ENSF 614 – Design Report

Group 09

Marko Mijovic (30019553)

Sarang Kumar (10081224)

Gabriel Mathias (10149399)

Nov 29, 2021

Contents

System Use Case Diagram

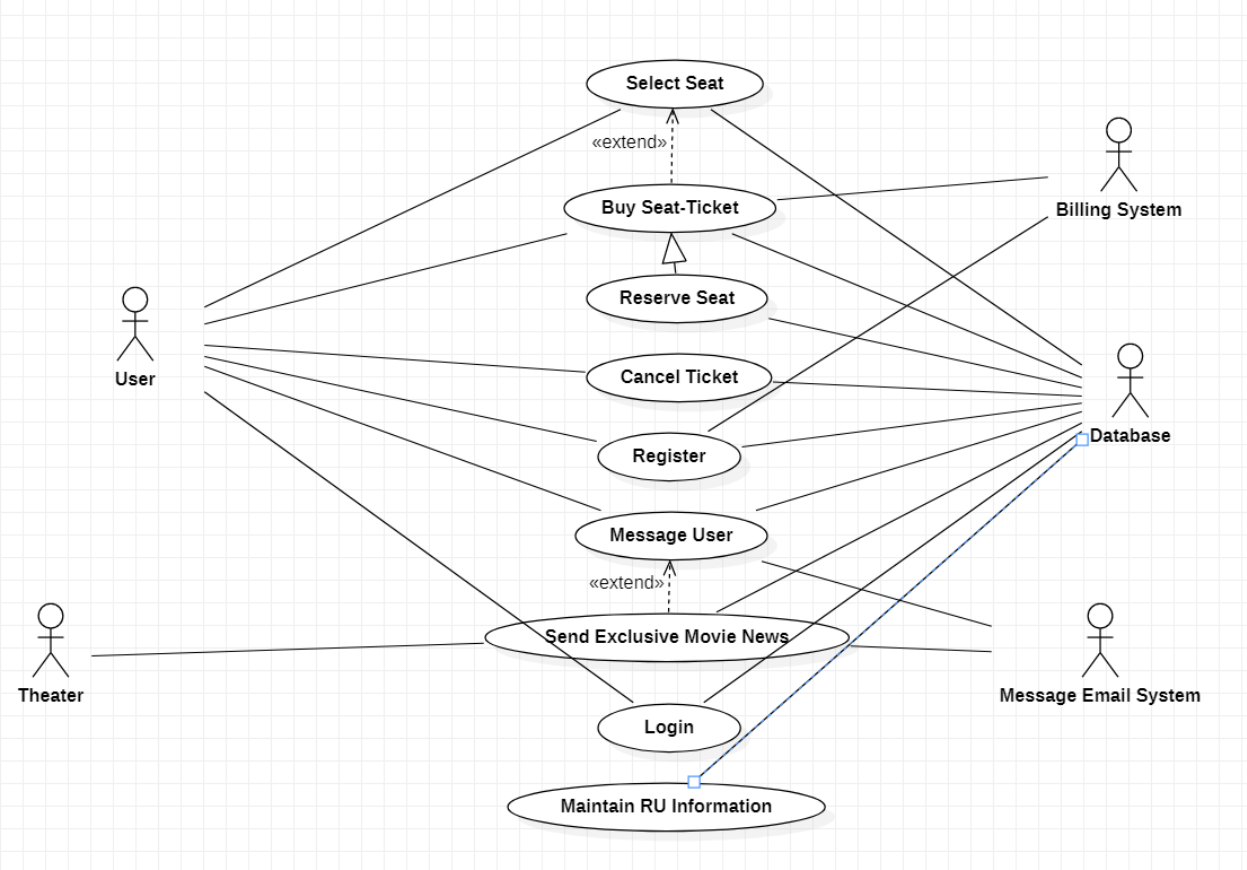

Figure 1 Use Case Diagram

System Activity Diagram

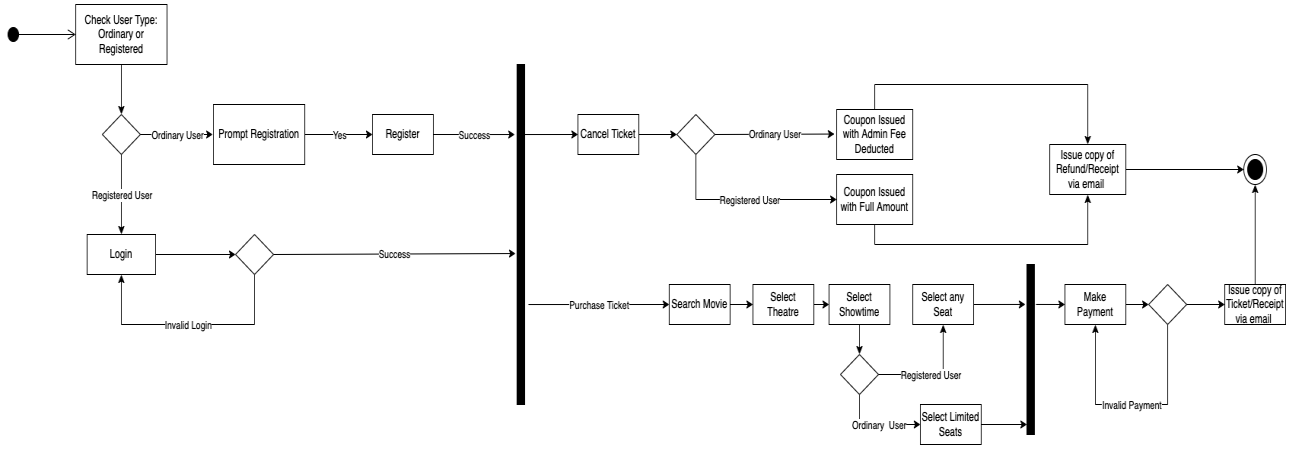

Figure 2 System Activity Diagram

Ticket State Transition Diagram

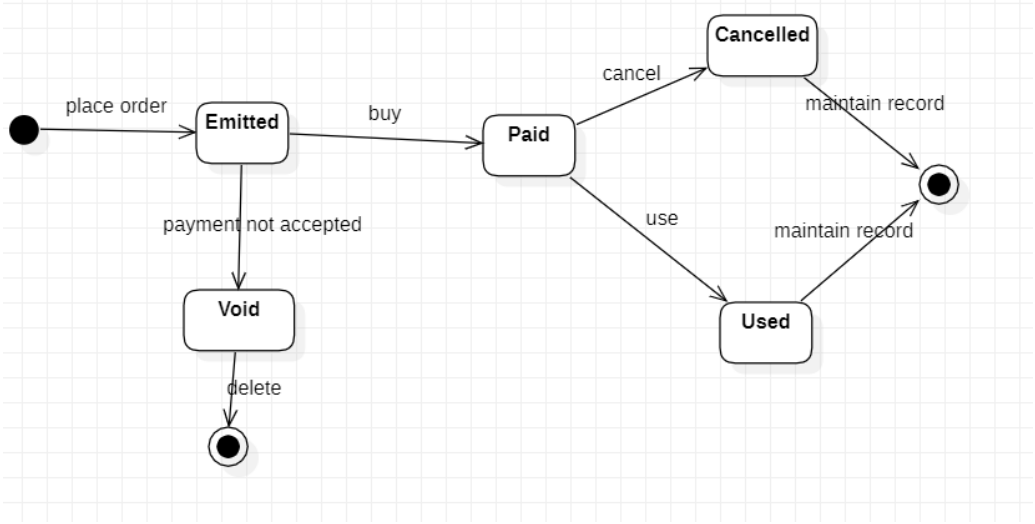

Figure 3 State Transition Diagram

Payment State Transition Diagram

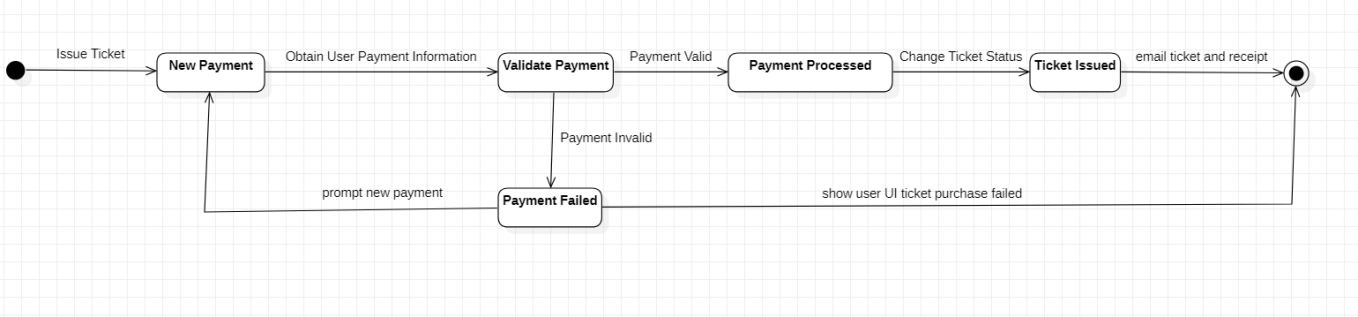

Figure 4 Payment Transition Diagram

Use Case Scenarios

Register User Scenario:

Tom wants to create an account to become a registered user in the theatre. Tom goes to the theatre application. There, at the login page, he is prompted to sign in, sign up, or continue as guest. He chooses to sign up. A screen pops up asking for his profile information: name, address, credit card, email, password. Tom fills out this information and hits the Complete Sign-Up button. The database is validates that the username is unique. Tom's profile is then created and sent to the database, where it is stored. Tom is now considered a Registered User. Tom is successfully signed in and the screen welcomes Tom using the name in his saved profile. A welcome email is sent from the messaging system to the new user.

Login User Scenario:

Tom goes to the theatre application. There, at the login page, he is prompted to sign in, sign up, or continue as guest. He chooses to sign in. A screen pops up asking for his username and password. Tom enters his login information and hits the Complete Sign-In button. The username queried but is not found in the database. The screen messages an error to Tom saying that the