

**Veer Narmad South Gujarat University,
Surat.**

**Department of Information and
Communication Technology**

B.Sc. (Information Technology)

Project Report

6th Semester

**M.Sc. (Information Technology)
5 Year Integrated Course**

Year 2023 – 2024

TheCinemaCub

Guided By:

Dr.Payal Joshi

Submitted By:

Arin Dhimar
SPID:2021013488
Bansi Bhimani
SPID:2021013472
Pratigna Savaliya
SPID:2021013596



Re-Accredited 'B++' 2.86 CGPA by NAAC

VEER NARMAD SOUTH GUJARAT UNIVERSITY

University Campus, Udhna-Magdalla Road, SURAT - 395 007, Gujarat, India.

વીર નર્મદ દક્ષિણ ગુજરાત યુનિવર્સિટી

યુનિવર્સિટી કેમ્પસ, ઉધના-મગદલ્લા રોડ, સુરત - ૩૯૫ ૦૦૭, ગુજરાત, ભારત.

Tel : +91 - 261 - 2227141 to 2227146, Toll Free : 1800 2333 011, Digital Helpline No.- 0261 2388888

E-mail : info@vnsgu.ac.in, Website : www.vnsgu.ac.in

Department of Information and Communication Technology

M.Sc.(Information Technology) Programme

Certificate

This is to certify that Mr./Ms. Dhimar Arin Avinash with exam seat number 120 and SPID 2021013488 has worked on his/her project work entitled as TheCinemaCub at **Department of ICT** as a partial fulfillment of requirement of B.Sc. (Information Technology) - 6th Semester, during the academic year 2023-2024.

A P P

Date: 18/6/2024

Place: Department of ICT, VNSGU, Surat

Internal Project Guide
BSc(IT) 6th Semester
Department of ICT
VNSGU, Surat

Head of the Department
Department of ICT
VNSGU, Surat

Index

Sr. No	Topics	Page no
1	Introduction	
	1.1 Project Profile	5
2	Proposed System	
	2.1 Scope	6
	2.2 Objective	7
	2.3 Advantages	8
	2.4 Future Enhancements and Potential Additions	10
3	Environment Specification	
	3.1 Hardware & Software Requirements	12
4	System Planning	
	4.1 Feasibility Study	15
	4.2 Software Engineering Model	17
	4.3 Risk Analysis	19
	4.4 Project Schedule	22
	4.4.1 Task Dependency	22
	4.4.2 Timeline Chart	23

5	System Analysis	24
	5.1 Detailed SRS	24
	5.2 UML Diagram	28
	5.2.1 Use Case Diagram	28
	5.2.2 Class Diagram	32
	5.2.3 Activity Diagram	33
	5.2.4 Sequence Diagram	37
	5.3 E-R Diagram	41
	5.4 DFD	42
6	Software Design	47
	6.1 Database Design	47
	6.2 Interface Design	51
7	Glossary	59
8	Reference	63

1) Project Profile

Project Title – TheCinemaCub

In-House Project Front-End Tool – Kotlin

Back-End Tool – Firebase (Authentication, Real-Time
Database, Storage)

Project Category - Mobile Application Development

Documentation Tool – Microsoft Word

Editor – Android Studio

Duration – 20 Weeks

2) Proposed System

2.1) Scope

The scope of "TheCinemaCub" project involves creating a versatile mobile application that caters to different roles within the cinema industry. This includes:

- **Application Owners:** Managing cinema owners, movies, and accessing feedback and ratings.
- **Cinema Owners:** Overseeing their cinemas, managing staff, leasing movies, and viewing booking statistics.
- **Cinema Admins:** Handling show schedules, bookings, and monitoring cinema ratings.
- **Normal Users:** Viewing current shows in their city, providing feedback, and managing their bookings.

The project focuses on delivering a user-friendly interface, ensuring data security, providing robust backend support, and integrating real-time updates to streamline operations and enhance user experience.

2.2) Objective

- **Develop a comprehensive mobile application** that supports multiple user roles within the cinema industry.
- **Provide an intuitive user interface** for seamless navigation and interaction.
- **Enable application owners** to manage cinema owners, movies, and access user feedback and ratings.
- **Facilitate cinema owners** in managing their cinemas, staff, and leasing movies while tracking bookings.
- **Empower cinema admins** to efficiently manage show schedules, bookings, and monitor ratings.
- **Offer normal users** easy access to current shows in their city, feedback submission, and booking management.
- **Ensure data security and privacy** for all users through robust security measures.
- **Integrate real-time updates** to keep information current and accurate for all stakeholders.

2.3) Advantages

- **Enhanced Visibility for Rural Cinemas:** The application allows rural cinemas to list their shows and manage bookings, increasing their visibility and attracting more patrons.
- **User-Friendly Interface:** Designed with ease of use in mind, the app simplifies the process for both cinema owners and users, making it accessible even to those with limited technical skills.
- **Real-Time Updates:** Users and cinema owners receive real-time updates on show availability, bookings, and feedback, ensuring accurate and timely information.
- **Efficient Management:** The app streamlines the management of cinemas, cinema admins, and shows, reducing the administrative burden and operational inefficiencies.
- **Integrated Feedback System:** Cinema owners can view user feedback and ratings, allowing them to improve their services based on customer insights.
- **Secure Authentication:** Utilizing Firebase Authentication, the app ensures secure login processes, protecting user data and providing reliable access control.
- **Scalable Solution:** Built using Firebase, the system can scale efficiently as the number of users and data grows, accommodating more cinemas and users without performance degradation.

- **Cost-Effective:** By offering a mobile-based solution, the app reduces the need for expensive infrastructure and manual processes, making it a cost-effective option for rural cinema owners.
- **Convenient Booking for Users:** Users can easily browse shows, select seats, and book tickets from the comfort of their homes, enhancing the overall user experience.
- **Accessibility:** The mobile application can be accessed from anywhere, making it convenient for users in rural areas to book tickets and for cinema owners to manage their operations remotely.

The
C I N E M A C U B

A P P

2.4) Future Enhancements and Potential Additions

- **Offline Access:** Implementing offline access functionality would allow users to view show schedules and make bookings even when they don't have an internet connection. This would be particularly beneficial in rural areas with limited connectivity.
- **Multiple Payment Options:** While the system currently supports payment processing, offering multiple payment options such as mobile wallets, UPI, or cash on delivery could cater to a wider range of users with varying preferences.
- **Customizable Cinema Pages:** Allowing cinema owners to customize their cinema pages with additional information, images, and branding elements would help them showcase their unique offerings and attract more customers.
- **Advanced Analytics:** Enhancing the analytics capabilities of the system to include more detailed insights, such as demographic data of users, peak booking times, and popular movie genres, would provide valuable information for strategic decision-making and marketing campaigns.
- **Integration with Social Media:** Integrating social media sharing and login features would enable users to share their booking experiences with friends and family, while also simplifying the login process by leveraging existing social media accounts.

- **In-App Messaging:** Adding an in-app messaging system would facilitate communication between cinema owners and users, allowing for real-time updates, special offers, and personalized notifications.
- **Rating System for Cinema Facilities:** In addition to movie ratings, implementing a rating system for cinema facilities (e.g., cleanliness, seating comfort, concessions) would provide valuable feedback to cinema owners for improving the overall customer experience.
- **Localization Support:** Providing support for multiple languages and localized content would enhance accessibility for users in diverse regions, ensuring a seamless experience for non-English speaking audiences.
- **Multiple Screens Management:** Implementing functionality to manage multiple screens within a single cinema would enable cinema owners to schedule and track shows across different screens, optimizing the use of their facilities and providing more viewing options for customers.
- **Mobile App Notifications:** Adding push notifications for mobile apps to alert users about upcoming shows, special offers, and booking confirmations would keep users engaged and informed in real-time.
- **Desktop Application or Website for Admins:** Developing a desktop application or a website for cinema owners, application owners, and cinema admins would streamline administrative tasks and management functionalities, providing a robust platform for handling operations, analytics, and customizations.

. 3) Environment Specification

3.1 Hardware & Software Requirements

Hardware Requirements:

Development Machines

Name	Specification
Processor	Quad Core Or Higher
RAM	16 GB or Higher
OS	Windows 10 Or Higher / Linux
SDD	30GB Space or Higher

Testing Machines

Name	Specification
Processor	Quad Core Or Higher
RAM	1 GB or Higher
OS	Android 7.2 or Higher
Storage	100 MB or higher

Software Requirements:

Name	Specification
Database/Authentication/Cloud Storage	Firebase
Development Tools	Android Studio
Documentation Tools	MS Word
Drawing Tool	Draw.io , PlantUML

The
C I N E M A C U B

A P P

4) System Planning

Problem Statement:

In rural areas, cinema halls often face challenges in meeting the requirements of popular booking platforms like BookMyShow, resulting in limited visibility, reduced patronage, and operational inefficiencies. These platforms typically cater to urban-centric theatres with advanced technological infrastructure, leaving rural cinemas underserved and struggling to compete. To bridge this gap, there is a need for a user-friendly, cost-effective mobile application that allows rural cinema halls to manage their operations, attract more patrons, and integrate seamlessly with modern booking systems.

A P P

4.1) Feasibility Study

The feasibility of the Cinema Management System depends on several factors, including technical, economic, operational, and schedule feasibility. Let's explore each of these aspects:

Technical Feasibility:

- This involves assessing whether the required technology and infrastructure are available and capable of supporting the system.
- Considerations include hardware requirements, software development, and integration capabilities with existing systems, data security, and scalability.
- It is important to ensure that the system can be effectively implemented and maintained within the existing technological framework.

Economic Feasibility:

- This involves evaluating the cost-effectiveness of implementing the system.
- It includes analysing the costs associated with software development or acquisition, hardware requirements, training, maintenance, and ongoing support.
- The benefits derived from the system, such as increased efficiency, improved customer satisfaction, reduced operational costs, and optimized resource allocation, should outweigh the costs over a reasonable period.

Operational Feasibility:

- This assesses whether the system can be easily integrated into the existing operational processes and workflows.
- It considers factors such as the ease of use for different types of users (Application Owners, Cinema Owners, Cinema Admins, and Normal Users), compatibility with existing work practices, and the impact on daily operations.
- The system should enhance operational efficiency, simplify tasks, and improve overall productivity without causing significant disruptions.

C I N E M A C U B

A P P

4.2) Software Engineering Model

Incremental Model:

The Incremental Model was chosen for the development of the Cinema Management System due to its flexibility and iterative nature. The benefits of using the Incremental Model for this project include early delivery of critical functionality, reduced risks, and the ability to incorporate user feedback during the development process.

Stages of the Incremental Model for Cinema Management System:

1. Initial Planning and Requirements Gathering:

- Conducted initial meetings with stakeholders (Application Owners, Cinema Owners, Cinema Admins, and Normal Users) to gather requirements.
- Identified core functionalities and prioritized them for the first increment.

2. Increment 1: Basic Authentication and User Management:

- Implemented user authentication (email/password and Google login).
- Set up Firebase for authentication and database management.
- Allowed Application Owners to manage Cinema Owners.
- Deployed and tested the increment to gather feedback.

3. Increment 2: Movie Management:

- Developed functionality for Application Owners to manage movies (add/view).
- Integrated Firebase Storage for storing movie posters.
- Enhanced the user interface for better usability based on feedback from Increment 1.
- Deployed and tested the increment to gather feedback.

4. Increment 3: Cinema and Admin Management:

- Enabled Cinema Owners to manage cinemas and cinema admins.
- Ensured functionality to check for email existence before adding a cinema admin.
- Integrated Firebase for database updates and storage.
- Deployed and tested the increment to gather feedback.

5. Increment 4: Show and Booking Management:

- Developed functionality for Cinema Admins to manage shows and view bookings.
- Implemented checks to avoid show schedule conflicts.
- Integrated payment gateway for processing bookings.
- Deployed and tested the increment to gather feedback.

6. Increment 5: User Features and Feedback:

- Enabled normal users to view shows, book tickets, view bookings, and provide feedback.

- Integrated Firebase Storage for fetching cinema and movie posters.
- Implemented feedback and rating features for cinemas and movies.
- Deployed and tested the increment to gather feedback.

7. Increment 6: Final Enhancements and Bug Fixes:

- Addressed any remaining bugs and issues from previous increments.
- Improved performance and optimized the application based on user feedback.
- Conducted final testing to ensure a smooth user experience.

By breaking down the project into these increments, we ensured that critical functionality was delivered early and that each increment was thoroughly tested and refined before moving on to the next. This approach allowed us to mitigate risks and incorporate user feedback continuously.

4.3) Risk Analysis

Below are the key risks identified for the Cinema Management System project and the measures taken to address them:

1. Technical Risks:

- **Risk:** Integration issues with Firebase for authentication and database management.
- **Mitigation:** Conducted thorough research and training on Firebase. Implemented and tested integration in early increments to identify and resolve issues early.
- **Risk:** Scalability and performance issues as the number of users grows.
- **Mitigation:** Designed the system architecture with scalability in mind. Performed load testing and optimized database queries.

2. Operational Risks:

- **Risk:** Resistance to change from stakeholders used to existing processes.
- **Mitigation:** Engaged stakeholders throughout the development process. Conducted training sessions and provided detailed documentation.

3. Project Management Risks:

- **Risk:** Delays in delivery due to unforeseen complexities.

- **Mitigation:** Used the Incremental Model to deliver critical functionality early. Maintained a buffer in the schedule for unexpected issues.

4. Security Risks:

- **Risk:** Unauthorized access to user data.
- **Mitigation:** Implemented robust authentication mechanisms using Firebase. Regularly updated security protocols and conducted security audits.

5. User Acceptance Risks:

- **Risk:** Low user adoption due to poor usability or functionality gaps.
- **Mitigation:** Gathered user feedback continuously and incorporated it into each increment. Conducted user testing sessions to ensure the application met user expectations.

6. Data Integrity Risks:

- **Risk:** Data loss or corruption in the Firebase database.
- **Mitigation:** Implemented regular data backups and integrity checks. Ensured transactional operations to maintain data consistency.

4.4) Project Schedule

4.4.1) Task Dependency

Task dependency outlines the relationships between different tasks in the project, indicating which tasks need to be completed before others can begin. For the Cinema Management System project, the tasks and their dependencies are as follows:

1. Initial Planning and Requirements Gathering:

- Dependencies: None

2. Increment 1: Basic Authentication and User Management:

- Dependencies: Initial Planning and Requirements Gathering

3. Increment 2: Movie Management:

- Dependencies: Increment 1

4. Increment 3: Cinema and Admin Management:

- Dependencies: Increment 2

5. Increment 4: Show and Booking Management:

- Dependencies: Increment 3

6. Increment 5: User Features and Feedback:

- Dependencies: Increment 4

7. Increment 6: Final Enhancements and Bug Fixes:

- Dependencies: Increment 5

-

4.4.2)TimeLine Chart

Task	Duration	Start Date	End Date	Dependencies	Responsible Team Member(s)
Initial Planning and Requirements Gathering	2 weeks	Week 1	Week 2	None	Project Manager, Business Analyst
Basic Authentication and User Management	3 weeks	Week 3	Week 5	Initial Planning and Requirements	Backend Developer, UI/UX Designer
Movie Management	3 weeks	Week 6	Week 8	Basic Authentication and User Management	Backend Developer, Frontend Developer
Cinema and Admin Management	3 weeks	Week 9	Week 11	Movie Management	Backend Developer, Frontend Developer
Show and Booking Management	3 weeks	Week 12	Week 14	Cinema and Admin Management	Backend Developer, Frontend Developer
User Features and Feedback	3 weeks	Week 15	Week 17	Show and Booking Management	Backend Developer, Frontend Developer
Final Enhancements and Bug Fixes	3 weeks	Week 18	Week 20	User Features and Feedback	QA Team, Project Manager

5) System Analysis

5.1) SRS (Software Requirement Specification)

Functional Requirements

Application Owners:

1. User Authentication:

- Application owners must be able to log in using email/password or Google authentication.
- Application owners must receive an email verification if their email is not verified.

2. Manage Cinema Owners:

- Add new cinema owners after checking if the email is already in use.
- View the list of registered cinema owners.

