Classics Theater System

Trace Crafton

System Implementation

IUPUI

Table of Contents

Problem Statement…………………………………………………………………….....3

Glossary of Terms……………………………………………………………………3

System Requirements………………………………………………………………3

Functional Requirements……………………………………………………………3

Nonfunctional Requirements………………………………………………………...4

User Interface Requirements…………………………………………………………5

Functional Requirements Specifications…………………………………………………..7

Stakeholders……………………………………………………………………………7

Actors and Goals……………………………………………………………………..7

Use Cases………………………………………………………………………………8

Use Case Diagram……………………………………………………………………9

Class Diagram………………………………………………………………………10

System Sequence Diagrams……………………………………………………………11

Movie Show Booking………………………………………………………………11

Payment Verification…………………………………………………………………12

Activity Diagram……………………………………………………………………..13

User Interface Specifications……………………………………………………..14

User Effort Estimation………………………………………………………...15

Plan of Work……………………………………………………………………………16

Problem Statement

A trip to the movie theater is supposed to serve as an escape from daily life, where people can go to relax and enjoy an interesting story. However, many issues surround the process of booking movie tickets today. The new movie ticket booking website is confronting the source of these issues by offering users an easy-to-use system that unifies the various booking services into one. Customers will be able to use this website for all their movie booking expenditures from a variety of nearby theaters. This will allow users to quickly find and reserve showings that are suitable without navigating through multiple websites. The unified approach will also eliminate booking errors such as double-booking. This option for customers could provide an increase in movie theater business, at a time when it is needed most. In order to ensure flawless interaction, some of the features have been reduced to accommodate seamless data management.

Glossary of Terms

Classics Theater application: System used to store user information, movie/theater information, and where users can manage reservations, and process payments.

Classics Theater at home: Website designed for users to access theaterbook from any web browser.

Classics Theater kiosk: Kiosk station inside movie theater where users can print tickets, make a new reservation, and even pay using cash.

System Requirements

Functional Requirements

No.	Priority Weight	Description
REQ-1	High	Web service should include movie listings from multiple different theaters in the area

REQ-2	High	Users should be able to book tickets from a variety of theaters
REQ-3	High	Users should be able to book multiple tickets
REQ-4	High	Users should be able to create TheaterBook account
REQ-5	High	Theaters should be able to indicate the capacity of each showing
REQ-6	High	Booking should become unavailable once desired showing reaches capacity
REQ-7	Med	Users should be able to access TheaterBook from inside the theaters
REQ-8	Med	Users should be able to view theater information and get directions
REQ-9	High	Theaters should be able to update their list of showings
REQ-10	High	All account, booking, showings, and theater information should be stored in a database

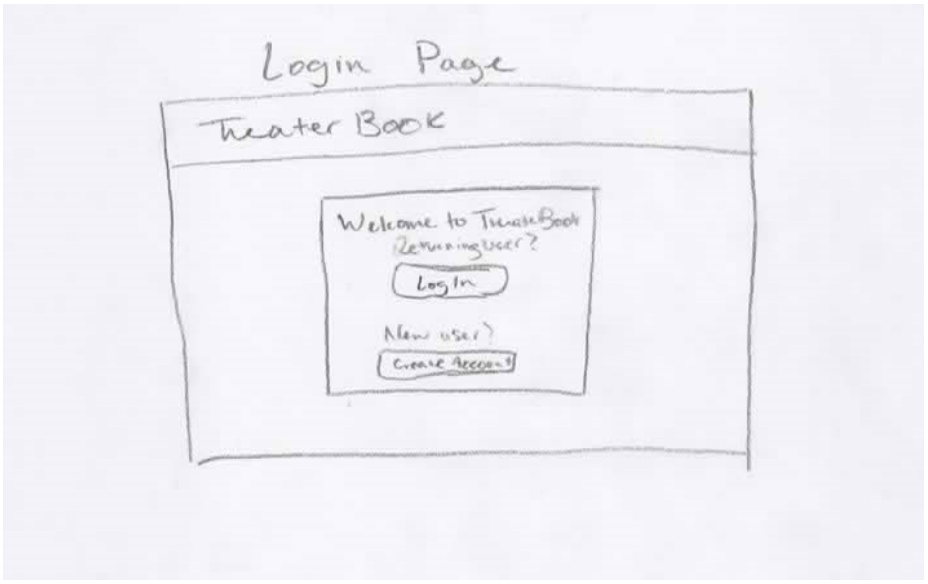
Nonfunctional Requirements

Usability	#1 High	The usability of this application should be of top priority to provide users a satisfactory experience
Functionality	#2 High	The application should function as intended with the requirements above