username or password is invalid. Tom notices a typo in the username, corrects that and hits the Complete Sign-In button again. The username is queried in the database and the profile is found. The password is validated against the one on file. The password matches with the one on file. Tom is successfully signed in and the screen welcomes Tom using the name in his saved profile.

Maintain RU Information:

Database system automatically stores information of registered users upon the completion of the *Register User Scenario* and maintain this information for future usage such as ticket purchases.

Select Seat Scenario:

Phil, a casual movie goer that does not have an account with our theater, goes to our website/UI and starts browsing a panel of the ongoing movies. He finds a movie he wants to watch. He clicks on the movie and is taken to a buy ticket UI. He is first prompted to select a date from the available movie showtime dates. Once database confirms the date, he is shown the current seat map for that showing. Seat map shows reserved, taken, and available seats. He then selects the seats he wants to purchase a ticket for and is taken to a checkout UI. From here, the buy ticket scenario begins.

Buy Ticket Scenario:

Once the user has selected a seat and is taken to the checkout page, the buy ticket for that seat scenario begins. On the checkout page user enters his payment information, email, and name. The payment is handled externally by the respective financial institution through APIs. Upon successful purchase, user is

shown his receipt, which is also emailed to him. In addition to the receipt, we also email user the digital tickets. User has successfully obtained the tickets for the movie.

