ENSF 619 Project

**Group 9**

Aidan Lee 30000074

Andrew Lee 30128102

Justin Leong 10127156

Patrick Linang 10122254

**Note:** All diagrams have been included in an attached folder if they are too difficult to read in this document

**Actors**

**Ordinary User**

A user that does not need to log into the system. They are able to search for movies, select theatres, make payments with a credit card, and receive an emailed ticket and receipt. Ordinary users are able to cancel their ticket up to 72 hours before showtime, and will receive a voucher for 75% of the ticket’s value, which is valid for the next year.

**Registered User**

A user that must log into the system, and must pay a $20 annual fee. They are able to perform the same activities as an ordinary user, but have different ticket cancellation behaviour. Registered users may cancel their tickets any time prior to showtime and receive a voucher for 100% of the ticket’s value. They also receive movie news prior to public announcements, during which time they may purchase up to 10% of the seats.

**Database**

A database that stores a list of movies, theatres, showtimes, seats, and tickets. Additionally, it stores the name and credit card information of registered users.

**Financial Institution**

A financial institution that makes payments to a payee on the request of an ordinary or registered user.

**Email Server**

An email server that sends emails to an ordinary or registered user. Such emails may contain tickets and receipts for ordinary or registered users, or movie news for registered users.

**Systems Use Case Diagram**



**Use Case Scenarios**

**Note:** Underlined words represent entities. Bolded, Italicized, Underlined words represent operations.

E.g. This is an entity; this is an ***operation***.

**Register a User**  
This use case begins when a User wishes to become a Registered User. The system ***prompts*** the User for personal information such as their name, address, credit and/or debit card account, and email. The system also ***prompts*** for new account login details including username and password. The system should ***check*** that user inputs are valid (e.g. does user input satisfy the criteria for username and password strings?). Additionally, the system should ***check*** that the inputted username doesn’t already exist in the database. If all input fields are valid, a new user account is ***created***. The system should ***store*** the account information in the database.

**Login/Logout**

This use case begins when a Registered User wishes to ***log into*** the ticket reservation app. The system ***prompts*** them to enter their username and password. After both have been entered, the system ***checks*** the validity of the login attempt. If both username and password are correct, the Registered User is ***granted access*** to the app. Otherwise, the system ***prompts*** them to try again. After successful login, if the Registered User wishes to logout they will ***click*** the logout button on the application and the system will ***remove*** any additional features that a Registered User has access to. The system will then ***return*** the user to the general user interface screen of the application.

**View and Select Theatres**

This use case begins when a user ***enters*** a search term and ***clicks*** the *Search* button. The system ***searches*** the database for theatre names matching the search term. If there are matching results, the system ***displays*** them in the UI for the user. If there are no results, the system ***displays*** a “no results” message to the user. The user ***clicks*** on one of the theatres from the list of search results to select it, then the system ***displays*** information about the selected theatre, such as name, and location. The list of available movies is also displayed.

**Select a Movie**

This use case begins when a user has selected a theatre, and the list of available movies is displayed. The user ***clicks*** on one of the available movies to select it. The system ***displays*** information about the selected movie, such as title and release date. The list of available showtimes is also displayed. The list of available showtimes displayed to Registered User will additionally include movies that have not yet been publicly announced if less than 10% of the seats have already been purchased.

**View and Select Showtimes**

This use case begins after a user has selected a movie and a theatre. The system will then ***display*** a list of showtimes. A showtime will have a movie name, the specific theatre, and the movie start time. The process is finished when the user either ***clicks*** the desired showtime icon to ***book*** a seat, or ***clicks*** a button to cancel and ***exit*** the process.

**View and Select Seat**

This use case begins after a user has booked a showtime. The system will then graphically ***display*** a birds-eye-view of the theatre with seat icons indicating the location of the available seats in the theatre. The user can ***enter*** a row character (A-Z) and a seat number (1-25) for a seat in the booking. Once a seat location is selected, the user can ***click*** a button to submit the selection to the system. If the submission is successful, the system will confirm the completion of the process and ***initiate*** the ticket payment process. If the submission is invalid, the system ***refreshes*** the seating chart and ***prompts*** the user to try again.

