Movie Ticket Booking System

Nick Merida

SP23: SYSTEM IMPLEMENTATION: 26578

Shawn Dai

3/9/2023

Table of Contents

Problem statement

As someone who wants to watch a movie.

I'm trying to book a movie ticket online.

But there is currently no system to do so online

because you can only do it in person.

This is very inconvenient.

Project Proposal

Objectives of the system

A Movie Ticket Booking System is used to allow customers to reserve a spot for a movie for the future. For the employers it allows them to see who and how many people booked for a specific movie. Employees would also be able to efficiently see how many people have booked in a movie allowing the proper number of reserved spots to be calculated. With this system it would allow people to more efficiently buy a movie ticket because they don't need to go in person which could lessen lines, require less labor for the employees, and not go to a movie and it be filled already because they can see if the booking is full or not.

In terms of price for a project like this It will vary depending on the end solution but for The current setup the bare minimum is required to be able to run the webserver, and the SQL server, this depends on how many people will be connecting to it.

Expected System requirements and Specs

The users for a movie ticket booking system would most likely be the manger with some sort of system training allowing them to operate the system, customer, and employees reserving spots for customers. For system requirements hardware can be any windows machine. The only network requirement is to have access to the internet. The software I'm going to be using is VS Code with python, my database is MySQL, and my front end is HTML with some CSS. Other programs required would be any addons or plugins needed to link the three together to work properly.

Weekly Plan

Week 1 To start out I will setup all of the environments I need.

Week 2 The next step I'm going to take is linking the front back and database together.

Week 3 After this I will start by setting up all of the basics, I need like the tables I need for the database.

Week 4-10 Then I will start writing the code to tie the python code to allow it to use the data I implemented into MySQL. During all of this process I will be checking for any bugs or glitches with the program.

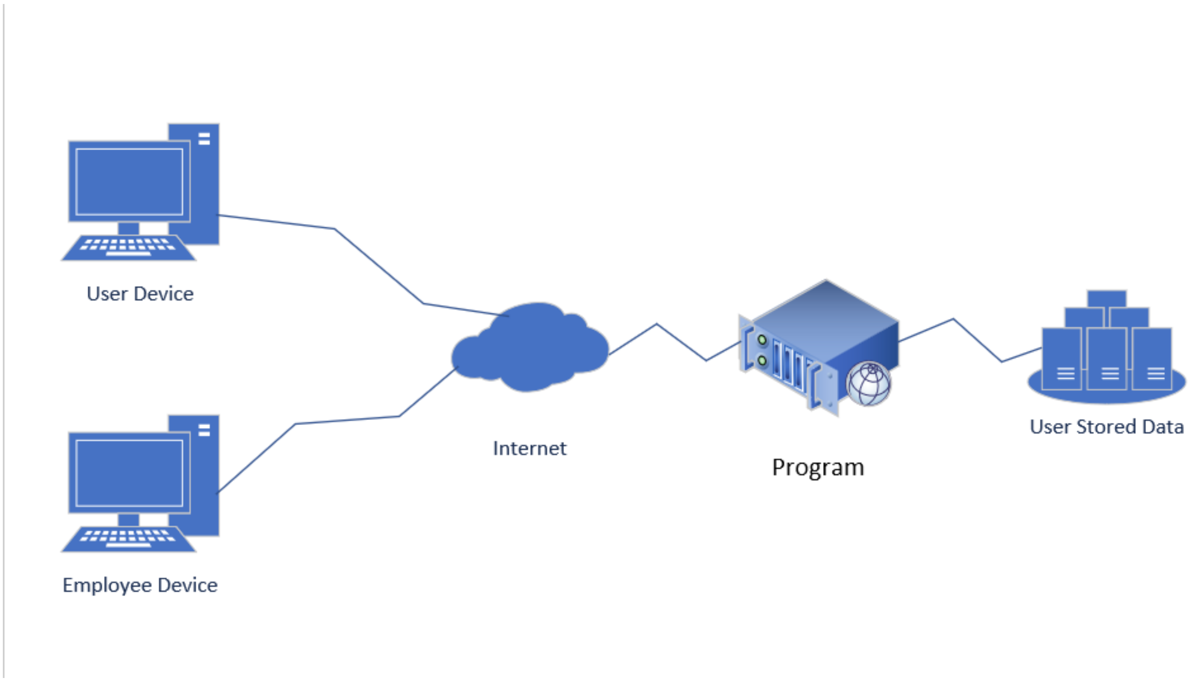
Week 10 Test system for Midterm.

Week 11 Finally, I will work on the front end building the gui and art for the page. Last, I will have my program tested and make any changes needed.

Week 12-13 add features mentioned in customer feedback

Week 14-15 Write test case for features and add polish to system. Then record the demo for the final program.

Glossary of Terms



Program: Front end and back-end code to get the system running. Front end- html, css

Back-end Python. This is run off of local server

User stored data: This is the MYSQl server ran locally allowing employees to access user data or the program to read data sent or received from the web-based site.

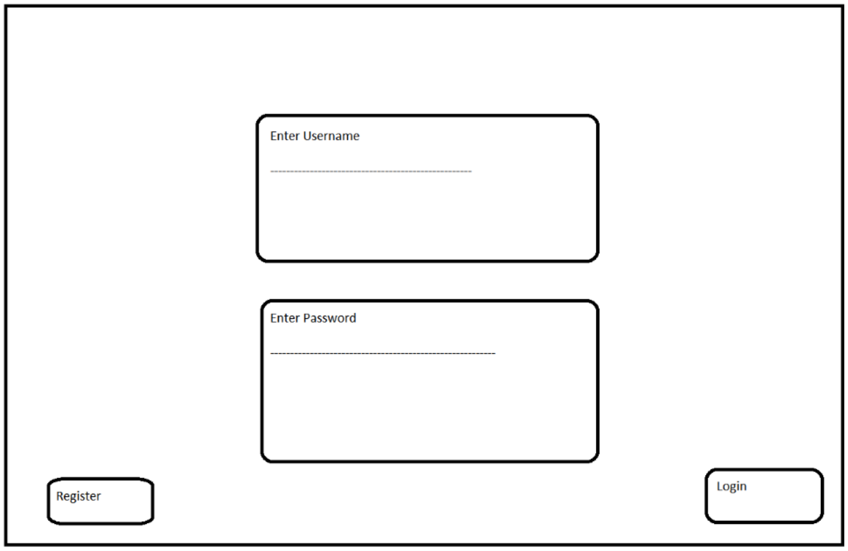
Employee Device: any device an employee logs into allowing changes to movie schedules or other scheduling need.

