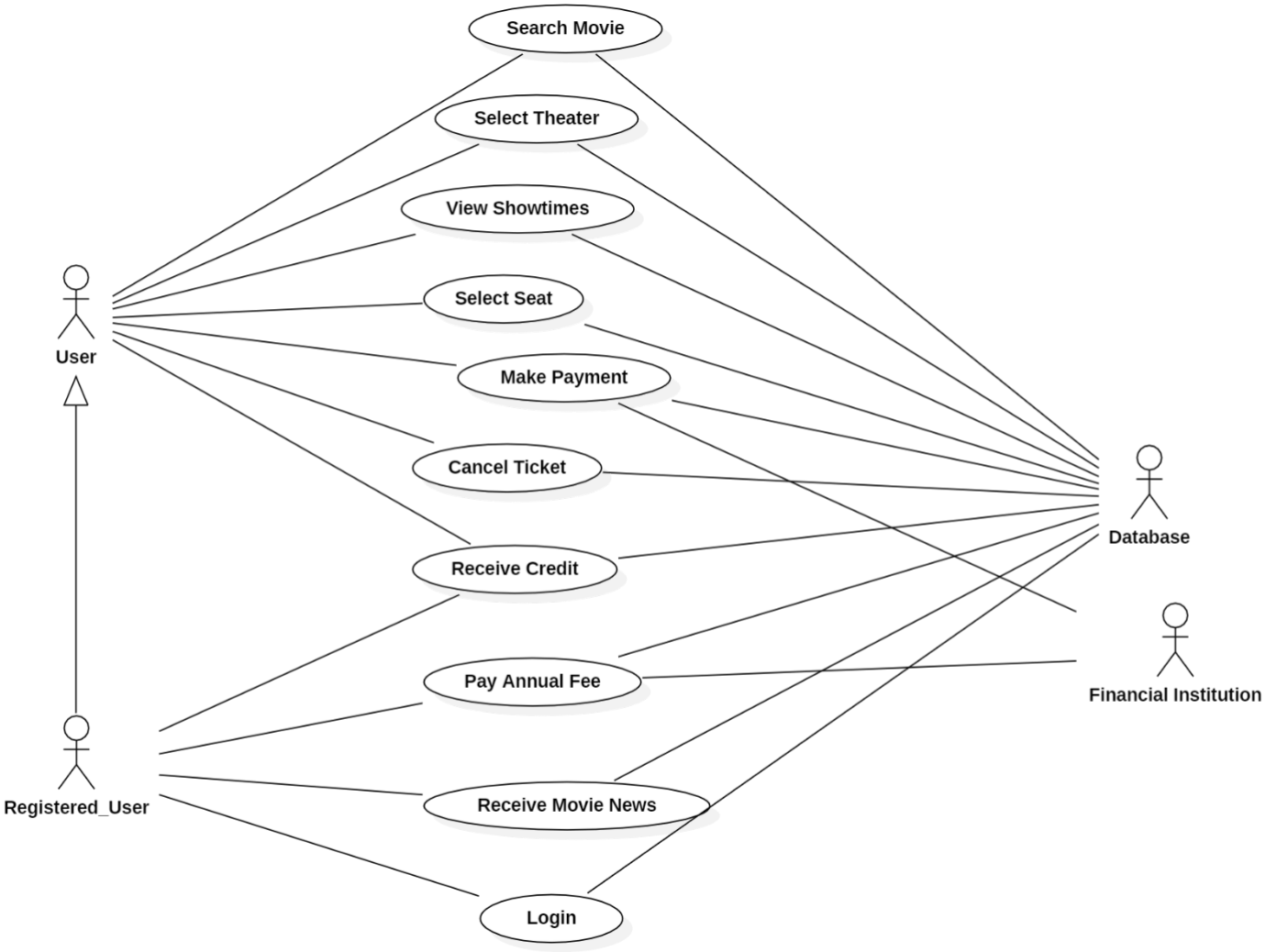
ENSF 619 Term Project
Design Phase

Group 14

Team Member:
Haixia Wu,
Jenny Tong Xu,
John Van Heurn,
Javier Vite

Contents

1. Systems use case diagram



1.1 Actors

1. User

User are those people who use Movie Theater Ticket Reservation APP to buy movie' s tickets.

2. Registered User

Registered User is a subclass of User. The personal and credit-card information of the Registered User are saved on the database, making it easier for them to buy movie tickets using Movie Theater Ticket Reservation APP. They have some privileges (no movie cancellation fee, receive movie new by email, etc.) by paying annual fee.

