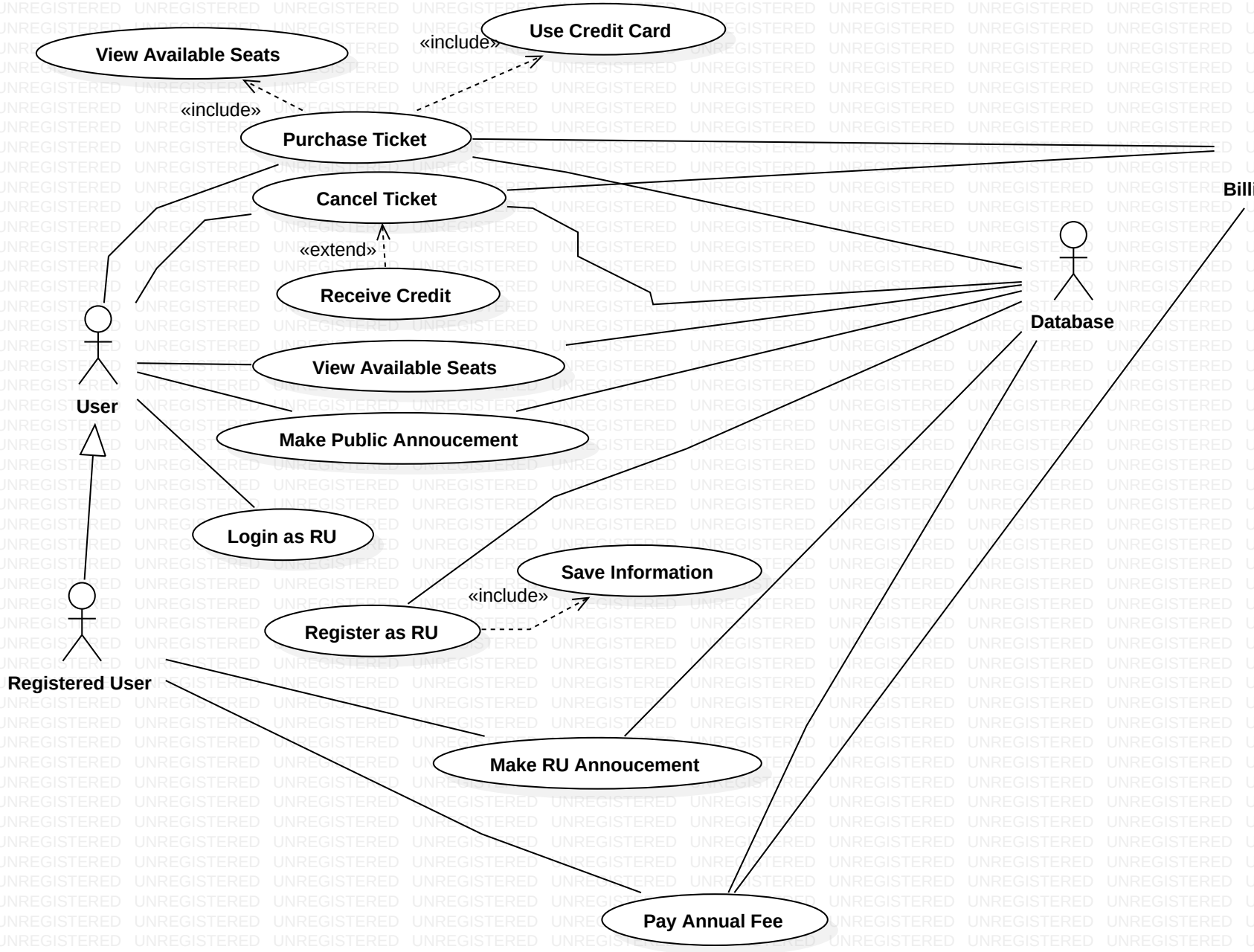
Model1::UseCaseDiagram1



Use Case Scenarios

Reserve Ticket

1. Check if user is registered user
2. Enter movie name
3. System displays available theatres
4. Select Theatre
5. System displays available times
6. Select Time
7. System displays available seats
8. Select Seat
9. Check if user has credit code
10. Enter credit code
11. Deduct credit value from checkout price
12. If credit is used up, delete credit code from database
13. If user is registered, use saved email and card information to process payment
14. Process payment
15. Payment successful
16. System emails user copies of ticket and receipt

Alternative:

2. Enter movie name
3. Movie is unavailable
4. Prompt user to enter another movie name until an available movie is found

Alternative:

10. No credit code so continue to payment

Alternative:

13. If user is not registered, prompt user for email and card information
14. Process payment

Alternative:

15. Payment fails (card declined)

Cancel Ticket

1. Select Ticket to Cancel
2. Check if it is 72 hours prior to the start of the show
3. Check if user is a registered user
4. Email user credit code with a 15% administration fee taken out of the credit if not a registered user

Alternative:

3. If within 72 hours of show, inform user that it's too late to cancel ticket
4. User acknowledges
5. Return to home page

View Available Tickets

1. Click movie button from homepage
2. Displays available movies
3. User selects movie from screen

4. Displays available theatres
5. User selects theatre
6. Displays available times
7. User selects time
8. Display available seats
9.

Make Public Announcement

1. Search for new movies
2. Ensure RU announcement has already occurred
3. Make announcement for all users

Register as RU

1. Enter information (name,address,credit/debit card)
2. Save information into database
3. Add Registration status to user

