Software Design Specification

For a Movie Theater Ticketing System

By: Nathan Gozun, Adam Larson, Anthony Barbaro, Kellen Murphy, Kobe Tran

Version: 1.0

Publish: March 1, 2023

System Description

For this project, we have decided to design a software system that aims to simplify the ticket buying and the seat choosing process by combining the systems into one system that is accessible on both a mobile app and a website. This system should simplify the process of finding a movie theater or a movie and subsequently purchasing a ticket and selecting a seat.

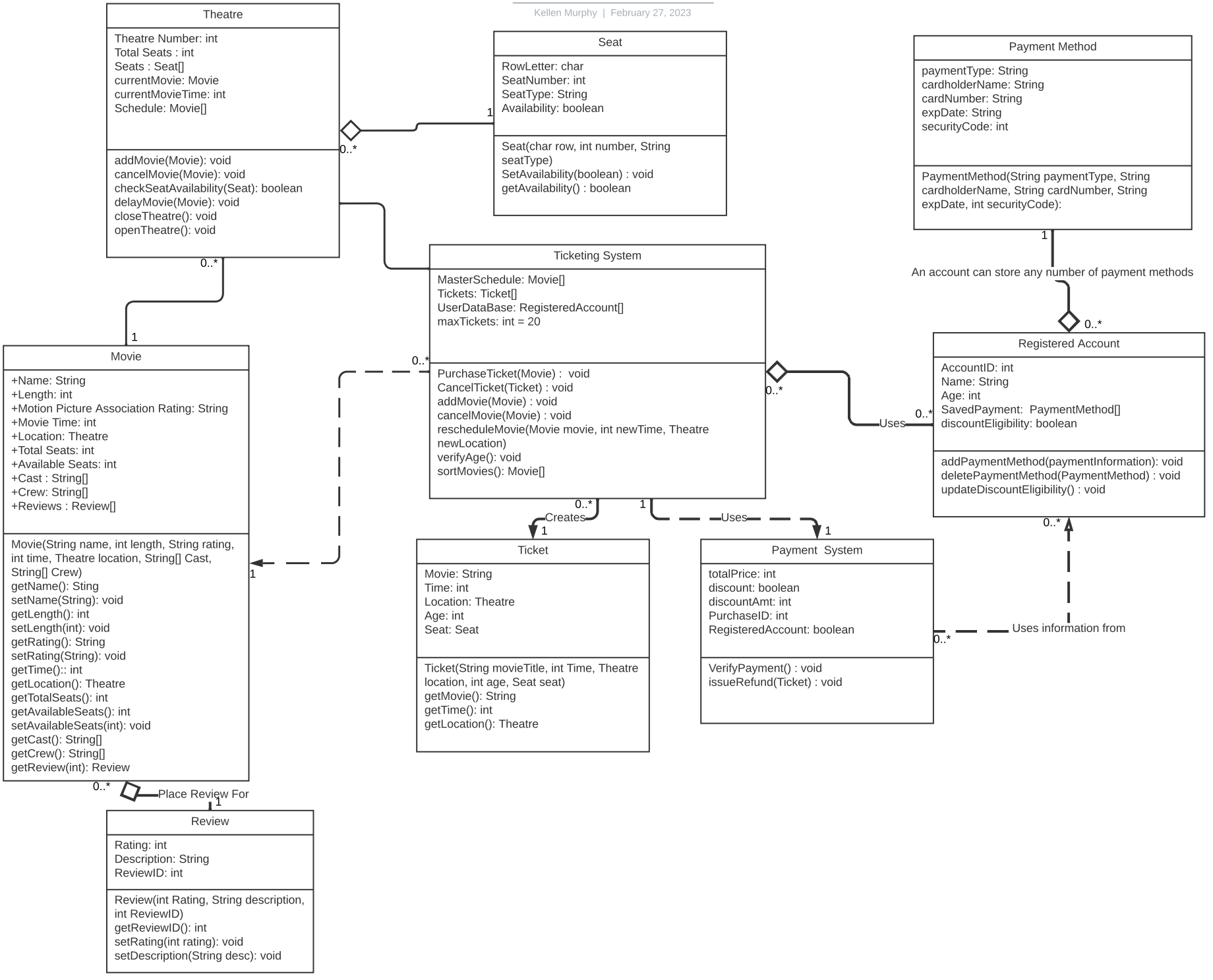
On the back end of the project, we plan on having the ticketing system be the core of the software. It will interact directly with the ticket software, payment system software, theater software, and the account software. The ticketing system should be able to get information about movies without having to go directly through the theater system. The theater system will interact with the movie system and the seat system. Finally, the account system will contain information about the accounts payment method. Using this structure will allow us to maintain control and limit cyberattacks as well as create a solid structure where each system can access the information that each needs.

On the front end of the project, the mobile app and the website should have exactly the same flow. The user should be able to see movie listings, register for an account, and login on the main page. The movie listings page will allow the user to view time and location, the seating arrangement of the theater, and have the option to purchase a ticket. From the registration page, the user should be able to create an account that will allow them to come back and purchase tickets in the future with ease. After logging in, the user should be able to view the tickets that they have and have the ability to cancel their tickets and get a refund if it is available.