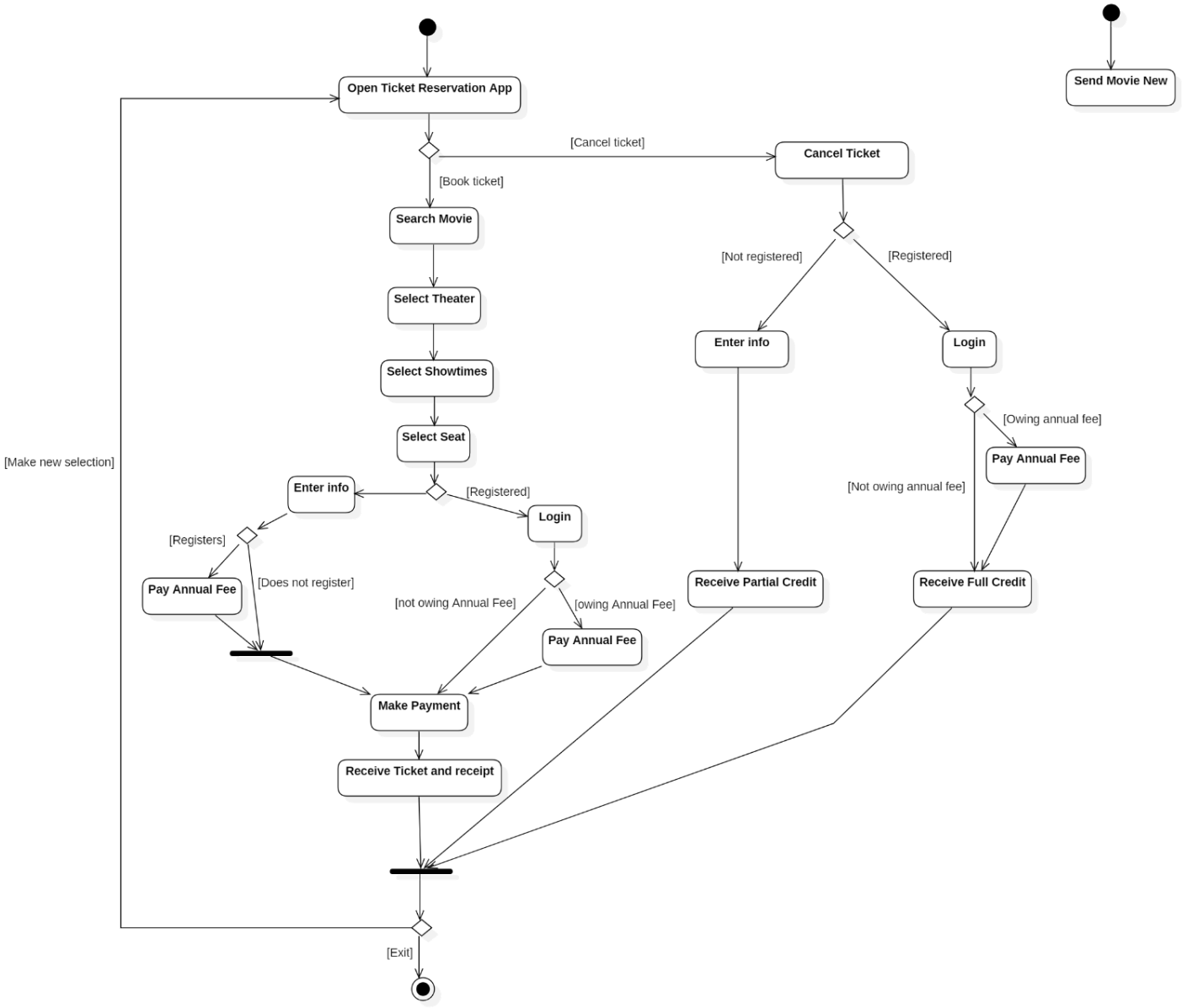
3. Database Engine

Database Engine is the underlying software component that recognizes and interprets SQL commands to access a relational database and interrogate data. It' s also referred to as a SQL database engine or a SQL query engine. There is a relational database that store the information related with Movie Theater Ticket Reservation APP. Database Engine helps users to interact with the relational database.

4. Financial Institute

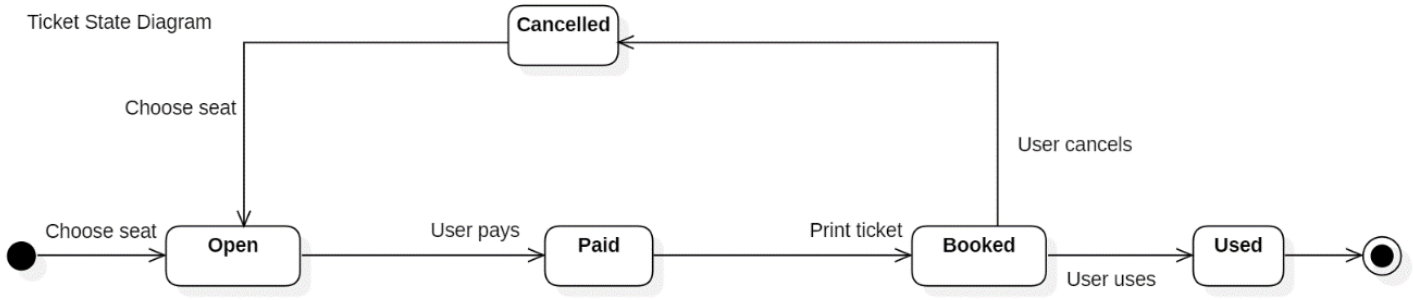
Financial Institute is referred to banks that offering banking and investment services to customers here. If User actors want to make a payment to a movie theater, the banks act as an intermediary between them.

2. Systems activity diagram

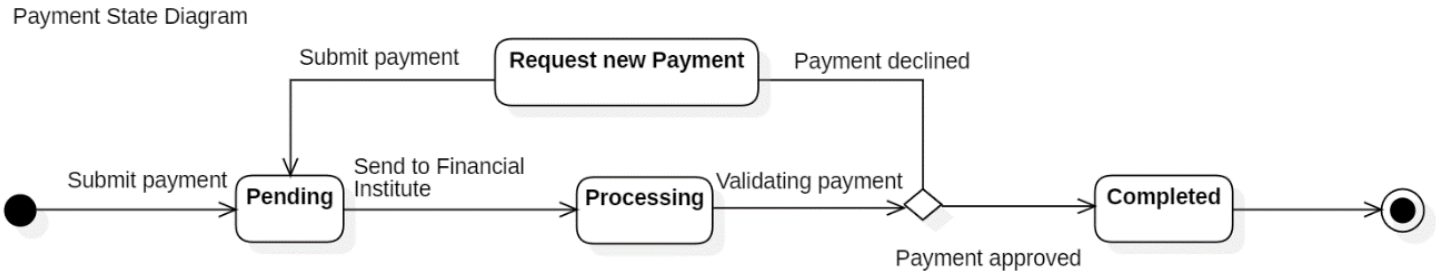


3. A state transition diagram

3.1 Ticket object



3.2 Payment object



4. A detailed "Scenario" for each use case

Notation: **candidate-objects** **candidate-operations**

1. Search Movie

This use case involves the user browsing the movie-catalogue. The user can choose to view-all-movies or search-movie and the program (booking-manager) will return a list of movies that matches the user's criteria. This use case ends with user selecting a movie (front-end).

