

# Software Engineering Program

ENSF614–Fall 2021

## Term Project: Design Phase Document

### **Group Members:**

Zach Frena

Ahmad Elkholy

Kelten Falez

Evan Cooksley

## **Table Of Contents**

<b>System requirements:</b>	<b>3</b>
<b>System Use Case Diagram</b>	<b>4</b>
<b>System Activity Diagram</b>	<b>5</b>
<b>State Transition Diagram</b>	<b>6</b>
<b>Scenarios</b>	<b>7</b>
Unregister Scenario:	7
Register Scenario:	7
Buy Ticket Scenario:	7
Cancel Ticket Scenario:	7
<b>System Interaction Diagrams</b>	<b>8</b>
<b>Design Level Class Specification</b>	<b>12</b>
<b>Class Diagram - No Relationships</b>	<b>13</b>
Model	13
View	14
Controller	15
Package Diagram	16
<b>Deployment Diagram</b>	<b>17</b>

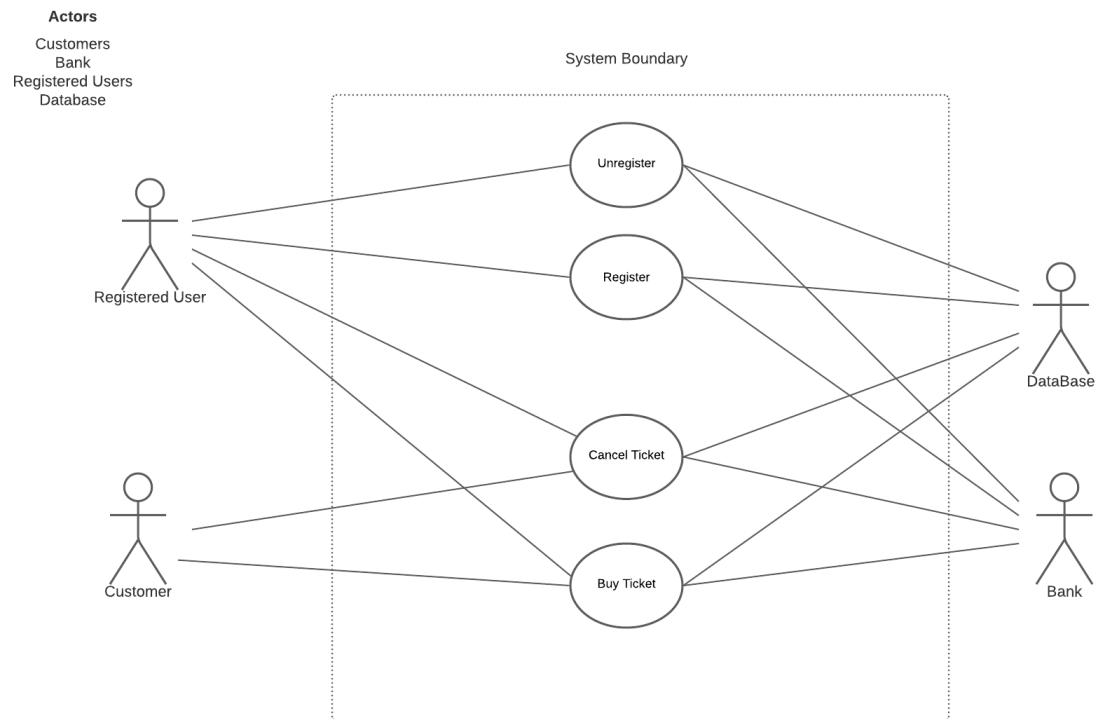
# System requirements:

Our task is to analyze, design, and develop a system that can be used by two group of users:

1. Ordinary users that can search for a movie, select an specific theater, view available show times, view graphically available seats, select the desired seat, make payment by credit card, receive a copy of ticket and the receipt, via email. Users also should be able to cancel their ticket only up to 72 hours prior to show and receive a credit with %15 administration fee for future purchase up to a maximum of one-year expiration date.
2. Registered Users (RUs) whose information such as name, address, credit and/or debit card account must be saved on the system's database. RUs must pay a \$20.00 annual account fee, but they don't have to pay a 15% admin fee for cancelling their tickets and will receive the movie news before public announcement. There is also one more constraint: Only 10% of the seats can be purchased by RUs on a first come first serve policy prior to public announcements.

The implementation of this system/application will be limited to a single theatre development.

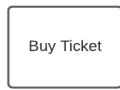
# System Use Case Diagram



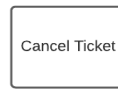
# System Activity Diagram



Enter Details  
Register (Check if  
already registered)  
Save details to db  
Charge annual fee  
Update permissions



