MOVIE SHOW TIME

Schwifty

Ashanti Russ
Edward Konienczy
Team coordinator: Dawit Abay
Jiawei Sun
Juwan Smith

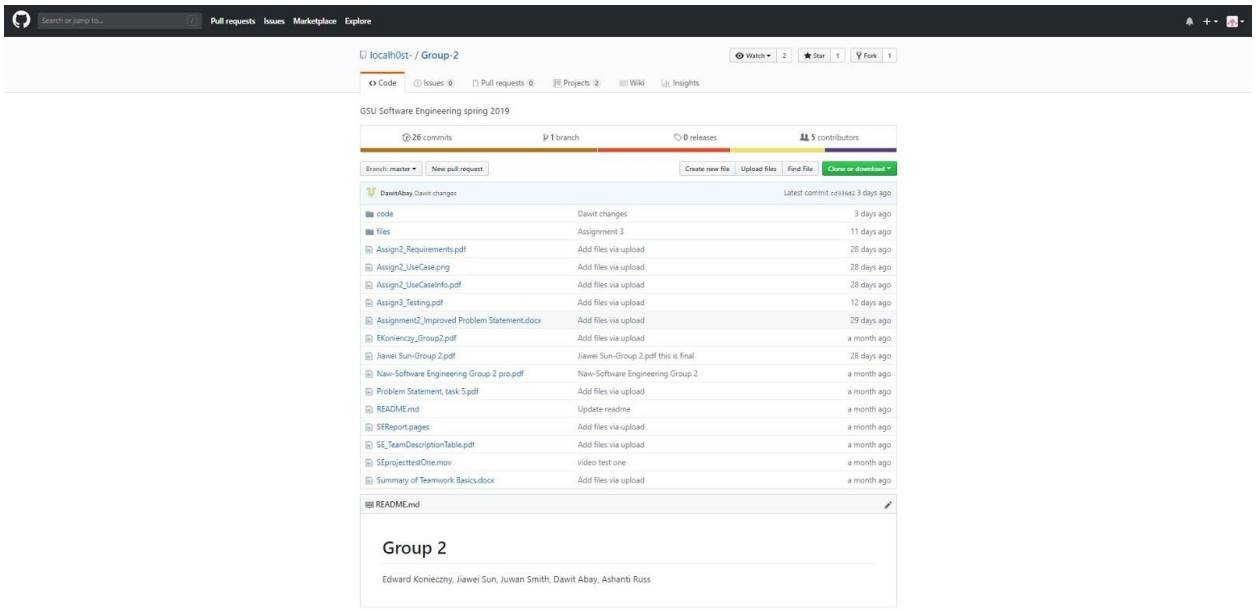
Spring 2019

TASK 1:
Planning Scheduling and Peer Evalua

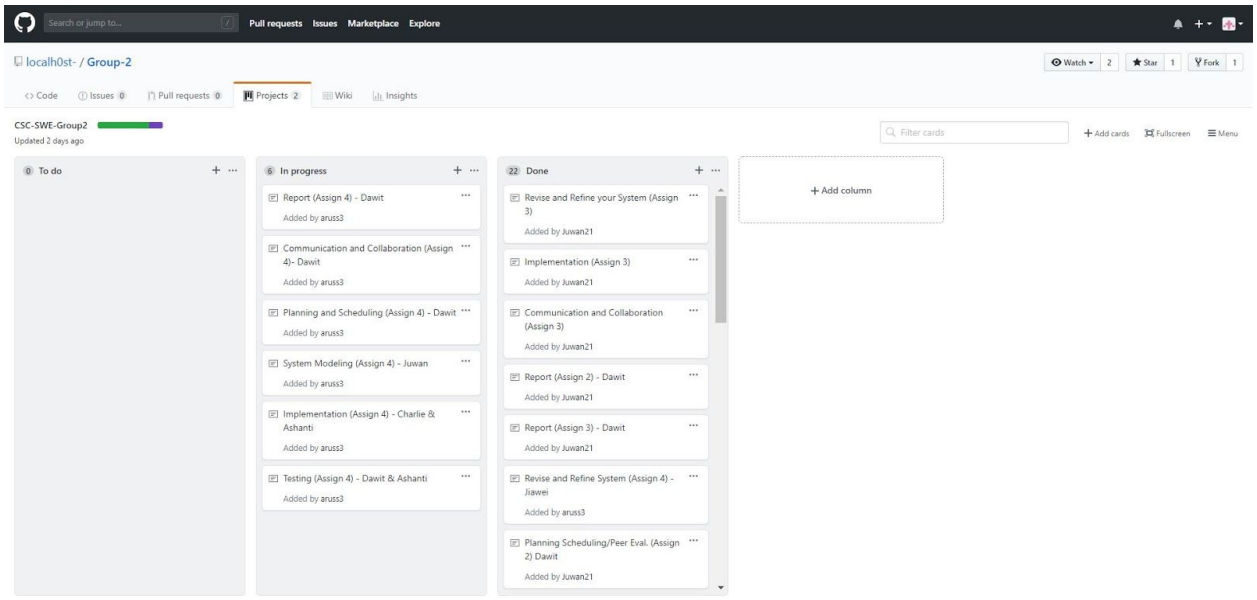
Name	Email	Task	Hour	Dependency	Date	Evaluation
Juwan Smith(TC)	jsmith405@student.gsu.edu	Task 4 System Modeling	5h	System Modeling	03/14/19	100% did a excellent job System Modeling
Dawit Abay	Dabay1@student.gsu.edu	Task1,2,6, 7 Planning/ Report/Github/ Testing	5h	Planning/ Report/Github/ Testing	03/14/19	100% did a excellent job Planning/ Report/Github/ Testing
Jiawei Sun	jsun8@student.gsu.edu	Task 3 Revise System	5h	Revise System	03/14/19	100% did a excellent job Revise System
Edward Konienczy	EKonienczy1@student.gsu.edu	Task 5 Implementation	5h	Implementation	03/14/19	100% did a excellent job Implementation
Ashanti Russ	aruss3@student.gsu.edu	Task 5 Implementation	5h	Implementation	03/14/19	100% did a excellent job Implementation

