# ENSF 480 TERM PROJECT

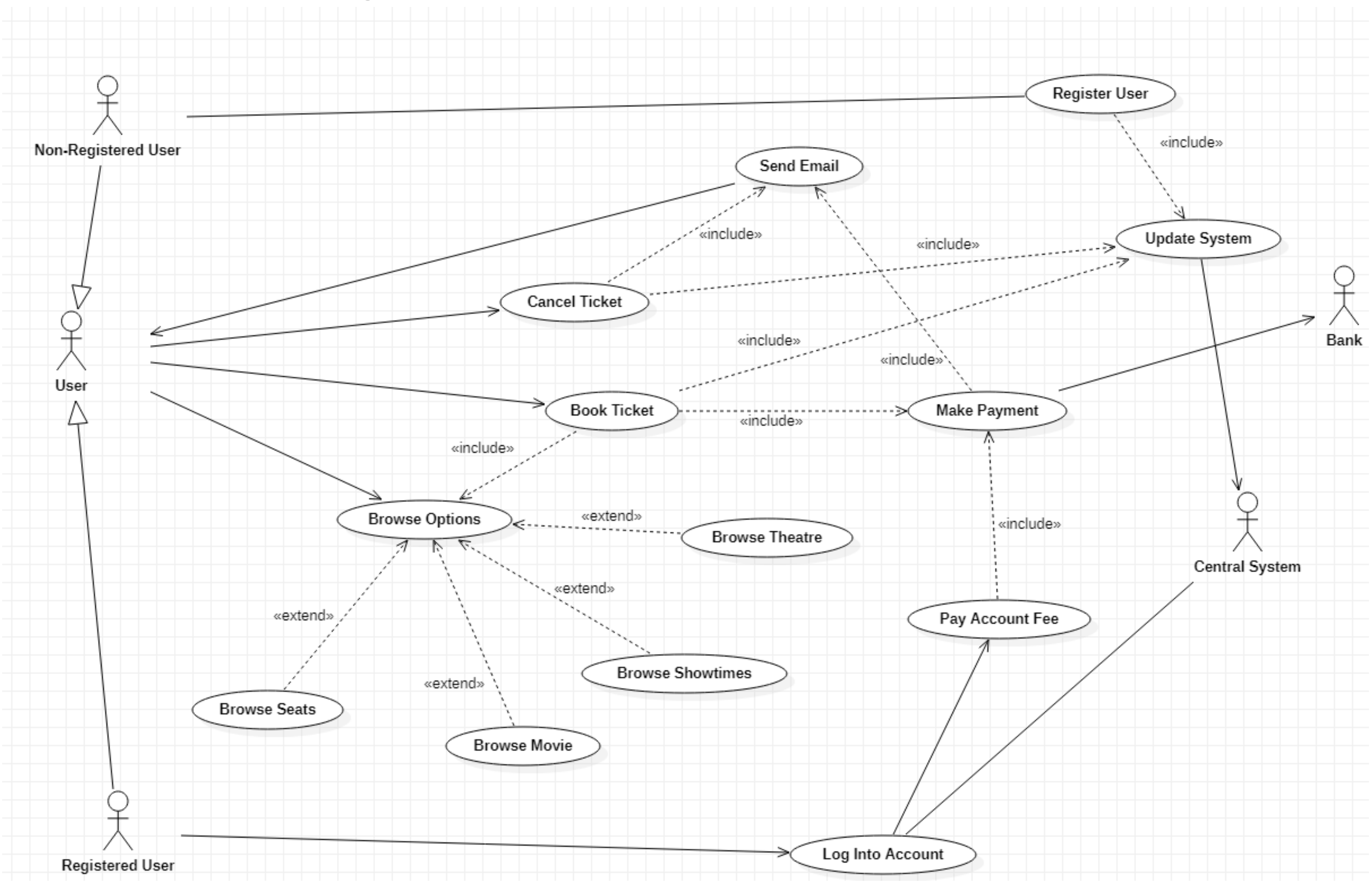
**NAME:** Minh Vo
**UCID:** 30061394

**NAME:** Matthew Wells
**UCID:** 30064094

**NAME:** Michael Vassilev
**UCID:** 30068475

**NAME:** Junhao Xue
**UCID:** 30064444

## Use case diagram:



## Actors:

**Registered User:** These users have created an account with the cinema website and will receive special member-only benefits such as zero administration fee when cancelling tickets, but have to pay an annual fee of $20.00. They have a 72 hour window to cancel their ticket and receive credits for the next purchase. They will also receive a movie announcement prior to the public announcement and can purchase tickets on a first come first serve basis.

**Non-Registered User:** A Non-Registered User is a user that does not have an account setup with Theatre System. They are able to purchase tickets for specific theatres, times, and seats, which they will receive via email. The email contains a link that allows the user to cancel their ticket. Non-Registered Users are subject to a 15% deduction on the credit they receive from canceling a movie ticket. They receive this credit via an email coupon, and are only able to cancel up to 72 hours before a show.

**User:** A user is a generic customer, and can be a registered user or a non-registered user. They are able to browse through movies, book and cancel tickets, and receive emails from the system.

**Central System:** A Central System is a database that keeps track of all user information, including registered user accounts, as well as information of booked and cancelled tickets. It is also responsible for sending movie announcements to all registered users' accounts.

**Bank:** A Bank is an actor that is responsible for processing the user's payment to the cinema. The bank will have information on the user's payment methods and personal information, as well as access to the user's card. The money paid by the user will go to the cinema after going through the bank.

## Use case scenarios:

**Note: Blue highlighted text represents operations*

**Browse Options:** This use case begins when the user has started the process of booking a ticket. The system will then prompt the user with a list of options by a GUI controller, including "browse movies", "browse showtimes" and "browse seats". The user can then click on the "browse movies" option to proceed.

**Browse Theatre:** This use case begins when the user has clicked on "Browse Options". The website managed by a theatre controller should display a list of theatres for the users to choose from. When the users have chosen their preferred theatre, they can browse showtimes, movies, and available seats.

