Group 13 - BasedMovies123

Team Members: Erasto Astudillo, Felix Marecaux, Aidan Shotland, Carlos Anguiano
GitHub: https://github.com/canguiano13/cs250-designSpecificationAssignment

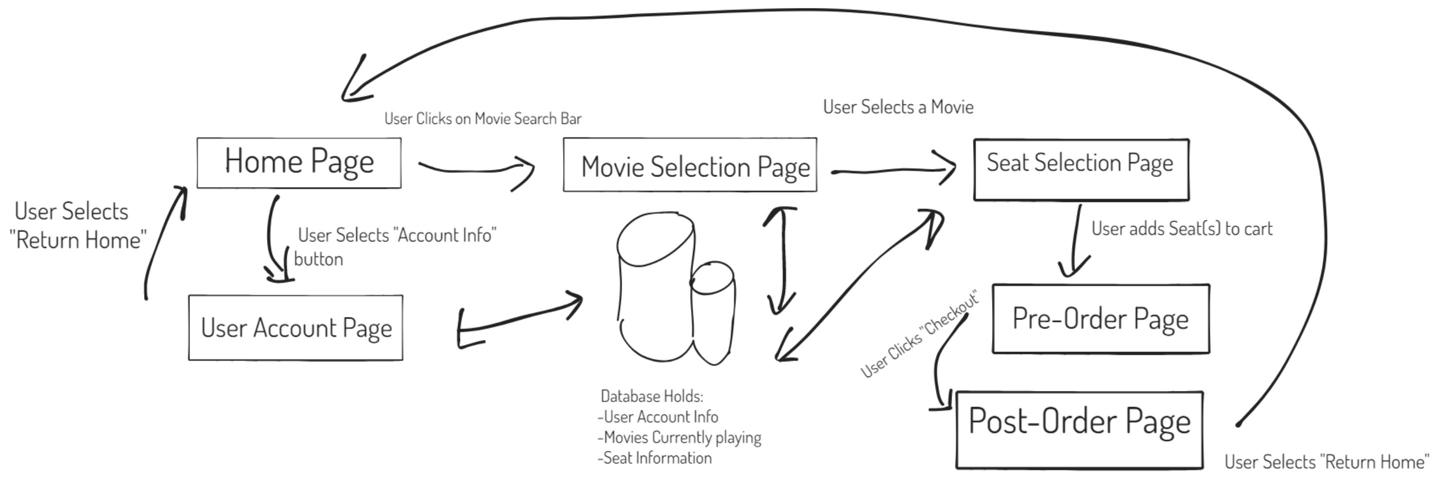
System Description:

<u>-Brief Overview of System:</u>

- Our system implements the ticketing system that is discussed in class. It begins with the home page, where the user has two places they can navigate to. They can first navigate to the “account” page, which displays the user’s account information such as payment methods, email address, and username. Internally, we also have the user’s AccountID.They can also instead navigate to the movie selection page. On this page, the user can search any movie that is currently playing in the theater near them. They can search for keywords or for a specific movie title, and can sort results by rating or genre.
- Once the user has selected a movie, they are then sent to the seat selection page. On the seat selection page, they can view which seats are currently available and which seats are currently taken. A user can only add un-reserved seats to their cart. Once they have added all of the seats that they wish to buy, they can navigate to the pre-order page. This page displays the seats that they currently have selected which allows them to verify their order and email address that the ticket information/receipt will be sent to. Once they have verified their order and put in their payment information, they can place the order. Once this is done, they are sent to the post-order page, which again displays their order information as well as a “thank you” message. Internally, we will be notified to send a receipt and the tickets to the user’s email address. Lastly, from this page, the user can return back to the home page.