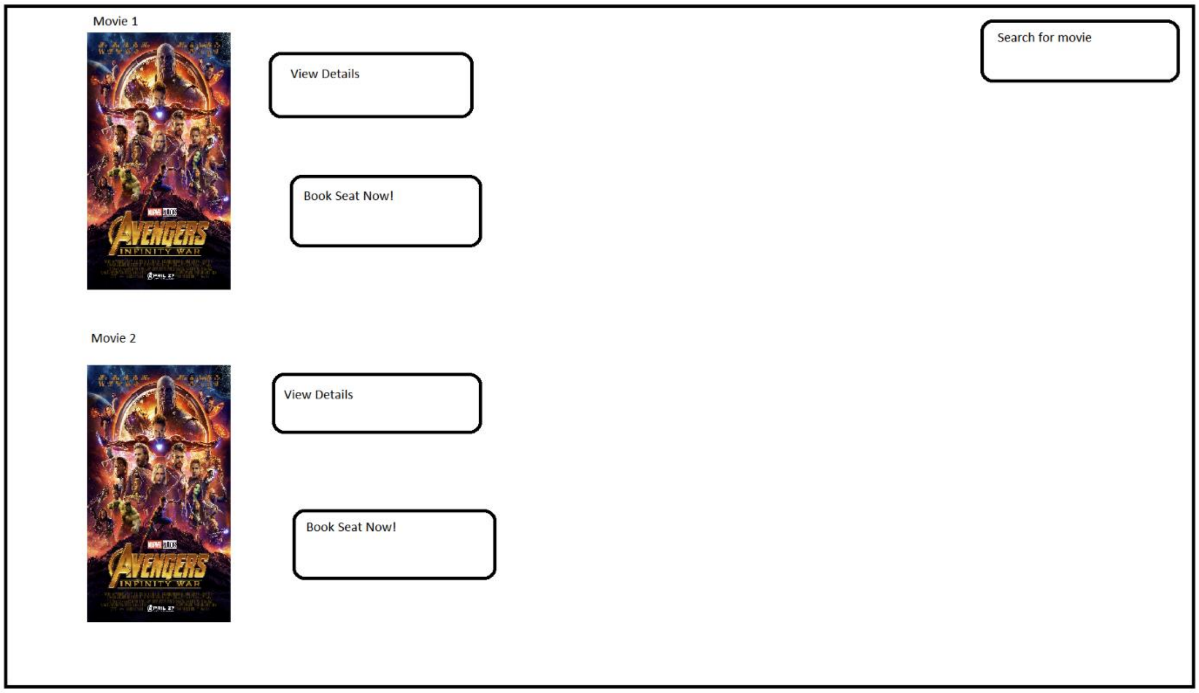
User Device: users' device that can access the internet allowing them to login, register, and book movies.

System Requirements

REQ-1	High	Can handle multiple users using the system at once
REQ-2	**High**	Allow employees to edit movies being played
REQ-3	**High**	Certain employees have raised access compared to customers to allow edits to be made
REQ-4	**High**	User can book tickets to a movie
REQ-5	**Low**	Appealing UI
REQ-6	**Low**	Register a specific seat in the theater.
REQ-7	**Medium**	Specialized employee accounts for different permissions in the system.
REQ-8	**High**	Functionality
REQ-9	**Medium**	**Usability**
REQ-10	**Medium**	Reliability
REQ-11	**Low**	Performance
REQ-12	**Low**	Supportability
REQ-13	**High**	Login screen where employees or users can login.
REQ-14	**High**	Main screen to select what movie to book.
REQ-15	**Low**	Interface for selecting a specific seat.

Very Rough Sketch of UI


Enter Username
Enter Password
Register
Login


Movie 1
View Details
Book Seat Now!
Search for movie
Movie 2
View Details
Book Seat Now!
AVENGERS
INFINITY WAR
AVENGERS
INFINITY WAR

Plan of work

~~**Week 1** To start out I will setup all of the environments I need.~~

~~**Week 2** The next step I'm going to take is linking the front back and database together.~~

Week 3 After this I will start by setting up all of the basics, I need like the tables I need for the database.

Week 4-10 Then I will start writing the code to tie the python code to allow it to use the data I implemented into MySQL. During all of this process I will be checking for any bugs or glitches with the program.

Week 10 Test system for Midterm.

Week 11 Finally, I will work on the front end building the gui and art for the page. Last, I will have my program tested and make any changes needed.

Week 12-13 add features mentioned in customer feedback

Week 14-15 Write test case for features and add polish to system. Then record the demo for the final program.

Stakeholders

Admins

Customers

Employees

Mangers

A company wanting to play their movie at this theater

The owner

Actors & Goals

Primary Actors

Customer: Can book a movie online or in person, login/register for an account and pay, View capacity.

Employee: Assist Customer, accept payment and hand out tickets to customer to allow customer to purchase a movie ticket in person.

Secondary Actors

Admin: Add, Edit, remove movies being shown, edit bookings/ capacity, update accounts.

System: The system takes care of generating the ticket for a seat and checking the capacity.

Use Cases

Customer (16)

- Pay for booking: pay for a booking online with credit or debit card (2)
- Cash for booking: pay in person for booking with cash (2)
- Receive ticket: Receive the ticket from the employee (2)
- Debit/Credit Card: pay for the booking via debit or credit card (2)
- Go to the registered room: go to the room playing the specified movie that was purchased (2)
- Login/register: login/logout for current user account (2)
- View account: View current user account (2)
- Update Account: Update current user account (2)

Employee (12)

- Check Booking: Check booking made by Customer via the system (2)
- Scan ticket: scan customers ticket (2)
- Allow customer in: if ticket is in order allow customer into movies (2)
- Reject customer: turn away customer with false booking (2)

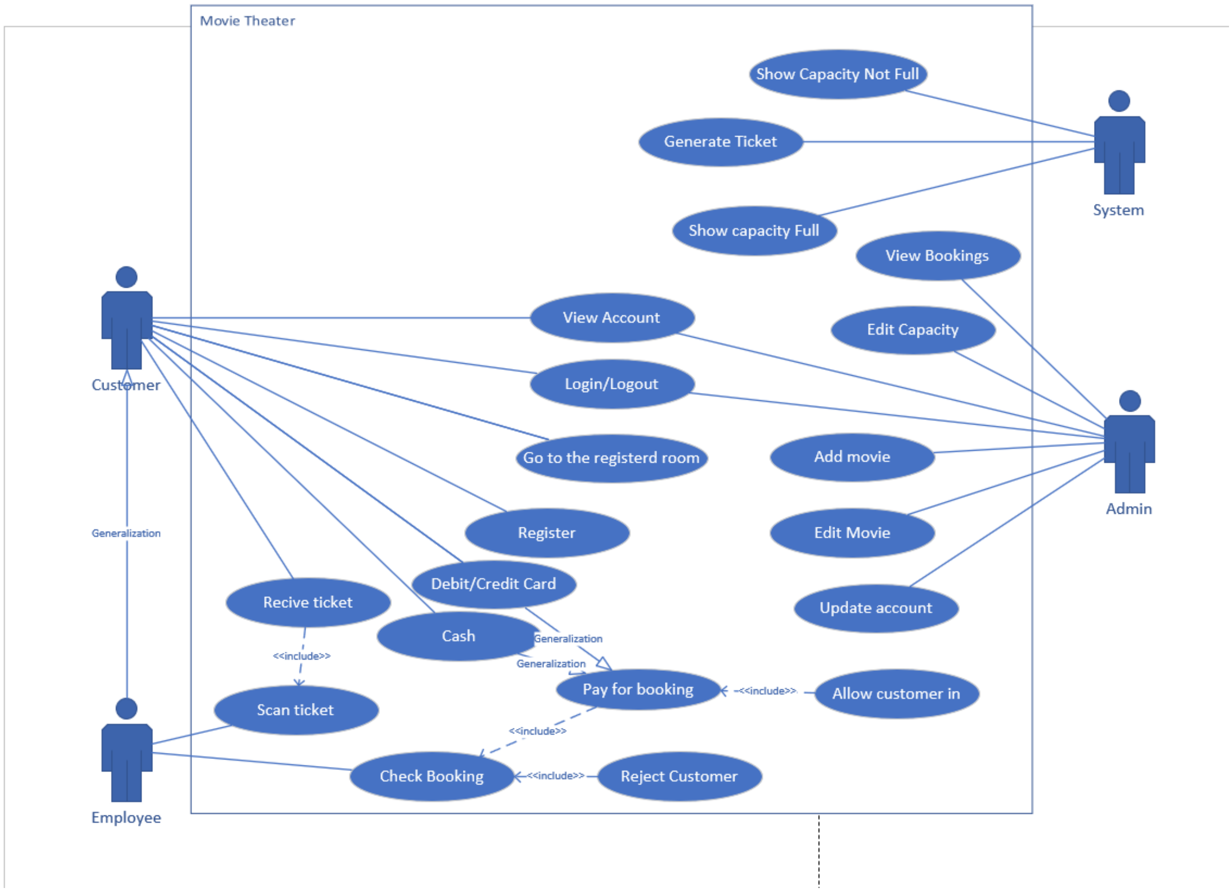
Admin (14)

