******

***CMPS 344 Software Engineering***

***Mohammad Qrah Mohammad-202203623***

***Mohammad Ayoub-202202663***

***Jamil Saleh-202201608***

***Tamer Rifai-202201180***

***Main Project Description:***

The idea of our project is to create an app where users can purchase movie tickets online and write reviews and give them a rating, rather than relying on a small group of professional critics. This approach aims to provide the Lebanese people an insight on price fluctuations and get their money’s worth through a fairer evaluation of movies, by considering the opinions of a larger and more diverse group of people.

The app would likely allow users to search for movies and get their current prices, read reviews written by other users, and write their own reviews. Users could rate movies on a scale, such as 1-5 stars, and the app would aggregate these ratings to provide an overall score for each movie.

Overall, our project aims to provide a way for people to know current prices, get discounts and give representative and accessible evaluation of movies, by giving everyone the opportunity to have their say.

***Problem Statement:***

The issue with internet ticket costs in Lebanon is that they have been steadily rising because of the country's economic problems, which include high inflation rates, the depreciation of the Lebanese pound, and political unrest. Many Lebanese individuals now find it impossible to afford or buy cinema tickets online, which restricts their access to leisure alternatives. Additionally, it has been difficult for online ticket sellers to maintain steady rates owing to variations in demand, availability, and economic situations, which has further aggravated and confused customers. Additionally, some people are unable to afford this level of enjoyment because of current economic hardships. Our app will provide savings for families and

***Project Goal:***

Our goal is to create an app for the Lebanese people to get access to movie tickets at their current prices with special offers and discounts meant for users to afford this level of entertainment. In addition to help the customer to choose which movie he/she wants to watch by looking at the consumer reviews and then writing their own after watching the movie.

***System and Domain Review:***

Our project is not something that doesn’t already exist; however, it is going to be centered around the Lebanese market and cater towards the declining economy of the country and the purchasing power of its people. Where customers can buy their tickets online from places such as the City Center, Beirut mall and the galaxy mall. However, the reviewing side of the application can be used on a global scale where people around the world can share their thoughts and comment on what they watched.

**Project Plan:**

***SDLC Model:***

A software development life cycle (SDLC) model is a conceptual framework describing all activities in a software development project from planning to maintenance. To implement our application making sure it’s foolproof, we will use the waterfall model. After numerous meetings and discussions, we have concluded that the waterfall model is the best approach towards building our project. Firstly, all the milestones we have set are well understood. Our project is not flexible and the probability of any changes throughout the project is almost non-existent. Second, similar apps already exist on the market, however they lack the features that our application Movie Critic provides. Finally, the waterfall approach has several benefits, such as predictability, explicit stage and task definitions, and simplicity and ease of use.

**Project Organization:**

We chose to work concurrently on each step because the waterfall model is a sequential approach, which means we couldn't enter one phase until fully completing the preceding one. With the help of our team, we were able to acquire requirements, analyze them, create one stage at a time, jointly, with testing and maintenance. This, in our opinion, would expedite and improve the efficiency of each stage. Additionally, every team member is free to contribute their own ideas, which helps us diversify the project and ensures that everyone is maximizing their potential.

**Ethical Standards and Guidelines:**

As a movie ticketing system, there are several ethical standards and guidelines that we should adhere to in order to ensure that our system is fair, transparent, and respects the privacy and security of our customers.

Here are some key considerations:

* Transparency: our movie ticketing system should be transparent in terms of how it operates, what fees and charges are levied, and how user data is collected and used. Our customers should be able to easily understand how our system works and what their obligations are.
* Fairness: our movie ticketing system should treat all customers fairly, without discrimination based on factors such as race, gender, age, or socioeconomic status. This means ensuring that ticket prices are reasonable and that everyone has equal access to tickets.
* Privacy: our movie ticketing system should respect the privacy of our customers, and collect and use their personal information only for the purposes of providing the service they have requested. We should also ensure that user data is stored securely and protected against unauthorized access.
* Security: Our movie ticketing system should be secure and protected against fraud, hacking, and other forms of malicious activity. This includes implementing strong passwords, encryption, and othe**r** security measures to prevent unauthorized access.

***Feasibility Study:***

**Risk Management:**

Since we are adopting the waterfall model risk management is essential for our project. Despite being straightforward, this model carries a lot of risk and unpredictability. To avoid such risks, we will try to lookout for any risks throughout the process. We worry that movie theaters will not be compliant with allowing college students to test our application. The greatest threat is not being able to finish the app due to coding problems. To combat these problems, we will pitch our idea to movie theaters to get their approval. Lastly, if we feel a lack of the necessary expertise to create the application, we shall seek professional assistance. Our development approach includes risk management in a very serious way, so perhaps by taking these precautions we won't run into issues in the future.

**Technical Feasibility:**

Since Movie Critic requires software and applications that are already available on the market, it is simple to say that it is technically feasible.