Software Architecture Overview:

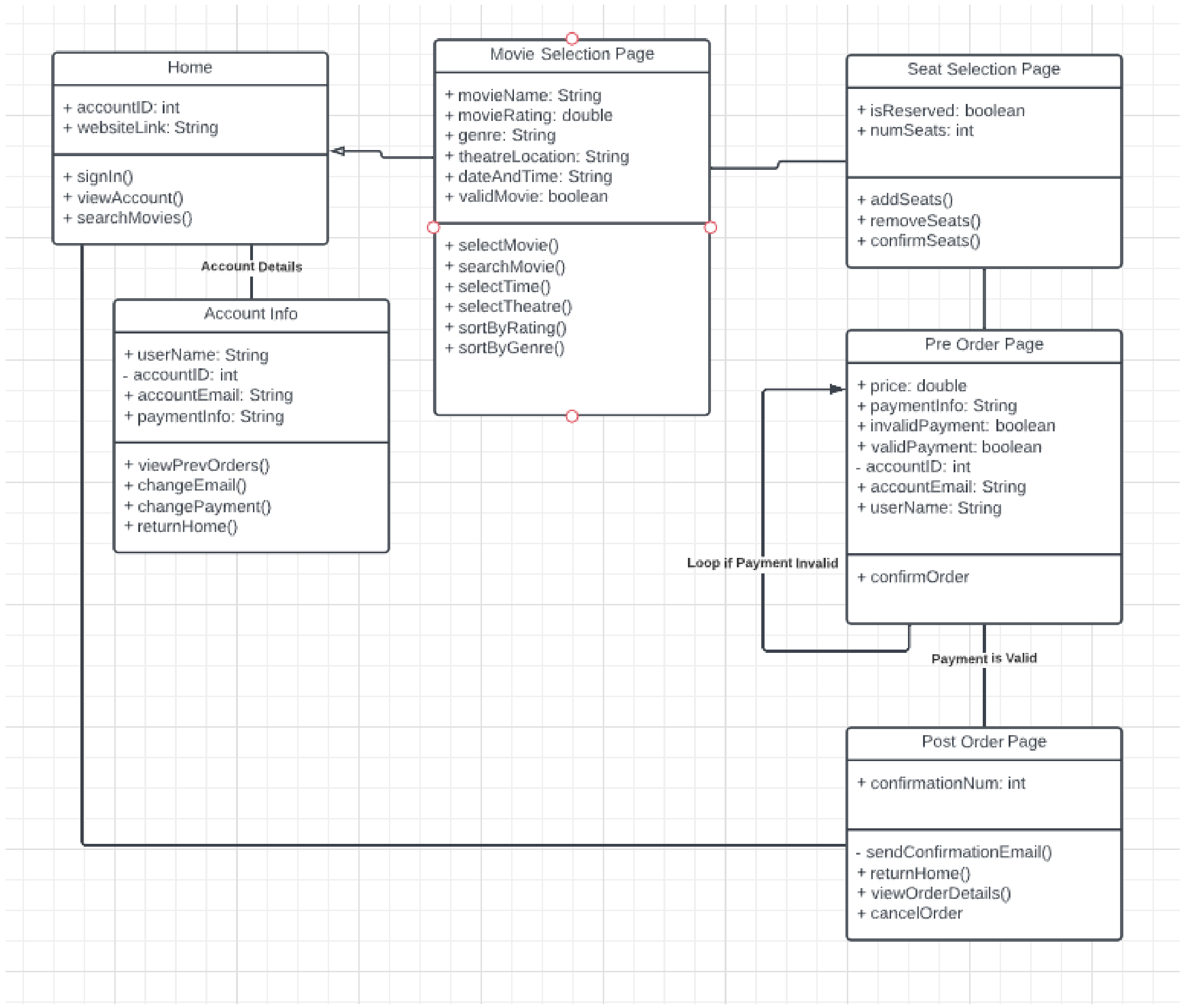
-Architectural Diagram of all Major Components:



-Description of Diagram:

- The above diagram highlights the major components of the software. We have six pages that the user can navigate to. The account page simply displays user information and allows the user to modify this information. The home page allows the user to navigate to the movie search or to their account information. Each of the other pages contribute to a step-by-step process that allows a user to select a movie, reserve seats, view their order information/place their ticket order, and receive confirmation for their order, respectively. We also require databases which hold information about which movies are currently playing and which seats are available for each movie. We also need a database which holds user information and information about confirmed purchases.

-UML Class Diagram



-Description of Classes

- Home
  - The first page that the user sees when the user opens the software. From this class, the user can view their account info or can use the search bar module. This allows users to search for specific movies or browse all available movies that they may buy tickets/reserve seats for.
- AccountInfo
  - The user may navigate to the AccountInfo page where they can view and edit any contact information, or payment information associated with their account. They can also view any previous purchases made under their accountID. Their previous purchases are imported from an internal database.