- Add movie: To add a movie to booking website (2)
- Edit movie: To edit movie on booking website (2)
- Login/logout: login/logout of admin, user account (2)
- View account: view admin, user account (2)
- Update account: update admin, user account (2)
- Edit capacity: Edit capacity for certain theater rooms. (2)
- View bookings: view current theater room bookings (2)

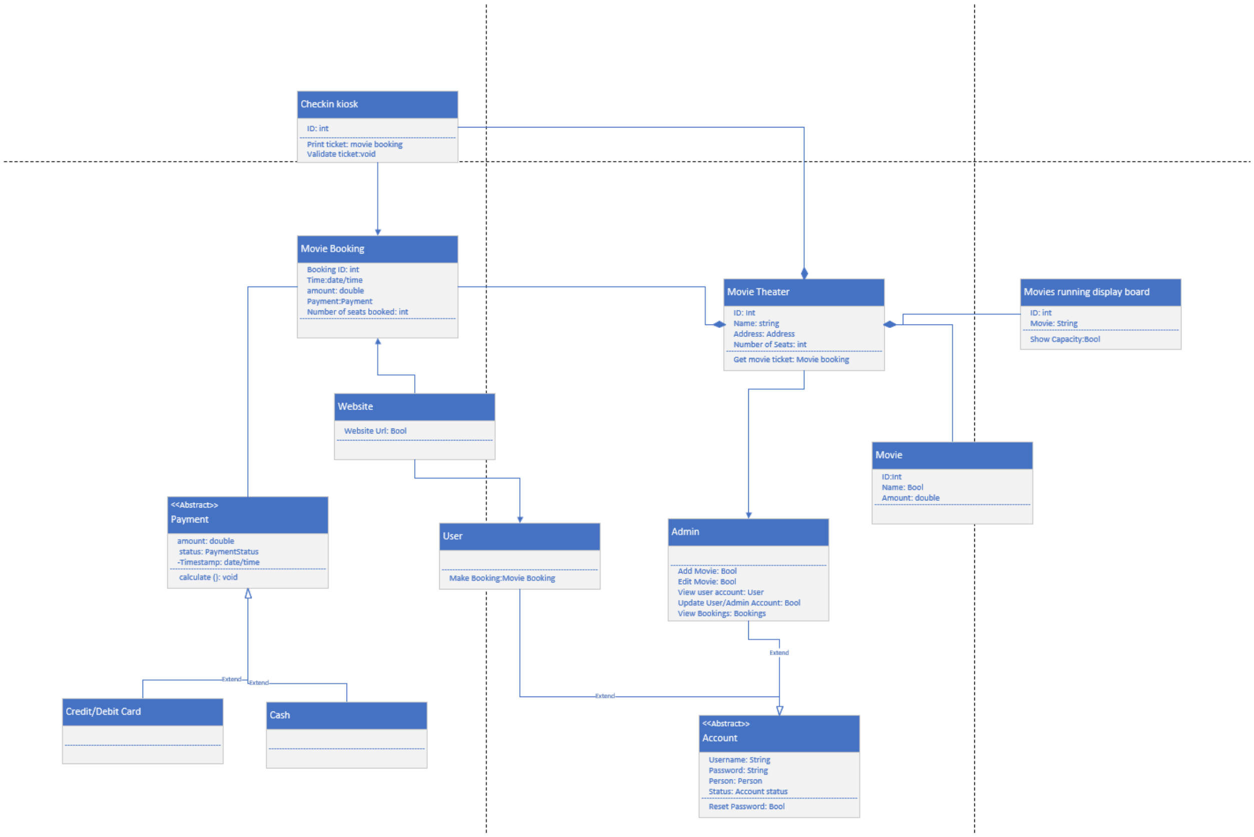
System (6)