Reserve Seat Scenario:

This use case begins when the user has already selected a showtime for a chosen movie and theater combination. At this point, the system will display a graphical representation of the seating map of the theater room, showing the seats that have already been sold in one colour and those that are still available in another. For the films that have not yet had a public announcement, registered users will be able to purchase up to 10% of seats on a first come, first serve basis. After this 10% of seats has been reserved, registered users will need to reserve their seats after announcement. For all users, if there are no more or not enough seats available, the system will inform a user that the show is sold out and encourage them to consider another film. Otherwise, the user will have the opportunity to reserve one or more specific seats or to simply press cancel to exist seat selection entirely. Any available seat selected by the user will be highlighted and if a user selects a seat that has already been reserved and confirmed, the system will display a message to the user and ask them to pick another seat. After the correct seats have been highlighted, the user will confirm their selection with the system. After the user confirms that these seats are the ones they want to reserve, the system will immediately ensure that said seats may no longer be accessed by future users. The user will then be informed that they have successfully selected their preferred seat(s) and can move forward with the process. The database engine will craft an update with this new information and both the available seat count and images displayed in the app will change accordingly. After this point, the use case 'Purchase Tickets' will become available to the user.

Cancel Ticket Scenario:

A user has decided to cancel a ticket that they've purchased. To do this, user goes to their email and finds the receipt email sent to them upon successful ticket payment. In that email there is a "to cancel" section with a cancel link. The user clicks the link which takes them to the cancel ticket page. On this page, user can either login if they are a registered user or proceed as an ordinary user.

If the user logs in, they are taken to their profile page showing ticket purchases. If the showing time is more than 72 hours away, they can cancel the purchase from that page for no fee.

If the user does not have a registered account, they click on the "proceed as ordinary user" button. This shows a prompt that specifies the ticket they are canceling (if the showing time is 72 hours away) and informs the user that canceling will yield them a voucher worth 85% of the original ticket cost that can be applied for future ticket purchases within a maximum of 1 year.

Message User Scenario:

Message user system automatically sends an email to a user when they purchase a ticket, cancel a ticket, register to become a registered user, or are charged annual $20 fee as a registered user.

In this scenario, an ordinary user Emily has decided to become a registered user and purchase a movie ticket. She first registers as specified in the *Register User Scenario.* Upon successful registration, she is charged the annual $20 fee. The message system will first email Emily with a congratulations email on

becoming a registered user, and then the message system will email Emily the receipt of the annual $20 fee charge.

Emily then selects a seat and buys the *ticket* for that seat as mentioned in the *Select Seat/Buy Ticket Scenarios*. As a result, the automated message system sends the receipt of the purchase, and another email with the digital tickets for the selected showing.

Send Exclusive Movie News Scenario:

Theater management can send exclusive information about a movie to the list of its registered users. In this scenario, a new movie *Die Hard III* is coming out next year. Theater management decided to send an automated email to its registered users with exclusive information about the movie's cast and show times, as well as the possibility to reserve a ticket for the opening night prior to the public release of ticket purchases. Theater management fills an email template with the exclusive information they want to send and triggers the message system to automatically send that email to all of the theater's registered users.

