Title page

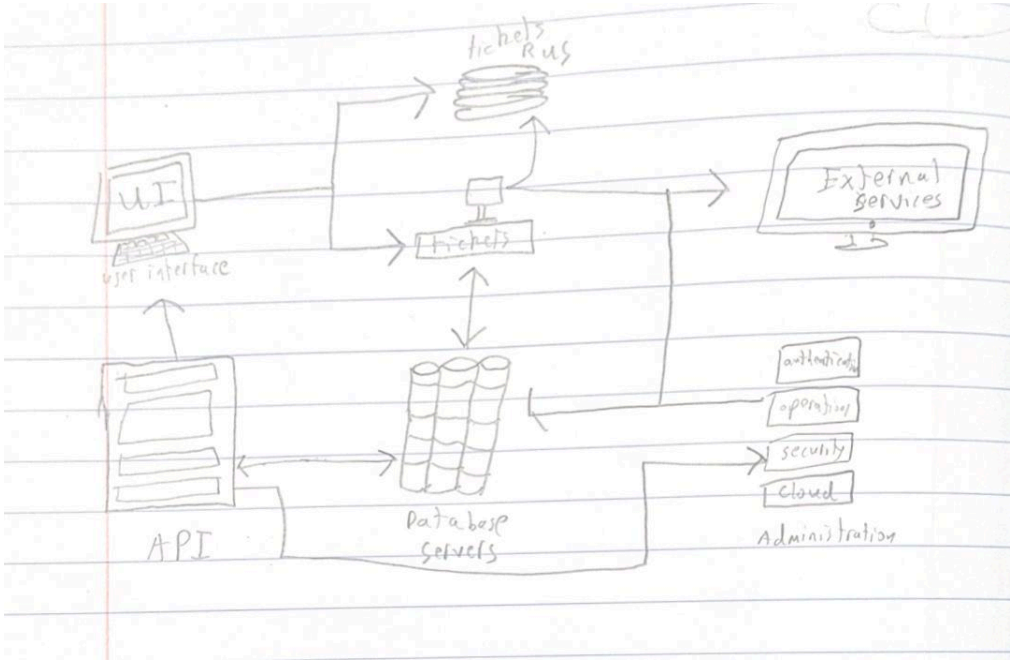
- Software title
- Team members

System Description

- Brief overview of system

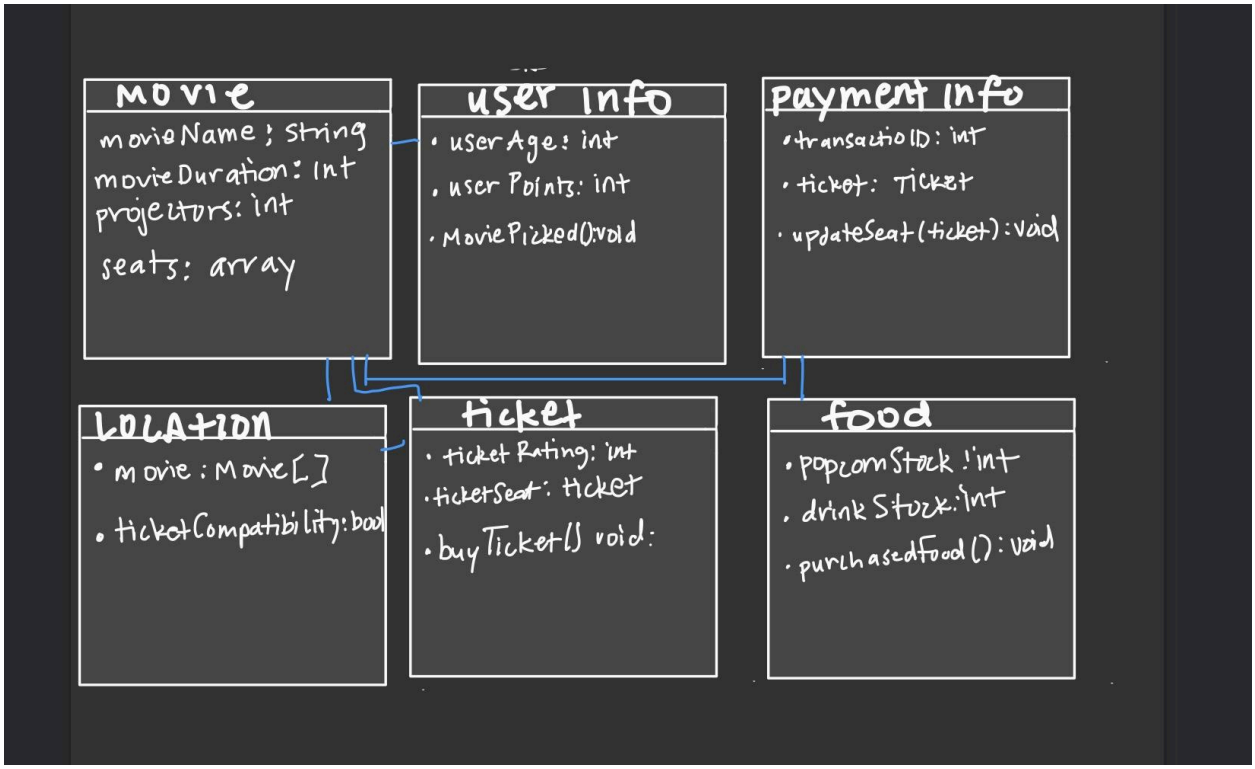
Software Architecture Overview

- Architectural diagram of all major components



  - **User Interface:** Represented by icons of a desktop indicating the software is accessible via web and mobile platforms.
  - **API Layer:** Shown at the base of the diagram, symbolizing the foundation that handles communication between the user interface and other system components.
  - **Tickets Module:** Positioned centrally above the API layer, representing the core functionality of the system where ticket-related operations are processed.
  - **Database Servers:** Illustrated with a stack of cylinders, suggesting data storage capabilities, possibly for user data, ticketing information, and performance details.
  - **External Services:** Depicted with cloud icons, which indicate integration with third-party services such as payment gateways, email services, or other APIs.
  - **Administration:** Represented by a stack, implying tools or dashboards for system administrators to manage and monitor the software's operations.

- UML Class Diagram



- Description of classes
  - Movie:
    - The Movie class entails the current film being projected, the duration of the movie, and the projectors displaying the movie. It also has an array of seats available and seats taken.
  - UserInfo:
    - The user info monitors the user's age which determines the rating of the movie they can view, and also keeps track of their account's points and the movie they picked as well.
  - PaymentInfo:
    - PaymentInfo covers the transaction ID of the ticket purchased, keeping a database of all tickets procured.The paymentinfo class also updates the seat chosen, altering the seating array in the Movie class.
  - Location:
    - The location is a simple class, containing the movie and the compatibility of the user's ticket and the location.
  - Ticket:
    - Ticket describes the rating of the ticket, such as Rated R or Rated E, and also keeps track of the seat and includes a buyTicket method which is the process of buying a ticket.
  - Food:

        - Food simply keeps track of the theater's food stock, which when it is low, means that popcorn, hot dogs, and drinks need to be restocked for additional customers who want to purchase food.
- Description of attributes