**Browse Movies:** This use case begins when the user has browsed and selected a theatre. The cinema website will display a list of movies available. At this point the system displays a message that prompts the user to either select his desired movie or go back to the previous stage. The user can then choose a movie and proceed to browse showtimes.

**Browse Showtimes:** This use case begins when the user has selected a theatre and the movie he wants to watch on the cinema website. At this point the system displays all available showtimes for this particular movie in the chosen theatre. The user can either select a specific time slot or go back to the previous stage. After they choose the showtime, they can then proceed to browse seats.

**Browse Seats:** This use case begins when users have selected their preferred showtime. The user will see a map of all available seats and the cost of each seat. The user can choose multiple seats by entering the row and column number of the seat they want. The total cost will be displayed on the side. There will be a button to proceed to book ticket(s).

**Book Ticket:** This use case begins when the user clicks Book ticket on the main page. The system will prompt the user to select their theatre, movie, showtime, and seat from "Browse Options". Once that is done, the system will use a ticket controller to generate a ticket for the selections. The system will then prompt the user for payment. The system will then update the central system with the information about the ticket, which the user receives via email.

**Make Payment:** This use case begins when the user has already selected his movie, showtime and seat, and decided to proceed to checkout, or when a registered user pays their membership fees. At this point the system asks the user to enter their payment information. Once they submit it, the system verifies and confirms the completion of the process. The user will then receive a confirmation email with a receipt.

**Register User:** This use case begins when an unregistered user chooses the option "Register" on the cinema's website. They will be given a pop-up where they are prompted to enter their name, phone number, address, email, and card number. The user must agree to terms and services to proceed. After that, they become a registered user and receive special benefits. The information of the user will be stored in the cinema's database.

**Cancel Ticket:** This use case begins when the user selects "Cancel Ticket" on the main page of the system. The system will first prompt the user to enter their ticket number that they received via email. The system will then use its ticket controller to create a search query for the ticket in the central system. Once this is done, the system will update the central database and the user will receive an email that contains a code with a credit that is equal to 85% of the value of the cancelled ticket if they are a non-registered user, and 100% of the value if they are a registered user. The user can then use this credit code the next time they want to book a movie ticket to receive a discount.

**Send Email:** This use case begins when the user makes payment after booking their ticket(s) or receives credit after cancelling their ticket(s). The email will be sent from the central system to the user. For a book ticket email, it includes the relevant information for the purchase such as ticket number, theatre, movie, showtime, seat, and total cost. For a cancel ticket email, it includes the relevant information such as ticket number, theatre, movie, showtime, seat, total cost, and credits received.

**Update System:** This use case begins whenever the user books a ticket, a new user registers or a ticket is cancelled. The system then uses its database controller to update the central system.

**Log Into Account:** This use case begins when a registered user clicks the login button of the main page of the system. The user is prompted to enter their login credentials, which are verified by the central database system. The registered user is then able to receive the benefits of being a registered user while using the system, such as no 15% deduction from refunds, and the ability to book tickets before public announcements. They are also notified of new movie announcements by the database controller that have not been made public.

**Pay Account Fee:** This use case begins after a user registers and logs in for the first time. They will be prompted for a $20 payment by the payment controller to activate the functionality of being a registered user. Every year, they will need to pay an account fee of $20. The user can then agree to this in the terms of services.

## List of candidate objects:

| Non-Filtered List | |
|---|---|
| User | (filtered: actor) |
| Central Database | (filtered: actor) |
| User information | (filtered: property of some sort of User object) |
| User status | (filtered: property of Some sort of User object) |
| Movie | (candidate object) |
| Theatre | (candidate object) |
| Showtime | (candidate object) |
| Seat | (candidate object) |
| Payment Method | (filtered: property of User) |
| Card Information | (filtered: property of User) |
| Ticket | (candidate object) |
| Receipt | (filtered: property of email) |
| Email | (candidate object) |
| Credit | (candidate object) |
| Movie Announcement | (candidate object) |
| Ticket Controller | (candidate object) |
| Theatre Controller | (candidate object) |
| Code | (filtered: property of ticket) |
| Account | (filtered: this is what a Registered User object is) |
| Cinema Website | (filtered: another name for the system) |
| Total Cost | (filtered: property of payment controller) |
| Membership Fee | (filtered: property of registered user) |
| Payment Controller | (candidate object) |
| DataBase Controller | (candidate object) |
| GUI Controller | (candidate object) |