2. Select Theater

This use case begins when the user has already selected the movie. At this point the user will browse the theater-catalogue. The user can search-theaters-for-movie and the program (booking-manager) will return the list of movie theaters where the movie that the user has selected is playing. The user can also search-theater and the theater-catalogue will return a list of theaters where the selected movie is playing and that matches the user's criteria. This use case ends with user selecting a theater (front-end).

3. View Showtime

This use case begins when the user has already selected the movie and the theater, the next step is to select a showtime. The front-end allows user to select a date for the upcoming week, and the booking-manager will get-showtimes for the user to select a showtime for the selected date (front-end).

4. Select Seat

This use case begins when the user has selected a showtime and decides to proceed. The booking-manager will display the seat-map for the theater-room where the movie will

be played. The front-end allows users to select a seat, provided that the selected seat has not been booked.

5. Make Payment

This use case begins when the user has selected his seat or is asked to pay annual fee to renew or activate their registered-user account. The booking-manager get-ticket-price from showtime and has either created or retrieved his user profile (discussed in more detailed in the Scenario "Login"). If user choose to proceed with the payment, the customer-manager send the transaction request to the bank-manager which will process-payment by validating user's bank and credit-card info and deduct the ticket price from the user 's credit-card. Once the transaction is approved by the financial institution, booking-manager book-seat (updates seat) and creates movie-ticket. Print-ticket sends the ticket info to user (via email).

6. Cancel Ticket

This use case begins when the user requests a cancellation and either created or retrieved his user profile (discussed in more detailed in the Scenario "Login"). The booking-manager validate-booking (check if booking exist) and verify-cancellation (check if the cancellation request is made up to 72 hours prior to the show). If cancellation can be processed, the booking-manager will cancel-ticket (remove ticket and updates the seat).

7. Receive Credit

This use case begins after the user has completed his movie cancellation. The customer-manager will provide a Voucher to users at 85% of the movie cost with a one-year expiration date. For registered user, the system will provide a Voucher for 100% of the

movie cost. This is done with a create-voucher function.

8. Pay Annual Fee

This use case begins after the user has logged-in to his account. The customer-manager, which tracks user information, check-fee-renewal-status for the registered-user. If payment is required, it will prompt-annual-fee which asks the user to pay a $20.00 annual account fee. In addition, if user choose to register-account, after entering their information, the program will prompt-annual-fee to set up account. If the user chooses to proceed, they will proceed with payment (refer to use case "Make payment" for more details).

9. Receive Movie News

Customer-manager have an operation to regularly send-movie-news before movie announcements to registered-users via email.

10. Login

The use case is used for booking tickets and occurs after book-seat and before process-payment. It is also used for cancelling ticket after validate-booking (check if booking exist) and verify-cancellation, and before cancel-seat. The user information is managed by customer-manager; He has the option of log-in (if he is a registered-user); otherwise, he can register-user or create-temp-account (user).

Candidate Objects

- Movie Catalogue
- Theater
- Seat Map
- Voucher
- Credit Card
- Booking Manager
- Movie
- Showtime
- Seat
- User
- Bank
- Customer Manager
- Theater Catalogue
- Theater Room
- Movie Ticket
- Registered User
- Bank Manager