**Make Ticket Payment**  
A ticket payment begins after a User ***selects*** a movie, theater, showtime, and seat. If the selections for the ticket reservation are valid, the User ***provides*** a valid method of pay. Payment method is either by credit card or by voucher. If the User ***pays*** by credit card, they must provide credit card information which can include name, billing address, and credit card number. If a User is a Registered User, their information could be ***retrieved*** from the database. The credit card information and amount to charge will be ***sent*** to the financial institution tied to the credit card. Alternatively, if the User wants to ***pay by voucher***, the User must provide a voucher ID number. A ticket reservation is finalized once the payment information is successfully ***sent*** to the financial institution or the voucher is confirmed to be valid. The transaction should also be ***recorded*** somewhere (e.g. a database) and ***ensure*** that no future payments can be made for the same ticket reservation (unless the ticket gets cancelled).

**Make Account Payment**  
A Registered User must ***pay*** a $20.00 annual account fee. An account payment begins when the Registered User provides their account information. If the account information ***matches*** an existing account, it should ***check*** that no account payment was previously made for the same year. If valid, the annual fee payment information will be ***sent*** to the financial institution tied to the Registered User’s card. The annual account fee payment process finishes once the payment is successfully ***sent*** and the Registered User’s account annual fee is ***marked complete*** for the current year.

**Receive Ticket and Receipt**  
This use case begins after a successful ticket payment. The movie ticket ***includes*** the movie name, theater, showtime, and seat information. It could also ***include*** a unique ticket ID number. The User also ***receives*** a receipt which can include the date and time of purchase, the ticket(s) paid for, and the payment information. This information is sent to the email address ***provided*** by the User. For a Registered User, their email address could be ***retrieved*** from the system’s database. The process is finished once an email attached with the movie ticket and receipt is successfully ***sent*** to the email server.

**Cancel Ticket** **and Receive Voucher**

This use case begins when the User has already purchased a ticket for a movie and has decided that they would like to ***cancel*** this purchase. At this point the system will ***ask*** the User for the ticket information for which they would like to cancel. The ticket information for cancellation would need to ***include*** the ticket number, ticket holder name, and email to ***send*** a credit voucher to. Depending on the uniqueness of the ticket, additional information such as movie name, date, and seat number may be ***provided*** to identify which ticket to be cancelled in the system. Once the cancellation has been ***submitted***, the system will ***determine*** whether the cancellation is valid (72 hours in advance). At this point the system will determine whether the ticket belonged to an Ordinary User or Registered User to determine the amount of the credit voucher to ***send*** back to the customer. For regular Users, a 15% administration fee will be ***deducted*** from the ticket price to ***compute*** the credit amount. For Registered Users, the full amount of the ticket price will be ***converted*** to a voucher. At this point, the system will ***generate*** a voucher containing the information of the credit amount, the date the voucher was generated, expiration date, and unique voucher number. This voucher will then be ***sent*** to the User through the email that they provided during cancellation.

**Receive Movie News**

This use case begins when movie news is to be ***generated*** and ***released*** to the public. At this point the system ***creates*** the movie news which contains information about the movies offered, release date of movies, show times, and prices. The system then ***determines*** the date in which the movie news will be ***released*** to the public as well as the early release date for Registered Users. On the early release date, a list of Registered Users will be ***collected*** by the system and a copy of the movie news will be ***sent*** to them through their email address.