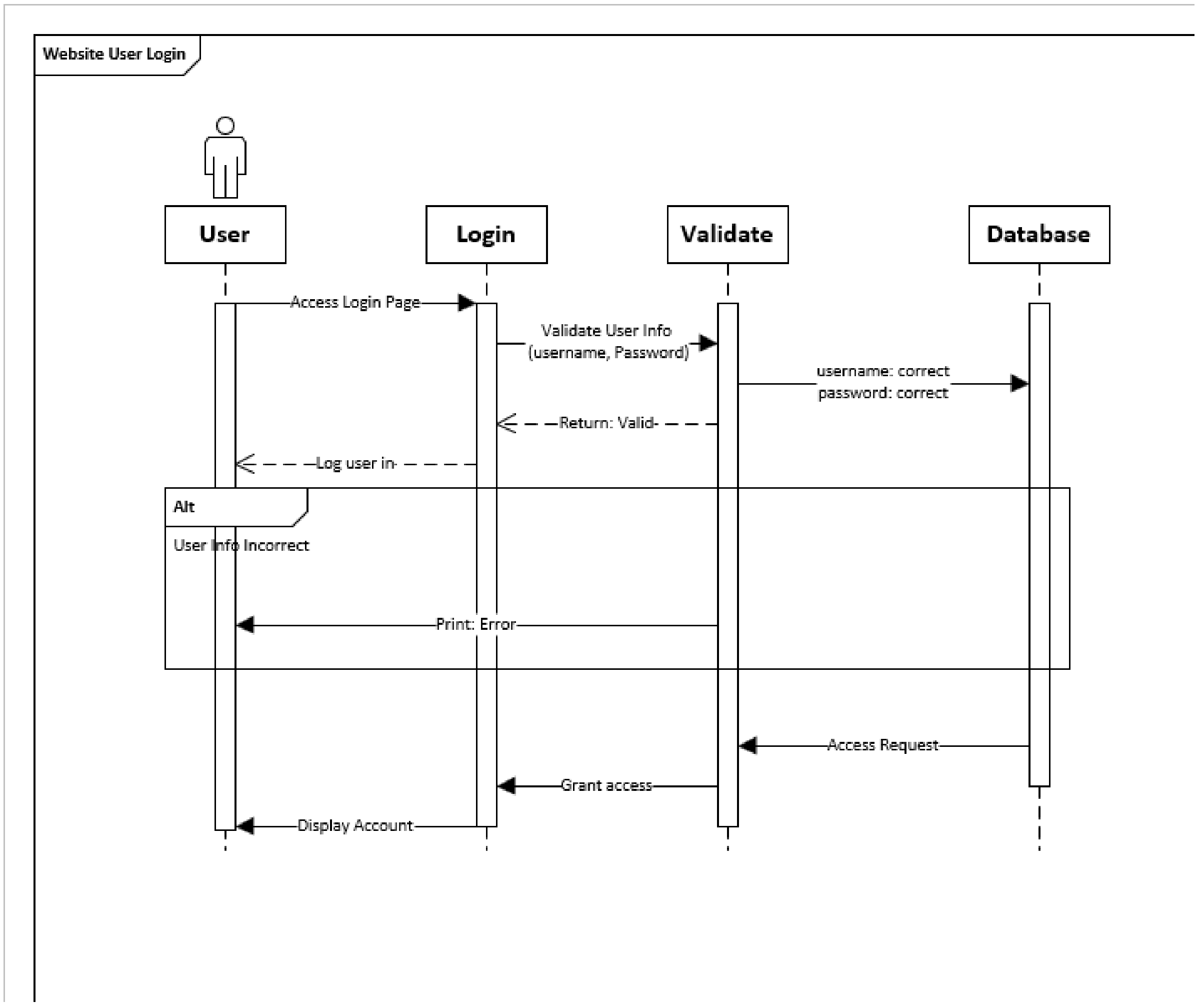
- Generate Ticket: generate physical ticket from information entered (2)
- Show Full: show capacity for a movie is full (2)
- Show available: show capacity for a movie is not full. (2)