We will be using a cross-platform app development such as Flutter. Flutter is Google's portable UI toolkit for crafting beautiful, natively compiled applications for mobile, web, and desktop from a single codebase. Flutter works with existing code, is used by developers and organizations around the world, and is free and open source.

The IDE that we will use is VS-code since its, easy-to-use code editor that provides excellent support for Flutter. We require a database to hold data because our users must be allowed to establish accounts in order to track their advancement. We shall utilize MySQL for that. We will also use a version control system such as Gitlab to allow every team member to work together remotely.

Lastly, modern project management software provides solutions to improve the management of waterfall projects, especially with software development. Smartsheet is an online project management software that we can utilize since it improves work visibility with its real time view of tasks across entire projects. We can quickly determine what stage our project is in, whether we are on schedule or in jeopardy of missing a deadline.

To correctly document and present our project ideas, we will also use Microsoft Word, Microsoft Excel, and Microsoft PowerPoint. Moreover, to create our interface we will be using bubble.io.

**Economic Feasibility:**

Since it is crucial for us to complete this project on schedule, we will outline the anticipated time frame for constructing Movie Critic. We want to spend as much time as we can making sure the waterfall model we're using is as accurate as possible.

With all the requirements gathered, the schedule will look like this:

* Requirements Gathering & Analysis: 3-4 weeks.
* Design: 2 weeks.
* Implementation: 1 month.
* Testing: 2-3 weeks.
* Maintenance: 2 weeks.

In summary, Movie Critic shouldn't take more than 4 months. Implementation, testing, and maintenance shouldn't take long if the earlier phases are done well. When it comes to the cost of developing our application, it won't be very expensive. We will have to pay for a variety of expenses in order to create this application from start to finish. They are listed below:

* Rent an office to work from: $200/month
* Hardware (laptops): $2400
* Subscriptions: $30/month
* Graphic designer: $280
* Server maintenance: $150/month

According to the calculations above, we would need a budget of about 4200$ to achieve our objectives in the span of 4 months for our app to be ready. Remember that the costs in this budget were determined based on what is typically expected in Lebanon.

**Delivery:**

We will distribute our application by making it accessible on both iOS and Android smartphones once it is finished. Delivery documentation aims to make the application more understandable for our clients. Even though the app provides a tutorial outlining the capabilities and how to use them, we will also send along some documents. But we will only submit a user guide along with our mobile application. This is because our application must be simple to use; users won't need any particular training to use it.

**Software Requirement Specifications:**

**Product Function:**

The MovieCritic system is a software application that provides moviegoers with several features for a better movie experience. The primary function of the system is to allow users to buy movie tickets online, which eliminates the need to wait in line at the cinema. The system enables users to reserve seats in advance, which is especially useful for popular movie screenings. Additionally, the system provides a movie rating and review functionality, allowing users to share their opinions and experiences about movies they have watched.

Furthermore, the system includes a point system that enables users to earn points with each ticket purchase, which can be redeemed to purchase discounted movie tickets. This feature is designed to help users save money on their movie experiences. The system also allows users to choose specific cinemas for movie screenings, making it convenient for users to select their preferred cinema locations.

In terms of software requirements, the system must have a user-friendly interface that is easy to navigate. It should also have a secure payment processing system to ensure users' financial information is protected. Additionally, the system must be highly available to ensure users can access it whenever they need it. Finally, the system must be compatible with both Android and iOS smartphones to reach the widest possible audience.

**User characteristics:**

The movie ticketing and rating system - MovieCritic - is designed to cater to users who are interested in going to cinemas but want to make it more affordable. Some of the user characteristics of the system that could help people afford going to cinemas include Families and groups, these users often go to the movies together and may find it expensive to purchase tickets for multiple people. The system offers the ability to reserve seats ahead of time and get group discounts, making it more affordable for families and groups to go to the cinema together. Students and young adults: These users are often on a budget and may not be able to afford full-price movie tickets. The system offers special student discounts and rewards for purchasing tickets, making it more affordable for this demographic to enjoy a night out at the cinema.

**Non-Functional Requirements:**

1. **Performance:**

* The system must exhibit exceptional performance, capable of handling an immense influx of users simultaneously, without any delays or performance issues. It must be able to process transactions swiftly and efficiently, even under heavy loads.

1. **Reliability**:

* The system must demonstrate utmost reliability and remain available at all times. In the event of any system failures, it must be able to recover quickly and seamlessly, without losing any data.

1. **Security**:

* The system must maintain the highest level of security, with robust measures in place to prevent any unauthorized access to the system and safeguard sensitive user data. It must comply with all pertinent data privacy regulations and guidelines.

1. **Usability**:

* The system must be exceedingly user-friendly and intuitive, with clear, concise, and user-friendly instructions on how to use it, regardless of the user's technical skill level.

1. **Scalability**:

* The system must be scalable and designed to handle rapid growth in user numbers and changing demand levels. It must be able to adjust to changing usage patterns and capacity requirements, without any interruptions or disruption to its service.

1. **Compatibility**:

* The system must be compatible with a wide range of devices, to ensure that users can access it from any platform, device, or operating system, without any compatibility issues.