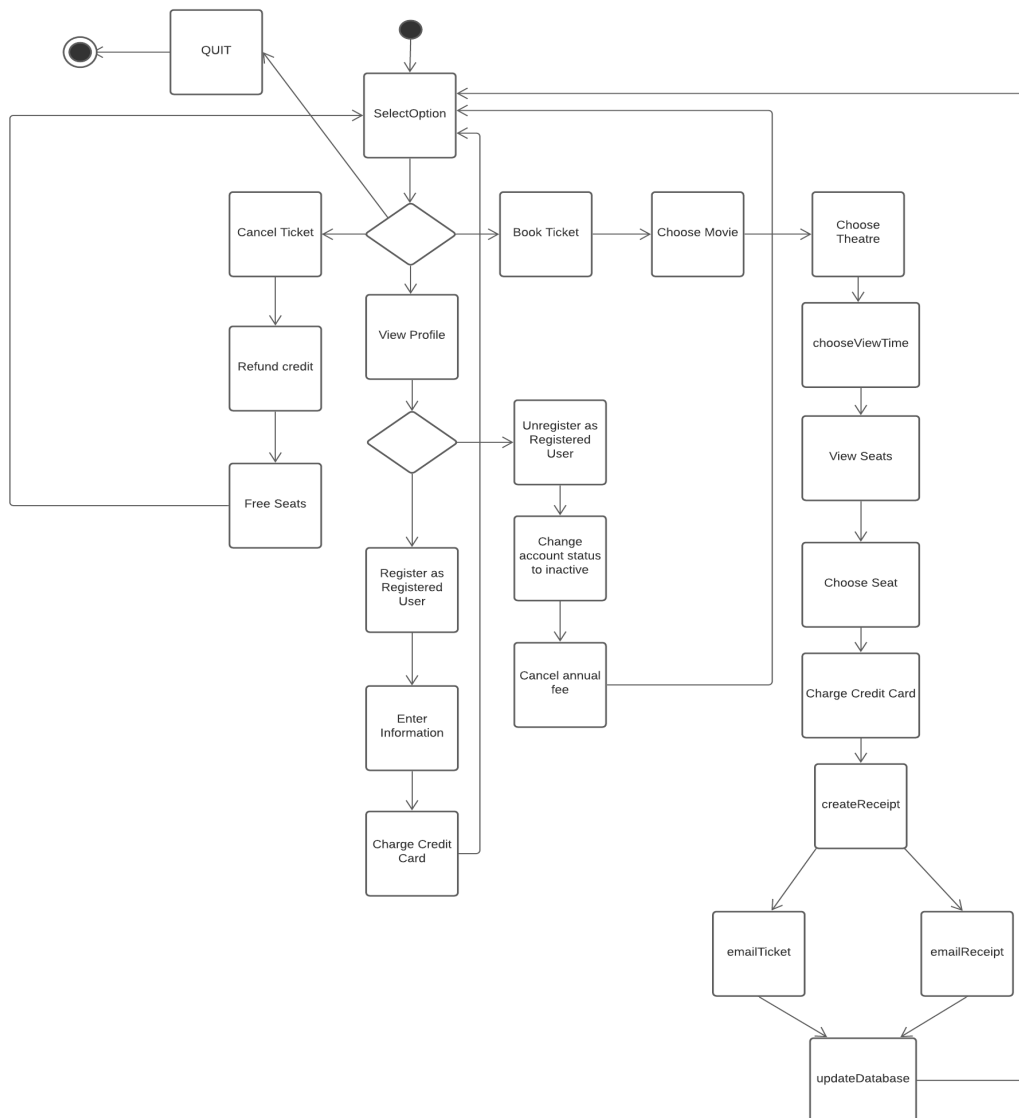
Search for movie  
Select Theatre  
Select Showtime  
View Seats  
Select Seats  
Make Payment  
Receive Ticket and  
Receipt



Choose ticket to cancel  
from active tickets  
Check if registered  
user  
Refund credit (charge  
admin fee if not RU)  
Delete reservation

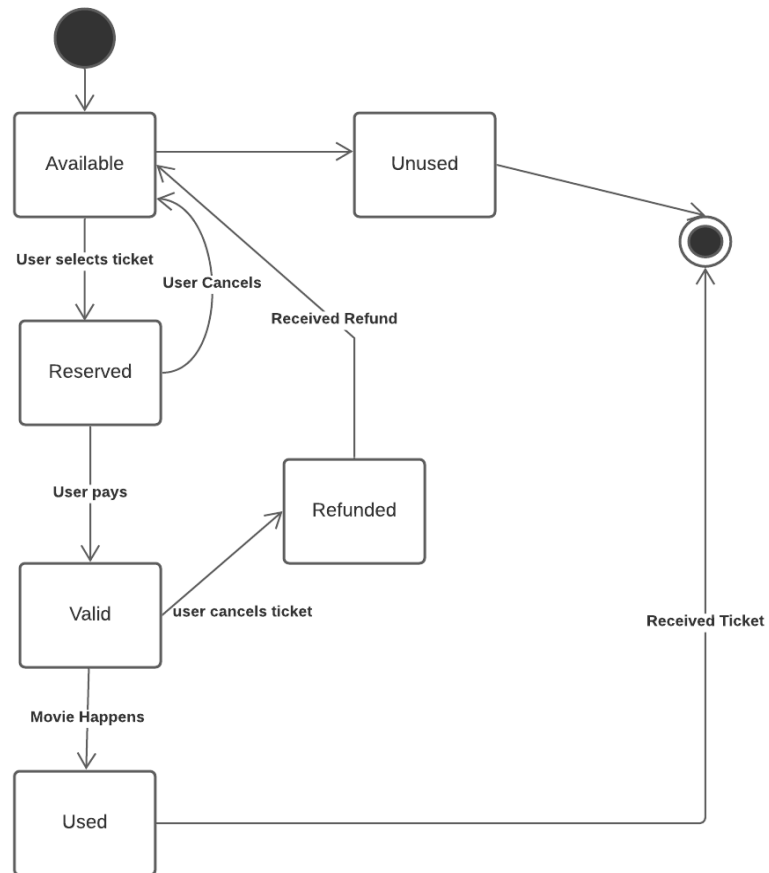


Log in  
UnRegister  
Save details to db  
Update permissions

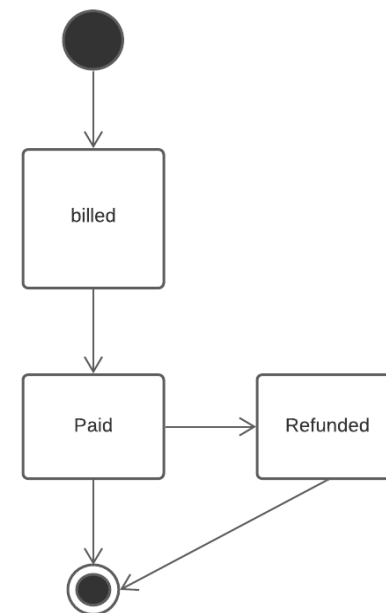


# State Transition Diagram

**Ticket Object**



**Payment Object**



# Scenarios

\*Assume that the names (Zach, Kelten, etc.) represent users of our system.

\*\*We used single-underline for objects and **BOLDED TEXT** for operations.

## Unregister Scenario:

Kelten no longer wants to be a registered user in the system. Kelten selects **unregister** on his profile. System **confirms** with Kelten that he no longer wants to be a registered user. Kelten **confirms**. System **sets Kelten's status to inactive and notifies the billing system** to no longer charge his account for annual fees.