System Interaction Diagram

Login Sequence

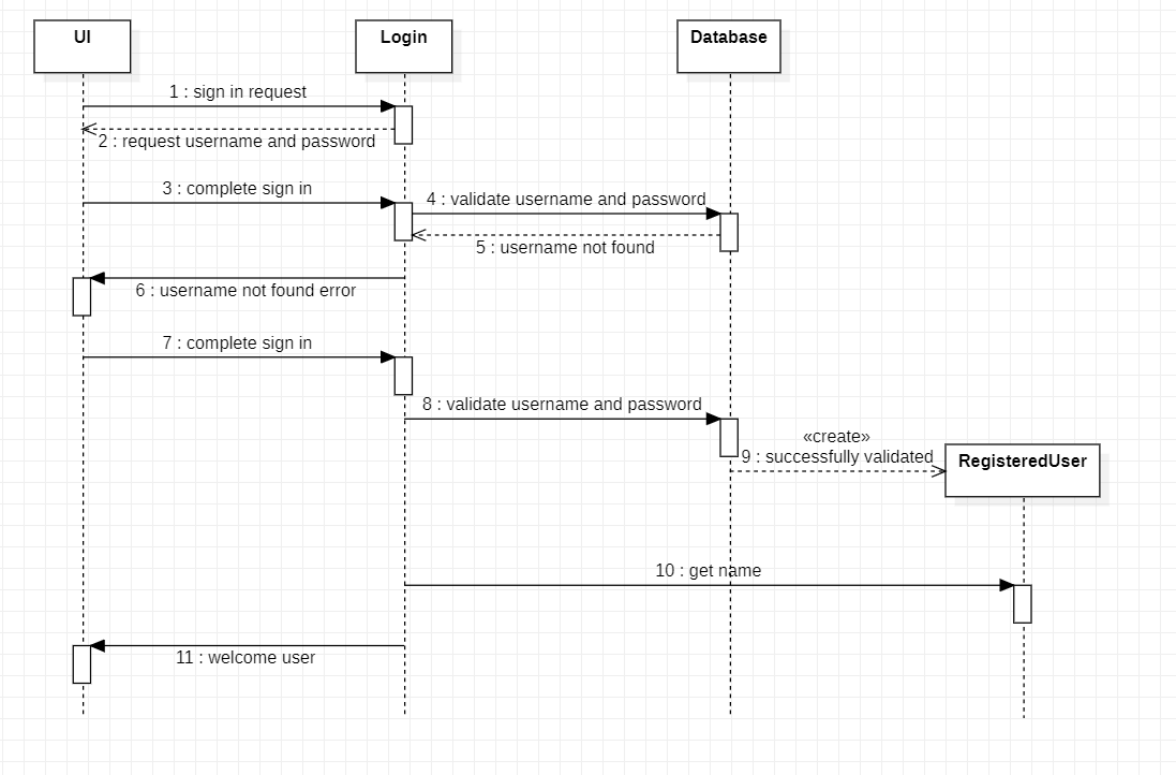

Figure 5 Login Sequence

Register Sequence

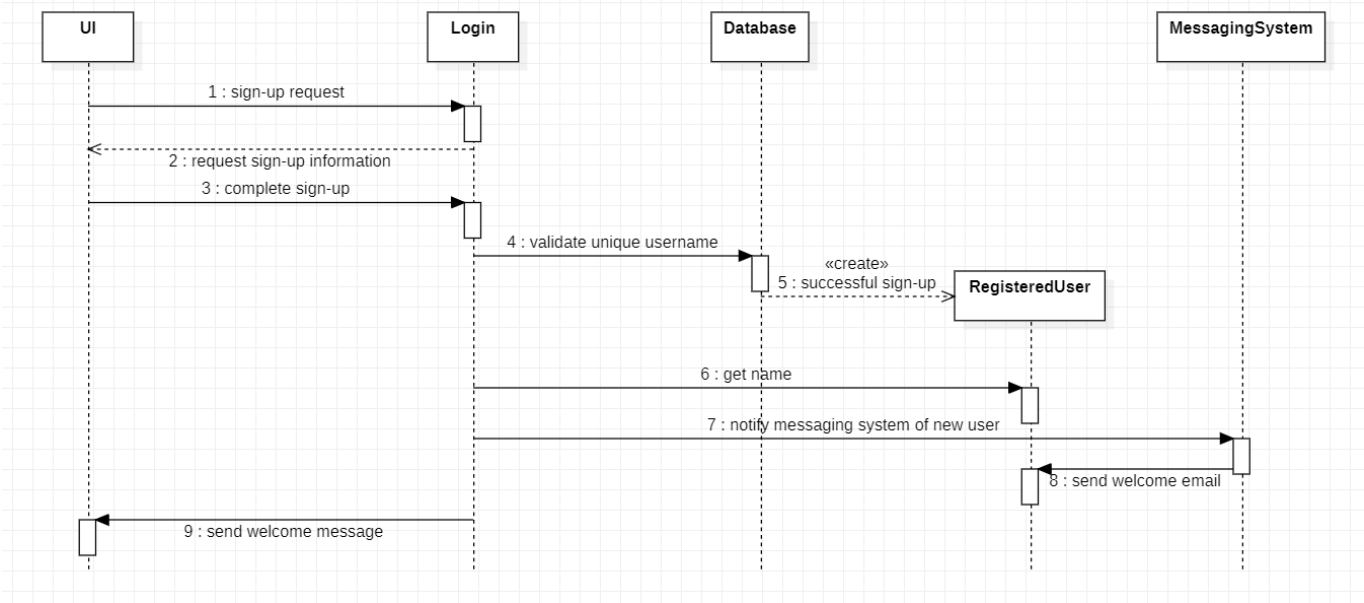

Figure 6 Register Sequence

Buy Ticket Sequence

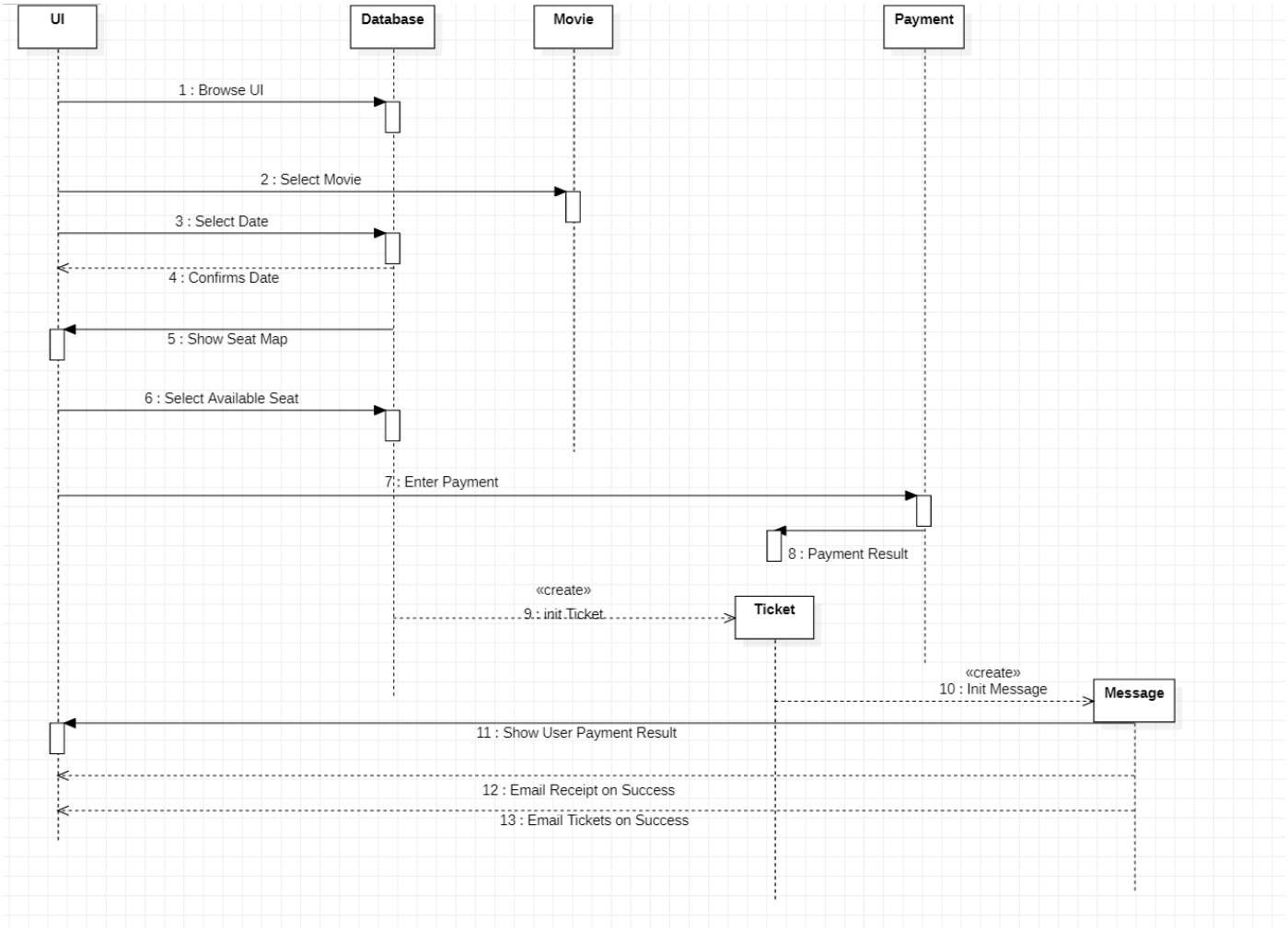

Figure 7 Buy Ticket Sequence Diagram

Cancel Ticket Sequence.

