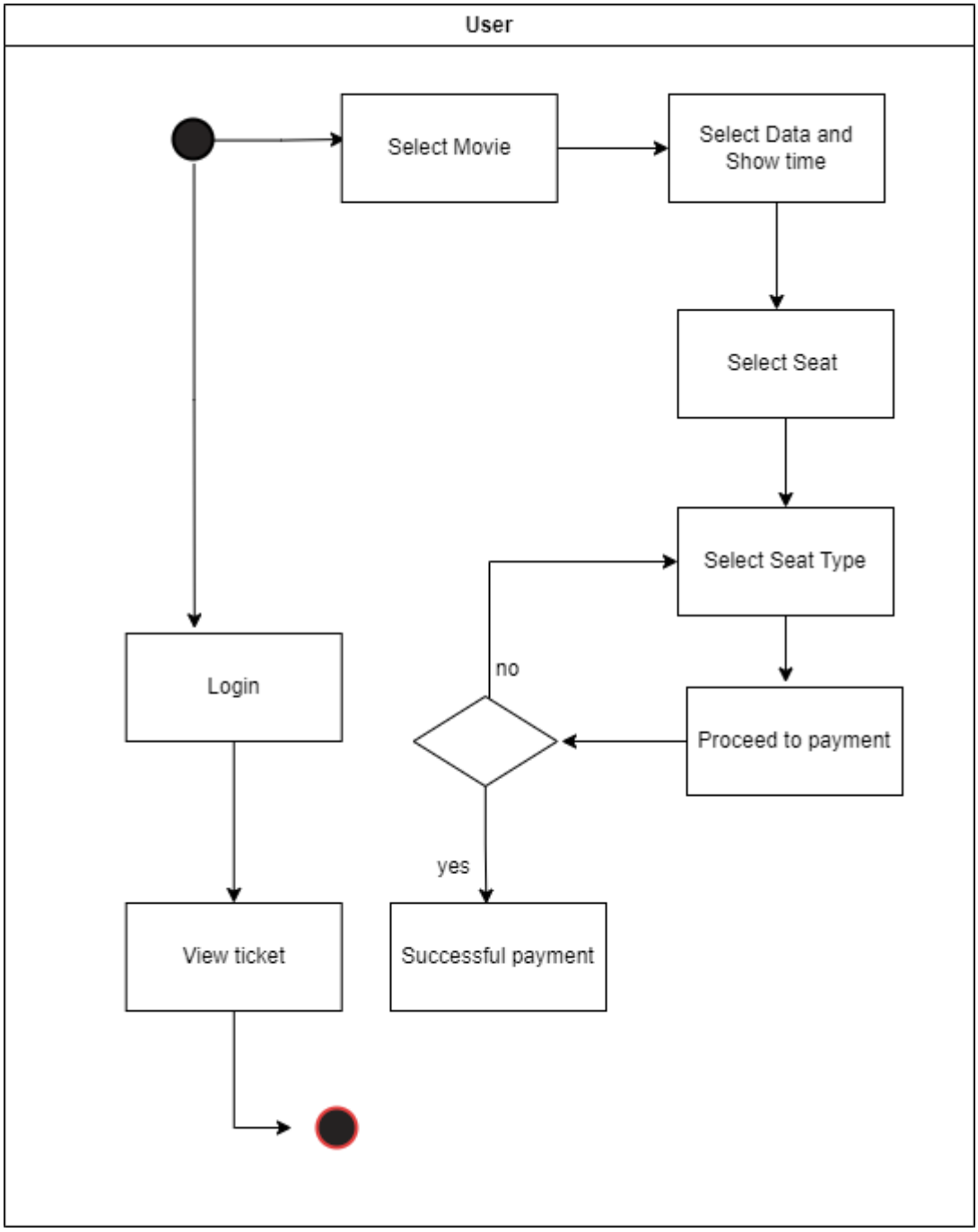


Figure 18: User's activity diagram

8.4 Requirement Catalogue

A requirements record is highly crucial and beneficial while developing a system. Required specifications must now be added to a catalog that follows a format. The requirements catalogue for this project is given beneath:

Purchasing tickets requirement catalogue

Source	Sing of	Priority	Requirement ID
Admin	All users	Must	U001
Functional Requirement **Purchase tickets** **All user can select movie, schedule and seat, seat type while purchasing tickets by downloading the application.**			
Non-functional requirement			
Description	Target value	Acceptance value	Comments
Purchasing tickets per day	500	1000	

Table 16: Requirement catalogue for purchasing ticket

Log in to view the tickets requirement catalogue

Source	Sing of	Priority	Requirement ID
Admin	All users	Must	U002
Functional Requirement **Login for user to view tickets** **All user must login to view purchased tickets otherwise they cannot find out their purchased tickets.**			
Non-functional requirement			
Description	Target value	Acceptance value	Comments
Login per day	500	1000	

Table 17: Requirement catalogue to view ticket

Requirement catalogue to apply for a refund

Source	Sing of	Priority	Requirement ID
Admin	All users	Must	R003
Functional Requirement			
Non-functional requirement **User can apply for refund policy by maintaining the given guidelines in the application**			
Description	Target value	Acceptance value	Comments
Apply for refund	50	200	

Table 18: Requirement catalogue to apply for a refund

8.5 Prioritized Requirements (PRL)

A MoSCoW prioritization procedure was used to make a list of the determined requirements. As follows is the focused requirements rundown for **Lion Cinema**

Must have requirements:

SL	Requirements	Priority
01	Login for all users	Must have
02	Manage upcoming movies information	Must have
03	Manage now showing movies information	Must have
04	Manage seat and movie schedule	Must have
05	Phone number verification	Must have

Should have requirements:

SL	Requirements	Priority
01	Manage theater information	Should have
02	Email verification	Should have

Could

SL	Requirements	Priority
01	Refund policy	Could have

8.6 Prototypes of the Application

Some prototypes of the Lion Cinema application are given beneath:

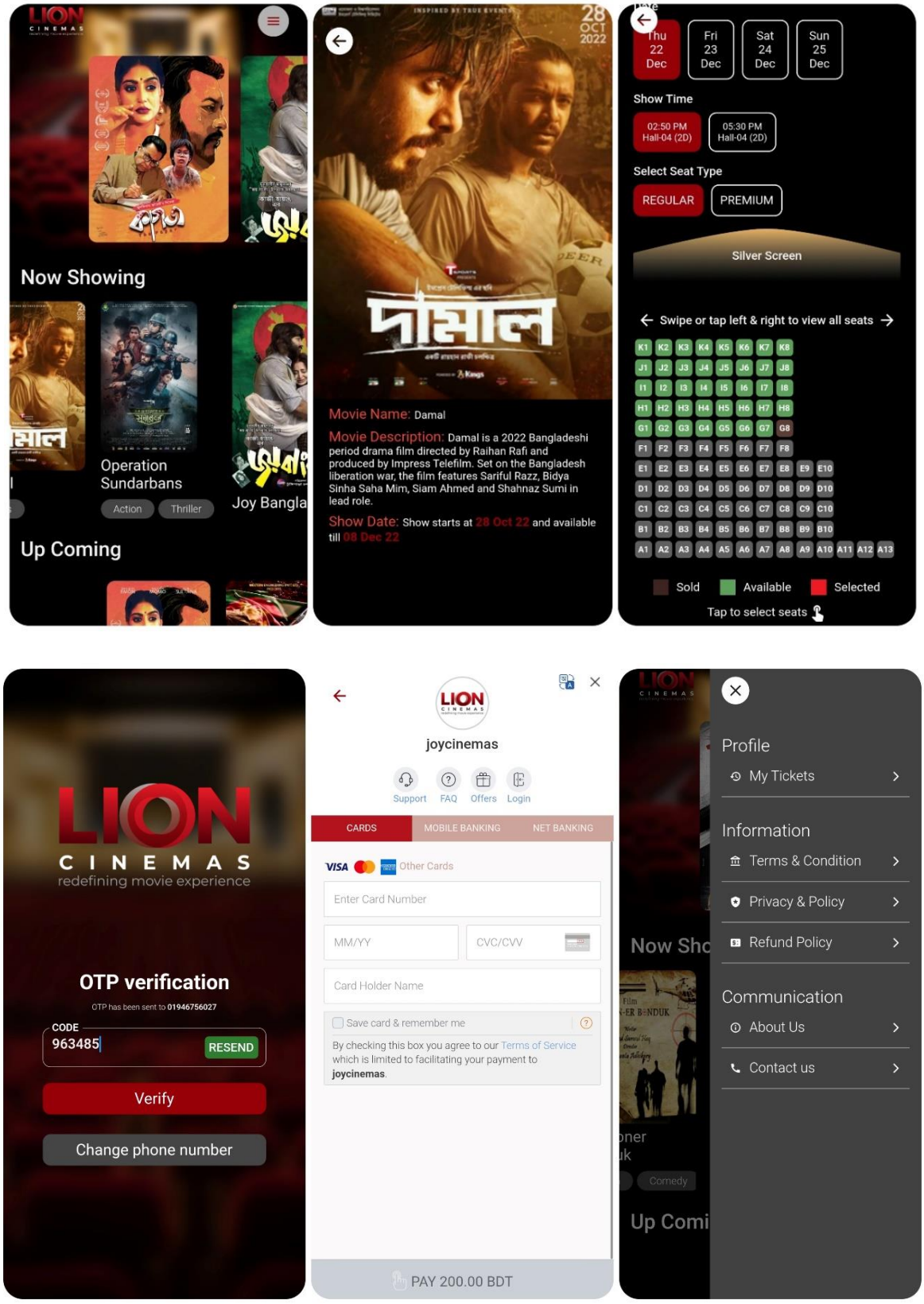


Figure 19: Prototype of the application

CHAPTER 9: ENGINEERING

9.1 Application Module

Ticket purchasing module:

SL No	Activities of user	SL No	Activities of application
01	The user chooses a movie and then clicks on it	01	The app will take the user to the ticket-purchasing page
02	The user selects a date	02	The app will show the next segment to select a showtime
03	The user selects show time	03	The app will show the next segment to select the seat type
04	The user selects the seat type	04	The app will provide available seats to select
05	User selects seat	05	The app will show an input text to provide a phone number
06	The user provides a phone number and clicks on the checkout button	06	The app verifies the phone number and takes the user to the payment segment
07	The user makes a payment	07	The app generates tickets

Table 19: Ticket purchasing module

View purchased tickets module:

SL No	Activities of user	SL No	Activities of application
01	The user clicks on my ticket	01	The app will show an input text to provide the user's phone number
02	The user provides the phone number	02	The app verifies the phone number and displays the purchased tickets by the user

