**Final Report**

Micky Driver

INFO C451: System Implementation

April 24, 2022

**Table of contents**

**Table of contents** 2

**1. Project Statement** 3

**2. Glossary of Terms** 3

**3. System Requirements** 3

**4. User Interface Specification** 12

**5. Domain Analysis** 14

**6. System Operation Contracts** 19

**7. Project Size Estimation** 22

**8. Plan of Work** 23

**9. Interaction Diagrams** 23

**10. Class Diagram** 30

**11. System Architecture** 27

**12. User Interface Design and Implementation** 32

**13. Design of Tests** 33

**14. Project management** 35

**15. References** 35

1. **Project Statement**

Most gyms today use a membership system that is outdated or doesn’t allow the user to complete tasks on their own. The current systems do not allow a user to sign in or make appointments with a trainer remotely. Most of the time a member just wants to get in, work out, and get out. A member can spend unwanted time at the front desk, along with a lot of time waiting for occupied machines. A perfect solution to this problem is an updated system to allow users to interact with the issues mentioned. The system will reduce the time that the members interact with the front desk in order to provide them with a hasty and satisfactory outcome.

1. **Glossary of Terms**

System – Generalization of user interface

Member – End user

Sign-up – User application and sign-up forum

Sign-in – User remote sign-in

Scheduler – Allows user to make appointments remotely

1. **System requirements**

**Functional Requirements**

|  |  |  |
| --- | --- | --- |
| **No.** | **Priority Weight** | **Description** |
| REQ-1 | 6 | Sign-up/Account creation |
| REQ-2 | 6 | Remote check-in |
| REQ-3 | 5 | Remote reservations |
| REQ-4 | 6 | Log-In/Sign-In |
| REQ-5 | 4 | Log-out/Sign-out |
| REQ-6 | 5 | Edit user information |
| REQ-7 | 6 | Retrieve user information |

**Non-functional Requirements**

|  |  |  |
| --- | --- | --- |
| **No.** | **Priority Weight** | **Description** |
| REQ-8 | 4 | Check log in history |
| REQ-9 | 6 | Show busy status of location |
| REQ-10 | 5 | User account verification through phone or email |
| REQ-11 | 3 | Internet connection |
| REQ-12 | 4 | Appointment time updates/changes |
| REQ-13 | 4 | Check payments |
| REQ-14 | 5 | Check user information for validation |

**Interface Requirements**

|  |  |  |
| --- | --- | --- |
| REQ-15 | 8 | Username and password for user account creation |
| REQ-16 | 6 | Device to access website/application |
| REQ-17 | 6 | Profile settings options to update user information |
| REQ-18 | 7 | Appointment scheduler |
| REQ-19 | 6 | Log-in UI |
| REQ-20 | 8 | Account information UI |
| REQ-21 |  |  |
| REQ-22 |  |  |

**Functional Requirement Specification**

**Stakeholders**

Stakeholders are those that have interest in the success of the business organization.

***Primary stakeholders:*** The primary stakeholders are those that use the system in order to monitor the database and increase efficiency of the organization.

*Management*

*Employees*

***Secondary stakeholders:*** The secondary stakeholders will use the system to increase satisfaction by increasing efficiency of checking in and scheduling remotely.

*Members*

*Potential Customers*

**Actors and Goals**

|  |  |
| --- | --- |
| **Actors** | **Goals** |
| User interface | Displays scheduling availability  Displays crowd density  Allows user to login  Allows user to sign up |
| System | Connects database to user interface  Updates user information  Collects user information  Collects user payments  Schedules appointments  Checks in user |
| Employee | Can read check-in logs  Can edit member information  Can schedule appointments for members |
| Customer | Can Log into system  Can view personal information  Can edit personal information  Can schedule appointments |
| Management | Can read check-in logs  Can manage system  Can edit member information  Can edit employee information |

**Use Cases**

UC1: Sign Up – Potential members can create an account

UC2: Log In - Members can access the site by logging in using their information

UC3: Check In – Members can check in remotely

UC4: Availability – Members can check to see how crowded the location is

UC5: Walk-In - Members can still sign up or check in at the front desk if they are not able to remotely

UC6: Scheduling – Member can schedule, edit, or cancel an appointment

UC7: Payment – Members can pay for their membership or products

UC8: Privacy - Encrypt member information for security

UC9: Updates - Send payment updates, scheduling, and change requests

UC10: User information – Members can view and edit their information or payment methods.