**Functional Requirements:**

Functional requirements are divided into two kinds: user requirements and system requirements.

A.User Requirements:

* Users will be able to create an account through credentials.
* Users will be able to enter their credit card information to book a ticket seamlessly.
* A narrow window of 12 hours will be provided for users to initiate a refund for their ticket, with an assured refund of 80% of the initial ticket price
* Users will be able use his points to get special offers
* Users will be able to search for movies currently available
* Users will be able to write reviews on movies
* Users will be able to rate the movie on a scale
* Users will be able to create an account through emails.
* Users will be able to update account info

B. System Requirements:

* The system should provide a secure user registration process and enable users to log in using their credentials.
* The system should display a comprehensive list of movies, along with their details, such as genre, rating, and reviews
* The system should allow users to book tickets for movies and reserve their desired seats at their desired Theaters.
* The system should have a refund policy that enables users to cancel their bookings and receive refunds if necessary.
* The system should provide a platform for users to share their reviews and ratings for movies they have watched.
* The system should ensure that user data is secure and protected against potential threats, such as hacking and data breaches.

**Use case Description**

We converted all of the user requirements into use cases in order to capture the functional needs of the system from the viewpoint of the user. The following use cases were used to generate a use case diagram. The relevant use case details were provided for each: sign in, sign out, update info, buy tickets, write review, give rating, and earn points.

1. Use case Name: Sign in

* Related Use Cases:
* Generalization of: Through Credentials, Through Email, includes buy ticket, get points, write reviews, give rating and Sign out.
* Actor: User
* Actor Actions:
* 1. Press on sign in method
* 2.Enter required fields
* System Response:
* 1.Sign in successful or sign in failed.

b) Use Case Name: Sign Out

* Related Use Cases: (inclusion)
* Actor: User
* Actor Actions:
* 1.Request log out
* System Response:
* 1.Redirect to homepage

C) Use Case Name: Update account info

* Related Use Cases: (inclusion)
* Actor: User
* Actor Actions:
  + 1.Enter new data
  + 2.Press on update account info
  + Response:1.Redirect to homepage

D) Use case Name: buy ticket

* Related Use Cases: refund ticket, get points.
* Actor: User
* Actor Actions:
* 1.choose a cinema
* 2.choose a movie
* 3.choose a seat
* 4.press on buy ticket
* 5.Add credit card info
* System Response:
* 1.Reserve seat

E) Use case Name: Refund ticket

* Related Use Cases: Buy ticket
* Actor User
* Actor Actions:
* 1.press on refund ticket
* System Response:
* 1. Refund 80 percent of the ticket price.
* 2.remove reserved seat

F) Use case name: Write reviews

* Actor: User
* Actor Actions:
* 1.press on write reviews
* 2.type in your review
* System Response:
* 1.Upload review to the public

G) Use case name: Give rating

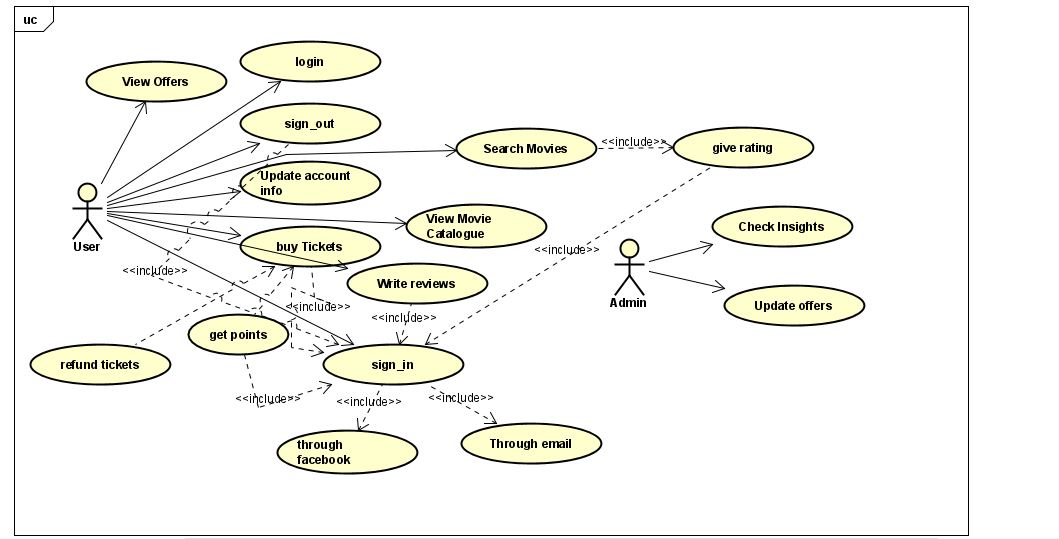
* Related Use Cases(inclusion)
* Actor:User
* Actor Actions:
* 1.Press on give rating
* 2.choose from the rate scale

H) Use case name: View offers

* Related Use Cases:None
* Actor:User
* Actor Actions:
* 1.Press on View offers
* 2. Choose desired offer.
* System Response
* 1.Display New offers
* Apply the offer on the users account.

