DOKUZ EYLUL UNIVERSITY

ENGINEERING FACULTY

DEPARTMENT OF COMPUTER ENGINEERING

CME2210 OBJECT ORIENTED ANALYSIS AND DESIGN

CINEMA MANAGEMENT SYSTEM

by

2017510070 Gaye Süner

Turkey

25.06.2020

Contents

Introduction

1.1 Identify the problem

In this project Cinema Management system will develop for CINEAGOL company. the project will be design for Android. It will also design to manage the record about movies schedule, movies collection, ticket sales.

This page is the starting point into series of pages that attempt to give an object-oriented analysis design and programming applied to a problem: cinema company management.

Since the existing system is manual so it is really difficult to maintain and to keep a full record about the daily purchase of tickets. The information is not up to date and manual system requires staff to maintain their records on the registers.

1.2 Goals

The aim of the project is to manage the cinema company system which provides some conveniences from different ways for the company and their customers.

The scope of the project is

- to record the customers data of the cinema company,
- income and expenses of this company,
- staff salary procedures and
- ticket sales system

in a certain order.

1.3 Stakeholders

The stakeholders can be noted when it comes to our application. Generally most explicit are those that requested for software projects: CEO, company board members. The critical details of what the company is how it will benefit them and their customers. Any local or nearby manager is likewise affected by the steadiness of the software program. How easy the software is for employees shows that the administrator can refer it.

1.4 Motivation

we are motivated with the system is to enhance and upgrade the existing system by increasing its efficiency and effectiveness. The software improves the working methods by replacing the existing manual system with mobile computers based system. Customer will The customers can buy movie ticket from CINEAGOL android application. The company staff can also use the system. he, or she can update the movies attributes, add and delete new movies.

With this system, it was our obligation to help them their time of needed. The software is a android based designed to advertise the company and meet the user needs. The project comes with new movies to become to cinemas, on show movie in cinemas. Users can also purchase available movie tickets and look up for price and starting time. User can also invite their relatives and friends through sharing the link of program in whatsapp, facebook and etc.

1.5 Process Flow Preview

Before the program started to be written, we determined the requirements of the program, the functions that should be included and the necessary features. In the next section, we listed the features and functions of the 5 classes of the program with java codes.

MOVIES ATTRIBUTES AND FUNCTIONS

A- Attributes

Movies ID all the movies are assigned a unique code which helps in search of each movie attributes.

Movies Name It gives facility to the user to select to movie by recognizing through its name.

Movies Description It provides the user with a description of the movie, allowing the user to learn more about the movie. It provides faster ticket sales by reducing the level of indecision of the user.

Movies Language Each movie is broadcast in different sessions in its original language or translation language. The user can buy tickets for the movie in any language and session.

Movies' Theatre each movie has its own hall where it is screened and when it is released, it is randomly displayed in an empty hall. The vision date of the movie is the specified date, but the cinema management will decides on the date when the movie will be released.

Movies' Tickets Prices The total amount of earnings to be added on the ticket price per person decided by the publisher for each movie is the net sales price of the ticket.

Movies Categories The categories and broadcasting press rules developed by the RTÜK in line with the Smart Signs Symbol System in order to protect the viewers from the negative effects of television broadcasts.

Movies score average score given by viewers around the world

B- Functions

AddMovie function is the function of adding to the database for viewing the movies to be released by the users.

```
public void addMovie(String name, String description, String lang,
double price) {}
```

DeleteMovie function is the function of deleting the expired movies. Allows the theater to be emptie.

```
public void deleteMovie(int movieID) {}
```

SearchMovie function is the function of searching a movie that the users want.

```
public Movie searchMovie(String movieName) {return movie;}
```

SearchCategorizeMovie function is the function of searching for a list of the same category movies that the users want.

```
public ArrayList<Movie> searchCategorizeMovie(String category) {return movielist;}
```

STAFF ATTRIBUTES AND FUNCTIONS

A- Attributes

Staff Id all the staffs are assigned a unique code which helps in search of each personal attributes.

Staff Name It keeps the name of the staff.

Staff Surname It keeps the surname of the staff.

Staff Gender It keeps the gender of the staff.

Staff Phone It keeps phone number of the personal and serves to communicate with the staff.

Staff Salary It keeps staff salary and provides determine to salary easily**.**

Staff Position It keeps staff duty.

Staff StartDate It keeps date on which staff entered the job.

Staff Birthdate It keeps birthdate.

Staff Password It provides to access to the management system.

B- Functions

Constructor for adding staffs

- `Staff( String Name,String Surname,String Gender,Phone phone, String position,Date startdate, Date birthdate) {}`
- `get/set functions`

CUSTOMER ATTRIBUTES AND FUNCTIONS

A-Attributes

Customer Id all the customers are assigned a unique code which helps in search of each personal attributes.

Customer Name It keeps the name of the customer.

Customer Surname It keeps the surname of the customer.

Customer Gender It keeps the gender of the customer

Customer Phone It keeps phone number of the customer.

Customer Mail It keeps mail of the customer.

Customer BirthDate It keeps birth date on which customer entered the job.

Customer Membership It keeps customer has a membership or not.

Customer WatchedFilms This list keeps the movies that the customer has gone before.

B- Functions

Constructor for adding customers

```
Customer( String Name,String Surname,String Gender,Phone phone,
Mail mail,Date birthdate) {}
```

```
public void addMember(int customerID) {}
```

```
public void addFilm(Film film) {}
```

SEAT ATTRIBUTES AND FUNCTIONS

A- Attributes

Seat Code It keeps the unique code of the seat for the helps to operate.

Seat Type It keeps the type of the seat to determine special quality.

Seat Status It keeps the status of the seat which gives information (empty or not)

B- Functions

```
Seat(String code, String type) {}
```

```
public boolean checkSeat(int seatCode) {}
```

THEATRE ATTRIBUTES AND FUNCTIONS

A- Attributes

Theatre Id It keeps the unique code of the theatre for the helps to operate.

Theatre Name It keeps the name of theatre (unique)

Seat[ ] seats It keeps seats of the theatre.

B- Functions

```
public void updateSeat(int seatCode) {}
```

```
public Seat searchSeat(int seatCode) {}
```

```
public Seat[] printAvailableSeats(){ }
```

For our process flow, we plan to constantly communicate between employees and customers. In order to plan all aspects of the project in detail, we thought that it was necessary to contact all organizations selling movie tickets while planning. Where we have gone wrong or felt like that, we have received information from the real people in this business.

Finally, we find people who need communication with the movie ticket purchase service. This will allow us to make minor changes to the interaction between the configured items and those who will use the software directly and have some kind of testing process. We think that allowing users to see how the software is made will facilitate the transition during the distribution phase.

Analysis and Design

2.1 Plan for Requirements Engineering

inception task

The need is to virtualized the display movies attributes system of CINEAGOL cinema. There are properties of the movies that help maintain and manage records through this project. The user can first Registration in enter the Personal details, and User login and If you want to update personal Details and you and update. The user collect all information like Movies details,Theatre Details.

In this stage we describe CEO's and the all team members responsibilities and understand the cases created by Stakeholders. We want to figure out that how often our customers use this app. Thanks to analysis of this question, we can answer how often the software will go into maintenance question.

Elicitation task

Our goal in this stage is identify the problem, purpose solutions and the talks amongst each other on the many different approaches. Meetings are arranged each other software developers online and this was repeated every Thursday until deadline. Every team member worked from their home and they finished their tasks on time.

elaboration tasks

The information collected from the initial stage is brought together in this group and cleaned. Here we explain how the software behaves. We set up the simplest scenarios so that the customer understands how it works with our mobile app. We define how each feature and each function interact with each other.

negotiation tasks

For any customer who will need a solution, the supply-demand relationship takes care of the team and stakeholders discuss them to find a solution. The requirements are listed from the most important and urgent to the trivial and urgent. Even if the goal is to save money, the goal can be ignored in this process, assuming that the returns in the 6-month period proceed positively.

specification task

During this task, we plan to create a software requirements specification template. In this template, we will note the overall purpose of the project and the audience. Explanations about product features, user classes / features, working environment and design will be included. Along with this written document, a 'model' and 'mockups' will be created to gain a visual perspective on the project.

validation task

The all stages and requirements are described clearly. After all work will end, the minus parts will be identified and the steps will be applied again.

requirements management

Any changes that may occur during the project phases will be made at the first moment the error occurs. Whether the time allotted for the construction of the project will allow such a change, that is, by the stakeholders and the software engineering team, is decided and the customer's right to engage in this work will be reviewed, discussed and determined.

2.2 Functional Requirements

hardware requirements

The software should be ran on Android Devices release after Android 4.4. Our program does not have a desktop application or website. Ios version will be released later for its usage to become widespread.

System development environment

Java is a general-purpose programming language that is class-based, object-oriented, and designed to have as few implementation dependencies as possible.

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. IntelliJ that Android Studio is built on supports all released Java versions.

declaration of functions in java

```
public class ClassName {
 public <ObjectType> functionName(<ObjectType> parameter) {
        ... //function.
        return <ObjectType>;    }
```

Android interface - Primary tasks

*View all movies which are in theatres

- Connects with the firebase database to call all movies and their attributes such as imdb point and description and display them.

*Book seat for a movie

- When the user picked an movie, available seats comes for that movie and when the user choose a seat, the dialog opens and requests information of the user.

*Allow administrators to login the management system

- Login display activity.

*Menu activity according to user logged (only admin)

- Menu displays which includes all the system management panel.

Android interface - Secondary tasks

*Allow for update movies

- Only for administrators can change the movie informations which are seen by the user.

*Allow for ticket operations

- Only for administrators can manage the ticket (delete,update)

*Authenticate any administrator logging in

- communicates with the firebase database to verify the inputted email and password is correct.

*Allow for staff operations

- Only for administrators can manage the staffs (add,delete,update).