**Use Case Diagram**

**Diagram

Description automatically generated**

**Traceability Matrix**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **REQ** | **PW** | **UC1** | **UC2** | **UC3** | **UC4** | **UC5** | **UC6** | **UC7** | **UC8** | **UC9** | **UC10** |
| **Functional** | | | | | | | | | | | |
| REQ1 | 6 | X |  |  |  |  |  |  |  |  |  |
| REQ2 | 6 |  |  | X |  |  |  |  |  |  |  |
| REQ3 | 5 |  |  |  |  |  | X |  |  |  |  |
| REQ4 | 6 |  | X | X |  |  |  |  |  |  |  |
| REQ5 | 4 |  | X |  |  |  |  |  |  |  |  |
| REQ6 | 5 |  |  |  |  |  |  |  |  |  | X |
| REQ7 | 6 |  |  |  |  |  |  |  |  |  | X |
| **Nonfunctional** | | | | | | | | | | | |
| REQ8 | 4 |  |  |  |  |  |  |  |  |  | X |
| REQ9 | 6 |  |  |  | X |  |  |  |  |  |  |
| REQ10 | 5 |  |  |  |  |  |  |  |  | X |  |
| REQ11 | 3 |  |  |  |  |  | X |  |  |  |  |
| REQ12 | 4 |  |  |  |  |  | X |  |  | X |  |
| REQ13 | 4 |  |  |  |  |  |  | X |  |  | X |
| REQ14 | 5 |  | X |  |  |  |  |  | X |  | X |
| **Interface** | | | | | | | | | | | |
| REQ15 | 9 | X | X | X |  |  |  |  |  |  | X |
| REQ16 | 6 |  |  |  |  |  |  |  |  |  |  |
| REQ17 | 6 |  |  |  |  |  |  |  |  | X | X |
| REQ18 | 7 |  |  |  |  |  | X |  |  |  | X |
| REQ19 | 6 | X | X |  |  |  |  |  |  |  |  |
| REQ20 | 8 |  |  |  |  |  |  |  | X |  | X |

**Fully-dressed Description**

|  |
| --- |
| **Use Case 1:** Sign Up |
| **Requirements:** REQ1, REQ6, REQ10, REQ15, REQ16, REQ19, REQ20 |
| **Actors:** User, System, User Interface |
| **Goal:** To create a member account for the organization |
| **Precondition:** User must have access to a device and be on the website |
| **Postcondition:** The member is allowed to sign into their account on the website |
| **Flow of events:**   1. User clicks sign up button 2. User fills out personal information 3. System processes request 4. User validates account creation through email 5. User is redirected to login page where they can then login |

|  |
| --- |
| **Use Case 2:** Login |
| **Requirements:** REQ4, REQ5, REQ7, REQ10, REQ14, REQ16, REQ19 |
| **Actors:** User, System, User Interface |
| **Goal:** To login to the website with credentials |
| **Precondition:** Must have an account created |
| **Postcondition:** The member is allowed to edit information and schedule appointments |
| **Flow of events:**   1. User clicks on login button 2. User enters credentials 3. System processes request 4. User is redirected to home page |

**System Sequence Diagram**

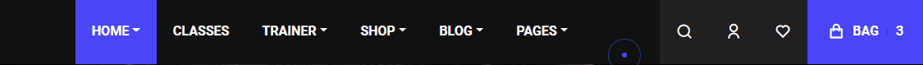
***Login as Member***

Diagram

Description automatically generated

1. **User Interface Specification**

**Preliminary Design**

When users first visit the site, they will be able to access most of the features. When they come across something that requires them to make an account it will prompt them. The navigation bar will look like the image below.

Members will be able to create a subscription plan for their membership and may decide on when the membership renewal should halt. The payment gateway will look like the image below.

Graphical user interface, application

Description automatically generated

Users will be able to view the status of an establishment. This status will show how busy it is currently depending on the number of members checked in. The image below reflects the status.