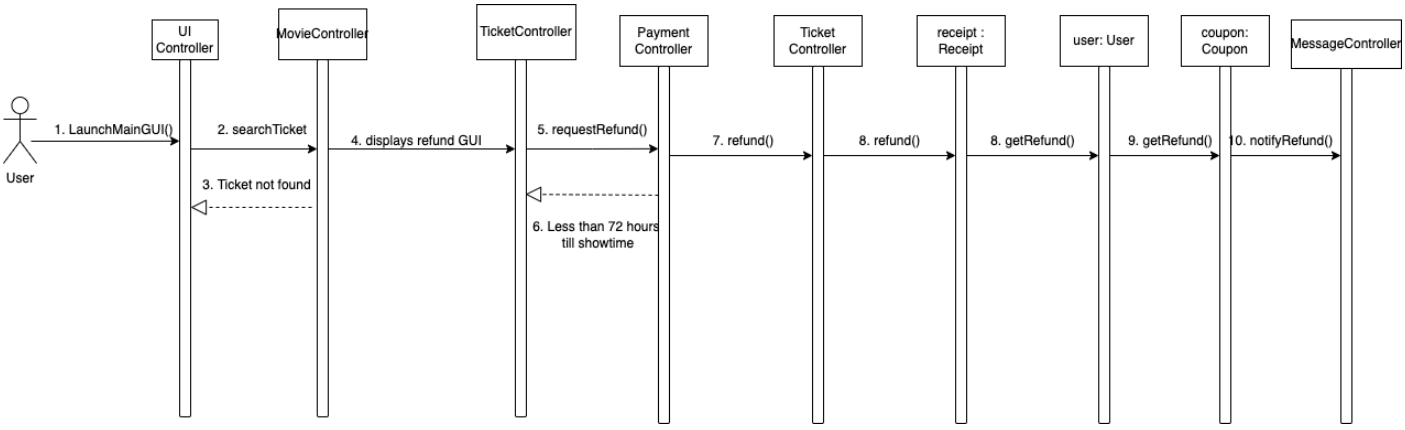

Figure 8 Cancel Ticket Sequence

Class Diagrams

Relation-Only Class Diagram

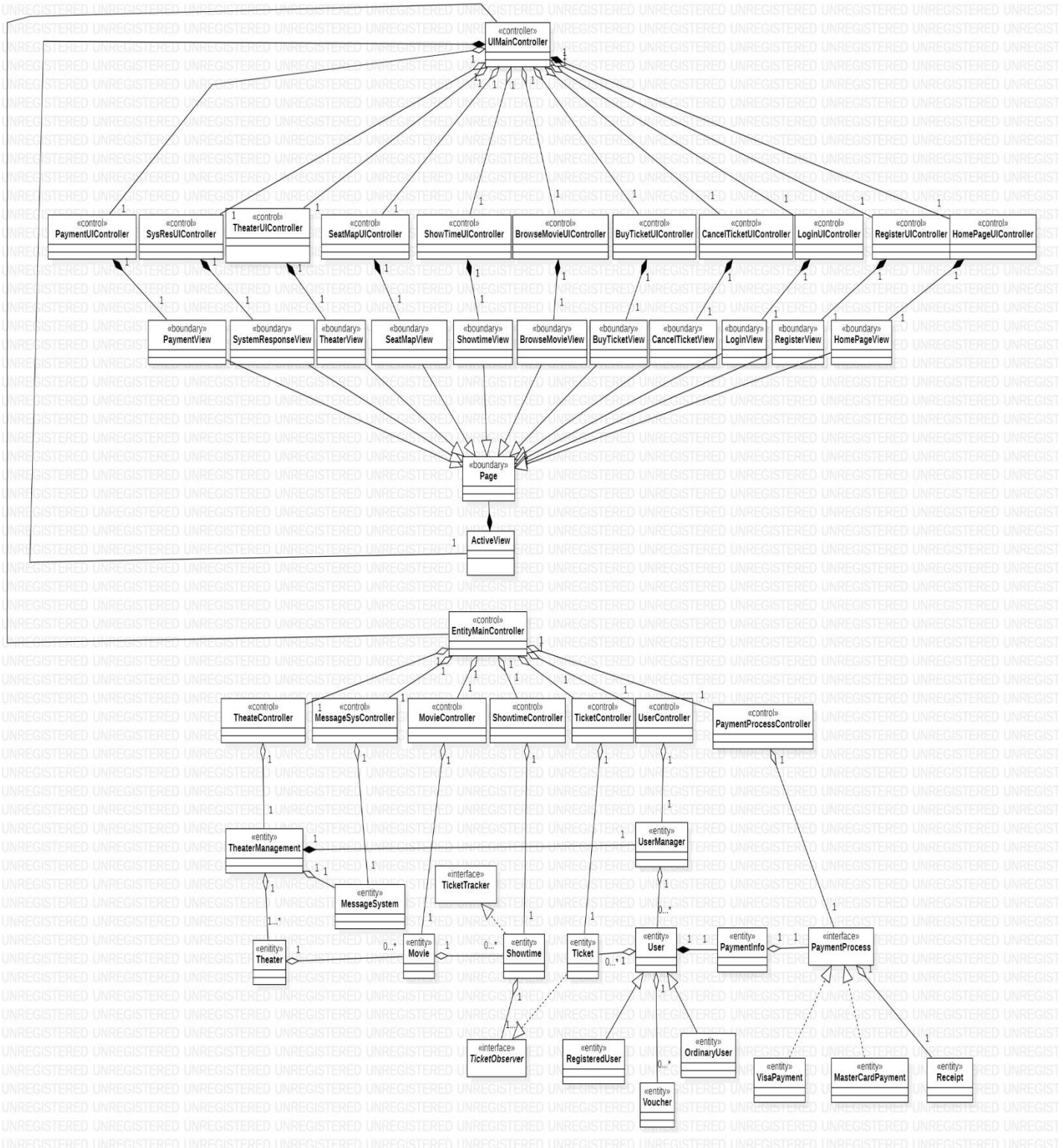