## 

## **Candidate Objects**

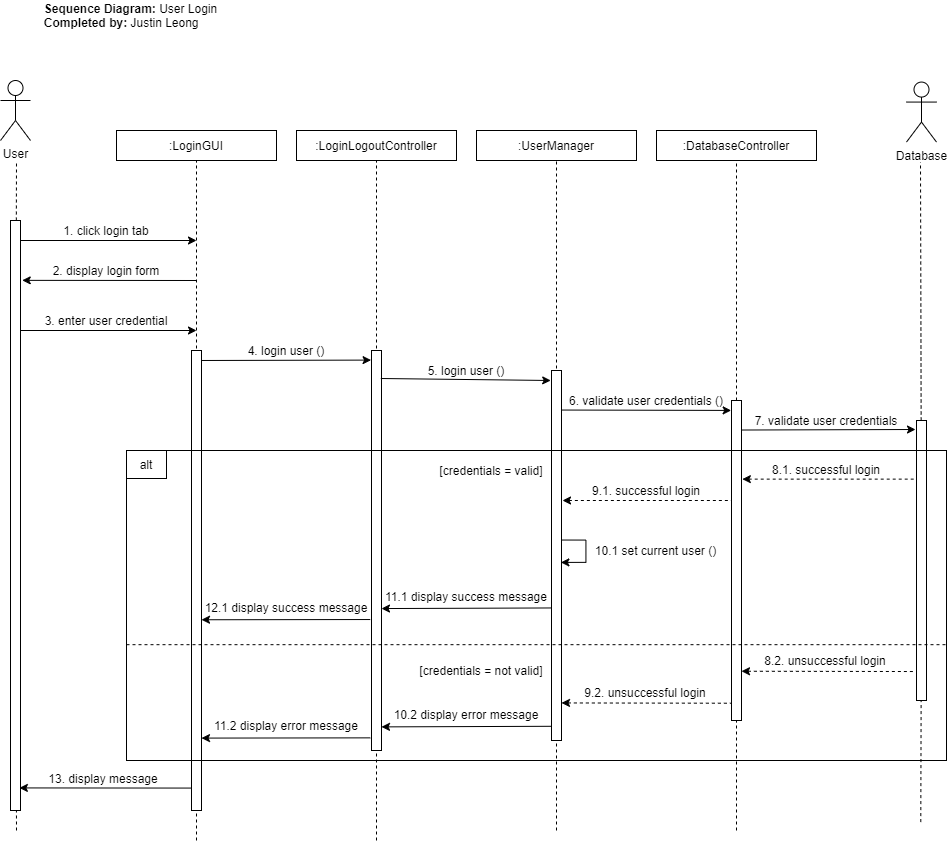
* User
  + Ordinary User
  + Registered User
* Theater
* Theater Room
* Seat
* Ticket
* Receipt
* Email
* Movie
* Movie List
* Voucher
* Showtime
* Movie News
* Card
  + Credit Card
  + Debit Card