Make RU Announcement

1. Search for new movies
2. Find Registered Users
3. Make announcement for Registered Users

Pay Annual Fee

1. Check user information
2. Apply $20.00 fee to the account
3. User confirms payment

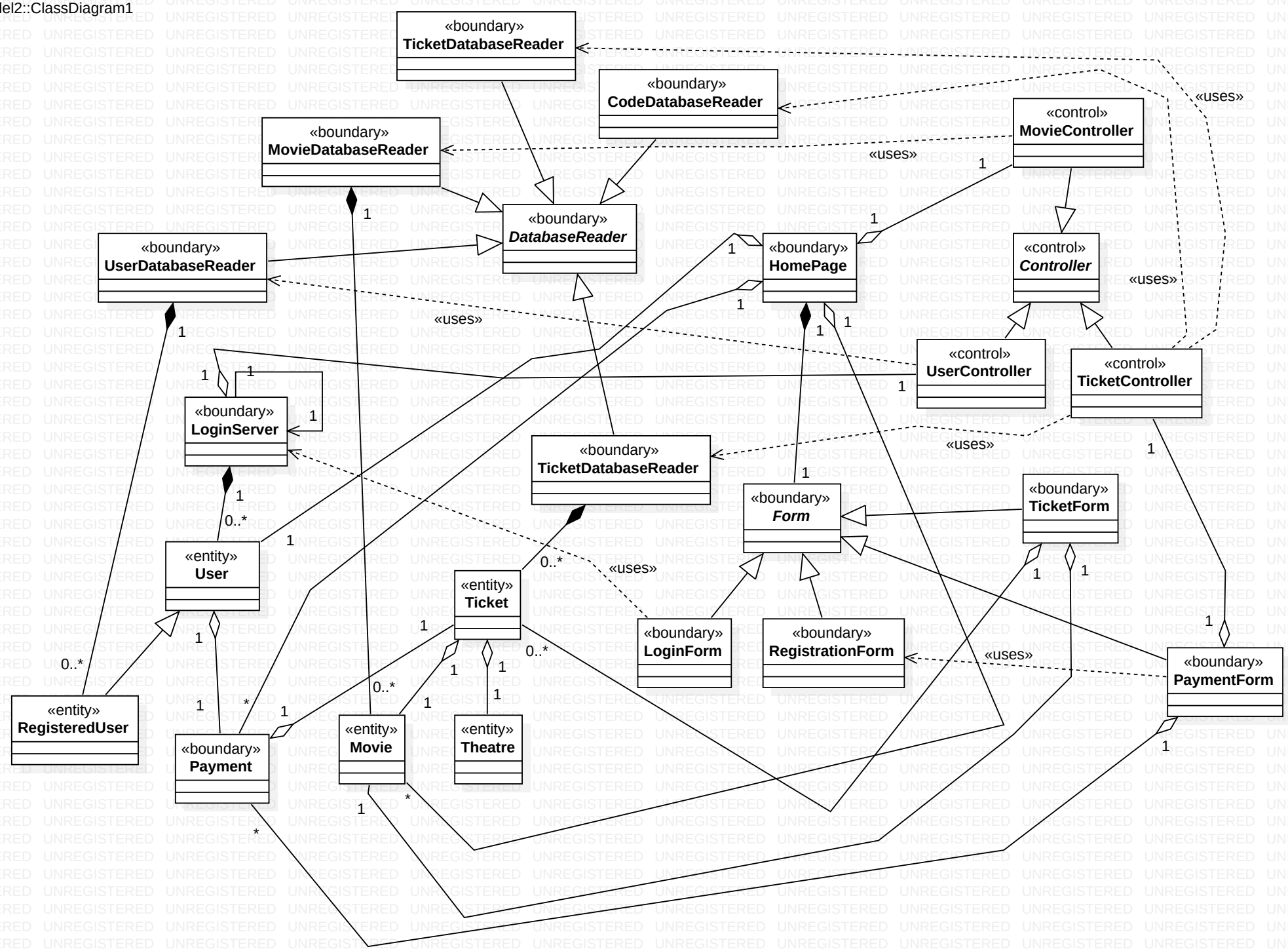
Receive Credit

1. Receive an email with the credit code to be used upon next purchase
2. Store the code with the amount of credit it has in the database

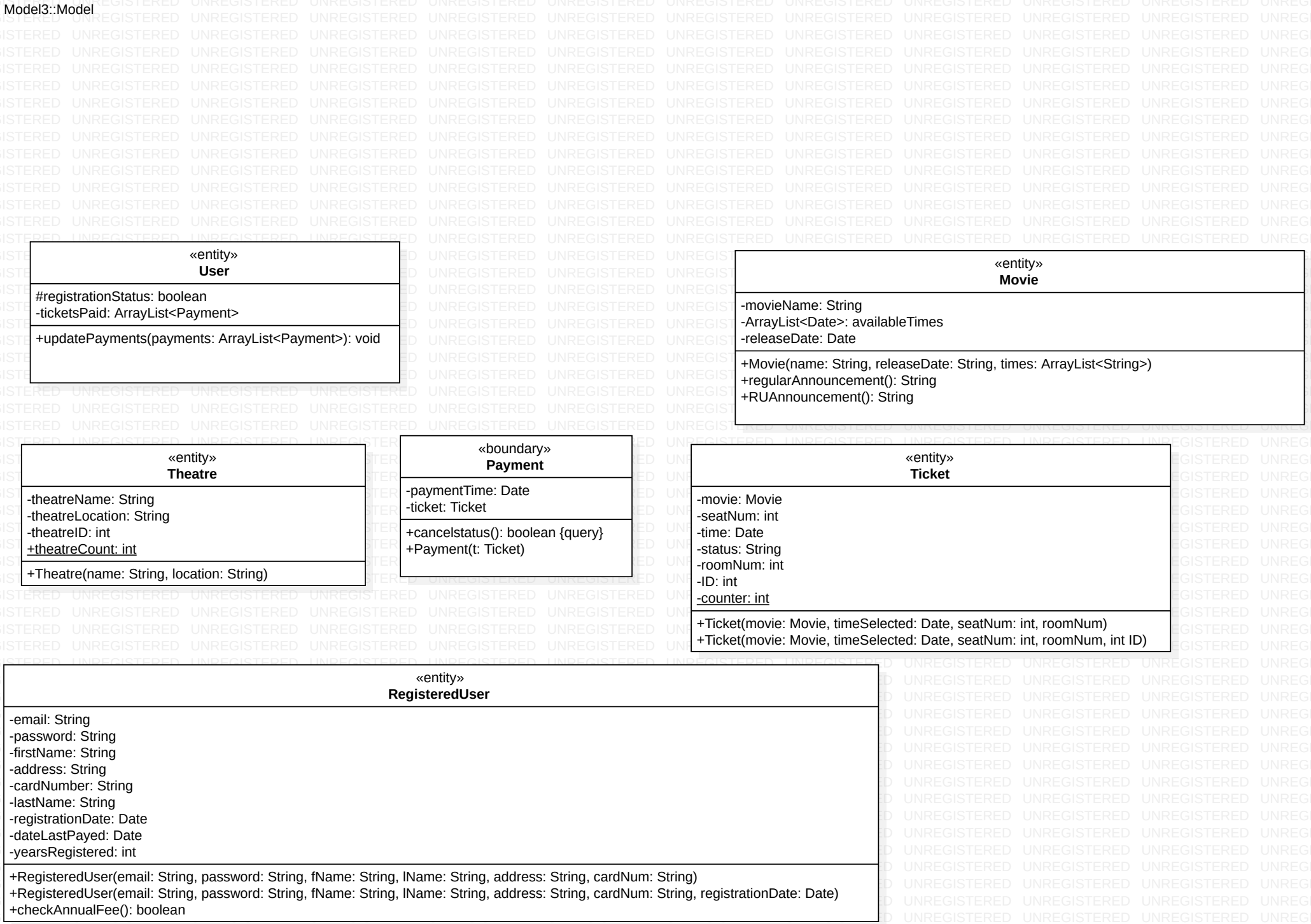
List of Candidate Objects that are traceable in the Use Case Scenarios

Noun	**Filtering Decision**
User	Filtered (actor)
Registered User	Filtered (actor)
Movie	Candidate Object
Theatre	Candidate Object
Time	Filtered (property of Movie)
Seat	Filtered (property of Room)
Ticket	Candidate Object
Payment	Candidate Object
Receipt	Filtered (property of Payment)
Credit Code	Candidate Object
Registration Status	Filtered (property of User)
Room	Filtered (property of Theatre)

Model2::ClassDiagram1



Model3::Model



Model1::Interface

«boundary»
HomePage

-homeFrame: JFrame
-mainPanel: JPanel
-loginButton: JButton
-movieSelect: JButton
-registerButton: JButton
#form: Form
~currentUser: User
~logoutButton: JButton
~annualPayment: JButton
~cancelButton: JButton
~selectButton: JButton
~movies: DefaultListModel<String>
~movieList: ArrayList<Movie>
~paidTickets: DefaultListModel<String>
-paidList: JList<String>
~pListTitle: JPanel
~announcements: JList<String>
-paymentSelected: Payment

+start(): void
+actionPerformed(e: ActionEvent): void
+setForm(Form: f): void

«boundary»
LoginServer

-instance: LoginServer
-users: ArrayList<RegisteredUser>

+getInstance(): LoginServer
+register(email: String, password: String, fName: String, lName: String, address: String, cardNum: String): boolean
+validate(email: String, password: String): RegisteredUser
+remove(ru: RegisteredUser): void

«boundary»
Form

#ticketFormOpened: boolean

+run(): void

«boundary»
LoginForm

-loginFrame: JFrame
-panel: JPanel
-loginButton: JButton
-emailText: JTextField
-passwordText: JPasswordField
-homeButton: JButton