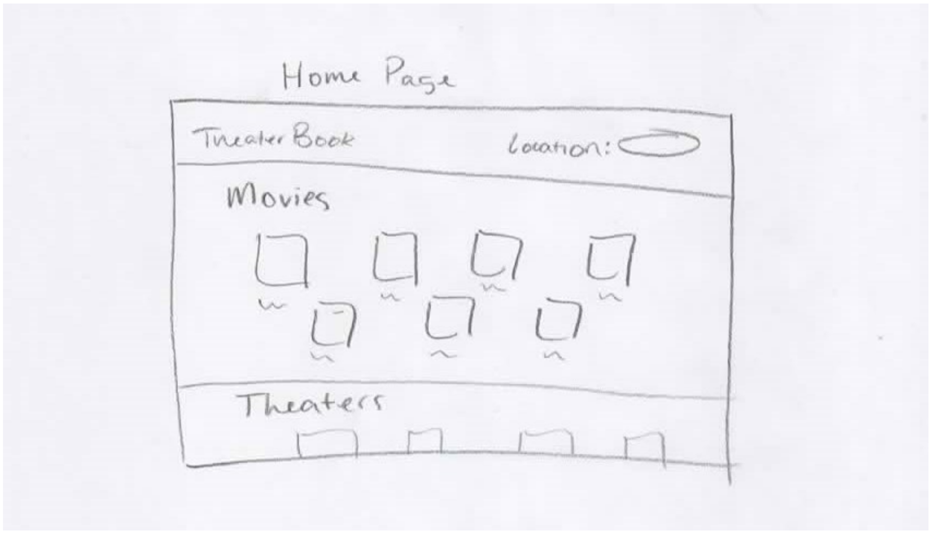
Reliability	#3 High	To build rapport with users, the application should be rid of any potential errors with a high level of accuracy in each process
Performance	#4 Med	Speed and efficiency should be a high priority, much of which will be coupled with the high priority of usability requirement
Supportability	#5 Med	The service should be flexible to accommodate various user needs and be able to provide support to users at any time

User Interface Requirements

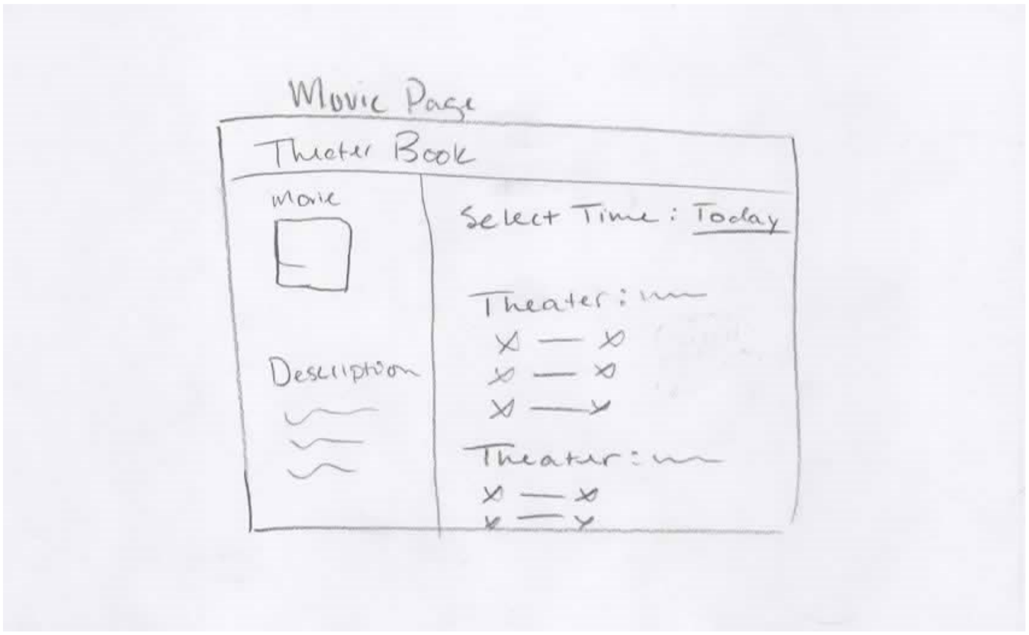
Users will be greeted to the webpage with the ability to log on or create a new account.