TASK 2:
GitHub

Screenshot of Group's GitHub page



Screenshot of Group's GitHub Task page



TASK 3:
Revise and Refine your System

1. Problem statement

1) What is your product, on a high level?
Our product provides a service of making showtimes of movies being presented at local movie theatres readily available for potential consumers of entertainment.

2) Whom is it for?
This product is available for any consumer that is in want of an entertaining experience that is within a relatively close environment of where they live and the necessary information, we provide to make that possible. Especially for movie fans who wants to watch the coming movie in ideal place for him or her.

3) What problem does it solve?
It provides readily available information for anyone who does not intuitively know or have access to information regarding the closest proximity movie theatre or showing times for movies in his or her area.

4) What alternatives are available?
Applications such as “Fandango” and “cinema” provide a relative competitive result to what we provide. Also, the more tedious methods are readily available such as finding theatres and showtimes manually through search engines.

5) Why is this project compelling and worth developing?
This project allows for some initial experience with being an individual within a development team and understand the quirks and intricacies required to work as a cohesive unit to achieve some common goal. Along with the previous statement, this provides an opportunity to test our current technical skills gained as students and find areas that we need to improve and hone these skills further.

6) Describe top-level objectives, differentiators, target customers, and scope of your product.
Top-level objectives: Provide easily accessible and free upcoming movie information including movie show time and locations to potential movie consumers.
Differentiators: Our product is limited to the relative Atlanta area so the amount of traffic would be less of an issue opposed to our competitors. And it costs less resources.
Target customers: Customers within the Metropolitan and greater Atlanta areas who interested in movies.

Scope of your product: The scope is relatively consistent with in the realm of movie Entertainment.

7) What are the competitors and what is novel in your approach?
Our competitors include “Fandango” and “cinema”, and other movie information websites. We provide useful information exclusively for the Atlanta area.

8) Make it clear that the system can be built, making good use of the available resources and technology.
There are many frameworks to build this system. For our project, we will Spring Boot as our framework, and use HTML, JavaScript and CSS to create our front-end, Java for back-end, then PostgreSQL for database.

9) What is interesting about this project from a technical point of view?
This project allows for our team to experience the use of full stack development in a single process by needing to create and environment to display information from databases that we create and then port said information over to a visual format to be used by potential consumers.

2. Requirements

(1) System Requirements

1) Check Movie Times
Actors: Customers, Developers, Movie Theatres, Database System
Description: The website contains a variety of showing times for films being shown at local theatres. This provides a free service to customers in need of pertinent information.
Alternate Paths: You could directly search for a specific film you believe to be in theatres and all the required info will be available along with, of course, the show times.
Pre-Condition: The local movie theatres have accurate information regarding the show times for their films. Also, our development team is consistent with the updating accurate information on the website.

2) Search Movies
Actors: Customers, Developers, Database System, Movie Theatres
Description: The website allows the ability to search for current available films that are being shown at local theatres if they’re indeed in our system.
Alternate Paths: You could visit your local theatre directly to find a specific film.
Pre-Condition: Information from the local theatres is crucial for it be accurate and displayed for customers.

3) Buy Tickets.
Actors: Customers, Develops, Movie Theatres
Description: The website contains the ability to purchase tickets electronically for showings at local theatres.

Alternate Paths: You could visit your local theatre and pay for tickets upon arrival.
Pre-Condition: The local movies theatres provide information and available tickets to be sold to customers, that is facilitated through our platform.

