Principles of Software Design - Final Project
ENSF 614 - Fall 2022 - Group 9

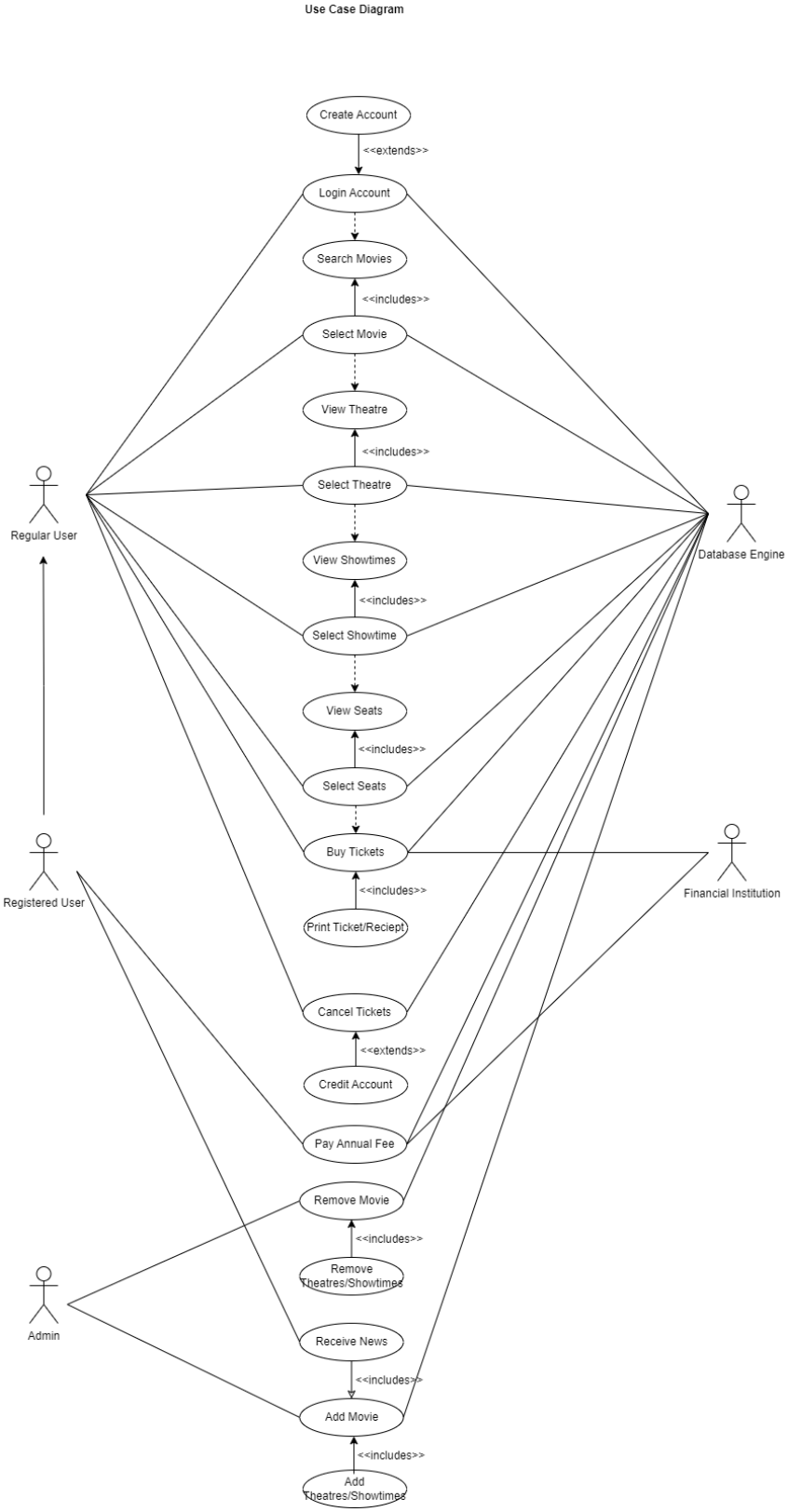
Instructor: Mahmood Moussavi

Group Members: Vishal Ravindran, Tyson Trail,
Kendall Reed, Aaron Manuel

Submission Date: December 04, 2022

Table of Contents

Use-Case Diagram



Use-Case Scenarios

Scenario 1: Login

The scenario starts with a user choosing to login as a Regular User, Registered User, or creating a new Registered User account. Choosing the Regular User option will proceed to the Search Movie use case. Logging in as a Registered User will verify with the database that the login is correct, then proceed to the Search Movie use case. Creating a new Registered User will require the user entering in their name, address, credit and/or debit card account information, and confirming registration with the database then proceeding to the Search Movie use case.

Scenario 2: Search Movies

The scenario starts with a user looking for a movie from a list of movies from the database using a movie search bar. The database returns the movie if it is found, or informs the user that the movie was not found if it is not in the database. The user then selects the movie they want to watch and then proceeds to select theatre. Registered users can search for and view unannounced movies (movies that have been flagged as not publicly announced).

Scenario 3: Select Theatre

The scenario starts with the user viewing the theatres that the movie is available, and select theatre to proceed to view showtimes from.

Scenario 4: Select Showtime

The scenario starts with the user viewing the showtimes for the theatre in the database. The user then selects the showtime that they want from the list of showtimes. If the seats in the showtime are full, then the user cannot select that showtime.

Scenario 5: Select Seats

The user will be able to view graphically available seats, supplied from the database. The user can select a single seat based on availability, and will proceed to the Buy Ticket Use Case. If the user is a Registered User, and the movie is not publicly announced, a Registered User will be able to select a seat only if 10% of the seats have not already been purchased.

Scenario 6: Buy Tickets

The scenario starts with the regular user entering in their credit card information into the form, then pressing confirm. The credit card information is then confirmed with the financial institution. The database then assigns a ticket with a ticket ID to the database. The regular user is then emailed the ticket with ticket information.

The scenario is the same for the registered user, except that their payment (credit card or debit card) information is saved in the database. The database populates the form before the user presses confirm.

Scenario 7: Cancel Tickets

The scenario starts with the user viewing their bought tickets. The user then selects the ticket that they want to cancel. If the showtime of the ticket is more than 72 hours away, the cancellation is processed. A Regular User then receives a payment of the original ticket price minus a 15% administration fee that can be credited to a future purchase from up to a year from the current date. A Registered User receives the full payment amount from the ticket purchase.

Scenario 8: Registered User Pays Annual Fee

The scenario starts with the Registered User choosing to upgrade their membership status, or has their membership status expire. If they are a new Registered User, they are prompted to make a payment of the annual fee. If the last payment of the Registered User's annual fee was a year ago, they are prompted to make another payment of the annual fee.