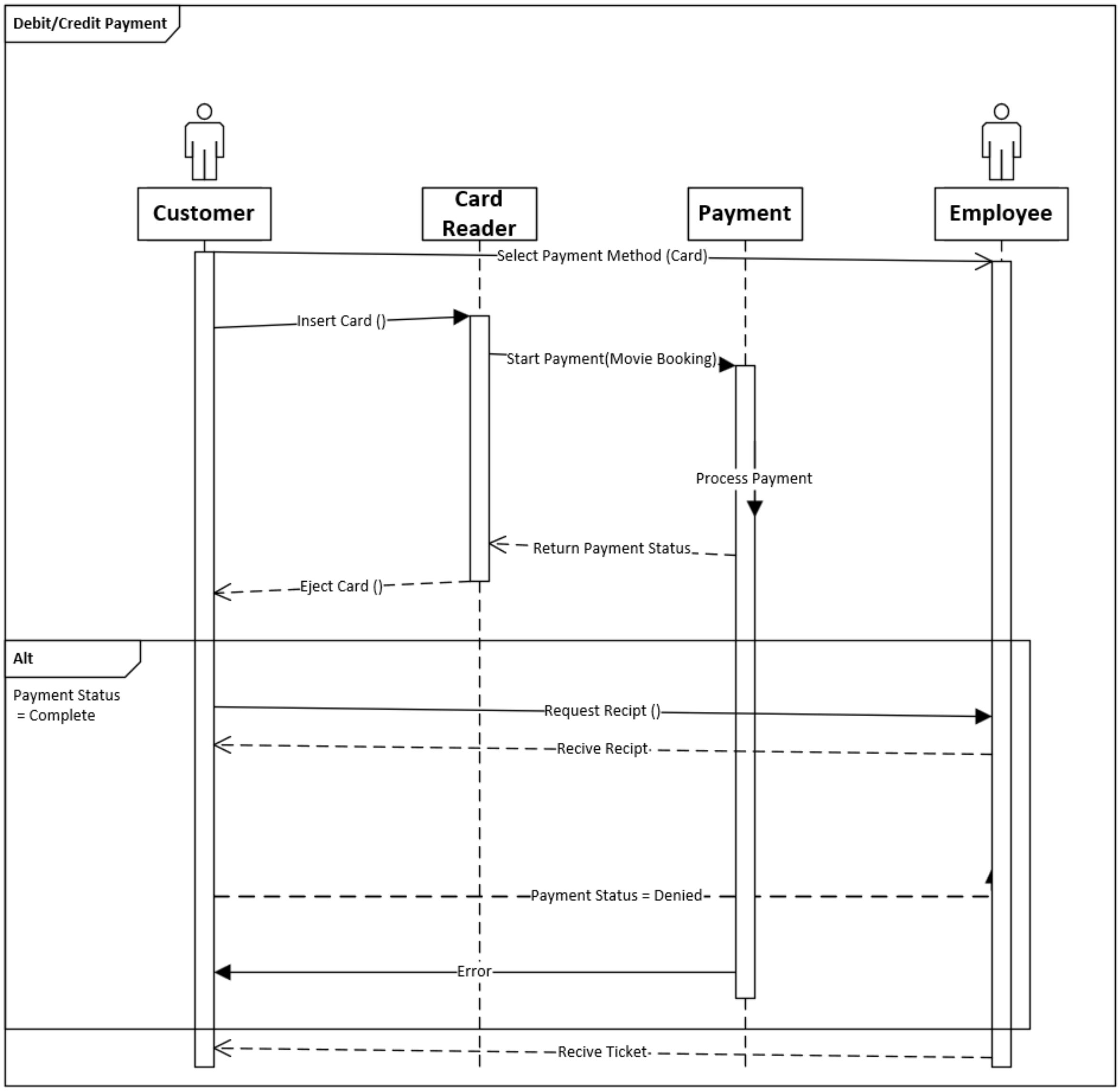
Use Case Diagram

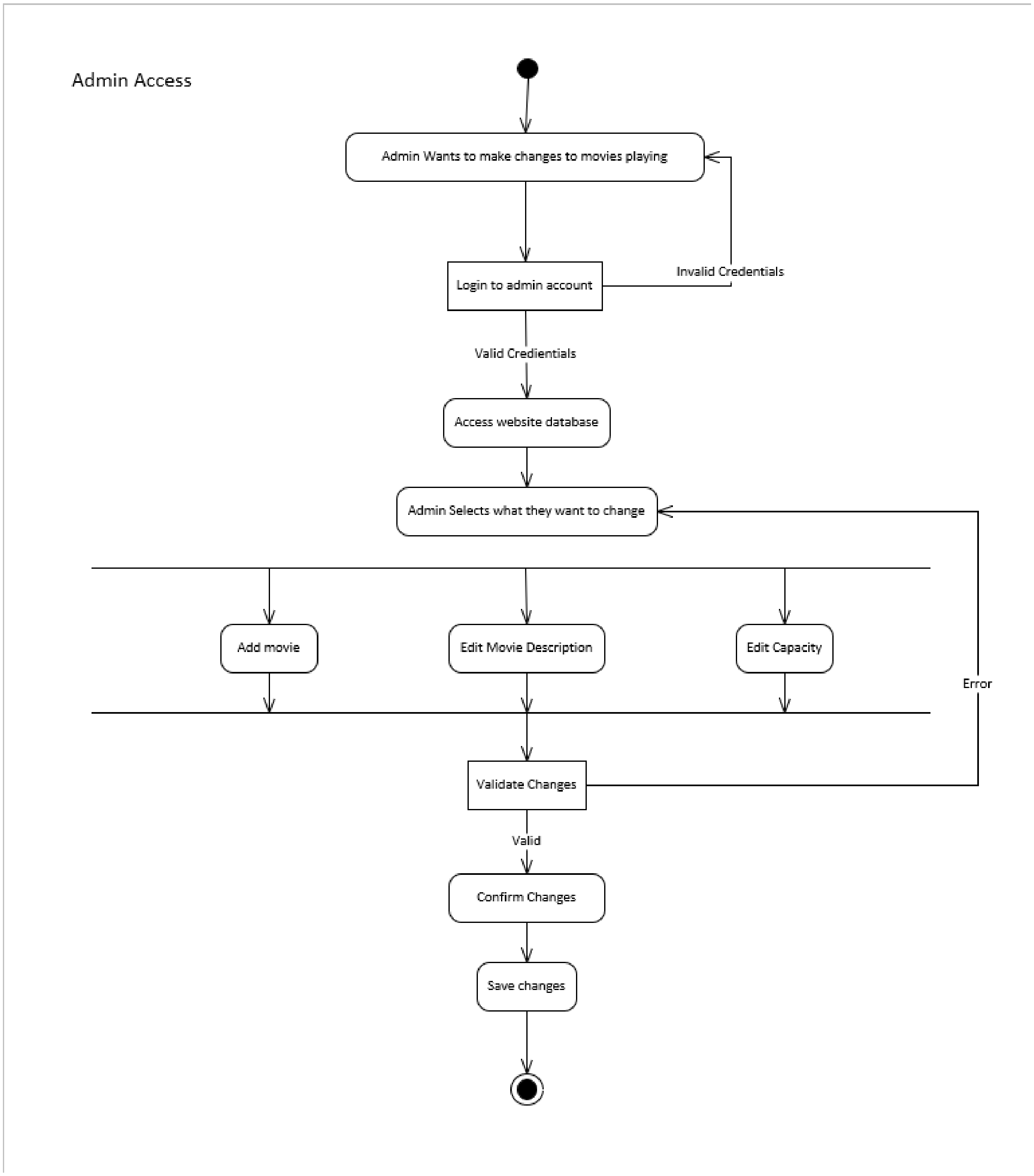


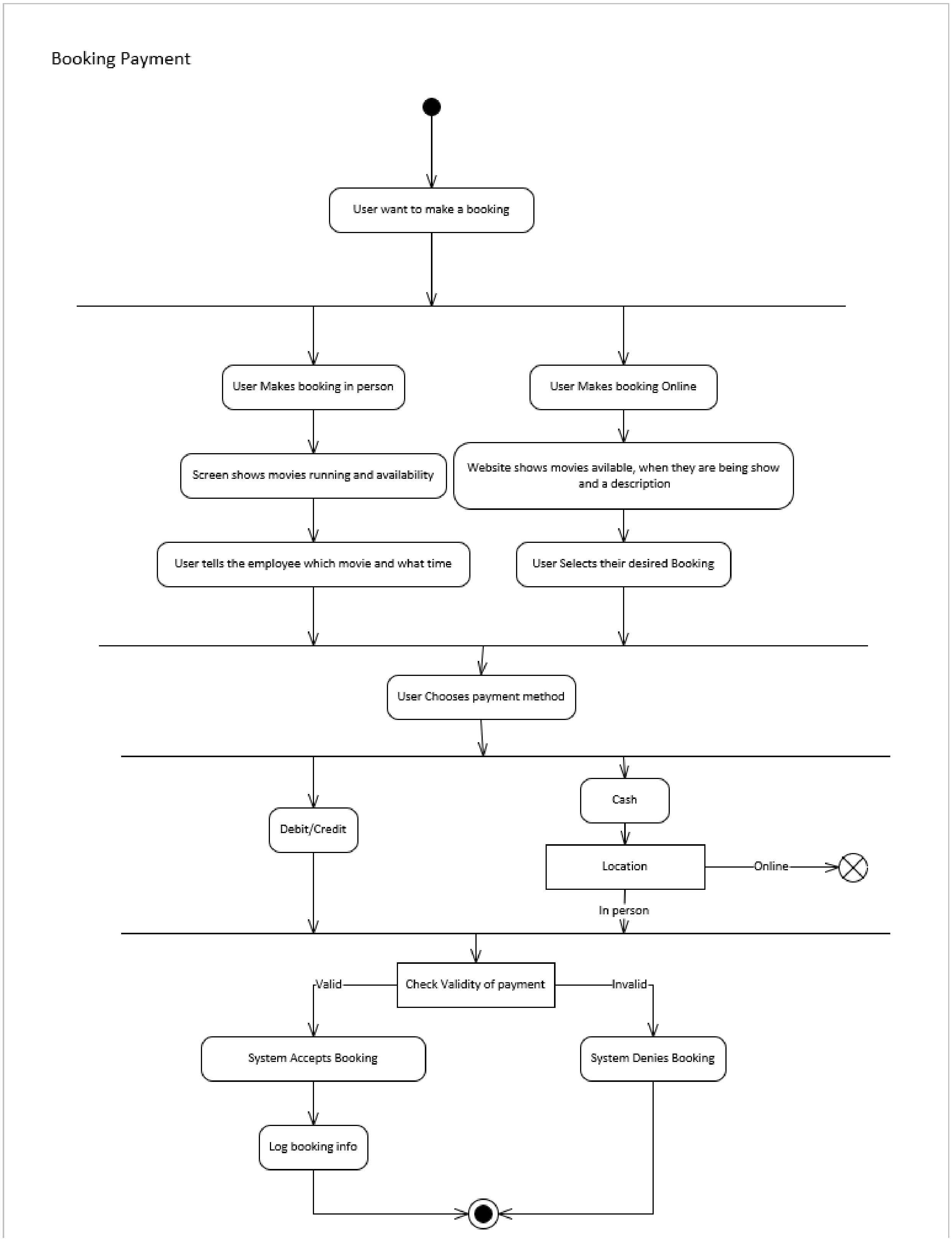
Class Diagram

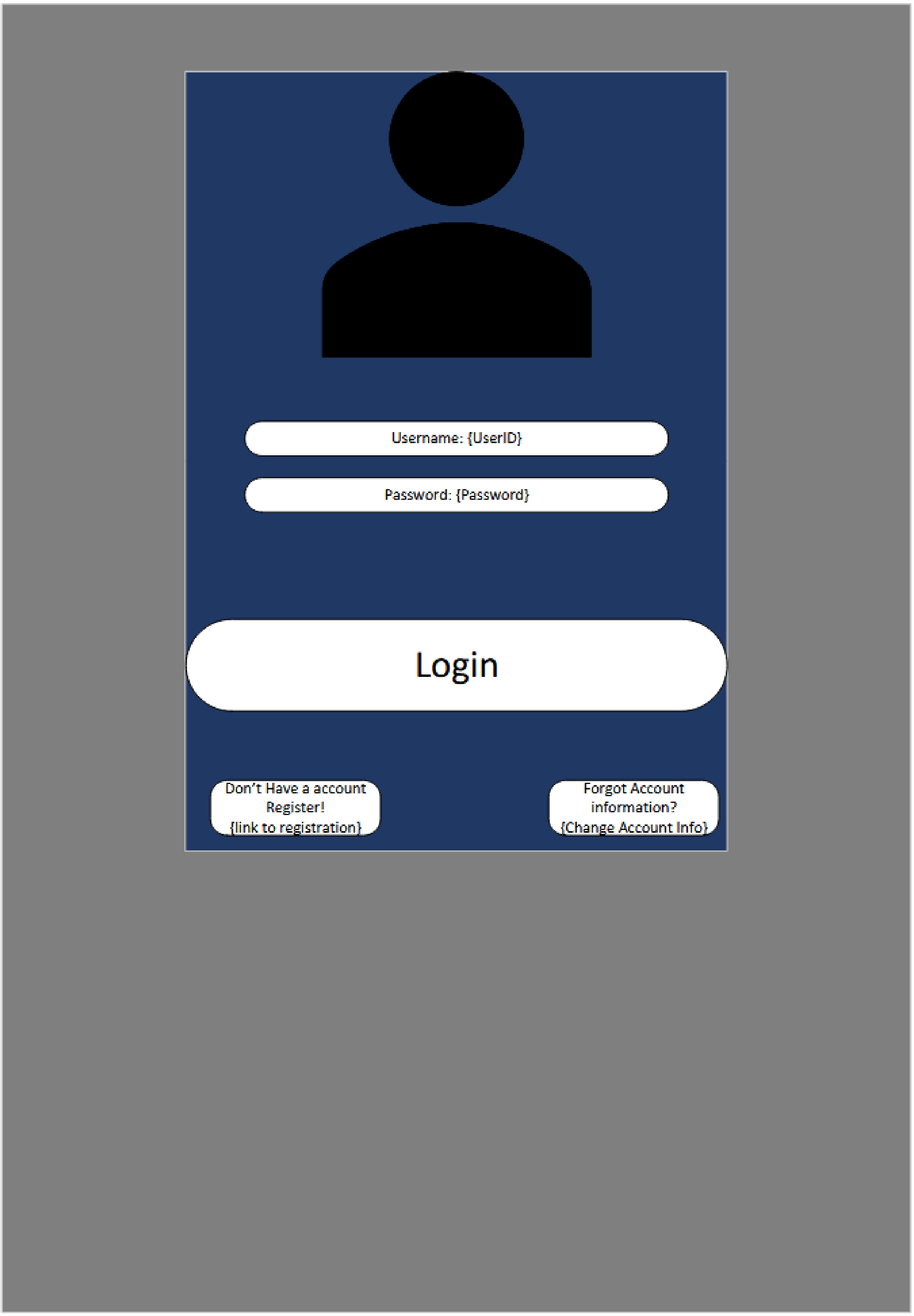


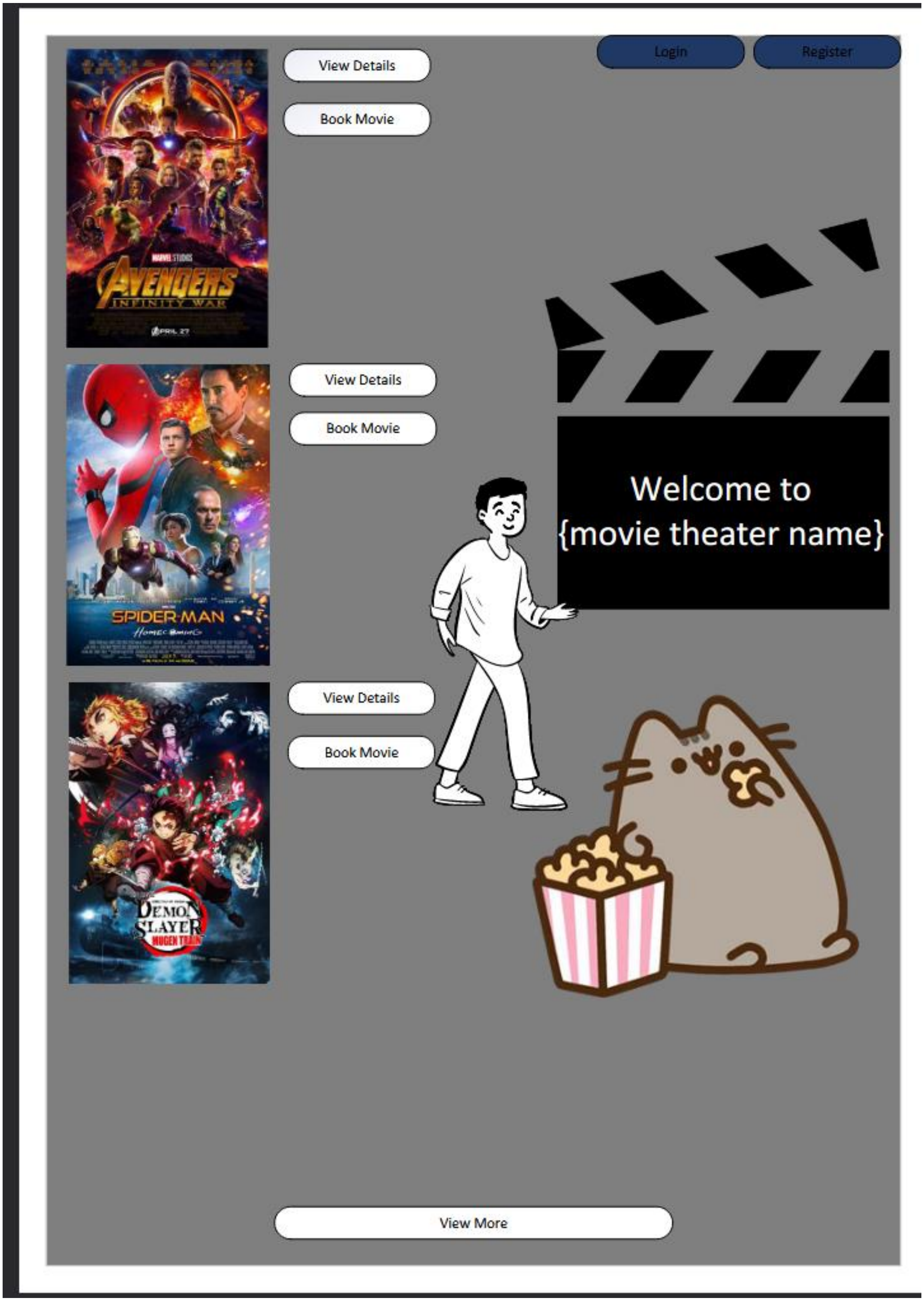

Website User Login
User
Login
Validate
Database
Access Login Page
Validate User Info
(username, Password)
username: correct
password: correct
Return: Valid
Log user in
Alt
User Info Incorrect
Print: Error
Access Request
Grant access
Display Account