Backend - tasks

*Connecting firebase database

- Managing and checking the connections between the android studio and firebase database.

*Checking the inputs

- adjusting and checking the inputs to the desired type.

*Managing interfaces

- Setting the user privileges and setting app interfaces according to that.

2.3 Non functional requirements

Performance requirements

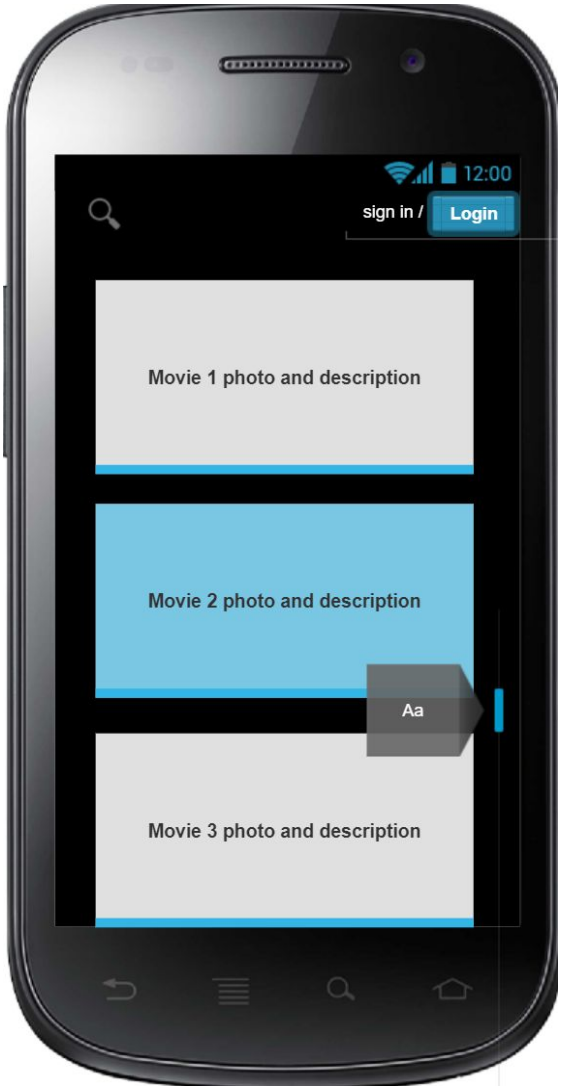
Performance requirements define how well the system performs certain functions under certain conditions. Examples are response speed, efficiency, execution time and storage capacity. Service levels that include performance requirements are often based on supporting end-user roles. Like most quality features, performance requirements are key when designing and testing the product.

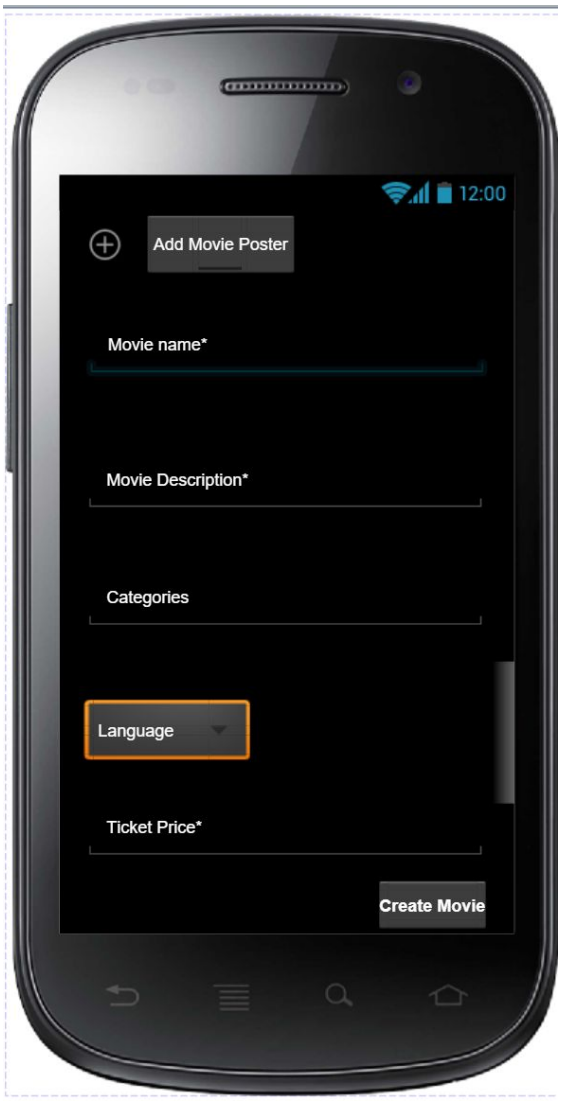
The requirements in our program are limited to the uses in the free version of the firebase database, which we use. We can only serve 100 employees until the agreement with customers is renewed. it can also be easily used for Android 4.4 and later versions.

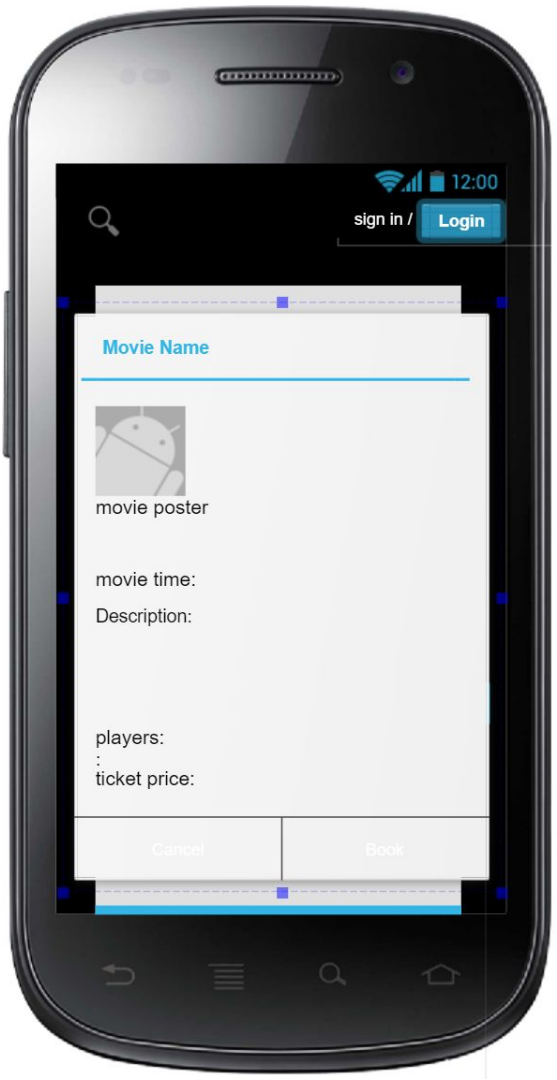
quality attributes

- it reaches the goal fast.
- simple and useful.
- user friendly.
- The design is stylish.
- dual platform for both the audience and the cinema owner.

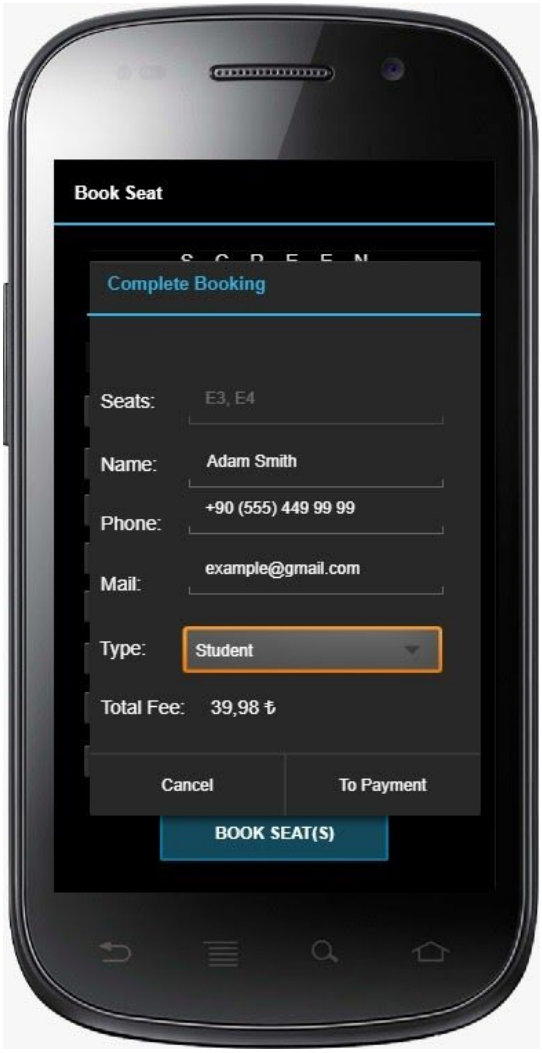
screenshot mockups




12:00
Add Movie Poster
Movie name*
Movie Description*
Categories
Language
Ticket Price*
Create Movie


12:00
sign in /
Login
Movie Name
movie poster
movie time:
Description:
players:
:
ticket price:
Cancel
Book


Book Seat
S C R E E N
A4 A7 A8
B1 B2 B6 B7 B8
C1 C2 C3 C6 C7 C8
D1 D4 D5 D6 D7 D8
E1 E2 E5 E7 E8
F1 F4 F5 F6 F7 F8
G1 G2 G3 G8
H1 H2 H3 H4 H5 H6 H7 H8
DS1 DS2
BOOK SEAT(S)


Book Seat
Complete Booking
Seats: E3, E4
Name: Adam Smith
Phone: +90 (555) 449 99 99
Mail: example@gmail.com
Type: Student
Total Fee: 39,98 ₺
Cancel
To Payment
BOOK SEAT(S)

2.4 Use Cases

Use Case #1: Checking book status

Primary actor: Seat

Goal : To see seat is available for booking or not.