Scenario 9: Registered User Receives Movie News

When new movie news is added by the admin to the database, the movie news is queried from the database and is sent to the Registered User's email.

Scenario 10: Admin Adds Movie

When a movie is added by the admin to the database, a message is sent to the registered users by email.

Scenario 11: Admin Updates to Public Movie

In this scenario an Admin updates a movies status in the database as publicly available. registered users are emailed news about the update. The movie is now viewable by regular users in the search movie use case, and registered users can now select and buy more than 10% of the available seats.

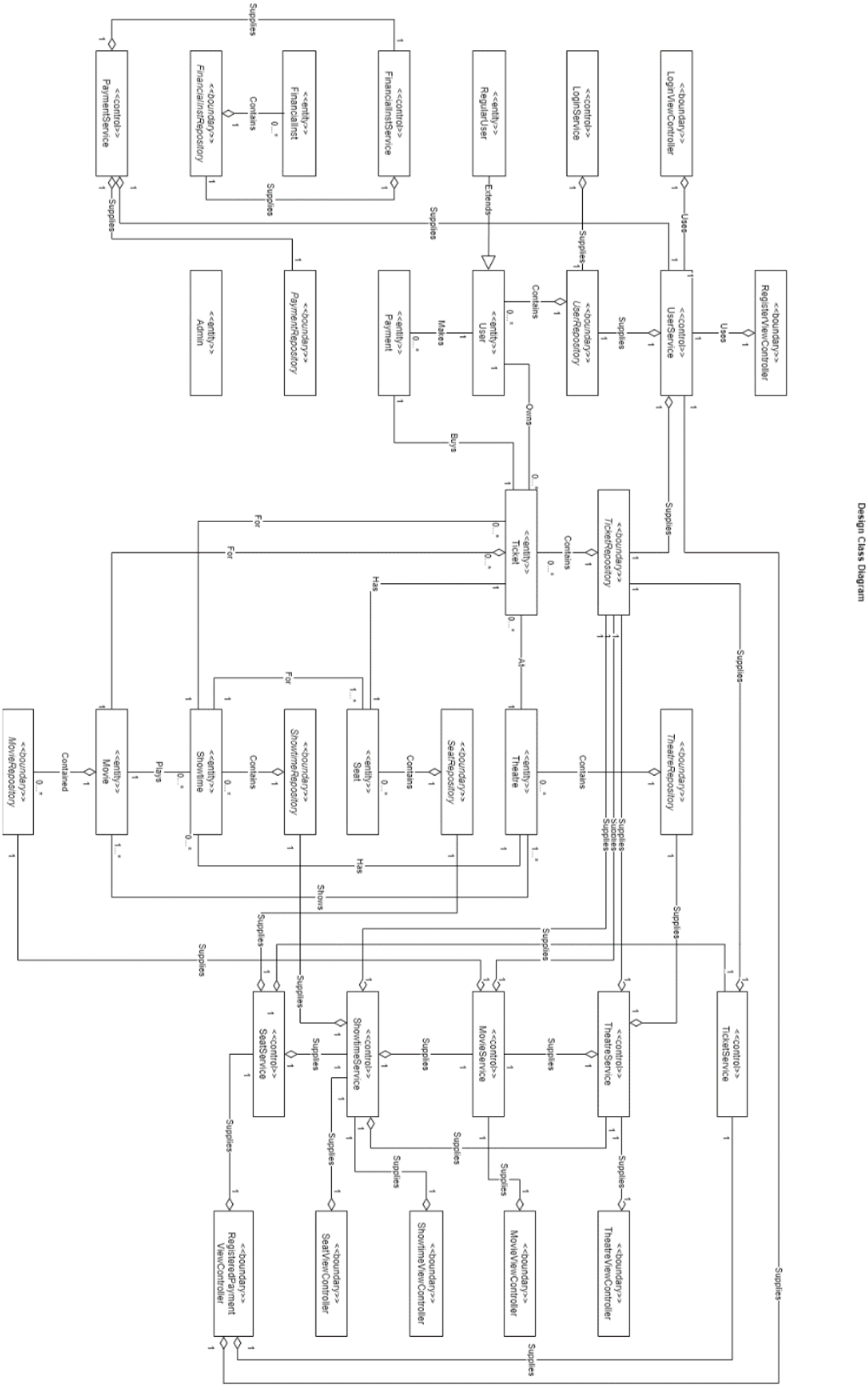
Scenario 12: Admin Removes Movie

In this scenario admin removes the movie from the database and a message is sent to the registered users by email.

List of Candidate Objects that are Traceable in Use-Case Scenarios

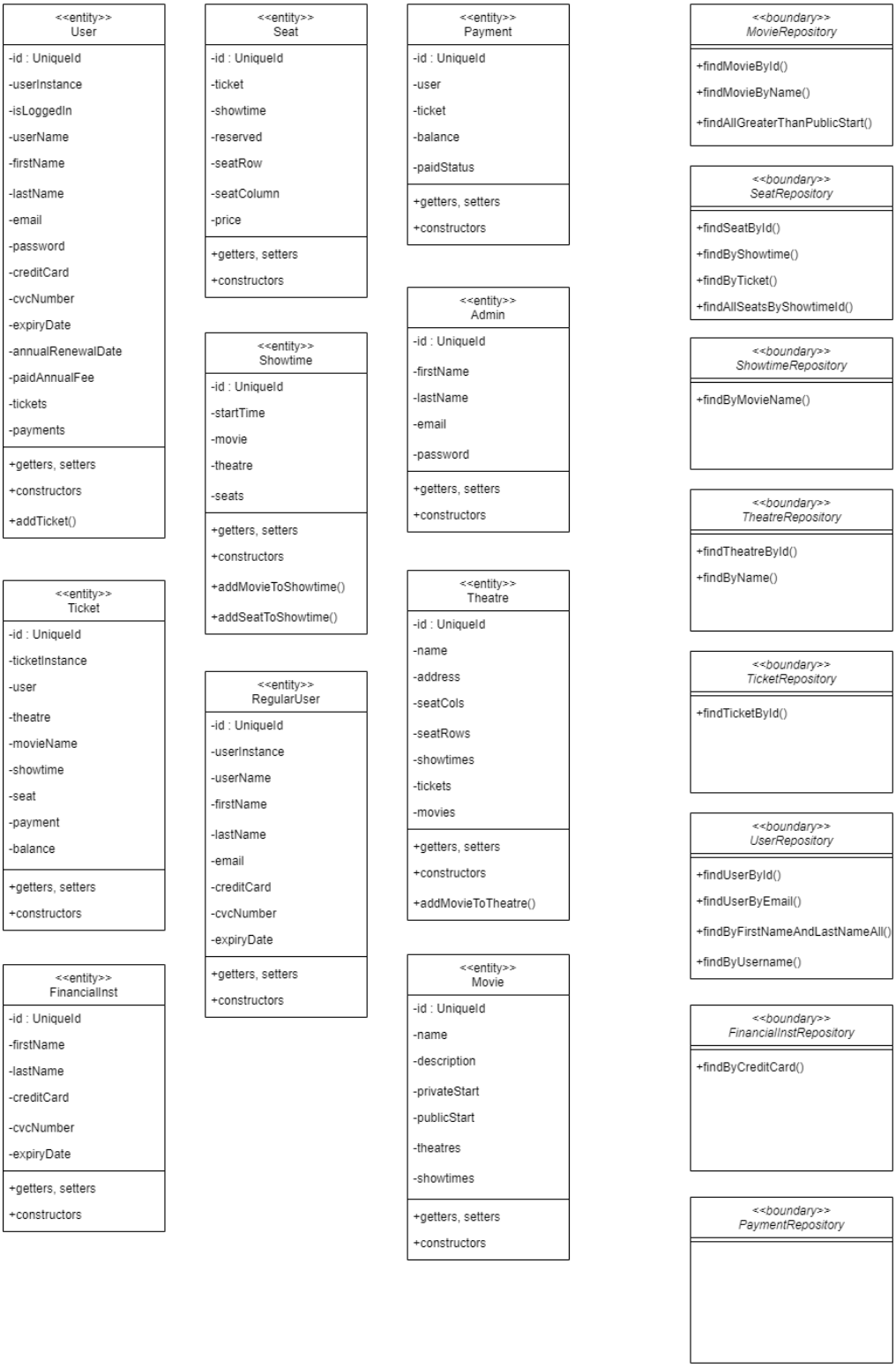
- User
- Regular User
- Registered User
- Admin
- Movie
- Movie Search bar
- Theatre
- Database
- Showtime
- Seat
- Payment
- Credit card
- Debit card