**Use Case diagram:**

We can better communicate a system's functioning by using the use case diagram. Our use case diagram contains all the aforementioned use cases. Although every use case represents an important functionality of our system, the most important use cases are: give rating, buy ticket, write reviews, and get points. The reason behind their importance is that these features are what make MovieCritic unique. A movie ticketing and rating system allows people to easily book tickets for movies online or through an app, without having to physically visit a theater. This saves time and effort and allows people to plan their movie-going experience in advance.Lastly, get points use case refers to functionality in a movie ticketing and rating system where users can earn points by engaging with the platform and redeem those points to buy tickets at a discounted price. The use case diagram contains examples of the core relationships that could be found in a use case diagram. Sign in is a generalization of through credentials and through email since a user can sign in through multiple options. Give rating, write reviews, buy tickets, get points, and sign out are inclusion of sign in since a user cannot do those things if he/she is not signed in.



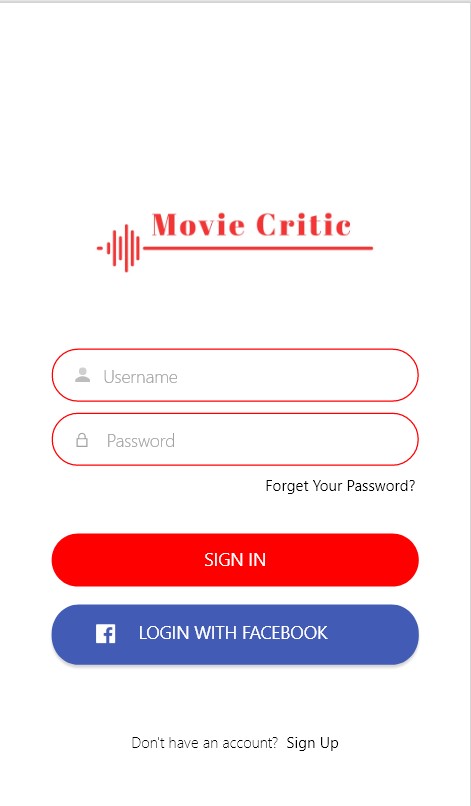
**Project Design**

**User interface Prototype**

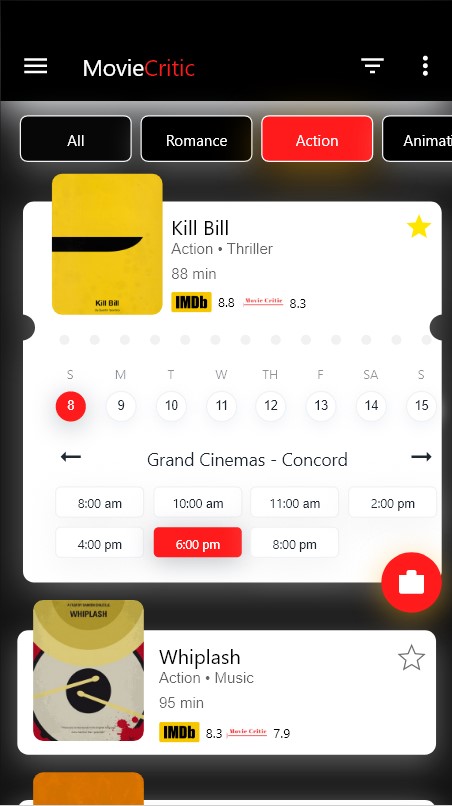
We built a prototype so that we could assess which elements are practical and which ones require improvement. The ability to test workflow with a variety of situations to make sure they're acceptable and to make sure the most frequent scenarios are optimized for little effort while being used is one of the main benefits of creating a user interface prototype. The MovieCritic has the pages listed below. Just simply looking at those UI designs, users can easily see how to perform our use cases:



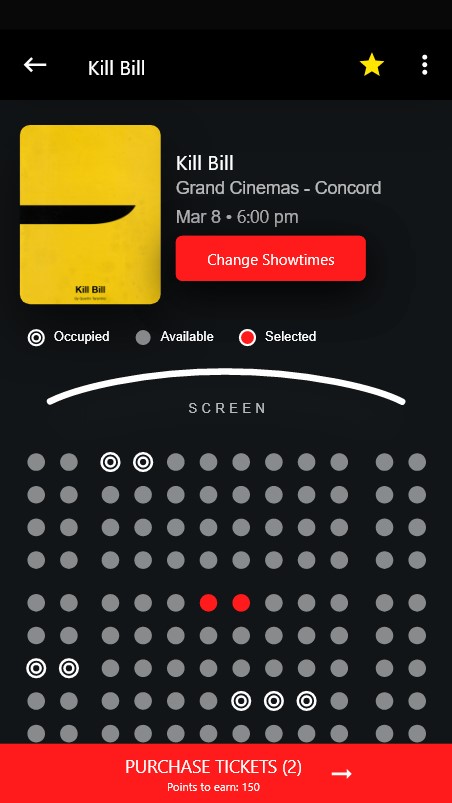
A) Startup Page



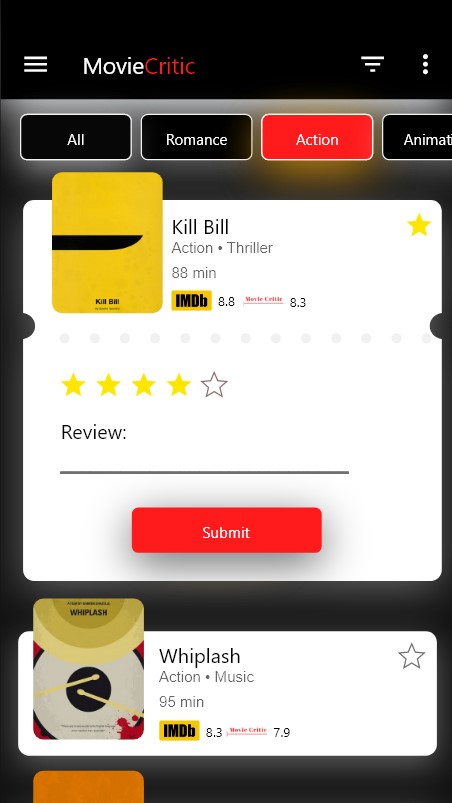
B)Log in page

`

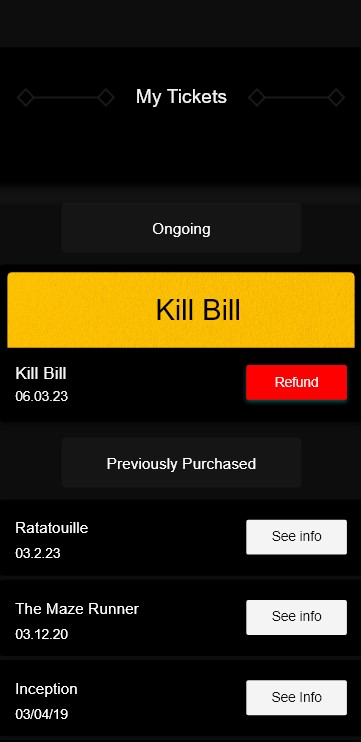
C) Home Page



D) Reservation



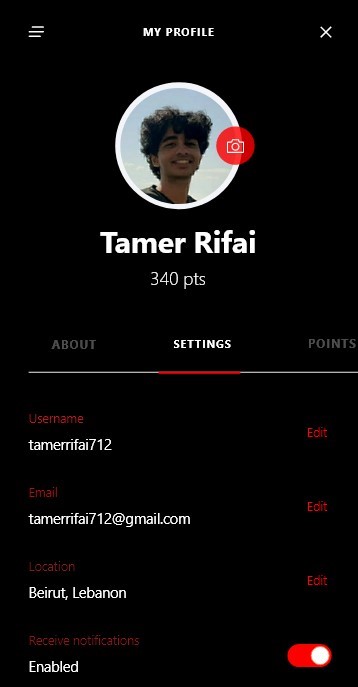
E) Review



F) Refund

D) Reservation

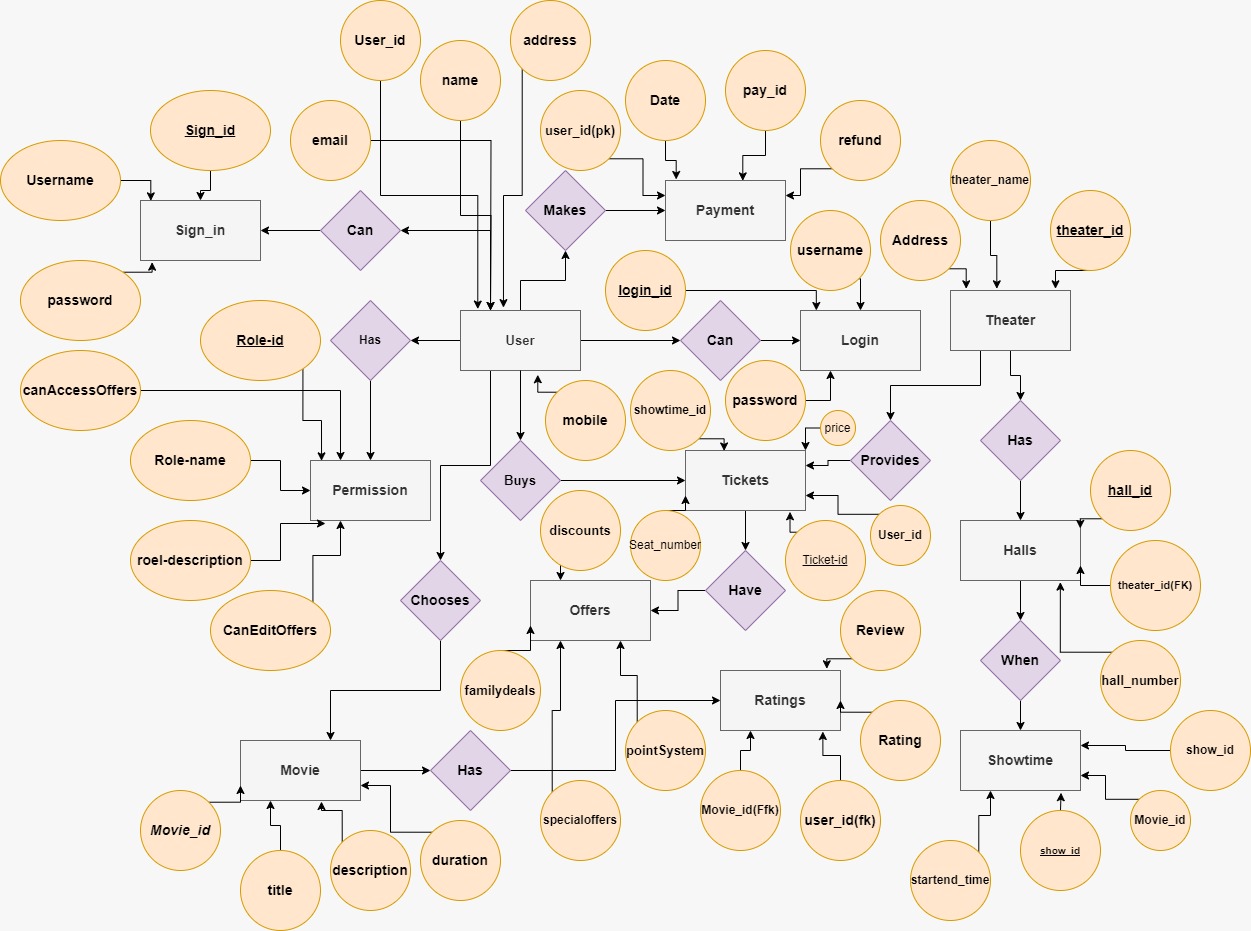
C) Home Page



G) Profile

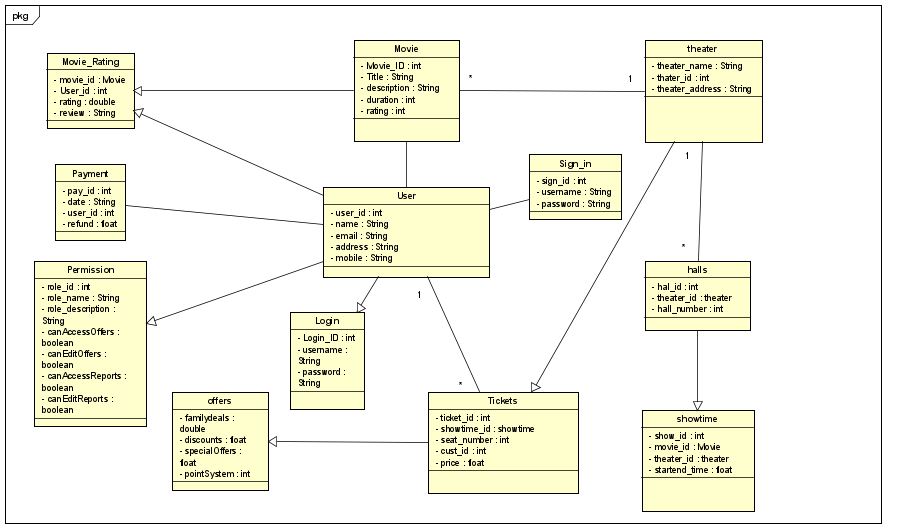