3. Manage Movies:

- Add new movies with details such as title, duration, genre, etc.
- Upload movie posters to Firebase Storage.
- View the list of added movies.

4. View Feedback/Ratings:

- Retrieve and display feedback and ratings from users for various cinemas and movies.

Cinema Owners:

1. User Authentication:

- Cinema owners must be able to log in using email/password or Google authentication.
- Cinema owners must receive an email verification if their email is not verified.

2. Manage Cinemas:

- Add new cinemas with details such as name, location, capacity, etc.
- Upload cinema images to Firebase Storage.
- View the list of cinemas owned.

3. Manage Cinema Admins:

- Add new cinema admins by selecting a cinema and ensuring the email is not already in use.
- View the list of cinema admins associated with their cinemas.

4. Rent/Lease Movies:

- View available movies added by the application owner.
- Lease movies based on agreements.
- View ratings of rented/leased movies.

5. View Bookings:

- Retrieve and display booking details for their cinemas.

Cinema Admins:

1. User Authentication:

- Cinema admins must be able to log in using email/password or Google authentication.
- Cinema admins must receive an email verification if their email is not verified.

2. Manage Shows:

- Add new shows by selecting a rented movie, specifying time and date, and ensuring no conflict with existing shows.
- View the list of shows for their assigned cinemas.

3. View Bookings:

- Retrieve and display booking details for their shows.

4. View Cinema Ratings:

- Retrieve and display ratings for their cinemas.

Normal Users:

1. User Authentication:

- Users must be able to log in using email/password or Google authentication.
- Users must receive an email verification if their email is not verified.

2. View Current Shows:

- Select a city to view available shows.
- Retrieve and display cinema and movie posters from Firebase Storage.

3. Book Shows:

- Select a show and view available seats.
- Select seats and proceed with payment.
- Payment gateway integration for processing payments.
- Display booking confirmation and store booking data in Firebase Database.

4. Provide Feedback:

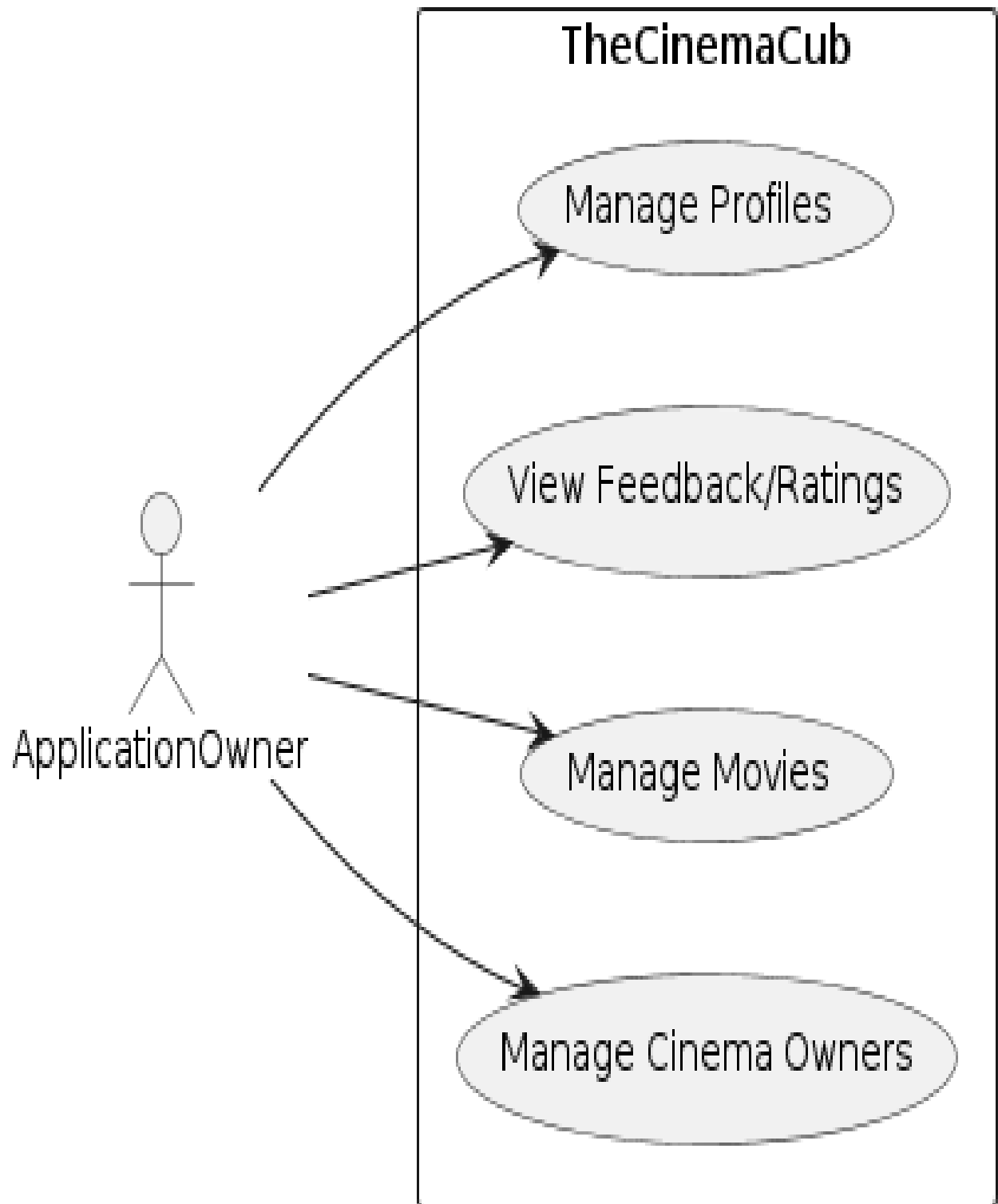
- Submit feedback and ratings for cinemas and movies after attending a show.
- Store feedback and rating data in Firebase Database.

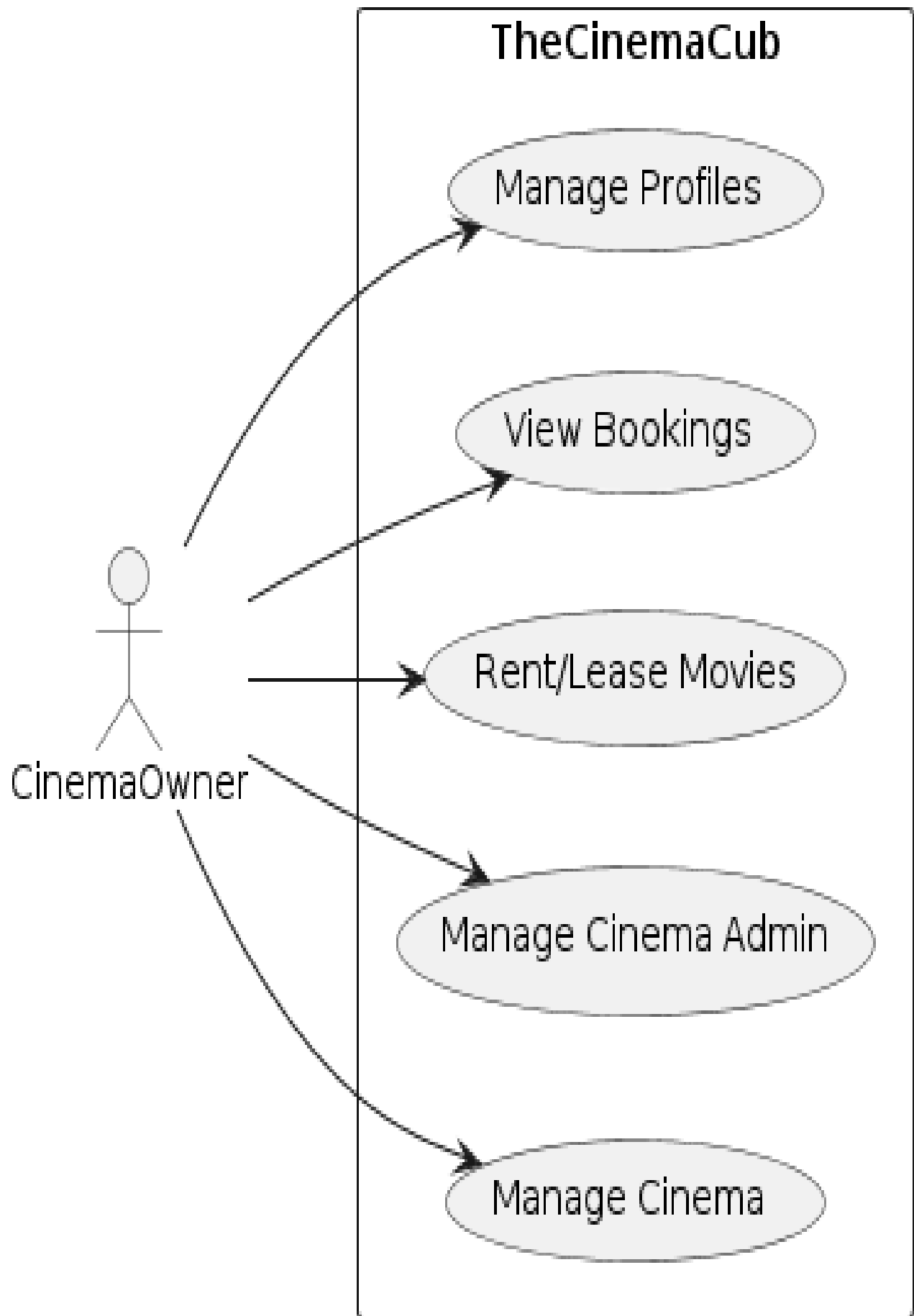
5. View Bookings:

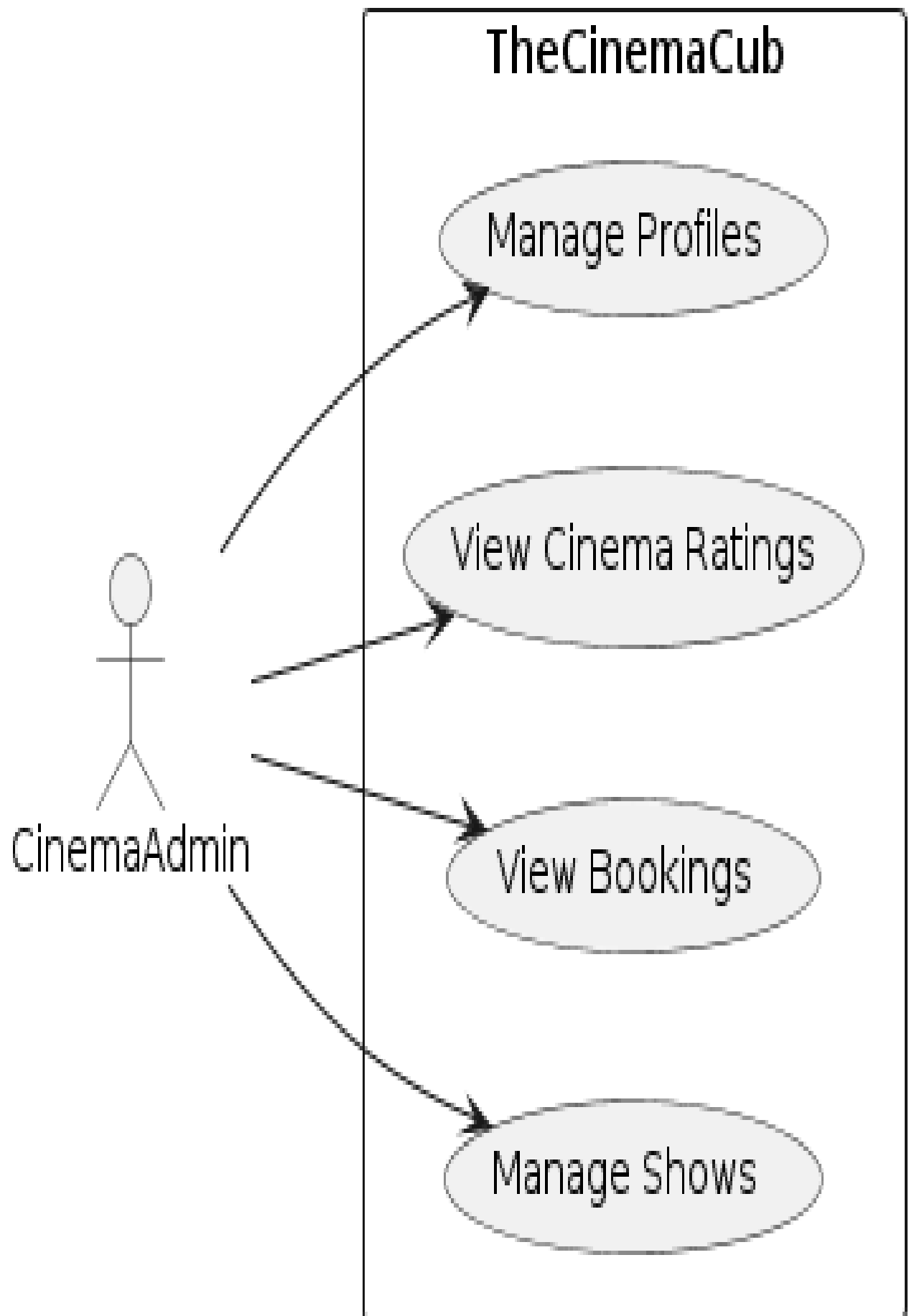
- Retrieve and display a list of their booked shows.