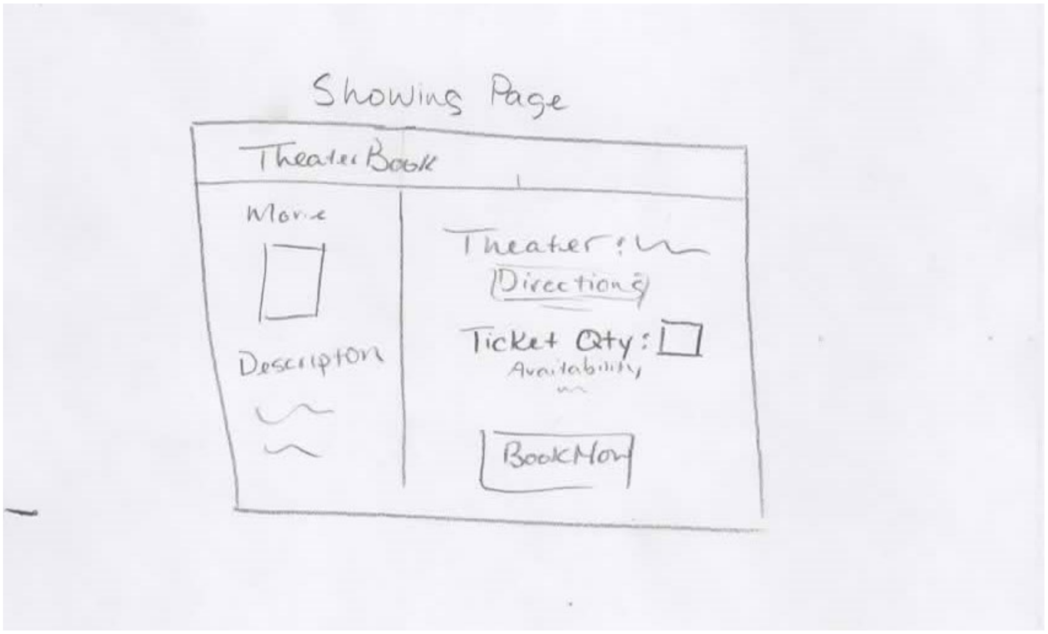
Following login process, users will be provided the various movies available at theaters near them, as well as the option to change their location at any time. This page will also give users the ability to browse specific theaters instead.



Once the user selects the movie they wish to view, they will then be brought to a page that shows a description of the movie as well as all of the showings available at each nearby theater.



Once they select a specific showing, users will be brought to a page that will go over all of the information they have selected, allow users to get directions to the theater, select the number of tickets desired, and book their tickets.



Functional Requirements Specification

Stakeholders

- Customers
- Theater staff
- Classics Theater Administration

Actors and Goals

Primary Actors

- Customer: This actor can browse showing information, theater information and directions, and book tickets while available.
- Theater staff: This actor can modify their showing listings and update capacity of each showing.

Secondary Actors

- Admin: Add or remove theater from web service, view/update user accounts.

- System: Shows actors the movie listing content, theater information, saves booking information, and limits booking based on theater capacity.

Use Cases

Admin

- Add/remove theater: Add/remove theater from showing listings.
- Add/remove theater employee: Add/remove theater employee with access to web services.
- Update account: Update account details.
- View account details: view the details/booking history of any account.
- Login/logout: Log in and out of administration account.

Customer

- Login/logout: Login and logout from customer account.
- View available listings: view current listings of movie showings.
- View showing details: view different times and movie information.
- Select timeframe: select showing timeframe that works best.
- Book showing: book showing for number of tickets desired (while available)
- Cancel booking: cancel previous showing booking.
- View account: view account details
- Modify account information: modify general information associated with account.