Table 20: view purchased tickets module

9.2 Use Case Diagram of Lion Cinema

Figure 20: Use case of Lion Cinema

9.3 Class Diagram of Lion Cinema

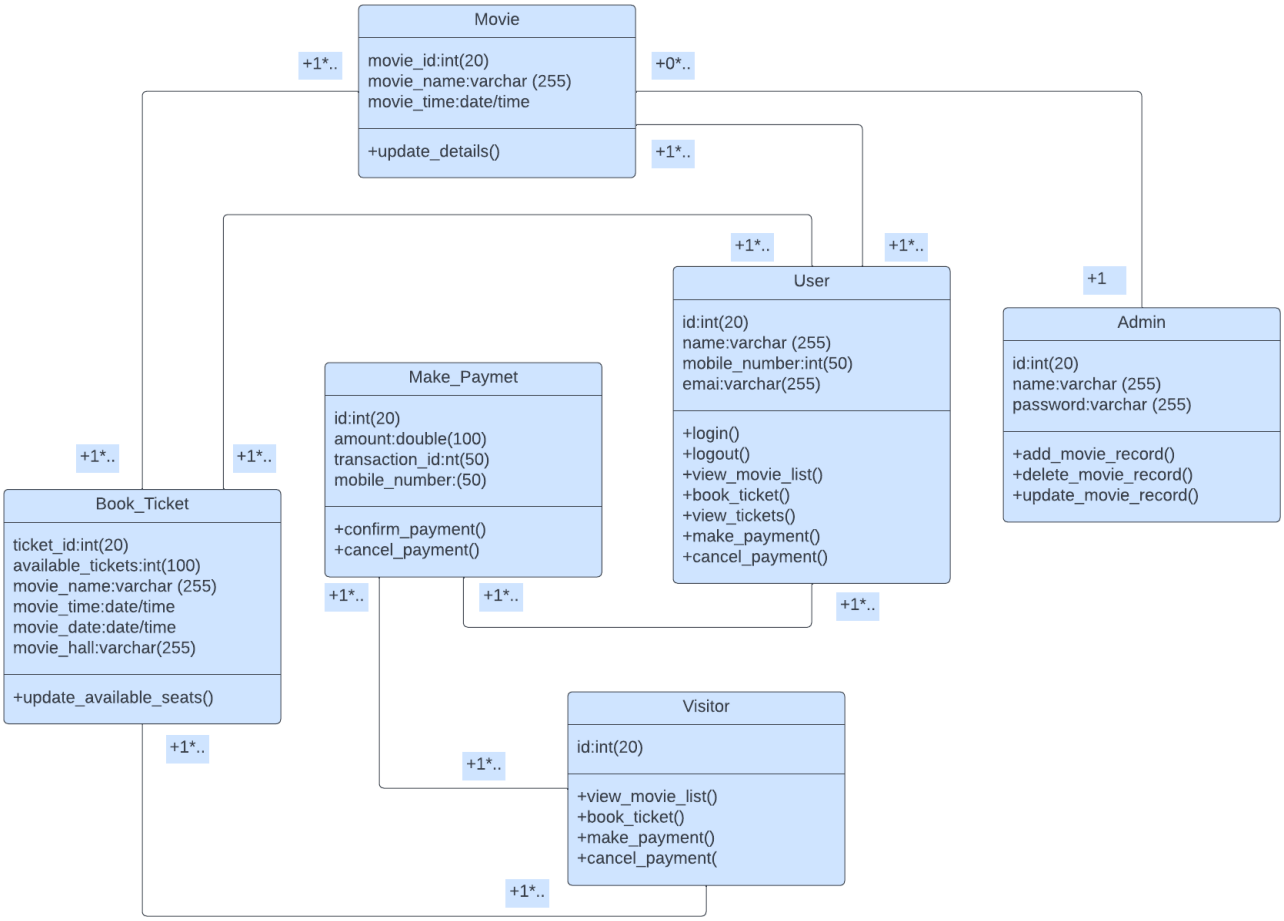


Figure 21: Class diagram of Lion Cinema

9.4 Entity Relationship Diagram of Lion Cinema

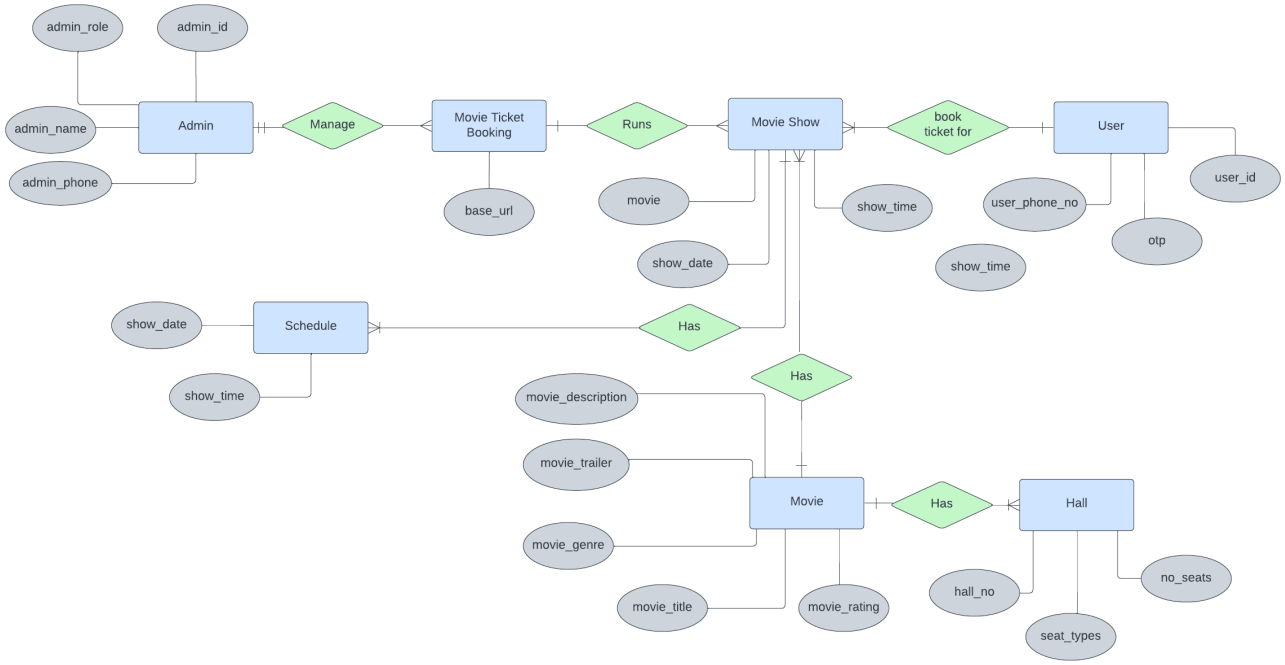


Figure 22: Entity relationship diagram of Lion Cinema

9.5 Sequence Diagram of Lion Cinema

Sequence diagram of a user

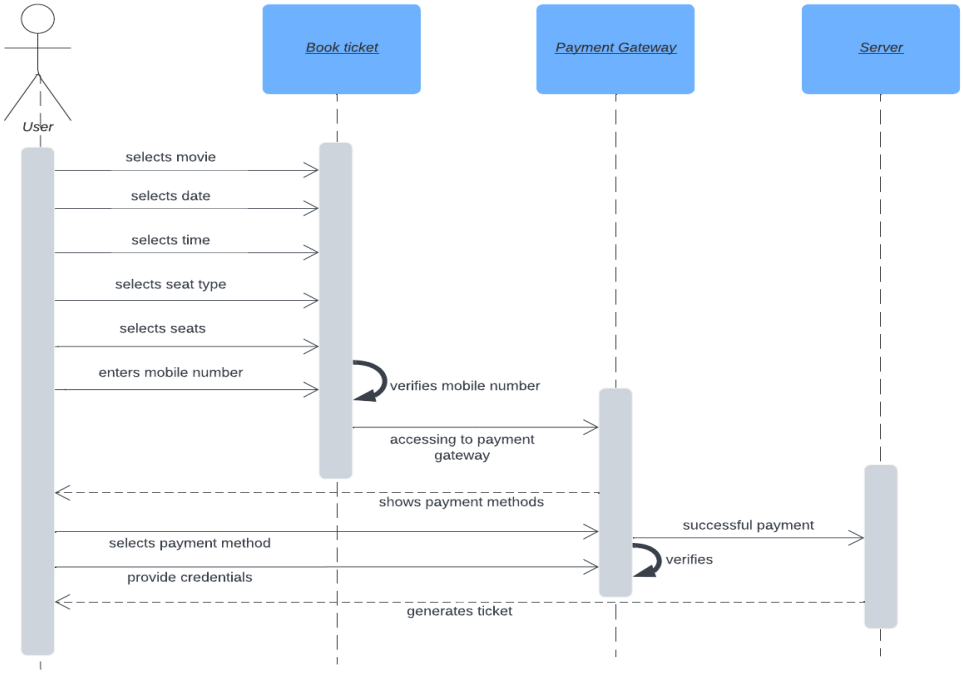


Figure 23: Sequence diagram of a user

Sequence diagram of admin

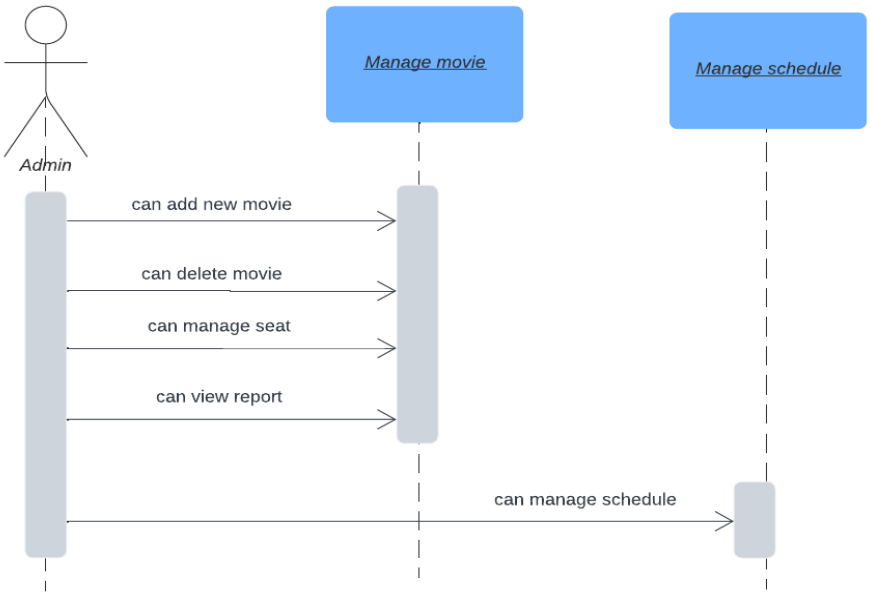


Figure 24: Sequence diagram of admin