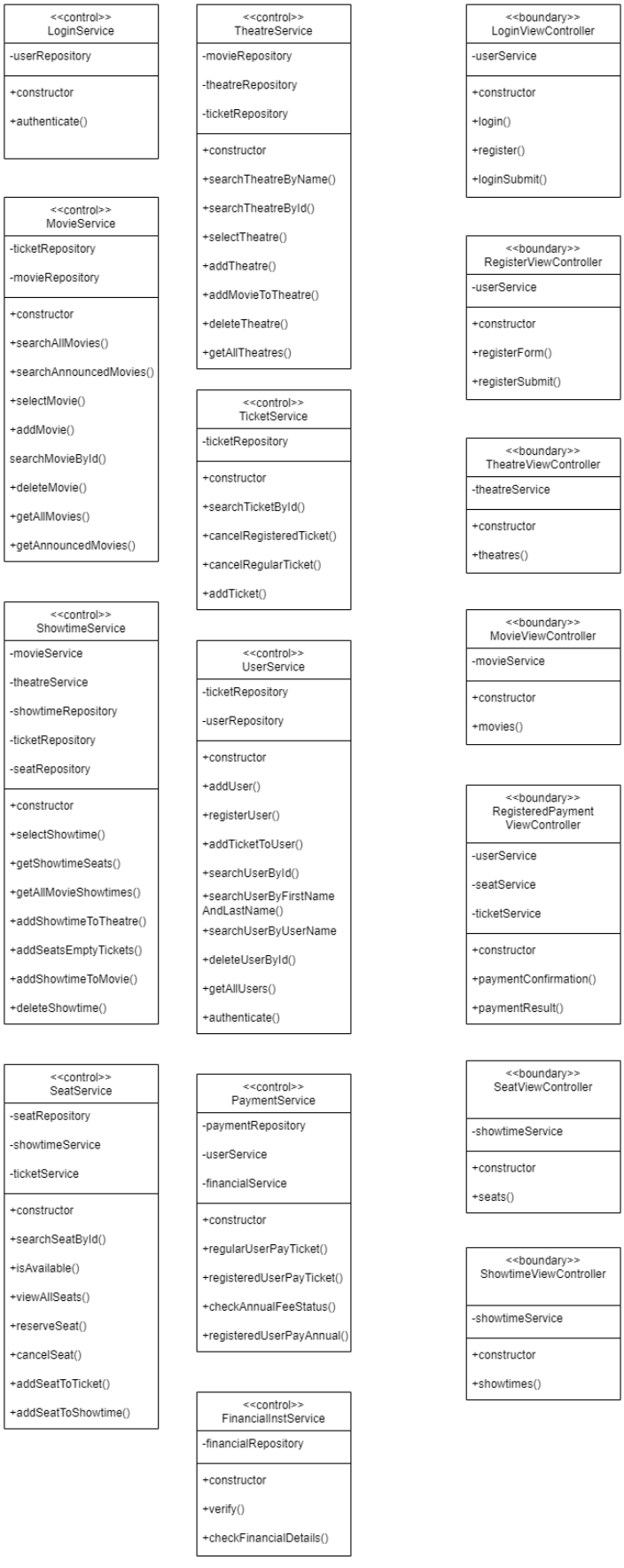
Design Class-Diagram



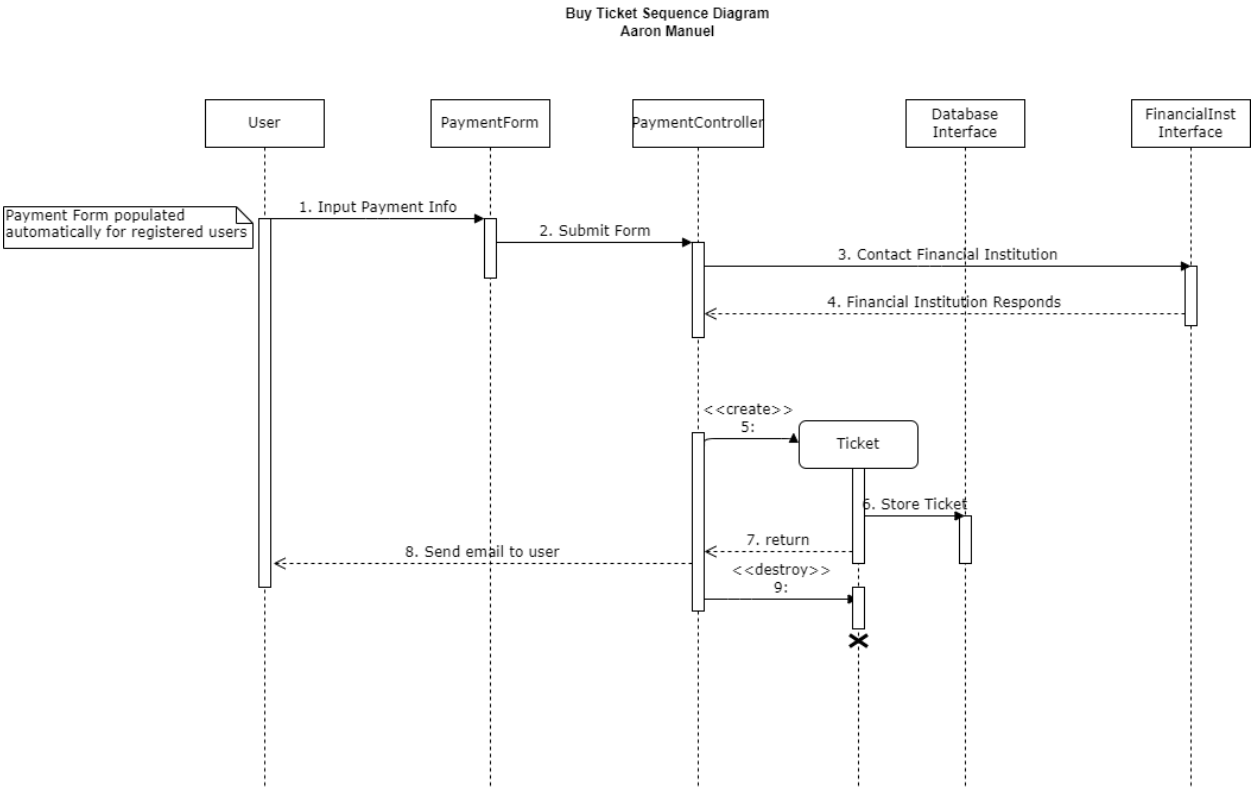
Class Diagrams Without Relationships

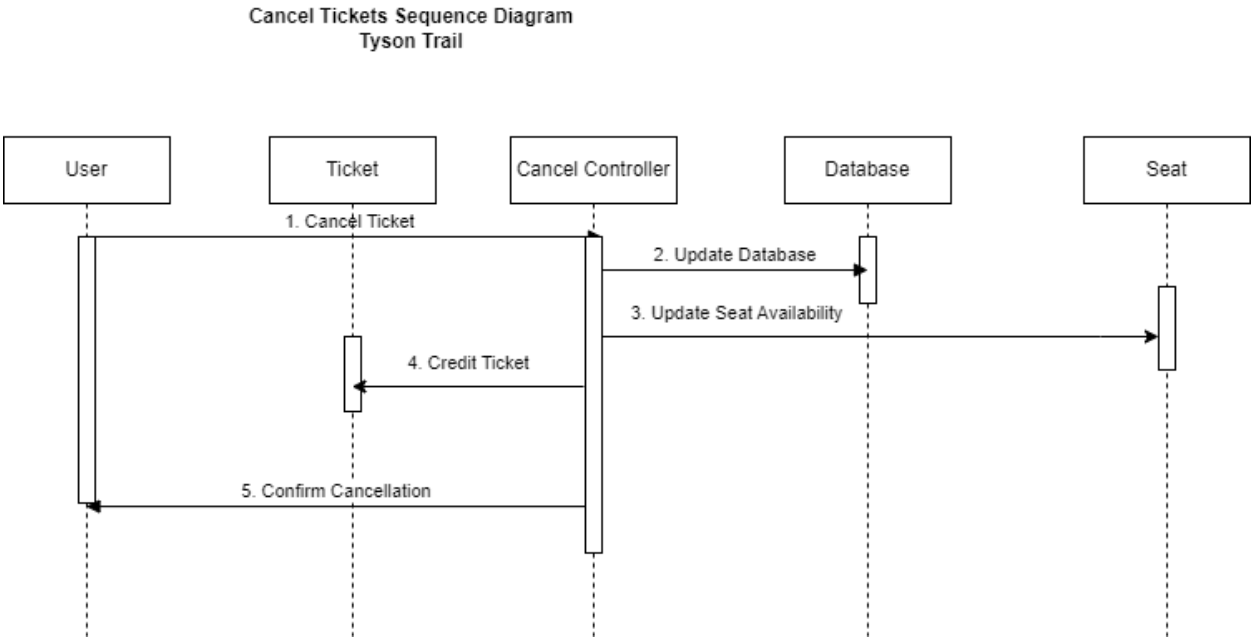
Class Diagrams (No Relations)