Preconditions: Seat must be exist.

Trigger: Request for the booking or searching.

Scenario:

1. Login to the system
2. Selects book a seat on app.
3. Selects avaliable seats on app.
4. Observes status of seat.

Exceptions:

1. Login error (Send error message: cause of wrong password or id)
2. Seat does not exist (Send error message)

Open issues:

-Should the seat status have an extra option to reservation?

Use Case #2: Booking a seat

Primary actor: Ticket

Goal: Ticket sales at the request of customers

Preconditions: Chosen seat must be empty and a request must be come

Trigger: Customer's request

Scenario:

1-Login to the system

2-Selects theatre and seat on app

Exceptions:

1-Login error

2-Choosing seat which is already full

Open issues:

-Does the purchase require a membership?

2.5 Models

System Diagrams

The Unified Modeling Language (UML) is a general-purpose, developmental, modeling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system.

Drawing UML diagrams is much cheaper than writing code. On the other hand, UML can be useful for creating roadmaps of large software structures. Such roadmaps give developers a quick way to find out which classes depend upon which others and provide a reference to the structure the whole system.

UML class diagrams allow us to denote the static contents of, and relationship between classes. In a class diagram we can show the variables and functions of classes.we can also show whether one class inherits from another, or whether it holds a reference to another.

Class Diagram

In diagram 1, person class is an abstract class. Staff and Customer classes are inherit from Person. Movie, Theatre and Person classes are inherit from ICRUD interface. All these classes has add, delete, update and search functions.

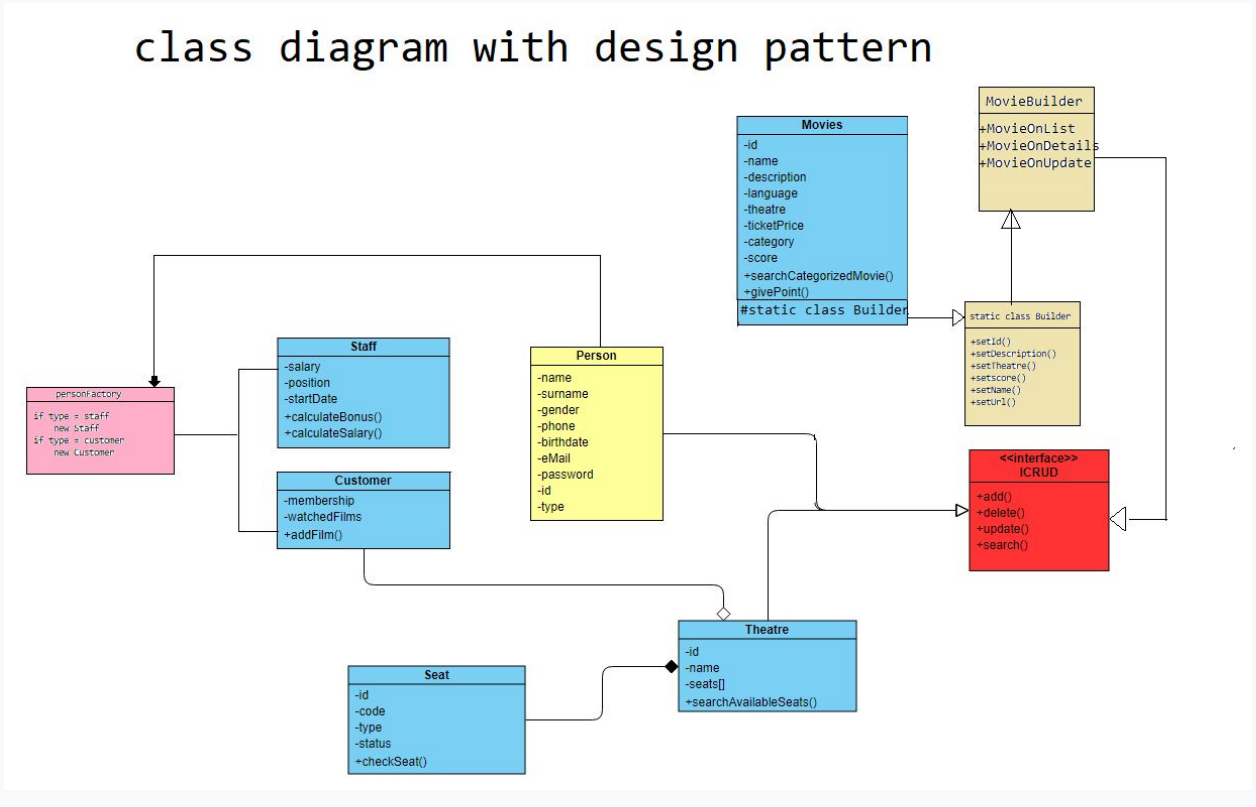


diagram 1 : cinema management system class diagram

Use case Diagram

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements.The use cases, which are the specific roles played by the actors within and around the system. As shown as diagram 2, the system has three actors and shows relationships between the actions that they can perform specifically.

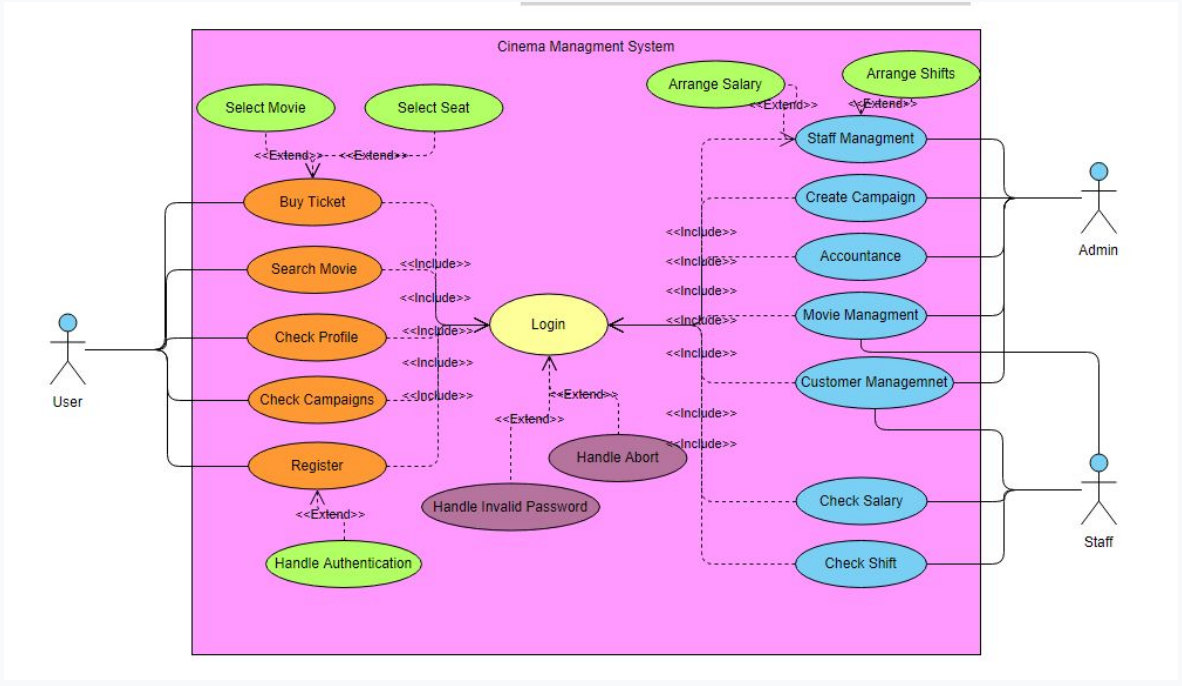


diagram 2 : cinema management system use-case diagram

Activity Diagram

In the diagram, nine activities are identified. It shows how admin panel system works. First step is login. When user login successfully, checking authority of user begins and some permissions are given. User can intervene the actions according to the given permissions as seen in diagram 3.1.