**Database Diagram**

MovieCritic requires a database to store our user’s data. Below is an EER Diagram of how the data is stored.

****

**Class Diagram:**

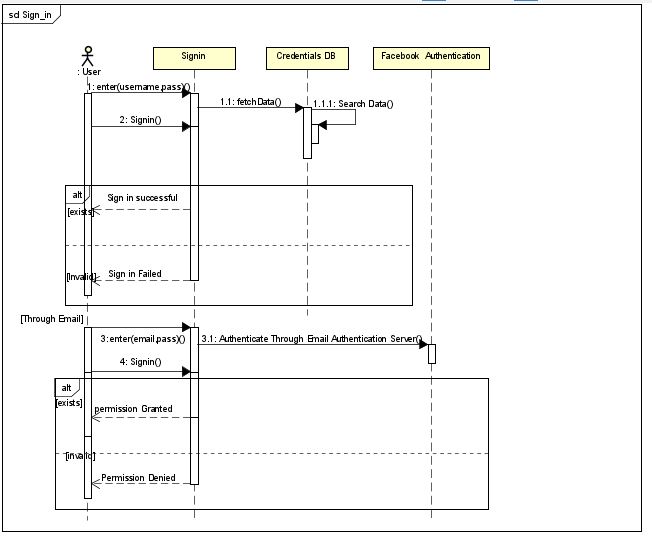
The class diagram is similar to the database diagram There are also twelve classes:User,Movie,MovieRating,Theater,SignIn,Halls,Payment,Permission,Login,Offers,Tickets,Showtime.The User has a one-to-man relationship with data just like in the database diagram.