Employee

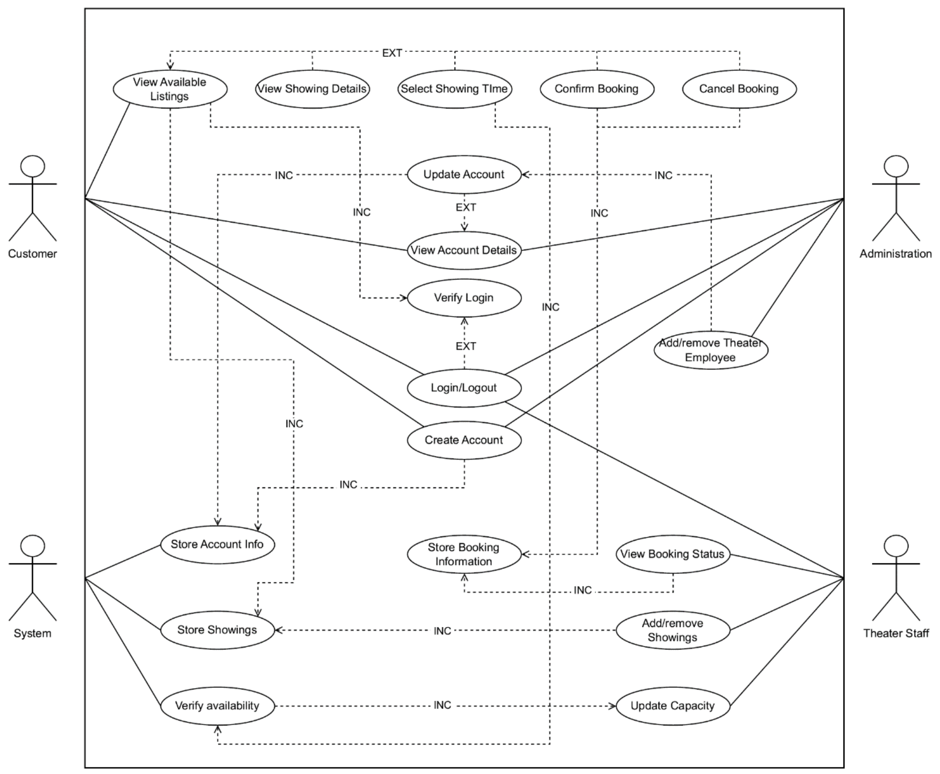
- Login/logout: log in and out from theater employee account.
- Add/remove showings: add and remove showings as new movies become available.
- Update capacity: update the capacity available for each showing.

- View showing booking status: bookings associated with each showing.

System

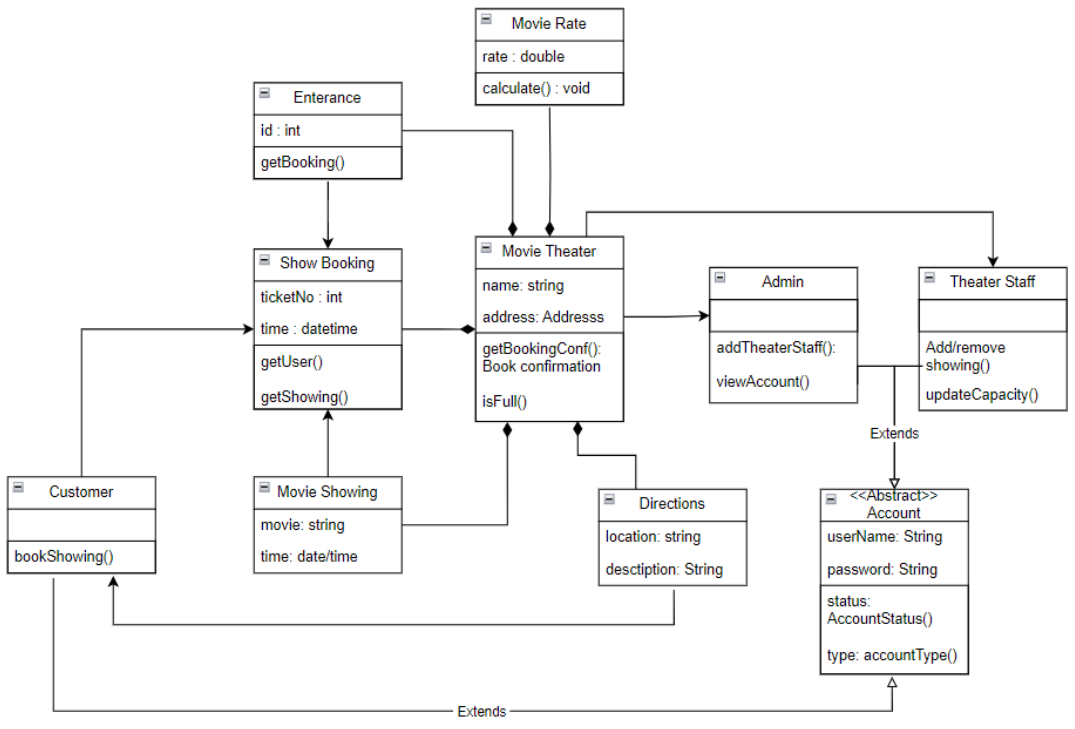
- Showings listing: provide users list of all current showings.
- Store booking information: save each booking into database.
- Verify availability: only allow for new bookings while seats are available.