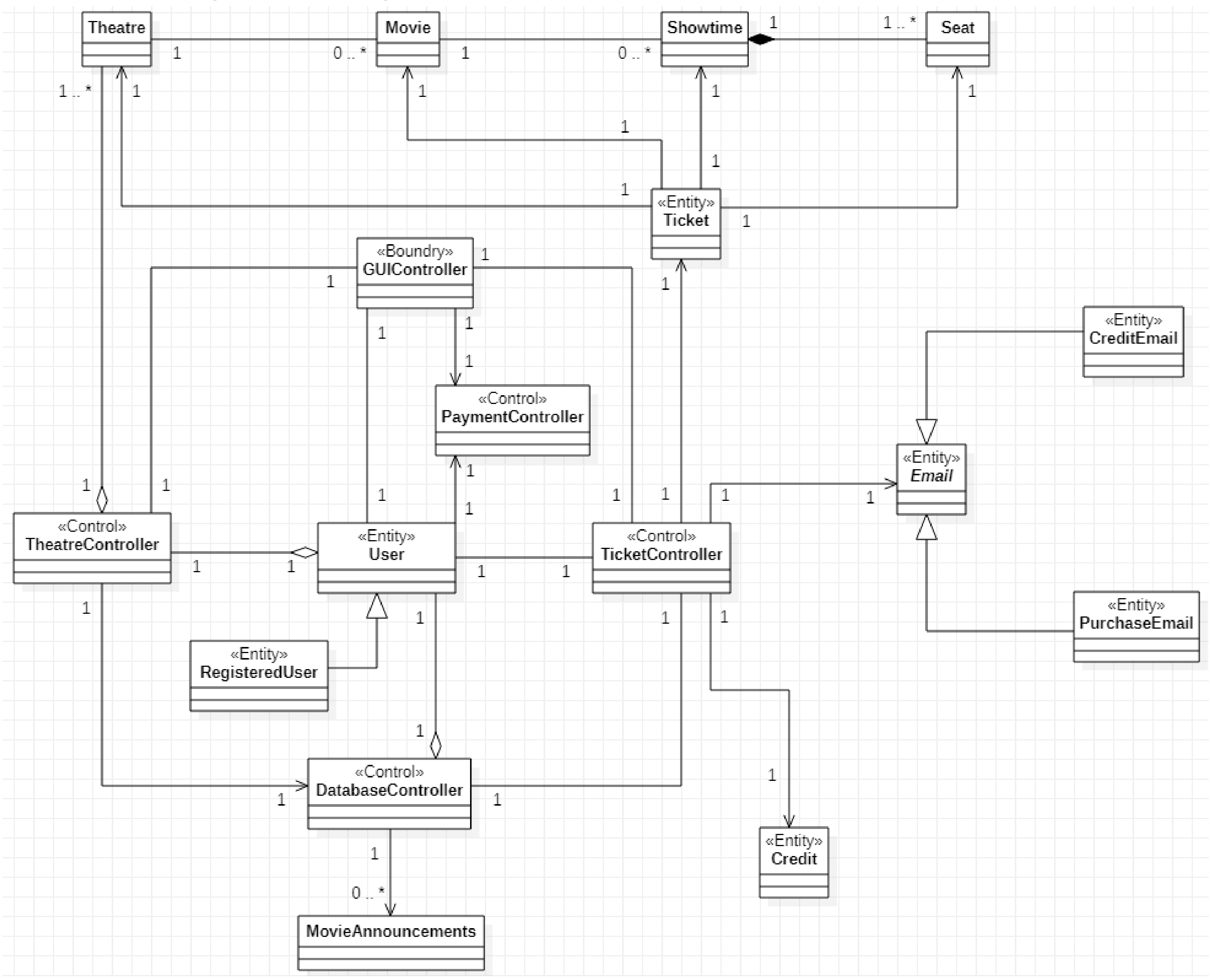
Good-Candidate-Objects:

Movie
Theatre
Showtime
Seat
Ticket
Email
Credit
Movie Announcement
Ticket Controller
Theatre Controller
DataBase Controller
Payment Controller
GUI Controller

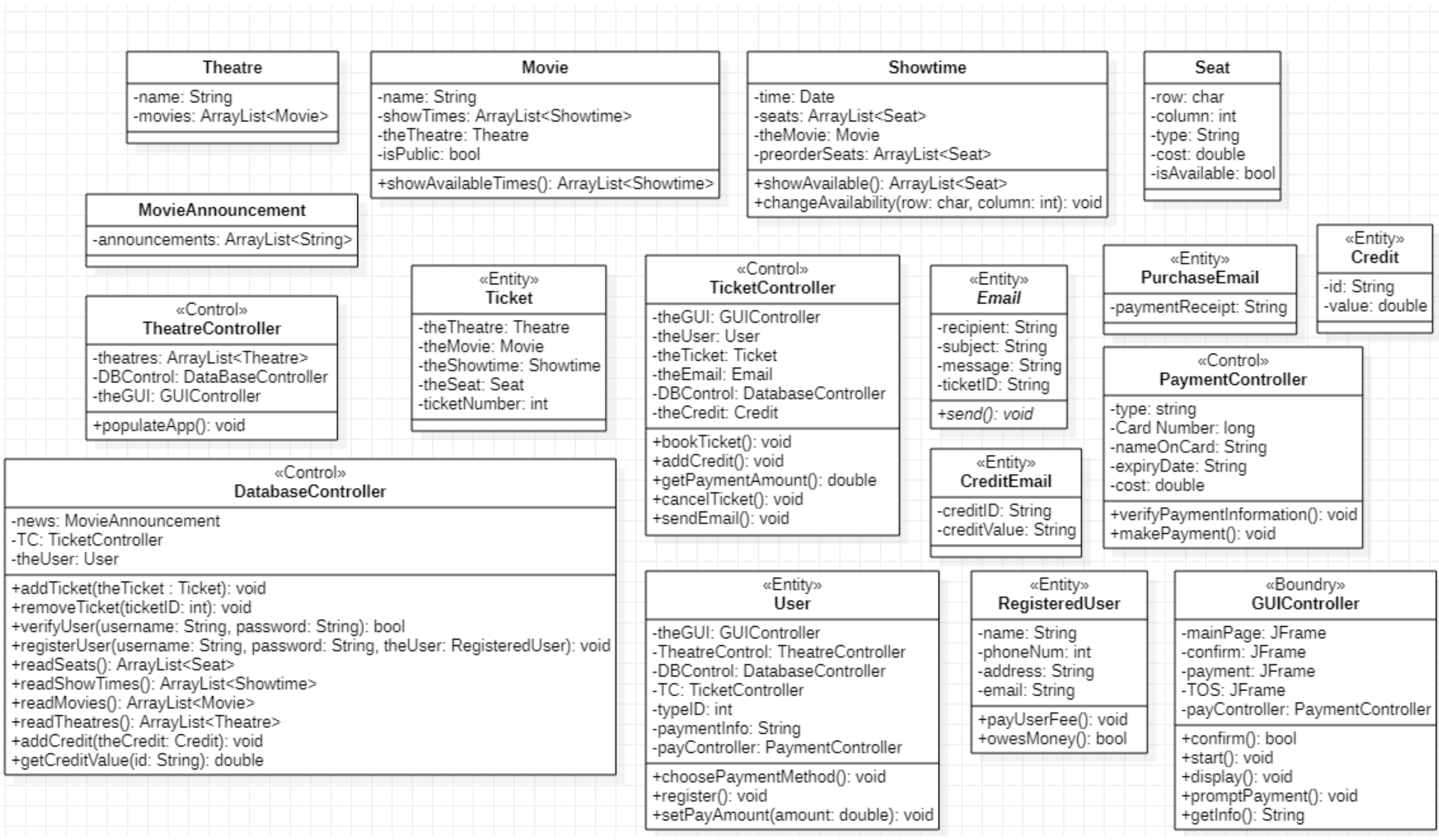
Operations for good-candidate-objects:

| | |
|---|---|
| Register for account | Operation of: Non-registered user |
| Confirm registration | Operation of: Non-registered user |
| Pay user fee | Operation of: Registered user |
| Display user profile | Operation of: Registered user |
| Browse theatre | Operation of: Theatre Controller |
| Browse movie | Operation of: Theatre |
| Browse showtimes | Operation of: Movie |
| Browse seats | Operation of: Showtimes |
| Confirm selections | Operation of: Ticket Controller |
| Choose payment method | Operation of: User |
| Confirm payment | Operation of: Payment Controller |
| Send email | Operation of: Ticket Controller |
| Book ticket | Operation of: Ticket Controller |
| Cancel ticket | Operation of: Ticket Controller |
| Update database | Operation of: DataBase Controller |
| Send credits | Operation of: Ticket Controller |
| Send movie announcement | Operation of: DataBase Controller |
| Generate Ticket | Operation of: Ticket Controller |
| Display various information | Operations of: GUIController |
| Prompt User for information | Operations of: GUIController |

## Design Class Diagram:

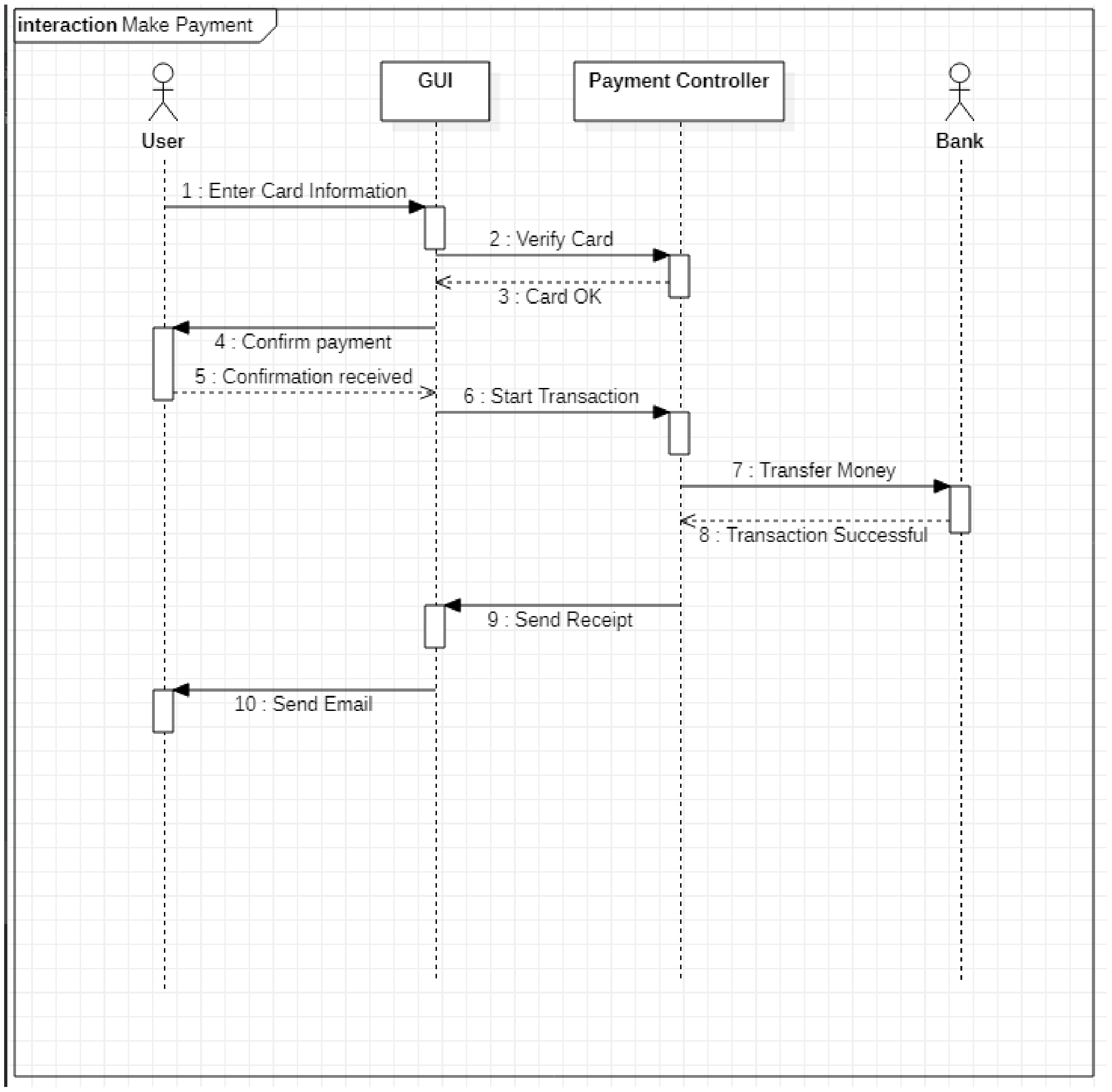