+run(): void
+actionPerformed(e: ActionEvent): void

«boundary»
TicketForm

-ticketFrame: JFrame
-panel: JPanel
-payButton: JButton
-cancelButton: JButton
-homeButton: JButton

+run(): void
+actionPerformed(e: ActionEvent): void

«boundary»
RegistrationForm

-registerFrame: JFrame
-panel: JPanel
-emailText: JTextField
-passwordText: JPasswordField
-fNameText: JTextField
-lNameText: JTextField
-addressText: JTextField
-cardText: JTextField
-registerButton: JButton
-homeButton: JButton

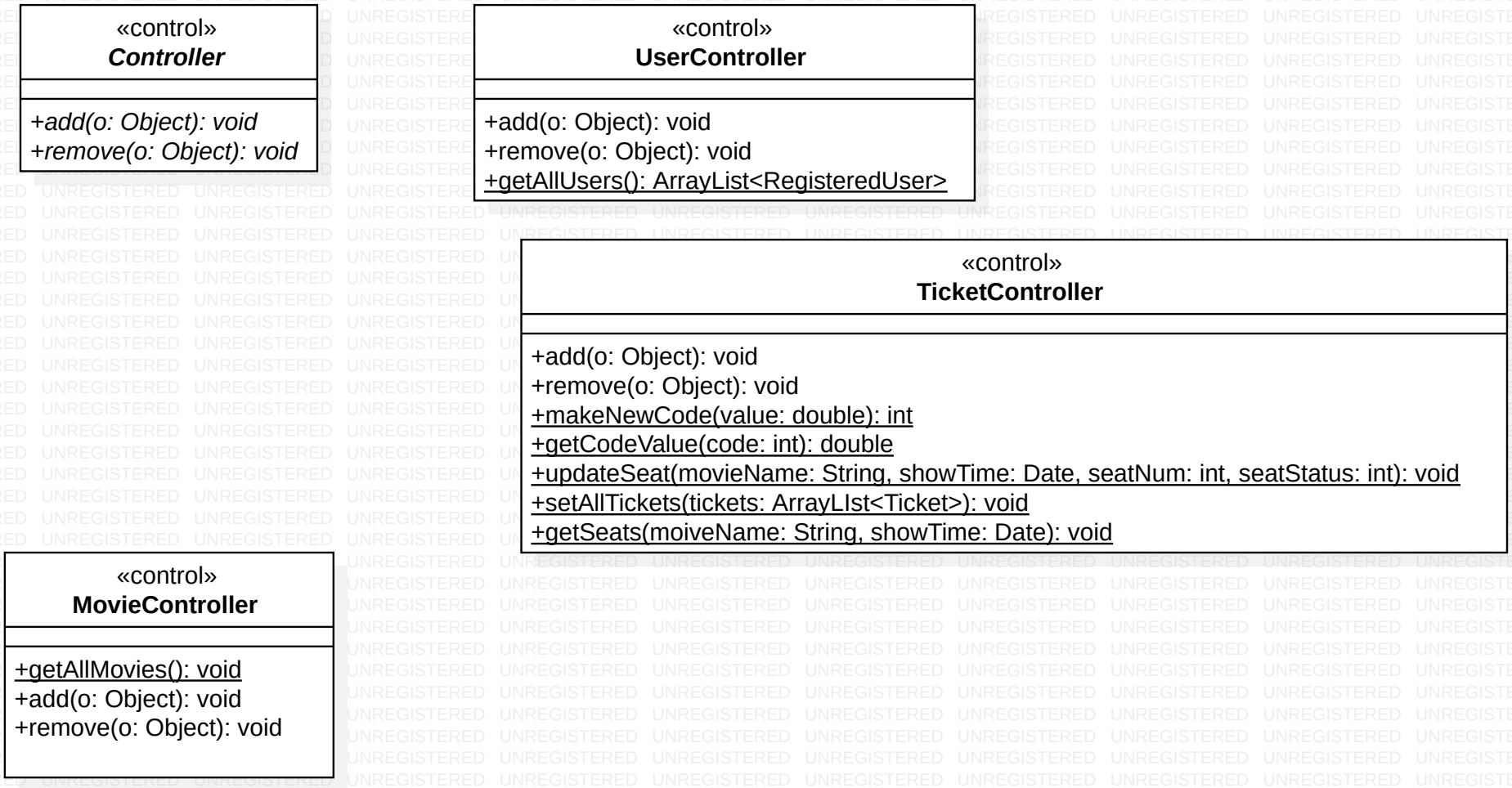
+run(): void
+actionPerformed(e: ActionEvent): void

«boundary»
PaymentForm

-paymentFrame: JFrame
-panel: JPanel
-confirmButton: JButton
-homeButton: JButton
-selectButton: JButton
-summText: JTextField
~tickets: DefaultListModel<String>
-ticketList: JList<String>
~payments: ArrayList<Payment>
~tc: TicketController

+run(): void
+actionPerformed(e: ActionEvent): void

Model2::Controller



Model4::Database

«boundary»
UserDatabaseReader

-allUsers: ArrayList<RegisteredUser>
-TABLE: String {readOnly}

-fetchAllUsers(): ArrayList<RegisteredUser>
+getUser(email: String, password: String): RegisteredUser
+addUser(user: RegisteredUser): boolean
+removeUser(email: String): boolean
+removeUser(user: RegisteredUser): boolean
+updateUser(updatedUser: RegisteredUser): boolean
+getAllUsers(): ArrayList<RegisteredUser>

«interface»
DatabaseReader

+DB_NAME: String {readOnly}
+DB_USERNAME: String {readOnly}
+DB_PASSWORD: String {readOnly}
#connection: Connection

«default»+connect(): Connection
«default»+disconnect(): boolean

«boundary»
CodeDatabaseReader

-TABLE: String {readOnly}

+makeNewCode(value: double): void
+getCodeValue(codeNum: int): double

«boundary»
TicketDatabaseReader

-TABLE: String {readOnly}
-allTickets: ArrayList<Ticket>

+getAllTickets(): ArrayList<Ticket>
-fetchAllTickets(): ArrayLIst<Ticket>
+addTicket(movieName: String, showTime: Date, seatNum: int): boolean
+addTicket(ticket: Ticket): boolean
+removeTicket(movieName: String, showTime: Date, seatNum: int): boolean
+removeTicket(ticket: Ticket): boolean
+getTicket(movieName: String, showTime: Date, seatNum: int): Ticket