Use Case Diagram



Class Diagram

This class diagram shows the various types of relationships and capabilities of the described movie booking system. As the diagram shows, most components stem off of the movie theater class. Items such as movie rate, show booking, movie showings, and directions all are a aggregation of the movie theater class. However, movie showing also answers to show booking as does directions when requested by user. Entrance to the showing is determined by the booking confirmation. There are three classes that also extend the account class: admin, theater staff, and customer.



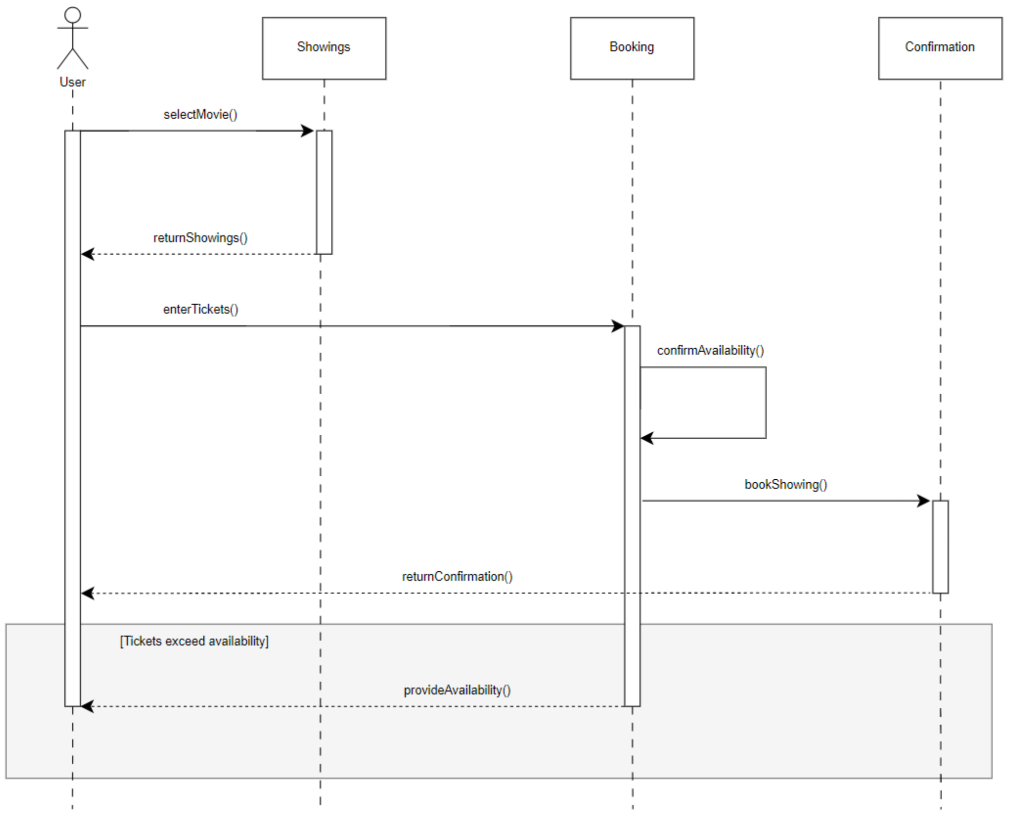
System Sequence Diagrams

Movie Showing Booking

Actor: User

Object: Showing, Booking, Confirmation

1. User selects movie
2. The system provides list of showings
3. Provide user list of showings
4. System prompts ticket number and confirm booking
5. User enters number of tickets and clicks book tickets
6. System verifies availability and provides confirmation
7. If ticket number is not available, system returns to booking page and indicates number available