## Register Scenario:

Zach wants to **register** in the system. Once this option is selected, the system prompts Zach to **enter** their personal information (email, billing address, payment details and name). Once valid information is entered, Zach will be **registered** into the system. His account will be **charged** its annual dues immediately and then again once every year. Zach can now **book** movies before they are **released** to the public and get full credit back when they **refund** a ticket.

## Buy Ticket Scenario:

Evan wants to **buy** a ticket in the theatre registration app. Evan will be able to **view** the available movies. If Evan is a Registered User he can **view and book** available movies before the general public, as long as less than 10% of the tickets are **sold** before general release. After Evan **selects** the movie, he will **select** the showtime, view his seat via a GUI, **select** his seat, and make the **payment** through his credit card. Once completed, he will be **sent** a copy of the ticket and the receipt in his email.

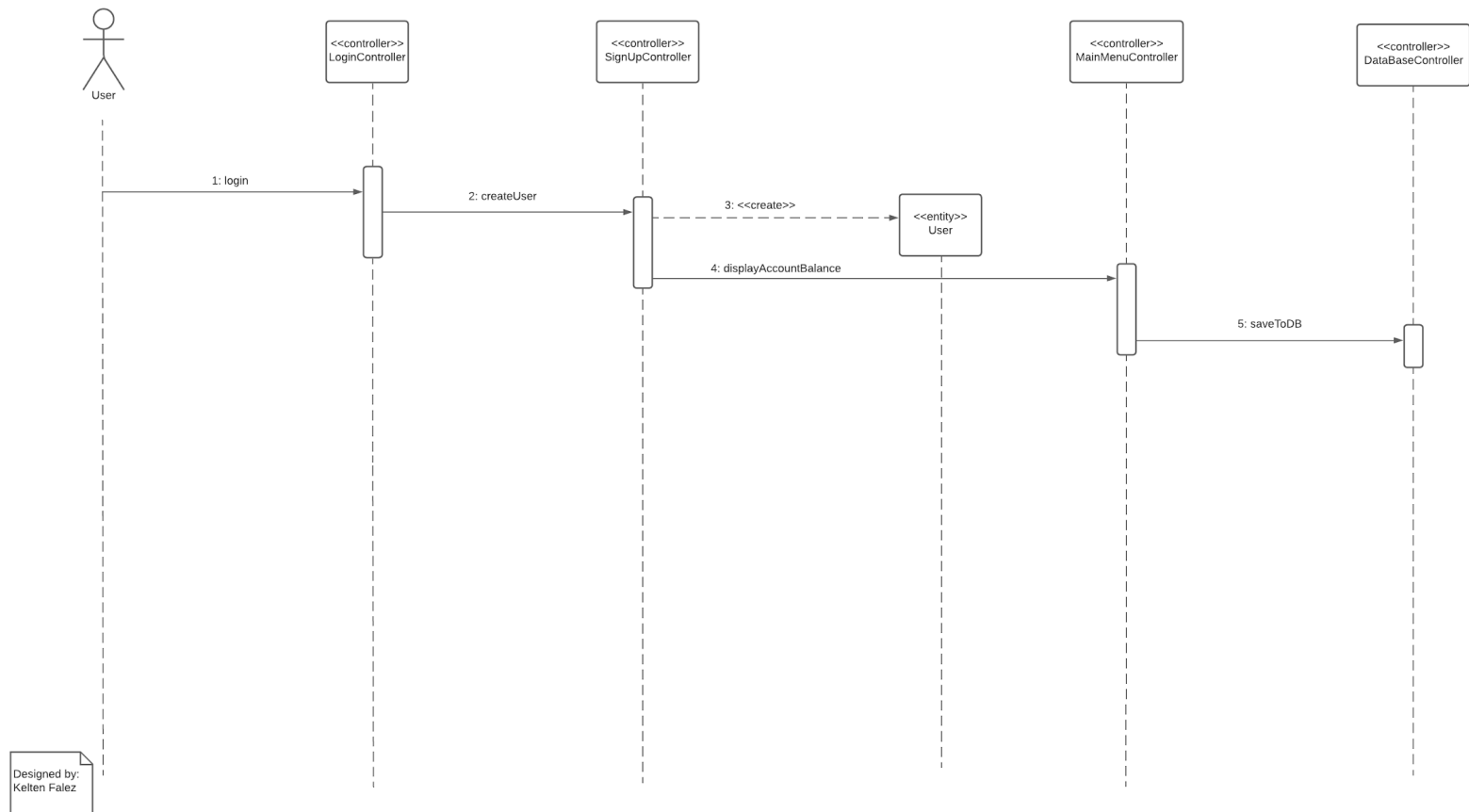
## Cancel Ticket Scenario:

Ahmad wants to **cancel** a ticket from the app. Ahmad will be able to **view** the ticket(s) he **bought**. He then selects which ticket(s) to be **canceled**. Ahmad will be **refunded** a credit for the ticket he bought. If Ahmad is a RU he will **receive** the full value of the ticket. Otherwise he will be **charged** a 15% admin fee for cancellation.

After the ticket has been **cancelled**, the chosen seats will be **released** and become available for others to choose/**reserve**.