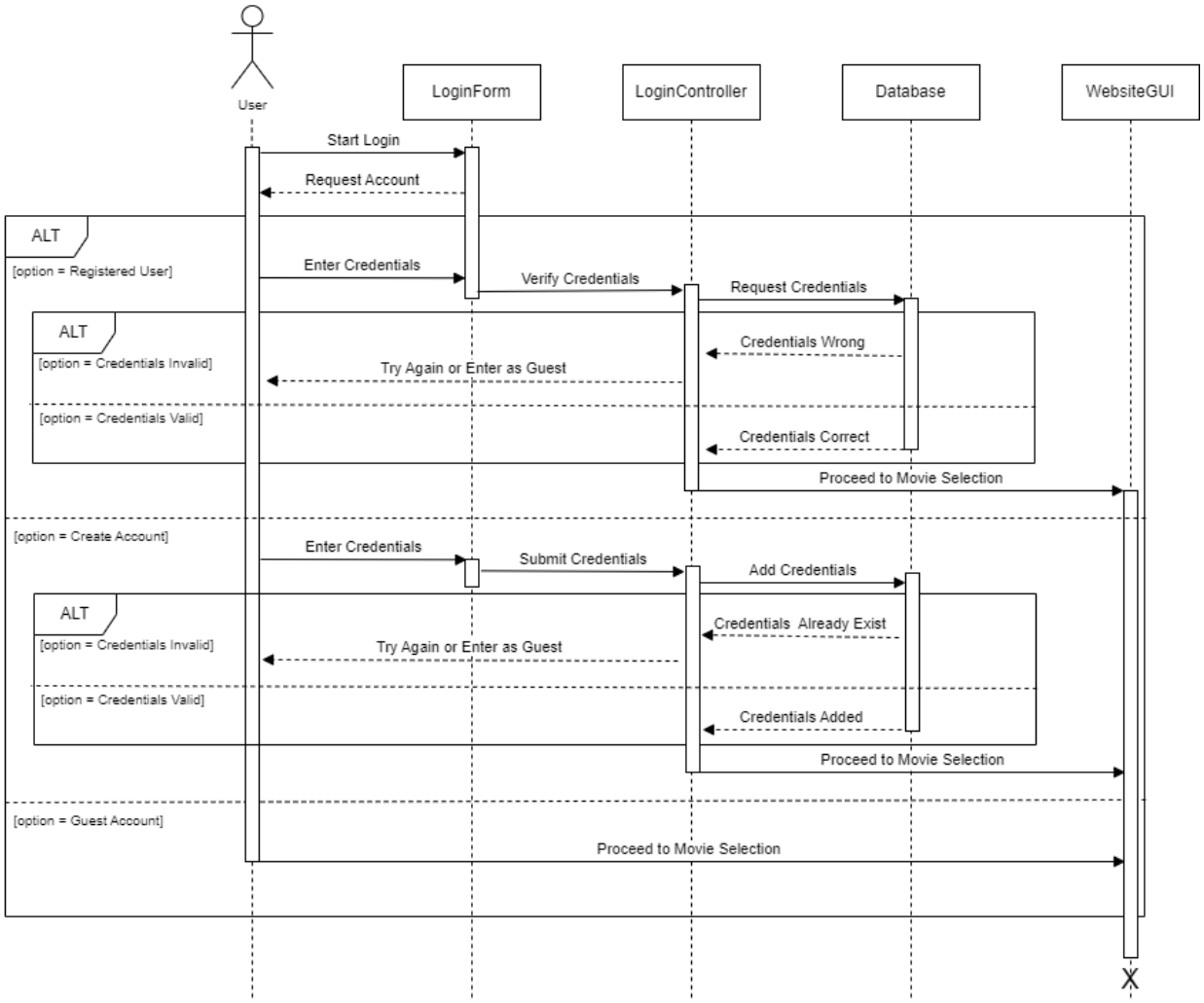

<<control>>
LoginService
-userRepository
+constructor
+authenticate()
<<control>>
MovieService
-ticketRepository
-movieRepository
+constructor
+searchAllMovies()
+searchAnnouncedMovies()
+selectMovie()
+addMovie()
searchMovieById()
+deleteMovie()
+getAllMovies()
+getAnnouncedMovies()
<<control>>
ShowtimeService
-movieService
-theatreService
-showtimeRepository
-ticketRepository
-seatRepository
+constructor
+selectShowtime()
+getShowtimeSeats()
+getAllMovieShowtimes()
+addShowtimeToTheatre()
+addSeatsEmptyTickets()
+addShowtimeToMovie()
+deleteShowtime()
<<control>>
SeatService
-seatRepository
-showtimeService
-ticketService
+constructor
+searchSeatById()
+isAvailable()
+viewAllSeats()
+reserveSeat()
+cancelSeat()
+addSeatToTicket()
+addSeatToShowtime()
<<control>>
TheatreService
-movieRepository
-theatreRepository
-ticketRepository
+constructor
+searchTheatreByName()
+searchTheatreById()
+selectTheatre()
+addTheatre()
+addMovieToTheatre()
+deleteTheatre()
+getAllTheatres()
<<control>>
TicketService
-ticketRepository
+constructor
+searchTicketById()
+cancelRegisteredTicket()
+cancelRegularTicket()
+addTicket()
<<control>>
UserService
-ticketRepository