Debit/Credit Payment
Customer
Card Reader
Payment
Employee
Select Payment Method (Card)
Insert Card ()
Start Payment(Movie Booking)
Process Payment
Return Payment Status
Eject Card ()
Alt
Payment Status = Complete
Request Recipt ()
Recive Recipt
Payment Status = Denied
Error
Recive Ticket

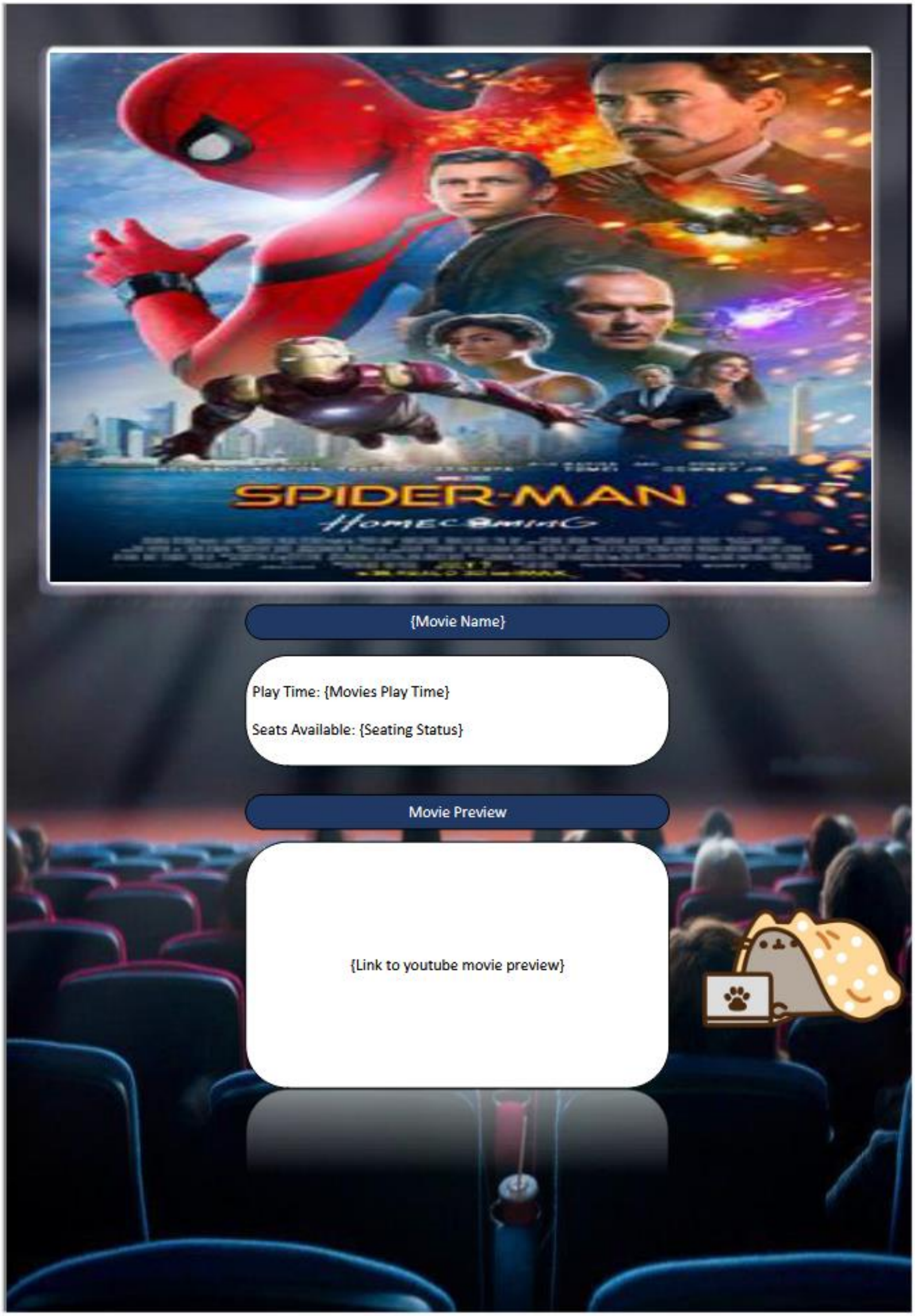

Admin Access
Admin Wants to make changes to movies playing
Login to admin account
Invalid Credentials
Valid Credientials
Access website database
Admin Selects what they want to change
Add movie
Edit Movie Description
Edit Capacity
Error
Validate Changes
Valid
Confirm Changes
Save changes


Booking Payment
User want to make a booking
User Makes booking in person
User Makes booking Online
Screen shows movies running and availability
Website shows movies avilable, when they are being show and a description
User tells the employee which movie and what time
User Selects their desired Booking
User Chooses payment method
Debit/Credit
Cash
Location
Online
In person
Check Validity of payment
Valid
Invalid
System Accepts Booking
System Denies Booking
Log booking info


Username: {UserID}
Password: {Password}
Login
Don't Have a account Register! {link to registration}
Forgot Account information? {Change Account Info}