## Classes Without Relationships with Operations and Attributes:

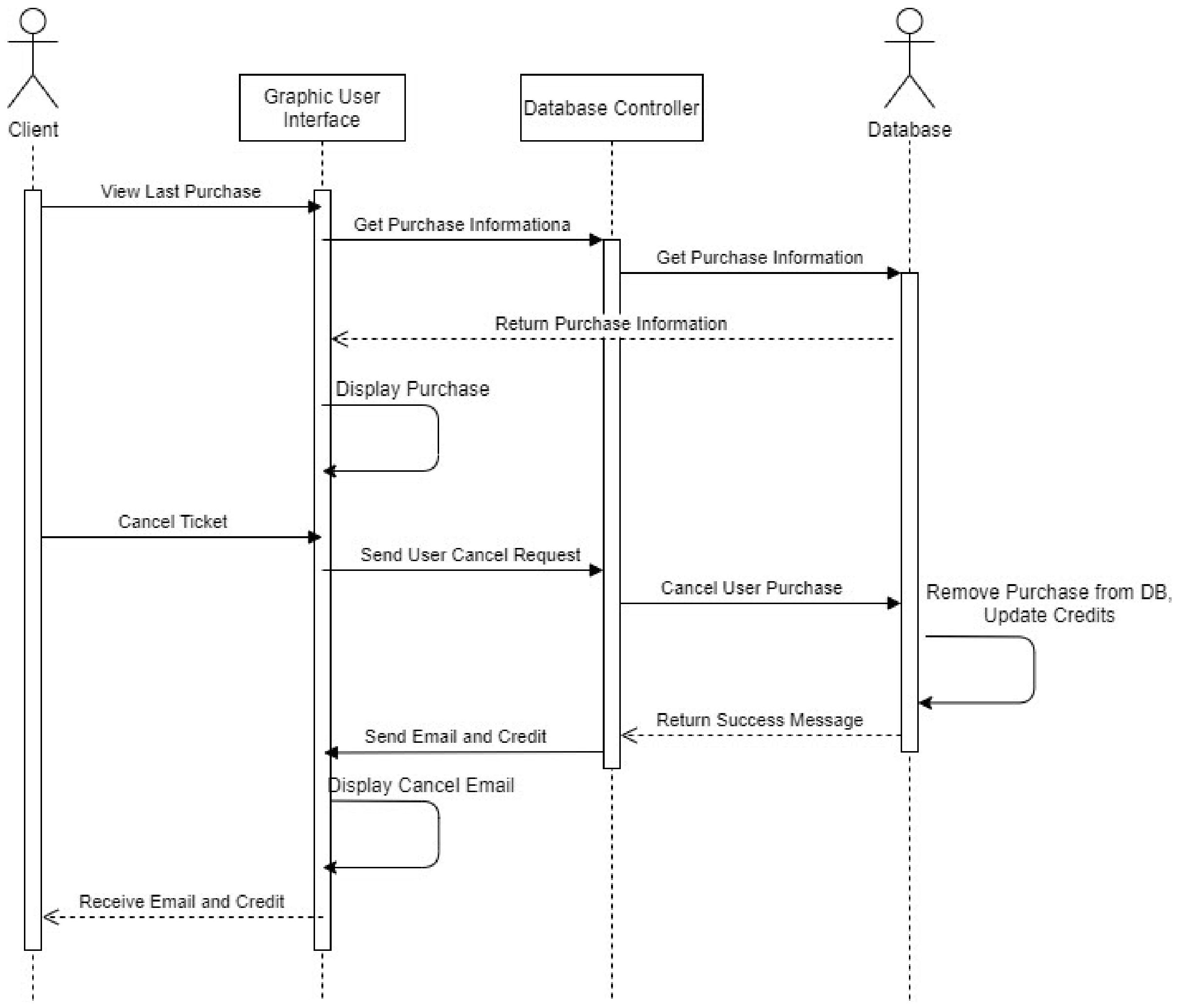


## Interactions diagrams:

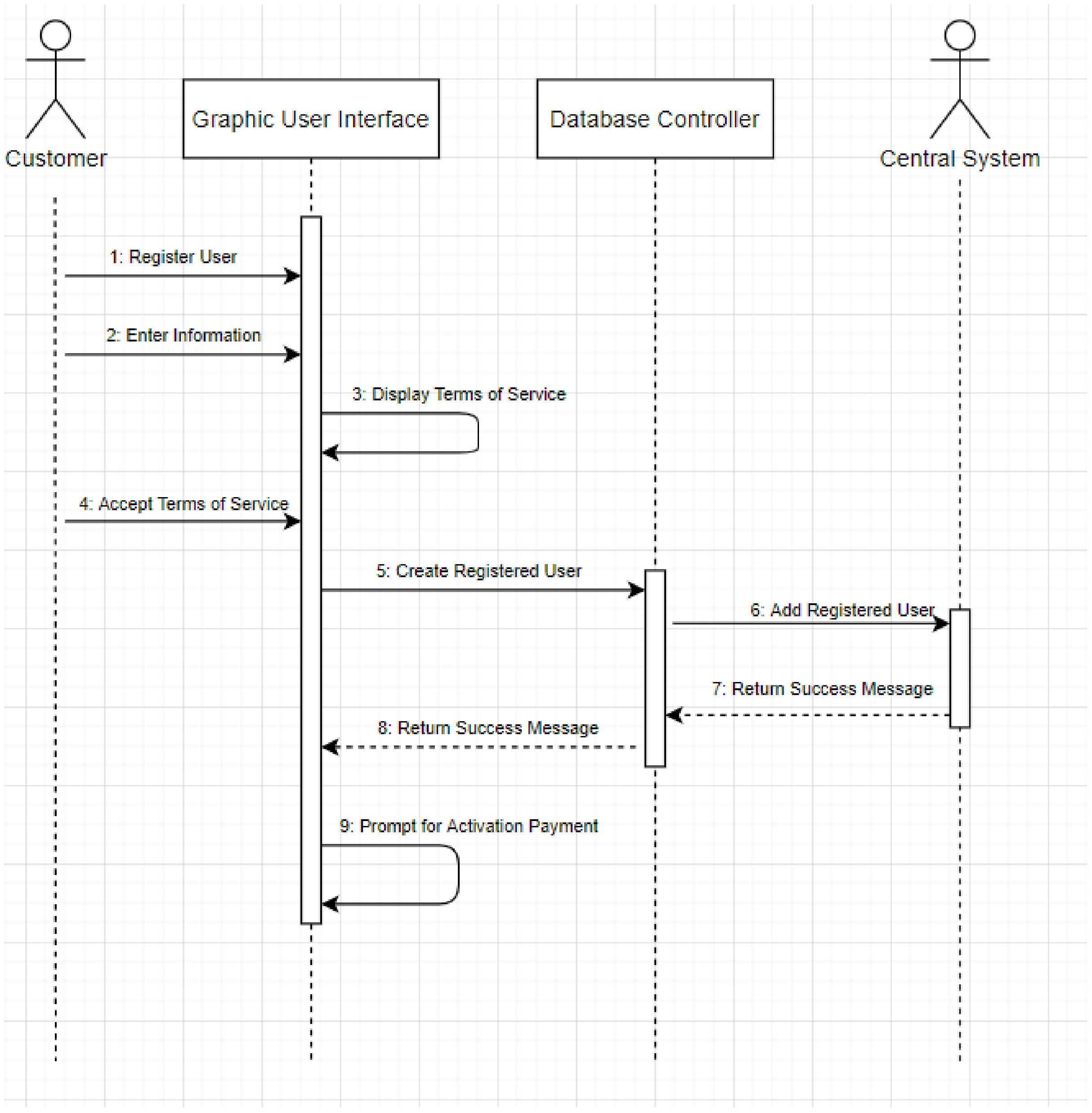
Junhao Xue - Make Payment



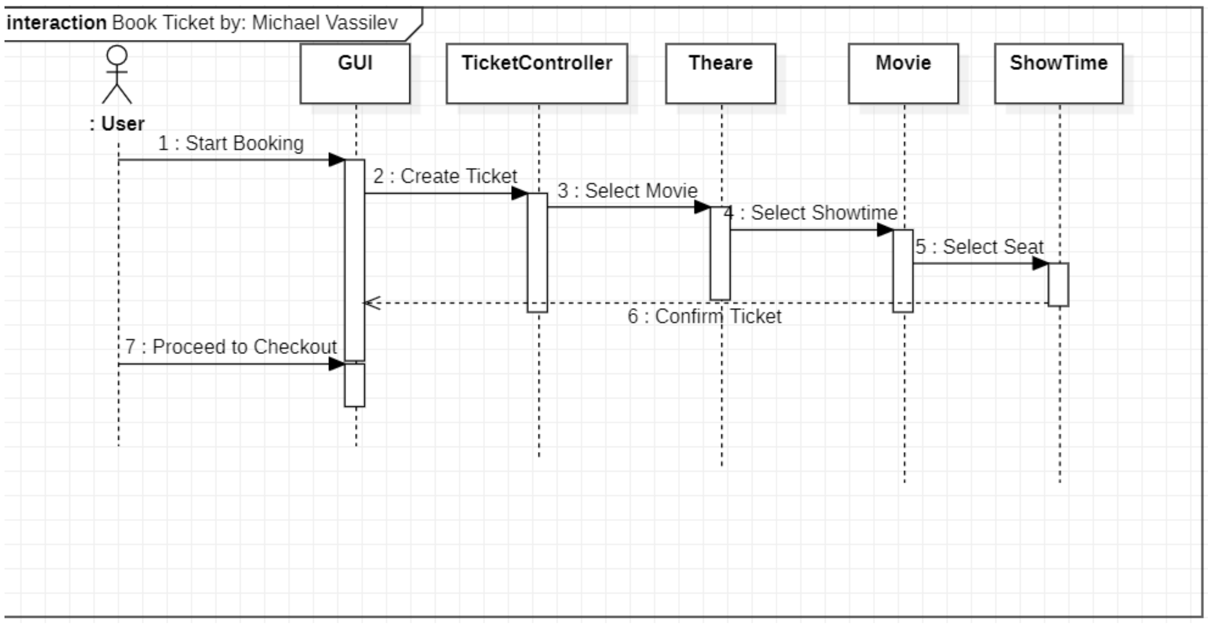
Minh Vo - Cancel Ticket



Matthew Wells - Register User

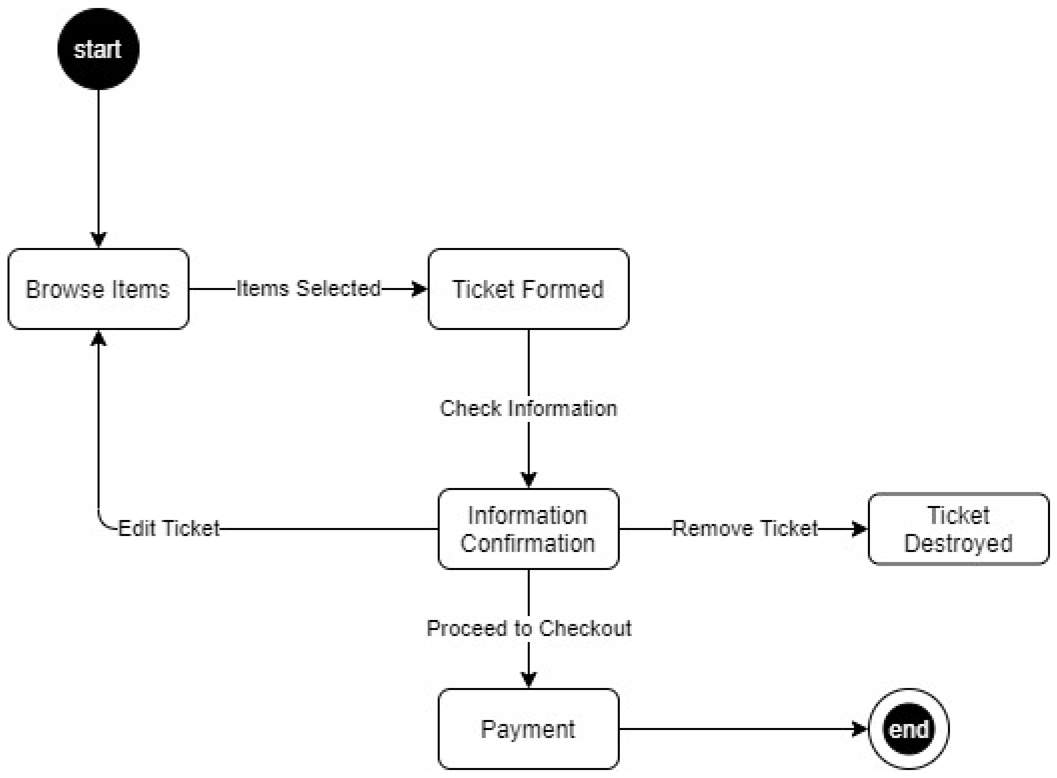

Customer
Graphic User Interface
Database Controller
Central System
1: Register User
2: Enter Information
3: Display Terms of Service
4: Accept Terms of Service
5: Create Registered User
6: Add Registered User
7: Return Success Message
8: Return Success Message
9: Prompt for Activation Payment