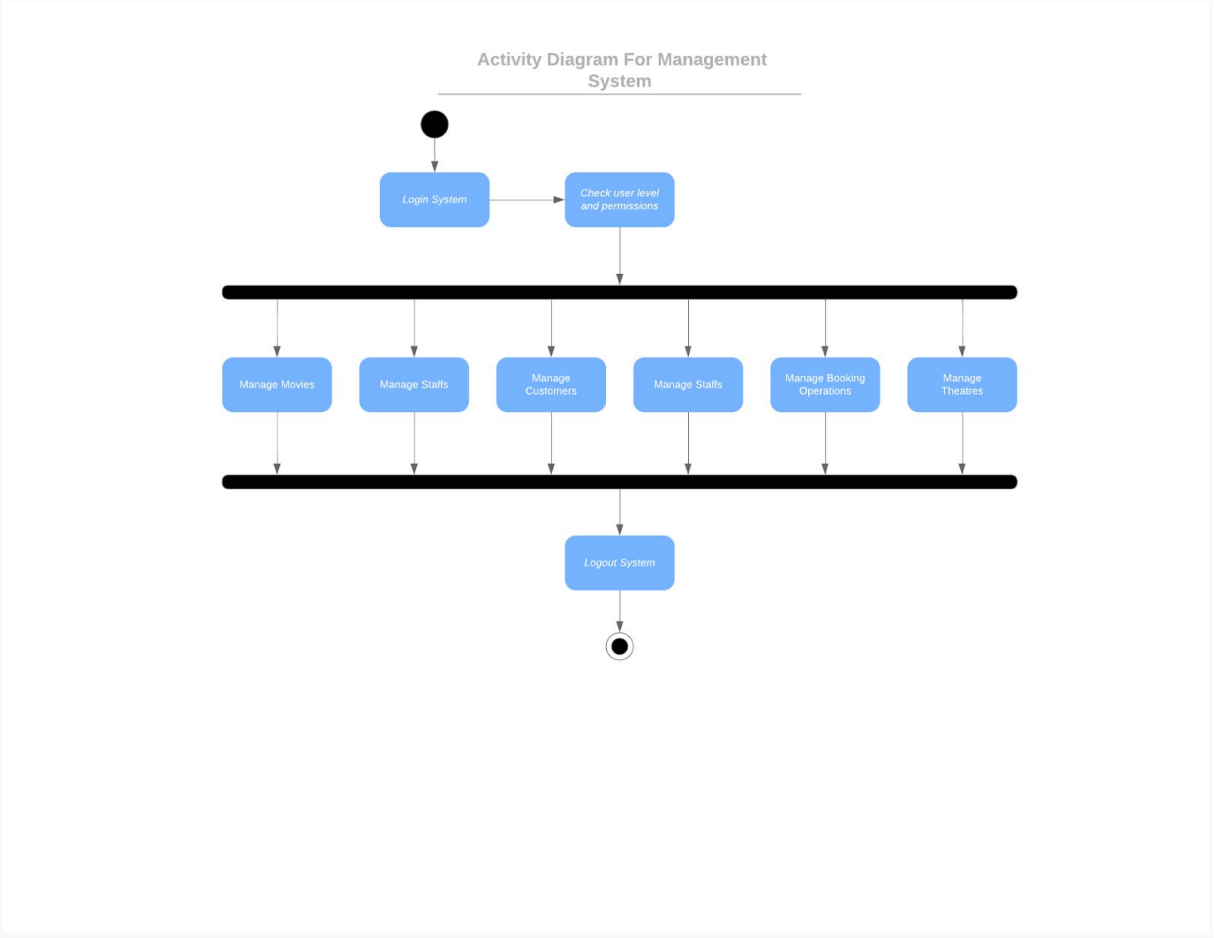


diagram 3.1 : admin person's login and authorized function

Diagram 3.2, shows the adding operation for the staffs. Firstly, required informations is requested such as name,phone etc. After that register actions starts and an account is created for the staff. Finally, position, work time and days are determined.

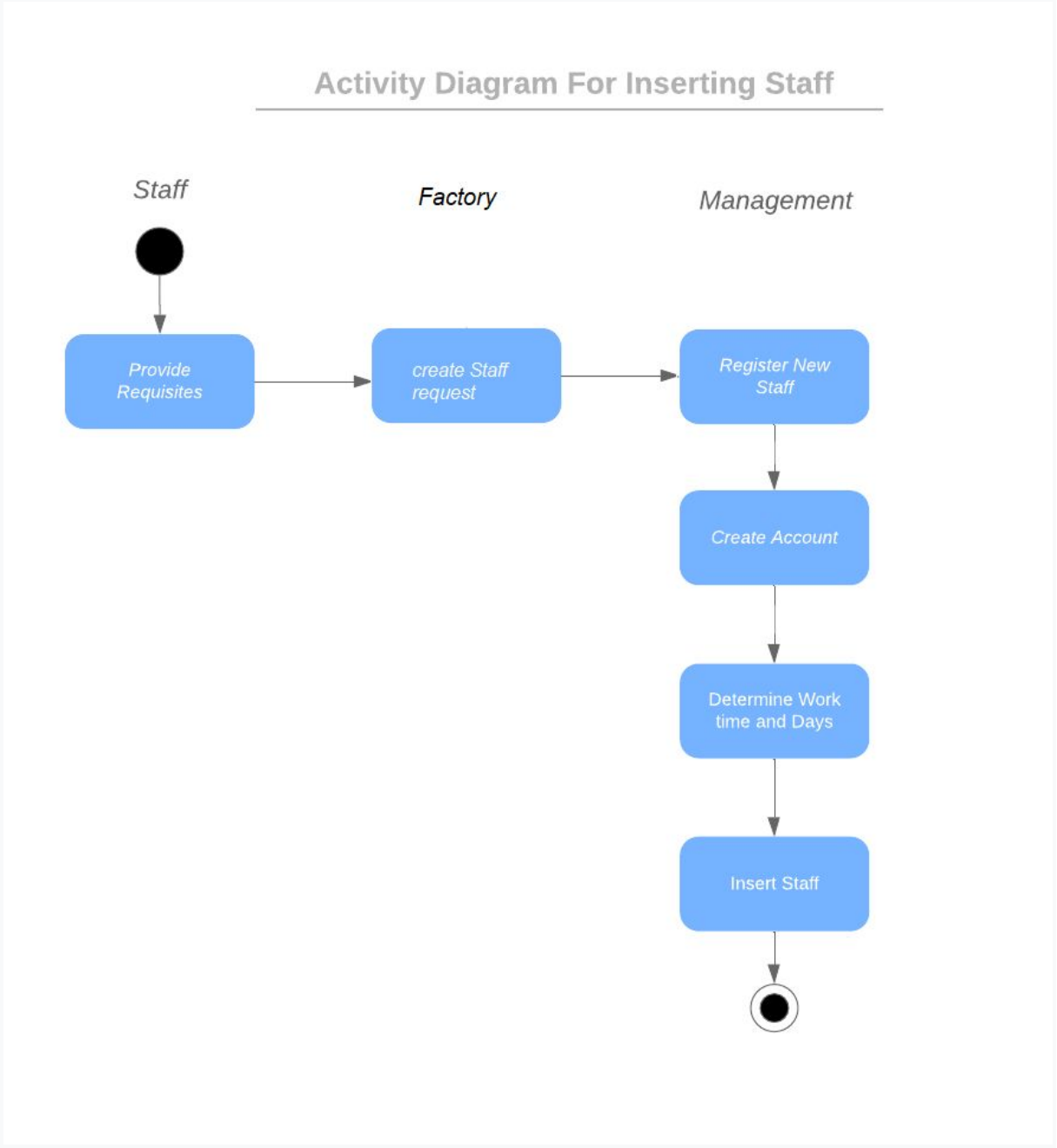


diagram 3.2 : staff registration made by admin

State Diagram

In diagram 4 shows a State diagram that describes a finite state machine that controls the way Booking Seat in the Cinema theatre. The rectangles represent states. In diagram, we enter the BookingSeat state, we invoke the window that seats available. The arrows between the state are called transitions. Each is labeled with the name of the event that triggers the transition. In diagram, if we are in the Payment booking state, and we get pay event, then we transition to the Release Ticket state and invoke create virtual ticket.

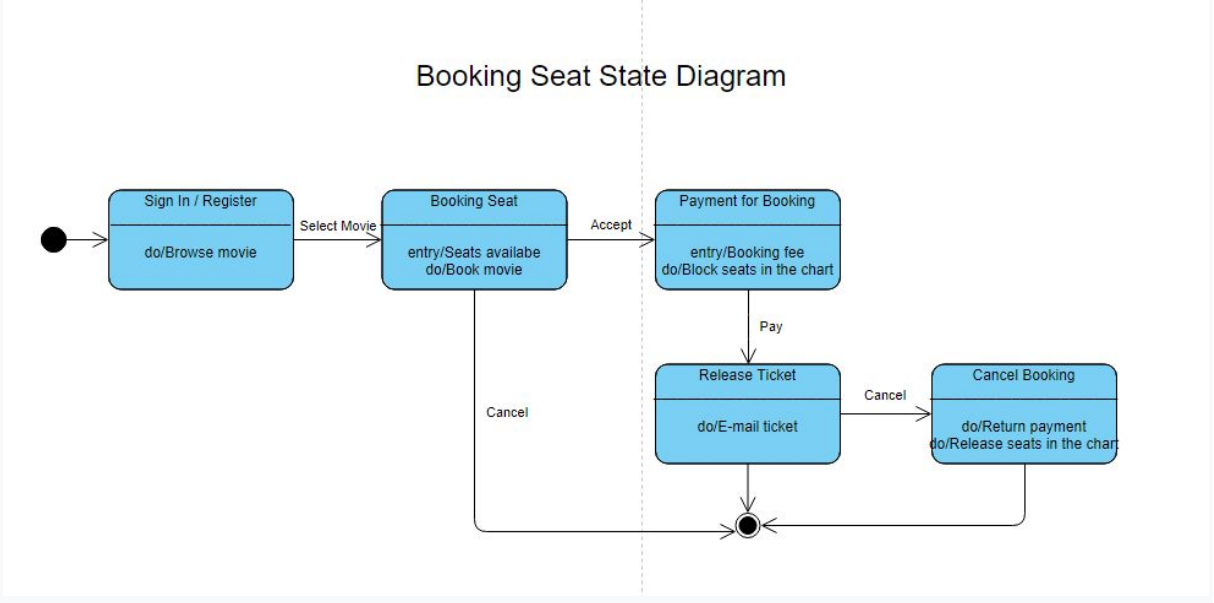


diagram 4 : booking seat state diagram

Sequence Diagrams

A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur.

In diagram 4.1 the admin saves the movie attributes to the database from movieAdding window that only admin can access. while the movie's attributes are recorded, the v,s,on date range of the movie is also entered. When the last day of the movie in vision comes, the movie is automatically deleted from the list.

Our system does not allow ticket sales without logging in. In diagram 4.2 for the customer to buy tickets, they must first log in to the system with their customer account. In addition to the option to search by the name of the movie, the logged in customer can also search for movies with the filtering option. After the customer chooses the movie he wants to watch, he should also choose an empty seat he/she wishes. After all these processes are completed, the customer is directed to the bank to make the ticket payment.