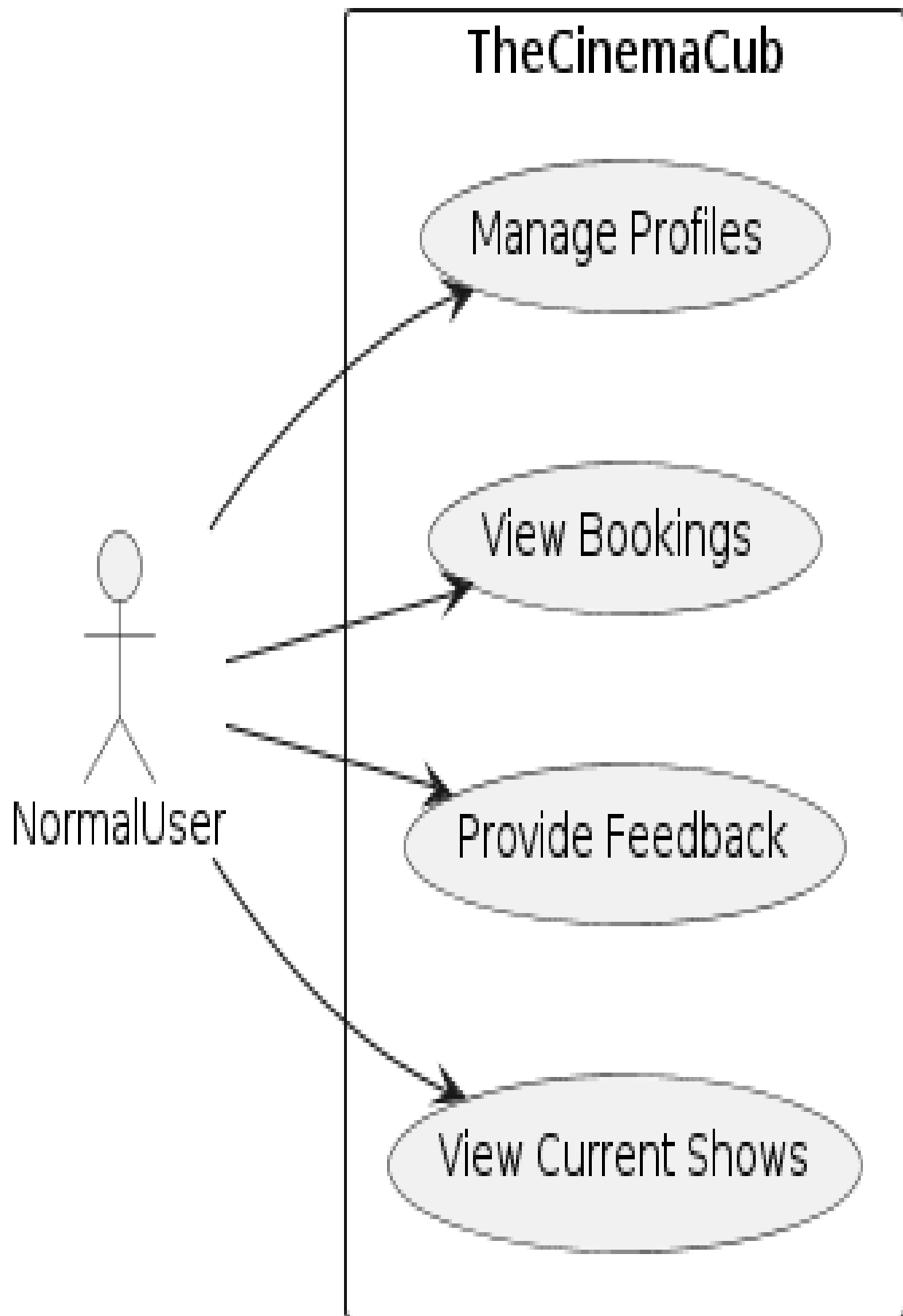
5.2) UML Diagram

5.2.1) Use Case Diagram

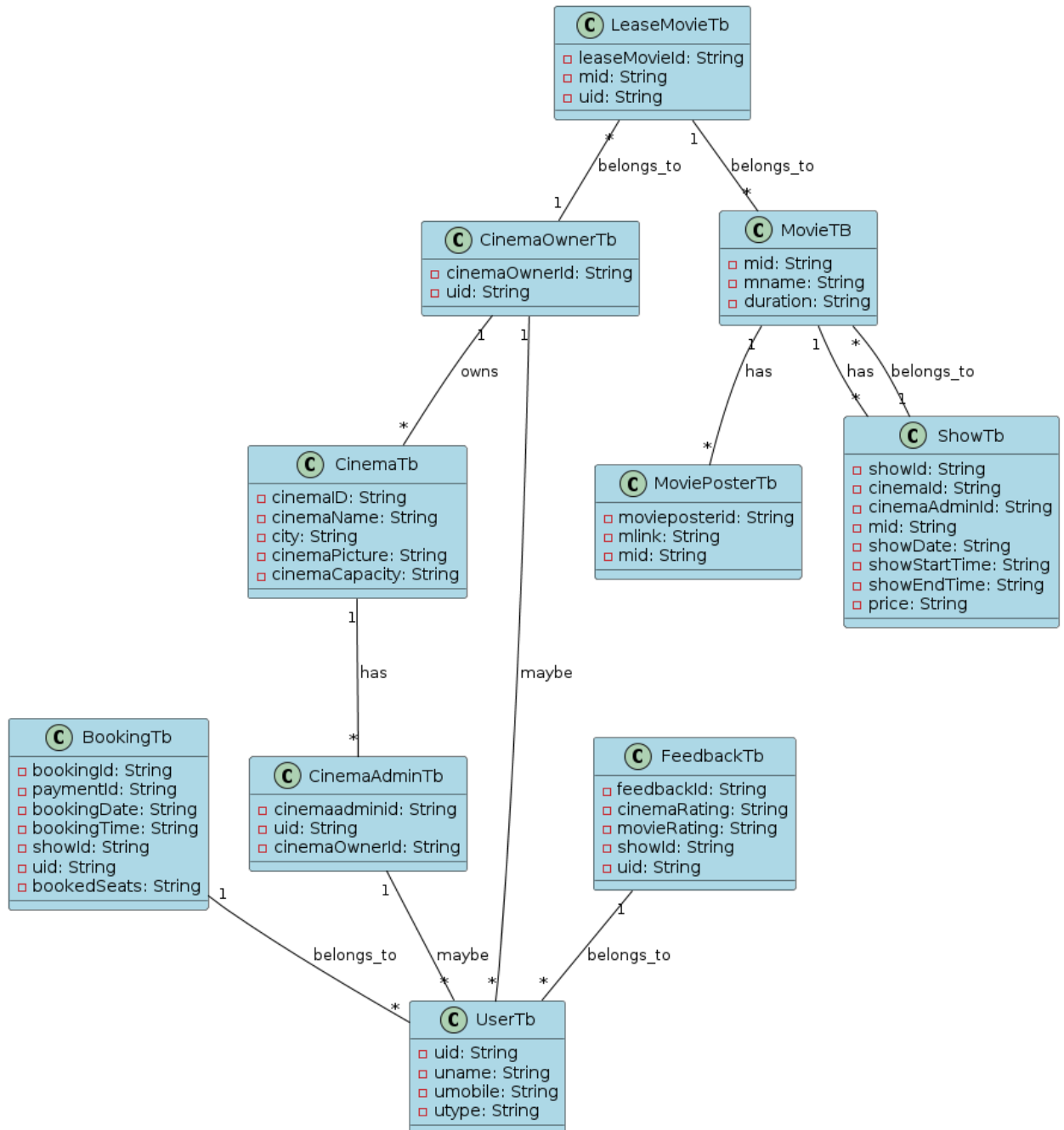




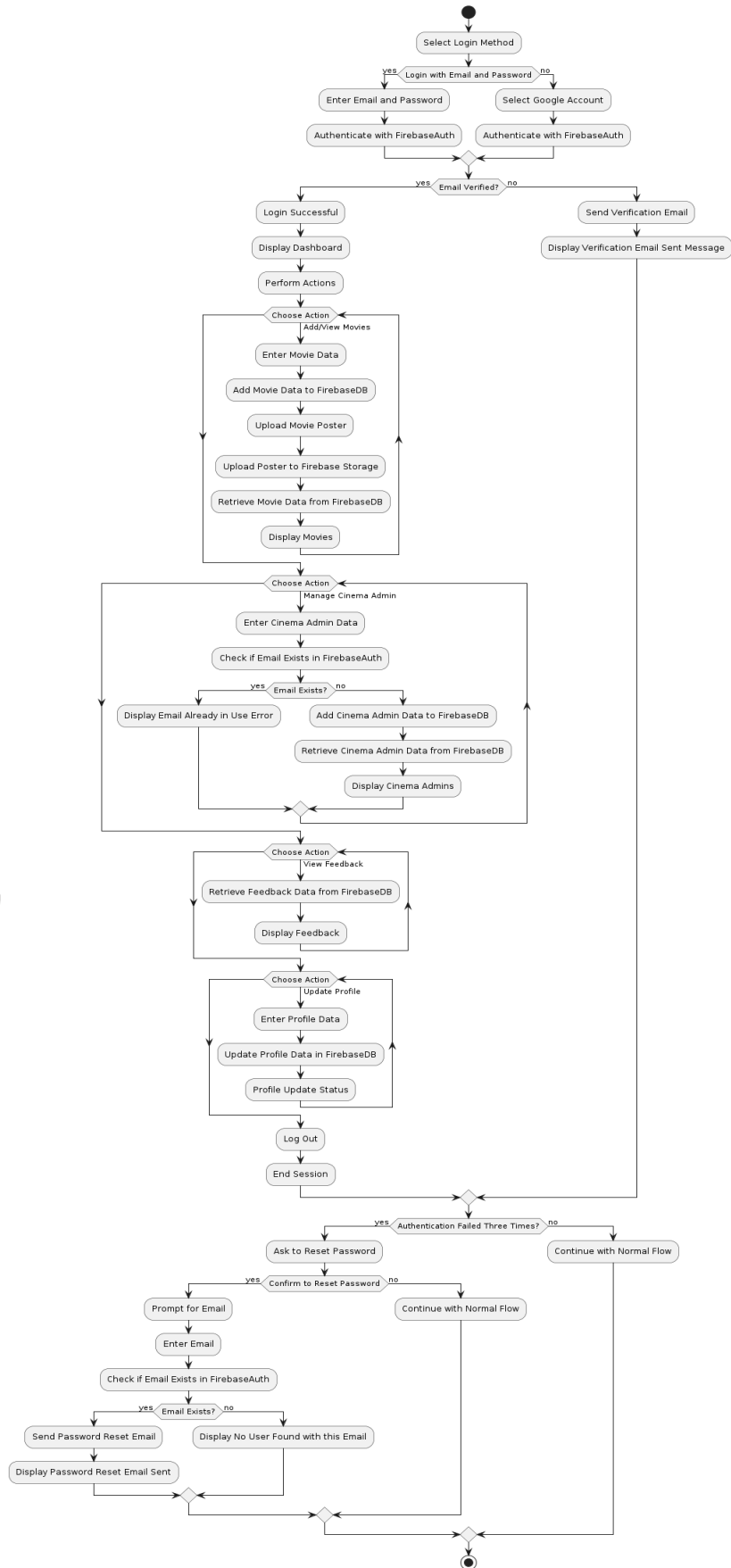


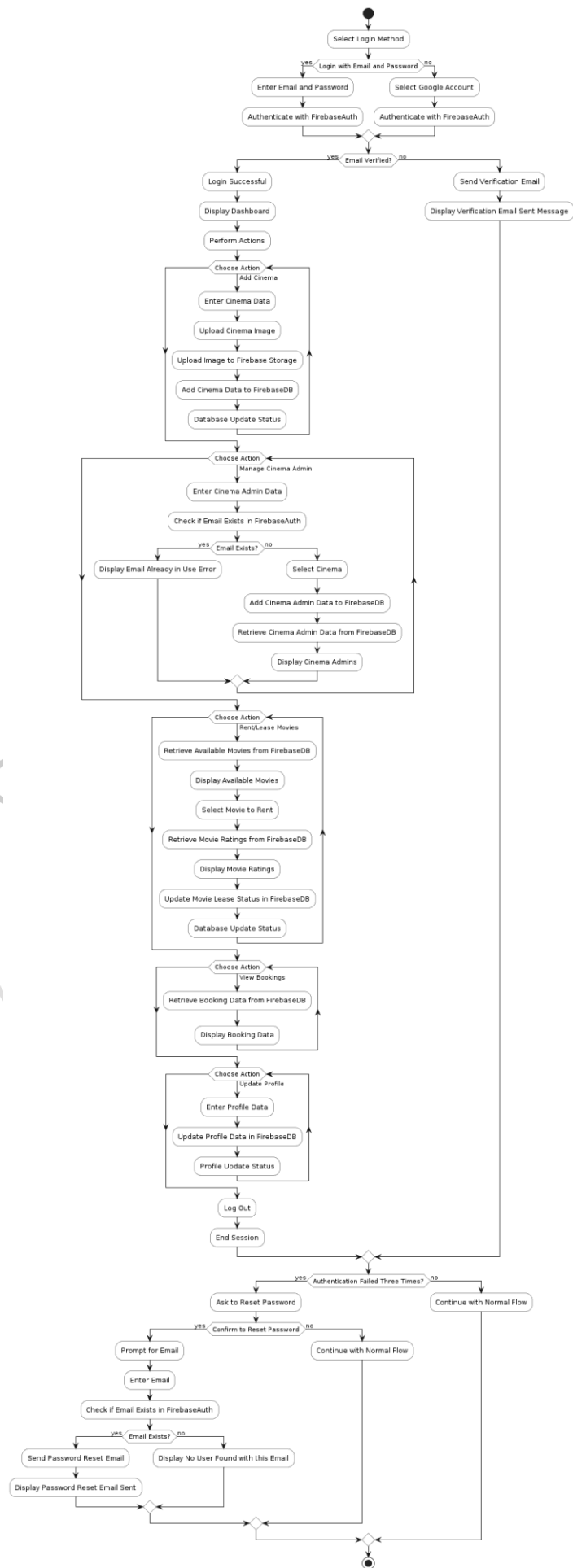


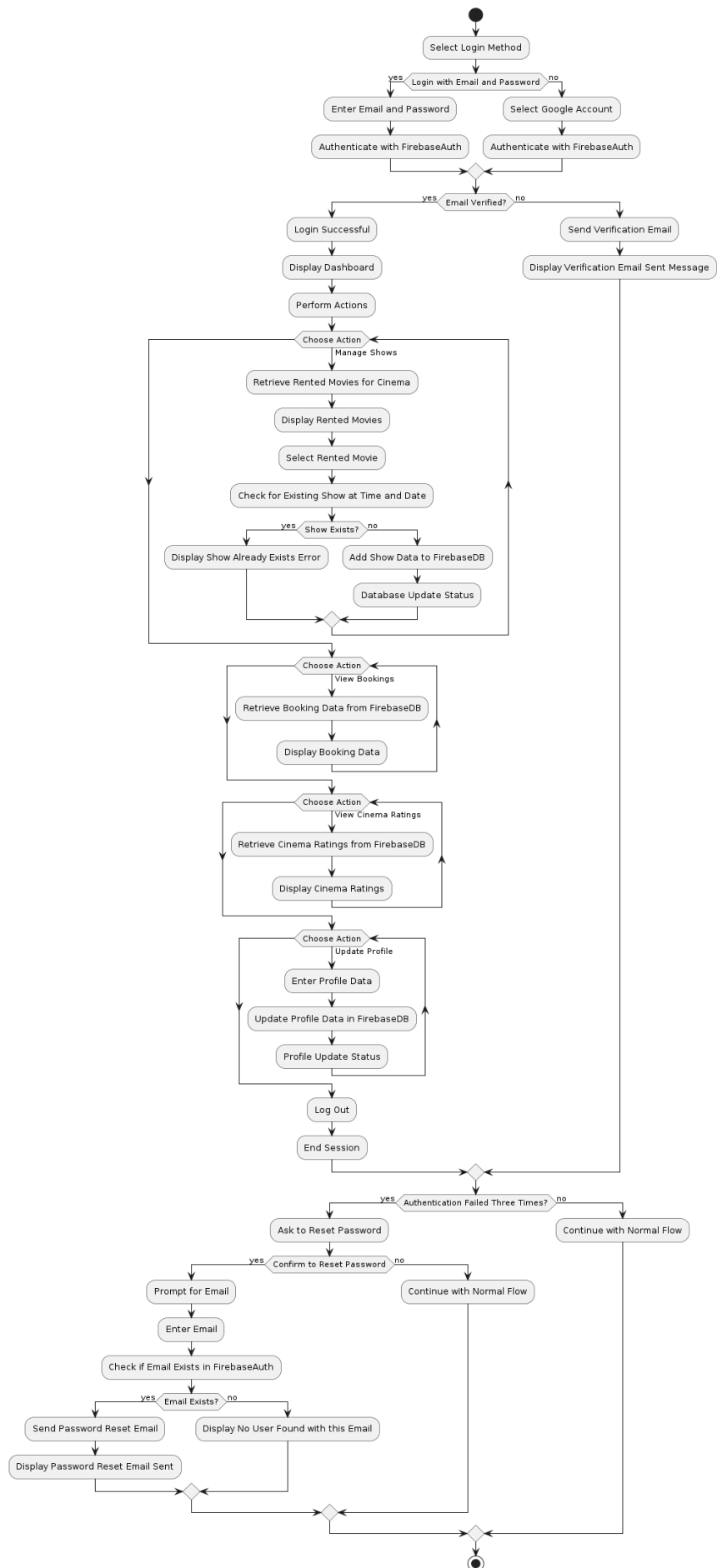
5.2.2) Class Diagram

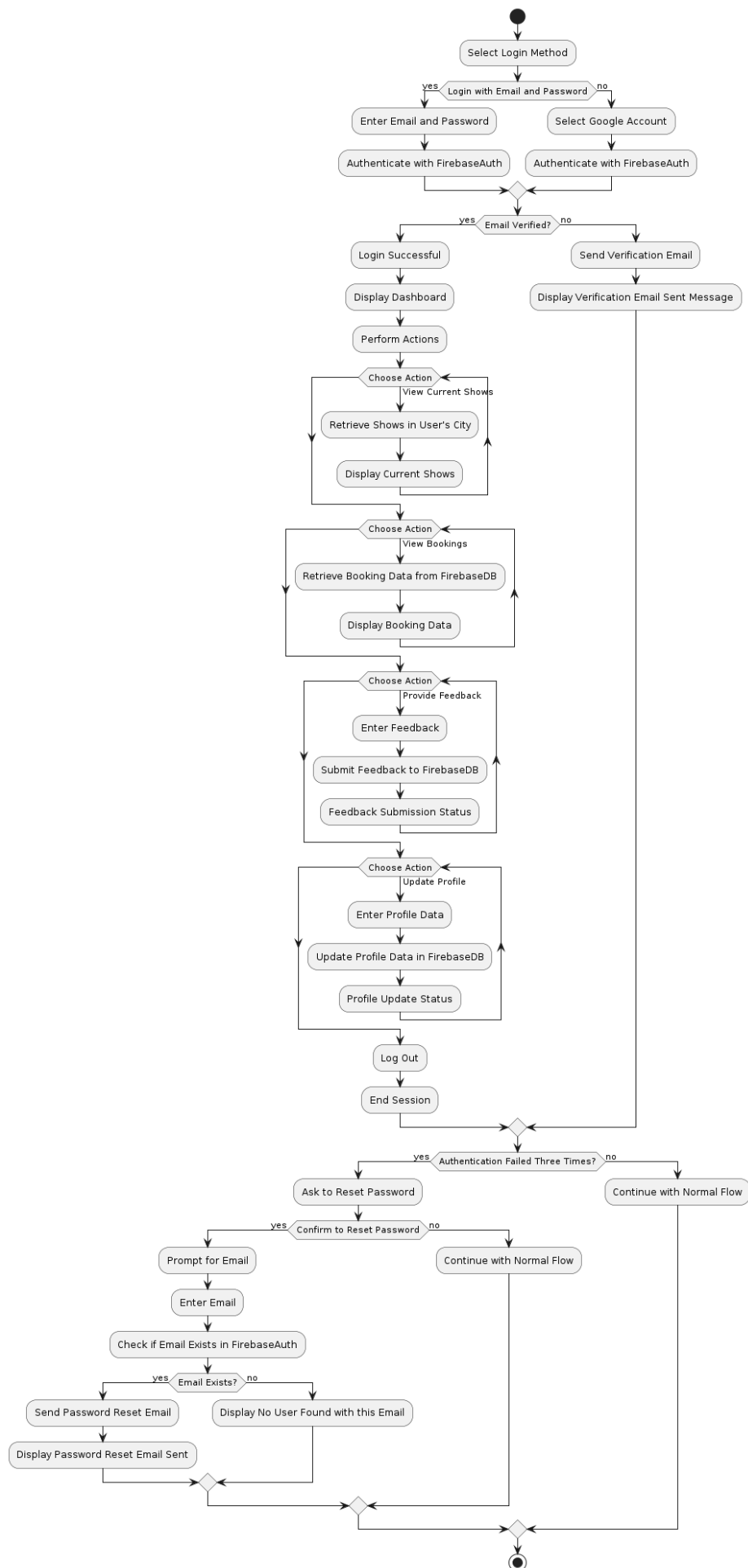


5.2.3) Activity Diagram