4) Watch Trailers
Actors: Customers, Developers Description: The website provides film trailers for current films being shown at local theatres.
Alternate Paths: None.
Pre-Condition: Developers are staying up to date on current films being shown and then acquiring the necessary resources to apply the trailers and info onto the platform. Group 2 CSC 4350

5) Development Team Information
Actors: Developers, Customers, Movie Theaters
Description: The website contains the necessary information on the development such as names, experience, background information etc. This is available to the public.
Alternate Paths: Going to the development team directly in order to gather the desired information.
Pre-Condition: The development continues to keep and update accurate and true information regarding the team and perceived qualifications and experience.

6) Find Theatre Locations
Actors: Customers, Developers, Movie Theatres, Database System
Description: The website contains a variety of information on local theatres that allow you to enjoy a safe, comfortable film experience.
Alternate Paths: None.
Pre-Condition: The developers keep up to date information on the locations of certain theatres and what films are currently being shown at those aforementioned theatres.

(2) User Requirements

1) Retrieve Movie Times
Introduction: This requirement is related to the need of displaying pertinent data for the films on the system.
Inputs: A name or date regarding a specific film you are trying to watch at a local theatre.
Requirements Description: A robust amount of data for films that are being shown at local theatres to be entered into a database system.
Outputs: The system will then display all results that are possibly related to the input and thus showing a schedule of films being run at a theatre.

2) Retrieve Specific Movie
Introduction: This requirement is related to the need of finding specific information in regards to a specified film.
Inputs: A name or date regarding a specific film you are trying to watch at a local theatre.

Requirements Description: Data from a multitude of different films being shown and input into a database to be displayed.
Outputs: The system will return a list of films that are related to the input and then give a certain amount of information on the searched film.

3) Ticket Retrieval
Introduction: This requirement is related to the need of providing an electronic version of a ticket to gain access toward attending a film at a theatre.
Inputs: A credit or debit card of some kind is needed in order to perform an epurchase of a ticket for a film at a local theatre.
Requirements Description: A secure encrypted system that allows a customer to make and charge a payment online.
Outputs: An e-ticket will be sent to a device of a customer that can be used to attend a film at a local theatre.

4) Stream Trailer
Introduction: This requirement is related to the need of providing a short clip displaying an introduction to films being shown at local theatres.
Inputs: A name of the film you are searching for.
Requirements Description: A robust amount of short video files that are uploaded to a database system in order to be displayed on the system.
Outputs: Shorts clips showing scenes from films that are being shown at local theatres.

5) Display Development Team Info
Introduction: This requirement is related to the need of providing information on the development team of the system.
Inputs: No input required. Simply click relevant menu option.
Requirements Description: Information that amounts to a synopsis on the relevant members with in the development team to be uploaded onto the database.
Outputs: Documents that display the abilities and skills of the development team members.

6) Show Theatre Locations
Introduction: This requirement is related to the need of providing locations and information of local theatres you would want to visit.
Inputs: A name of a film you believe to be currently shown at a local.
Requirements Description: Information in regard to films being shown currently, also operating hours and locations of local theatres that have said films available.
Outputs: Times relating to the input of the film being searched and local theatres that are currently showing the film.

3. Use Case Diagram

In this part, we use buy tickets as our use case to show class diagram, which is shown below.

Use case: Buy Tickets
Actors: Customers, Database, Payment Service
Description: The website contains the ability to purchase tickets electronically for showings at local theatres. As customer goes to the website, he or she would use UI to browse the recommendations for movies, then select movie time and pay for tickets.
Alternate Paths: You could visit your local theatre and pay for tickets upon arrival.
Pre-Condition: The local movies theatres provide information and available tickets to be sold to customers, that is facilitated through our platform.
Class diagram:

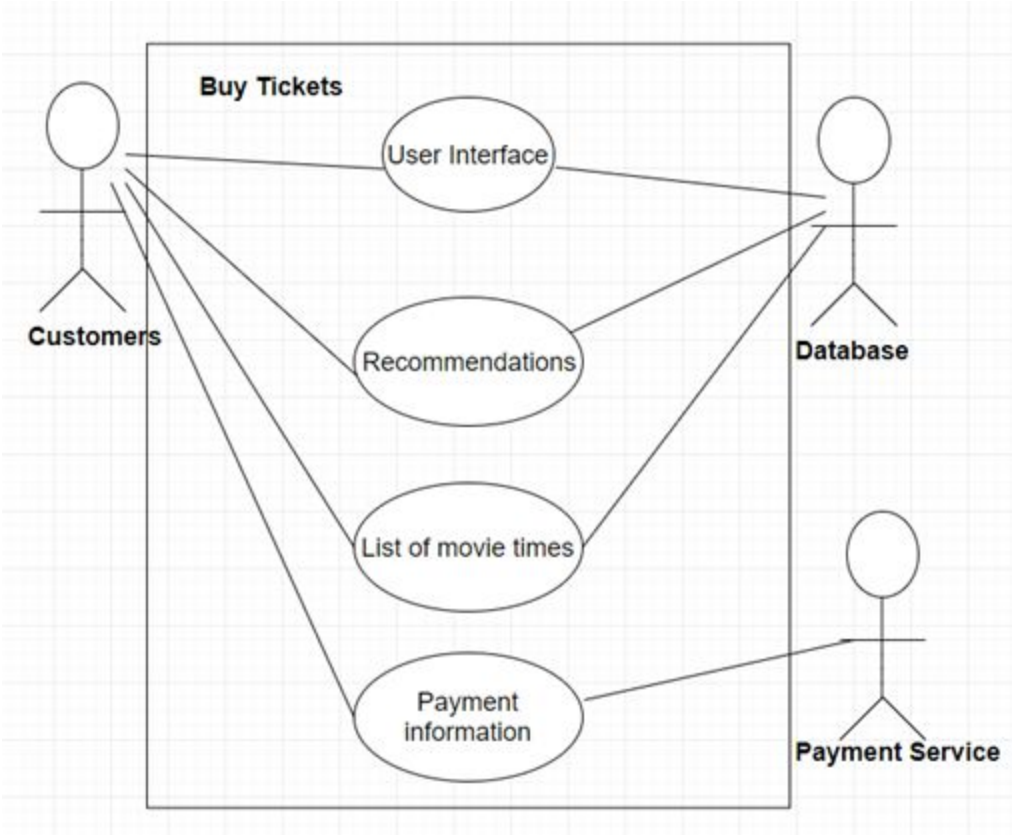
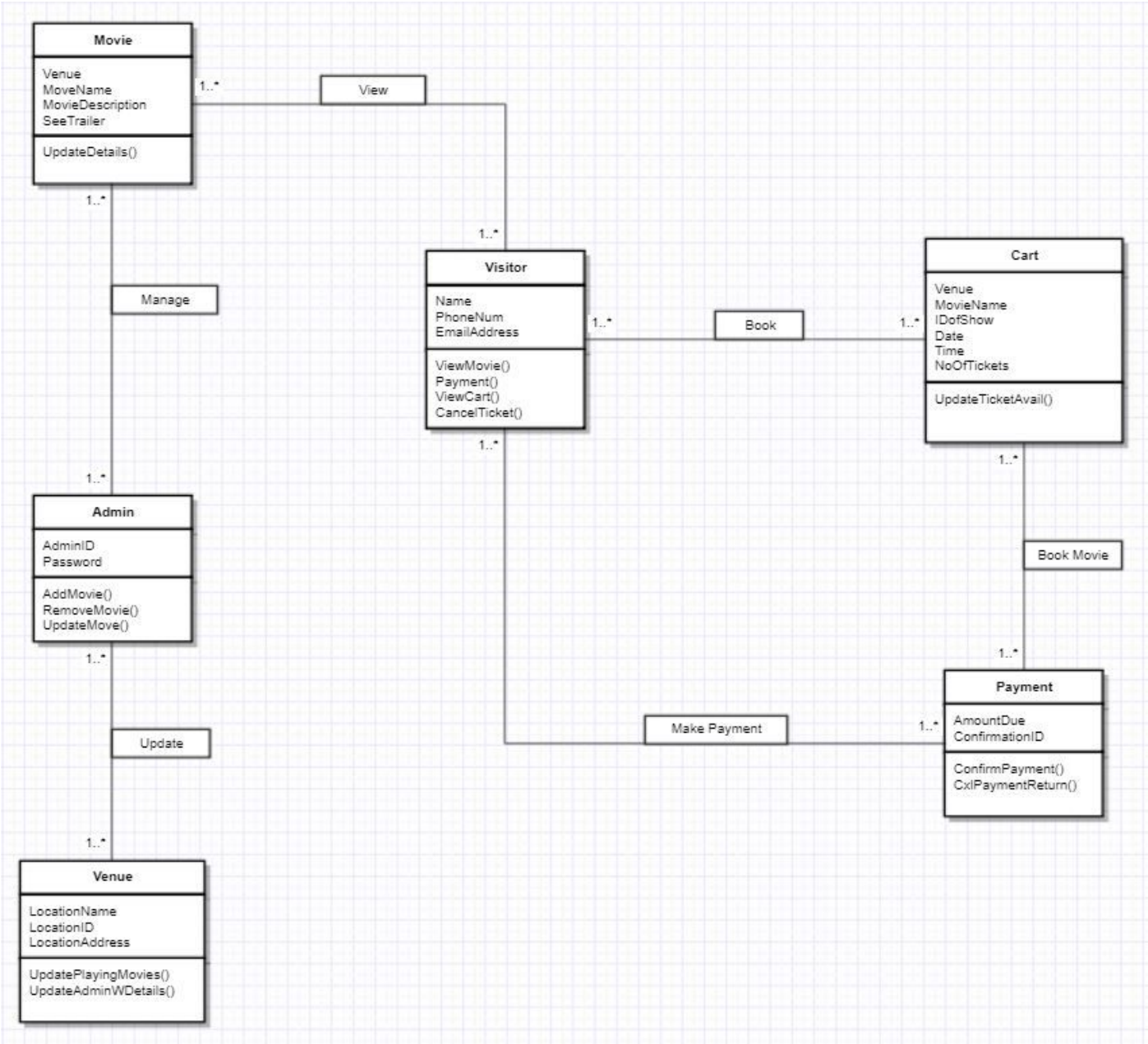