«boundary»
MovieDataBaseReader

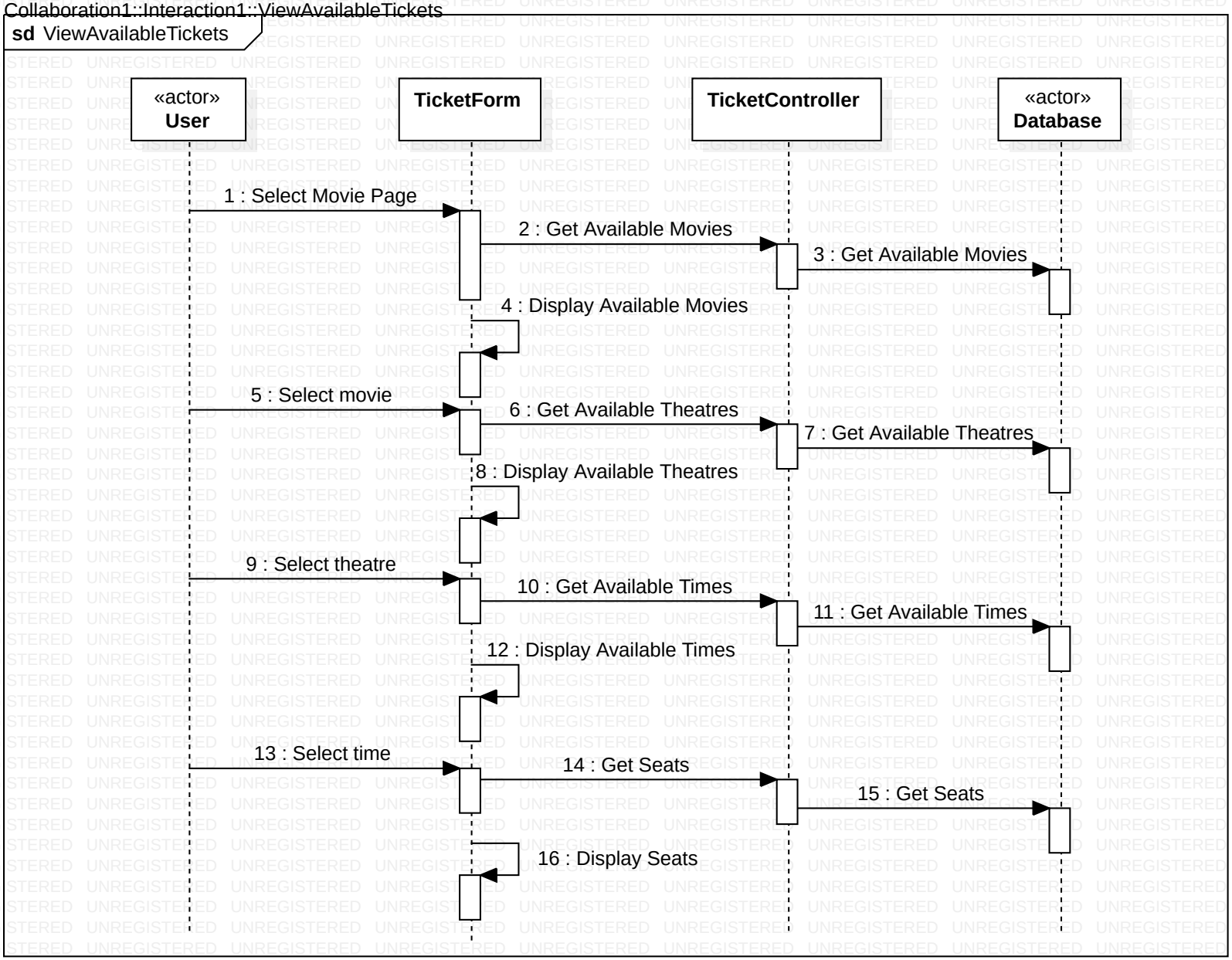
-TABLE: String
-NUM_OF_SEATS: int {readOnly}
-allMovies: ArrayList<Movie>

+getAllMovies(): ArrayList<Movie>
+setAllMovies(movies: ArrayList<Movie>): void
+setTable(theatreName: String): void
+getAllShowTimes(movieName: String): ArrayList<Date>
+getSeats(movieName: String, showTime: Date): ArrayList<Integer>
+getMovie(movieName: String): Movie
-fetchAllMovies(): ArrayList<Movie>
+updateSeat(movieName: String, showTime: Date, seatNum: int, seatStatus: int): boolean
+removeMovie(movieName: String, showTime: Date): boolean
+addMovie(movieName: String, showTime: Date, releaseDate: Date): boolean

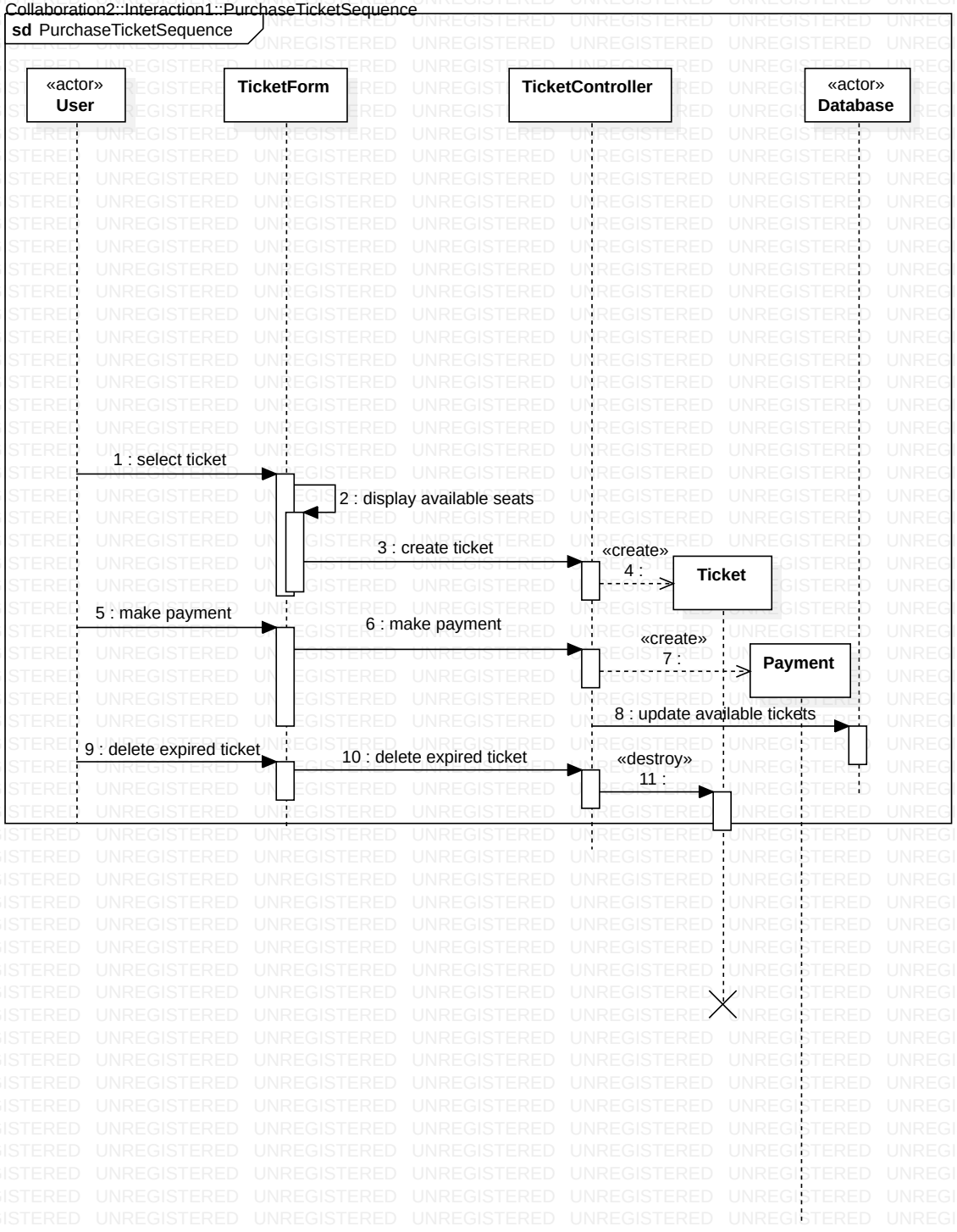
Collaboration1::Interaction1::ViewAvailableTickets