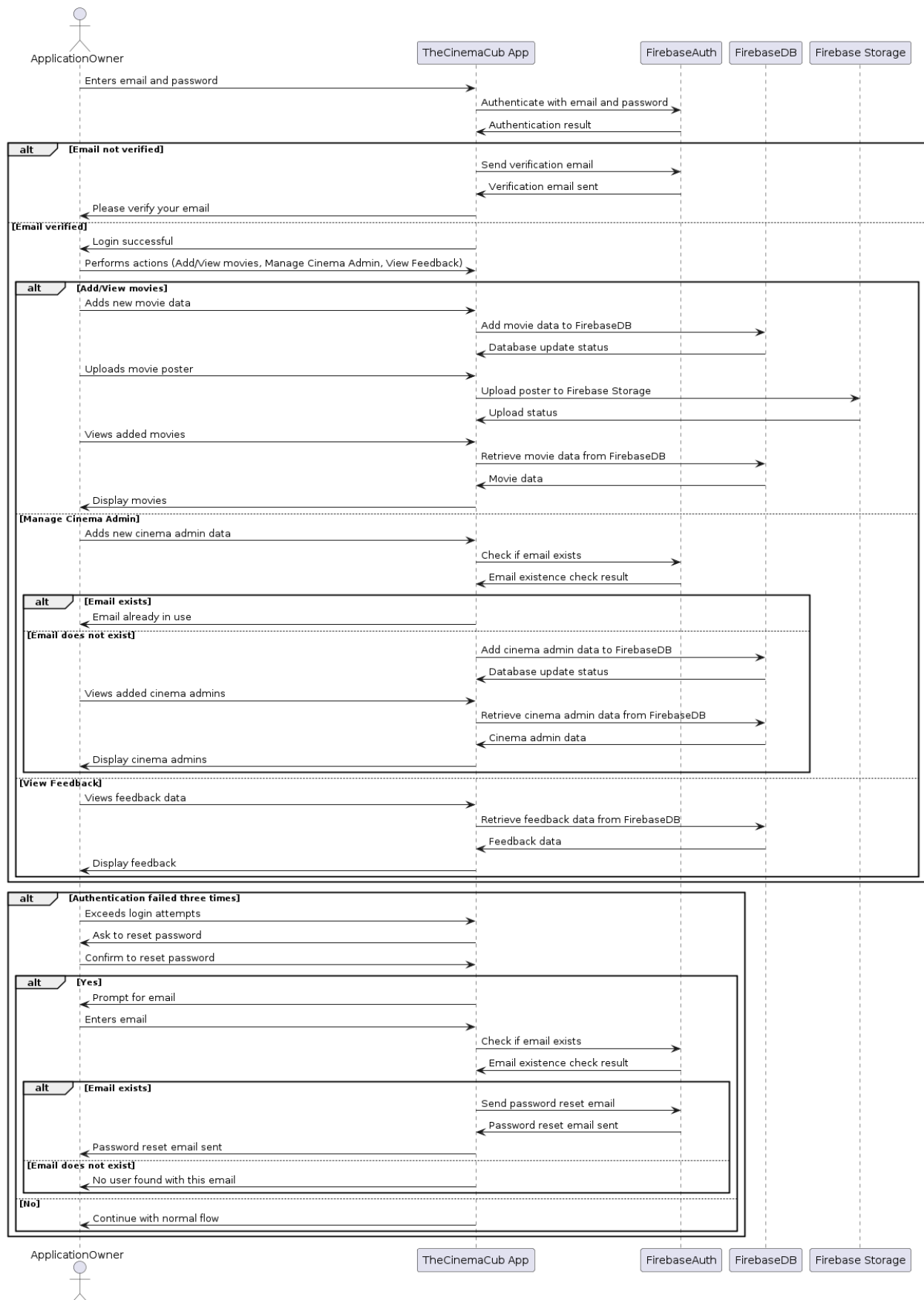


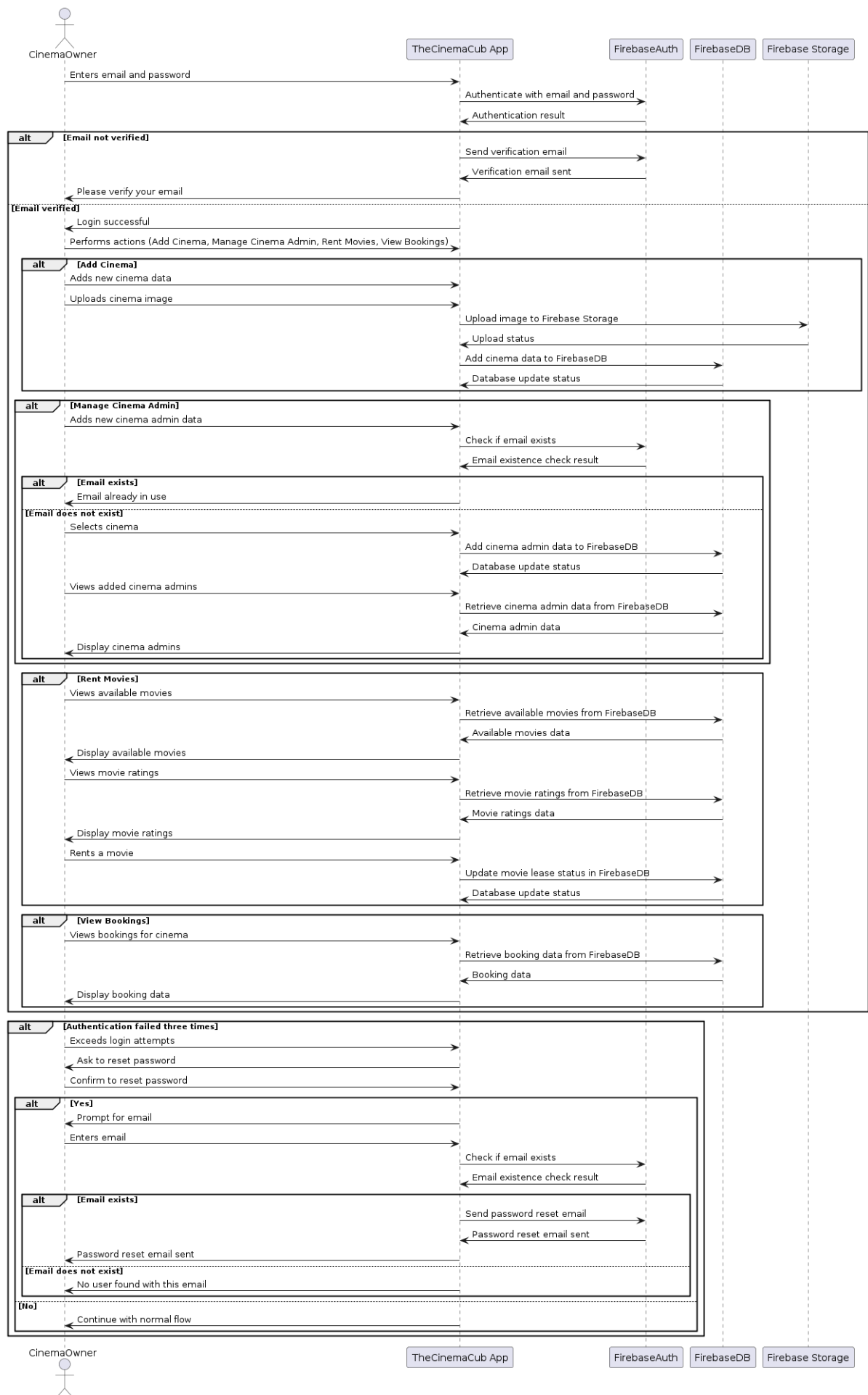


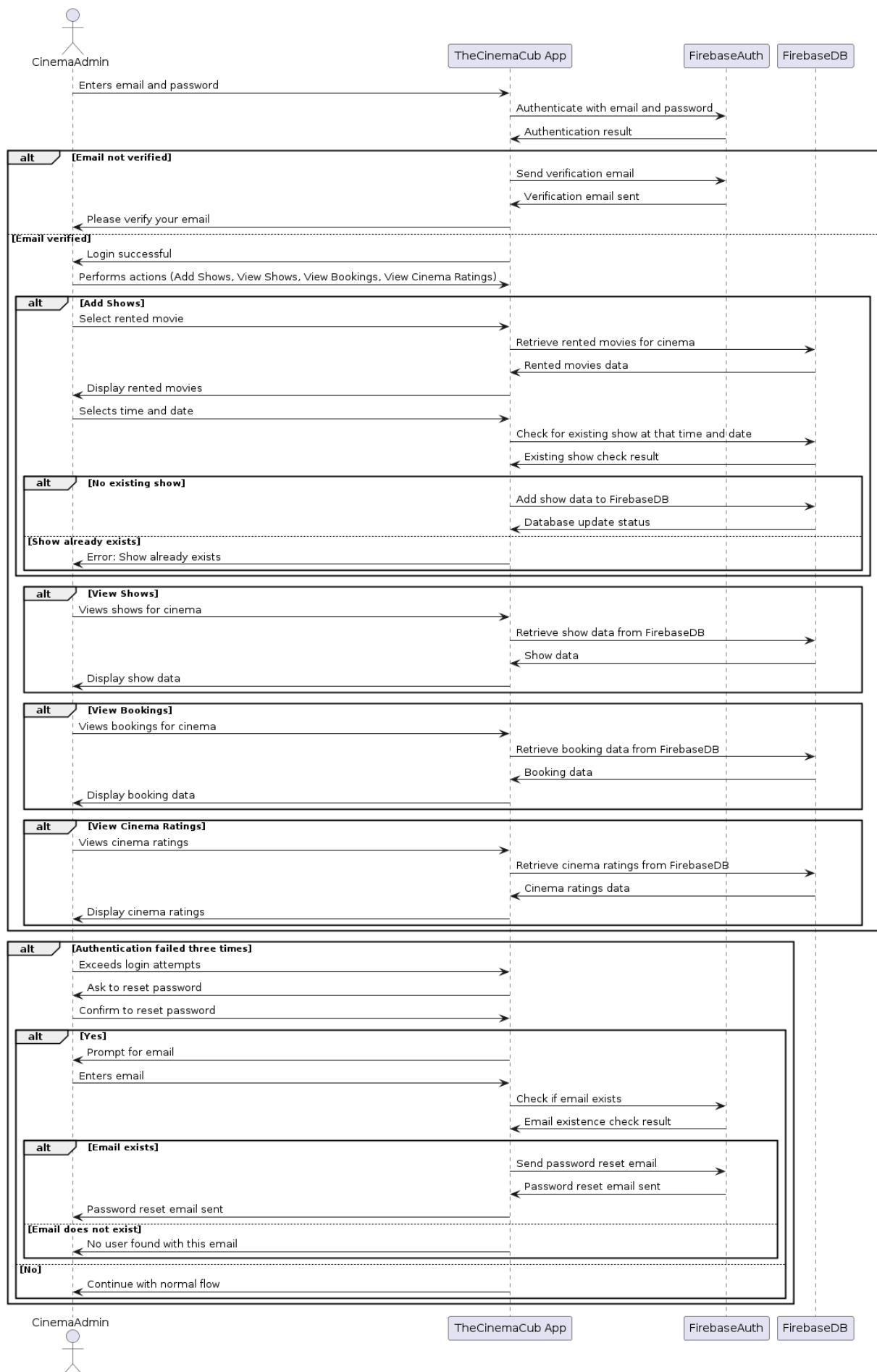


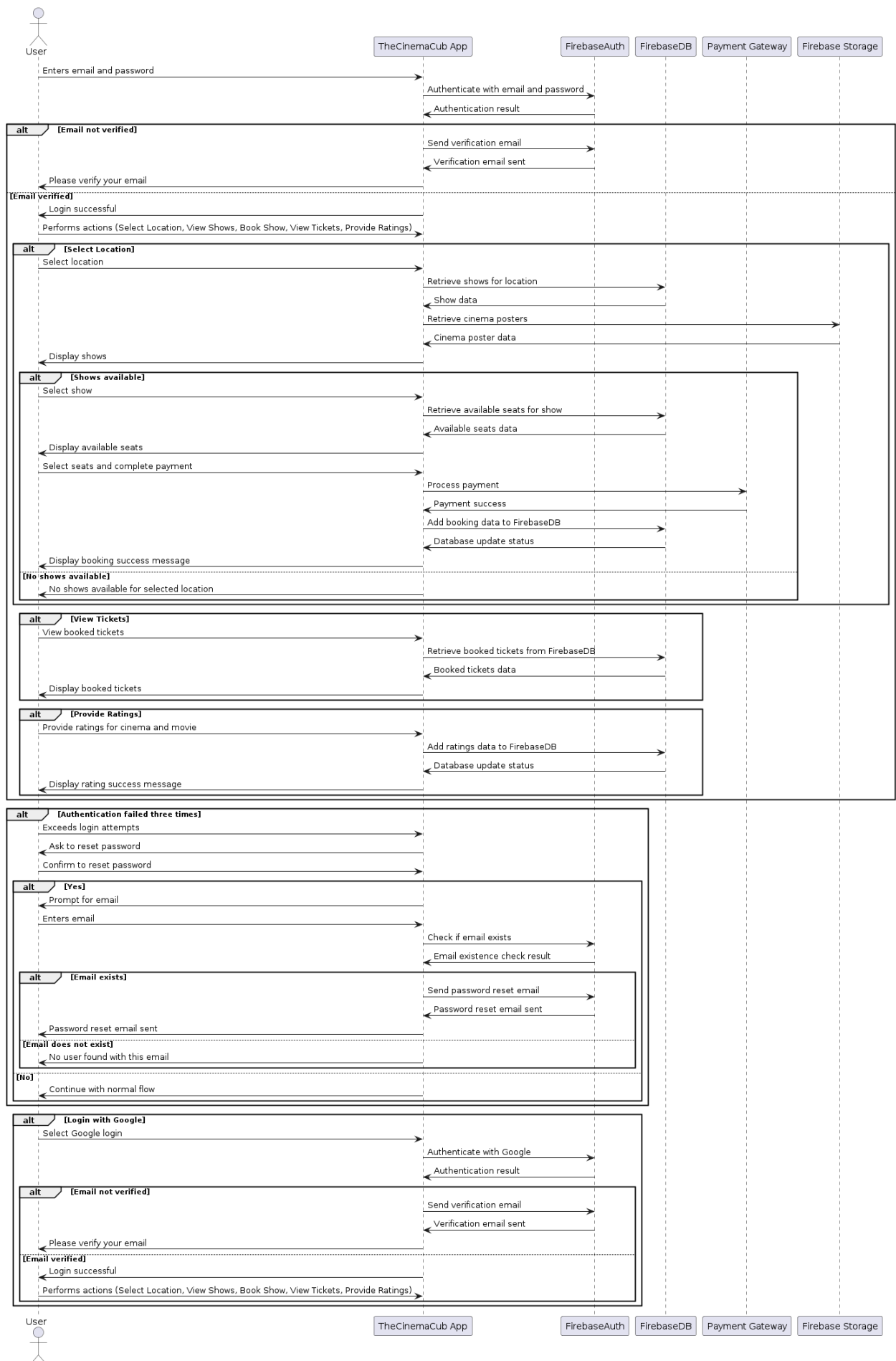


5.2.4) Sequence Diagram

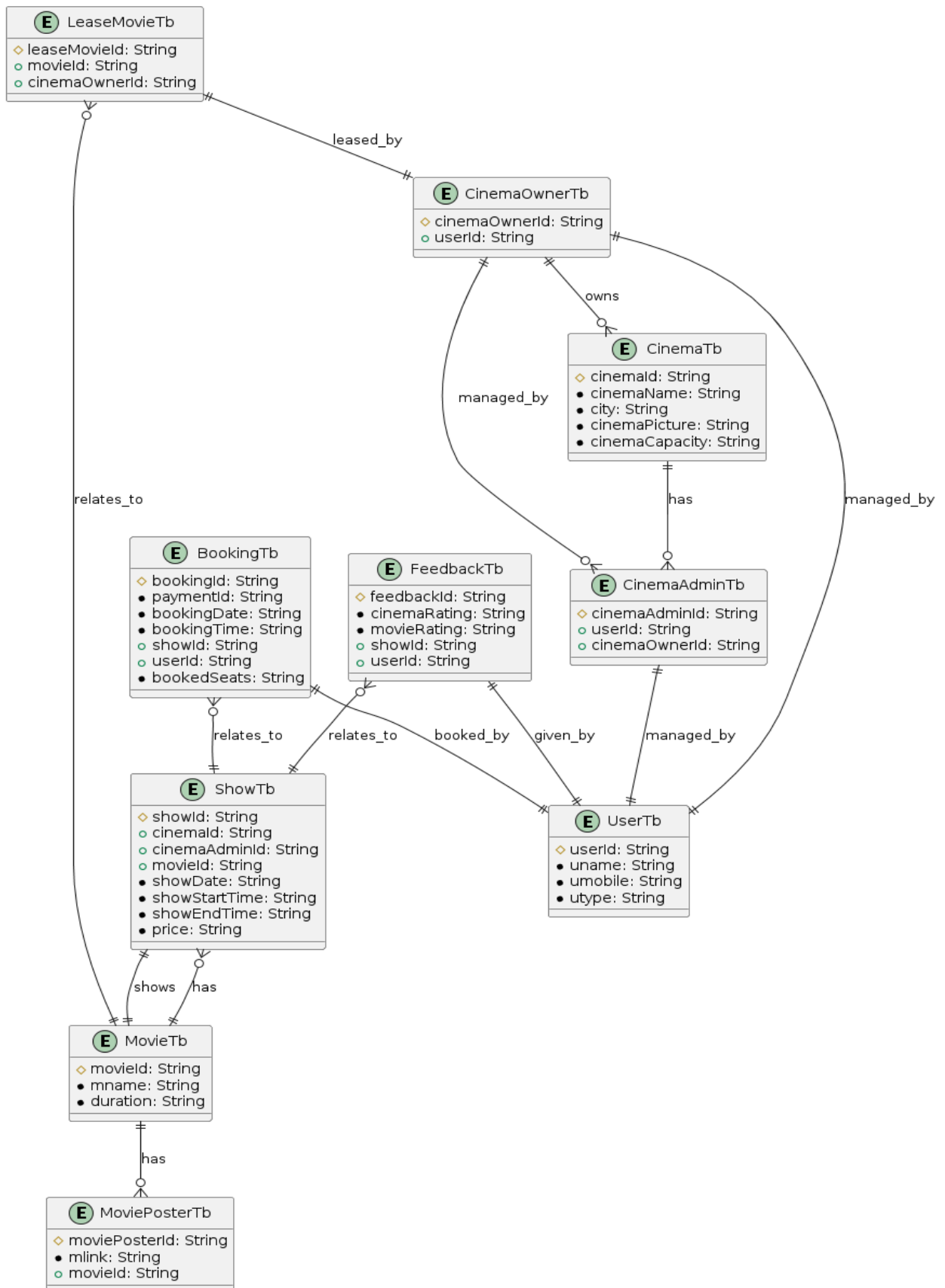




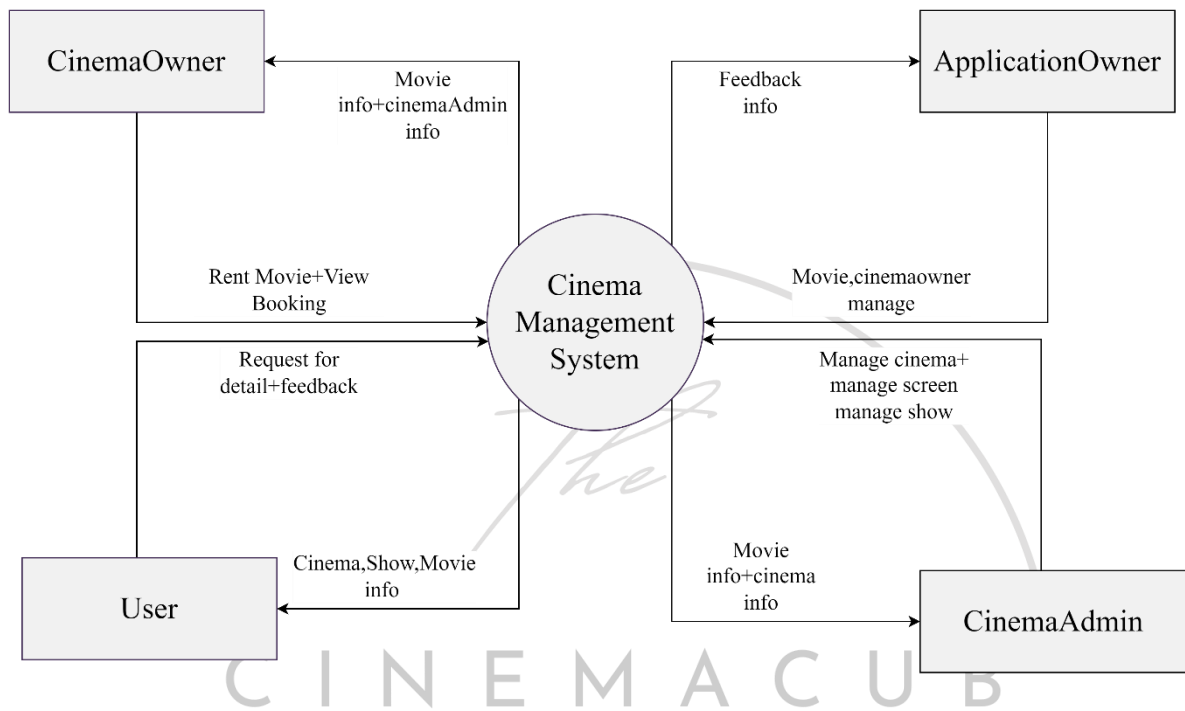




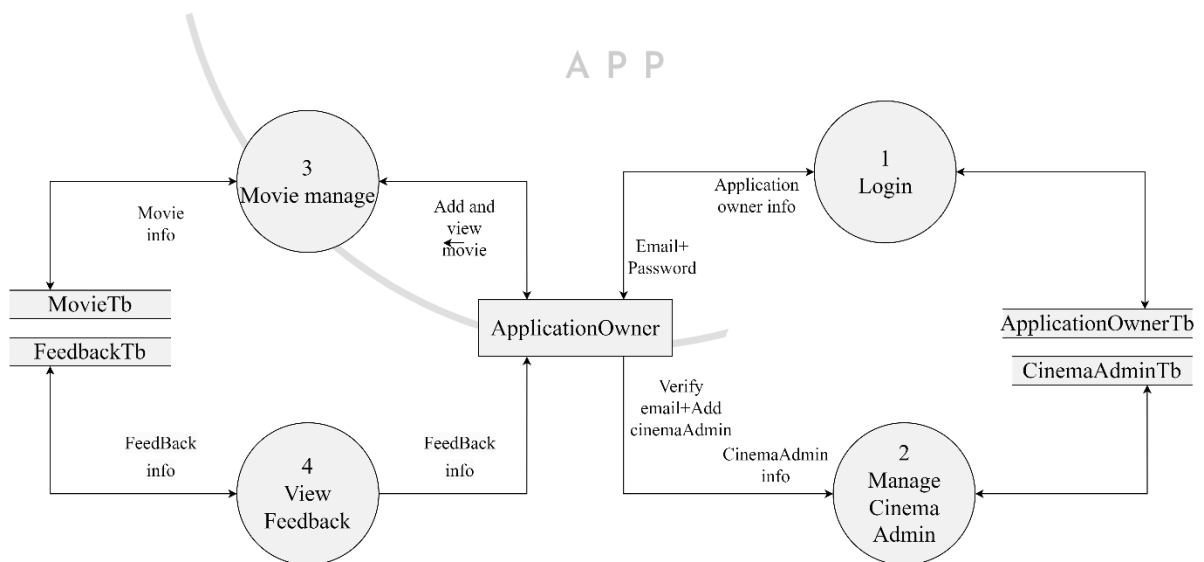
5.3) ER Diagram