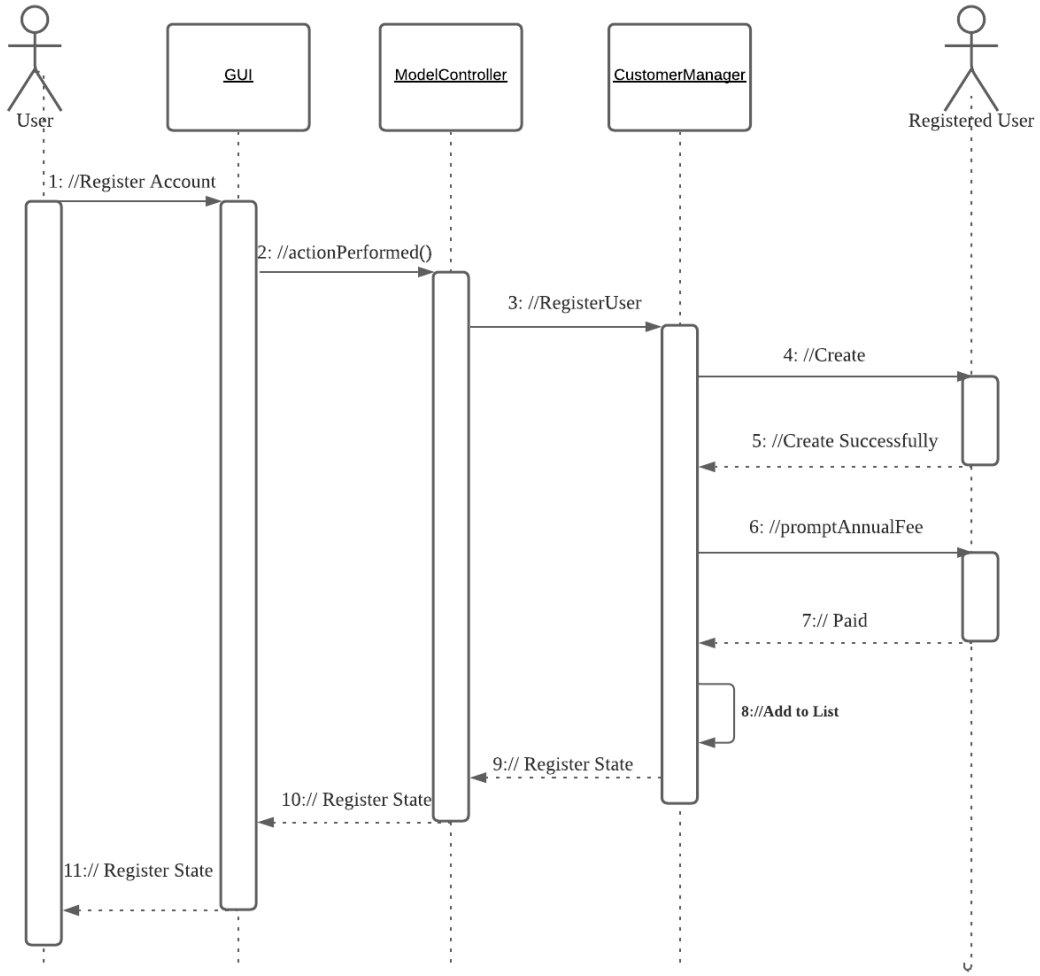
Candidate Operations

- view-all-movies
- search-theater-by-name
- get-ticket-price
- verify-cancellation
- create-temp account
- send-movie-news
- process-payment
- search-movie
- get-showtimes
- book-seat
- cancel-ticket
- log-in
- check-fee-renewal-status
- search-theaters-for-movie
- Display-seats
- validate-booking
- register-user
- print-ticket
- create-voucher