-userRepository
+constructor
+addUser()
+registerUser()
+addTicketToUser()
+searchUserById()
+searchUserByFirstName AndLastName()
+searchUserByUserName
+deleteUserById()
+getAllUsers()
+authenticate()
<<control>>
PaymentService
-paymentRepository
-userService
-financialService
+constructor
+regularUserPayTicket()
+registeredUserPayTicket()
+checkAnnualFeeStatus()
+registeredUserPayAnnual()
<<control>>
FinancialInstService
-financialRepository
+constructor
+verify()
+checkFinancialDetails()
<<boundary>>
LoginViewController
-userService
+constructor
+login()
+register()
+loginSubmit()
<<boundary>>
RegisterViewController
-userService
+constructor
+registerForm()
+registerSubmit()
<<boundary>>
TheatreViewController
-theatreService
+constructor
+theatres()
<<boundary>>
MovieViewController
-movieService
+constructor
+movies()
<<boundary>>
RegisteredPayment ViewController
-userService
-seatService
-ticketService
+constructor
+paymentConfirmation()
+paymentResult()
<<boundary>>
SeatViewController
-showtimeService
+constructor
+seats()
<<boundary>>
ShowtimeViewController
-showtimeService
+constructor
+showtimes()

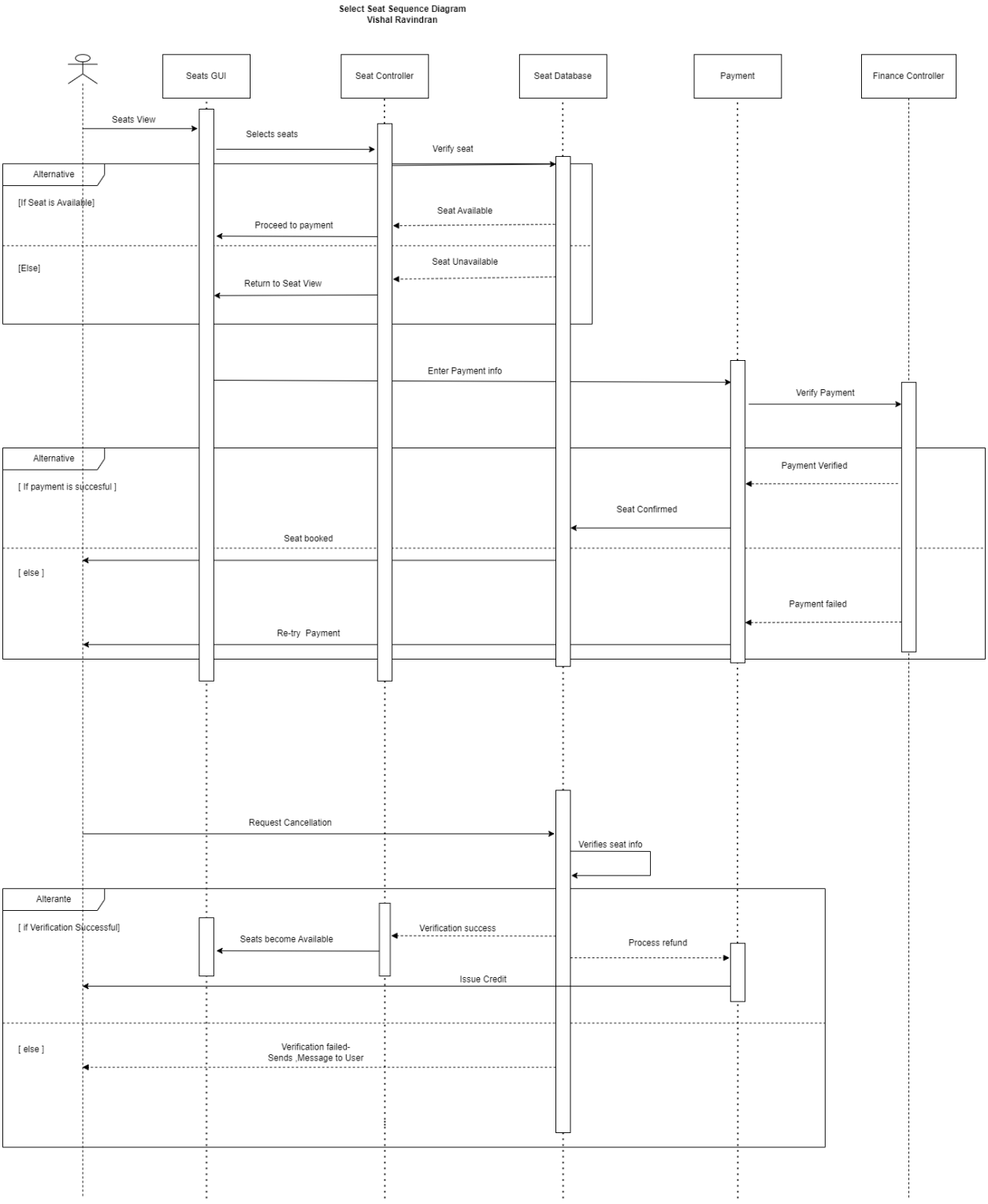
Interaction Diagrams (Sequence Diagram)