9.6 Component Diagram of Lion Cinema

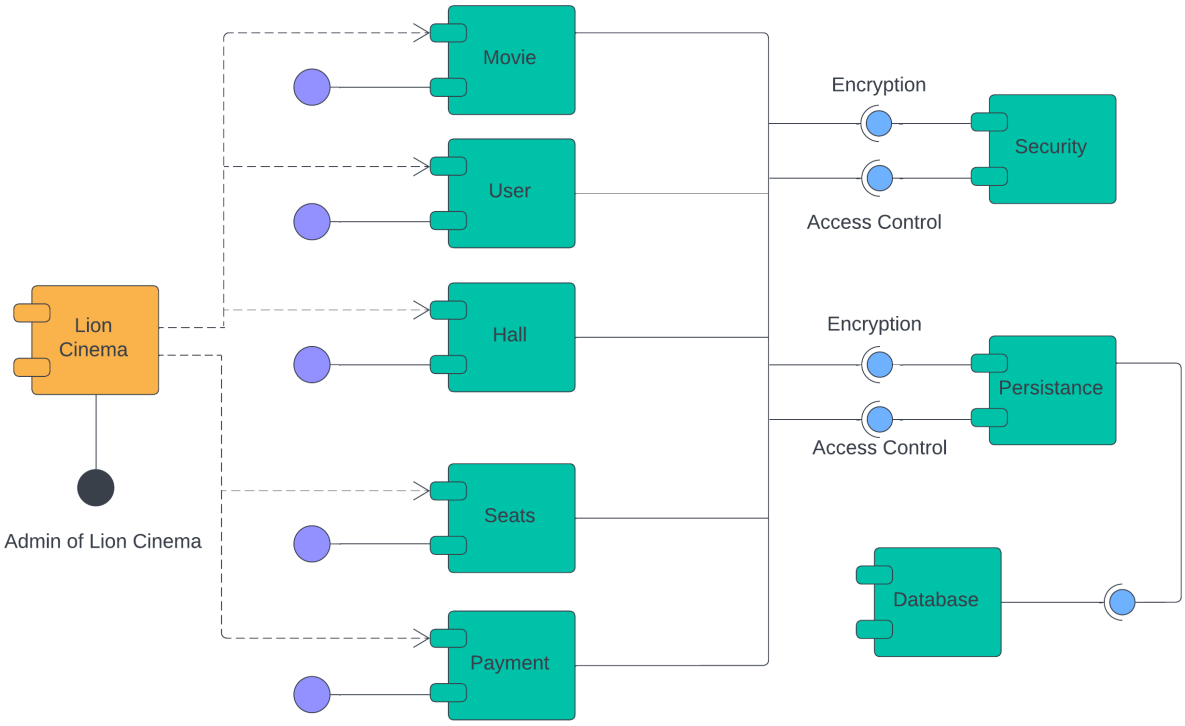


Figure 25: Component diagram

9.7 Deployment Diagram of Lion Cinema

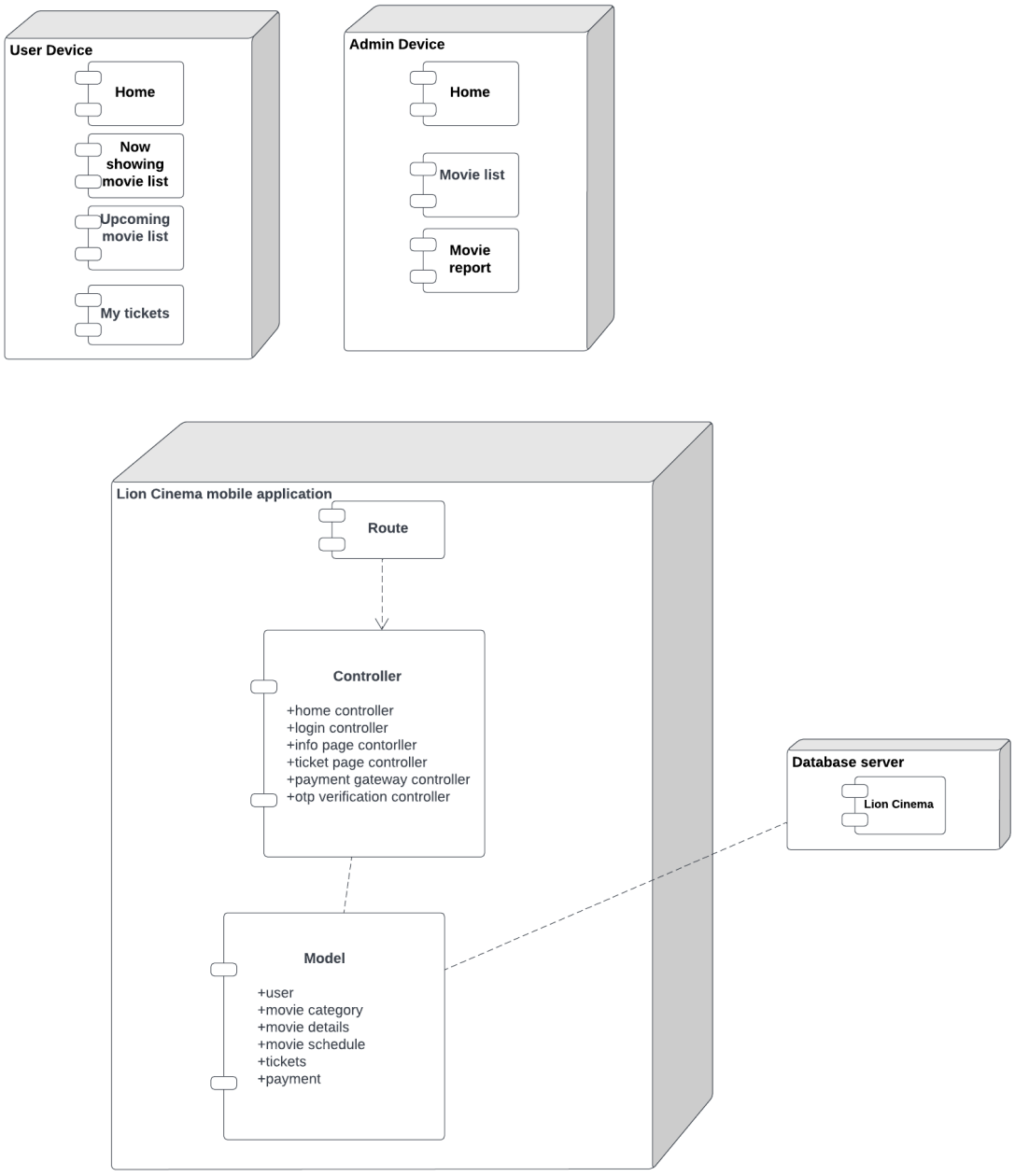
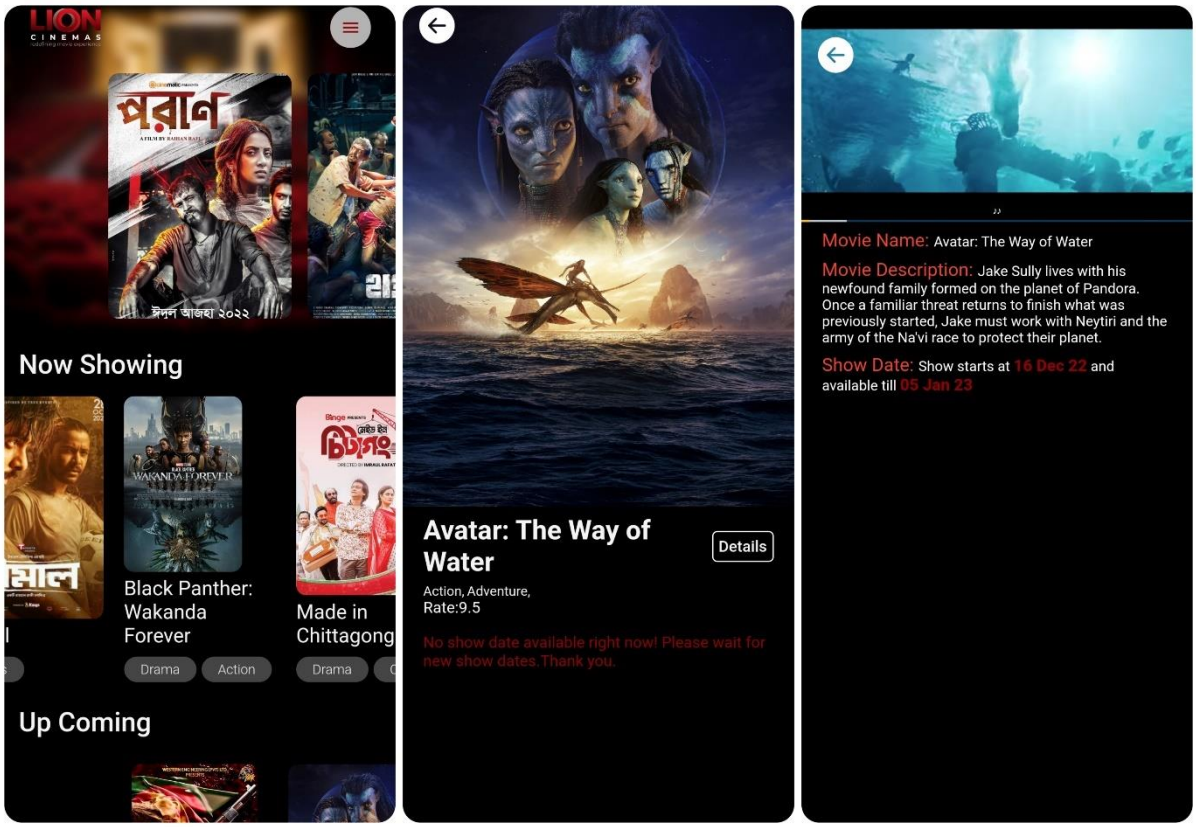
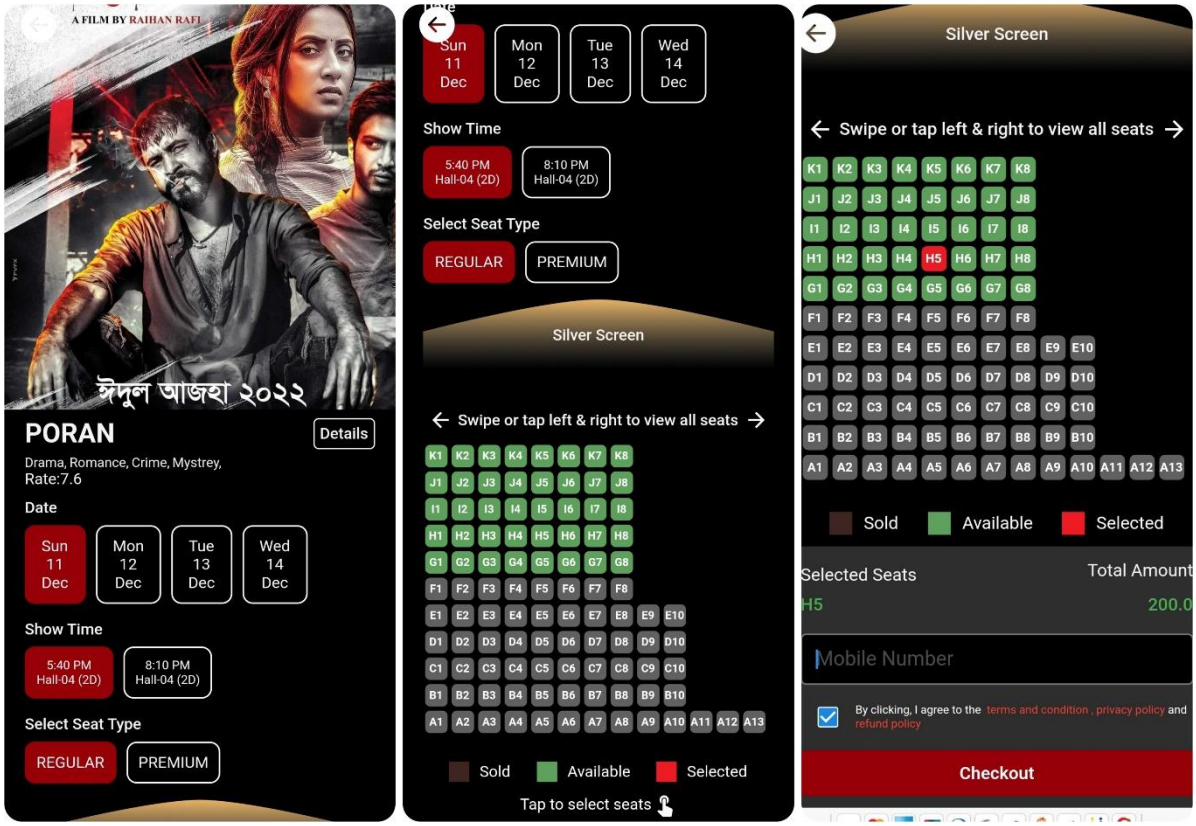


Figure 26: Deployment diagram

9.8 Interface Design of Lion Cinema





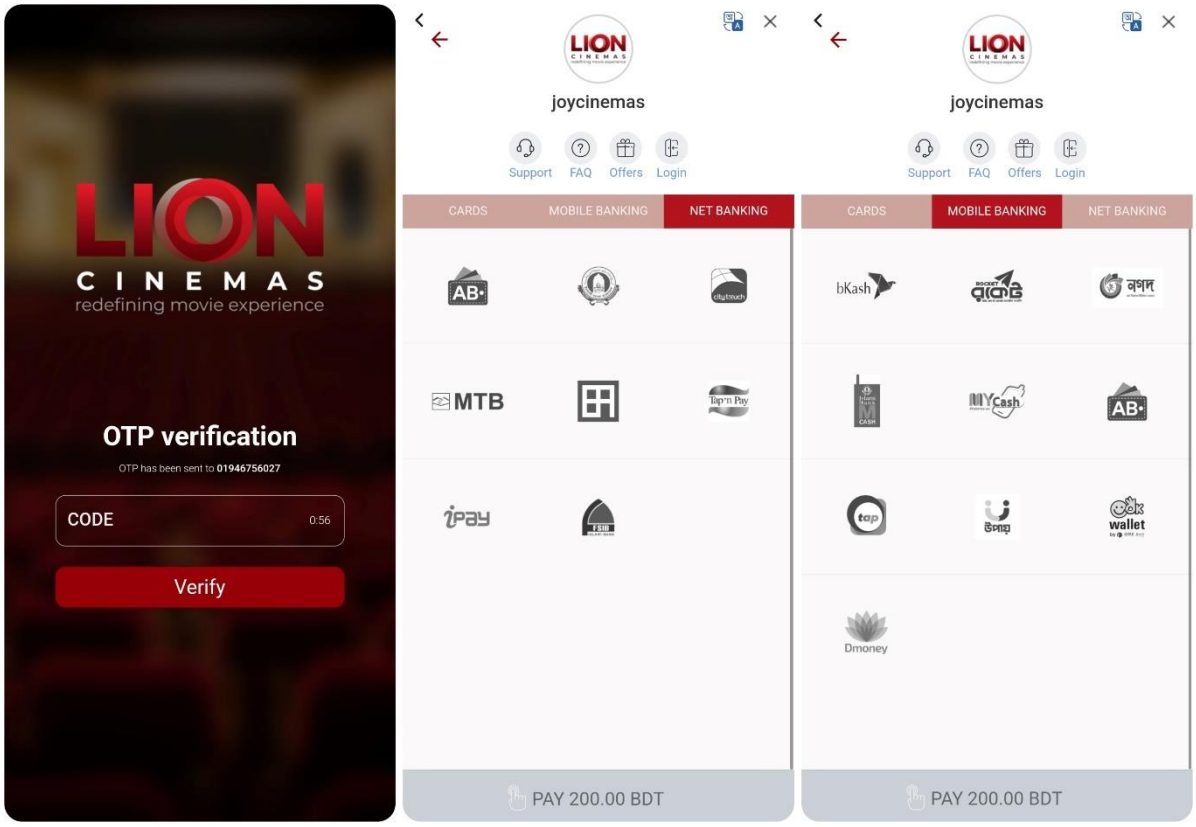


Figure 27: User interface example

CHAPTER 10: DEPLOYMENT

10.1 Core Module Coding Samples

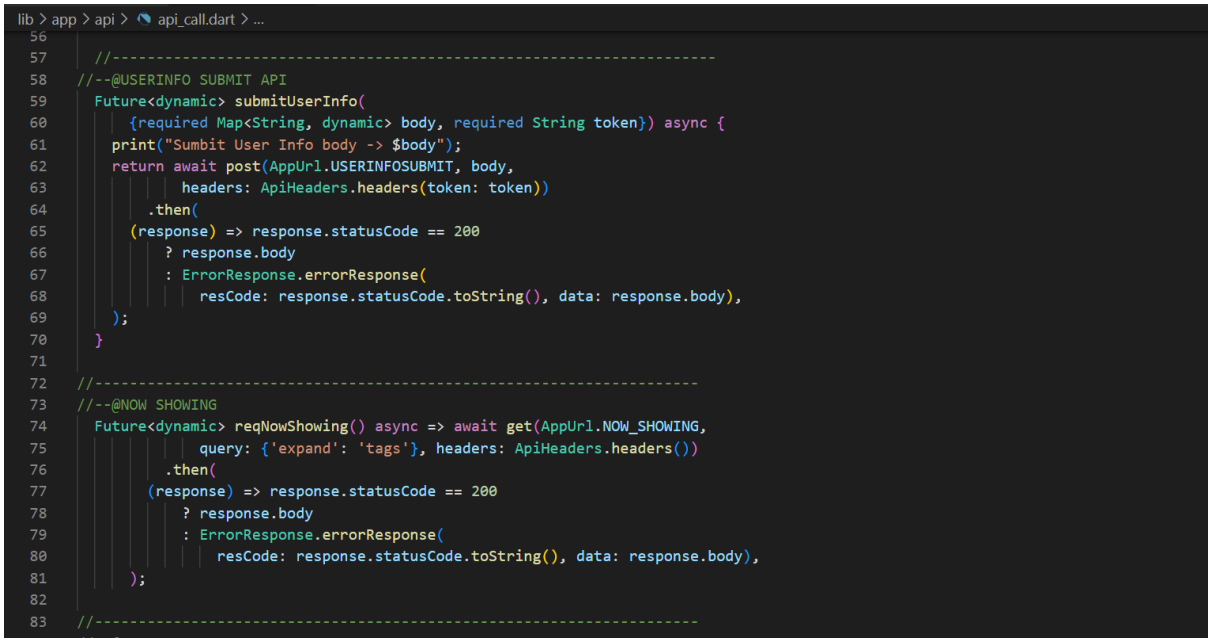
This project has been developed using dart language in flutter and multiple dependencies have been used to make it more optimized.