Chart, histogram

Description automatically generated

**User Effort Estimation**

|  |  |
| --- | --- |
| **Use Case** | **Number of Clicks** |
| Sign up | 5 |
| Log in | 3 |
| Remote check in | 6 |
| Availability | 2 |
| Schedule | 7 |
| Payment | 8 |
| Edit Information | 6 |

1. **Domain Analysis**

Diagram

Description automatically generated

Diagram

Description automatically generated

**Concept definitions**

|  |  |  |
| --- | --- | --- |
| **Class** | **Description** | **Type (Do, Know, Neither)** |
| Website | Schedule a meeting with trainer | Do |
| Website | Check in remotely | Do |
| Website | Cancel meeting with trainer | Do |
| Database | Retrieve payment information | Know |
| Database | Retrieve status of trainer meetings | Know |
| Database | Retrieve membership information | Know |
| Management | View member information | Neither |
| Management | View membership status of members | Neither |
| Management | Approve access | Do |
| Payment | Preapprove payments | Do |
| Payment | Make payments | Do |
| Payment | Change membership status | Do |
| User interface | Login/logout | Do |

**Association definitions**

|  |  |  |
| --- | --- | --- |
| **Class** | **Name** | **Description** |
| User to Website | Sending data | User inputs information requesting access from the website. The website allows users to do actions such as checkin and schedule. |
| Website to Database | Sending and retrieving data | Website requests authcheck to determine if a user can access the website. The Database receives requests from the website to validate information. |
| User to Payment | Sending data to update payment information and make payments | User inputs payment information allowing them to make payments on their membership. The user can also preapprove their payments. |
| Payment to Database | Sending Data to update membershipStatus and validate | Payment sends payment information to database requesting the updating of membershipStatus allowing the user to be approved. |
| Database to Management | Sending Data | Database sends Member information to determine if they are authorized to access the facility |
| Management to Doorcheck | Sending Data | Management sends member data to the front door physical check to determine if the member is who they say they are |

**Attribute definitions**

|  |  |  |
| --- | --- | --- |
| **Class** | **Attribute** | **Description** |
| Website | Schedule Meeting | Schedule a meeting with the trainer |
| Retrieve member info | Retrieve member and payment information |
| Remote Checkin | Check in to establishment remotely |
| Database | userID/Password | Validate userID and PW |
| membershipStatus | Gives current status of membership |
| meetingStatus | Gives status of trainer meetings |
| Payment | Preapproved | Preapprove future payments for membership |
| makePayment | Allows user to make a payment to activate their membership |
| User Interface | Login/logout | Allows user to log in and out of their member account |
| Management | View | Checks member information to determine if eligible |
| Approve | Approves checkin to transfer over to front door check |
| Door check | grantAccess | Checks membership status and member info to determine if access should be granted to facility |

1. **System Operation Contracts**

|  |  |
| --- | --- |
| **Contract CO1: enterUserID** | |
| **Operation** | enterUserID |
| **Cross Reference** | UC Login |
| **Preconditions** | Trying to log into system |
| **Post conditions** | UserID is associated with member account |

|  |  |
| --- | --- |
| **Contract CO2: enterPassword** | |
| **Operation** | enterPassword |
| **Cross Reference** | UC Login |
| **Preconditions** | Trying to log into system |
| **Post conditions** | Password is associated with userID |

|  |  |
| --- | --- |
| **Contract CO3: clickLogin** | |
| **Operation** | clickLogin |
| **Cross Reference** | UC Login |
| **Preconditions** | Trying to log into system |
| **Post conditions** | UserID is associated with member account  Password is associated with UserID |

|  |  |
| --- | --- |
| **Contract CO4: validateLogin** | |
| **Operation** | validateLogin |
| **Cross Reference** | UC Login |
| **Preconditions** | Trying to log into system |
| **Post conditions** | UserID is associated with member account & Password is associated with UserID & found in database |

|  |  |
| --- | --- |
| **Contract CO6: checkIn** | |
| **Operation** | checkIn |
| **Cross Reference** | UC Check In |
| **Preconditions** | Trying to check into facility |
| **Post conditions** | Member account has a valid membershipStatus  Member matches membership credentials |

|  |  |
| --- | --- |
| **Contract CO7: checkAvailability** | |
| **Operation** | checkAvailability |
| **Cross Reference** | UC Availability |
| **Preconditions** | Trying to check facility crowd meter |
| **Post conditions** | Database has updated list of active members currently within facility |

|  |  |
| --- | --- |
| **Contract CO8: scheduleAppointment** | |
| **Operation** | scheduleAppointment |
| **Cross Reference** | UC Scheduling |
| **Preconditions** | Trying to schedule appointment with trainer |
| **Post conditions** | Schedule has open slots  membershipStatus is active |