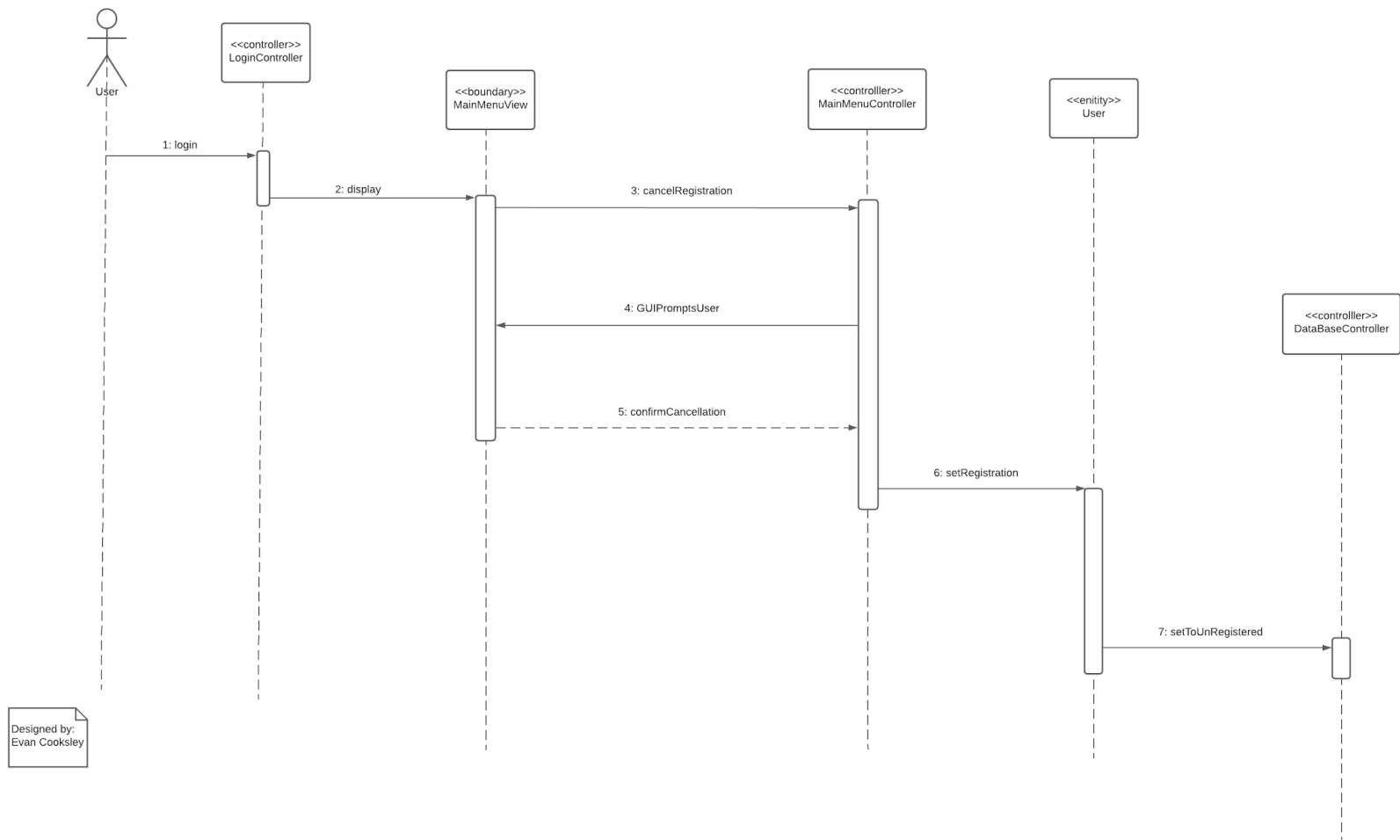
# System Interaction Diagrams

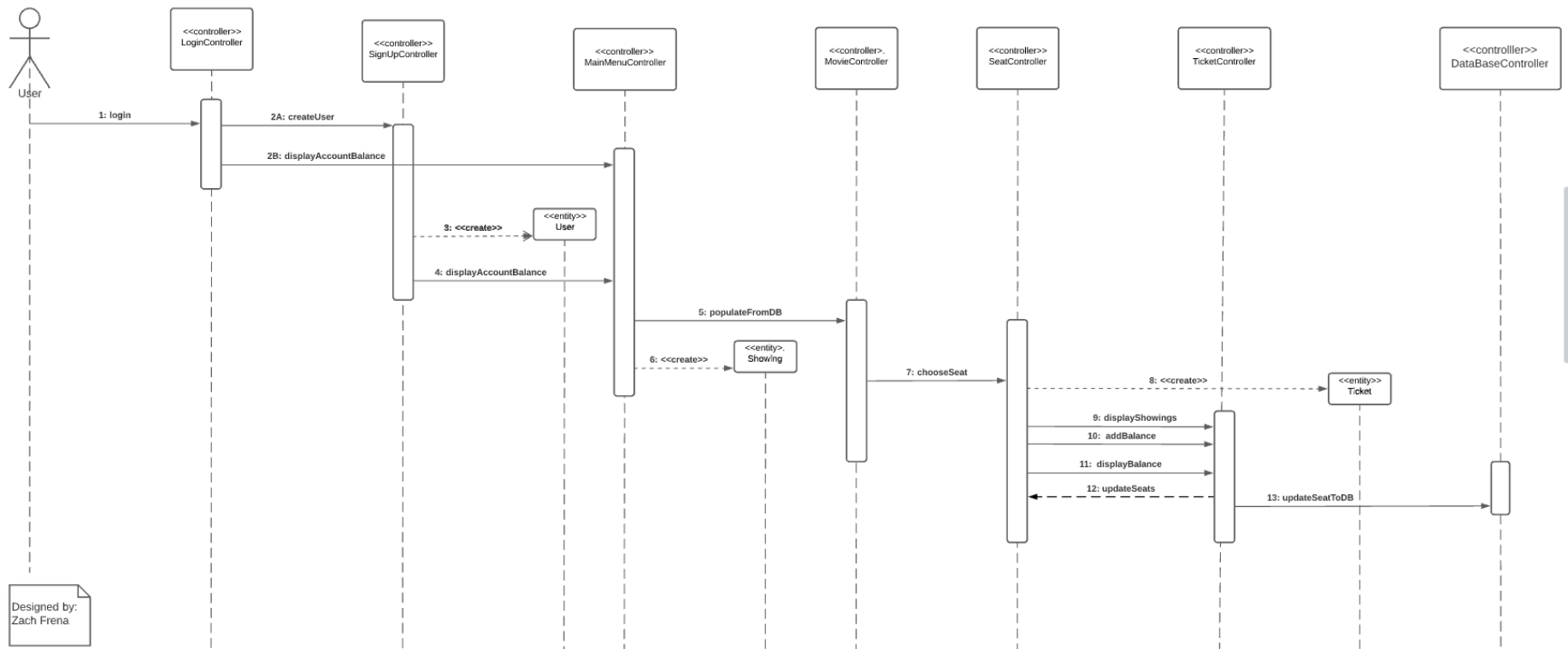
## Use Case: Register



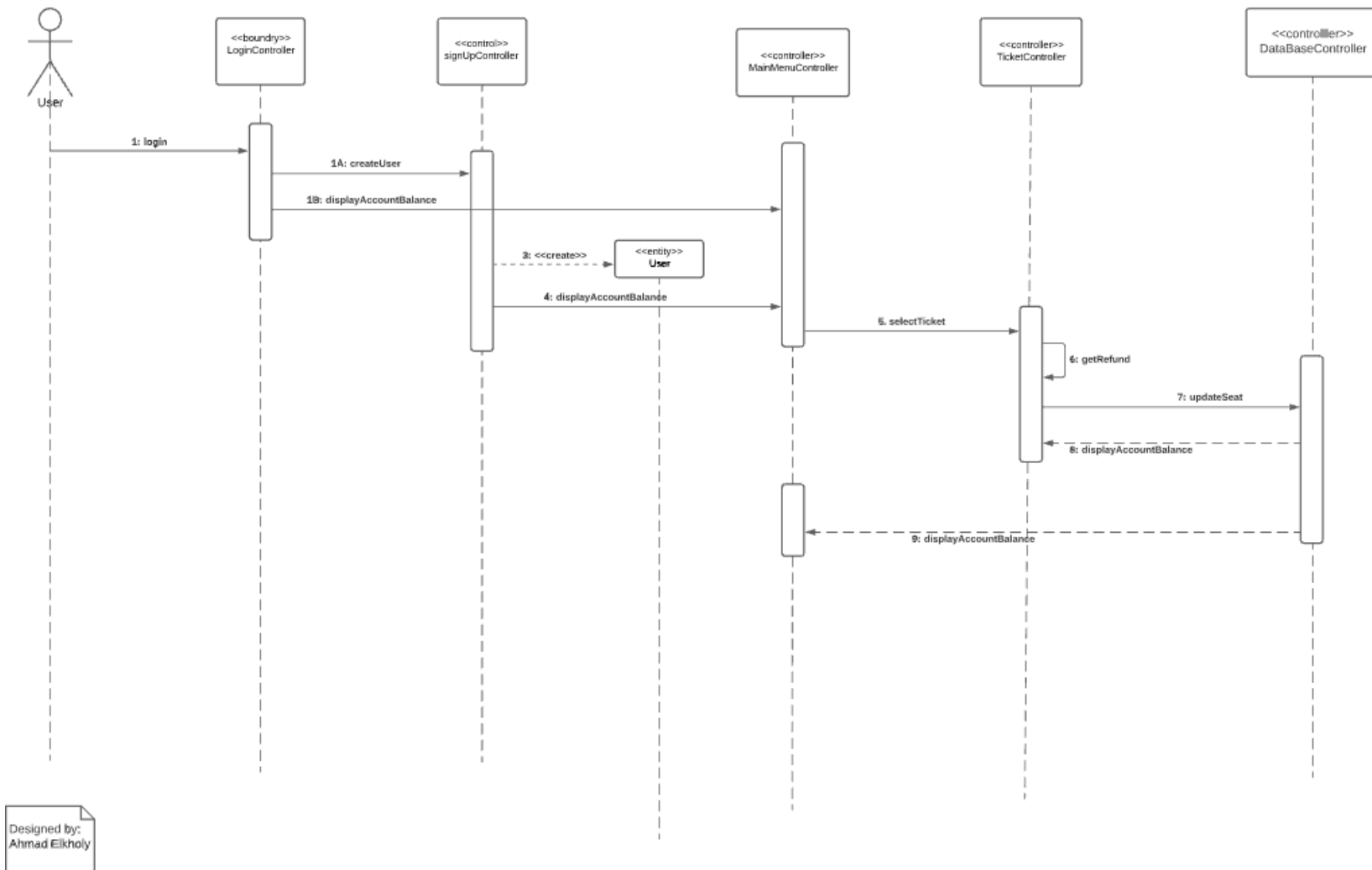