Michael Vassilev - Book Ticket

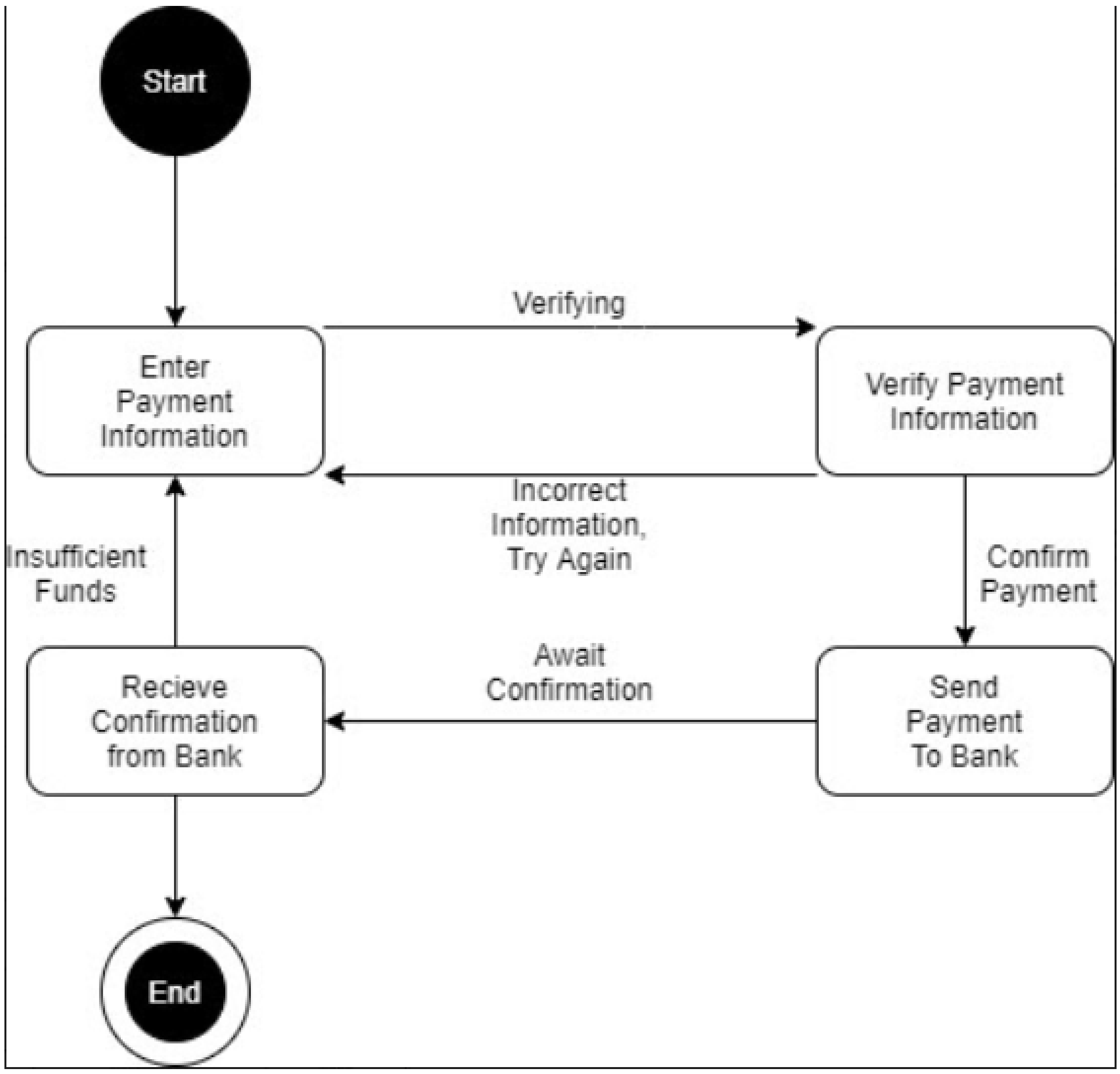


## Statement transition diagrams:

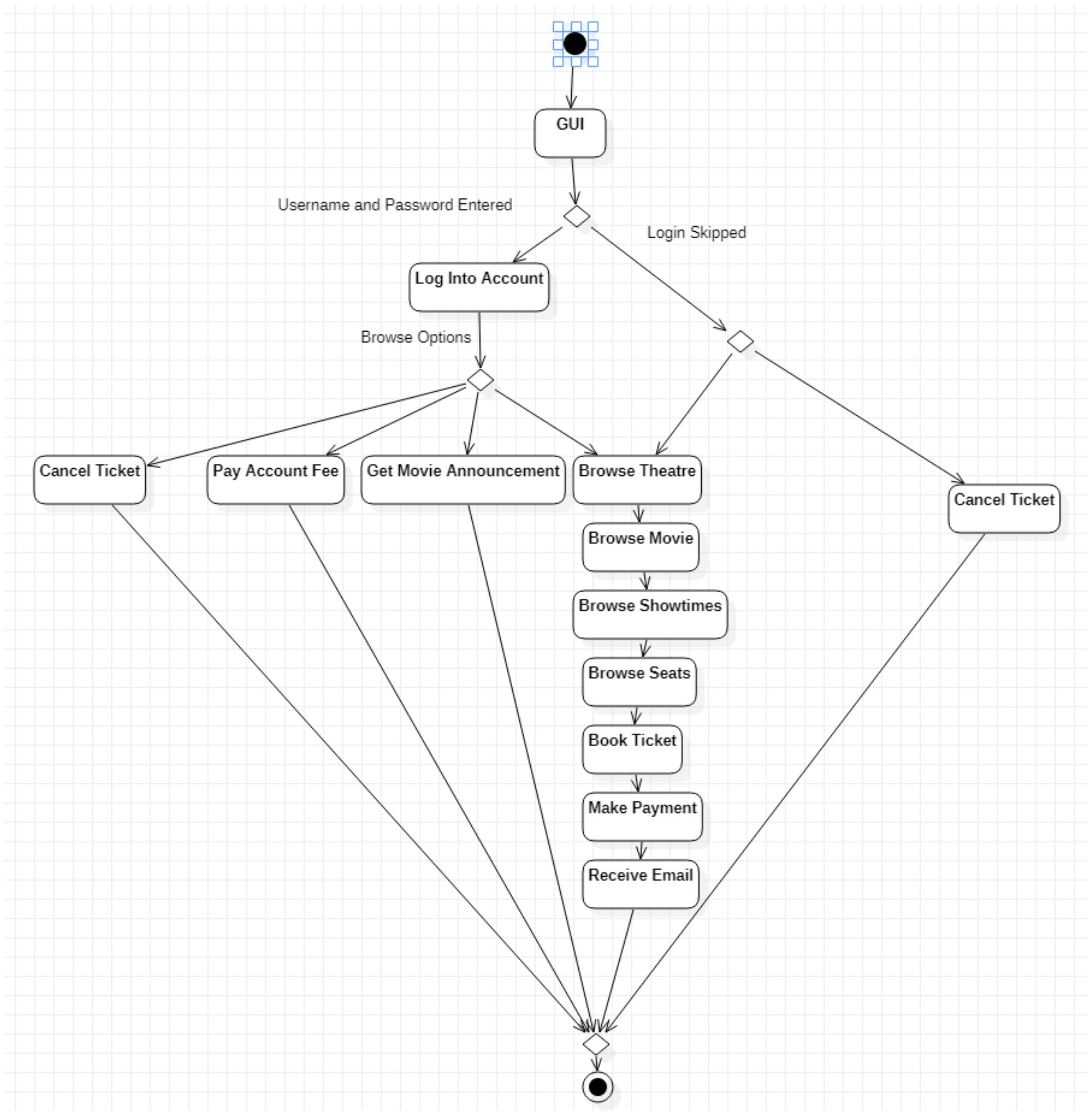
Ticket



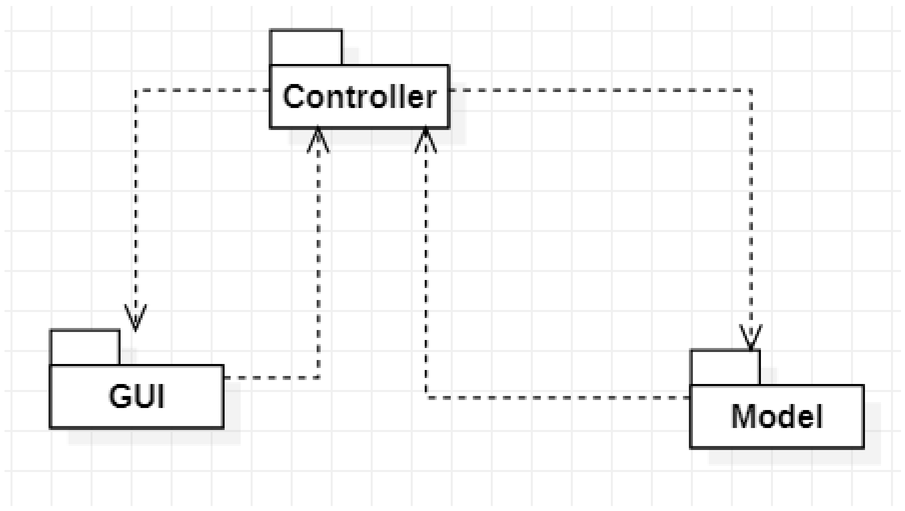
Payment


Start
Enter Payment Information
Verifying
Verify Payment Information