5. Systems interaction diagram

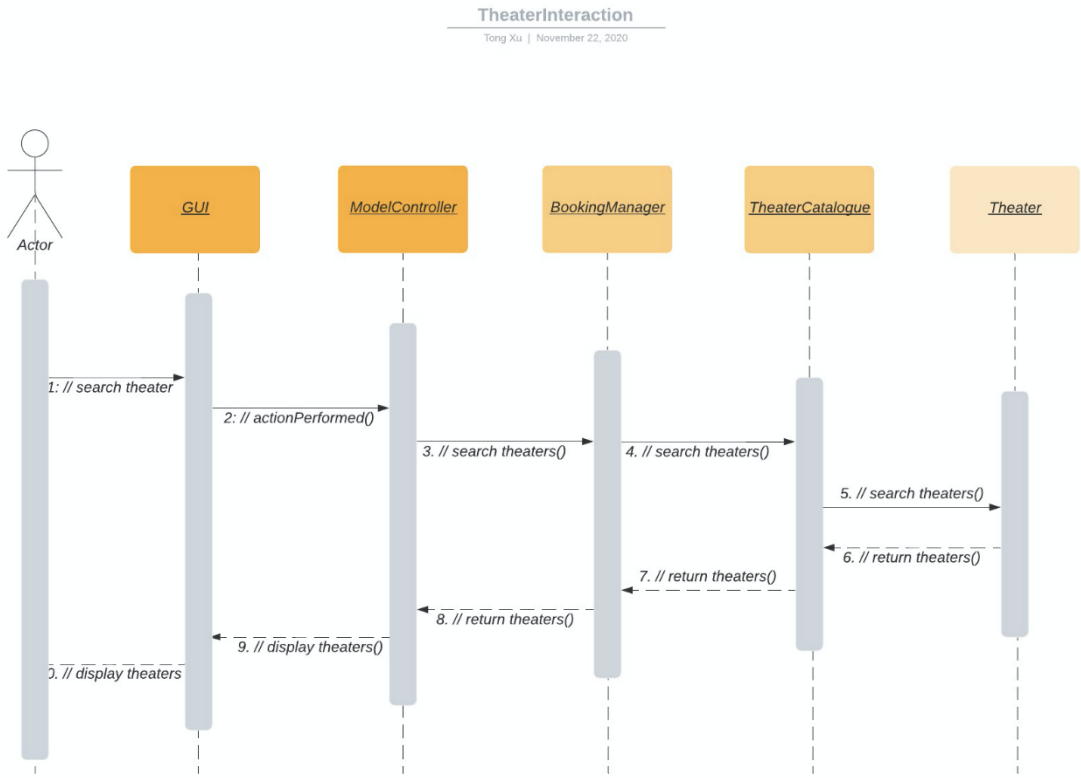
5.1 Haixia Wu

Register for Account

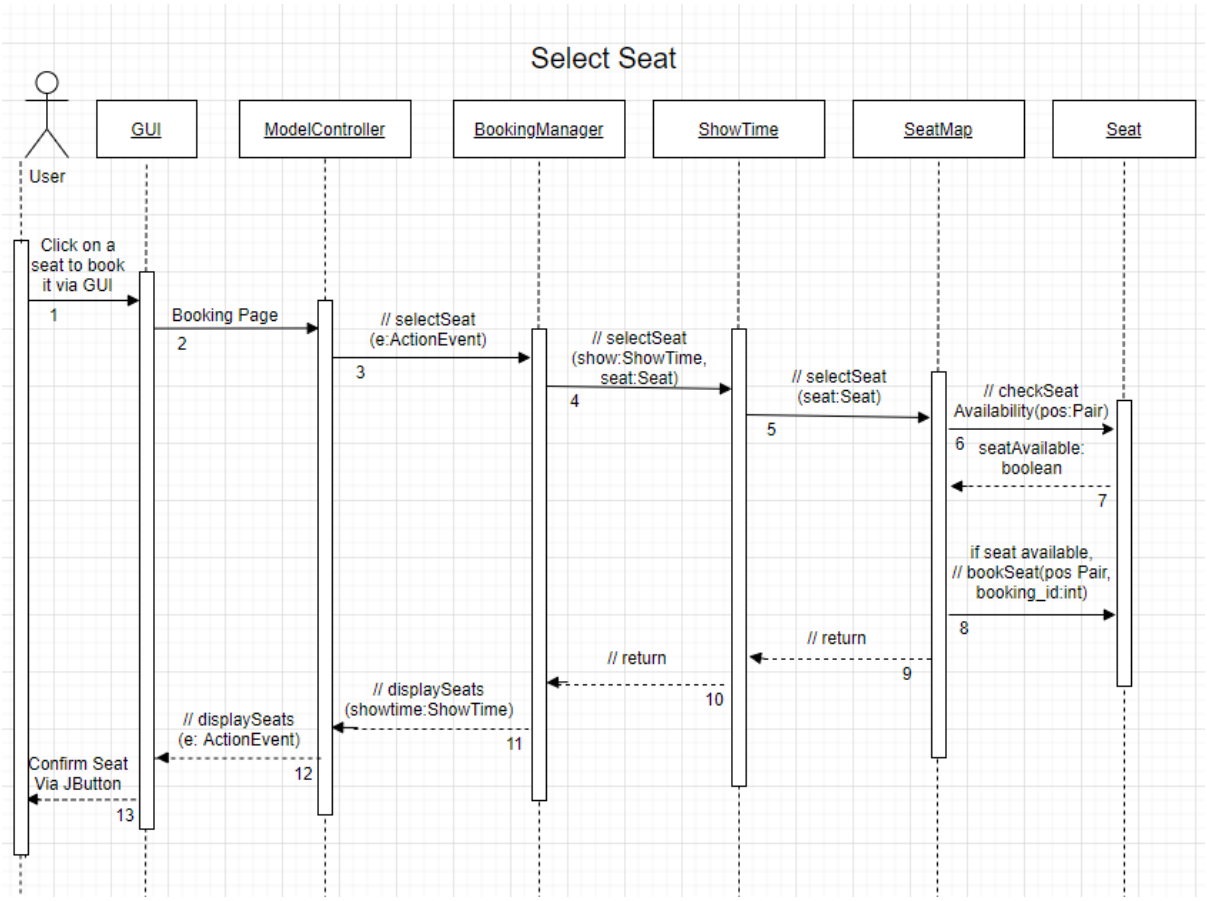


5.2 Jenny Tong Xu

Select Theater

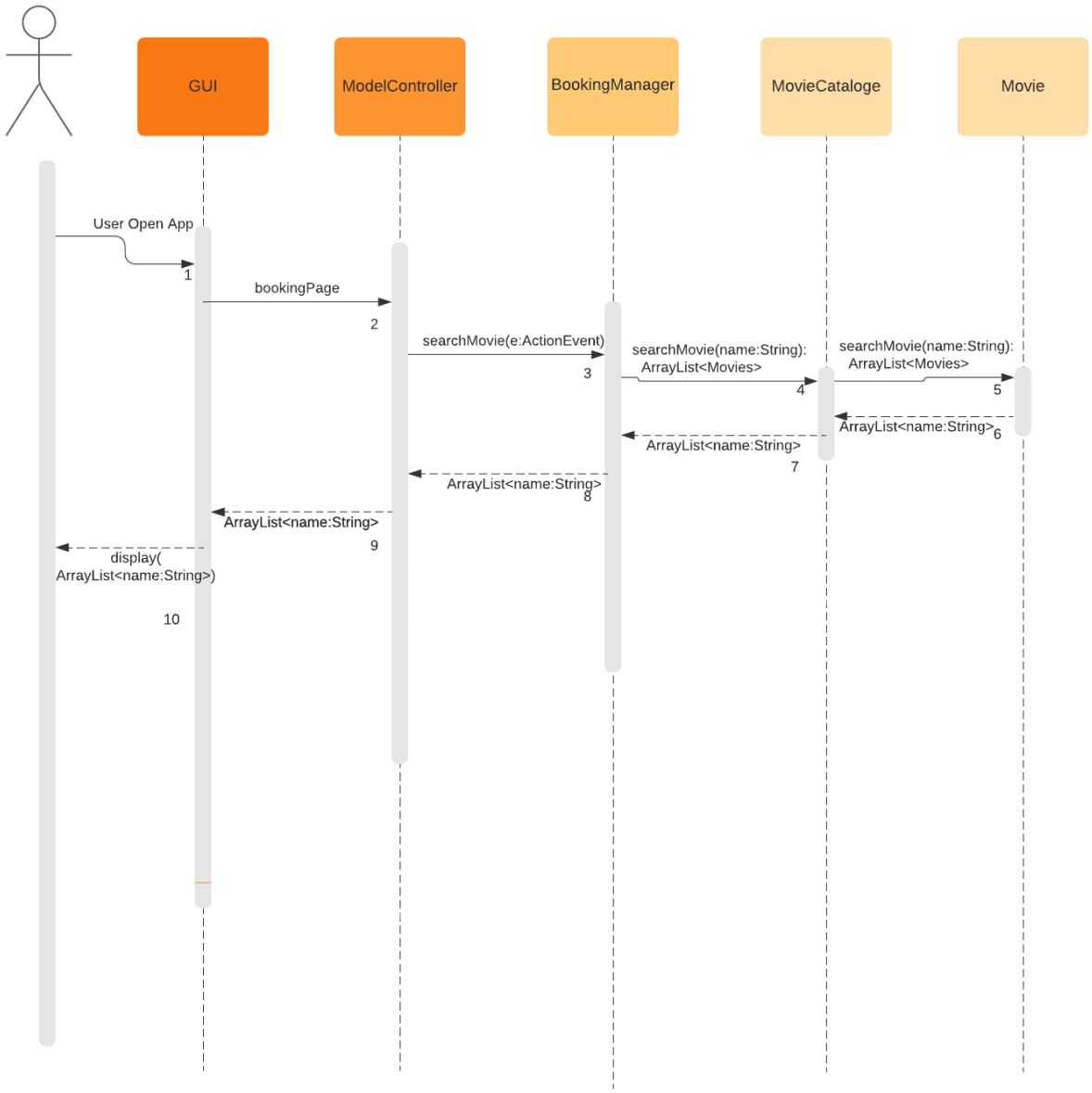


5.3 John Van Heurn