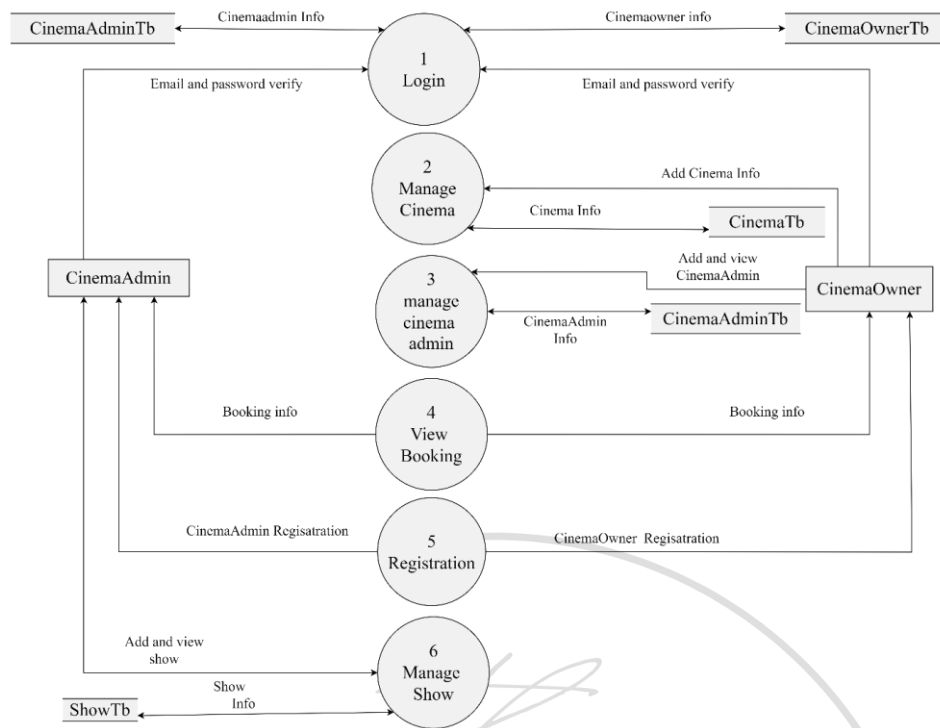
5.4) DFD



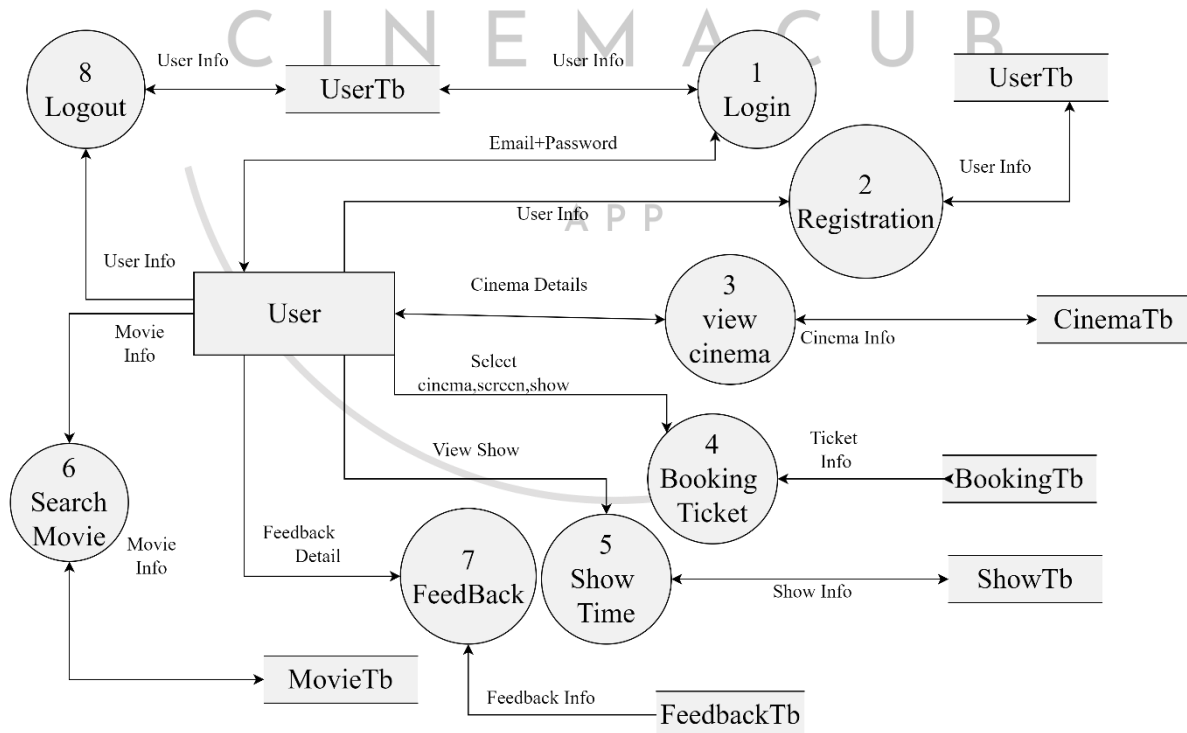
Level 0 DFD



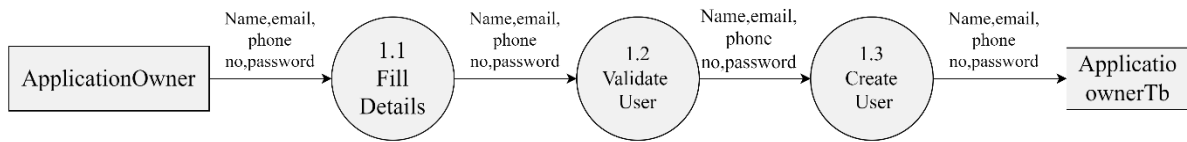
level 1 DFD ApplicationOwner



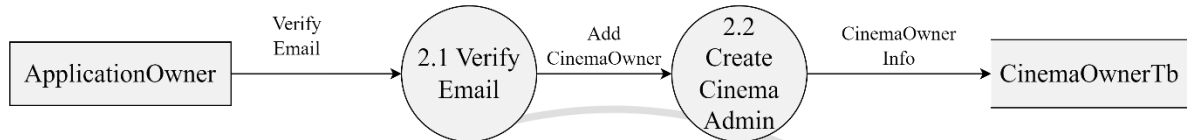
Level 1 DFD CinemaAdmin and CinemaOwner



Level 1 User DFD



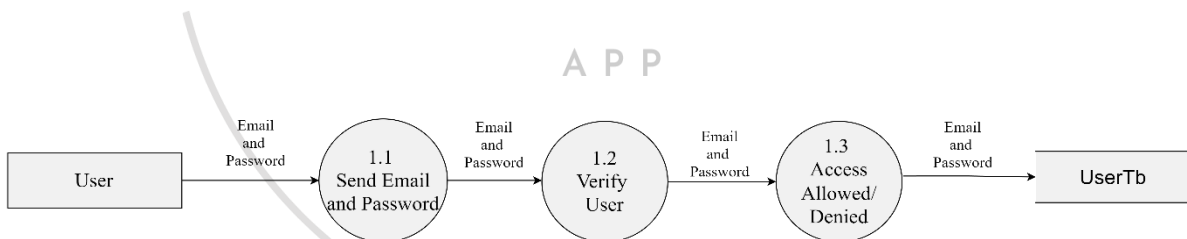
Level 2 DFD of ApplicationOwner (Login)