Figure: Class Diagram

4. Class diagram

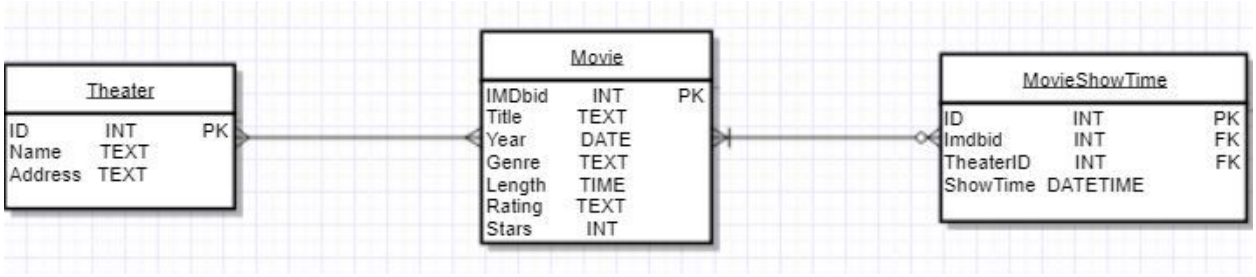
Class Diagram



Class Diagram Analysis				
Objects	Associations	Multiplicity (Instances)	Attribute	Defined Operations
Venue	Admin	Update Admin: One or more	Location Name Location ID LocationAddress	UpdatePlayingMovies() UpdateAdminWDetails()
Admin	Venue Movie	Update by Venue:One or more instances Manage Movie: One or more	AdminID Password	AddMovie() RemoveMovie() UpdateMovie()
Movie	Admin Visitor	Manage by Admin: One or more View by Visitor: One or more	MovieName MovieShow Venue SeeTrailer	UpdateDetails()
Visitor	Movie Cart Payment	View Movie: One or more Book via Cart: One or more Make a Payment: One or more	FullName PhoneNum EmailAddress	ViewMovie() Payment() ViewCart() CancelTicket()
Cart	Visitor Payment	Book for Visitor: One or more Book Payment: One or more	Venue MovieName IDofShow Date Time NoOfTicketsAvailable	UpdateTicketAvail()
Payment	Visitor Cart	Make Payment via Visitor: One or more Book Movie via Cart: One or more	AmountDue ConfirmationID	ConfirmPayment() CxlPaymentReturn()

5.Database Specification and Analysis

System Database Table Diagram



The above database tables and their attributes (including types) are:

- Theater,
    - ID, the ID for the theater (Type: INT)
    - Name, the name of the theater (Type: TEXT, string)
    - Address, the street address of the theater provided in string (Type: TEXT, string)
- Movie
    - IMDbid, the IMDBd ID of the movie (Type: INT)
    - Title, the title of the movie (Type: TEXT, string)
    - Year, the year the movie was released (Type: DATE, int in YYYY-MM-DD)
    - Genre, the genre of movie (Type: TEXT, string)
    - Length, the length of the movie (Type: TIME, int in HH:MM:SS)
    - Rating, the rating of the move (Type: TEXT, string)
    - Stars, the number of stars the movie has; the intended scale 0-10 to be converted into 0-5 stars (Type: INT)
- MovieShowTime
    - ID, the row ID incrementing from one (Type: INT)
    - IMDbid, the IMBd ID of the movie (Type: INT)
    - TheaterID, the ID of the theater (Type: INT)
    - ShowTime, the date and time of all showtimes of a provided movie at a theater provided in an array of datetime (Type: DATETIME)

Concerning the relationships between the tables, the Theater table has a many-to-many relationship with the Movie table, as a theater can show multiple movies and movies can be viewed in multiple theaters. As for the relationship of the Movie table and MovieShowTime table, it is a many option-to-many mandatory since for movie show times to exist, multiple movies must be available to be seen and multiple movies will provide various showtimes. Primary keys that are notable include IMDbid of the Movie table, and ID of the Theater table and MovieShowTime table. Notable foreign keys are IMDbid, as it references to Movie.IMDbid, and TheaterID, as it references to Theater.ID. We will be using PostgreSQL database as our database management system.