## 

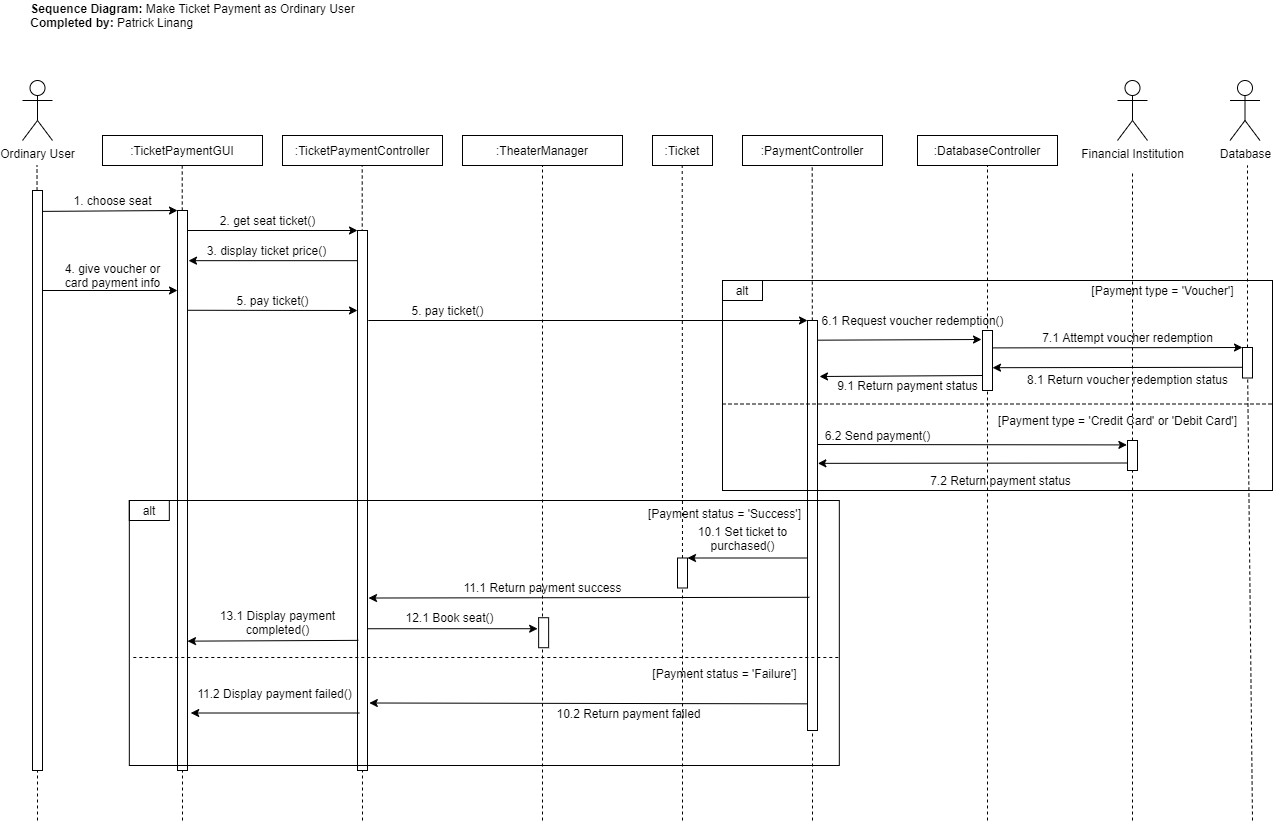
## **Potential Operations**

* Prompts
* Check
* Created
* Store
* Log into
* Granted access
* Log out
* Select
* Remove
* Return
* Enters
* Clicks
* Searches
* Displays
* Initiate
* Exit
* Refresh
* Provides
* Pays
* Retrieve
* Sent
* Pay-by-voucher
* Record
* Ensure
* Matches
* Marked complete
* Includes
* Cancel
* Submitted
* Deducted
* Compute
* Converted
* Generate
* Released
* Creates
* Collected

# **System Interaction Diagram - User Login (Justin)**



# **System Interaction Diagram - Make Ticket Payment (Patrick)**

****

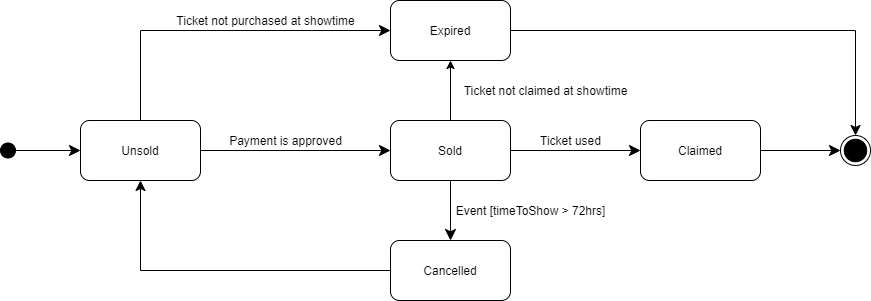
# **System Interaction Diagram - Cancel Ticket (Aidan)**

# 

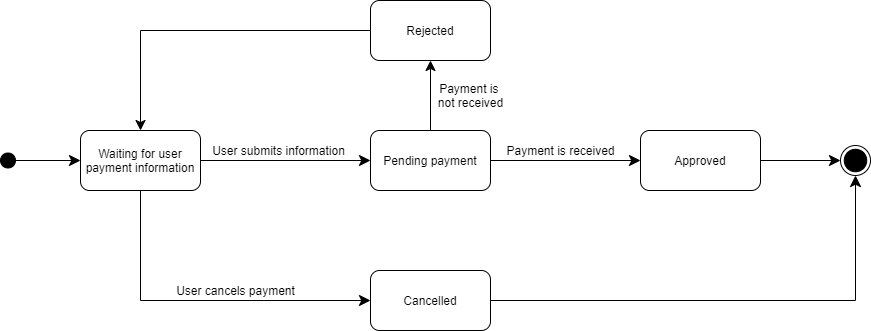
# **System Interaction Diagram - View Seats (Andrew)**