sd ViewAvailableTickets

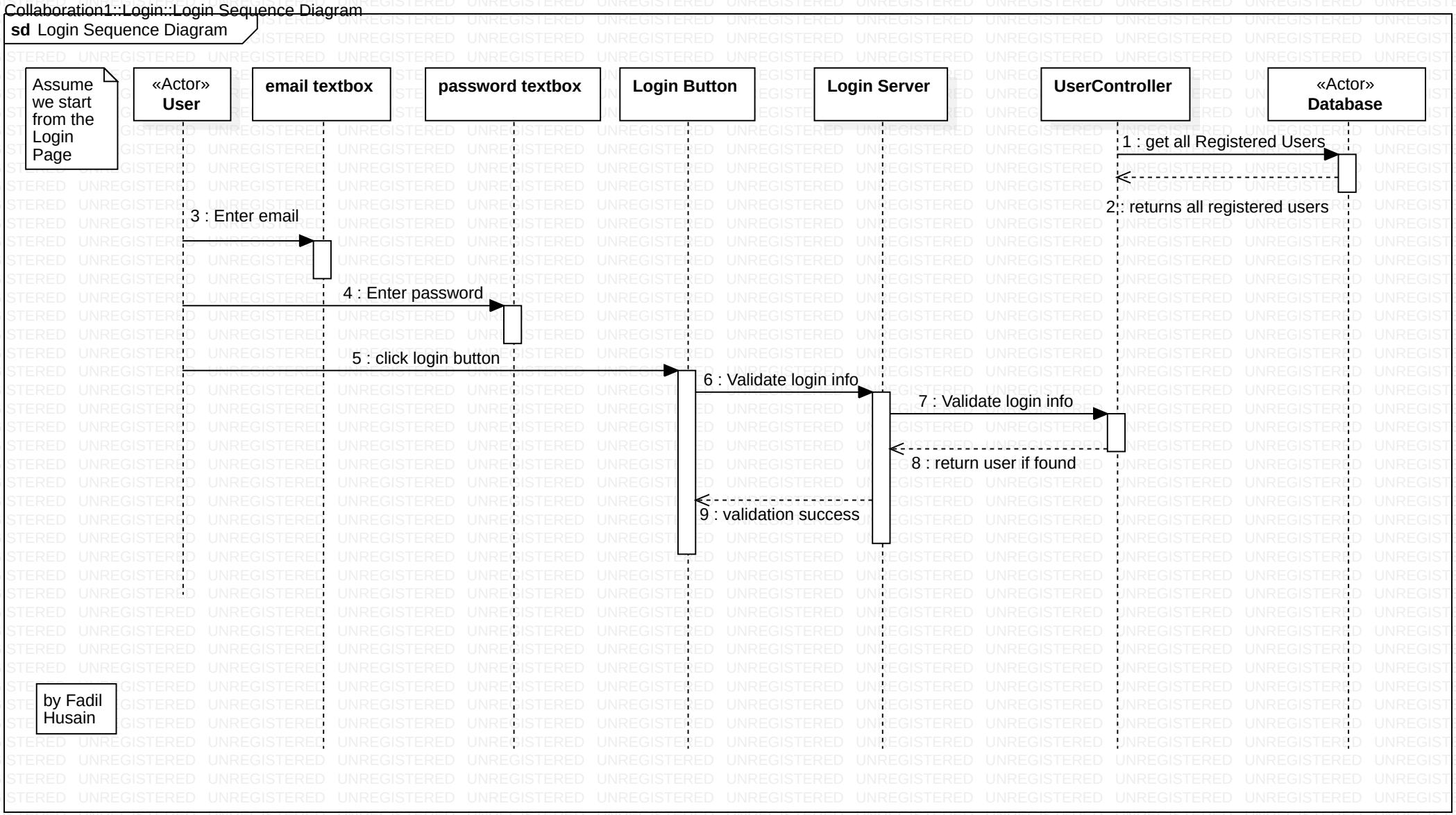
Jaron



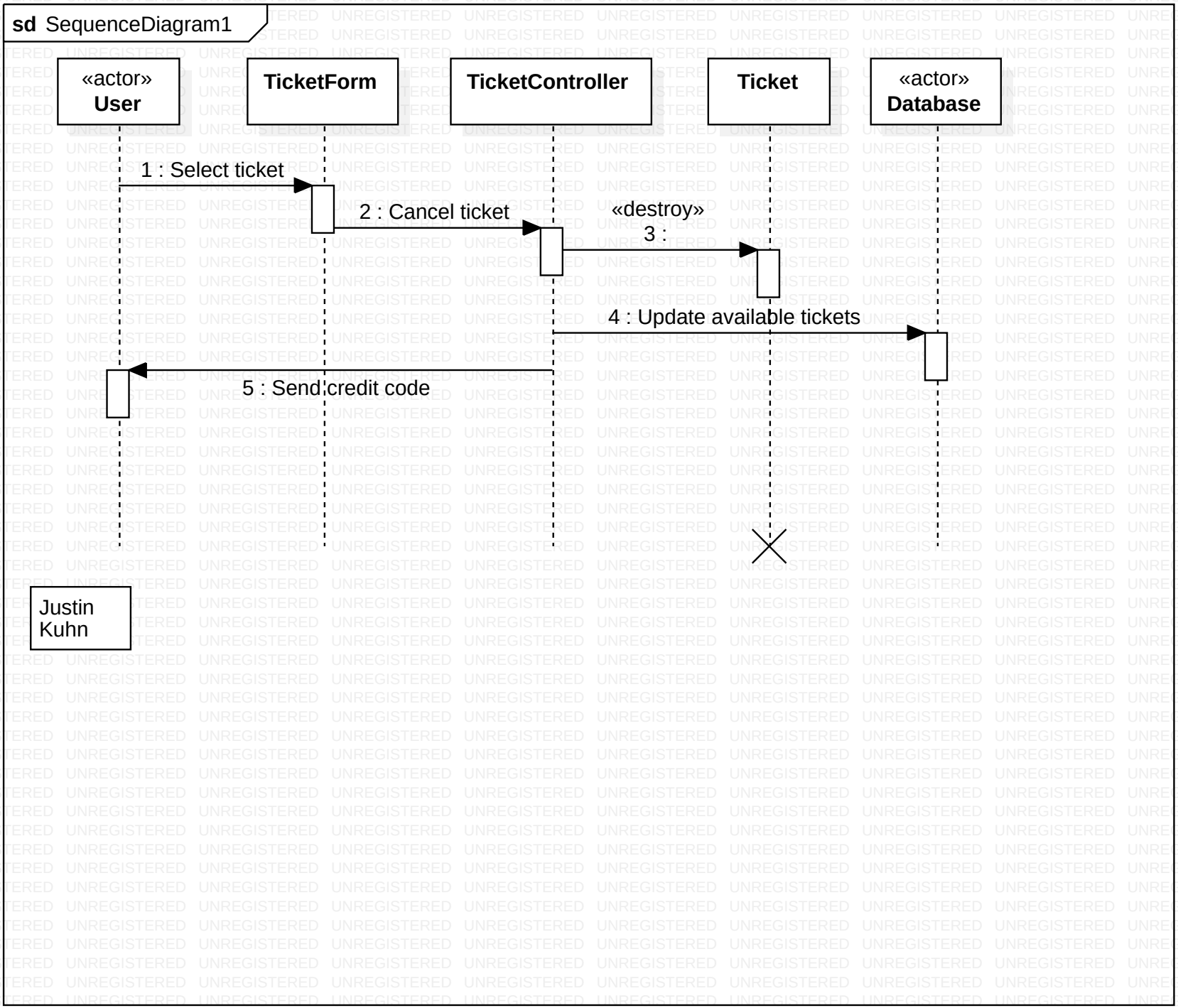
Collaboration2::Interaction1::PurchaseTicketSequence