API calling sample code:

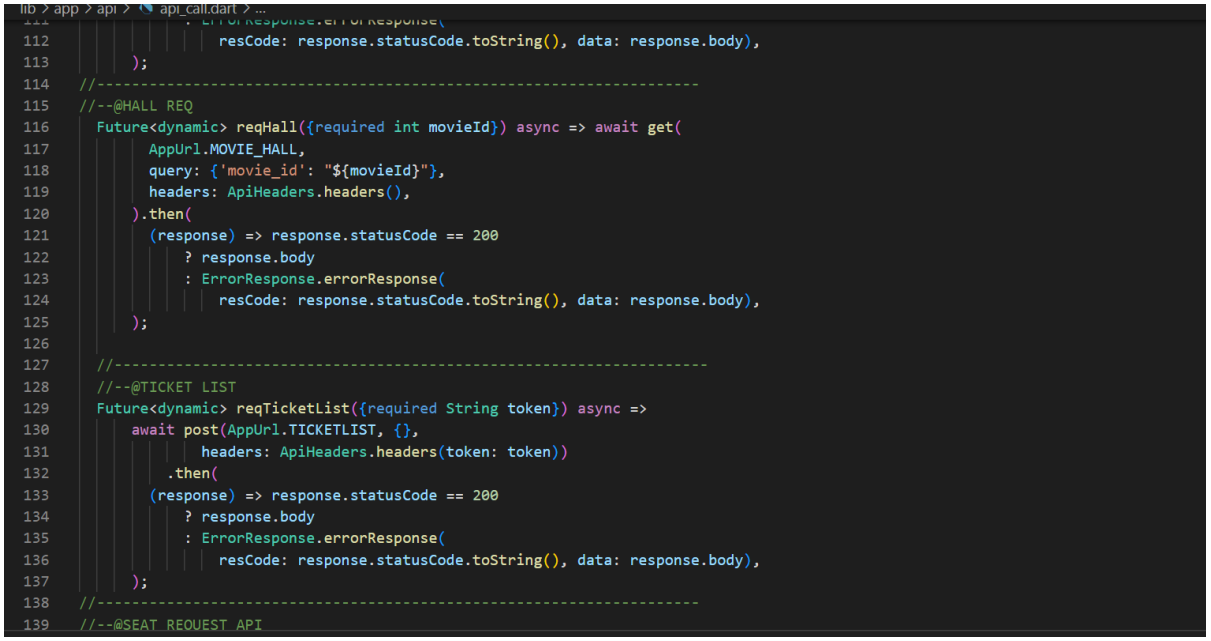
```dart
import 'package:get/get.dart';
import 'package:lion_cinema/app/api/url.dart';

import '../api_data_model/error_response.dart';
import 'api_header.dart';

class ApiCall extends GetConnect {

//---------------------------------------------------------------------
//--@OTP REQUEST API
  Future<dynamic> reqOtp({required Map<String, dynamic> body}) async =>
      await post(AppUrl.OTP, body, headers: ApiHeaders.headers()).then(
        (response) => response.statusCode == 200
            ? response.body
            : ErrorResponse.errorResponse(
                resCode: response.statusCode.toString(), data: response.body),
      );

//---------------------------------------------------------------------
//--@SEAT SUBMIT API
  Future<dynamic> submitSeat(
          {required Map<String, dynamic> body, required String token}) async =>
      await post(AppUrl.TICKETSUBMIT, body,
              headers: ApiHeaders.headers(token: token))
          .then(
        (response) => response.statusCode == 200
            ? response.body
            : ErrorResponse.errorResponse(
```

lib > app > api > api_call.dart > ...

```dart
            ? response.body
            : ErrorResponse.errorResponse(
                resCode: response.statusCode.toString(), data: response.body),
      );

  //---------------------------------------------------------------------
//--@LOGIN API
  Future<dynamic> requestLogin({required Map<String, dynamic> body}) async {
    print("login body -> $body");
    return await post(AppUrl.LOGIN, body, headers: ApiHeaders.headers()).then(
      (response) => response.statusCode == 200
          ? response.body
          : ErrorResponse.errorResponse(
              resCode: response.statusCode.toString(), data: response.body),
    );
  }

  //---------------------------------------------------------------------
//--@USERINFO VIEW API
  Future<dynamic> requestUserInfo({required String token}) async {
    return await post(AppUrl.USERINFOVIEW, {},
            headers: ApiHeaders.headers(token: token))
        .then(
      (response) => response.statusCode == 200
          ? response.body
          : ErrorResponse.errorResponse(
              resCode: response.statusCode.toString(), data: response.body),
    );
```



```
lib > app > api > api_call.dart > ...
56
57    //-----------------------------------------------------------------
58  //--@USERINFO SUBMIT API
59    Future<dynamic> submitUserInfo(
60        {required Map<String, dynamic> body, required String token}) async {
61      print("Sumbit User Info body -> $body");
62      return await post(AppUrl.USERINFOSUBMIT, body,
63              headers: ApiHeaders.headers(token: token))
64          .then(
65        (response) => response.statusCode == 200
66            ? response.body
67            : ErrorResponse.errorResponse(
68                resCode: response.statusCode.toString(), data: response.body),
69      );
70    }
71
72  //-----------------------------------------------------------------
73  //--@NOW SHOWING
74    Future<dynamic> reqNowShowing() async => await get(AppUrl.NOW_SHOWING,
75              query: {'expand': 'tags'}, headers: ApiHeaders.headers())
76          .then(
77        (response) => response.statusCode == 200
78            ? response.body
79            : ErrorResponse.errorResponse(
80                resCode: response.statusCode.toString(), data: response.body),
81      );
82
83  //-----------------------------------------------------------------
```



```
lib > app > api > api_call.dart > ...
112               resCode: response.statusCode.toString(), data: response.body),
113       );
114   //-----------------------------------------------------------------
115   //--@HALL REQ
116     Future<dynamic> reqHall({required int movieId}) async => await get(
117         AppUrl.MOVIE_HALL,
118         query: {'movie_id': "${movieId}"},
119         headers: ApiHeaders.headers(),
120       ).then(
121         (response) => response.statusCode == 200
122             ? response.body
123             : ErrorResponse.errorResponse(
124                 resCode: response.statusCode.toString(), data: response.body),
125       );
126
127     //-----------------------------------------------------------------
128     //--@TICKET LIST
129     Future<dynamic> reqTicketList({required String token}) async =>
130         await post(AppUrl.TICKETLIST, {},
131                 headers: ApiHeaders.headers(token: token))
132             .then(
133         (response) => response.statusCode == 200
134             ? response.body
135             : ErrorResponse.errorResponse(
136                 resCode: response.statusCode.toString(), data: response.body),
137       );
138   //-----------------------------------------------------------------
139   //--@SEAT REQUEST API
```

Figure 28: Code for API calls

Payment gateway sample code:

```
class PaymentgatewayController extends GetxController {
  final webViewController = Rxn<InAppWebViewController>();
  InAppWebViewGroupOptions options = InAppWebViewGroupOptions(
      crossPlatform: InAppWebViewOptions(
          javaScriptEnabled: true,
          preferredContentMode: UserPreferredContentMode.RECOMMENDED,
          useShouldOverrideUrlLoading: true
          // // transparentBackground: true,
          // javaScriptCanOpenWindowsAutomatically: true,
          // useShouldInterceptAjaxRequest: true,
          // useShouldOverrideUrlLoading: true,
          // mediaPlaybackRequiresUserGesture: false,
          ), // InAppWebViewOptions
      android: AndroidInAppWebViewOptions(
        safeBrowsingEnabled: false,
        useHybridComposition: true,
      ), // AndroidInAppWebViewOptions
      ios: IOSInAppWebViewOptions(
        allowsInlineMediaPlayback: true,
      )); // IOSInAppWebViewOptions // InAppWebViewGroupOptions

  setWebViewController(con) {
    webViewController.value = con;
    update();
  }

  final pullToRefreshController = Rxn<PullToRefreshController>();

  enableRefresh() {
    pullToRefreshController.value = PullToRefreshController(
      options: PullToRefreshOptions(
        color: Colors.blue,
      ), // PullToRefreshOptions
      onRefresh: () async {
        if (Platform.isAndroid) {
          webViewController.value?.reload();
        } else if (Platform.isIOS) {
          webViewController.value?.loadUrl(
              urlRequest:
                  URLRequest(url: await webViewController.value?.getUrl()));
        }
      },
    ); // PullToRefreshController
  }

  RxBool isPageLoading = true.obs;

  setLoadingStatus(status) {
    isPageLoading.value = false;
    update();
  }

  @override
```

Figure 29: Code of payment gateway

View purchased ticket page sample code:

```
class TicketpageController extends GetxController {
  RefreshController refreshController =
      RefreshController(initialRefresh: false);
  dynamic argumentData;
  TextEditingController phoneNumber = TextEditingController();
  RxString token = "".obs;
  tokenUpdater({required String value}) {
    token.value = value;
    update();
  }

  RxBool isLoading = false.obs;

  RxList<dynamic> offlineTicketList = <dynamic>[].obs;
  requestTicketList({required String token}) async {
    isLoading.value = true;
    update();
    await ApiCall()
        .reqTicketList(
            token: GetStorage().read('Login_info')['token'].toString())
        .then((value) {
      if (value['error-occured'] != null) {}
      {
        if (value['success'] == true) {
          offlineTicketList.value = value['data']['items'] ?? [];
          offlineTicketList.refresh();
        } else {}
```

```
      isLoading.value = false;
      update();
      print("Ticket List from API -> ${value}");
      print("--------------------------------------");
      print("Ticket List from OFFLINE -> ${offlineTicketList}");
    });
  }

  RxBool isOtpLoading = false.obs;
  reqOtp({required String phone}) async {
    isOtpLoading.value = true;
    update();
    await ApiCall().reqOtp(body: {
      'phone': phone,
    }).then((value) {
      if (value['error-occured'] == null) {
        if (value['data']['success'] == true) {
          // print("here");
          isOtpLoading.value = false;
          Get.toNamed(
            Routes.OTPVERIFICATIONSCREEN,
            arguments: {"phone": phoneNumber.text, "msg": "Login"},
          );
        } else if (value['data']['success'] == false) {
          Get.snackbar("Warning", " ${value['data']['msg']} ",
              snackPosition: SnackPosition.BOTTOM,
              colorText: Colors.white,
```

Figure 30: Code of view purchased ticket page

10.2 Possible Problem Break Down

To make the framework enhancement simpler and more interesting, the project should be divided into manageable tasks. The anticipated breakdowns for the proposed project will follow:

Analysis and Database design

- Identification and completion of requirements,
- data gathering and normalization of the obtained data
- ERD and data dictionary production
- Schema creation following the data dictionary

Front End design

- Create the fundamental dart code using flutter for Lion Cinema.
- Associate with the various pages
- Use the appropriate icon, color, and graphics
- Create a responsive user interface

Ticket Purchasing Management

- User selects their desired movie for ticket purchasing
- Selects the schedule and seat
- Proceed to payment
- Choose a favorable payment option
- Make successful payment