# 

**State Transition Diagram - Ticket Object**

****

**State Transition Diagram - Payment Object**

****

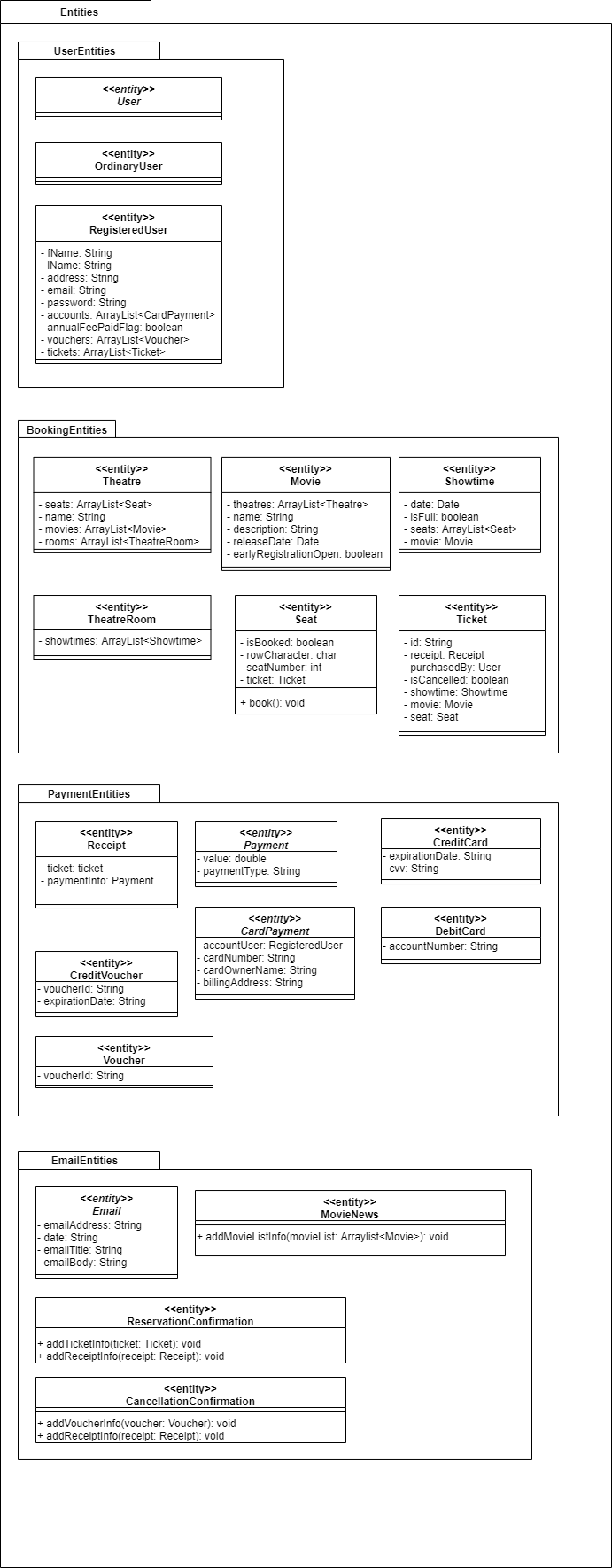
# 

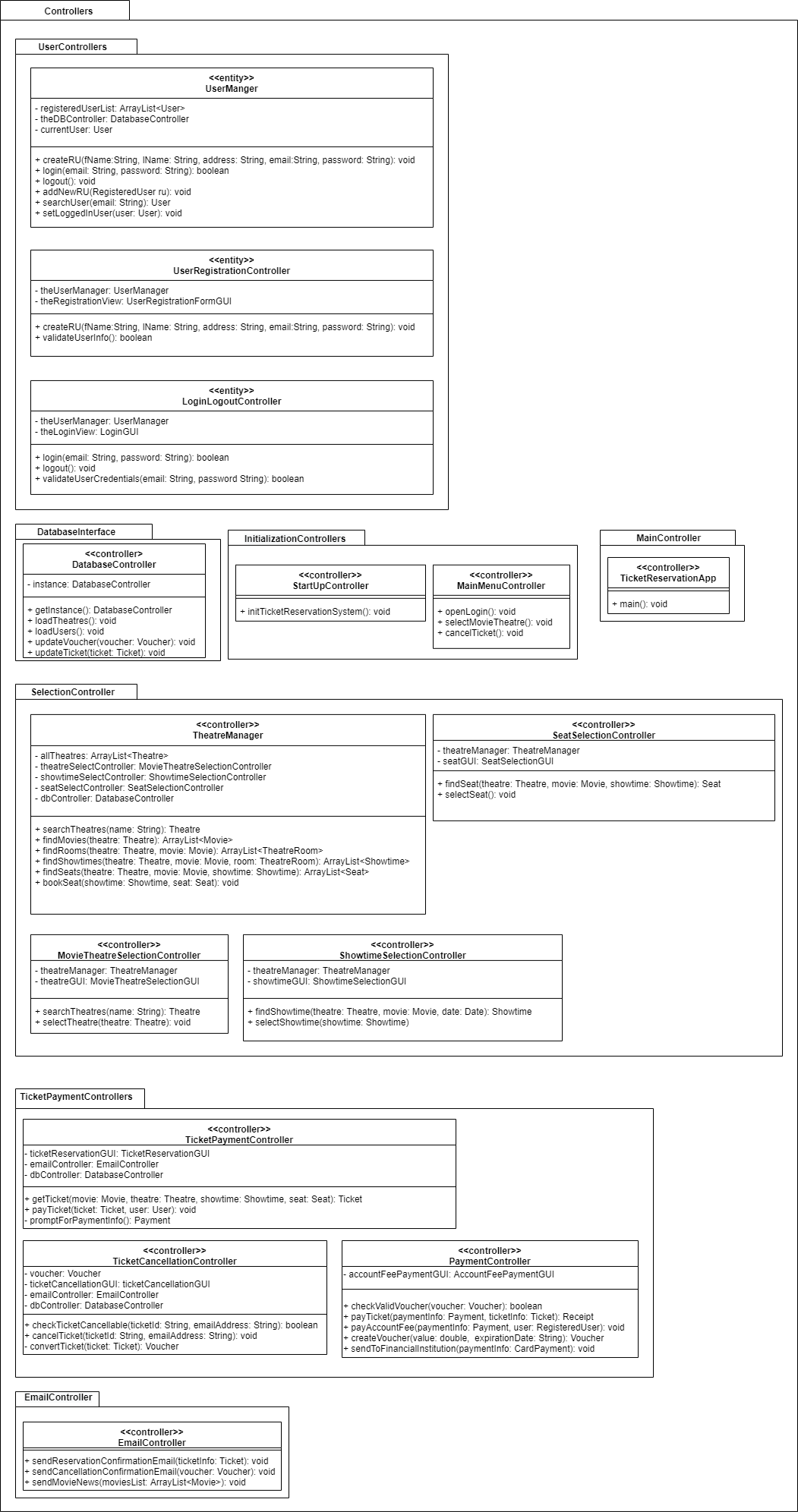
# **Design Level Class Specification Note:** All diagrams have been included in an attached folder if they are too difficult to read in this document