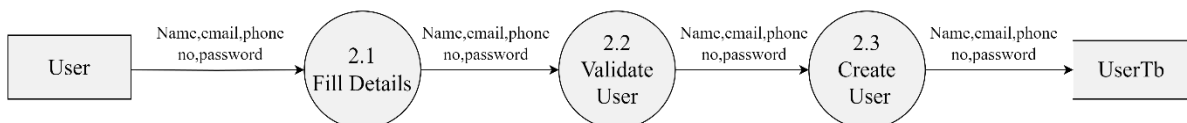
Level 2 DFD of ApplicationOwner (Manage CinemaOwner)



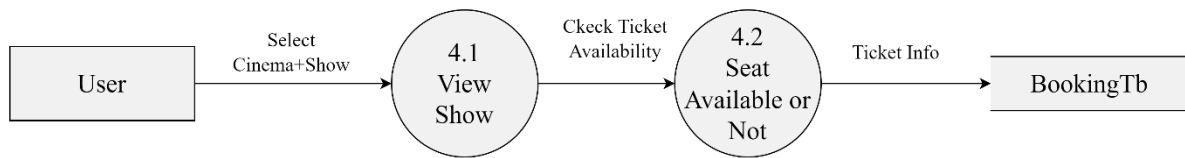
Level 2 DFD of ApplicationOwner (Manage Movie)



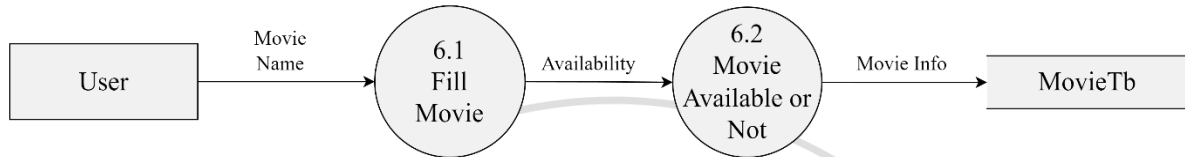
Level 2 DFD of User (Login)



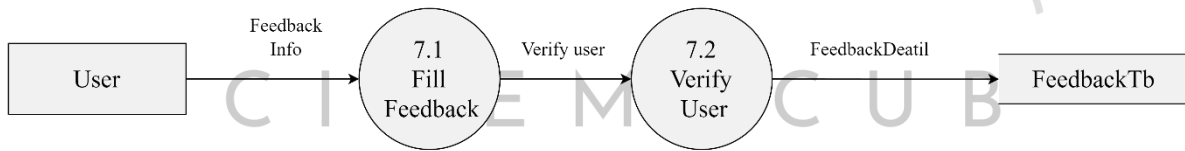
Level 2 DFD of User (Registration User)



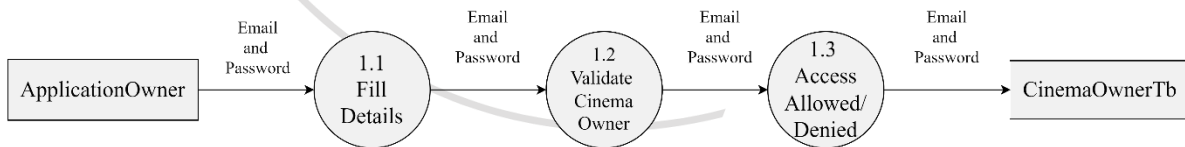
Level 2 DFD of User (Book ticket)



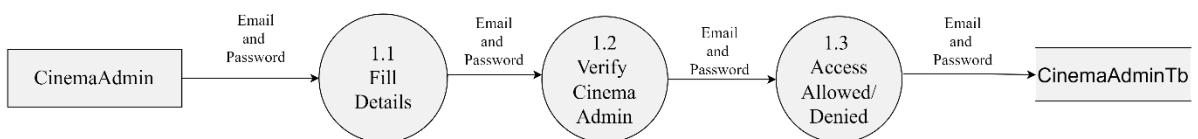
Level 2 DFD of User (Search Movie)



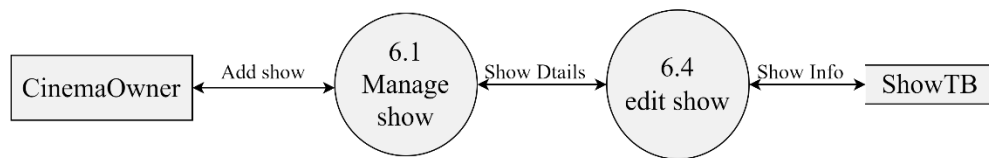
Level 2 DFD of User (Feedback)



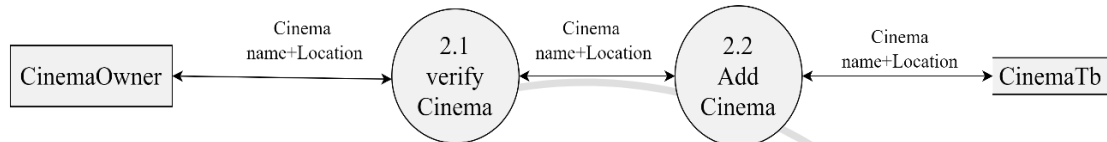
Level 2 DFD of CinemaOwner (Login)



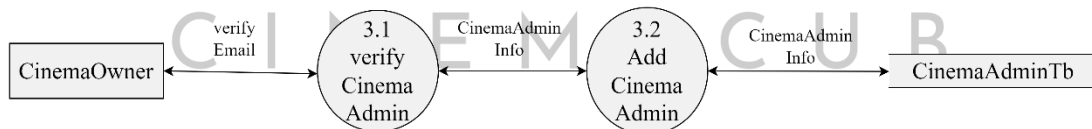
Level 2 DFD of CinemaAdmin(Login)



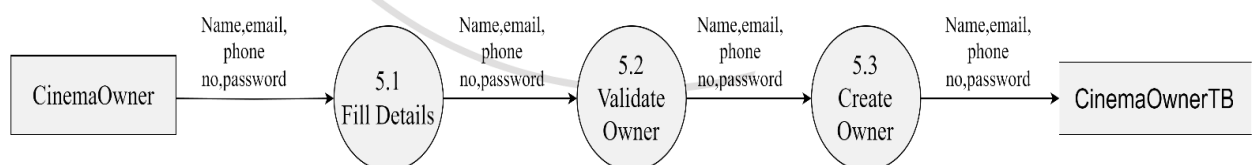
Level 2 DFD CinemaOwner and CinemaAdmin(Manage show)



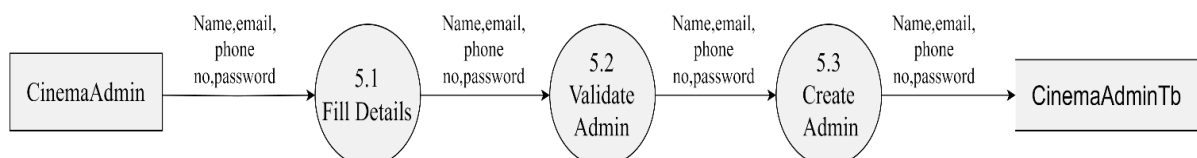
Level 2 DFD CinemaOwner And CinemaAdmin(Manage Cinema)



Level 2 DFD CinemaOwner and CinemaAdmin(Manage CinemaAdmin)



Level 2 DFD of CinemaOwner(Registration)



Level 2 DFD of CinemaAdmin(Registration)

6) Software Design

6.1) Database Design

UserTb:

Element Name	Data Type	Description	Constraints
UserId	String	Unique Id of user	Primary key
UserName	String	Name of user	Not null
UserEmail	String	Email of user	Not null
UserPassword	String	Password of user	Not null
UserType	String	Type of user	Not null

MovieTb:

Element Name	Data Type	Description	Constraints
MovieId	String	Unique Id of Movie	Primary key
MovieName	String	Name of Movie	Not null
MovieDuration	String	Duration of Movie	Not null

MoviePosterTb:

Element Name	Data Type	Description	Constraints
MoviePosterId	String	Unique Id of cinema	Primary key
MoviePosterLink	String	Link for the movie poster	Not null
MovieID	String	Unique Id of cinema	Foreign

LeaseMovieTb:

Element Name	Data Type	Description	Constraints
LeaseMovieId	String	Unique Id of Leased Movie	Primary key
MovieID	String	Unique Id of Movie	Foreign Key
CinemaOwnerID	String	Unique Id of Cinema Owner	Foreign Key

CinemaTb:

Element Name	Data Type	Description	Constraints
CinemaId	String	Unique Id of cinema	Primary key
CinemaName	String	Name of cinema	Not null
City	String	Name of city	Not null
CinemaPicture	String	Link for the cinema picture	Not null
CinemaCapacity	String	Capacity of cinema	Not null

CinemaOwnerTb:

Element Name	Data Type	Description	Constraints
CinemaOwnerId	String	Unique Id of Cinema Owner	Primary key
UserID	String	Unique Id of User	Foreign Key
CinemaId	String	Unique Id of Cinema	Foreign Key

CinemaAdminTb:

Element Name	Data Type	Description	Constraints
CinemaAdminId	String	Unique Id of Cinema Admin	Primary key
UserID	String	Unique Id of User	Foreign Key
CinemaOwnerId	String	Unique Id of Cinema Owner	Foreign Key

ShowTb:

Element Name	Data Type	Description	Constraints
ShowID	String	Unique Id of show	Primary key
CinemaID	String	Unique Id of cinema	Foreign Key
CinemaAdminID	String	Unique Id of cinema admin	Foreign Key
MovieId	String	Unique Id of movie	Foreign Key
ShowDate	String	Date of show	Not null
ShowStartTime	String	Show start time	Not null
ShowEndTime	String	Show end time	Not null
ShowPrice	String	Show price	Not null

BookingTb:

Element Name	Data Type	Description	Constraints
bookingID	String	Unique Id of booking	Primary key
userId	String	Unique Id of user	Foreign Key
showID	String	Unique Id of show	Foreign Key
bookingDate	String	Date of Booking	Foreign Key
bookingTime	String	Time of Booking	Not null
bookedSeats	String	Seats Booked	Not null
paymentID	String	Unique Id of Payment	Not null

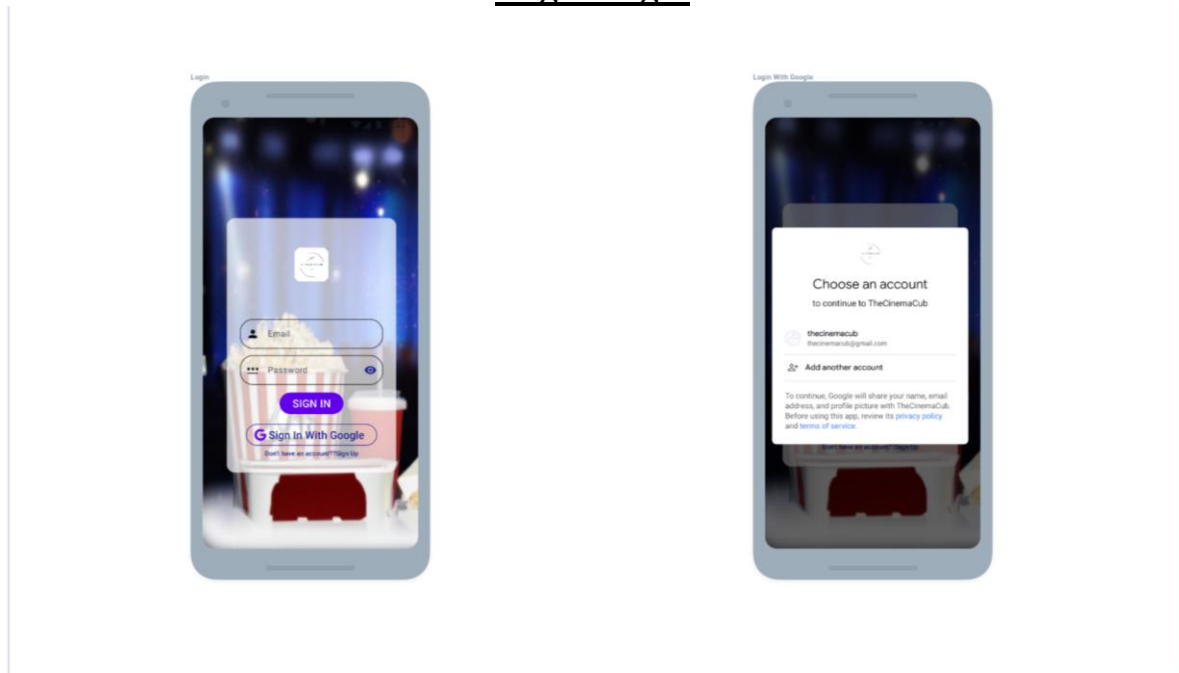
C I N E M A C U B

FeedbackTb:

Element Name	Data Type	Description	Constraints
feedbackId	String	Unique Id of feedback	Primary key
cinemaRating	String	Rating for cinema	Not null
movieRating	String	Rating for movie	Not null
showId	String	Unique Id of show	Foreign Key
userId	String	Unique Id of user	Foreign Key