The following diagrams will go into deeper detail about both the back end and the front end of the software. Then we have listed an expected timeline and the tasks that each member of the team will be responsible for.

CS250 Group 2 Assignment 2 UML Class Diagram

Kellen Murphy | February 27, 2023



The UML Class diagram represents a Ticketing System, which consists of two main components - a MasterSchedule and Tickets. The MasterSchedule is an array of Movie objects, while Tickets is an array of Ticket objects.

The Ticketing System has three main operations - CreateTicket, addMovie, and cancelMovie, which are used to create a new ticket, add a new movie to the MasterSchedule, and cancel an existing movie from the MasterSchedule, respectively. There is also an operation called rescheduleMovie, which is used to reschedule an existing movie in the MasterSchedule.

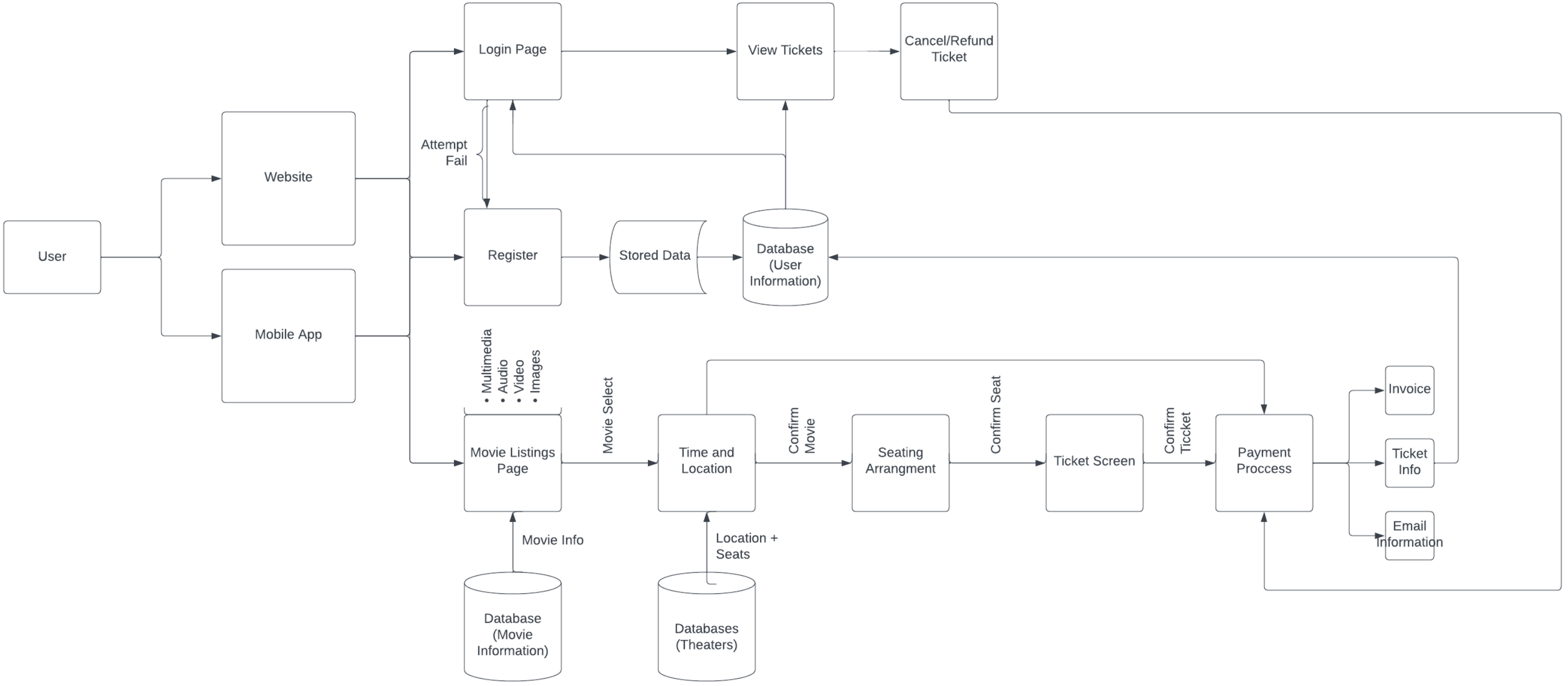
The Ticket class has several attributes, including Movie (a string that represents the name of the movie), Time (an integer that represents the movie's start time), Location (a string that represents the theater where the movie is being shown), Age (an integer that represents the minimum age required to watch the movie), and Payment System (an object that represents the payment system used to buy the ticket). The Payment System object has several attributes, including totalPrice (an integer that represents the total price of the ticket), discount (a boolean that indicates whether a discount was applied), discountAmt (an integer that represents the amount of the discount applied), VerifyPayment (an operation that verifies the payment for the ticket), and issueRefund (an operation that issues a refund for the ticket).