6. Refined sequence diagrams

Two major use cases shown are: adding a movie's showtime (which originates from the venue) and the purchasing of a ticket (which originates from the visitor). Additional details concerning the components of the diagram is provided below.

Lifelines:

- Admin
- Visitor
- Movie
- Book Ticket
- Payment
- Venue

Synchronous messages:

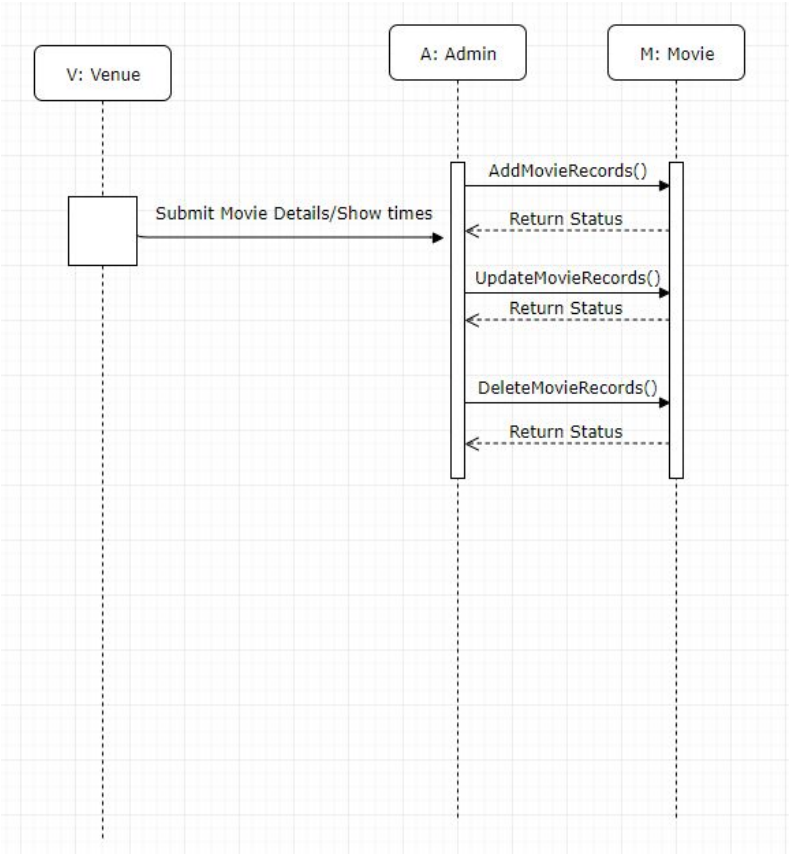
- AddMovieRecords()
- UpdateMovieRecords()
- DeleteMovieRecords()
- MakePayment()
- ValidateUserInfo()
- View Confirmation
- Submit Movie Details/Showtimes

Asynchronous messages:

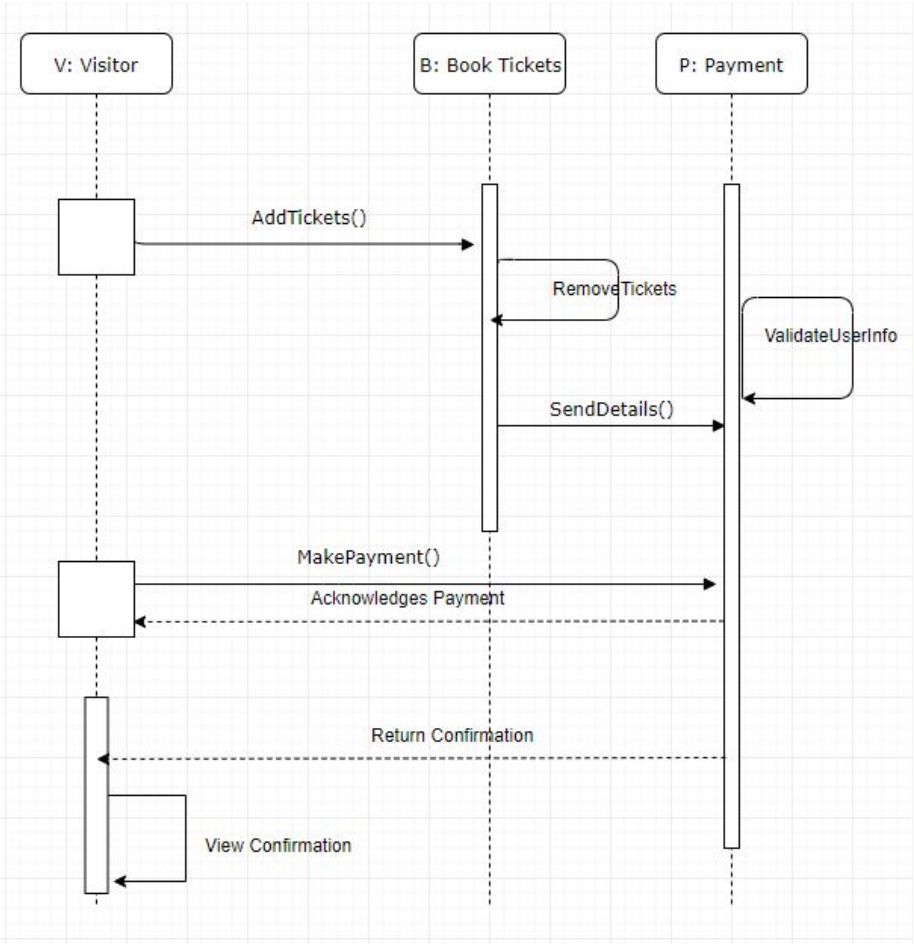
- AddTickets()
- ViewMovie()
- RemoveTickets()
- SendDetails()