The diagram 4.3 is most general representation and summary of the diagram 3.1 and diagram 4.2

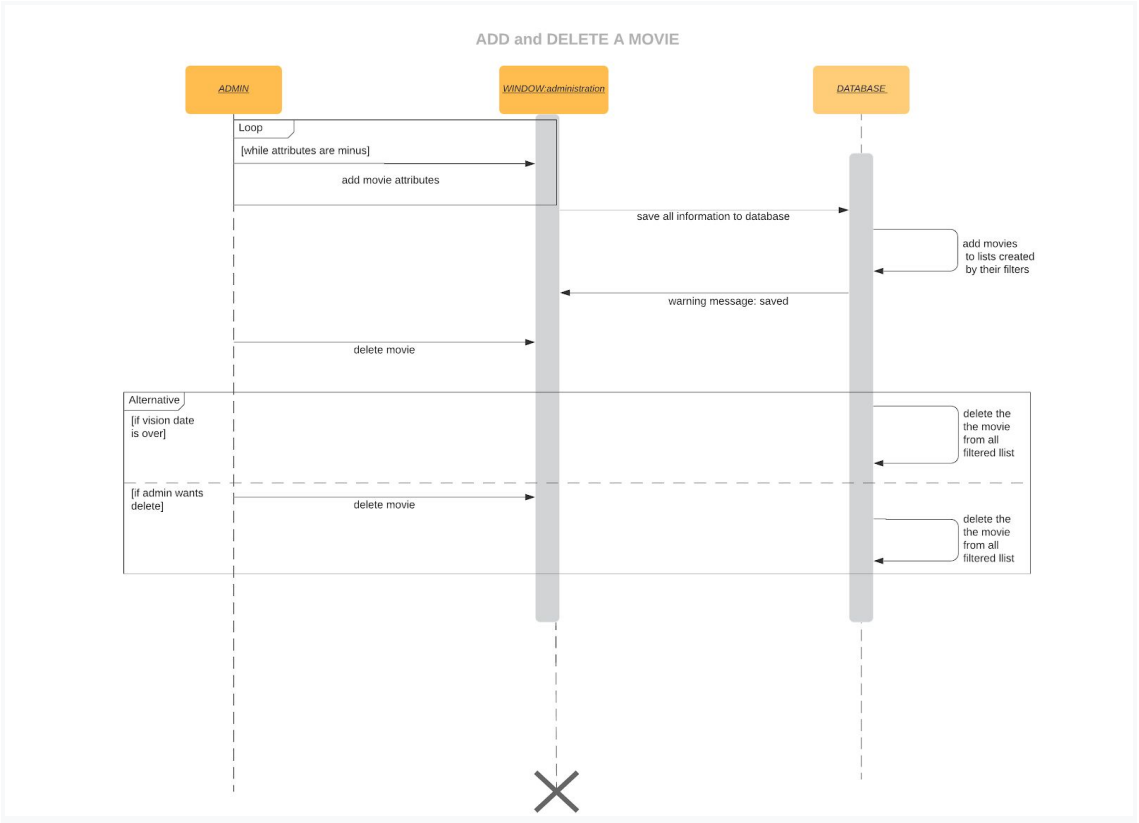


diagram 4.1 : adding and deleting movie

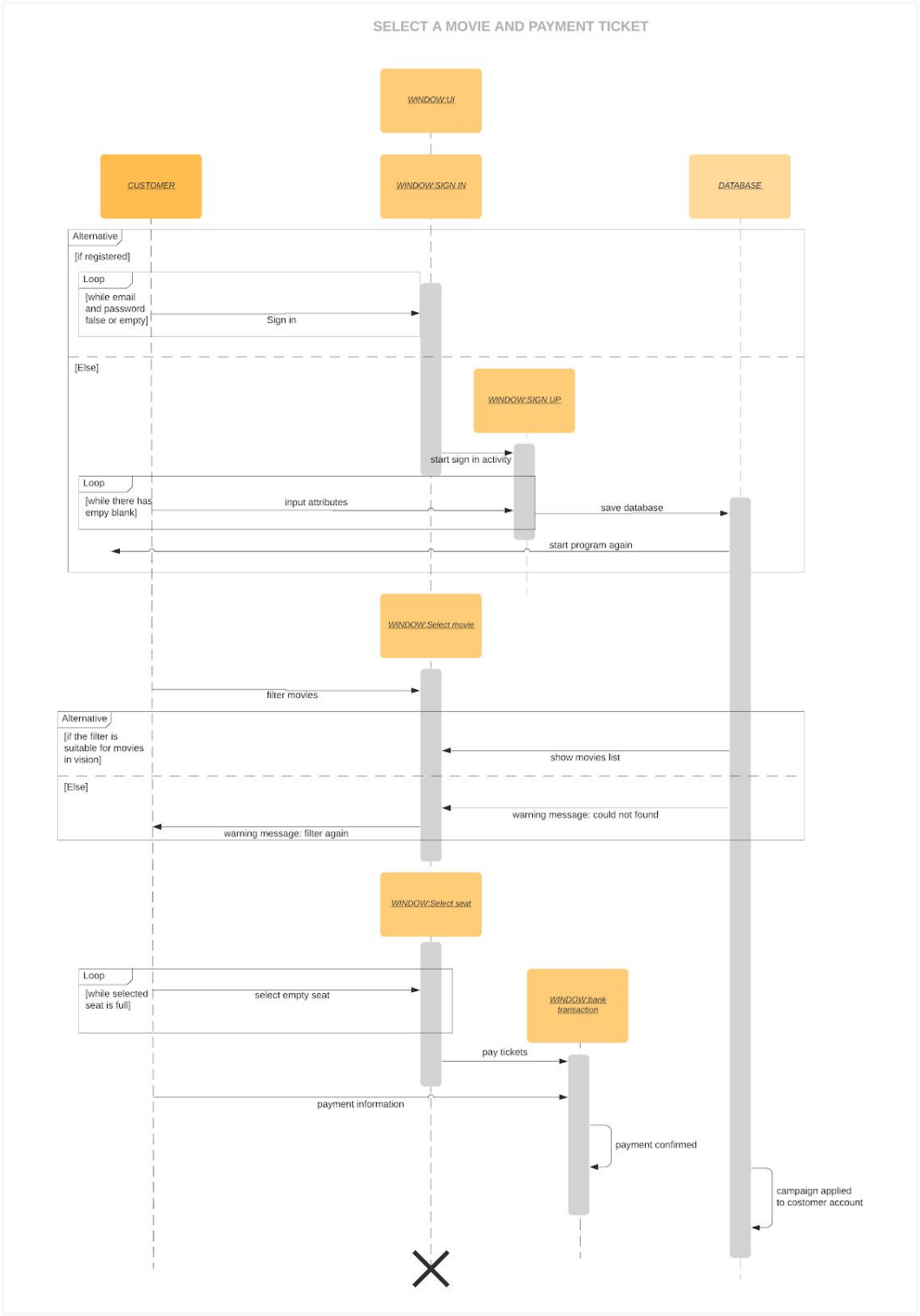


diagram 4.2 : login, selecting movie, booking seat and pay ticket price

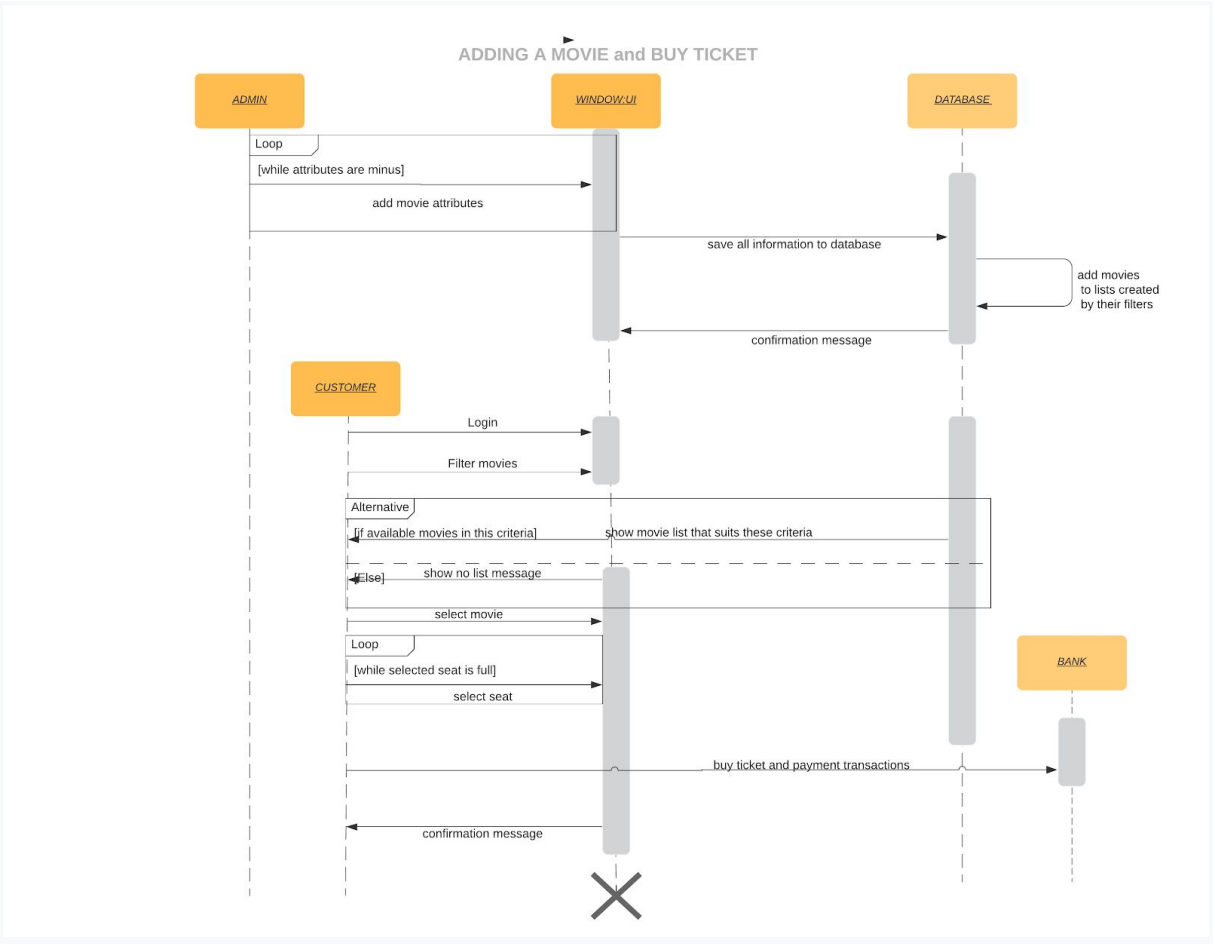


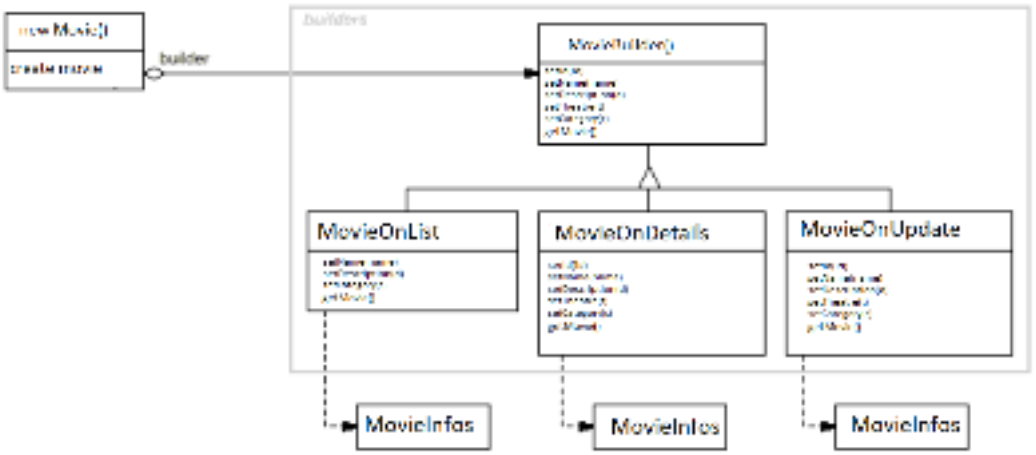
diagram 4.3 : adding movie and buy ticket

2.6 Design Patterns

When developing this mobile application, the values entered with an interface are recorded in the database. When recording a new movie or updating an existing movie, some values may not be written. In such a case, more than one constructor would have to be written to create a new object. We can use Builder design pattern to correct this situation, that is, to reduce the use of constructor.

Builder Design Pattern

Separate the construction of a complex object from its representation so that the same construction process can create different representations.



The Builder pattern captures all these relationships. Each converter class is called a **builder** in the pattern, and the reader is called the **director.**

Participants

BUILDER : Movie.MovieBuilder()

In Movie class, construction wants to different representation for the object that's constructed. This class prevents writing more constructors to create objects errorless with a different number of attributes.

CONCRETEBUILDER : MovieOnList(), MovieOnDetails(), MovieOnUpdate()

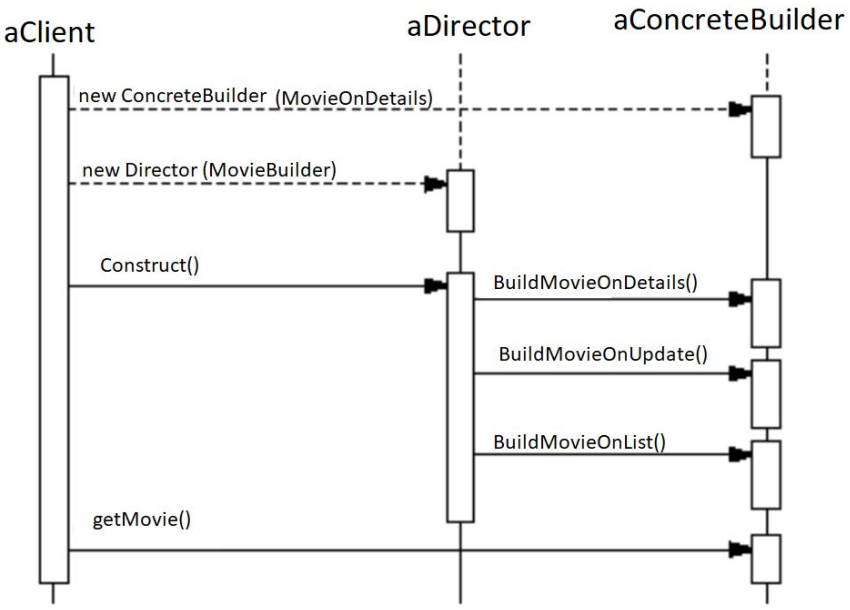