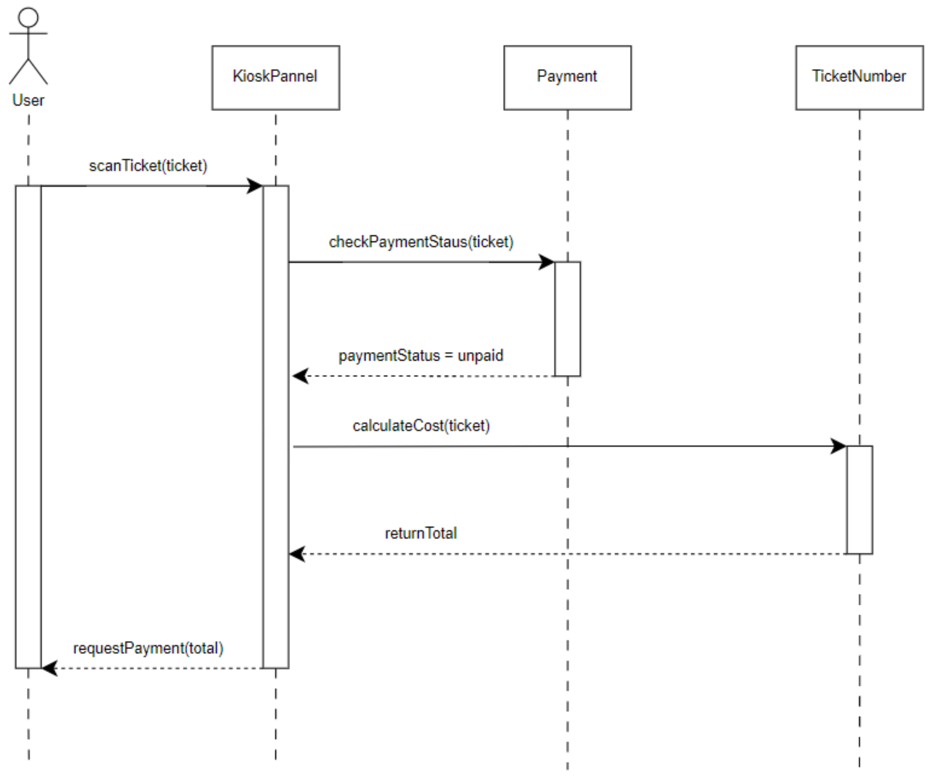


Payment Verification

Actor: User

Object: KioskPannel, Payment, TicketNumber

1. User scan ticket
2. Kiosk checks if ticket has been paid.
3. Kiosk reviews total tickets
4. TicketNumber determines total.
5. Kiosk requests payment based on the total.



Activity Diagram

Movie Showing Booking

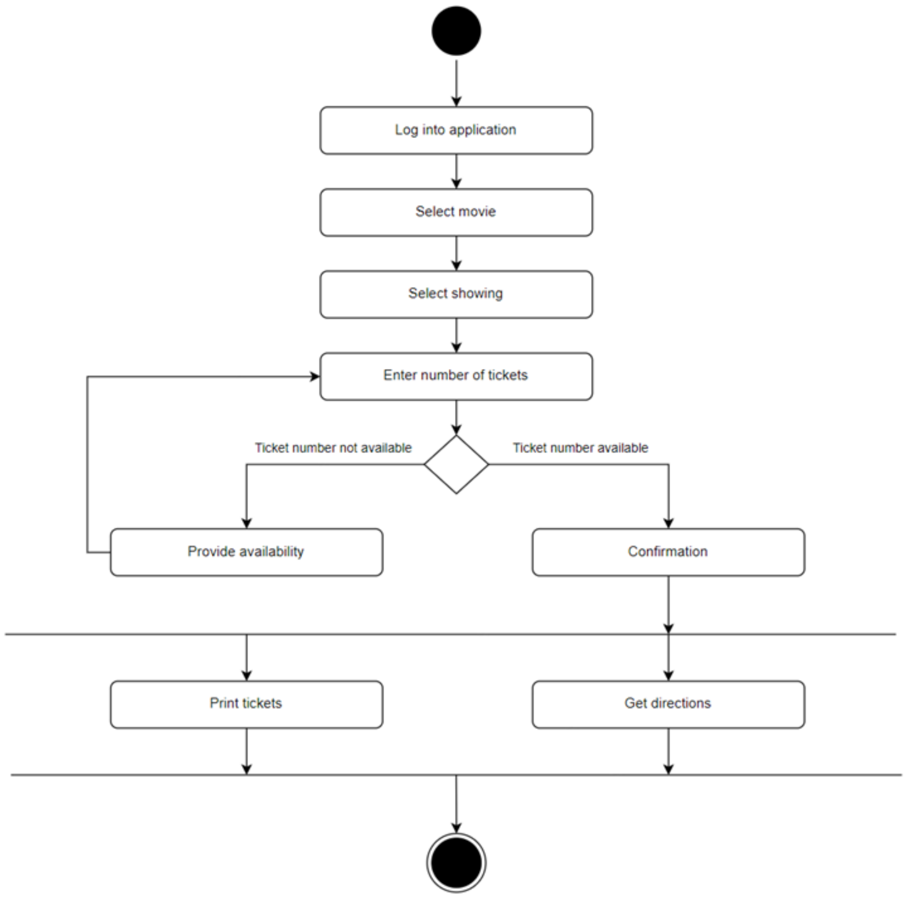
States:

Initial State: User logs into application

Final State: User books showing

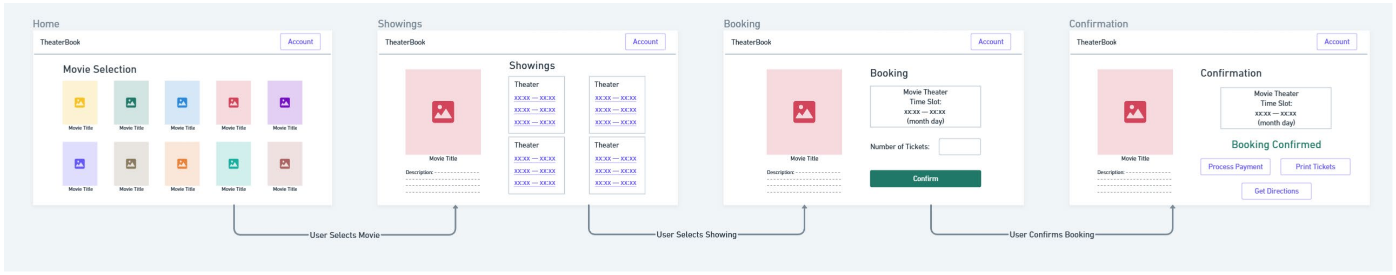
Actions:

The user logs into application and selects a movie they would like to see showings for. The available showings for the movie appear on the next screen, where users can choose one that works best for them. The user is then prompted for ticket numbers and to confirm booking. If ticket number is not available, system returns to booking page and indicates availability. If ticket number is available, user receives booking confirmation, where they can print tickets and get directions.