The Movie class has several attributes, including Name (a string that represents the name of the movie), Length (an integer that represents the length of the movie in minutes), Motion Picture Association Rating (a string that represents the movie's rating), Movie Time (an integer that represents the movie's start time), Location (a string that represents the theater where the movie is being shown), Total Seats (an integer that represents the total number of seats in the theater), available seat (an integer that represents the number of available seats in the theater), cast (an array of strings that represents the cast of the movie), crew (an array of strings that represents the crew of the movie), and Review (an array of Review objects that represent reviews of the movie). The Review class has several attributes, including Rating (an integer that represents the rating of the movie on a scale of 1 to 10), Description (a string that represents a description of the movie), Review (an operation that adds a new review for the movie), SetRating (an operation that sets the rating of the movie), and setDescription (an operation that sets the description of the movie).

The Movie class is also connected to the Theater class, which has several attributes, including Theater Number (an integer that represents the number of the theater), Total Seats (an integer that represents the total number of seats in the theater), Seats (an array of Seat objects that represent the seats in the theater), currentMovie (an object that represents the current movie being shown in the theater), currentMovieTime (an integer that represents the start time of the current movie), and Schedule (an array of Movie objects that represent the schedule of movies for the theater). The Seat class has several attributes, including RowLetter (a character that represents the row of the seat), SeatNumber (an integer that represents the number of the seat), SeatType (a string that represents the type of the seat), Availability (a boolean that indicates whether the seat is available or not), SetAvailability (an operation that sets the availability of the seat), and getAvailability (an operation that returns the availability of the seat).

In this Ticketing System, users can cancel a ticket using the CancelMovie operation, which takes a Movie object as input. If the user cancels the ticket before the movie starts, they may be eligible for a refund. The Ticket class has an operation called VerifyPayment, which verifies the payment for the ticket and issues a refund if necessary. Payment methods can be registered to the user's account, allowing for a faster and more convenient checkout process in the future. Users can create a registered account, which stores their personal and payment information for future purchases. This also allows the system to provide a personalized experience for the user, with recommendations based on their purchase history and preferences.

Overall, this UML class diagram represents a Ticketing System, which allows users to create, view, and manage movie tickets, movies, theaters, and reviews. It provides a clear overview of the main components and operations of the system, as well as the relationships and dependencies between them.


User
Website
Mobile App
Login Page
Attempt
Fail
Register
Stored Data
Database
(User
Information)
View Tickets
Cancel/Refund
Ticket
• Multimedia
• Audio
• Video
• Images
Movie Listings
Page
Movie Select
Time and
Location
Confirm
Movie
Seating
Arrangment
Confirm Seat
Ticket Screen
Confirm
Ticcket
Payment
Proccess
Invoice
Ticket
Info
Email
Information
Movie Info
Database
(Movie
Information)
Location +
Seats
Databases
(Theaters)

The SWA diagram represents the system architecture of an online movie ticket booking application, showing the different components and how they are connected.

The user is the primary component and connects to the website and mobile app. The website and mobile app are connected to the login page, register page, and movie listings page. The login page is connected to the register page in case of failed login attempts. The register page is connected to the stored data, which contains user registration information.

The movie listings page is connected to the database that contains movie information, including multimedia such as audio, video, and images. The movie listings page is also connected to time and location, which shows available showtimes and theaters. The time and location component is connected to databases containing theater information, which displays available seats for each theater.

Once the movie and seats are confirmed, the seating arrangement component is connected to the ticket screen component. The ticket screen component is connected to the payment process component, which handles payment information and generates an invoice. The payment process component is also connected to time and location to ensure that the transaction is associated with the correct showtime and theater.

The view tickets component is connected to the cancel/refund ticket component, which allows users to cancel or refund their tickets. The cancel/refund ticket component is connected to the payment process component to handle refunds.

Finally, the ticket information component is connected to the database containing user information and email information to send ticket confirmation and invoice details to the user.

Overall, the SWA diagram shows the flow of information and interactions between the different components of the movie ticket booking application.

Our software system is a Ticketing System that allows users to purchase and manage movie tickets. The system has several components, including a Master Schedule of movies, Tickets that users can purchase, and a Payment System to process transactions. The system also includes features for cancelling tickets, registering payment methods, and creating a registered account.

List of Tasks:

1. Develop user interface for the Ticketing System (Anthony, Kellen)
2. Implement Master Schedule component to store and display movies (Nathan)
3. Implement Ticket component to manage ticket purchases (Kobe)
4. Implement Payment System component to process transactions (Anthony)
5. Implement cancelling ticket feature (Nathan)
6. Implement registering payment methods feature (Kobe)
7. Implement creating registered account feature (Adam)
8. Integrate all components and features of the Ticketing System (All team members)

Estimated Timeline to Complete the Tasks:

1. 2 weeks
2. 3 weeks
3. 4 weeks
4. 3 weeks
5. 1 week
6. 1 week
7. 2 weeks
8. 4 weeks

Team Member(s) Responsible for Each Task:

Anthony: User interface development, Payment System component, project integration
Kellen: User interface development, Ticket component, registering payment methods feature, project integration
Nathan: Master Schedule component, cancelling ticket feature, project integration
Adam: Creating registered account feature, project integration
Kobe: Ticket component, registering payment methods feature, project integration