## Use Case: Unregister

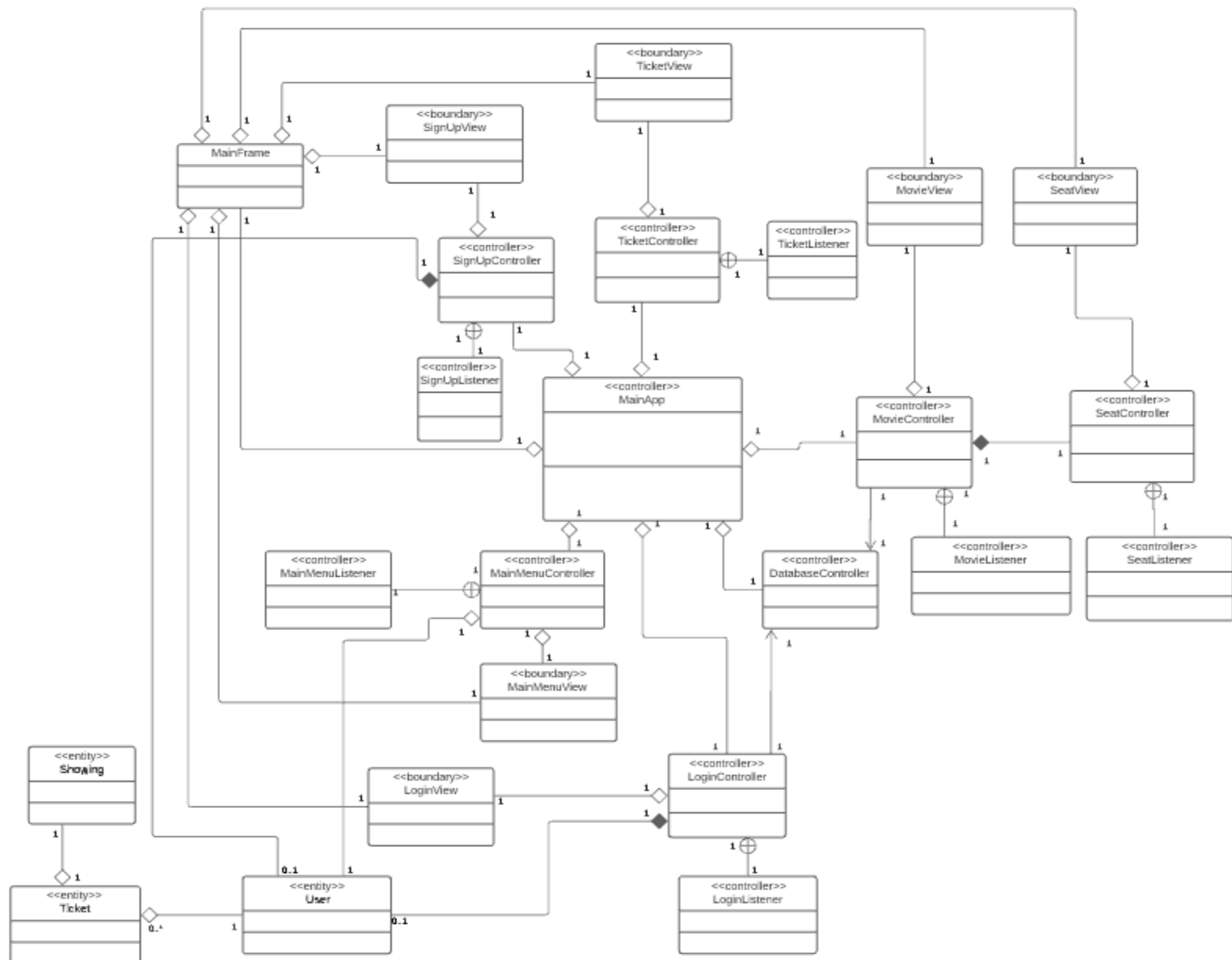




## Use Case: Cancel Ticket

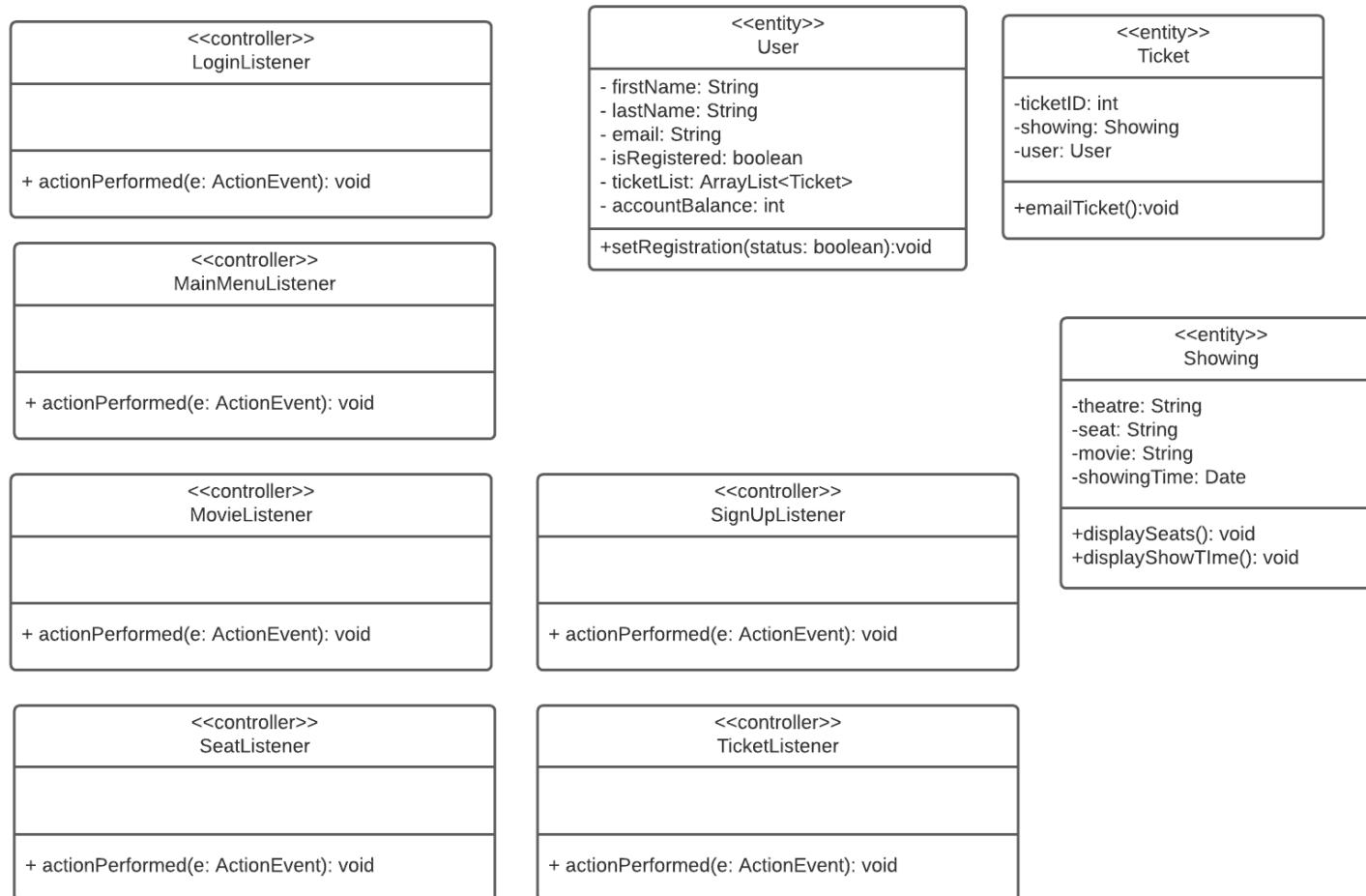