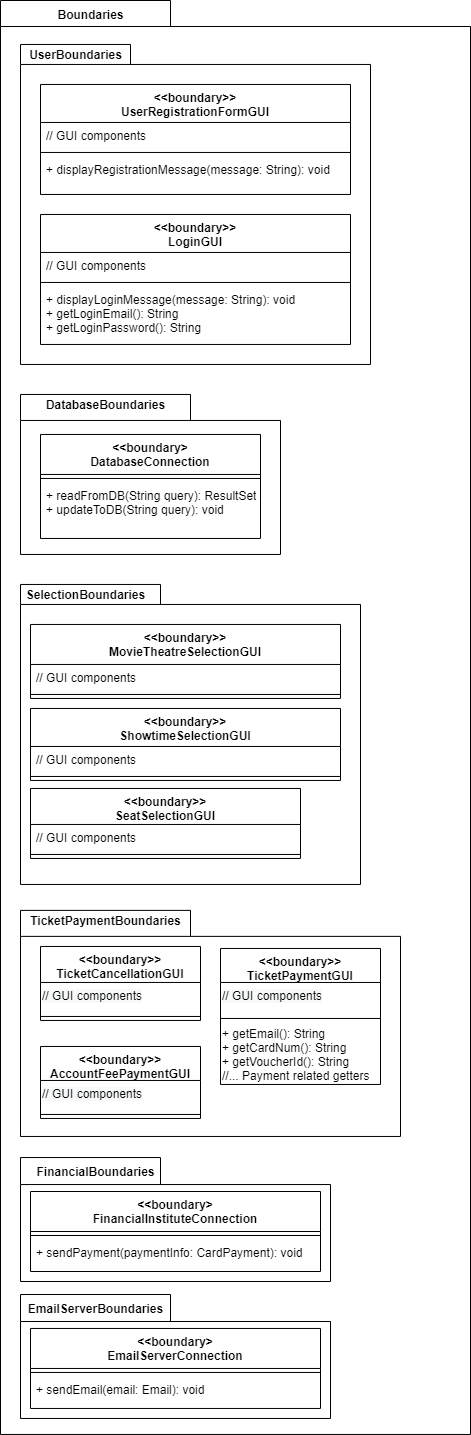
# 

# 

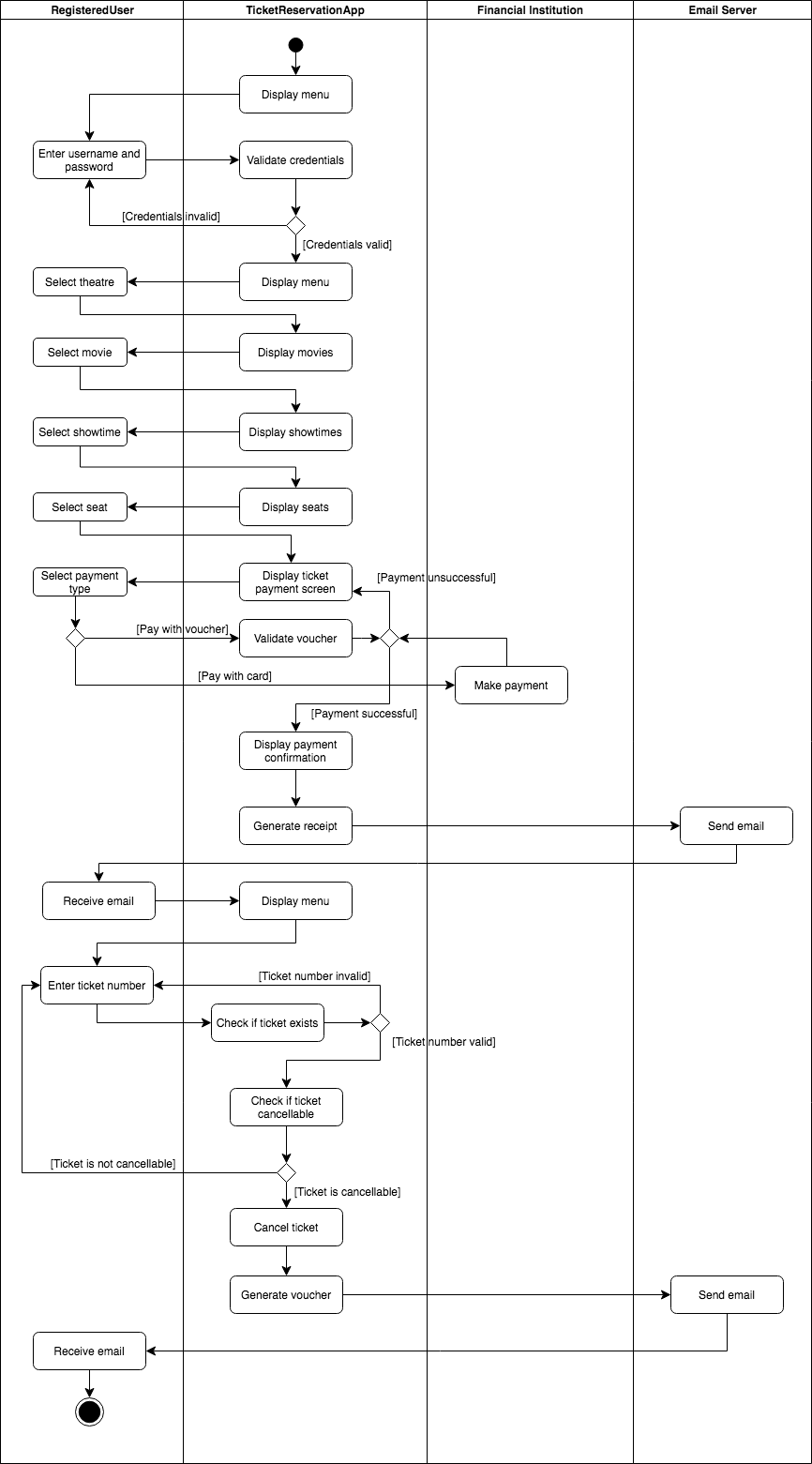
# **Detailed Design Level Class Specification Note:** All diagrams have been included in an attached folder if they are too difficult to read in this document