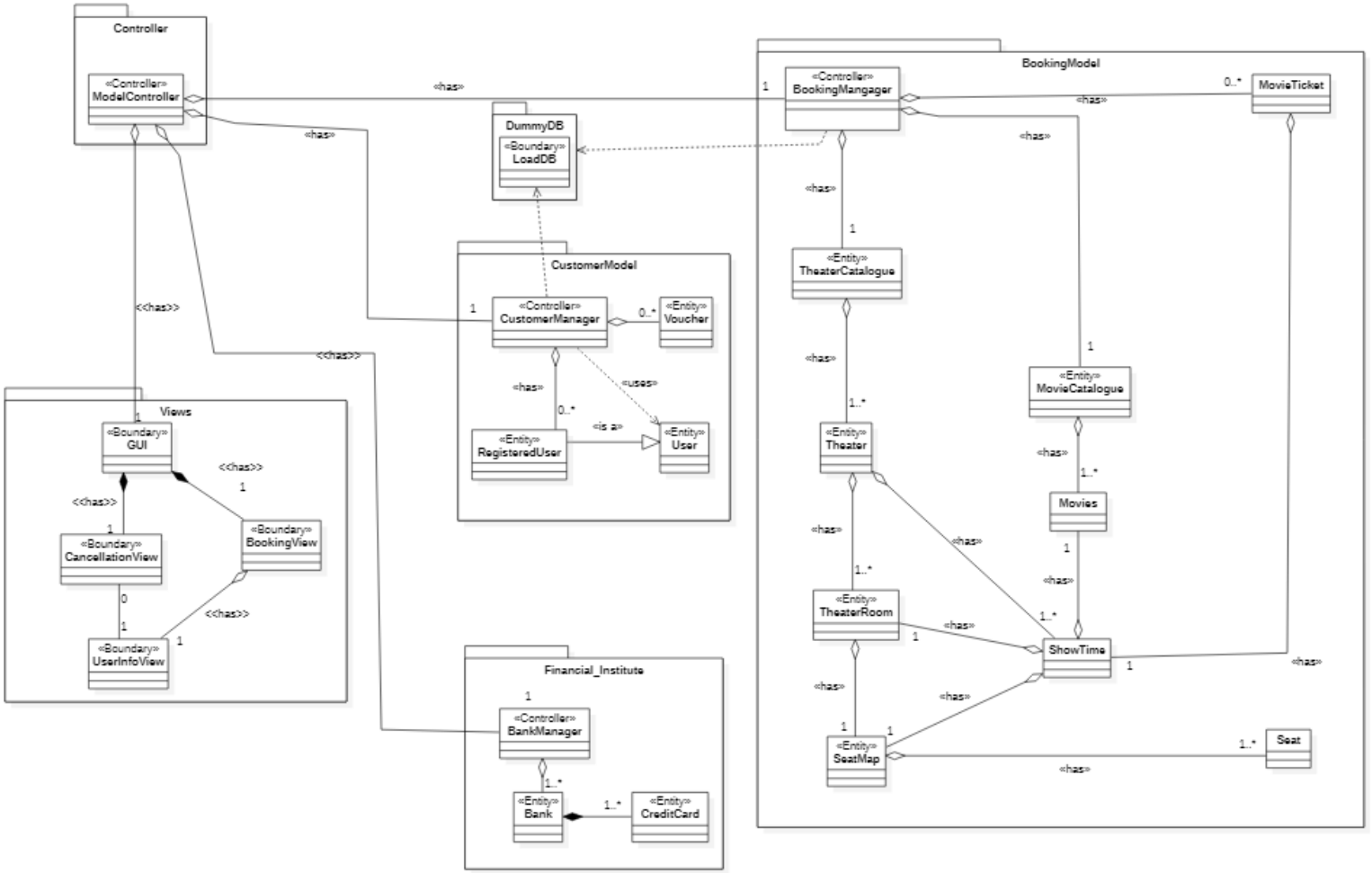
5.4 Javier Vite

Search movie



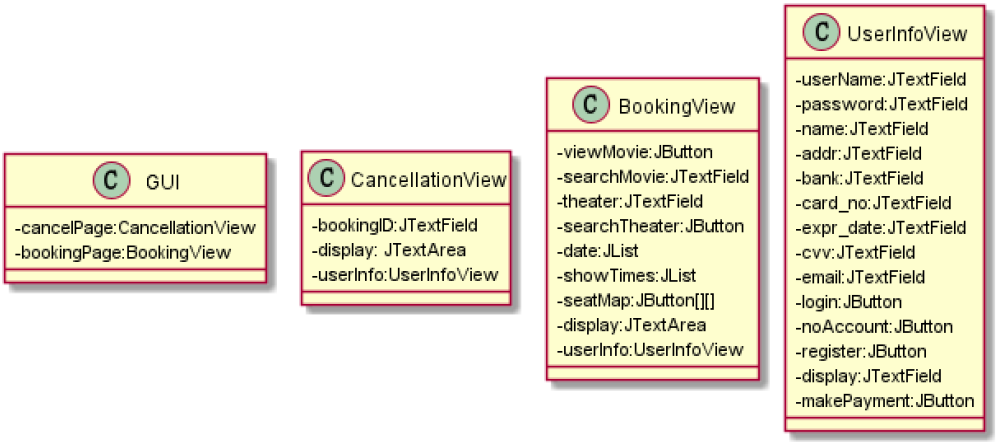
6. A Design Level Class Specification

6.1 A class diagram without attributes and behavior that only shows the class name and the relationships among them