# Design Level Class Specification

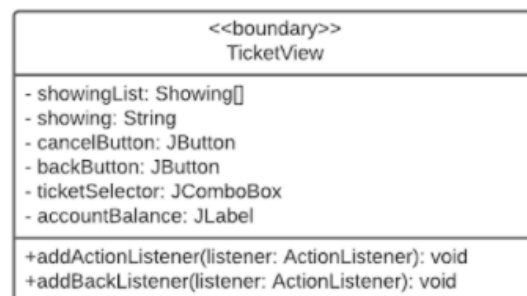
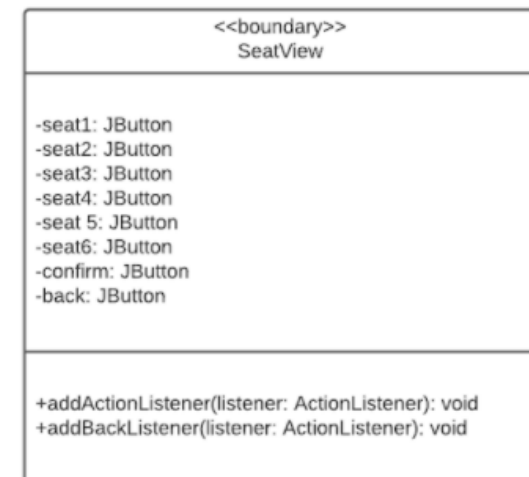
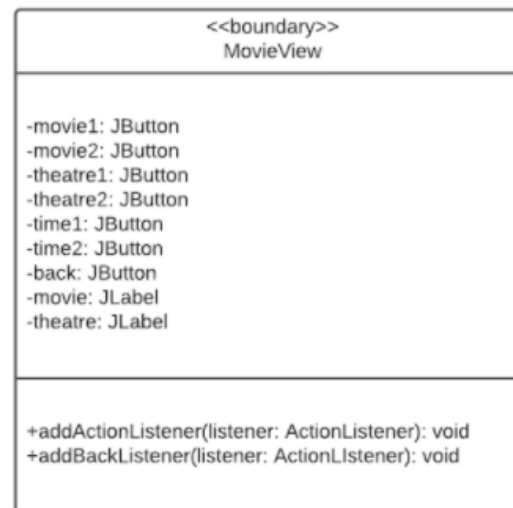