Figure 9 Relational Class Diagram

Attribute-and-Methods Class Diagram

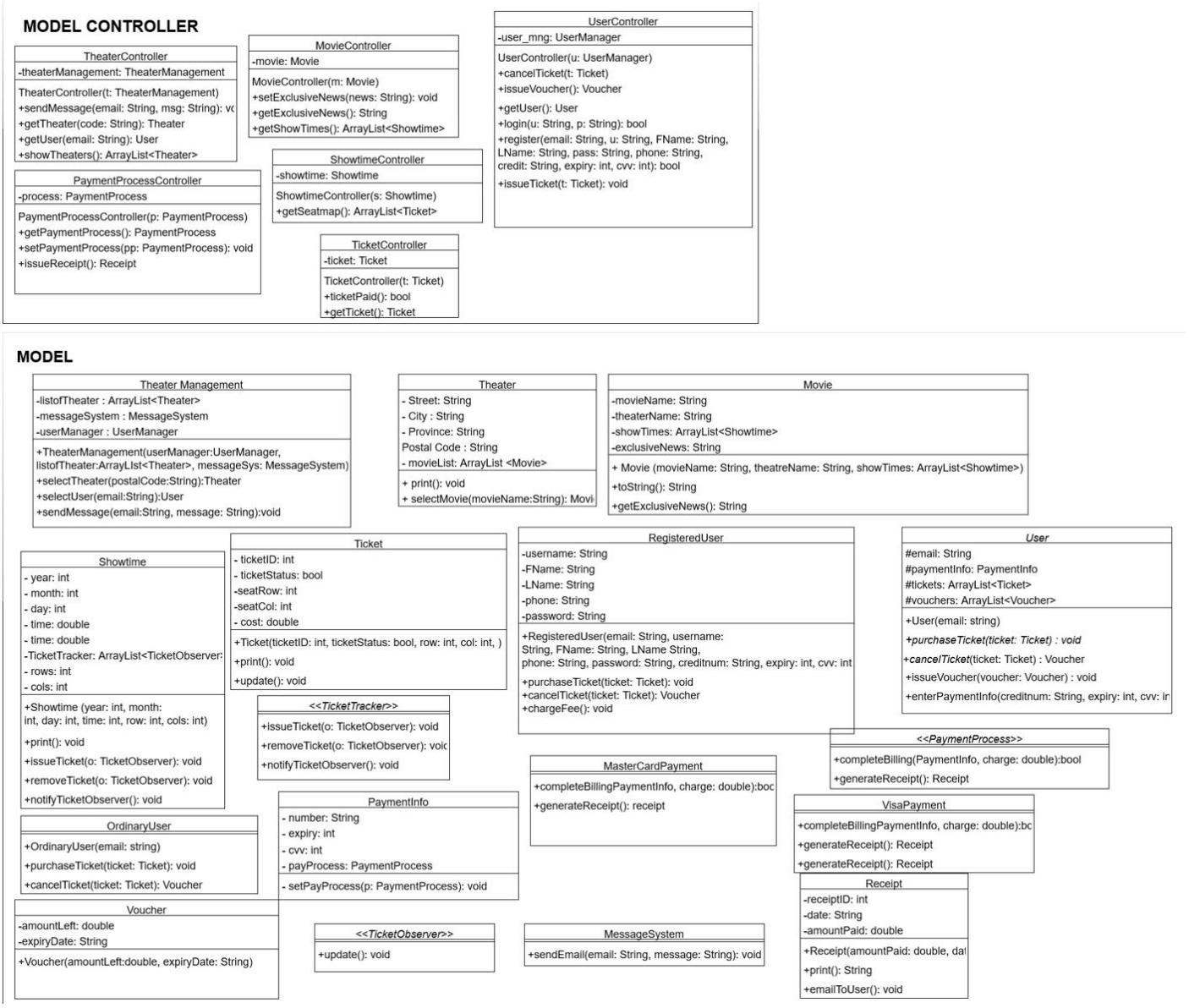