Incorrect Information, Try Again
Confirm Payment
Insufficient Funds
Await Confirmation
Send Payment To Bank
Recieve Confirmation from Bank
End

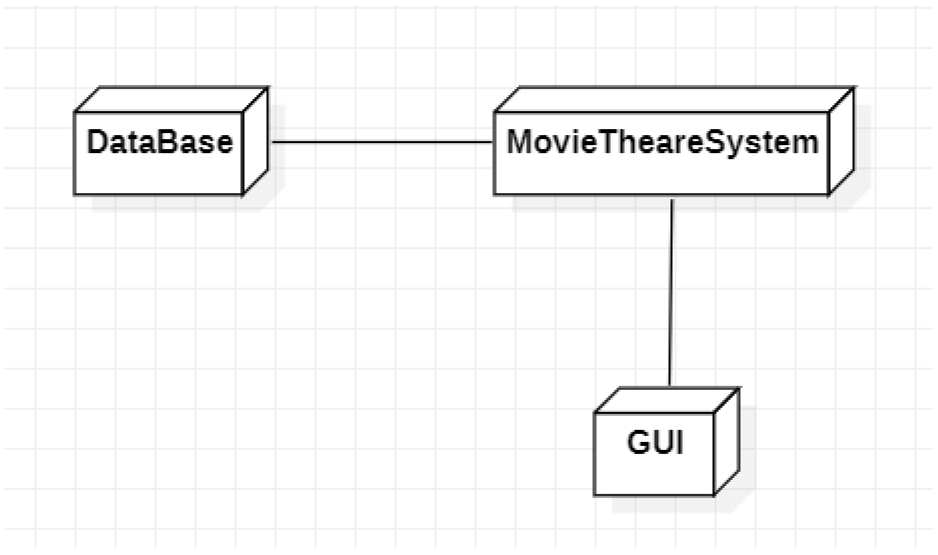
## System activity diagram:



## System package diagram:



## System deployment diagram:


DataBase
MovieTheareSystem
GUI