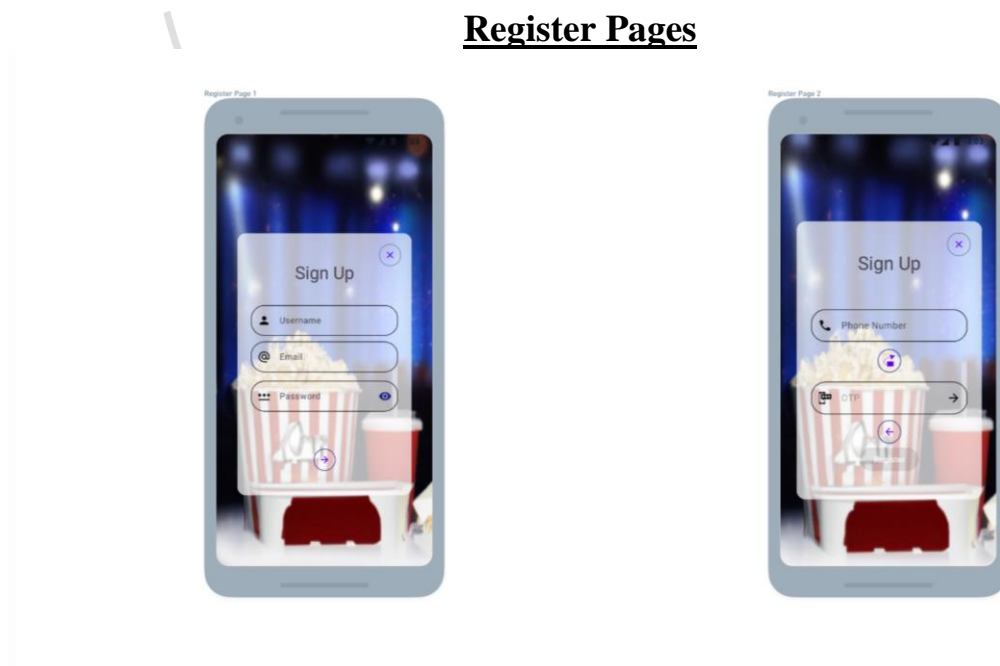
6.2) Interface Design

Login Pages

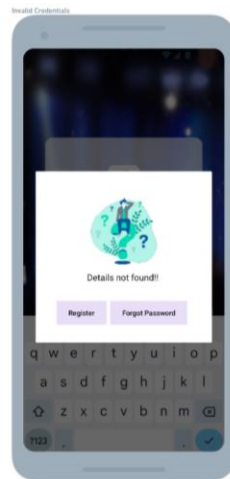


C I N E M A C U B

Register Pages

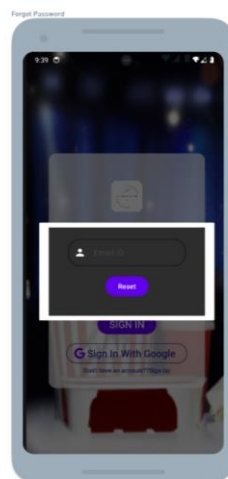


Invalid Credentials

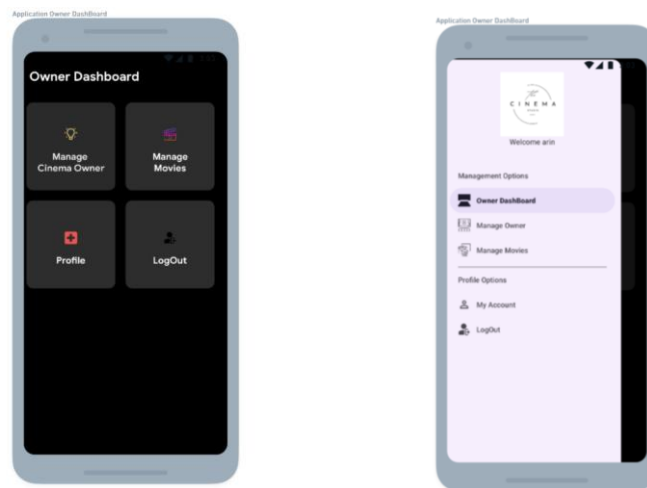


C I N E M A C U B

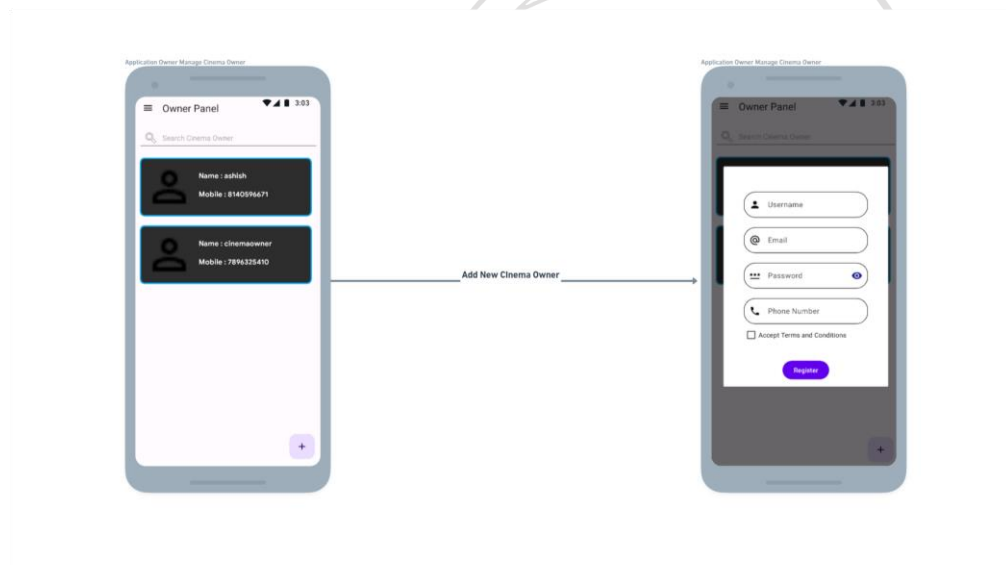
Forgot Password



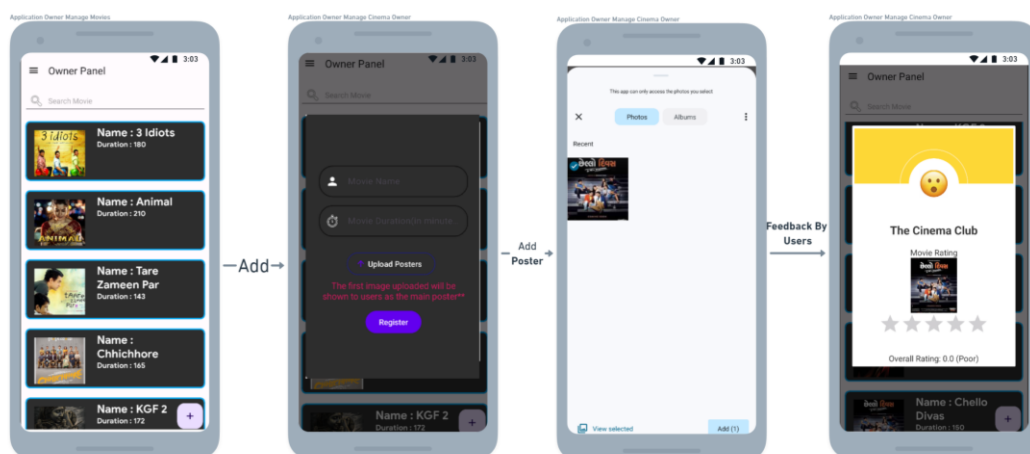
Cinema Owner Dashboard & Navigation



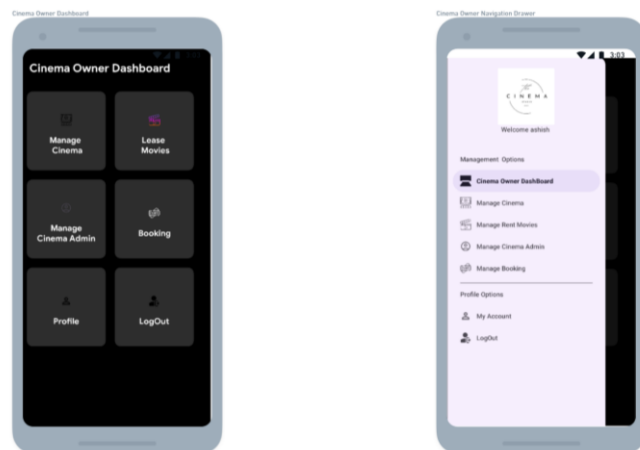
Application Owner Manages Cinema Owner



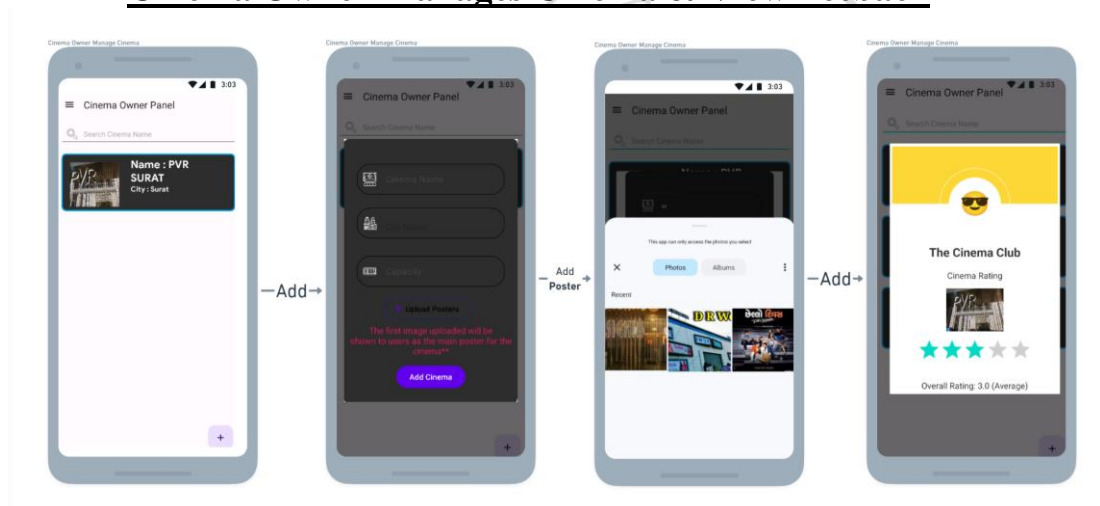
Application Owner Manages Movies



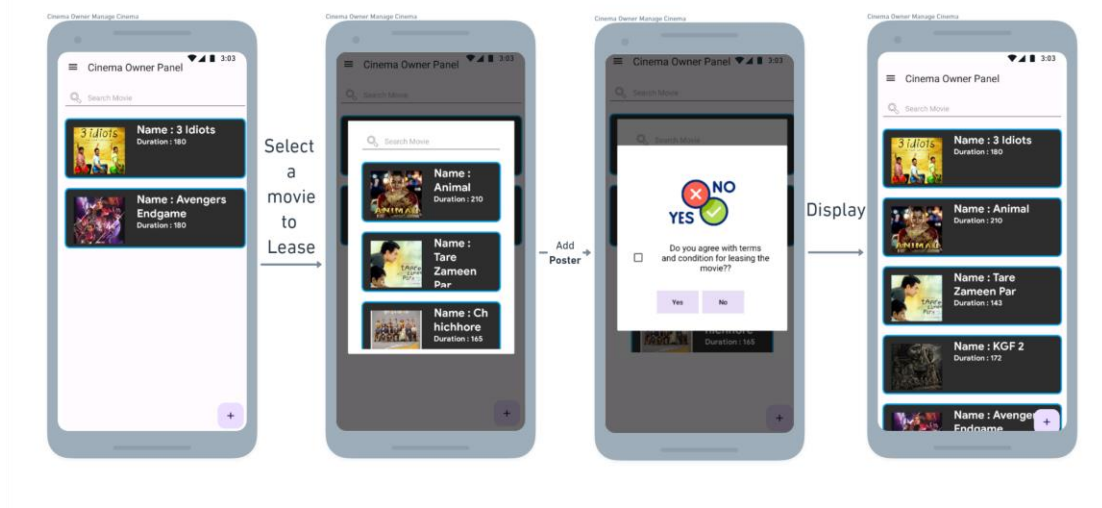
Cinema Owner Dashboard and Navigation Drawer



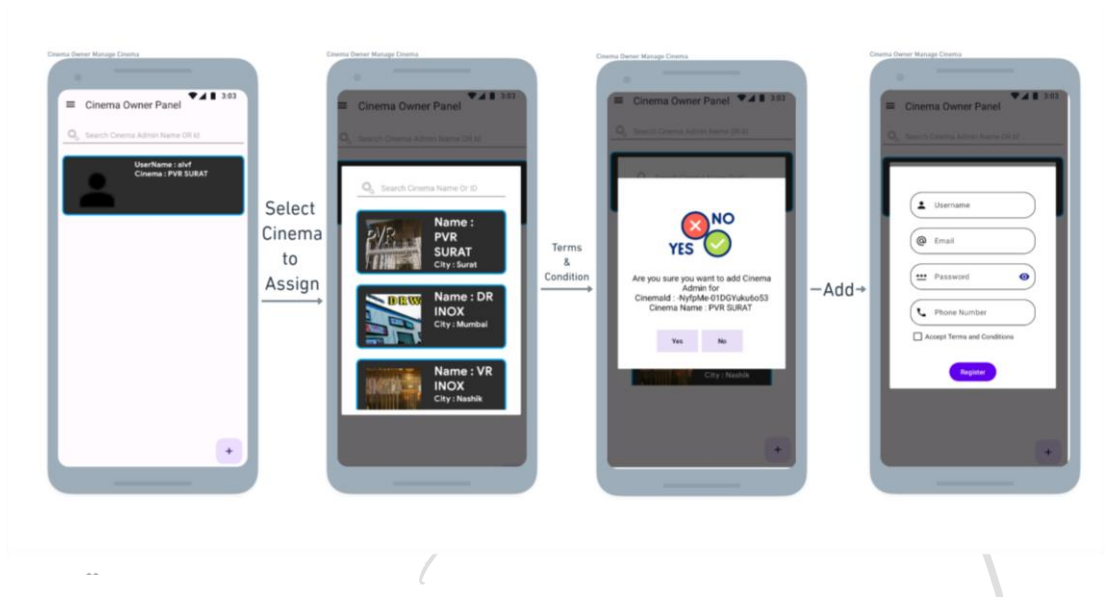
Cinema Owner Manages Cinema & View Feedback