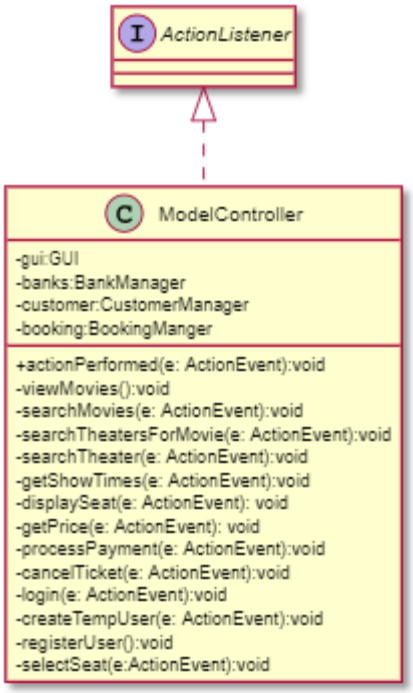


6.2 A class diagram with no relationships (no lines), only showing the class details: attributes and behaviors

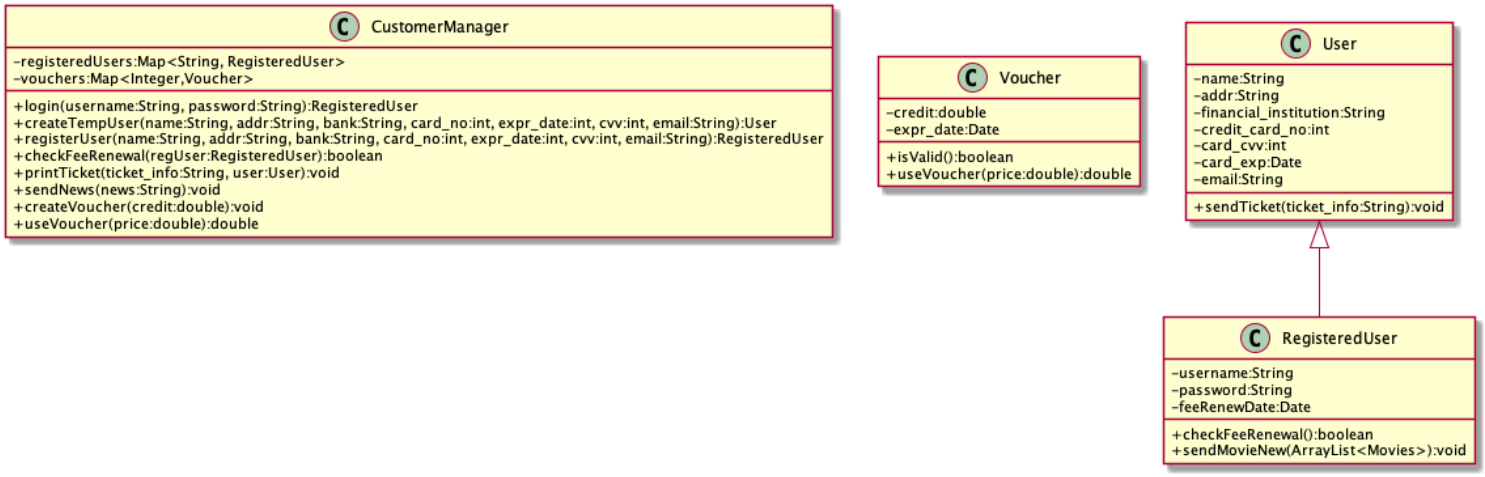
1. Views



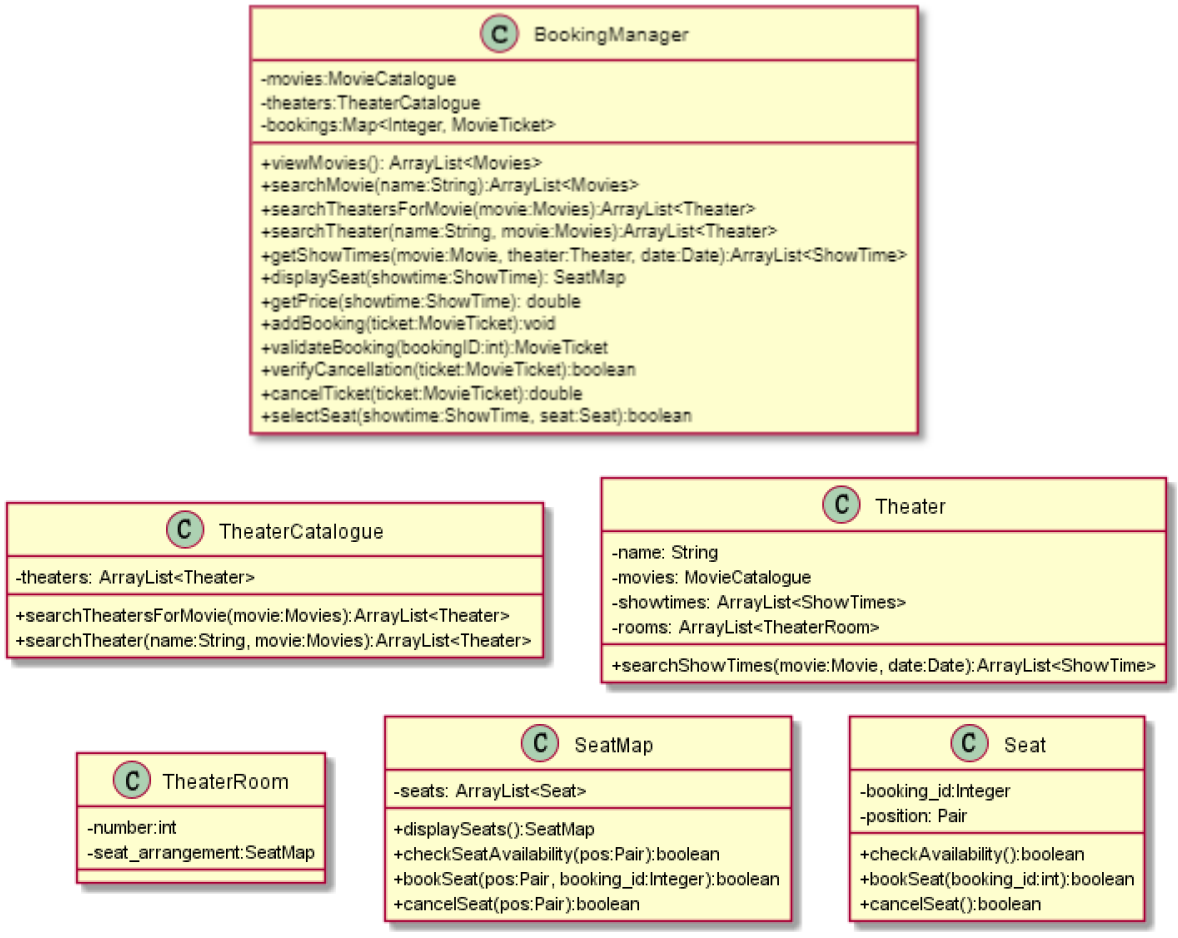
2. Controller

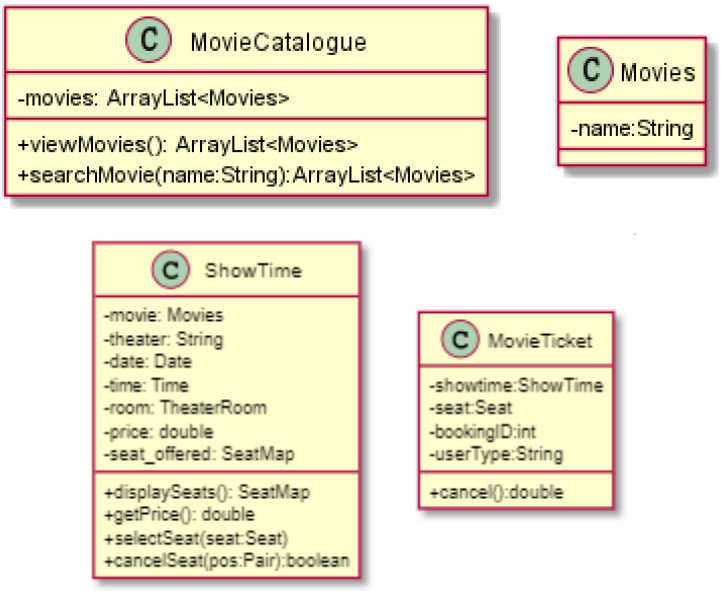