The functions here create Movie objects that will be used in different places. For example, while only the title, description, poster and score attributes of the movies are displayed in the movieOnList function on the main page; In addition to these attributes in the MovieOnDetails function, it displays attributes such as id, theater hall where it is played.

DIRECTOR : Movie()

This class keeps the concreteBuilder functions staticly.

PRODUCT : Movie Object

Objects with the same ancestor that will work in different places. Represents the complex object under construction. concretebuild(ex: MovieOnList) is builds the product's internal representation and defines the process by which it's assembled.



-The client creates the Movie object and configures it with the desired Builder object.

-Director notifies the builder whenever a part of the product should be built. (MoviesOnList)

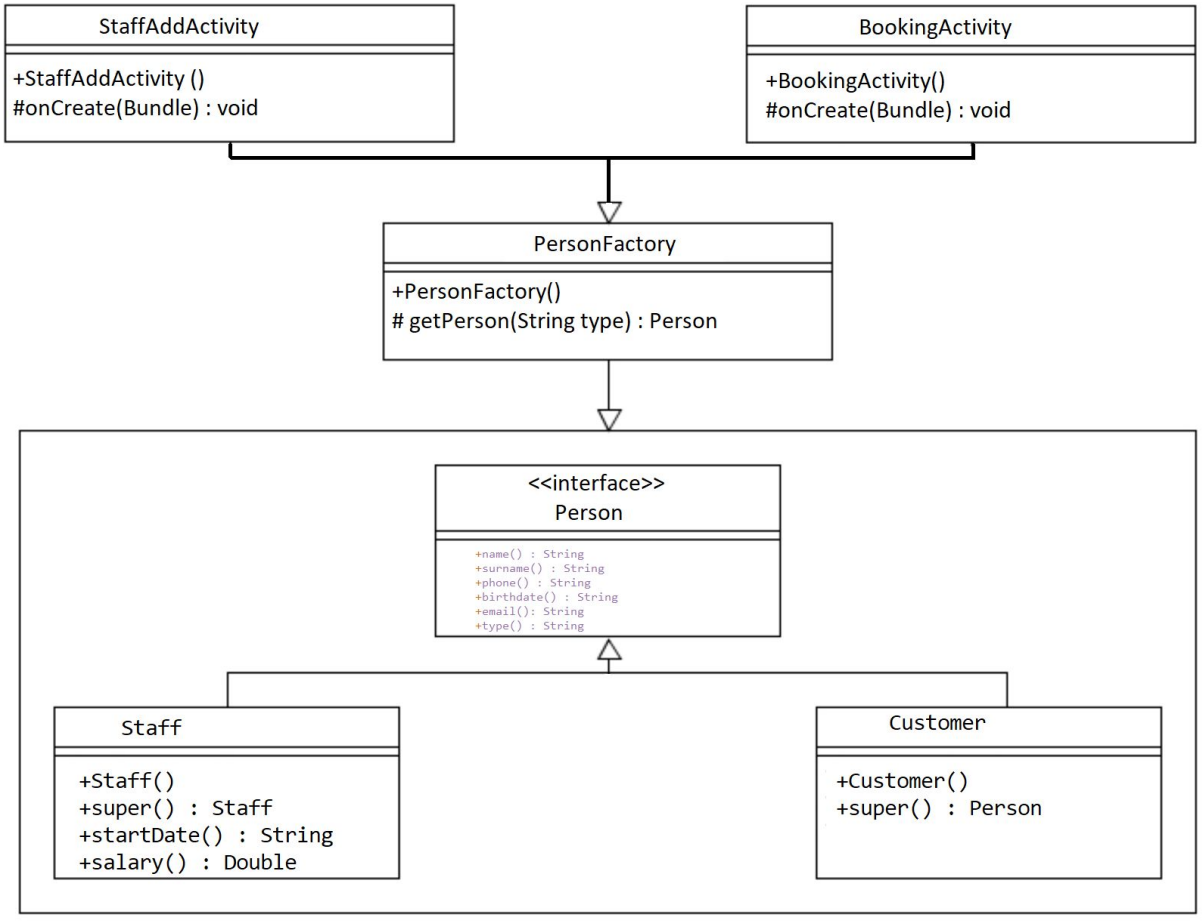
-Builder handles requests from the director and adds parts to the product.

-The client retrieves the product from the builder.

Each ConcreteBuilder contains all the code to create and assemble a particular kind of product. The code is written once; then different Directors can reuse it to build Product variants from the same set of parts.

Factory Design Pattern

In Factory pattern, we create object without exposing the creation logic to the client and refer to newly created object using a common interface. Factory methods eliminate the need to bind application-specific classes into our code. The code only deals with the Product interface.



We must notice that the last thing this operation does is call Create on the only parent class. That's because FactoryPerson class handles only type=staff and type = Customer.

Participants

PRODUCT : Person

defines the interface of objects the factory method creates.