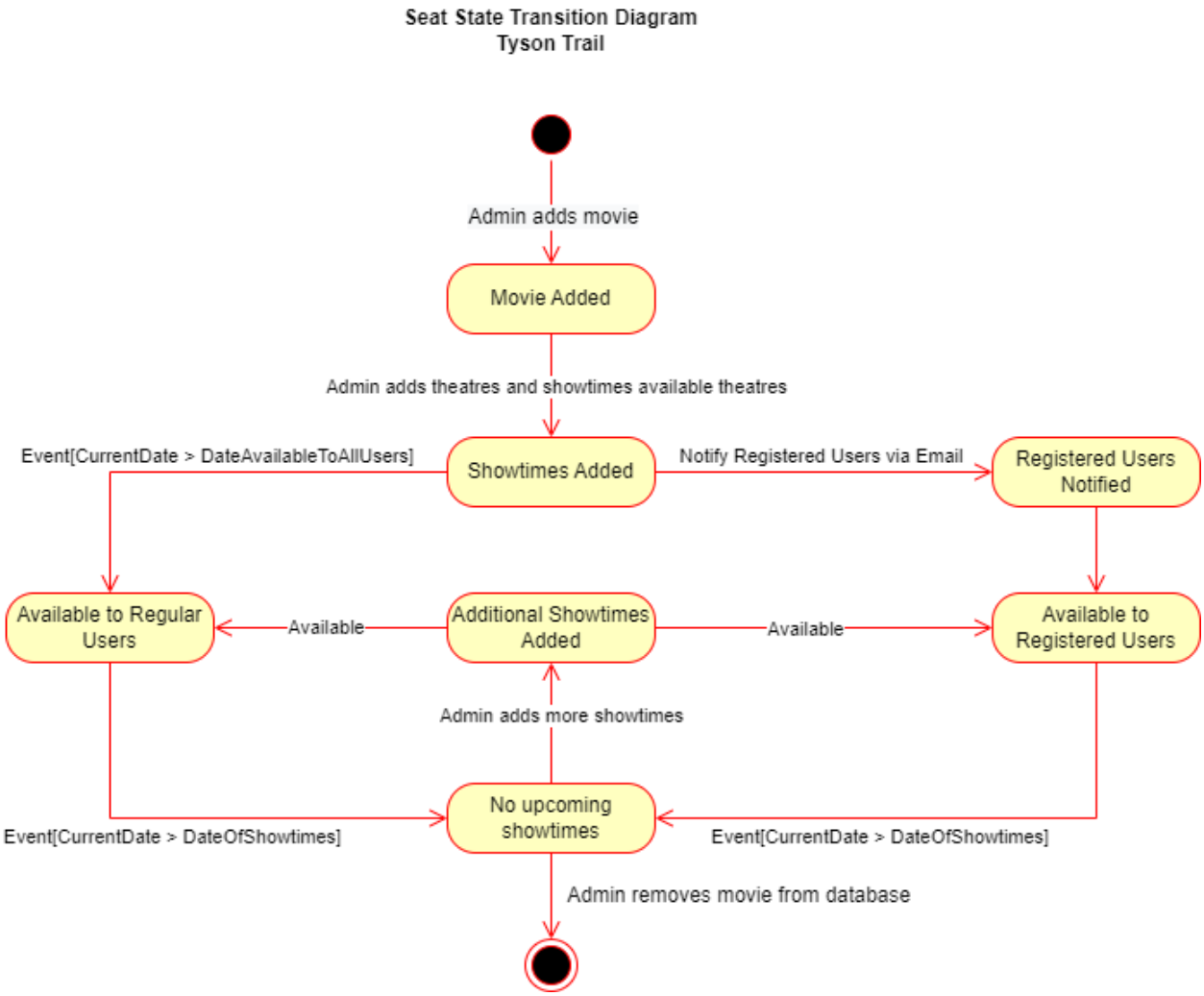


Login Sequence Diagram
Kendall Reed

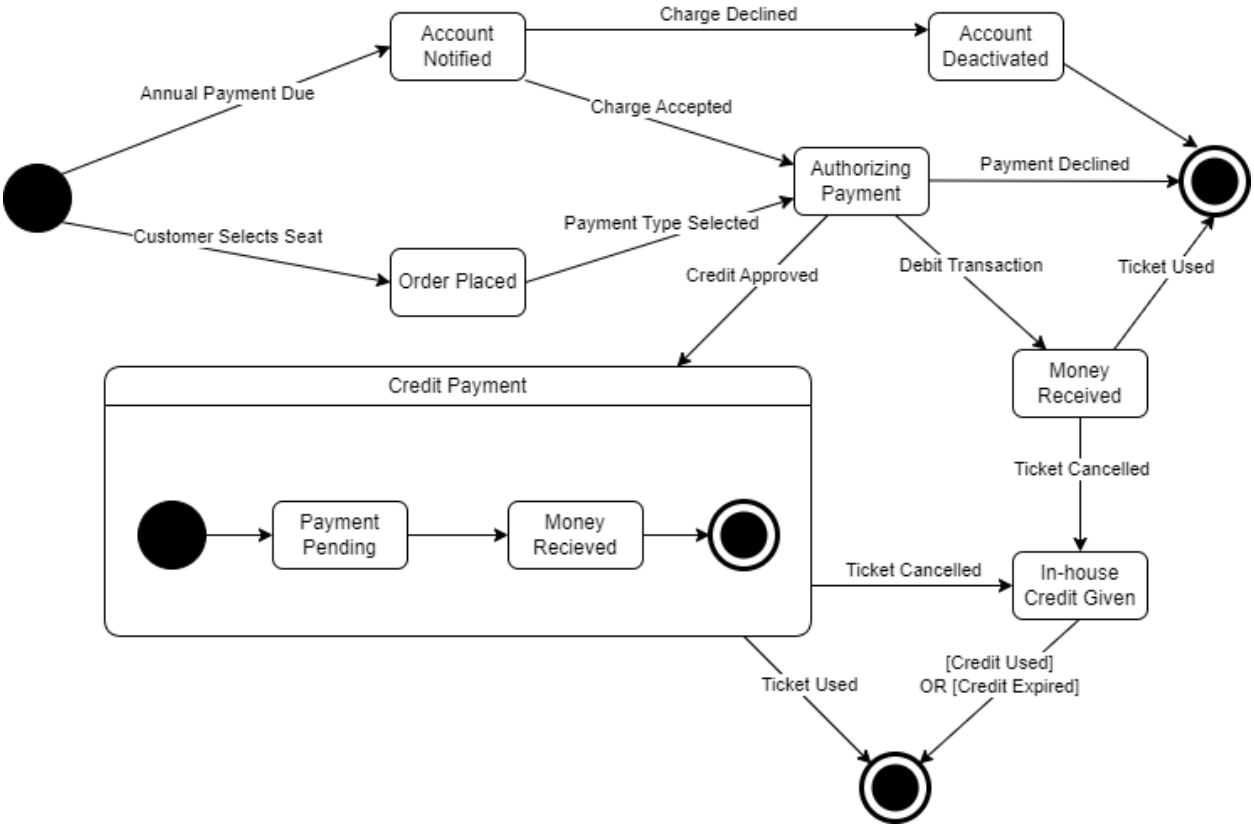



Select Seat Sequence Diagram
Vishal Ravindran
Seats GUI
Seat Controller
Seat Database
Payment
Finance Controller
Seats View
Selects seats
Verify seat
Alternative
[If Seat is Available]
Seat Available
Proceed to payment
[Else]
Seat Unavailable
Return to Seat View
Enter Payment info
Verify Payment
Alternative
[If payment is succesful]
Payment Verified
Seat Confirmed
Seat booked
[else]
Payment failed
Re-try Payment
Request Cancellation
Verifies seat info
Alterante
[if Verification Successful]
Verification success
Seats become Available
Process refund
Issue Credit
[else]
Verification failed-
Sends ,Message to User