- MovieSelection
  - This page allows the user to search for a specific movie using the movie search bar, or to browse available movies. They can also sort available movies by several factors such as genre or rating.
- SeatSelection
  - This class allows the user to see/reserve available seats. They can see which seats are currently open or reserved. They may also select open seats and add them to their cart. Once they have selected all the seats they wish to buy, they can go to the pre-order page to verify their order.
- PreOrderDisplay
  - This class allows users to browse restaurants that are open and available for delivery. From this class, the user can sort the available restaurants by rating, type of food, or distance. The user can also use viewRestaurant() to access the Restaurant class which would allow them to place and manage orders
- PostOrderDisplay
  - This class sees users place and manage orders. Here, the user has already selected the specific restaurant that they wish to order from. The menu is displayed and they can add/remove items from their order. Once they have selected all of the items they wish to purchase, they can use confirmOrder() to navigate to the payment class. They may also add special instructions for their

-Description of Attributes:

- Home Class
  - accountID:
    - This ID will be an integer variable which is unique to every user that uses the ticketing software system. It is a way to distinguish between every user.
  - websiteLink:
    - This will be a string integer which is used to access the website once a user is on a search engine.
- MovieSelection Class
  - validMovie:
    - This is a boolean variable which when set to true means that the movie is available at our theaters, and when it is set to false, that means we are not showing that movie.
  - movieName:
    - This is a string variable which will hold the names of the movies that are currently showing at the theaters.
  - movieRating:

        - This will be a double variable which will hold a rating for each of the movies that are currently showing. This movieRating variable will be used to sort a list of movies based on their rating.
    - genre:
        - This variable will be of type string and is used as a descriptor of the movies, holding a one or two word string which states the genre of the movie.
    - theatreLocation:
        - this will be a variable of type string, it holds the address of the different movie theater locations. Location will be a way to sort through the list of movies that are available.
    - dateAndTime:
        - This variable is of type string, and holds the day which the movie is playing, as well as the time it will be showing.
- SeatSelection Class
    - isReserved:
        - This is a boolean value which will prevent you from selecting a seat that is currently reserved, if the value of this variable is set to true.
    - numSeats:
        - This variable will be of type int and holds the number of seats that the user is selecting to purchase.
- PreOrderDisplay Class
    - price:
        - This variable will be of type double and shows the final price for the movie and number of seats selected.
    - paymentInfo:
        - This holds the information of the debit/credit card that is used to make the purchase. It can also be saved as part of the user's info for their account.
    - invalidPayment:
        - this will be a boolean that causes the user to re-enter payment information if it is not valid.
    - validPayment:
        - if the payment is correctly received by the ticketing software system's servers, then this boolean variable allows the user to reach the next screen.
    - accountID:
        - This is a private int variable that is used to make sure that our system knows who is making this purchase, and that the correct user will receive the reservations for the seats.
    - accountEmail:

        - this is a string variable used to email a receipt and confirmation to the user once payment is confirmed
    - userName:
        - This is the personal information of the user, which will be used during checkout to make sure tickets can not be resold. It is a string variable.
- PostOrderDisplay Class
    - confirmationNum:
        - This is an integer variable that is used to make each order unique, that way if there is an issue, the correct ticket can be brought up quickly, as well as serving as proof of purchase.
- Account Class
    - userName:
        - This is the personal information of the user, which will be used during checkout to make sure tickets can not be resold. It is a string variable.
    - accountID:
        - This is a private int variable that is used to make sure that our system knows who is making this purchase, and that the correct user will receive the reservations for the seats.
    - accountEmail:
        - this is a string variable used to email a receipt and confirmation to the user once payment is confirmed
    - paymentInfo:
        - This holds the information of the debit/credit card that is used to make the purchase. It can also be saved as part of the user's info for their account.

-Description of Operations:

- Home Class
    - signIn()
        - This function will be used to allow users to sign into their account, which will be able to save the user's information such as email, payment info, etc.
    - viewAccount()
        - This function will move the user from the home page to the account page, where they can view their information and change it.
    - searchMovies()
        - This function will bring the user to the movie selection page.
- Account Class
    - viewPrevOrders()
        - This function allows users to see the previous purchases they made on the website, only if they are signed in.