|  |  |
| --- | --- |
| **Contract CO9: makePayment** | |
| **Operation** | makePayment |
| **Cross Reference** | UC Payment |
| **Preconditions** | Trying to make payment to member account |
| **Post conditions** | membershipStatus is non active OR preapprovedPayments are ON |

|  |  |
| --- | --- |
| **Contract CO10: encryptData** | |
| **Operation** | encryptData |
| **Cross Reference** | UC Privacy |
| **Preconditions** | Database encrypts data making it more secure |
| **Post conditions** | Sensitive data such as payment and member info are encrypted to provide better security of member information |

|  |  |
| --- | --- |
| **Contract CO11: sendUpdates** | |
| **Operation** | sendUpdates |
| **Cross Reference** | UC Updates |
| **Preconditions** | System sends updates on any upcoming events within the member account |
| **Post conditions** | Payment reminders  Schedule updates  Change requests |

|  |  |
| --- | --- |
| **Contract CO12: authorizeMember** | |
| **Operation** | authorizeMember |
| **Cross Reference** | UC Walk-In |
| **Preconditions** | Management authorizes members via walk-in to give access to facility |
| **Post conditions** | membershipStatus is active  memberID matches credentials |

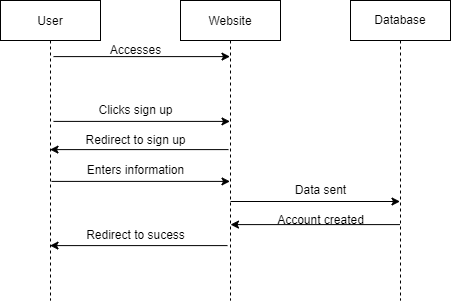
1. **Project size estimation**

It would be estimated about 12 hours per use case point. There may be more added than what is listed, but there are 10 Use Case points which accumulates to about 120 hours of work at the current evaluation.

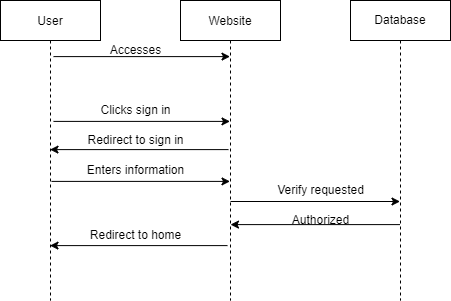
1. **Table

   Description automatically generatedPlan of work**
2. **Interaction Diagrams**

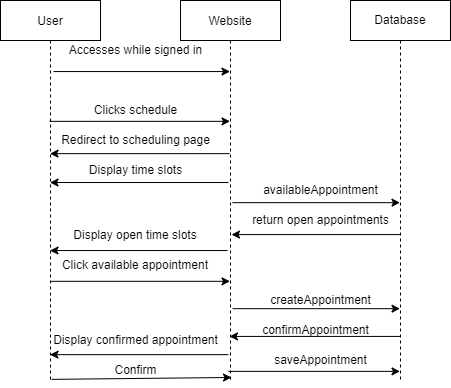
|  |
| --- |
| **Use Case 1:** Sign Up |
| **Requirements:** REQ1, REQ6, REQ10, REQ15, REQ16, REQ19, REQ20 |
| **Actors:** User, System, User Interface |
| **Goal:** To create a member account for the organization |
| **Precondition:** User must have access to a device and be on the website |
| **Postcondition:** The member is allowed to sign into their account on the website |
| **Flow of events:**   1. User clicks sign up button 2. User fills out personal information 3. System processes request 4. User validates account creation through email 5. User is redirected to login page where they can then login |