Eric

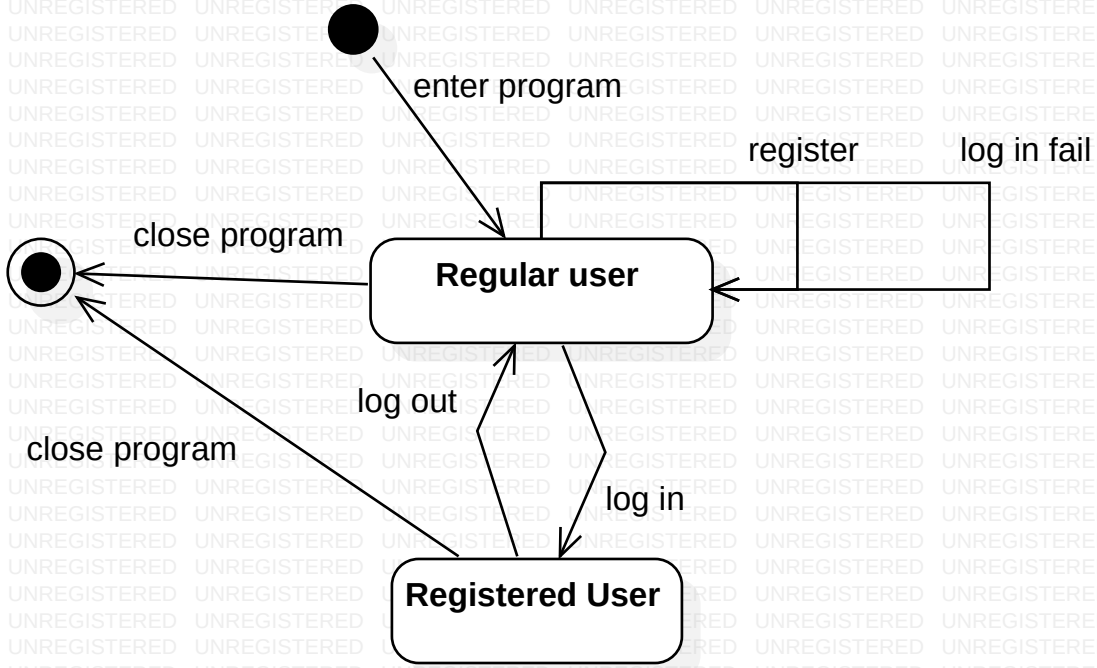

Collaboration1::Login::Login Sequence Diagram
sd Login Sequence Diagram
Assume we start from the Login Page
«Actor» User
email textbox
password textbox
Login Button
Login Server
UserController
«Actor» Database
1 : get all Registered Users
2 : returns all registered users
3 : Enter email
4 : Enter password
5 : click login button
6 : Validate login info
7 : Validate login info
8 : return user if found
9 : validation success
by Fadil Husain