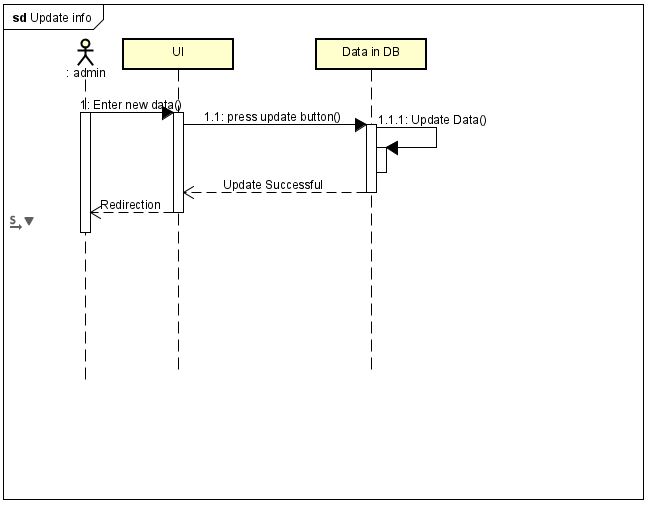
**Sequence Diagram:**

MovieCritic features a wide range of use cases, each of which has been thoroughly described before. During our lab sessions, we created a sequence diagram for each use case in order to clearly depict a user's progression through the system. Several sequence diagrams that show the dynamic nature of the user's engagement with the system are shown below.

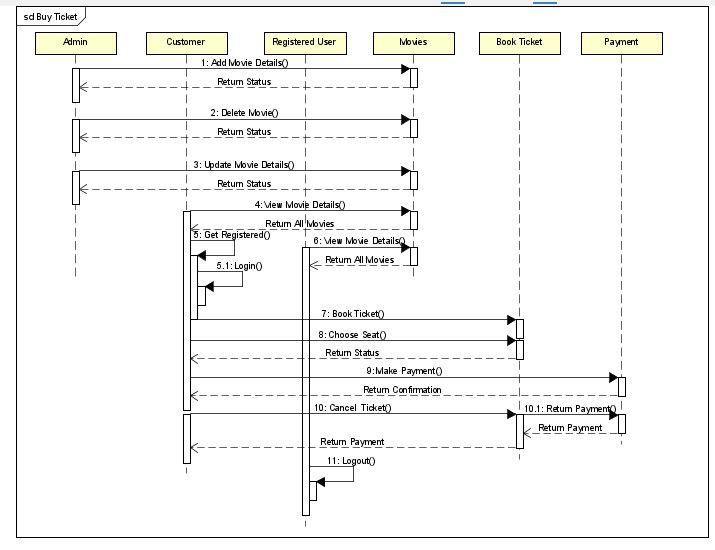
* Sequence diagram for “Sign in” – through credentials or through email



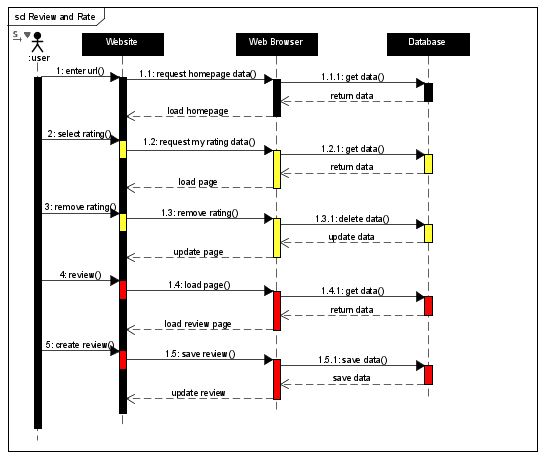
* Sequence diagram for “Update Info”



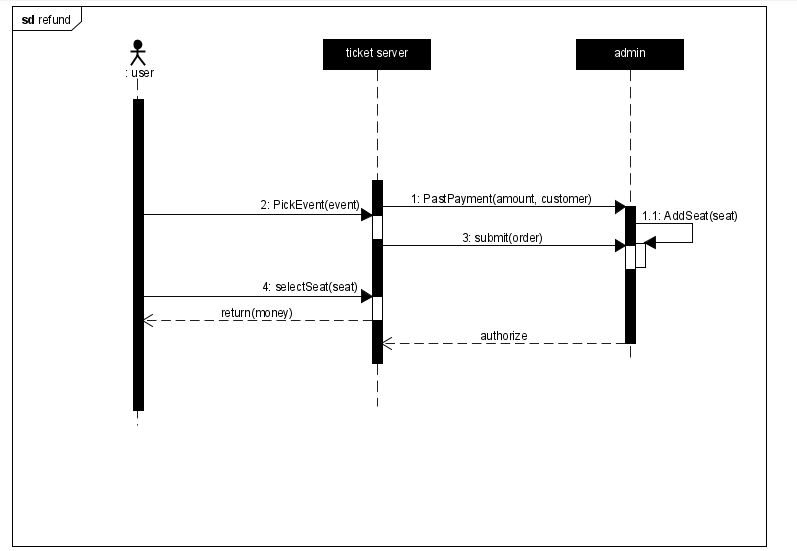
* Sequence diagram for “Buy Ticket”