Figure 10 Attribute and Methods P1

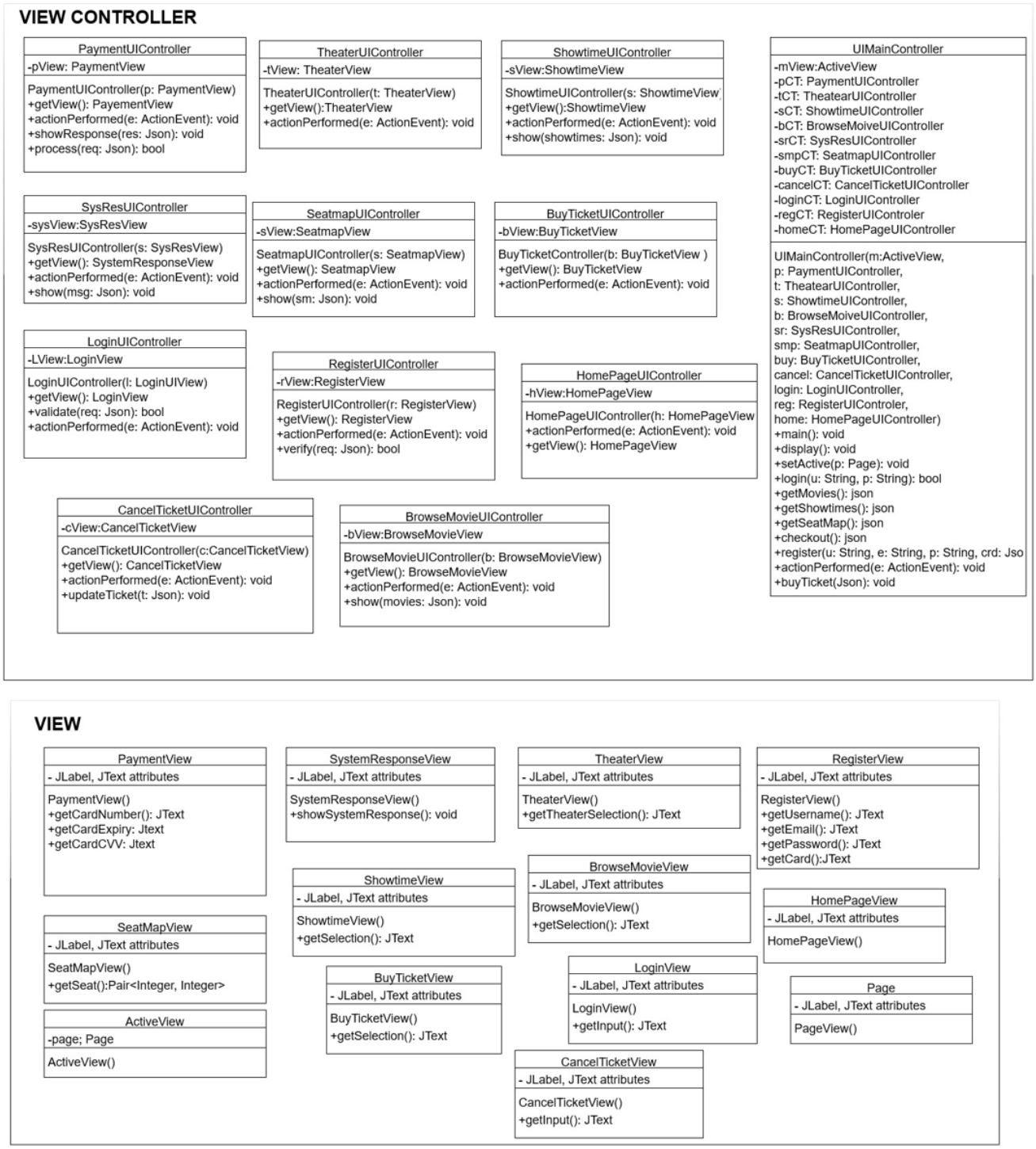

Figure 11 Attribute and Methods Part2

Package Diagram