Admin Management

- Show report based on daily or monthly ticket purchasing
- Control movie update (add or delete) facility

10.3 Prioritization while Developing

In the identification segment requirements list has already been prioritized by using MoSCoW prioritization now it's time to implement those in the development stage. The tasks needed to be implemented are:

- Login facility implementation
- Implementation of mobile number verification
- Ticket purchasing and viewing facility implementation
- Movie schedule selection facility implementation
- Seat selection facility implementation
- Now showing and upcoming movie list implementation

CHAPTER 11: TESTING

11.1 Test Acceptance Plan

Testing is a crucial step in creating effective software. A testing strategy needs to be created during the analytical stage. These plans should be approved by both the client and the developer. (*Different Types Of Mobile App Testing | LambdaTest*, n.d.)The essential testing requirements will be accomplished according to the project's test strategy. Two types of testing exist:

Functional Testing

- **Unit Testing**
  - Mobile number validation
  - Showing payment procedures after mobile number validation
- **Module Testing**
  - The process of payment without data
  - The process of payment with the wrong credential
  - The process to credential with valid data
- **Integration Testing**
  - Successful login to view purchased tickets
  - Users can successfully select movie, schedule, and seat type

Non-Functional Testing

- **Acceptance Testing**
  - Resend OTP if it’s not delivered within 60 seconds
  - Changing mobile number
- **Security Testing**
  - OTP validation
  - Checking terms and conditions acceptance
- **Accessibility testing**
  - Color blindness testing by users
  - Testing by old aged users

11.2 Test Case

Test cases for the application are given below:

Unit testing: test case

Test Case Name	Unit Test		
Test Class			
Test Description			
Data Source	Test Steps	Expected Result	Actual Result

Module testing: test case

Test Case Name	Module Test		
Test Class			
Test Description			
Data Source	Test Steps	Expected Result	Actual Result

Integration testing: test case

Test Case Name	Integration Test		
Test Class			
Test Description			
Data Source	Test Steps	Expected Result	Actual Result

11.3 Unit Testing

Test 1: Mobile number validation

Test Case Name	Unit Test		
Test Class	Login controller		
Test Description	Mobile number validation		
Data Source	Test Steps	Expected Result	Actual Result
	Processing to checkout without providing mobile number	An error message should display to provide 11 digits mobile number	An error message is displaying to provide 11 digits mobile number

Table 21: Unit test case 1

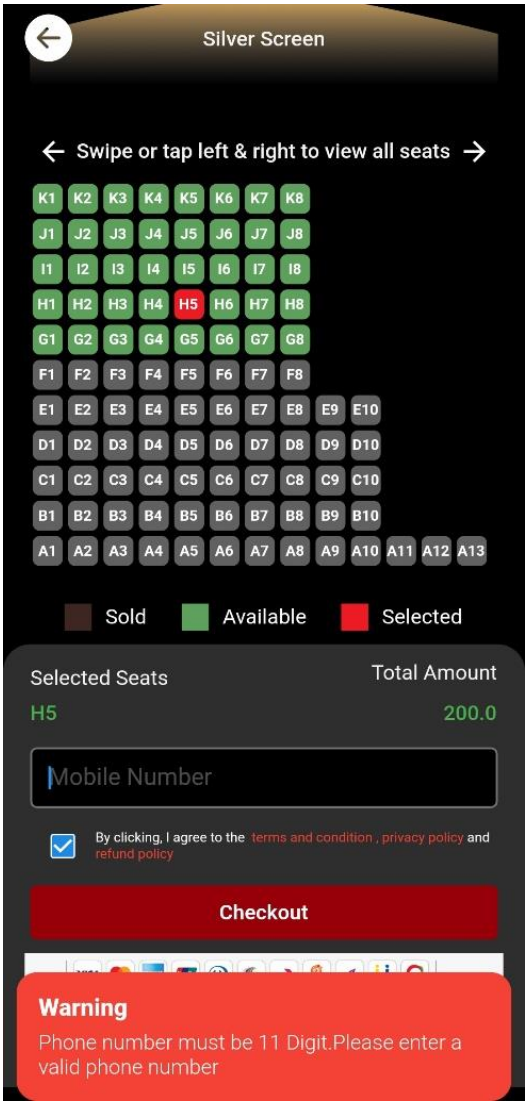


Figure 31: Mobile number validation

Test 2: Processing to make payment after mobile number validation

Test Case Name	Unit Test		
Test Class	OTP verification controller		
Test Description	Showing payment procedures after OTP validation		
Data Source	Test Steps	Expected Result	Actual Result
	Processing to payment segment	Showing payment segment	It shows expected result

Table 22: Unit test case 2

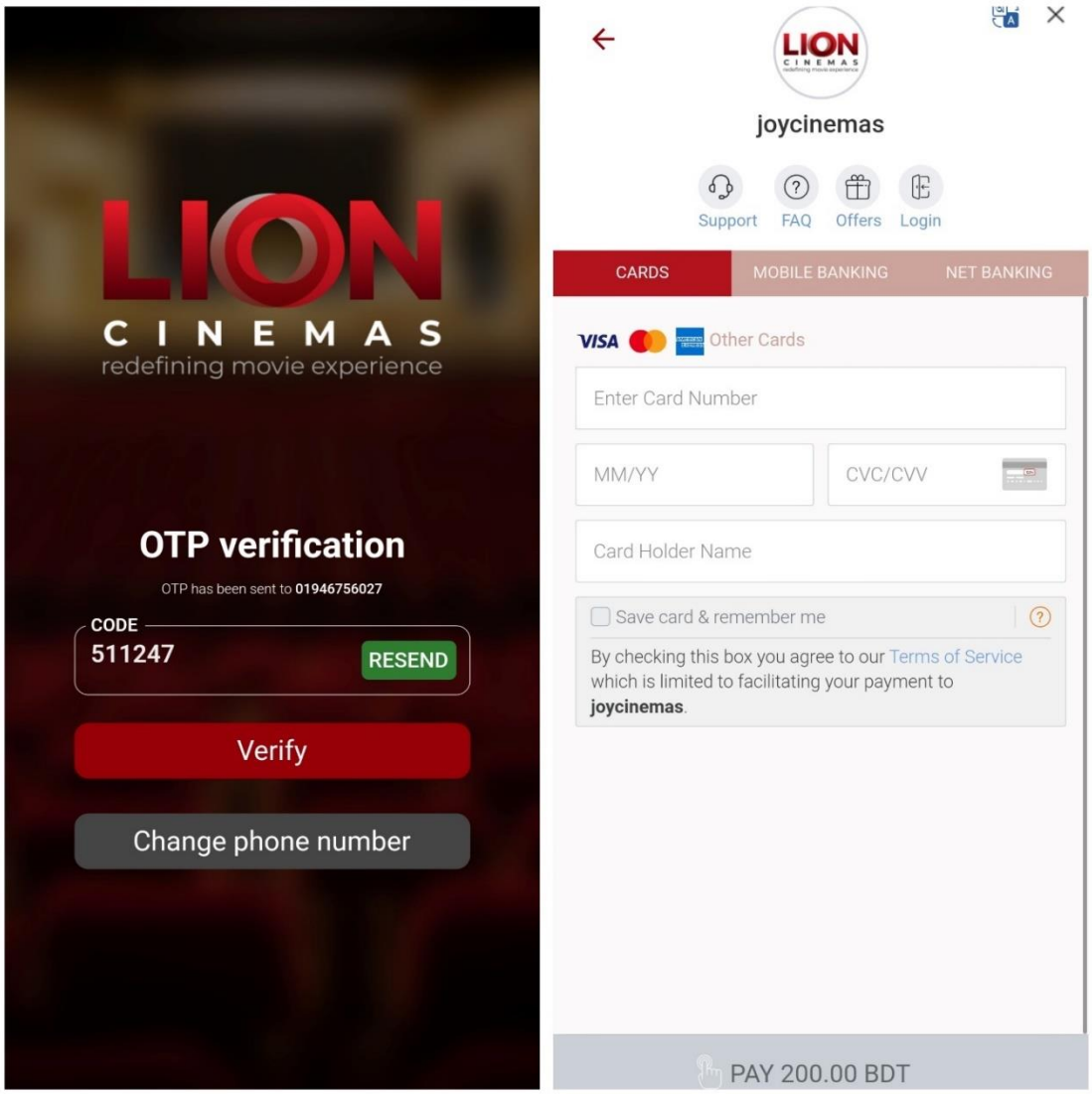


Figure 32: Payment segment after OTP validation

11.4 Module Testing

Test 1: Process to checkout without data

Test Case Name	Module Test		
Test Class	Login controller		
Test Description	Process to payment without data		
Data Source	Test Steps	Expected Result	Actual Result
	Enter to data	Provide a warning message to provide data	Showing a warning message to provide valid data

Table 23: Module test case 1

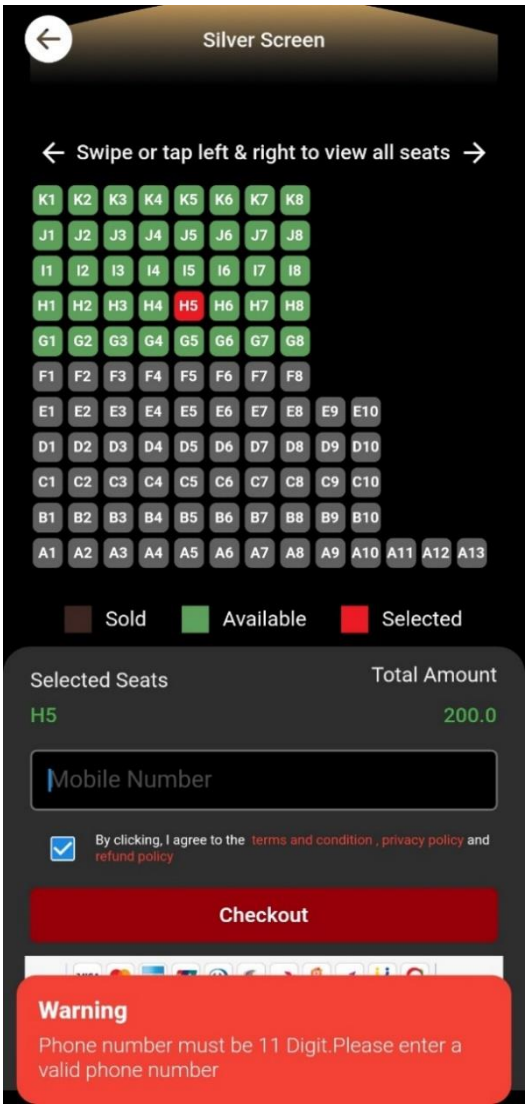


Figure 33: Data checking

Test 2: Process to checkout with wrong data

Test Case Name	Module Test		
Test Class	Login controller		
Test Description	Process to payment with wrong data		
Data Source	Test Steps	Expected Result	Actual Result
	Enter wrong data	Provide a warning message to provide valid data	Showing a warning message to provide valid data

Table 24: Module test case 2

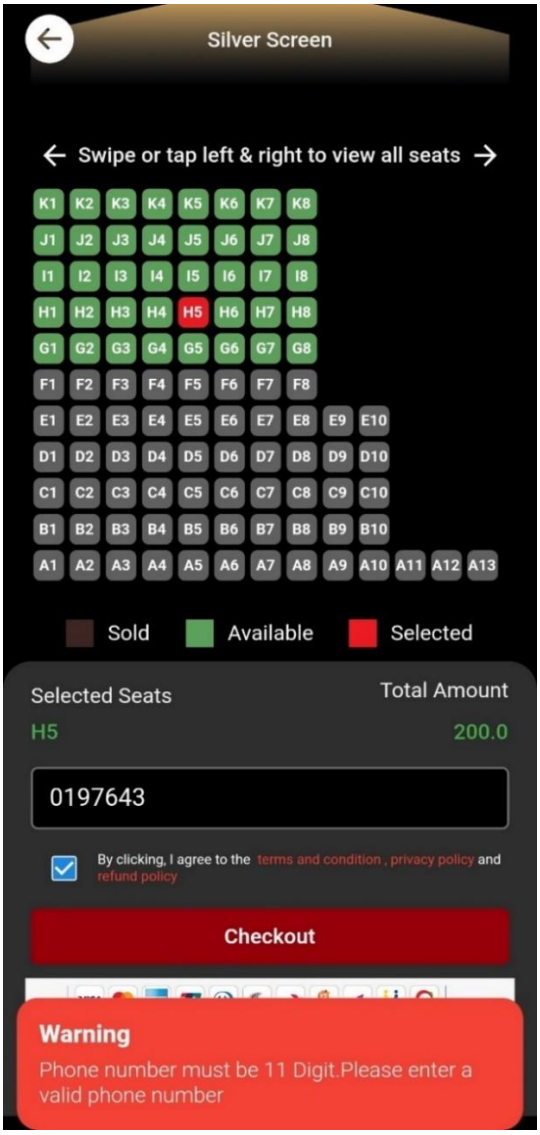


Figure 34: Data checking

Test 3: Process to checkout with valid data

Test Case Name	Module Test		
Test Class	Login controller		
Test Description	Process to payment with valid data		
Data Source	Test Steps	Expected Result	Actual Result
	Enter valid data	Showing payment segment after OTP confirmation	It shows expected output