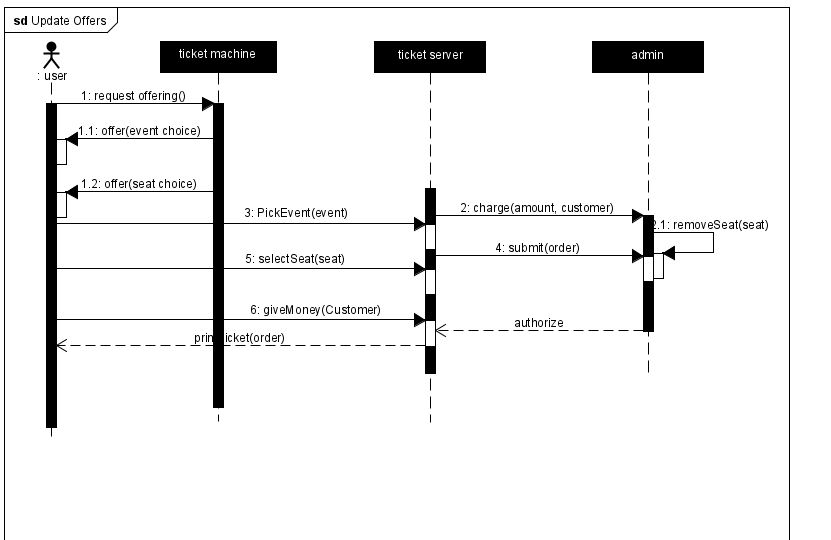
* Sequence diagram for “Review and Rate”



* Sequence diagram for “refund”



* Sequence diagram for “Update Offers”



**Methodology:**

**Implementation:**

We will utilize Flutter to implement our application, as we already described in the Technical Feasibility section. We selected Flutter because it offers a wide variety of advantages for speedy development and more user-friendly mobile applications. We want to construct cross-platform mobile applications. Rather than using kotlin to develop our application on android and swift for ios

Since Flutter was created using the Dart programming language, it enables native apps for iOS and Android to use the same scripts without any changes.

We will utilize VS code since it is free easy to use code editor that offers top-notch support.

We chose MySQL for our database since it has several benefits over other database management systems. It makes easier efficient database administration through software integration. It is a strong, dependable, and stable relational database system with cutting-edge capabilities including high performance, data security, and scalability.

**Testing:**

After implementing a movie ticketing and rating system, it's essential to conduct thorough testing to ensure that the application works as expected and meets the requirements of its users. First Functional testing, Functional testing verifies that the application's primary features, such as purchasing cinema tickets, rating films, and posting reviews, are operating as intended. All conceivable situations should be included in test cases, including legitimate input, unsuccessful purchases, and handling of errors. Second, User experience testing, User experience testing focuses on how the application feels to use. This includes testing the application's responsiveness, ease of use, and overall usability. User testing can be conducted with real users or using simulated users to ensure that the application meets user expectations. Third Performance testing, this type of testing ensures that the application can handle the expected user load and that it performs optimally under various conditions, such as high traffic volumes or slow internet connections. Fourth Security testing, Security testing ensures that the application is secure from potential threats such as hacking, data breaches, and unauthorized access. This includes testing the application's authentication and authorization systems, encryption methods, and error handling. Fifth Compatibility testing, Compatibility testing ensures that the application works as expected across different devices, operating systems, and web browsers. This testing should cover a range of popular devices and configurations to ensure that the application works correctly on all platforms.

By accomplishing these testing goals, the movie ticketing and rating system can be validated as a reliable and high-quality application that meets the needs of its users.

**Maintenance:**

As previously mentioned, before we will be using gitlab for version control.

We will require a version control system to help us manage and to facilitate teamwork since our team consists of several people and we will all be working on each stage of the project at the same time. Gitlab was our choice because it helps us deal with the misunderstanding that frequently results when several members of our team are modifying the same files at once. It makes it easy to see any modifications that any team member made. In this manner, everyone on the team will constantly be informed.

**Conclusion:**

In conclusion, we will go through a drawn-out process of gathering requirements and creating models in order to develop our application, MovieCritic. After that, MovieCritic will be applied to Flutter. We will test and maintain our application when everything is finished to ensure its security. MovieCritic will be prepared to launch on the market for our users after it has completed each step. To ensure that everyone has access to it, it will be available on both Android and iOS smartphones. We believe that by establishing MovieCritic, we will help family and friends to afford this level of entertainment and help cinemas to bounce back.