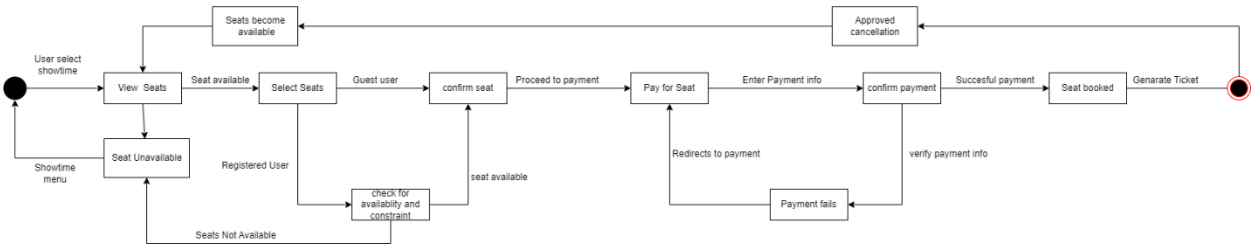
State Transition Diagrams

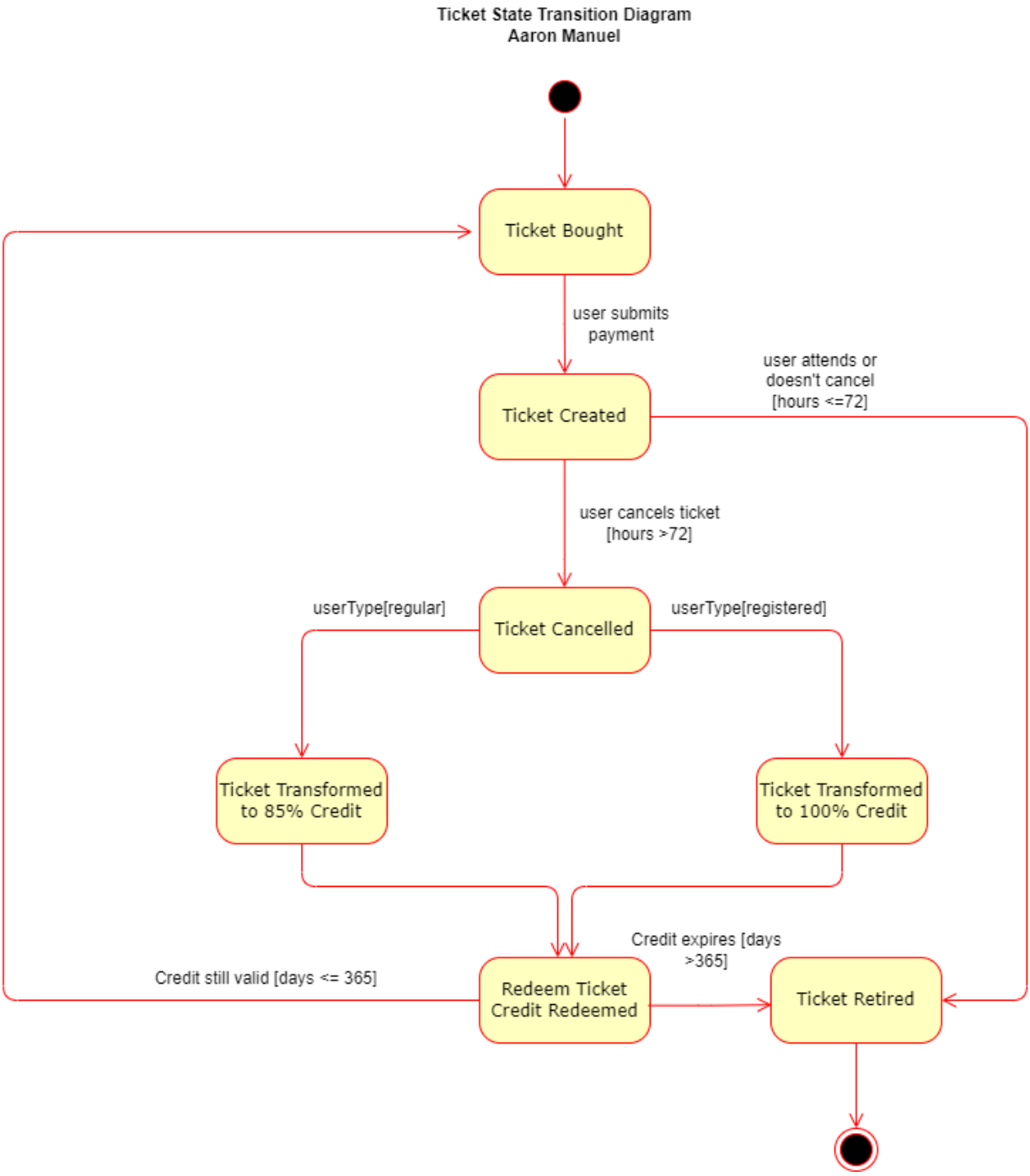


Payment State Transition Diagram
Kendall Reed

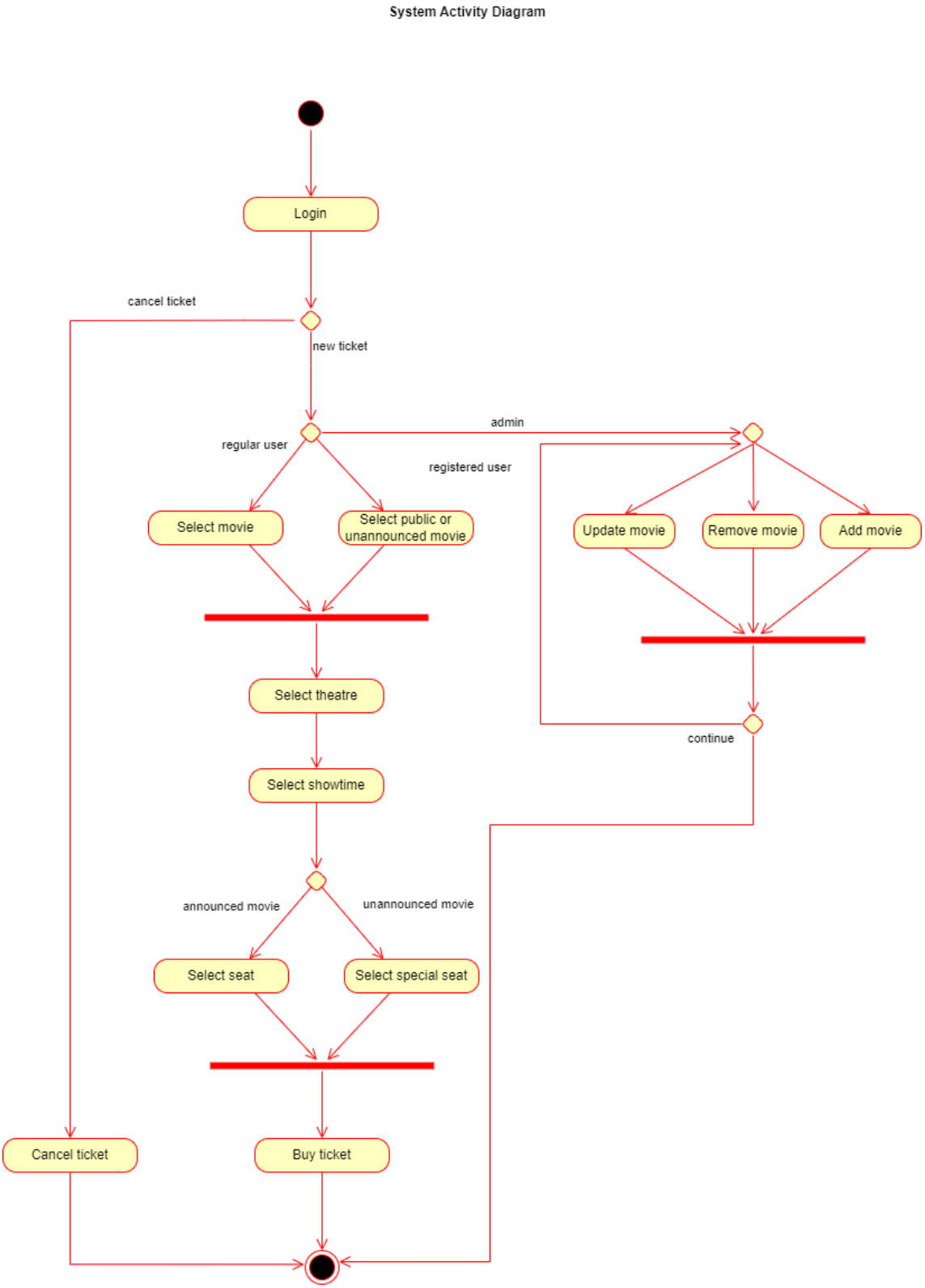


Seat State Transition Diagram
Vishal Ravindran

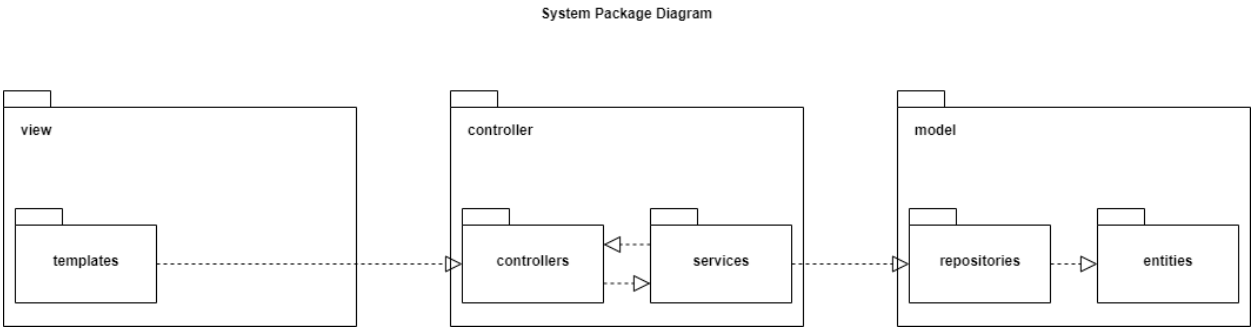



Ticket State Transition Diagram
Aaron Manuel
Ticket Bought
user submits payment
Ticket Created
user attends or doesn't cancel [hours <=72]
user cancels ticket [hours >72]
Ticket Cancelled
userType[regular]
userType[registered]
Ticket Transformed to 85% Credit
Ticket Transformed to 100% Credit
Credit expires [days >365]
Credit still valid [days <= 365]
Redeem Ticket Credit Redeemed
Ticket Retired

System Activity Diagram



System Package Diagram



System Deployment Diagram