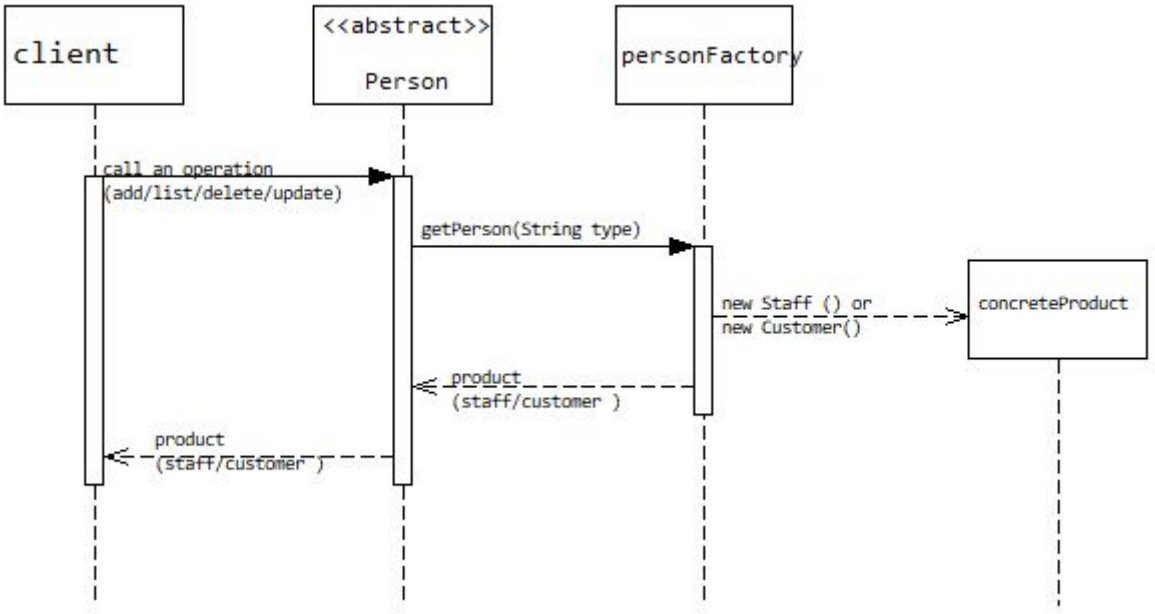
CONCRETE PRODUCT : Persons' inheritor (Staff / Customer)

CREATOR : Person()

Declares the factory method, which returns an object of type Person. Creator may also define a default implementation of the factory.

CONCRETE CREATOR : PersonFactory()

Creator relies on its subclasses to define the factory method so that it returns an instance of the appropriate ConcreteProduct.



The Factory pattern type that we used is Parameterized factory methods. The factory method takes a parameter that identifies the kind of object to create. When we call the FactoryPerson class, we must specify the person type: Staff or Customer. All objects the factory method creates will share the Factory interface.

Project Plan

3.1 Task Description

Stakeholder meetings

Stakeholders and software engineering team hold meetings to fully understand the problem and obtain the necessary information. During the project development phase, due to the extraordinary reason happening in the world, these meetings were held online and held every Thursday.

Design Models and Mockups

Designing models and mock-ups helps to provide clarity about how the project works as well as the project. Throughout the software process, these models can constantly change in line with supply and demand.

Database Creation

In order to publish the first version of our project in a short time, we used google product firebase (nosql) while the database was being created.

Firebase provides a real-time database and back-end as a service. The service provides application developers an API that allows application data to be synchronized across clients and stored on Firebase's cloud. It includes a user management system whereby developers can enable user authentication with email and password login stored with Firebase.

Android Creation

Our Android application was first designed by drawing Mockups and then coded in the Android Studio program in java language. As soon as users open the application, they will see a list of movies that are in the vision. In this way, when they are put into practice, they will be able to get what they want without getting lost. Cinema directors, on the other hand, can perform the tasks belonging to the step after logging in with the admin login at the top.

Testing

We have prepared and implemented several unit tests to apply to this program. We will consider the tests performed more broadly in the testing section. A unit test can verify different behavioral aspects of the system under test, but most likely it will fall into one of the following two categories: state-based or interaction-based. Verifying that the system under test produces correct results, or that its resulting state is correct, is called state-based unit testing, while verifying that it properly invokes certain methods is called interaction-based unit testing.

Finalization and report

All test and function processes are concluded at this stage. Reports will be created to ensure that all information and functionality are clear to make the user manual and to make it easier for employees to use the software. For this, be sure to consult the user manual on the last page.

3.2 Task Assignment

İbrahim made;

Stage 1: Description of booking seat and theatre classes.

State 2: Modelling class, use case and state diagrams.

In last stage fixed the problems. Testing the whole project and made a testing table. Fixed the UI design

Ayberk made;

Stage 1: Description of management system,

Stage 2: Modelling activity diagrams,

Last Stage: Firstly, firebase management such as queries(adding,deleting,updating,reading),designs and its connections between the java classes. Secondly, movie activities such as movie visualization and update. Additionally it's design.

Thirdly, Theatre and it's some specific functions. Fourthly, activities of booking system and it's design.

Lastly, Ticket control system and its design.Furthermore, login authentication and it's design.

Gaye made In the first stage, after the determination of the subject, she made a description of the 'Movie' category in the first draft for solution. In the second stage tasks of drawing UML diagrams, she modeled the functions required for the 'Movie' category, she started doing with sequence diagram. In the task of creating the skeleton of the classes, which is the third stage task, she continued from the same category and wrote the attributes and functions of the 'Movie' class. While doing this, she benefited from the principles of object-oriented programming. As for the fourth stage, it filled the functions determined in the

previous stage and made them available. He also created the queue data structure to hide movies and theaters. When it came to the fifth stage, CRUD operations of the Staff class were made through the Android Studio program. CRUD (Create, Read, Update, Delete). The data to be saved are recorded in real time on the Firebase database provided by google.

3.3 Deliverables and Milestones

We had five major Milestones in this project:

1- Description of the problem and solution suggestions.

2- Realization of structural design and making UML diagrams.

3- the skeleton of the code structure

4- coding and testing

5- UI-UX design

3.4 Project Schedule

Whole project was carried out in 5 stages.

1- At the first stage, we identified the structures needed by the project. These structures required us to take advantage of the following principles of Object-oriented programming:

-The structure must be designed in Encapsulation principle.

- The system must contain examples of Polymorphism and Inheritance (pure inheritance,

abstract class and interface usages) principles.

- The system must contain the usage of List, Stack or Queue data structures which you will implement.

2- In the second stage, UML diagrams were created according to the working logic of all the functions of the project to be done before starting the coding and the communication between the classes.

The diagrams created are as follows:

- Class Diagram
- Use case Diagram
- Activity Diagram
- State Diagram
- Sequence Diagram

After explanation of the diagrams created, the second stage ended.

3- In the third stage, we have created and coded the classes,interfaces and abstract classes of the project in accordance with the decisions we made at the meetings. Since we entered the third stage, we started the coding task.

4- In the fourth stage, we successfully completed the coding parts and all functions as console application. The entire project was completed in accordance with Object-oriented programming principles. We kept our data in queue data structure for a more effective use.

5- In the fifth stage, database and interfaces were added to the project. necessary white box tests were done and errors were corrected. At the end finalization and reporting.

Testing

4.1 Features to be tested

We have subjected our project to two different tests. The first of these; it was the tests we did depending on the scenarios. The second test was the speed tests we performed according to the number of data in the database. Tables containing the scenarios and results of both tests are given below.

4.2 Test Cases

1-) Login test

TEST CASE ID	TEST SCENERIO	TEST CASE	PRE-CONDITION	TEST STEPS	TEST DATA	EXCEPTED RESULT	ACTUAL RESULT	STATUS
TC_LOGIN_001	Verify the Admin Login	Enter Valid e-mail and valid password	1.Need a valid account to login	Enter e-mail	Valid e-mail	Login succesfull	Management page is shown	PASS
				Enter Password	Valid password			
				Click Login button				
TC_LOGIN_002	Verify the Admin Login	Enter Valid e-mail and invalid password	1.Need a valid account to login	Enter e-mail	Valid e-mail	A message "e-mail and password don't match" shown.	A message "e-mail and password don't match" shown.	PASS
				Enter Password	Invalid password			
				Click Login button				
TC_LOGIN_003	Verify the Admin Login	Enter invalid e-mail and valid password	1.Need a valid account to login	Enter e-mail	Invalid e-mail	A message "e-mail and password don't match" shown.	A message "e-mail and password don't match" shown.	PASS
				Enter Password	Valid password			
				Click Login button				
TC_LOGIN_004	Verify the Admin Login	Enter invalid e-mail and invalid password	1.Need a valid account to login	Enter e-mail	Invalid e-mail	A message "e-mail and password don't match" shown.	A message "e-mail and password don't match" shown.	PASS
				Enter Password	Invalid password			
				Click Login button				

Conclusion

5.1 The Problem and Solution

Cinema Booking java project report Nowadays, traditional reservation ways of cinema ticketing is dying. It's new age where technology dominates human life. With the software and technological devices, exceptions are reduced and even terminated. Also, people prefer easy, quick and safe way for every part of his life. This project is designed to meet the requirements of a cinema ticket booking system. It has been developed in Android application with java. In our project: with this cinema ticketing system; cinema companies can satisfy comfortable facilities to their customers.

5.2 The Team and the SE Project

The Software Engineering process we used was the waterfall method. however, we realized that this method is not very efficient for remote work. The workflow of this methodology is as follows:

1. System and software requirements: captured in a product requirements document
2. Analysis: resulting in models, schema, and business rules
3. Design: resulting in the software architecture

4. Coding: the development, proving, and integration of software
5. Testing: the systematic discovery and debugging of defects
6. Maintenance

5.3 The Stakeholders and Benefited

All active stakeholders took advantage of the software after the product was launched. This list is not limited to these; company shareholders, customers, project development team, senior management, department managers, consultants and much more.

5.4 The Organization's Benefits

Our organization benefits from the production of this software is the recognition of our organization. We are pleased with this project and the work we do, which will be an advertising face for our future projects.