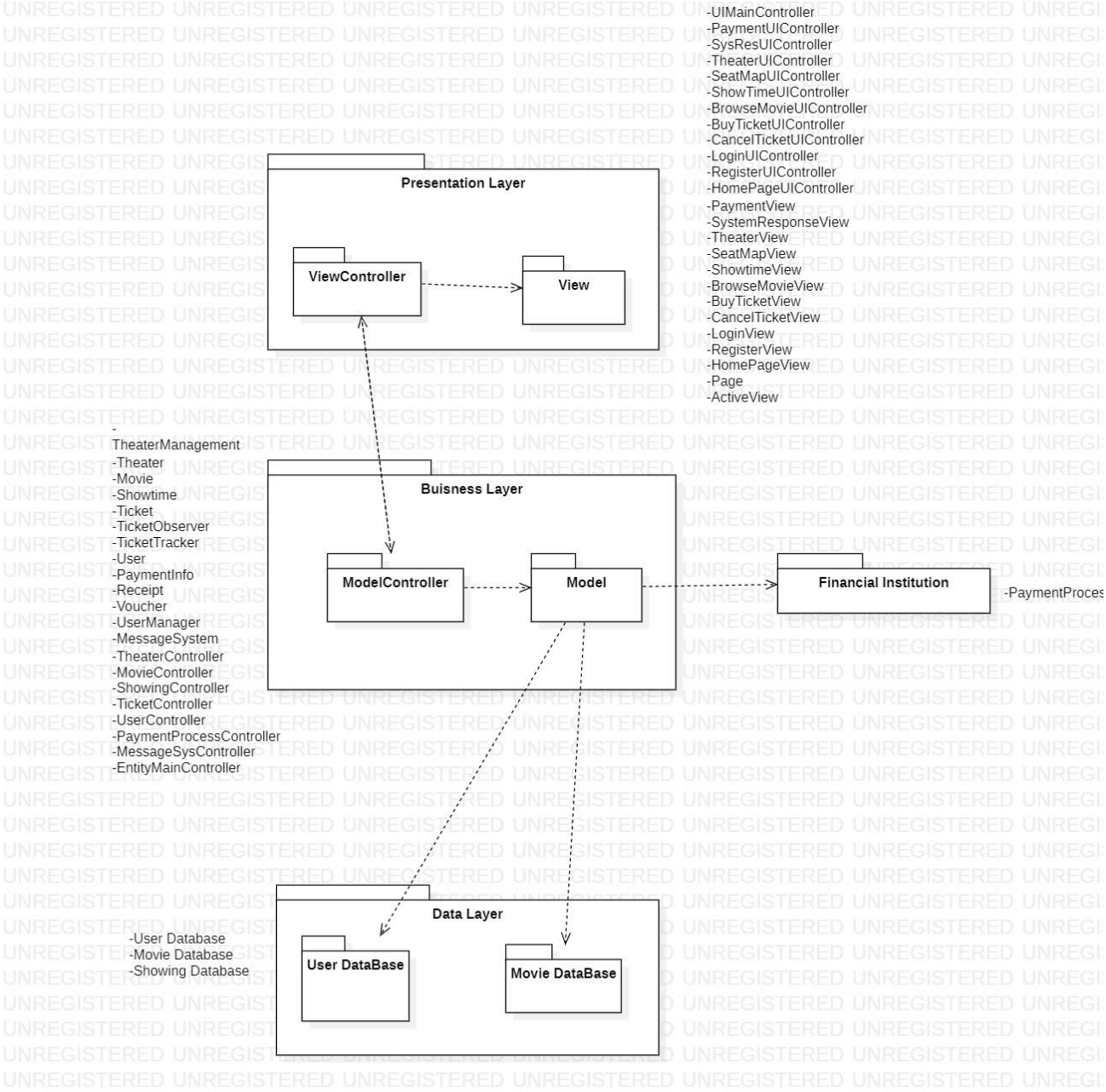

Figure 12 Package Diagram

Deployment Diagram

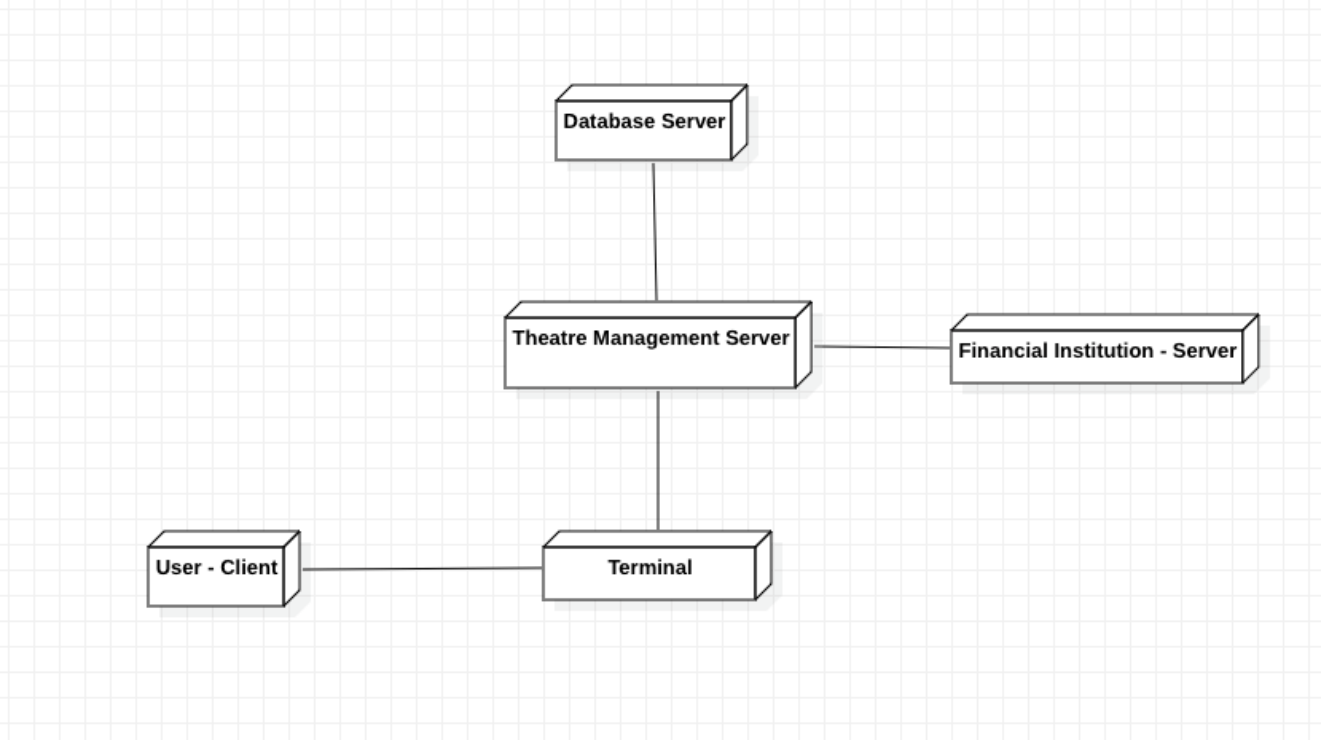

Figure 13 Deployment Diagram