The two sequence diagrams are shown as below

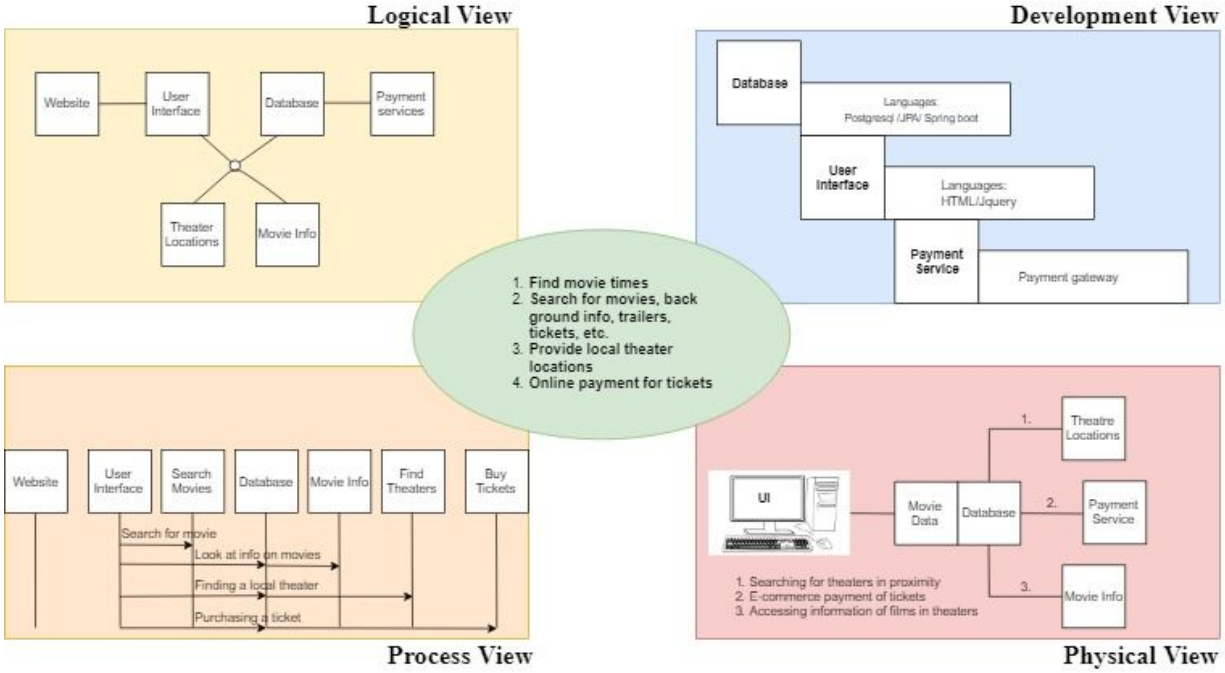
Sequence diagram for adding a movie's show time