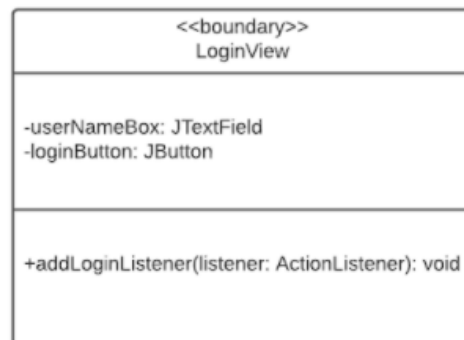
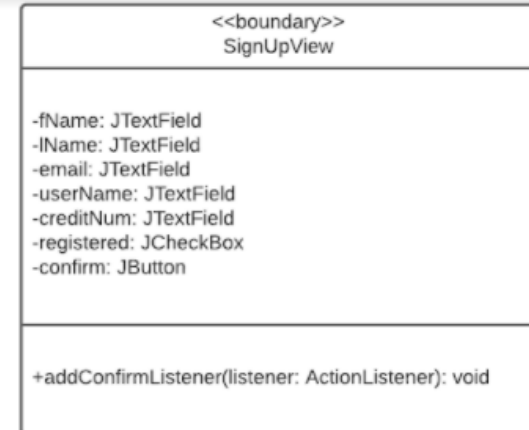
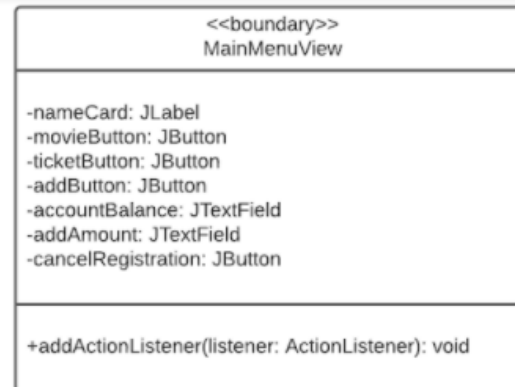
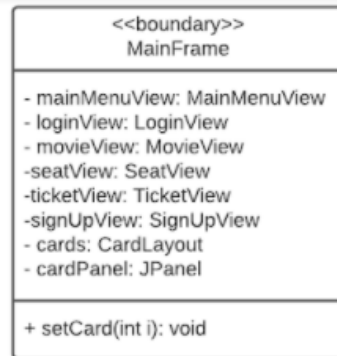