User Manual

6.1 Software Description

People will be able to book the cinema tickets by just sitting at home or in the office. If you are planning for a surprise to watch a movie with your friends and the family then you can get the information regarding the seat availability in the particular theatre. It will help people to book the movie tickets before the release of the films. This allows you to see the show of the film.

The features that can be included in the cinema booking system area as follows:

Customer database management: The tickets that have been booked by the customers need to be stored in the database in an efficient way. The Customer can take advantage of the cinema campaign.

Theatres list: The list of the theatres that are present in the nearby areas should be displayed properly.

on show movies: The details about the upcoming and on show movies can also be mentioned and can be booked in advance.

Price of the ticket: The ticket prices should also be specified.

and etc.

6.2 How to use the Software

A few adjustments are required before you can install the program on your phone first.

Prerequisite

Following are some pre requisite for this:

Android Studio (if you don't have it, see the previous tutorial for installation of Android Studio).

USB cable

Android device

Step 1) Enable USB debugging

The very first step is to enable USB debugging on your Android device. To do this follow these steps

On your phone (or tablet) go to Settings=> About Phone

Tap Build Number 7 times, after 7th time it will say You are now a developer.

You will notice Developer's Options are now available.

Go to the Developer option and enable USB debugging

Step 2: Install USB driver

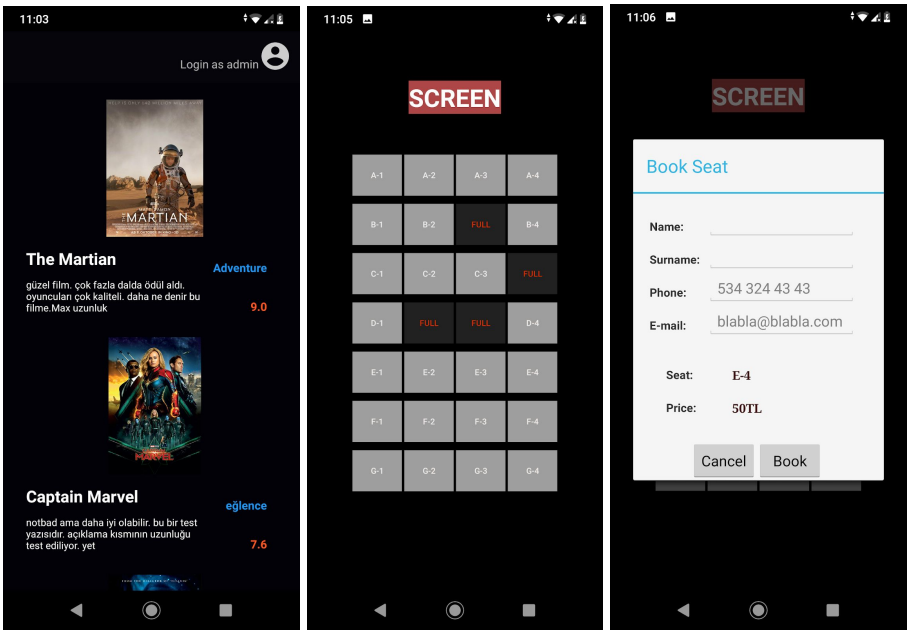
Go to the Control Panel => Device Manager then locate and right click your Android device and click Update driver software.

Note: Make sure your Android device is not sleeping while connected through USB cable.

Step 3: Run your app

No you can run your Android app. Right click on the app and click Run. Or simply select run option from the tool bar menu shown below.

As it is known, our program is a program where both movie viewers can buy tickets and Cinema companies can see some of their works about cinema.

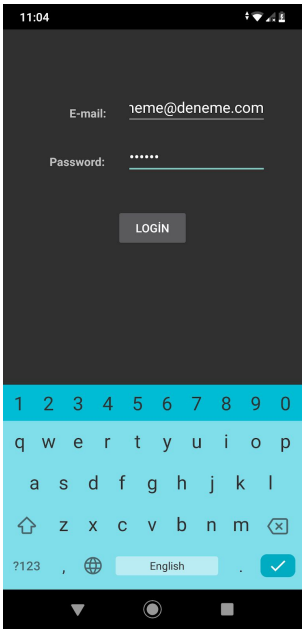


pic 1 **pic 2** **pic 3**

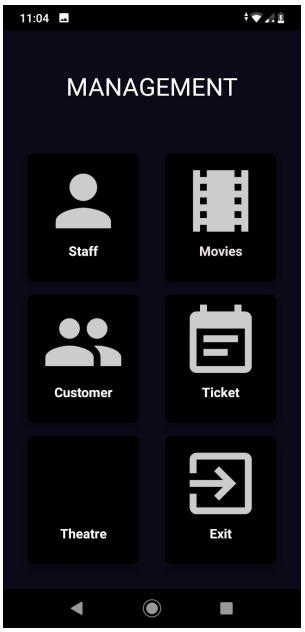
In pic 1, When the program is opened, the "movie list interface" that the customer will encounter first opens. the customer can select a movie that is in vision.

In pic 2, the seats of the selected movie are shown and the customer can select an empty seat they want.

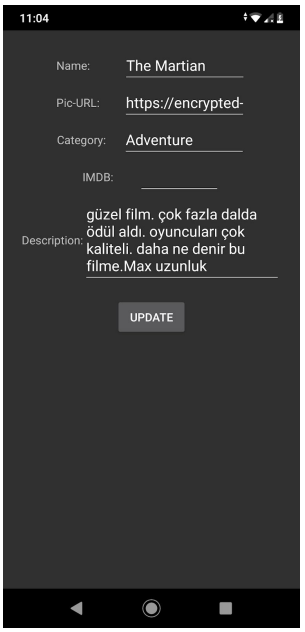
In pic 3 After the seat is selected, ticket purchases are made.



pic 4



pic 5



pic 6

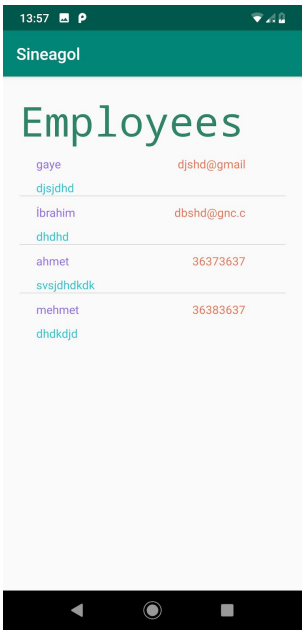
In pic 4 The cinema operator admin can log in from the "login as admin" button at the top of the "movie list interface" page. We have temporarily created an admin account. Please enter the following information:

email: deneme@deneme.com

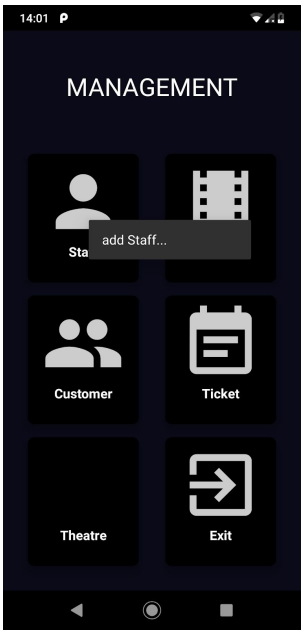
password: 123456

In pic 5 here is the interface that the administrator will use. (Some changes were made after the picture was taken.)

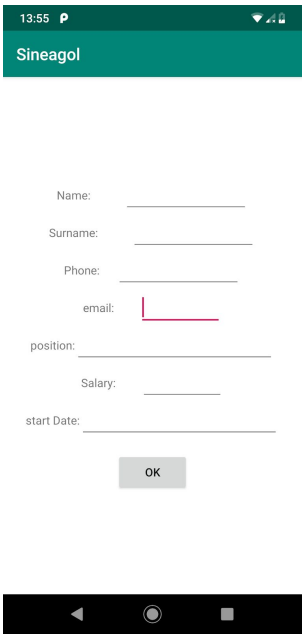
In pic 6 here, when a different movie arrives at the same theater instead of movies from the vision, the movie update.



pic 7



pic 8

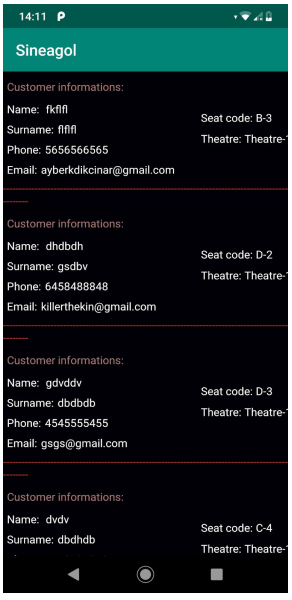


pic 9

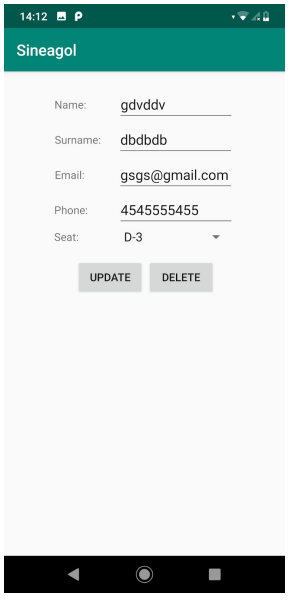
In pic 7 Staffs' information is listed in the staff section.

In pic 8 Long press the employees button to add a new employee information.

In pic 9 new employee attachment.



pic 10



pic 11

6.3 Troubleshooting Common Problems

The software will also be easily available when it comes out later in the June. The software has a minimal version for tablets for those working for the company to easily navigate through customer orders . Tablets are not required for full use the software. However, in case the deadline of project does not allow from them. the compatibility will still be available whenever they wish to provide their employees with them nonetheless.