Welcome to
{movie theater name}



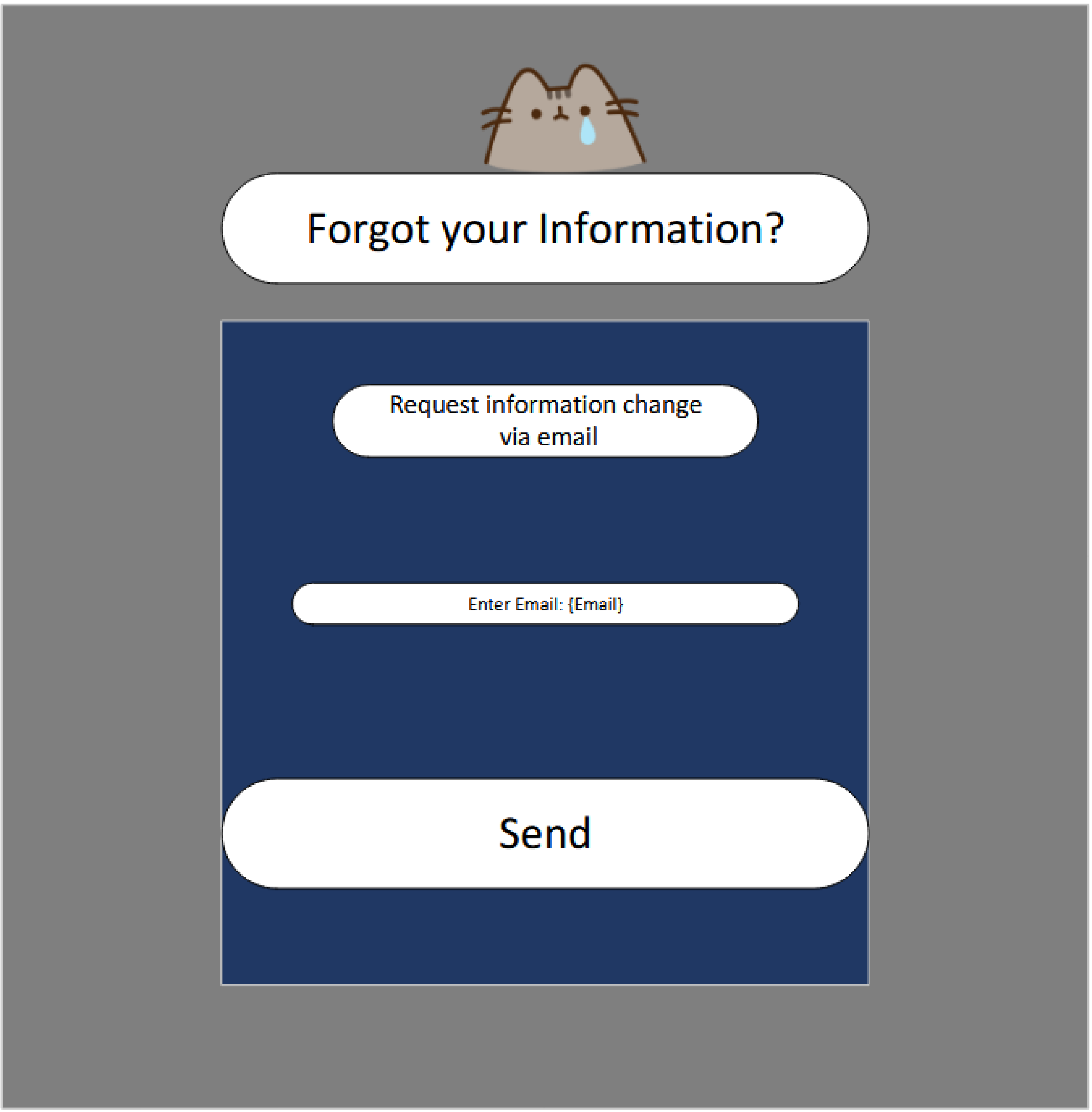
{Movie Name}

Play Time: {Movies Play Time}

Seats Available: {Seating Status}

Movie Preview

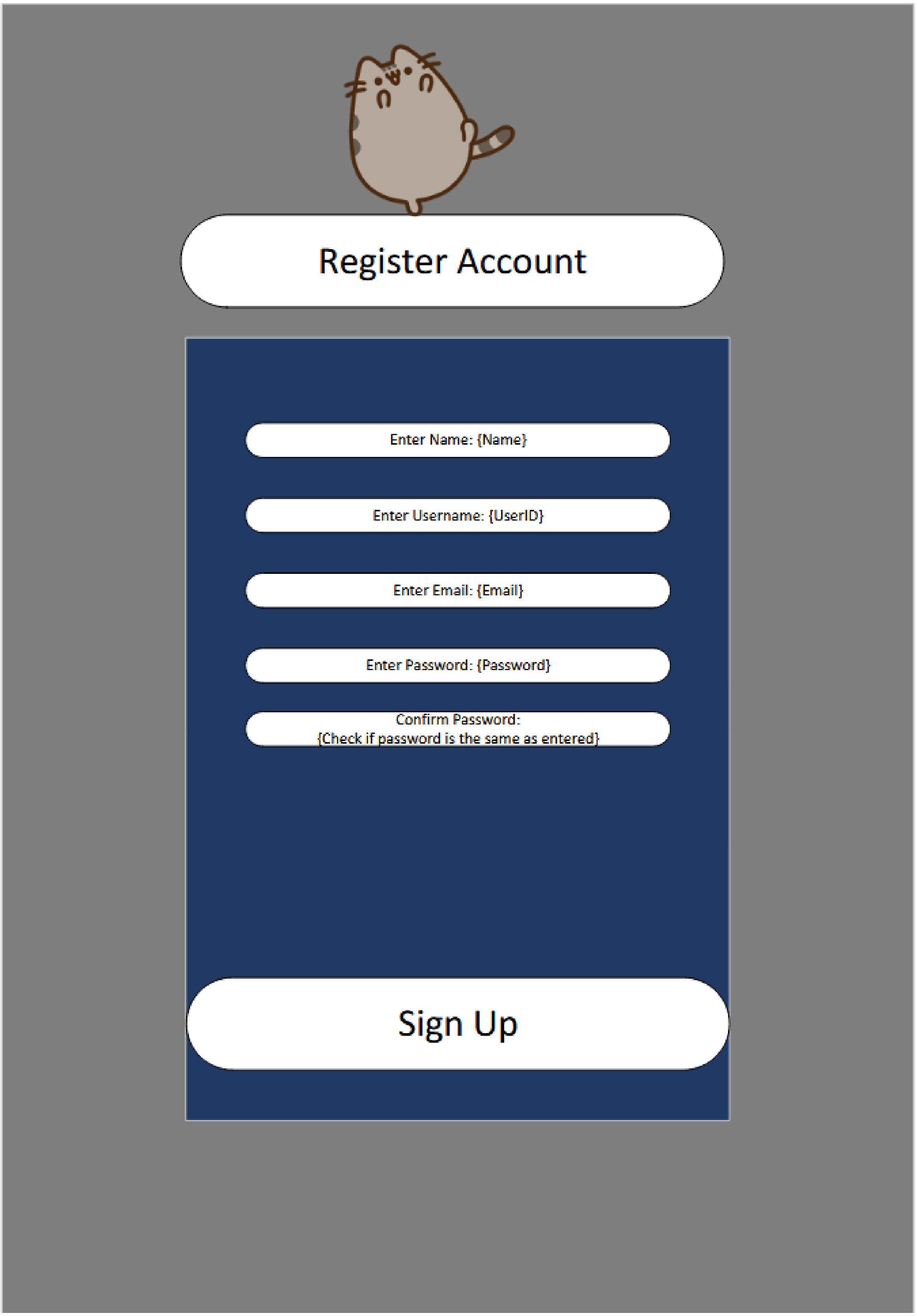
{Link to youtube movie preview}



Forgot your Information?

Request information change via email

Enter Email: {Email}



Register Account

Enter Name: {Name}

Enter Username: {UserID}

Enter Email: {Email}

Enter Password: {Password}

Confirm Password:
{Check if password is the same as entered}

Sign Up

Work Cited

Python tutorial. (n.d.). Retrieved March 9, 2023, from https://www.w3schools.com/python/