# Class Diagram - No Relationships

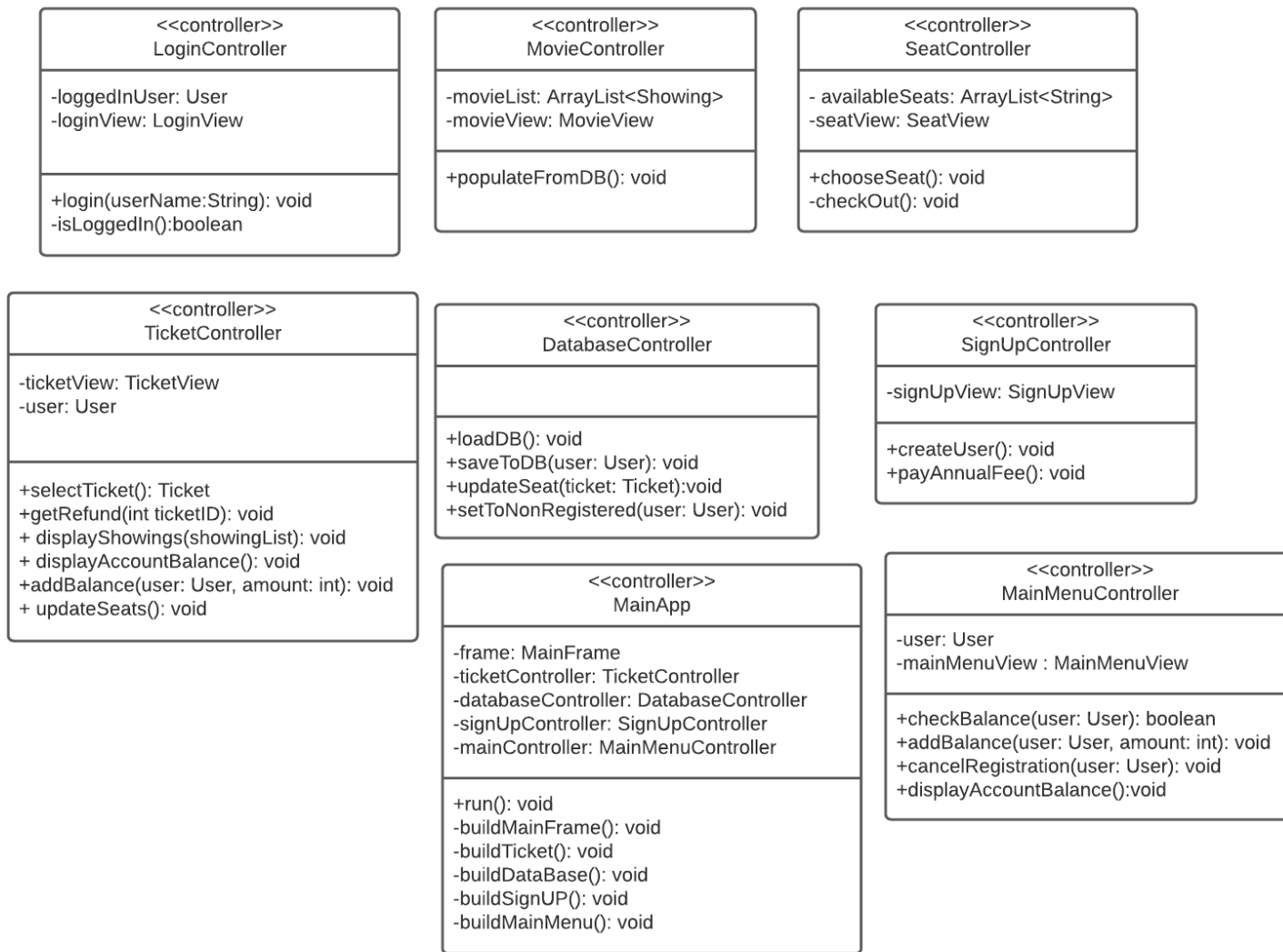
## Model



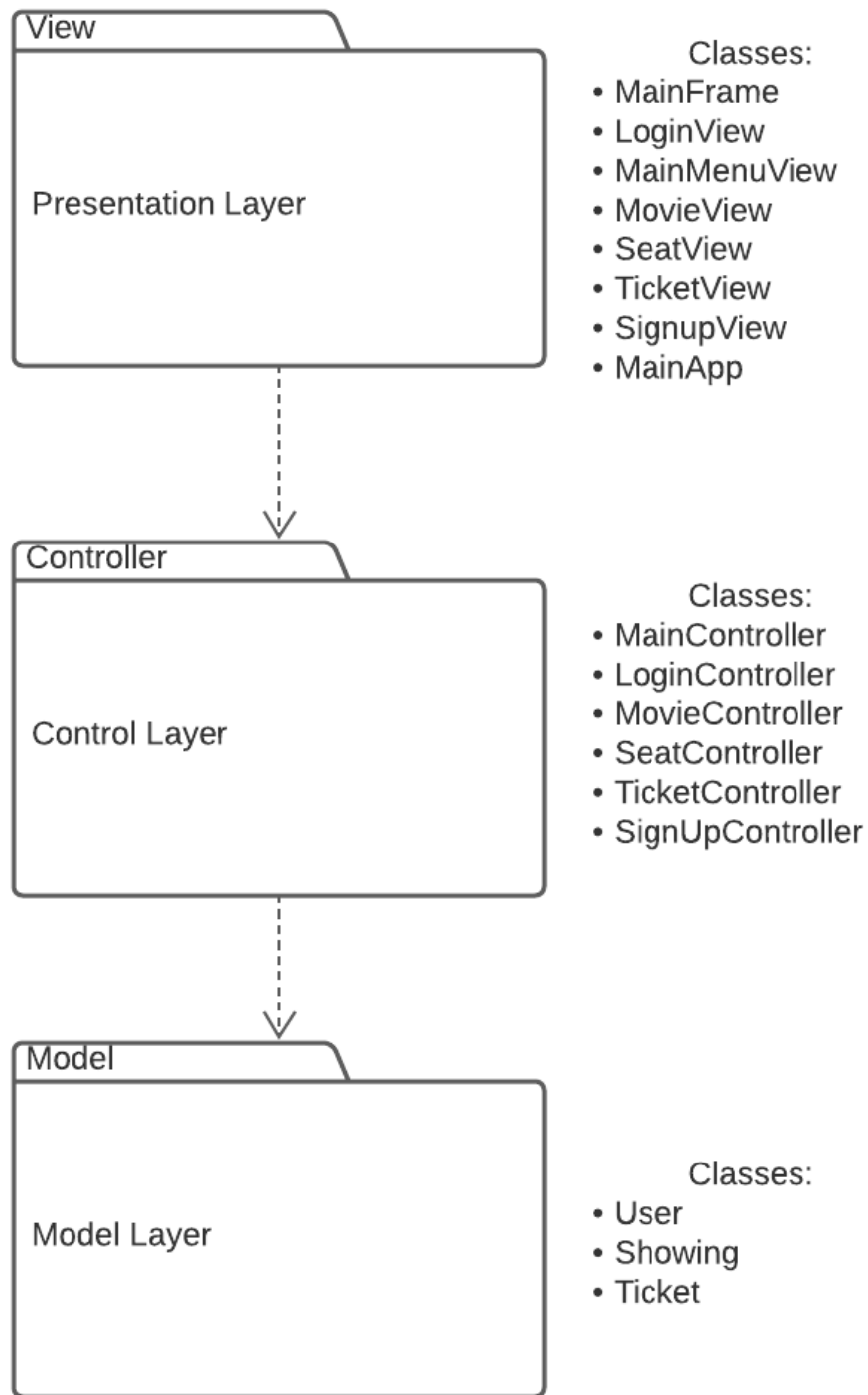
# View



# Controller



## Package Diagram





# Deployment Diagram