****

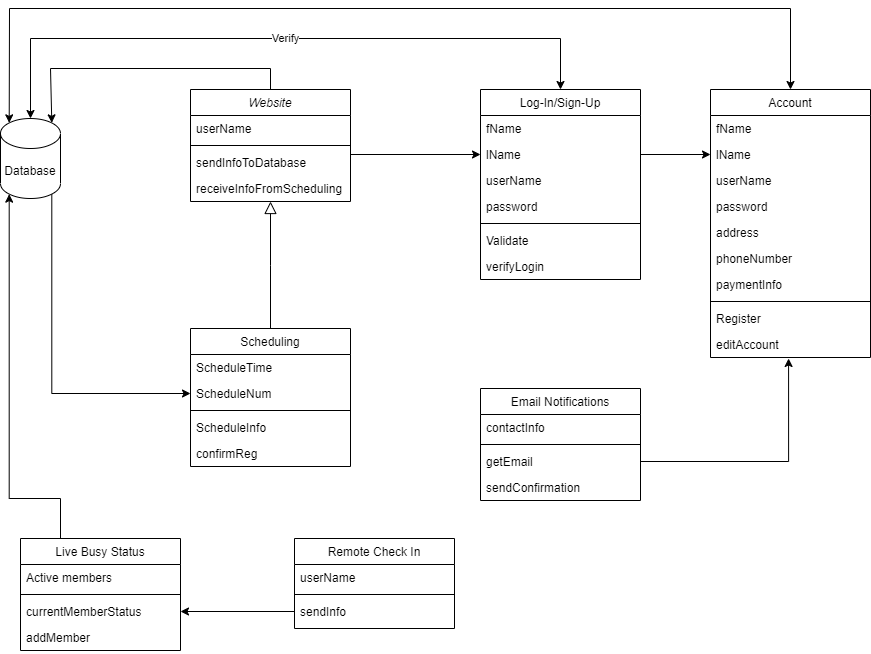
|  |
| --- |
| **Use Case 2:** Login |
| **Requirements:** REQ4, REQ5, REQ7, REQ10, REQ14, REQ16, REQ19 |
| **Actors:** User, System, User Interface |
| **Goal:** To login to the website with credentials |
| **Precondition:** Must have an account created |
| **Postcondition:** The member is allowed to edit information and schedule appointments |
| **Flow of events:**   1. User clicks on login button 2. User enters credentials 3. System processes request 4. User is redirected to home page |

****

|  |
| --- |
| **Use Case 3:** Scheduling |
| **Requirements:** REQ3, REQ6, REQ7, REQ11, REQ14, REQ16, REQ18 |
| **Actors:** User, System, User Interface |
| **Goal:** To schedule an appointment for the user |
| **Precondition:** User must have an account on the database |
| **Postcondition:** The member can schedule an appointment |
| **Flow of events:**   1. User signs in to account 2. User creates schedule request 3. System processes request 4. User validates schedule through email 5. User is redirected to success page |