Table 25: Module test case 3

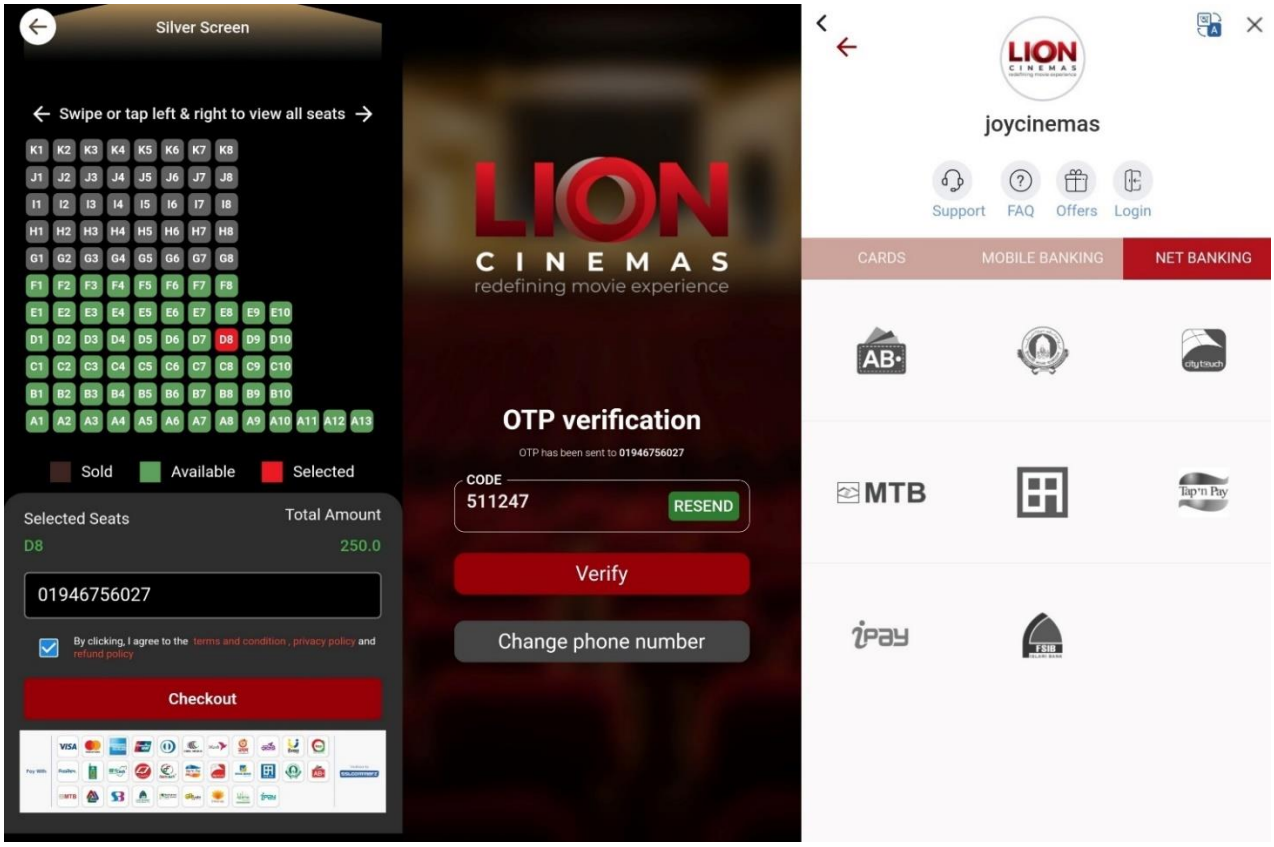


Figure 35: Data checking

11.5 Integration Testing

Test 1: View purchased tickets after login

Test Case Name	Integration Test		
Test Class	Tickets page controller		
Test Description	Login with phone number to view purchased tickets		
Data Source	Test Steps	Expected Result	Actual Result
	Enter phone number to login	Showing purchased tickets	It shows expected output

Table 26: Integration test case 1

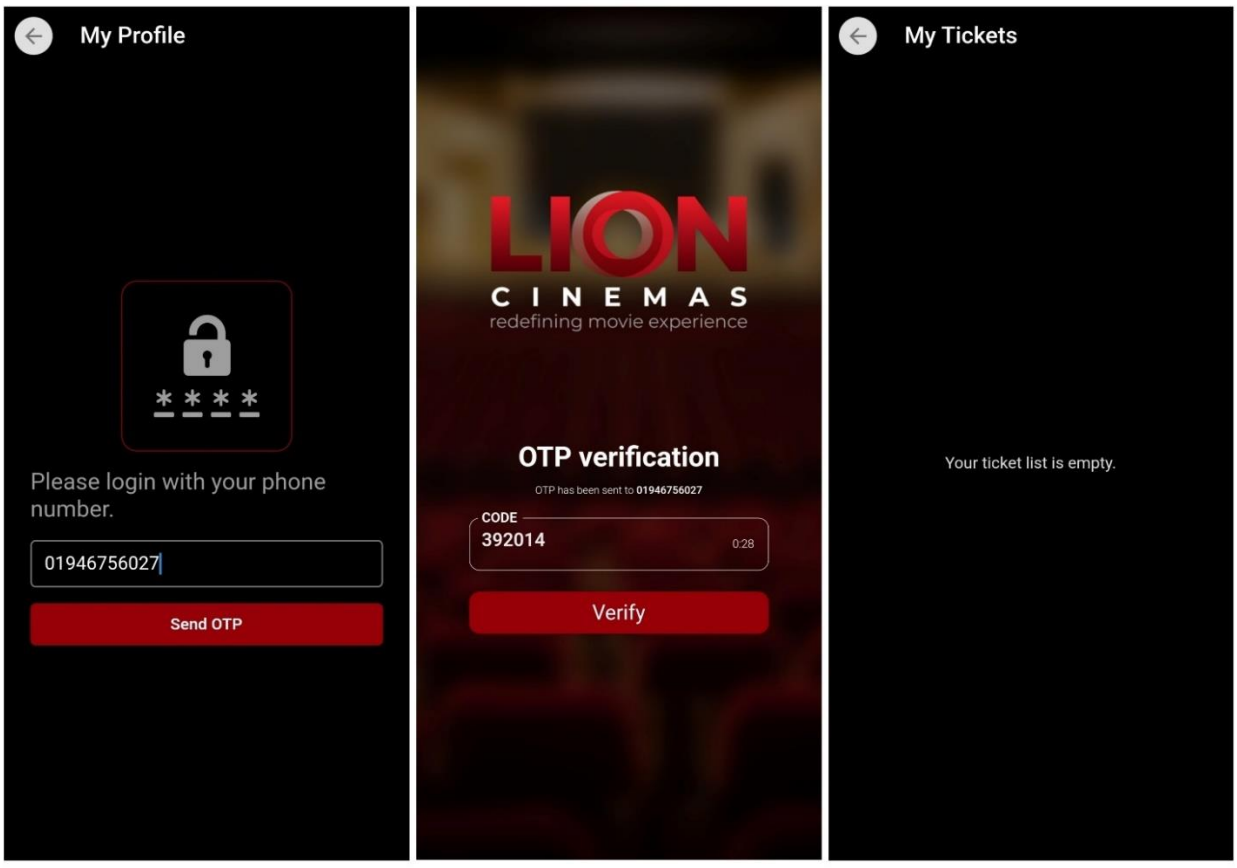


Figure 36: View purchased tickets

Test 2: Selecting movie, schedule, seat types

Test Case Name	Integration Test		
Test Class	Details page controller		
Test Description	Selecting movie, schedule, and seat types		
Data Source	Test Steps	Expected Result	Actual Result
	Select movie, schedule, seat types	Selecting movie, schedule, seat types working fine	It's working as expected

Table 27: Integration test case 2

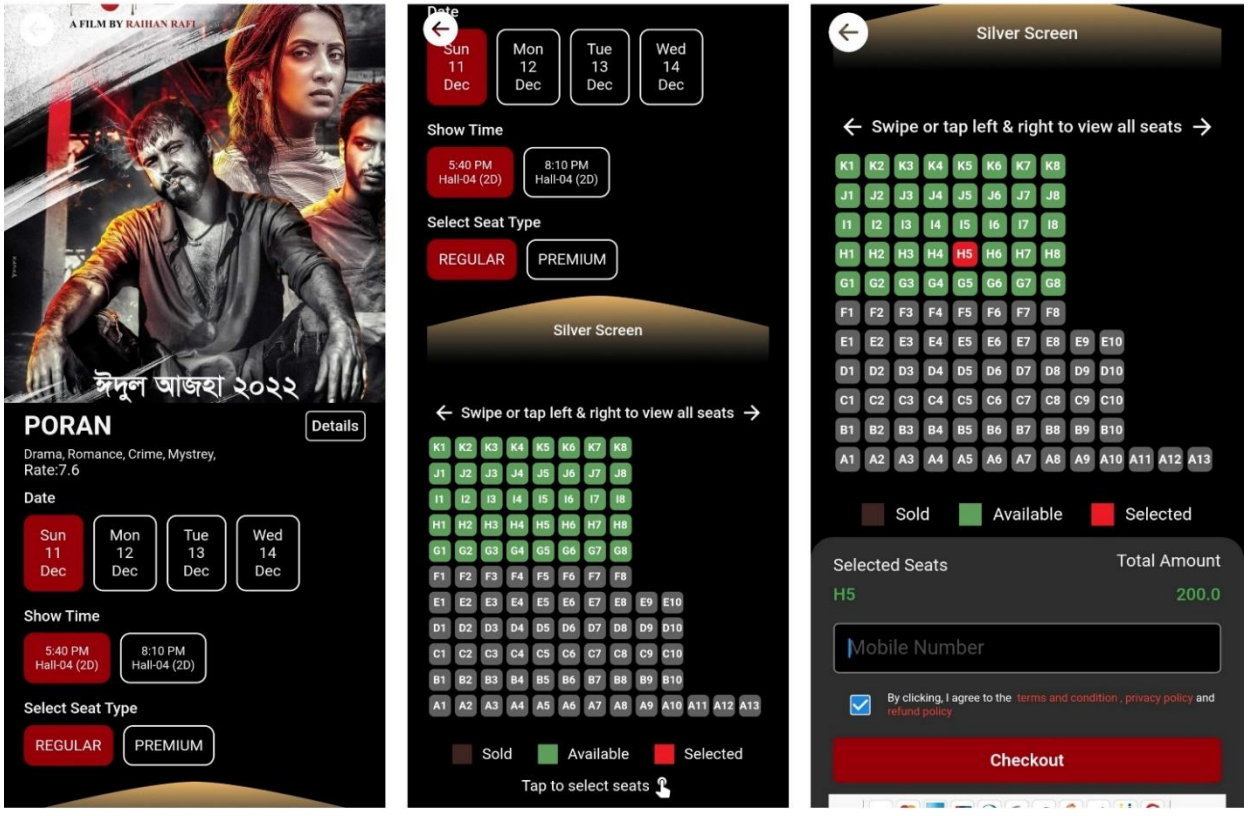


Figure 37: Movie schedule and seat types selection

11.6 Acceptance Testing

Test 1: Resend OTP

Test Case Name	Acceptance Test		
Test Class	OTP verification controller		
Test Description	Resend OTP		
Data Source	Test Steps	Expected Result	Actual Result
	Click on verify to send OTP	If OTP doesn't come in 60 seconds resend option will appear	It's working as expected