    - changeEmail()
        - This function will allow the user to change their email address, so they can change where the confirmation email will be sent.
    - changePayment()
        - Allows the user to change their payment information, or discard it completely so that our system no longer retains the information.
    - returnHome()
        - Will send the user back to the home screen
- MovieSelection Class
    - searchMovie()
        - Allows users to use a search bar to look up keywords and try to find if a movie is available or not at our theaters.
    - selectMovie()
        - This will bring the user to the seat selection screen, as this function confirms the movie that the user has selected.
    - selectTime()
        - Allows the user to search for movies by time
    - selectTheater()
        - Allows the user to search for movies showing at a specific theater
    - sortByRating()
        - Sorts the current movie selection by their ratings
    - sortByGenre()
        - Allows the user to specify a specific genre that they would like to see
- SeatSelection Class
    - addSeats()
        - This allows the user to add a selected seat to their cart and therefore temporarily reserves it while they continue to check out
    - removeSeats()
        - This allows the user to remove a selected seat if they have changed their mind or would like a different seat.
    - confirmSeats()
        - This will confirm the seats that are currently selected, and send the user to the next screen, which is the checkout page.
- PreOrderDisplay Class
    - confirmOrder()
        - Once all the information for the payment has been filled out, users can confirm their order. If it was valid, they will go to the next page, if it was invalid they will have to try again.
- PostOrderDisplay Class
    - sendConfirmationEmail()

        - This will send an email to the user’s email address with a confirmation number, the information of their seats, theater location, movie information, etc.

- returnHome()
    - Sends the user back to the home page
- viewOrderDetails()
    - Allows the user to review their purchase
- cancelOrder()
    - Allows the user to cancel their order, which means seats will be unreserved, payment will be sent back, and another confirmation email will be sent out.

Development Plan and Timeline:

-Partitioning of Tasks

Erasto Astudillo - Requirements Gathering - **deadline 3/1/23**

- Establish needs of the customer

Felix Marecaux - Design

- Determine structure of software -**deadline 3/1/23**
- Create useable UML diagram for software

Carlos Anguiano - Implementation **- deadline 4/1/24**

- Build software
- Code and test

Aidan Shotland - Verification / Maintenance **- deadline 4/20/24**

- Test if system works
- Are stakeholders happy?
- Fix bugs

-Team Member Responsibilities

Regardless of individual responsibility, entire team will work and help each other

Erasto Astudillo will find out what the stakeholders and customers need. He will find out how well the system serves its purpose. He will ask the stakeholders specific questions regarding budget, manpower, deadlines, functionalities, etc. To find out what the customer needs he will research similar platforms with similar functionalities that have had success in the past. In addition, he will collect data by conducting surveys on all social media platforms along with asking relatives and peers their opinions. The collection of data will be completed by **March 1st 2023.**

Felix Marecuix is in charge of design. He needs to specify the structure of the software along with designing the architecture. He will make a shared document with his team members to help brainstorm ideas on the design. This document will include a few elements, a brief overview, a UML and architecture diagram, description of classes and attributes and operations, and finally this timeline of responsibilities. The overview is essentially a summary or introduction to the functionalities of the software. The architecture and UML diagram will provide team members with a visual representation of the design. The descriptions will breakdown every step of the implementation of code. This responsibility will have a deadline of **March 1st, 2023**.

Carlos Anguiano will be in charge of the actual implementation of the code. He is in charge of the team at this stage but everyone will be responsible for contributing something to implementation. Each member of the group will be assigned a different job by Carlos and he will assist everyone in need. Depending on individual strengths, the jobs will be divided into: classes, attributes, and operations. This division of jobs will allow the job to be done more efficiently and effectively with constant communication among team members. The actual implementation of the code's timeline is TBD. The hard deadline for the stakeholders is **April 1st 2024.**

Aidan Shotland will be responsible for the failures. He will be constantly testing and debugging the program to ensure that it is exactly what the stakeholders and customers require. The main goals of testing will be to detect, locate, fix, failures, faults and errors. Any problems such as bad customer reviews, unhappy stakeholders, and angry management will all go through him. He will be assigned this position for the foreseeable future however, the service will need to be available for consumers by **April 20th, 2024.**