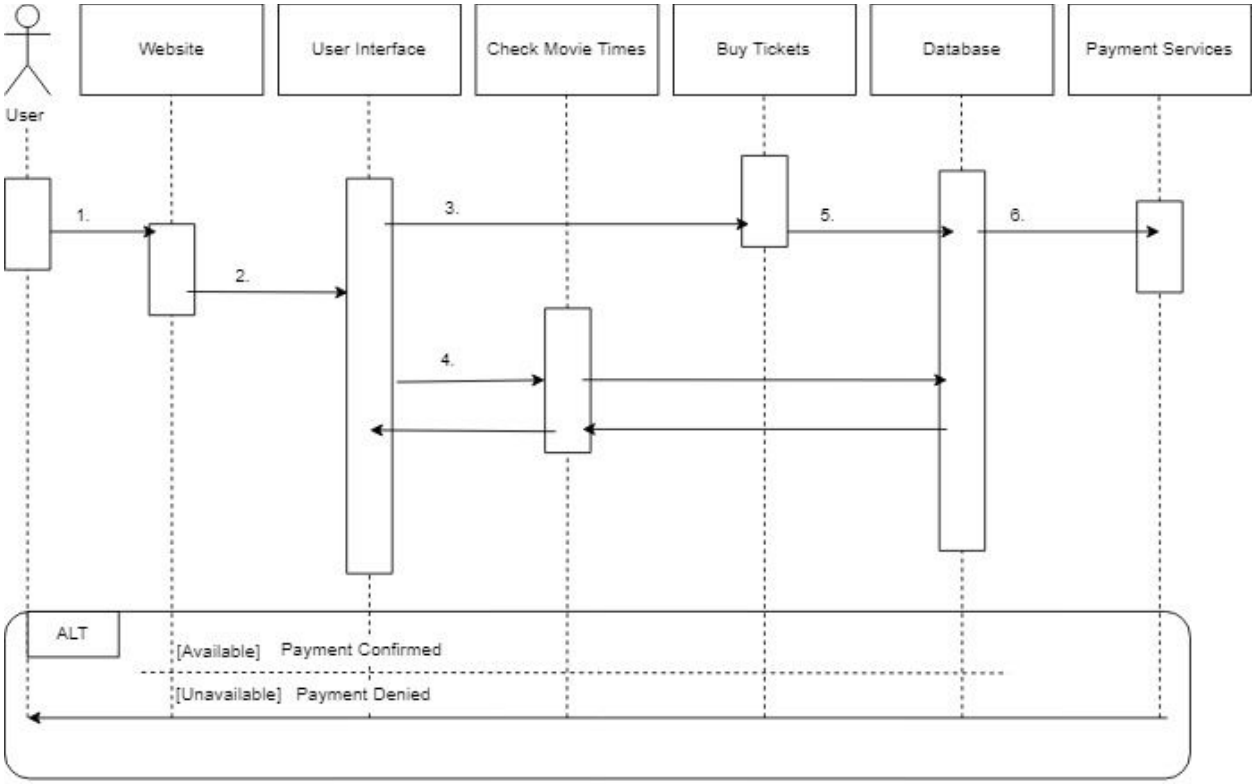
Sequence diagram for buying tickets



TASK 4:
System Modeling (Analysis)



○



1. Access the website
2. Interact with the User Interface
3. Purchasing Request for a ticket
4. Accessing info about showing times
5. Checking for available ticket info
6. Processing Purchase Request

TASK 5:
Implementation

(1) Implementing the Database Design

As needed, the chosen tables have been implemented. All tasks criteria for this task have been completed as per specifications.

(2) Implement the Class Diagram Design

As needed, additional classes have been created as well as testing the communication between the Frontend and Backend, which was a success, has been completed. To compile the code, one can do so by going to the necessary directory and running the script gradlew.

TASK 6:
Testing

1. Test the Environment

- probably the most complete way to do it would be postman in combination of Junit.
    - Test Cases

1. Develop a collection of requirements
    a. Identify features (functionality)
        i. Search bar
        ii. Nav bar
        iii. Home page
        iv. About us page
        v. Report page
        vi. Movie trailer page
        vii. Search text /button
    b. Partition inputs into
        i. Ant-m -error
        ii. Iron ma -error
    c. One possible partition
        i. Ant-man correct
    d. One possible partition
        i. Ant-man correct
    e. Test Specification
        i. If the search text is empty nothing will happen it stay in on the home page

2.Implementing the Test cases

• **Watch Trailers**: The website provides film trailers for current films being shown at local theatres.

• **Search Movies**: The website allows the ability to search for current available films that are being shown at local theatres if they're indeed in our system.

Test ID	Search Movies	Watch Trailers
purpose of Test	Search for movie by name	Watch movie trailers
Test Environment	It on the navigation bar	It on the movie info page
Test Steps	Press the search bar and click on the search button	Press the search text "Ant-Man" and click on the search button
Test Input	Search movie "Ant-Man"	Search movie "Ant-Man"
Expected Result	background, time, & ticket info	Show movie info with movie trailer video
Likely Problems/Bugs Revealed	Find the movie info	Find the movie trailers

3.Test Documentation

Bug	The test uncovered the bug	Description of the bug	Action was taken to fix the bug
Movie container	not display the container	not display the container	Create JS/CSS for movie container
Not finding backend	Not connect to frontend to find the movie	Not connect to frontend	Create psql
Not connect to javascript	HTML in not connect to the script	HTML in not connect to the script	Forgot to add .js
Tomcat connector configured	listen on port 8080 failed to start	listen on port 8080 failed to start	Verify the connector's configuration, identify and stop any process that's listening on port 8080
type=Bad Request, status=400	Required String parameter 'title' is not present	Required String parameter 'title' is not present	/api/search?title=root

Appendix

Group's GitHub link: https://github.com/localh0st-/Group-2
Screenshot of GitHub Task page