Table 28: Acceptance test case 1

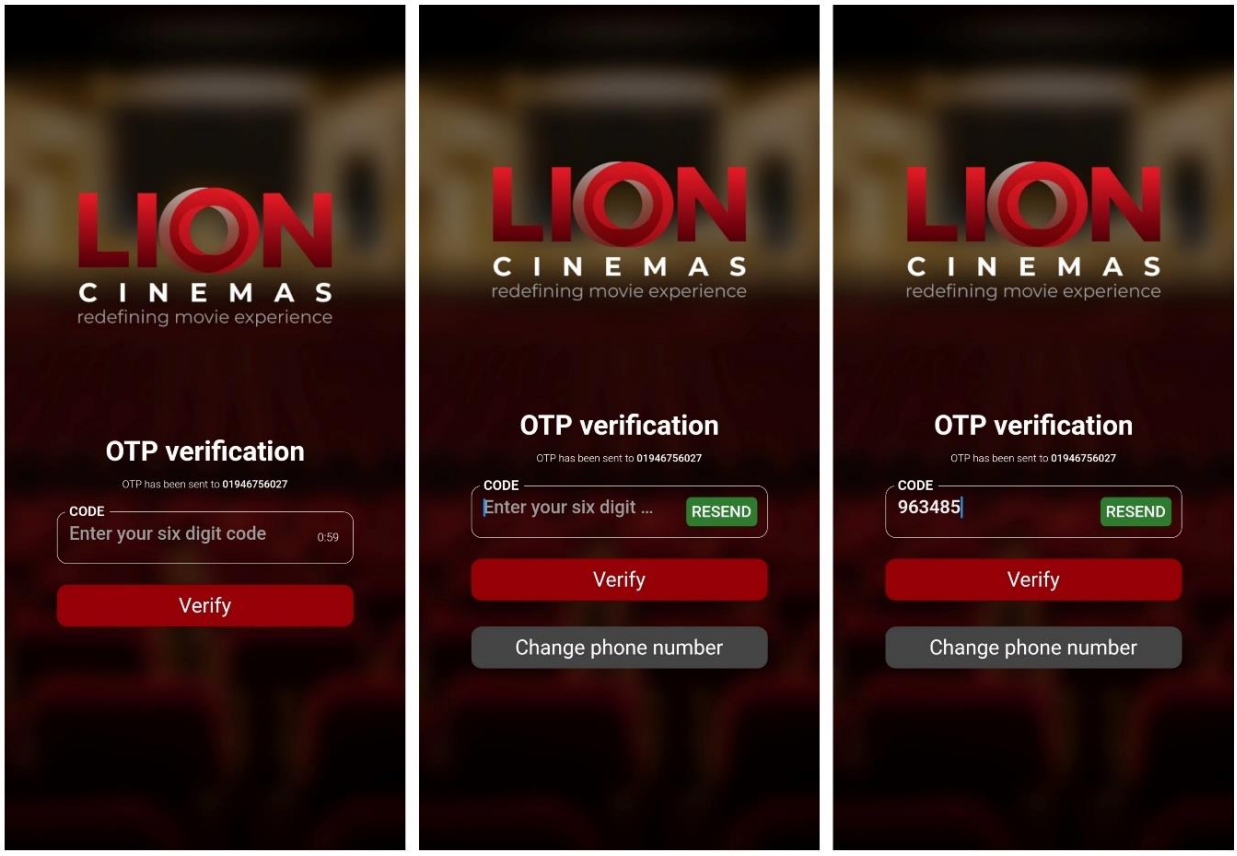


Figure 38: Resend OTP sample

Test 2: Change phone number

Test Case Name	Acceptance Test		
Test Class	OTP verification controller		
Test Description	Resend OTP		
Data Source	Test Steps	Expected Result	Actual Result
	Click on verify to send OTP	If OTP doesn't come in 60 seconds resend option will appear	It's working as expected

Table 29: Acceptance test case 2

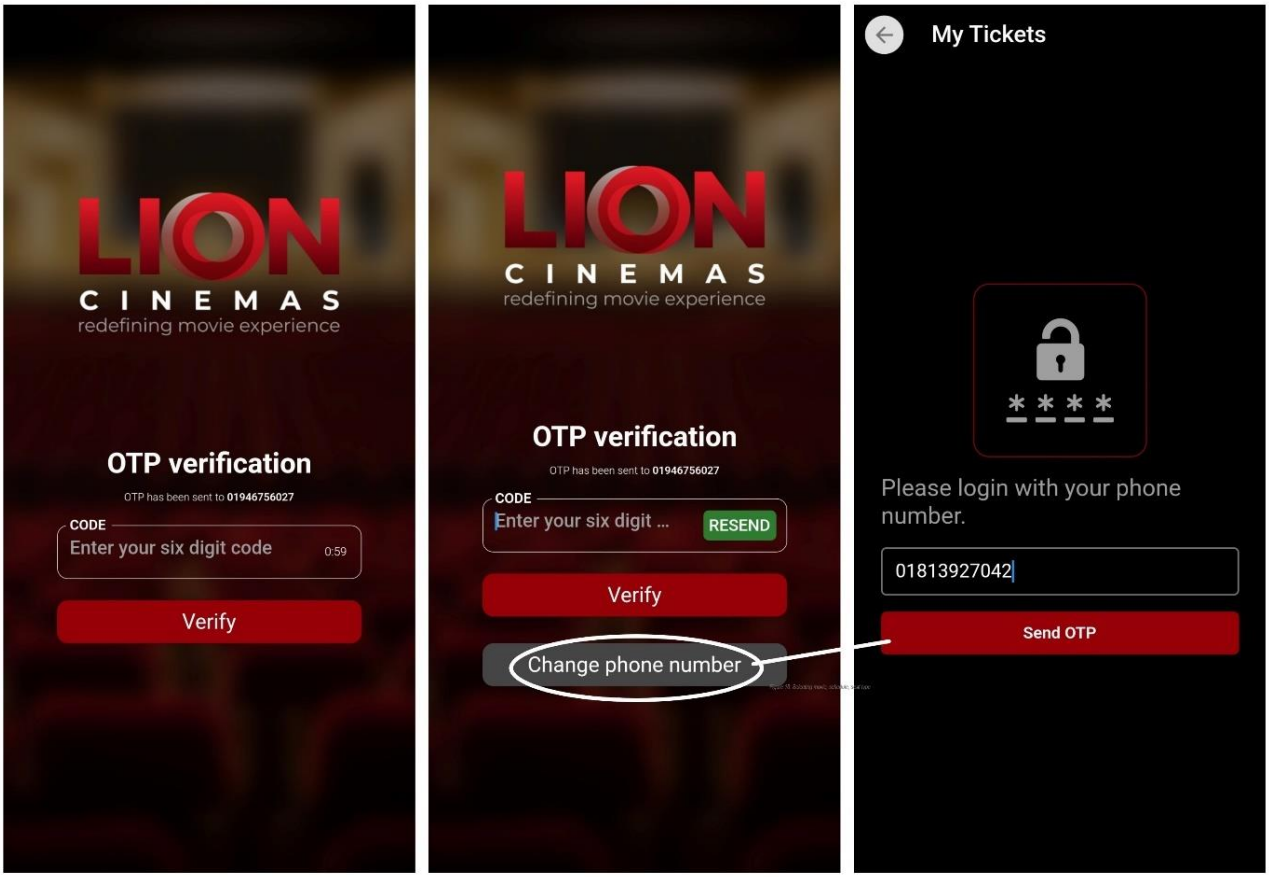


Figure 39: Changing phone number sample

11.7 Security Testing

Test 1: Check OTP

Test Case Name	Acceptance Test		
Test Class	OTP verification controller		
Test Description	Checking OTP validation		
Data Source	Test Steps	Expected Result	Actual Result
	Provide invalid OTP	A warning message should show for providing incorrect OTP	It's working as expected

Table 30: Security test case 1

Figure 40: Checking OTP

Test 2: Checking terms and condition checklist

Test Case Name	Acceptance Test		
Test Class	Details page controller		
Test Description	Checking terms and condition checklist		
Data Source	Test Steps	Expected Result	Actual Result
	Provided mobile number but didn't accept terms and condition on checklist	A warning message should show for accepting terms and condition	It's working as expected

Table 31: Security test case 1

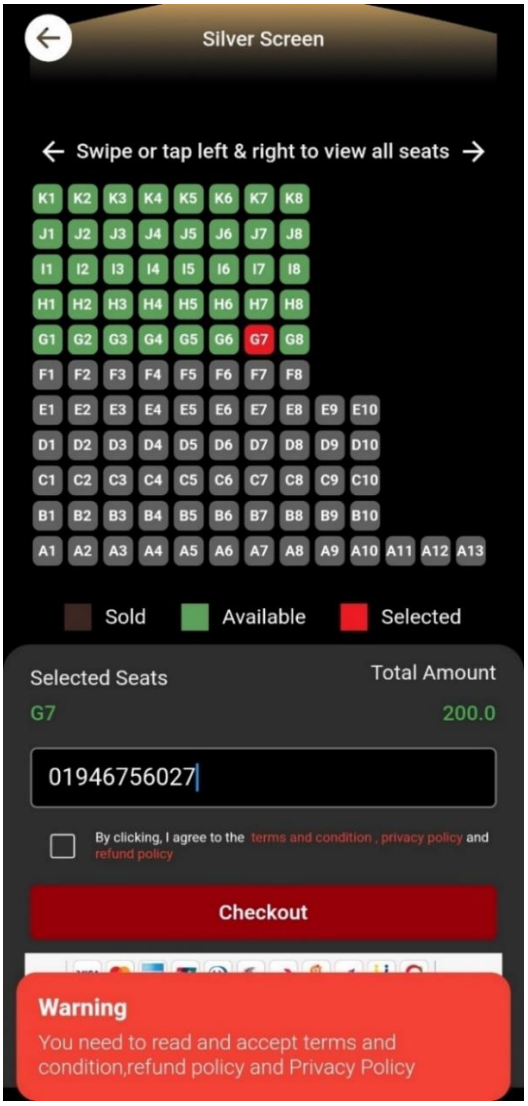


Figure 41: Checking terms and conditions

11.8 Accessibility Testing

Test 1: Testing the application by a color blind user

Test Case Name	Accessibility Test		
Test Class	App pages and routes		
Test Description	Test on color blind people		
Data Source	Test Steps	Expected Result	Actual Result
	Giving a colorblind user to test the application	No problem should be found	It's working as fine for colorblind user

Table 32: Accessibility testing 1

Test 2: Testing the application by an old aged user

Test Case Name	Accessibility Test		
Test Class	App pages and routes		
Test Description	Test on old aged user		
Data Source	Test Steps	Expected Result	Actual Result
	Giving an old aged user to test the application	No problem should be found	It's working as fine for old aged user

Table 33: Accessibility testing 2

CHAPTER 12: IMPLEMENTATION

12.1 Training

Users might become more engaged with the system through training. Everything is planned up in advance. Training is used to guide users. New users will quickly learn how to get around the system after being introduced to it. A chart for training is given below:

SL	User	Training segment	Duration	
01	User	Purchasing a movie ticket (selecting movie, selecting time and date, selecting seat type, selecting seat, processing to payment)	30 minutes	The user understands the purchasing procedure easily
		View purchased tickets	10 minutes	The user understands the process
02	Admin	Manage the whole system (add a new movie, delete a movie, manage schedule, manage theater details)	40 minutes	Admin understands the process easily

Table 34: User training

12.1 Implementation Scheme

Big Bang

The way the Big Bang operates is by quickly activating each new system after turning down the earlier one. This approach is far faster compared to any other. Following testing, this scheme immediately places the new system into use. Data loss during the manual process could cause harm and difficulty. In light of a single site's minimum efforts, it is integrated there.

12.3 Scaling

Since it was unnecessary and not required by the organization, no plans were established that took application scaling into account.

12.4 Load Balancing

To balance the response time of the application, users' needs must be taken into account. User impact refers to how often the application is used and how many users are using it at a time. This process of balancing loads is called load balancing. To keep the system running fast, it distributes the load among other servers.

CHAPTER 13: CRITICAL APPRAISAL AND EVALUATION

13.1 Objectives That could be met

The following list contains the initially stated aims:

- Easy movie ticket purchasing process
- View purchased tickets
- Updated movie list

Objective 1:

Users can easily select a movie, movie schedule, seat, payment process, etc. which ensures that the purchase process is completed successfully. The **Sslcommerz** payment gateway has been used for this project. Here the user is verified by verifying the mobile number which ensures the user validation process has been successfully met.

Objective 2:

Users can easily view their purchased tickets by just logging in with their mobile number. Mobile number verification is also done here to verify the validity of the user. By logging in the user can view their purchased tickets otherwise they can not view them but still, the tickets will be stored in the database.

Objective 3:

Movies list is updated daily. Users will get to view the latest released movies and the latest upcoming movies. Users can easily view movie details to decide which movie ticket to buy.

Due to the project's rushed completion, several features and functions are not being implemented. However, the main functionalities have been developed properly.

14.1 Objectives That Are Not Touched

The project’s conclusion is well delivered as activity work. However, the best portable activities for following have a specific objective that isn't disclosed which is to log in with email via email verification.

14.2 Why It Could Not Be Touched

Although the organization had expected this, they had also set a deadline for finishing the entire project. Additionally, they mention that it is not necessary if it cannot be completed in time. Email verification could not be carried out in the app since there were too many necessary activities to complete.

CHAPTER 14: LESSONS LEARNED

14.1 Pre-project- Review- Closing

To develop the **Lion Cinema** mobile app, I had to complete the project proposal, which details the actual functionality of the app. The approach mostly focuses on booking movie tickets through mobile applications, where users can buy tickets as they wish. I have also completed extensive documentation of the app.