    - movieName is a string as it just includes the name of the movie, and movie duration + projectors includes the length of the movie and the amount of working projectors in the theater. Seats are ordered in arrays containing individual seats and their availability.
    - userAge and userPoints are both integer values as they only require numbers. MoviePicked is a method without a return value.
    - transactionID returns an integer value while ticket returns a ticketID called "ticket" which is used in updateSeat() function which updates the seating in the theater.
    - Movie is an array which has a list of movies showing for the week associated with the corresponding location for the showing of each movie. Ticket Compatibility is a boolean value which returns true if the ticket is valid in the location and false if it is not.
    - ticketRating returns an integer value of the rating type, e.g. rating R is a value of 1, etc. ticketSeat returns the "ticket" value as well which is used to determine the location of the seat. buyTicket() is a function used when the user is in the process of purchasing a ticket.
    - The integer values in food are keeping track of the current food in the stock, and purchasedFood() is an example of a receipt which confirms the food has been purchased.
- Description of operations

* descriptions should be detailed and specify data types, function interfaces, parameters, etc..

Development plan and timeline

- Partitioning of tasks
- Team member responsibilities

Just3Guys

Theater Ticketing Software

Software Design Specification

Version 1.0.0

Feb 29, 2024

Group 12
Matthew Tran, Samuel Walls, Victor Tepordei

Prepared for
CS 250- Introduction to Software Systems
Instructor: Gus Hanna, Ph.D.
Spring 2024

Description, development plan, and timeline.

All information present including an overview of the software system, a list of tasks, an estimated timeline to complete the tasks, and which team member(s) is responsible for each task.

Development plan
Project will be divided into modular subsections which will each be completed by their corresponding team.

1.1 Frontend (Samuel Walls)

Front-end development plan from beginning to end.

1.1.1 UI

1.1.1.1 Wide-screen (Desktop)

1.1.1.2 Small-screen (Mobile)

1.1.2 QA

1.2 Backend (Victor Tepordei)

1.2.1 API Server

1.3 Testing system (Matthew Tran)

1.3.1 Unit testing
1.3.2 Integration testing

1.3.3 System testing

1.3.3.1 Alpha

1.3.3.2 Beta

1.3.4 User acceptance testing