Cinema Owner Rent/Lease Movies

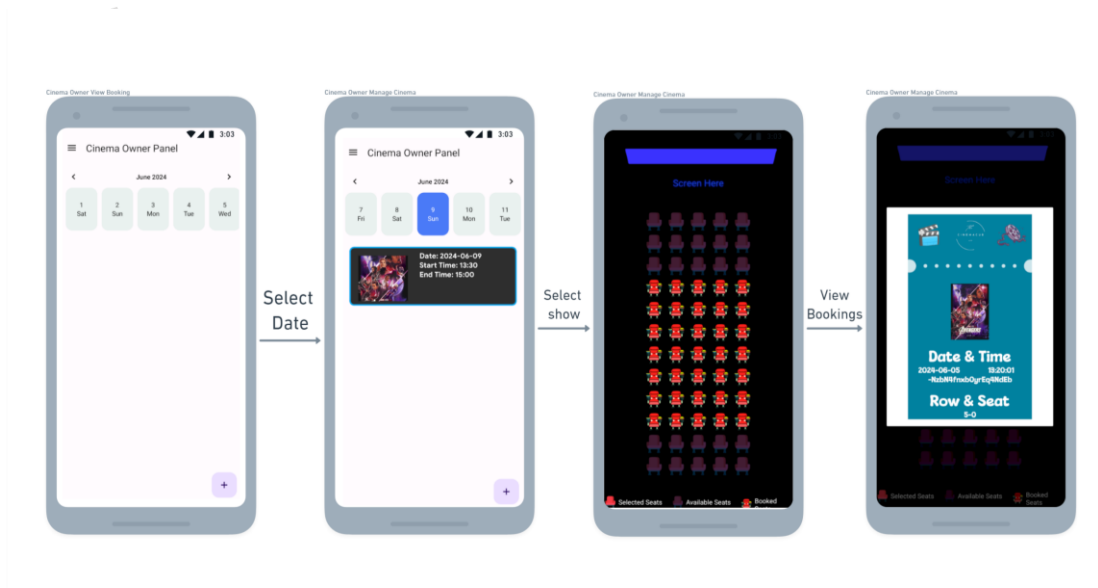


Cinema Owner Manages Cinema Admin



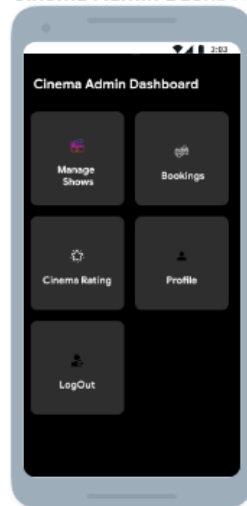
CINEMACUB

Cinema Owner View Bookings

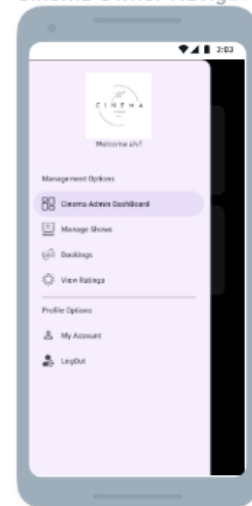


Cinema Admin Dashboard and Navigation Drawer

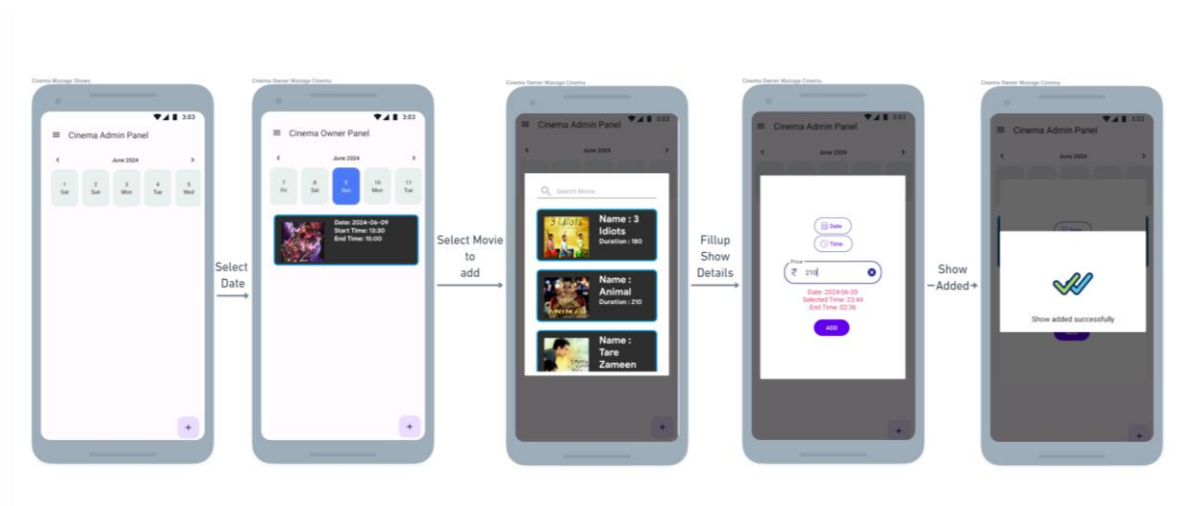
Cinema Admin Dashboard



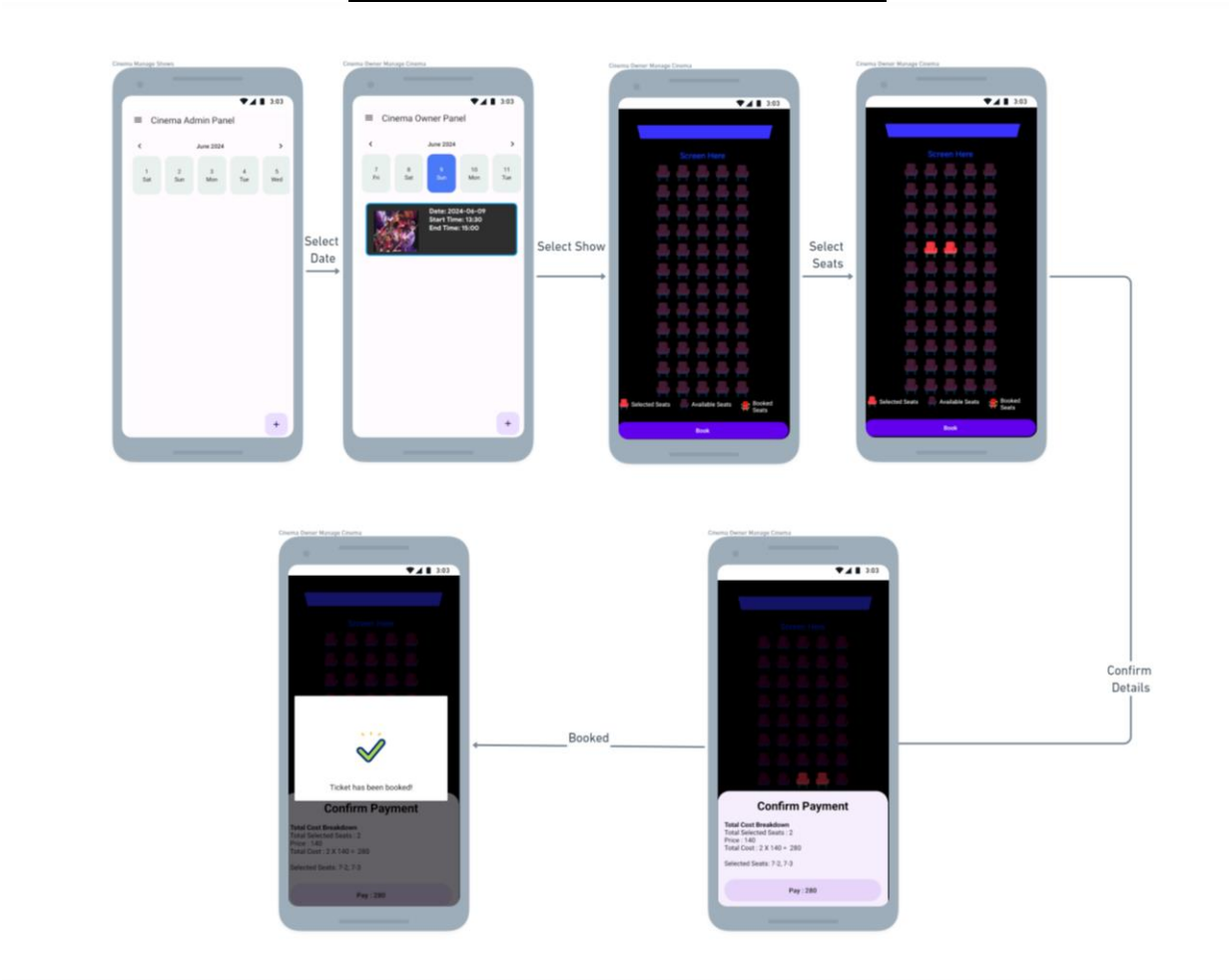
Cinema Owner Navigation



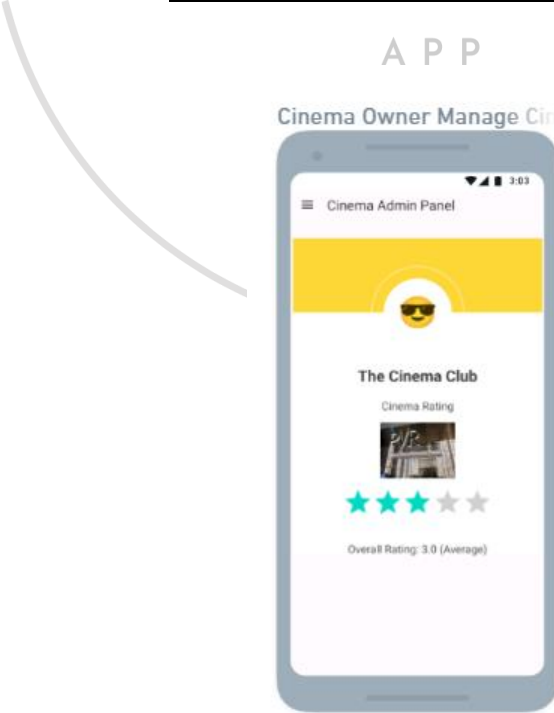
Cinema Admin Manage Shows



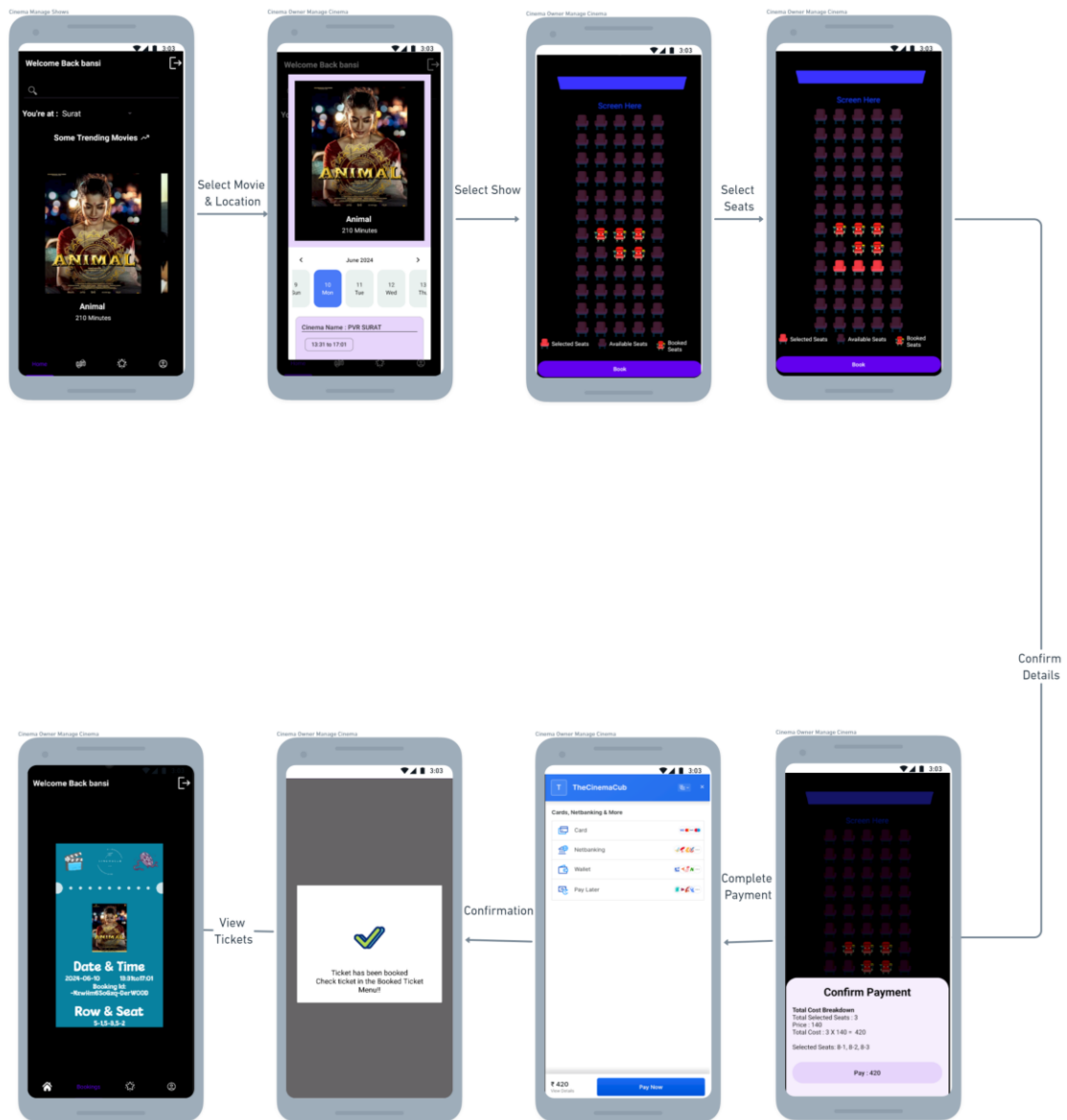
Cinema Admin Manage Bookings



Cinema Admin View Feedback



Normal User Booking Ticket



8) Glossary

1. **Application Owner:**

- The individual or entity responsible for managing the overall application, including overseeing cinema owners, managing movies, and accessing feedback and ratings.

2. **Cinema Owner:**

- The person or entity who owns and manages one or more cinemas, responsible for leasing movies, managing cinema staff, and overseeing booking statistics.

3. **Cinema Admin:**

- A person appointed by the cinema owner to manage show schedules, bookings, and monitor cinema ratings and performance.

4. **Normal User:**

- General users of the application who view current shows, provide feedback, and manage their bookings.

5. **Firebase:**

- A platform developed by Google for creating mobile and web applications, providing services such as authentication, real-time databases, and cloud storage.

6. **Incremental Model:**

- A software development model that builds a system in small, manageable increments, allowing for early delivery of critical

functionality and incorporating user feedback throughout the development process.

7. UML Diagram:

- Unified Modelling Language diagram, used to visualize the design of a system. It includes various types such as Use Case Diagrams, Class Diagrams, Activity Diagrams, and Sequence Diagrams.

8. ER Diagram:

- Entity-Relationship Diagram, a data modelling technique used to illustrate the relationships between entities in a database.

9. SRS (Software Requirement Specification):

- A detailed description of the software system to be developed, including functional and non-functional requirements.

10. Scalability:

- The capability of a system to handle a growing amount of work or its potential to accommodate growth.

11. Usability:

- The ease with which users can learn to operate, prepare inputs for, and interpret outputs of a system or component.

12. Load Testing:

- A type of performance testing to determine a system's behaviour under both normal and anticipated peak load conditions.

13. User Authentication:

- The process of verifying the identity of a user before allowing access to the application.

14.Booking System:

- The module of the application that handles ticket reservations, seat selection, and payment processing.

15.Cloud Storage:

- A service model, in which data is maintained, managed, backed up remotely, and made available to users over a network, typically the internet.

16.Google Authentication:

- A method of verifying users based on their Google account credentials, allowing them to log in using their Google email and password.

17.Real-Time Database:

- A database system that provides real-time data updates, ensuring that users always have access to the most current information.

18.Payment Gateway:

- A service that authorizes and processes payments, allowing users to complete transactions within the application.

19.Data Integrity:

- The accuracy and consistency of data over its lifecycle, ensuring that data remains unaltered and reliable.

20.Stakeholders:

- Individuals or groups with an interest in the project's outcome, including application owners, cinema owners, cinema admins, and normal users.

21. Feasibility Study:

- An analysis of the viability of a project, assessing technical, economic, operational, and schedule feasibility.

22. Task Dependency:

- The relationship between tasks in a project where one task must be completed before another can begin.

C I N E M A C U B

A P P

9) References

- **Android Developers Documentation:**

- The official documentation for Android development, including resources for using Android Studio, Kotlin, and integrating Firebase.

- **Firebase Documentation:**

- Comprehensive guides and references for using Firebase services such as Authentication, Real-Time Database, and Cloud Storage.

- **PlantUML:**

- A tool for creating UML diagrams from plain text descriptions.

- **Google Firebase Authentication with Kotlin:**

- Tutorials and code examples for integrating Firebase Authentication with Kotlin in Android applications.

- **Google Material Components for Android:**

- Library providing material design components.

- **Oops No Internet:**

- Library to detect internet connectivity and show custom error dialogs.

- **OkHttp:**

- HTTP client for Android.
- **Gson:**
 - Library to convert Java Objects into JSON and back.
- **Glide:**
 - Image loading and caching library.
- **Surround CardView:**
 - A custom card view with surround animations.
- **Android GIF Drawable:**
 - Library for displaying GIFs.
- **Horizontal Calendar Date:**
 - A library to create horizontal calendar date pickers with click listeners.
- **Kotlin Standard Library:**
 - Core Kotlin standard library.
- **Animated Bottom Bar:**
 - Library for creating animated bottom navigation bars.
- **Dropdown Library:**

- Library for creating dropdown menus.
- **ViewPager2:**
 - Library for creating swipeable pages.
- **Androidx Browser:**
 - Library to support opening web pages within an app.
- **Google Play Services Auth:**
 - Library for authentication with Google services.
- **Google Play Integrity:**
 - Library for integrating Google Play Integrity API.
- **Razorpay Checkout:**
 - Payment gateway integration library.
- **Android Mail:**
 - Library for sending emails in Android.