14.2 What I Have Learned

I have used a ton of important techniques for developing a structure. I've learned how to work with a structure in a variety of areas. Importance of **get cli** package while developing in flutter. To help me in my future career, I also enhanced my project planning skills and testing skills. I've learned a ton of testing methods that I might apply to my future projects. The first time I learned how to use API in a mobile application and learned the flexibility of using API. Learned how to implement a payment gateway in a mobile application. The project helped me to acquire a significant amount of information that is genuinely necessary for my professional career.

14.3 Problems That I Have Faced

Numerous challenges I faced throughout my development lifetime. The project is based on Flutter but I have never developed an application using Dart language before because I have knowledge of Java language development. So, it was a challenging part for me. I had to deal with a variety of issues during the complete development life cycle if that makes sense. Setting the time frame was a major challenge because a complete system cannot be developed in such a short time. Another issue was choosing a methodology for the application's development because it can be difficult to follow a certain approach for this project. I was also unfamiliar with working with API. So, it was also a challenging part for me but gained knowledge from it.

14.4 Solutions That Occurred

When I encountered a challenge, I developed a solution to that challenge as well. I chose a DSDM methodology to create the application as a whole because it would make the process much easier. I've encountered many new things, but I've found solutions on Google and learned from You tube. For example, I had no idea about how to use **get cli** package but I learned it. Used SSL commerce payment gateway to make payments through the app. I had to carefully follow the time box timetable to create the application's extensive tasks.

CHAPTER 15: CONCLUSION

15.1 Summary of the Project

For consumers to buy movie tickets whenever they want, **Lion Cinema** is a mobile application created in flutter utilizing the dart language for both android and IOS mobile devices. By installing this application, users may watch upcoming films, get a list of the movies that are now playing in theaters, and conveniently plan their schedules to fit their preferred times. People can also read the information on the film, including its genre, trailer, title, and any available showtimes and dates. Users can make payments through debit/credit cards, mobile banking, or net banking easily.

15.2 Goal of the Project

The project's goal is to develop innovative solutions that meet end-user requirements, hence improving the system's accuracy, speed, and digitalization. To achieve this:

- Make it easier for users to choose a movie and simply book tickets and make payments through their mobile phone using the internet whenever they want.
- To cut back on the number of workers at the ticket counter.
- To provide advice and recommendations for the best theater in the neighborhood.
- Users can quickly check current movie details (such as genre, trailer, and movie description) and make snap judgments to provide a better user experience.

15.3 Success of the Project

The achievement of the goals concerning the objective validates the success of the project development objective. Users can select a movie, can select a seat, and seat type, and can book tickets from the app easily. Users can choose the payment option and make the payment that suits them. The project is successful since the application satisfies all of the initial requirements.

15.4 What I Have Done in the Documentation

From the beginning, I completed all of the tasks that the documentation's content required. I've completed the time-box, many diagrams, testing, etc. Numerous more

plans were made, and they are shown and explained in this documentation. Essentially, this document contains all the necessary elements needed to finish a project.

15.5 My Experience

In the span of working on this project, I acquired a lot of knowledge. I dealt with a variety of situations that I had, which gave me a wealth of knowledge. I gained valuable experience by learning how to complete all of the project's requirements in a short amount of time and manage the entire project.

APPENDIX A

Module test case 1

Test Case Name	Module Test		
Test Class	Sign out controller		
Test Description	Click on sign out to stay as viewer		
Data Source	Test Steps	Expected Result	Actual Result
	Click on sign out option	User signed out	It's working as expected

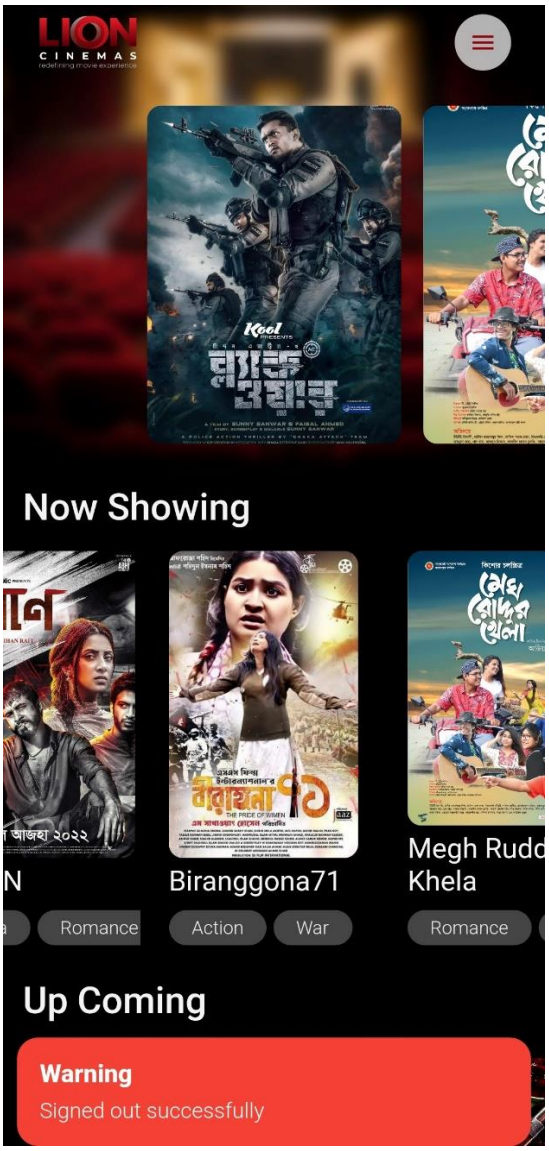


Figure 42: Module test case 1

Module test case 2

Test Case Name	Module Test		
Test Class	Profile controller		
Test Description	Login with phone number to view profile		
Data Source	Test Steps	Expected Result	Actual Result
	Enter phone number to login	Showing user profile	It's working as expected

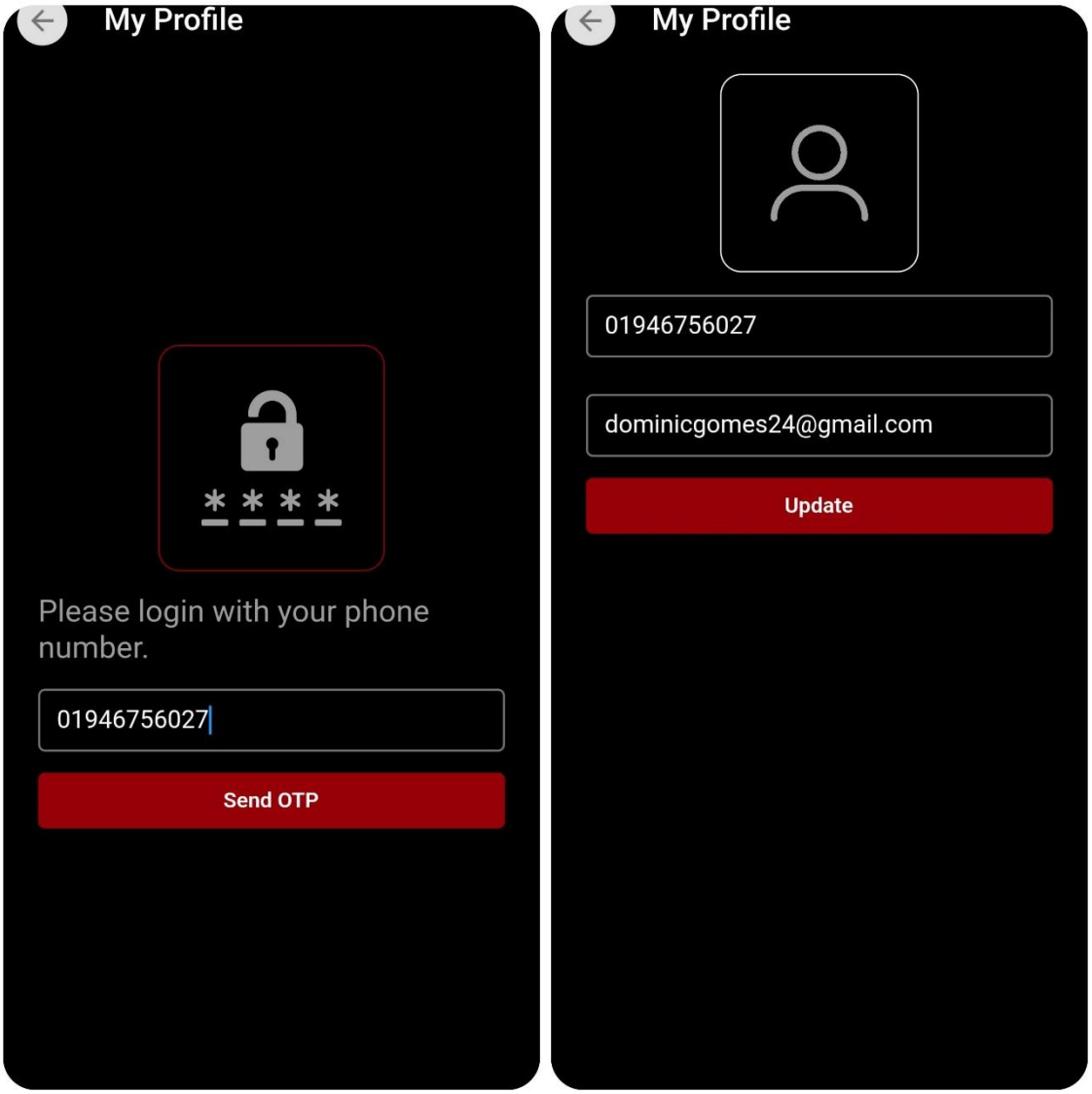


Figure 43: Module test case 2

USED REFERENCES

5 Best Apps For Booking Movie Tickets Online in India. (n.d.). Retrieved January 10, 2023, from https://mygadgetreviewer.com/apps-for-booking-movie-tickets/

34 Pros & Cons Of Starting A Movie Ticket Booking App Business (2023). (n.d.). Retrieved January 10, 2023, from https://www.starterstory.com/ideas/movie-ticket-booking-app-business/pros-and-cons

Bangladesh Electronic Fund Transfer Network. (n.d.). Retrieved January 15, 2023, from https://www.jb.com.bd/about_us/beftn

Cineplex Web. (n.d.). Retrieved January 10, 2023, from https://www.cineplexbd.com/about-cineplex

Client-server Application - OOSE. (n.d.). Retrieved January 10, 2023, from https://madooei.github.io/cs421_sp20_homepage/client-server-app/

Cloud-based integration - Wikipedia. (n.d.). Retrieved January 10, 2023, from https://en.wikipedia.org/wiki/Cloud-based_integration#cite_note-GartnerGlossary-2

Difference Between Web server and Application server - GeeksforGeeks. (n.d.). Retrieved January 10, 2023, from https://www.geeksforgeeks.org/difference-between-web-server-and-application-server/

Different Types Of Mobile App Testing | LambdaTest. (n.d.). Retrieved January 10, 2023, from https://www.lambdatest.com/blog/different-types-of-mobile-app-testing/

Dynamic Systems Development Method (DSDM) - GeeksforGeeks. (n.d.). Retrieved January 10, 2023, from https://www.geeksforgeeks.org/dynamic-systems-development-method-dsdm/

Quality Assurance vs Quality Control: Definitions & Differences | ASQ. (n.d.). Retrieved January 10, 2023, from https://asq.org/quality-resources/quality-assurance-vs-control

SDLC - Waterfall Model. (n.d.). Retrieved January 10, 2023, from https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm

Sheikh Parvez Mahamud, B. (2019). *Auction Management System*.

Software Engineering | Prototyping Model - GeeksforGeeks. (n.d.). Retrieved January 10, 2023, from https://www.geeksforgeeks.org/software-engineering-prototyping-model/

Software Engineering | Risk Management - GeeksforGeeks. (n.d.). Retrieved January 10, 2023, from https://www.geeksforgeeks.org/software-engineering-risk-management/

Unit Testing Tutorial – What is, Types & Test Example. (n.d.). Retrieved January 10, 2023, from https://www.guru99.com/unit-testing-guide.html

PLAGIARISM REPORT

dominic_2

ORIGINALITY REPORT

13%	13%	2%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	dspace.daffodilvarsity.edu.bd:8080 Internet Source	13%
2	Govindharaj Yoganandham. "Making Decisions for the Future in a Globalized Society with Erratic Circumstances, Collective Action and Transformational Paths to Enhance Human Development: An Assessment", Journal of Human Resource and Sustainability Studies, 2022 Publication	<1%
3	ir.unimas.my Internet Source	<1%
4	deepblue.lib.umich.edu Internet Source	<1%
5	shop.1300intech.com.au Internet Source	<1%
6	www.abor.asu.edu Internet Source	<1%
7	pdxscholar.library.pdx.edu Internet Source	<1%

8	scholar.uc.edu Internet Source	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		