Model::Collaboration1::Interaction1::SequenceDiagram1



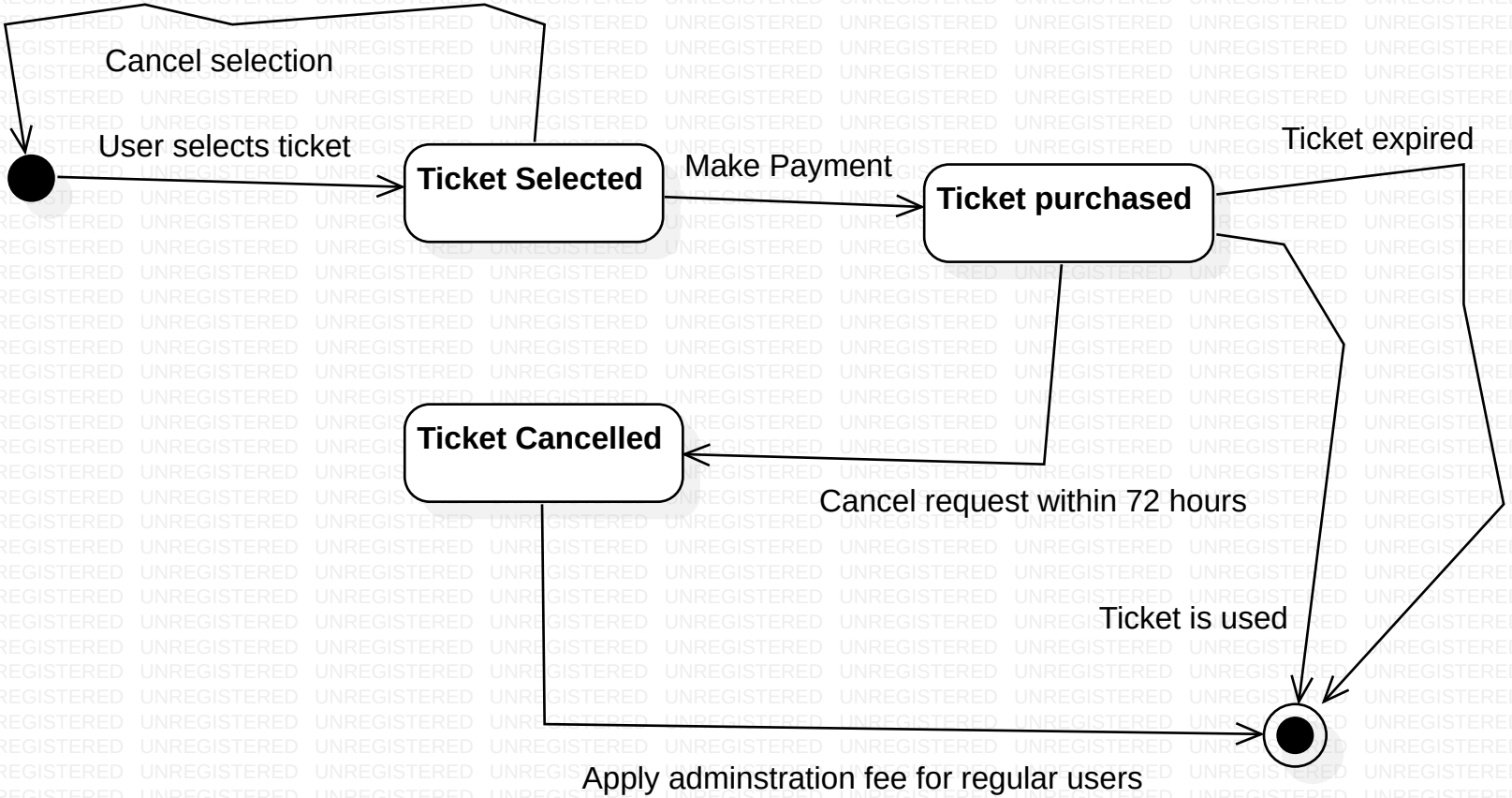
user/registedUser::register as RU

By Fadil Husain

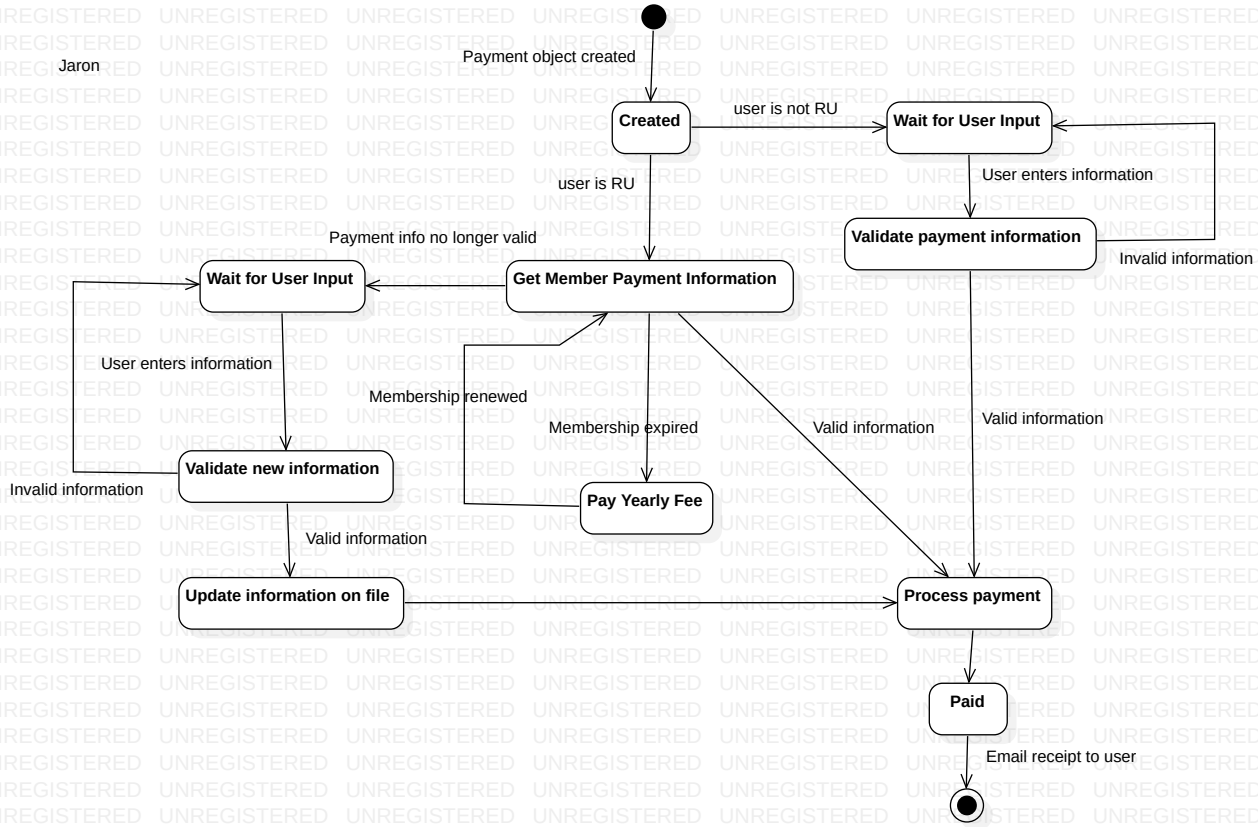


StateMachine1::TicketState

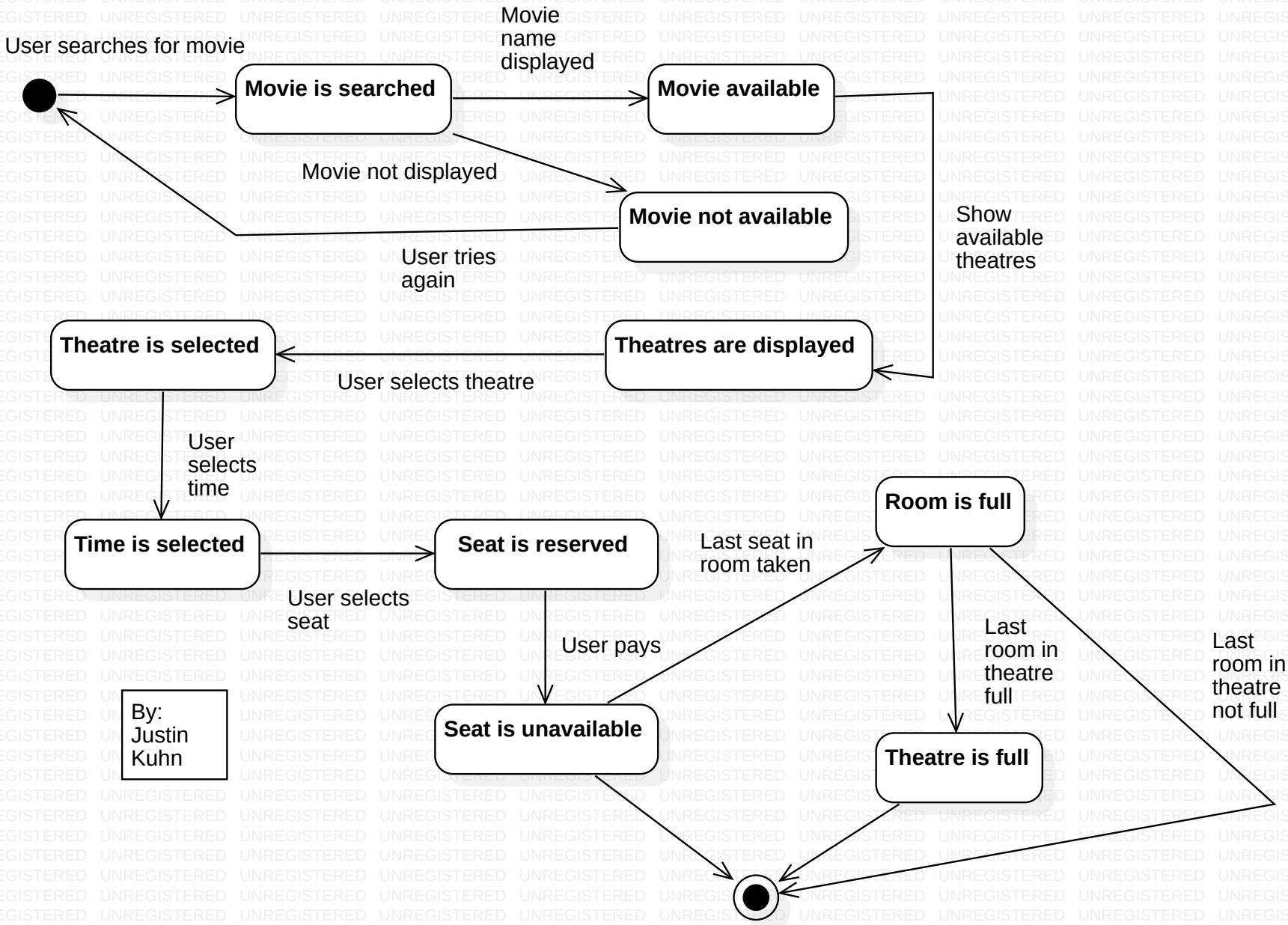
Eric
Wong

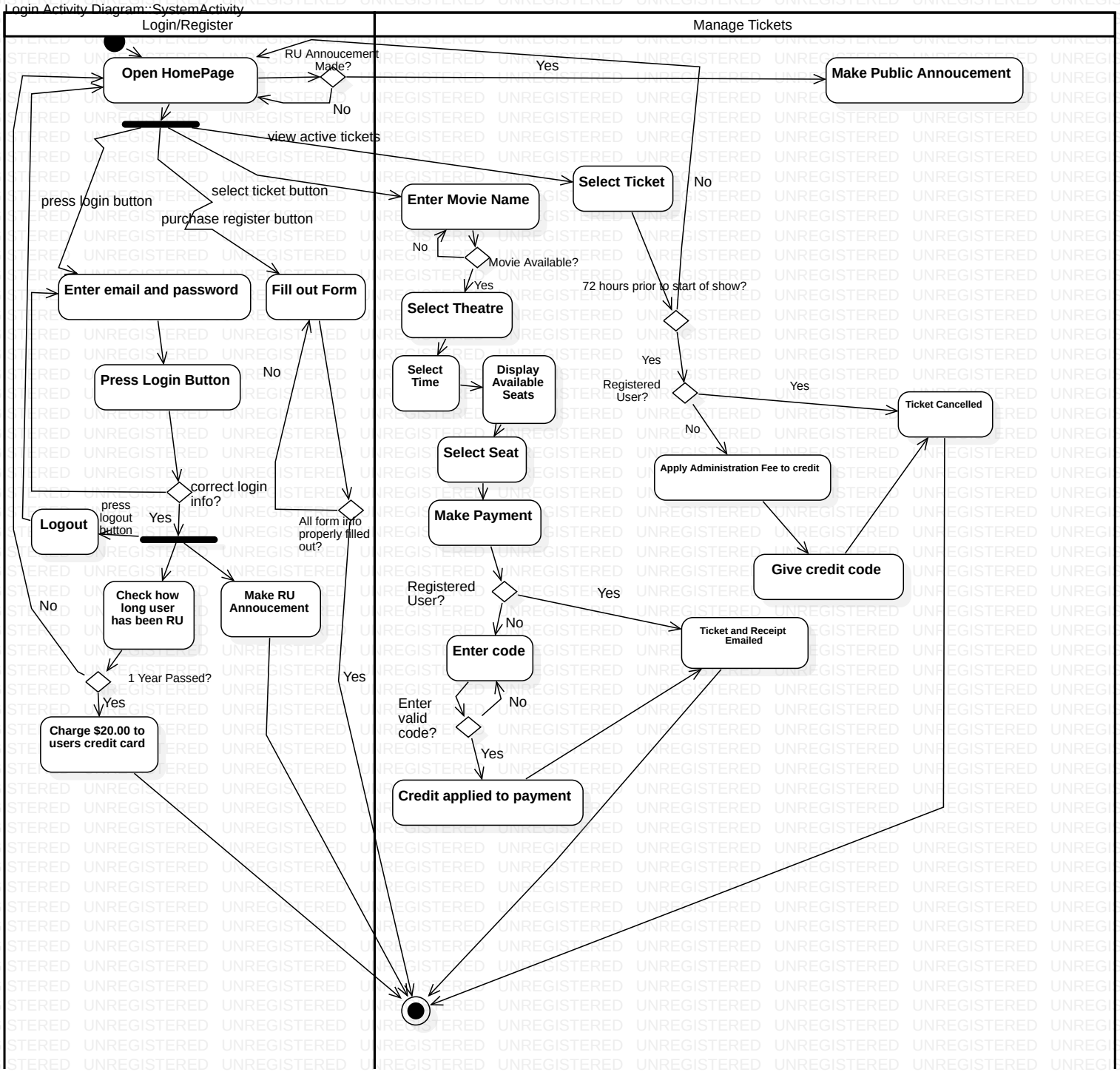


StateMachine1::StatechartDiagram1

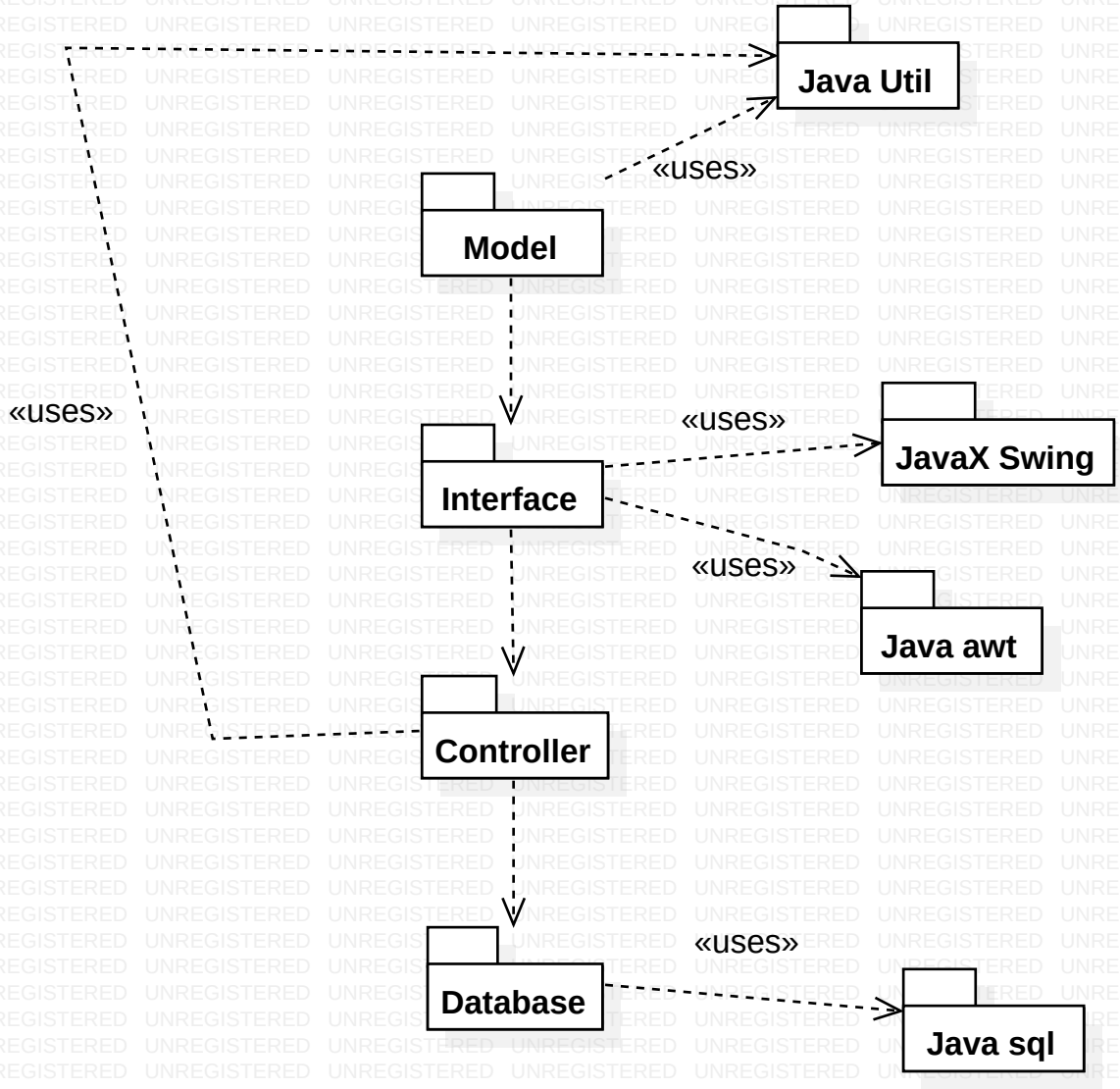


Model::StateMachine1::StatechartDiagram1




Login Activity Diagram::SystemActivity
Login/Register
Manage Tickets
Open HomePage
RU Annoucement Made?
Yes
No
Make Public Annoucement
view active tickets
press login button
select ticket button
purchase register button
Enter email and password
Fill out Form
Press Login Button
No
correct login info?
Logout
press logout button
Yes
All form info properly filled out?
Check how long user has been RU
Make RU Annoucement
No
1 Year Passed?
Yes
Charge $20.00 to users credit card
Yes
Enter Movie Name
No
Movie Available?
Yes
Select Theatre
Select Time
Display Available Seats
Select Seat
Make Payment
Registered User?
Yes
No
Enter code
Enter valid code?
No
Yes
Credit applied to payment
Ticket and Receipt Emailed
Select Ticket
No
72 hours prior to start of show?
Yes
Registered User?
Yes
Ticket Cancelled
No
Apply Administration Fee to credit
Give credit code

Model::PackageDiagram1



Model::DeploymentDiagram1