1. **Class Diagram and Interface Specification**



**Data Types and Operation Signatures**

1. **Account**

*Attributes*

fName: first name of the member

lName: last name of the member

username: username of member

password: password of member

address: email of member

phoneNumber: phone number of member

paymentInfo: payment information of member

*Operations*

Register: User can create new information to be sent to the database

editAccount: Member can update their account information

1. **Website**

*Attributes*

userName: username of currently signed in member

*Operations*

sendInfoToDatabase: sends new information to database

receiveInfoFromScheduling: receives updated info from scheduling for uers

1. **Scheduling**

*Attributes*

ScheduleTime: current timeslots for scheduling

ScheduleNum: scheduling number for appointment

*Operations*

scheduleInfo: displays scheduling information

confirmReg: confirms scheduling appointment

1. **Log In/Sign Up**

*Attributes*

fName: first name of the member

lName: last name of the member

username: username of member

password: password of member

*Operations*

validate: makes sure user information is not already in use

verifyLogin: verifies to make sure login information is correct

1. **Remote Check In**

*Attributes*

userName: username of checked in user

*Operations*

sendInfo: sends check in information to database and status

1. **Live Busy Status**

*Attributes*

Active members: members currently at facility

*Operations*

currentMemberStatus: displays current amount of members

addMember: adds member to current amount of members that use remote check in

1. **Email Notification**

*Attributes*

contactInfo: signed in user’s account information

*Operations*

getEmail: gets signed in user’s email address

sendConfirmation: sends scheduling and other confirmations to user email

**Traceability Matrix**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **REQ** | **PW** | **UC1** | **UC2** | **UC3** | **UC4** | **UC5** | **UC6** | **UC7** | **UC8** | **UC9** | **UC10** |
| **Functional** | | | | | | | | | | | |
| REQ1 | 6 | X |  |  |  |  |  |  |  |  |  |
| REQ2 | 6 |  |  | X |  |  |  |  |  |  |  |
| REQ3 | 5 |  |  |  |  |  | X |  |  |  |  |
| REQ4 | 6 |  | X | X |  |  |  |  |  |  |  |
| REQ5 | 4 |  | X |  |  |  |  |  |  |  |  |
| REQ6 | 5 |  |  |  |  |  |  |  |  |  | X |
| REQ7 | 6 |  |  |  |  |  |  |  |  |  | X |
| **Nonfunctional** | | | | | | | | | | | |
| REQ8 | 4 |  |  |  |  |  |  |  |  |  | X |
| REQ9 | 6 |  |  |  | X |  |  |  |  |  |  |
| REQ10 | 5 |  |  |  |  |  |  |  |  | X |  |
| REQ11 | 3 |  |  |  |  |  | X |  |  |  |  |
| REQ12 | 4 |  |  |  |  |  | X |  |  | X |  |
| REQ13 | 4 |  |  |  |  |  |  | X |  |  | X |
| REQ14 | 5 |  | X |  |  |  |  |  | X |  | X |
| **Interface** | | | | | | | | | | | |
| REQ15 | 9 | X | X | X |  |  |  |  |  |  | X |
| REQ16 | 6 |  |  |  |  |  |  |  |  |  |  |
| REQ17 | 6 |  |  |  |  |  |  |  |  | X | X |
| REQ18 | 7 |  |  |  |  |  | X |  |  |  | X |
| REQ19 | 6 | X | X |  |  |  |  |  |  |  |  |
| REQ20 | 8 |  |  |  |  |  |  |  | X |  | X |

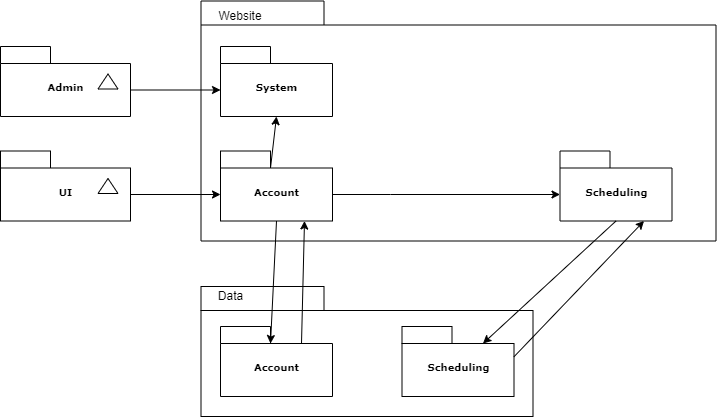
1. **System Architecture and System Design**

**Architectural styles**

The gym management system follows the Component-Based Development design style. The CBD focuses on production of individual components that can be used on their own. The reusability of these components is one of the primary objectives of component-based development. The components within this design style aim to have minimal dependency on other components.

Along with Component-Based Development design styling, the system also follows an event-driven style. The scheduling system reacts to the events that a user takes to update the scheduling data. The user is also able to create an account which is an event triggering the storage of data within the database.

**Identifying Subsystems**



**Mapping Subsystems to Hardware**

This system will be able to work on multiple computers and devices. The system will be able to communicate with the UI on the site and send the data received to the database. The database will verify all information and save it accordingly.

**Persistent Data Storage**

The database will store user information on remote drives allowing data to be retrievable if the system were ever to fail. The database allows users to change data while still being able to meet persistent requirements.