3. CustomerModel

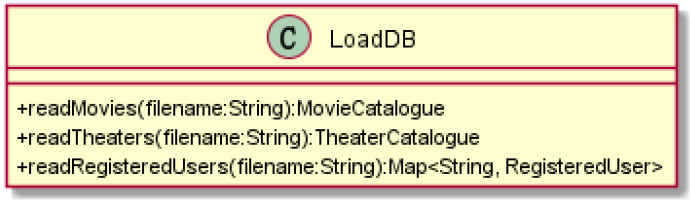


4. BookingModel

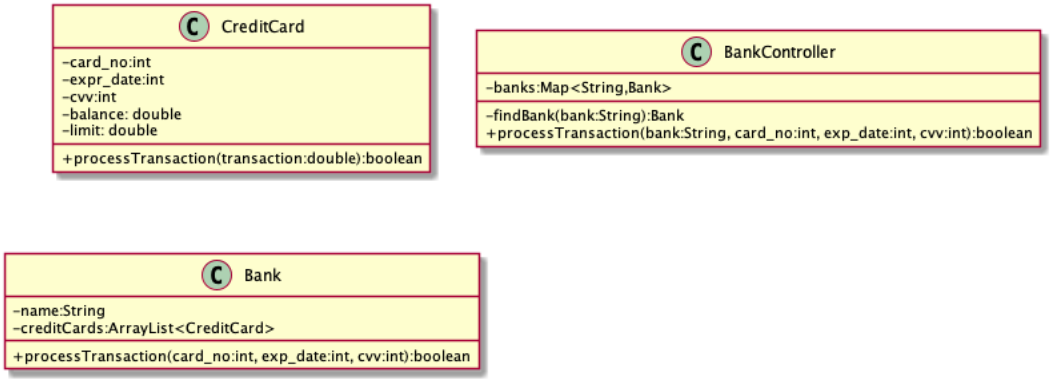




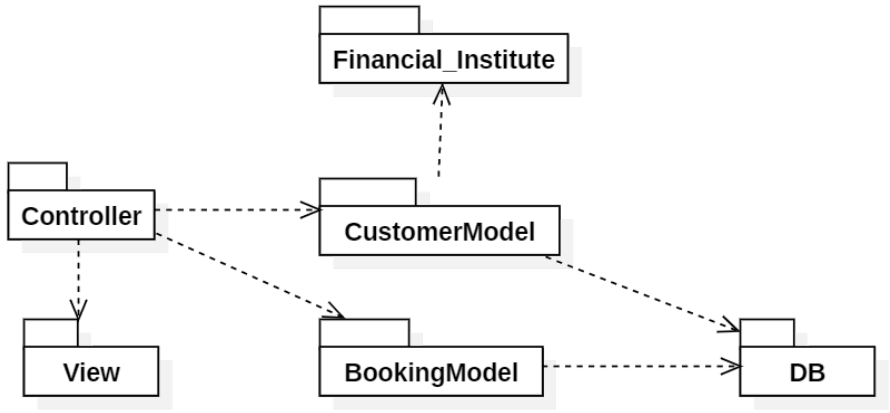
5. DummyDB



6. Financial_Institute



7. A Package diagram



8. A Deployment diagram