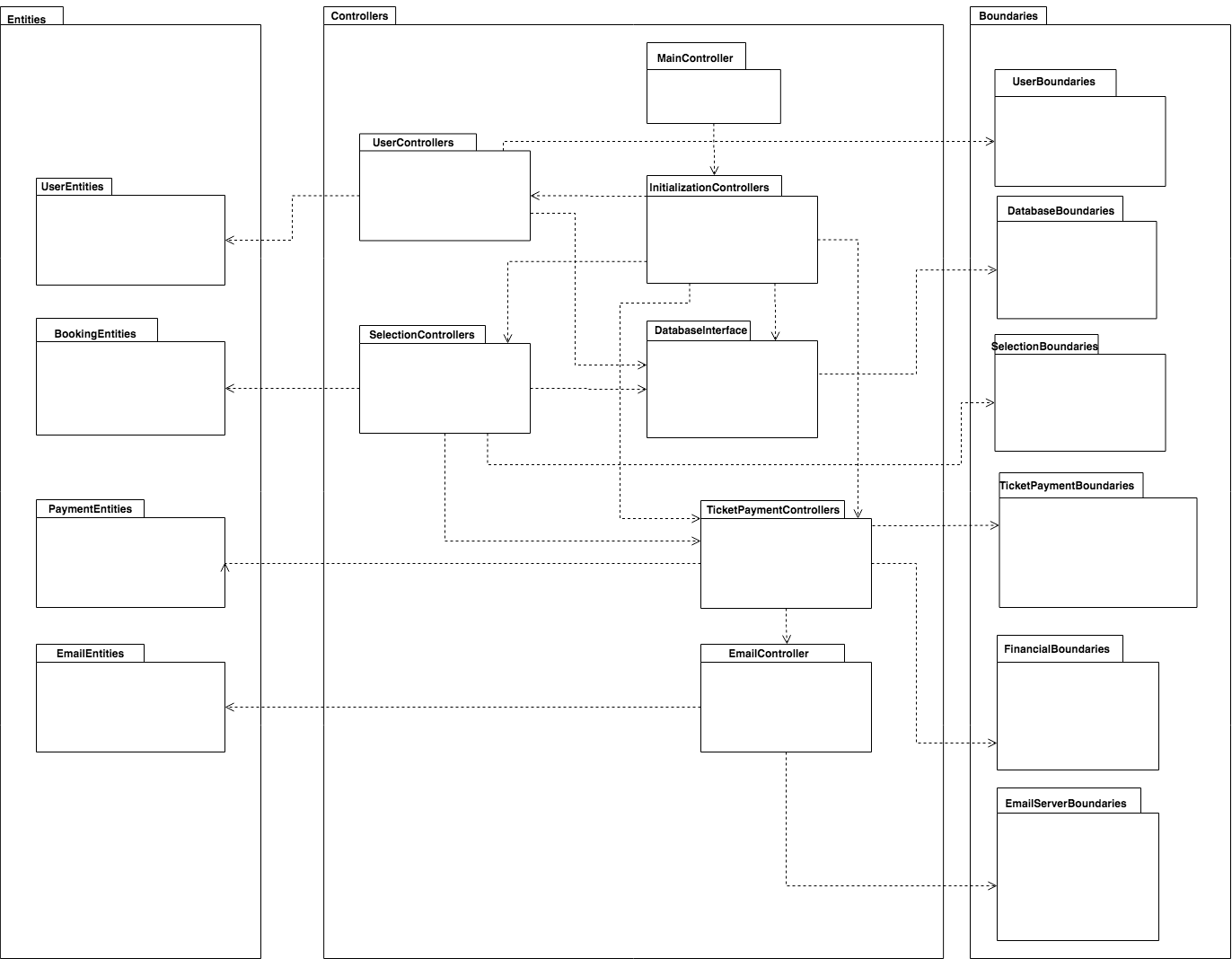






**Systems Activity Diagram**

**Package Diagram**

****

**Deployment Diagram**

# 

# 

# 