**Global Control Flow**

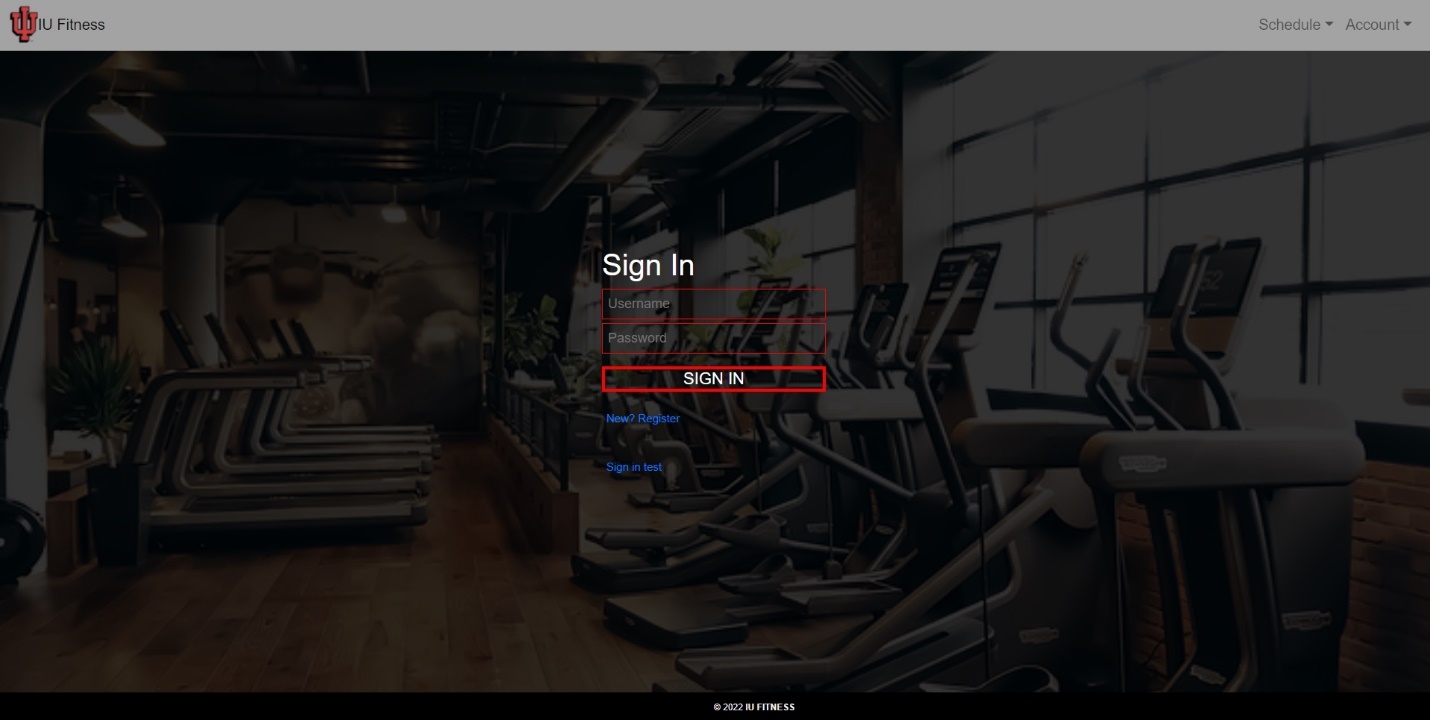
The system uses time dependency to use scheduling and real-time appointment updates. The scheduling subsystem is based off of real-time events created by the user and maintained by administrators.