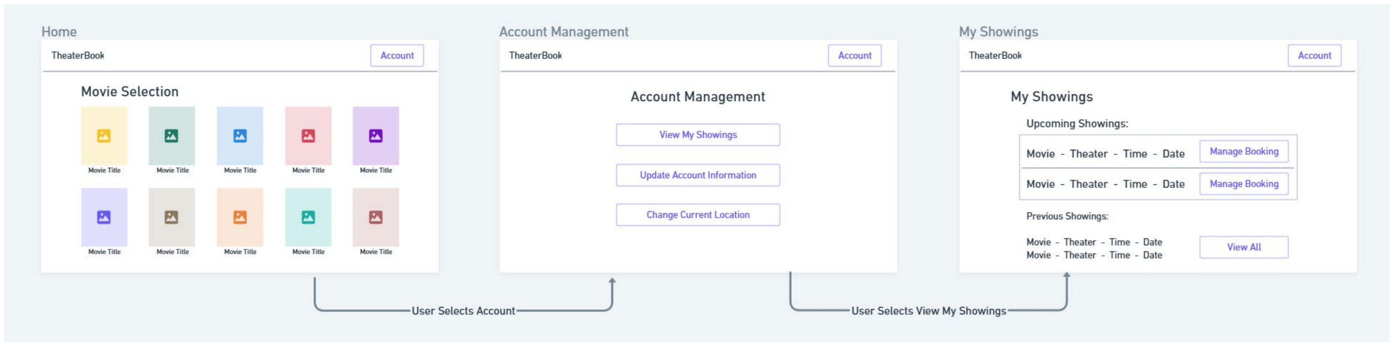


User Interface Specifications

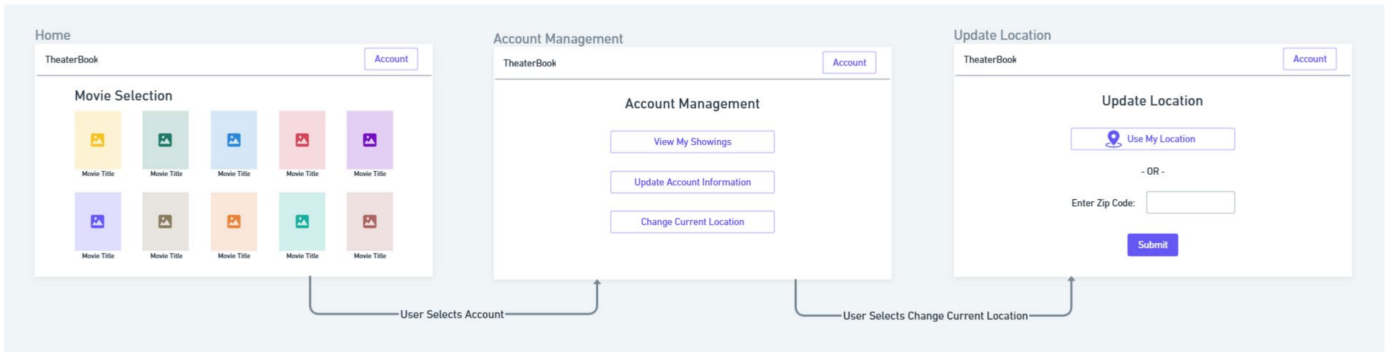
Use Case: User Book Showing



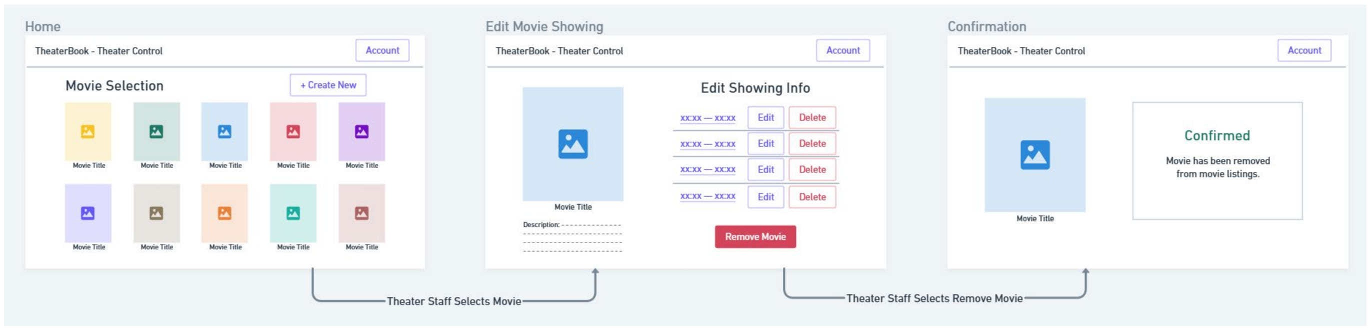
Use Case: User View Booked Showings



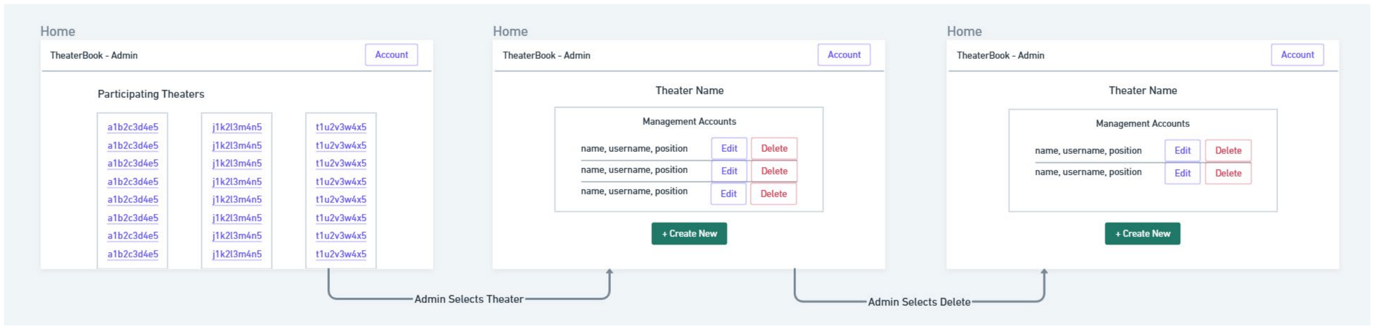
Use Case: Update Current Location



Use Case: Theater Staff Remove Movie from Showings



Use Case: Admin Remove Theater Staff



User Effort Estimation

Usage Scenario	Navigation	Clicks	Keys
User selects showing to book	Homepage, showings page, overview, confirmation	3	<10
User views booking history	Homepage, account, my showings	2	0
User updates current location	Homepage, account, change current location	3	<10
Theater staff removes movie from listings	Homepage, showings, confirmation	3	0
Admin removes theater staff	Homepage, theaters, theater page	2	0

Plan of Work

Completed:

Prepare software for project: I have downloaded and/or updated the software components that will comprise this system. I have also conducted research on how to best use these systems in tandem to ensure maximum efficiency between the development server and database.

Develop user options: (View movie showings, details, book showings, view up-coming bookings, manage reservations)

Outline all various processes of the system: The primary processes of the system have been outlined and accounted for. Subprocesses have been outlined but not yet officially published.

Develop and run test project.

Create login/registration page for users and theater staff.

Develop staff user options (update theater info, add/remove showings, modify capacity)

Integrate payment option.

Incorporate staff features into database and website.

Refine staff features on front end.

Incorporate customer features into database and website.

Refine customer features on front end.

Refine front end entirely.

Run tests and ensure everything in the system is working correctly.