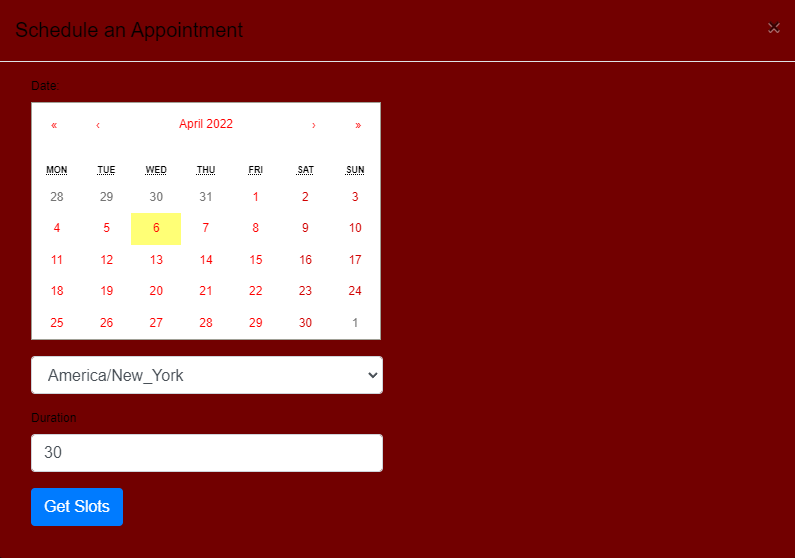
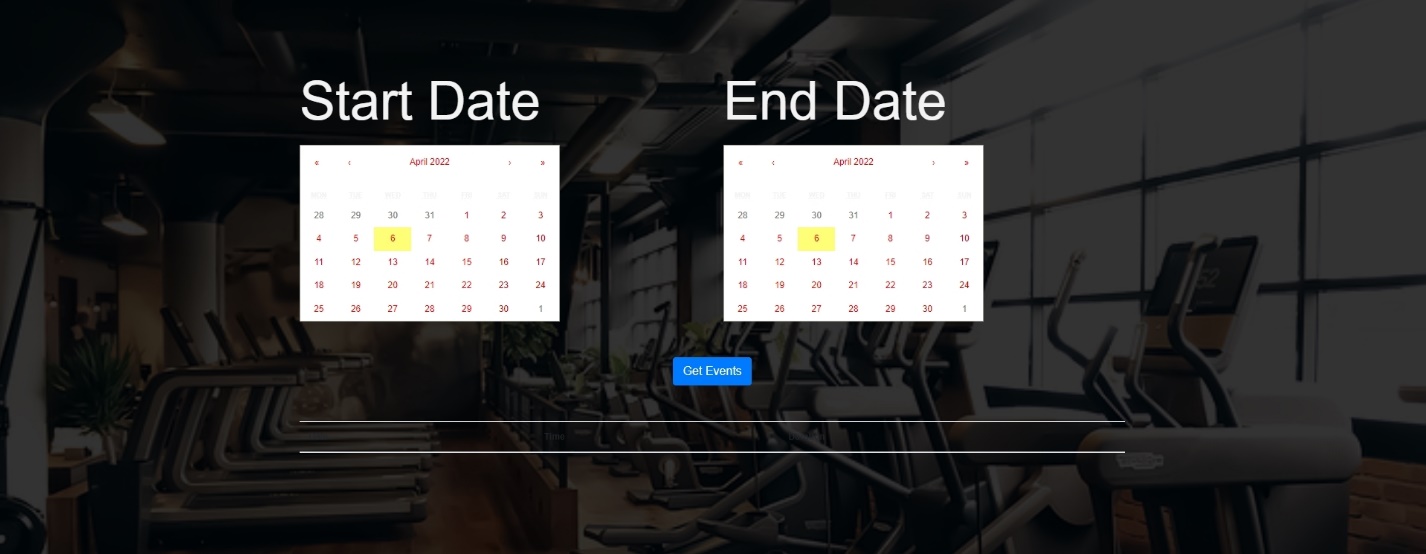
**Hardware Requirements**

1. **Processor:** 1.9 gigahertz (GHz)
2. **Memory:** 2GB RAM
3. **Display:** Super VGA with a resolution of 1024 x 768
4. **Bandwidth:** greater than 50 KBps
5. **Latency:** under 150ms
6. **Hard disk:** 1 Gbyte
7. **User Interface Design and Implementation**

A screenshot of a video game

Description automatically generated with medium confidence





1. **Design of Tests**

Units that are to be tested include:

* Website
* Database
* Scheduling

**Website**

The website is used for the user interface and can display all information from scheduling and the database. The website also includes sign in, sign up, and remote check in for users.

The test cases that are to be used include:

* Signing up

Check to see if new users can create an account and send information to the database

* Sign In

Check to see if signed up users can access their account and information associated

* Scheduling

Check to see if the scheduling information is correct and in real time

* Remote Check In

Check to see if remote check in works and updates current member count

* Edit Information

Check to see if updated user information is correctly processed and stored

**Database**

The database is used to store information about members and other stakeholders within the system. The database also stores and retrieves scheduling information for member appointments.

The test cases that are to be used include:

* Appointments

Check timeslots available for scheduling appointments

* Account linking

User account is linked to information and stored correctly within the database

**Scheduling**

Scheduling is used to schedule appointments for members based on availability.

The test cases that are to be used include:

* Calendar

Check if calendar is setup correctly and dates are selectable as well as timeslots

* Data

Check if scheduling can send and receive updated information from the database to display

1. **Project Management**

The basics of user authentication have already been implemented and the main purpose and functionality of the site (to create scheduling and remote check-ins) is currently being worked on. The big use case that is very exciting and in depth is the scheduling functions.

Use cases that have already been implemented:

UC1: Sign Up

UC2: Log In

Use cases that are in progress:

UC3: Remote Check In

UC6: Scheduling

1. **References**

Tilley, S., & Rosenblatt, H. (2017). Systems analysis and design (11th ed). Boston, MA: Cengage. ISBN-13: 9781337338035

W3schools.com. 2017. W3Schools Online Web Tutorials. [online] Available at: <https://www.w3schools.com/>

Schwalbe, K. (2019). *Information Technology Project Management*. Cengage Learning.

Project Management Institute. (2017). A guide to the Project Management Body of Knowledge (PMBOK guide) (6th ed.).