

GymBuddy

Your path to a successful workout!

Group 6

Ahnaful Arephin

Vallabhan Kudlu Ramakrishnan

Bhavan Mehta

Table of Contents

[**Project Presentation YouTube Link**](#_7qb1i3v4f9t4) **2**

[**Project Summary**](#_ntvzlkh0bml1) **3**

[1. Executive Summary](#_hqz6ghpyc4i9) 3

[2. Problem Statement](#_ob3p53p3qirm) 4

[**Analysis**](#_ttmsbetgtv49) **4**

[3. BPMN Diagrams](#_rxmw6me2ac1m) 4

[4. Context Diagram](#_28ctez613t26) 6

[5. Process Model](#_ctzyfwg579gx) 7

[Use Case Diagram](#_rso7c6534r8d) 7

[Use Case Descriptions](#_tac0uwvnjxcd) 8

[Use Case Description 1: Update Check-In Information](#_vo2dy79j2of0) 8

[Use Case Description 2: Calculate Occupancy Rate](#_gqy9e4v4rtfe) 8

[Use Case Description 3: Estimated Wait Time](#_1gq5aehl24m2) 9

[Use Case Description 4: Display Wait Time](#_ld1niomfa38c) 9

[Use Case Description 5: Register as a GymBuddy Member](#_hkwpuk6i0w9) 9

[Use Case Description 6: Authorize Log-In](#_8tcdnscbwcxl) 10

[Use Case Description 7: Search Exercise](#_qcz6msvrcna) 10

[Use Case Description 8: Rate Gym](#_1a5f4owb73g5) 11

[Data Dictionary](#_e0i41oe43v5n) 11

[Use Case: Update checkout information](#_amfw1z3bx7hr) 11

[Use Case: Average Wait time](#_8ljbe0qcymwo) 11

[Use Case: Display wait time](#_ycskvymogp24) 11

[Use Case : Sign Up](#_nilvtvrw8bnh) 12

[Use Case : Authorize Login](#_2q2ju4g7orkz) 12

[Use Case: Review Gym](#_3up1b35dzo05) 12

[6. Class Diagram (without Methods)](#_p5607pphvhv5) 12

[7. Sequence Diagrams for Major Use Cases](#_7r2jf2qmir2p) 15

[8. Functional Specifications for the proposed system](#_we5vlgjmzabk) 17

[9. Interface Design](#_3sjkclmcqp0) 18

[**Design**](#_fvnj0l1smo0r) **20**

[10. Database Design](#_k6gazpfjmu0v) 20

[Entity Relationship Diagram](#_dp9l5p2ikwvc) 21

[Database Constraints](#_xrq76uv2cakf) 23

[Member Table Constraints](#_4v10tjgokpg4) 23

[Gym Table Constraints](#_vgawixogaemv) 23

[Wait Timetable Constraints](#_102yk7klh5g) 23

[Capacity and Exercise table Constraints](#_90hr5vkg6s7n) 24

[Member Table Constraints](#_67sozeyw1rsj) 24

[Gym\_Buddy Constraints](#_gtgnb5kc1d6n) 24

[Rating Constraints](#_rg5ysr6xjzml) 25

[11. Complete Class Diagram (with Methods)](#_js37txg66dbw) 25

[12. Software Design:](#_40iwdtmktl6) 27

[13. Minutes of project meetings](#_urp7niwm9x5w) 31

[**References**](#_ii2en8c1c1fh) **35**

# Project Presentation YouTube Link

<https://youtu.be/ykbj8D05yUw>

# Project Summary

## 1. Executive Summary

Staying healthy is a vital part of everyday life for many people. Going to the gym is one method people use to stay in shape and stay healthy. According to Statista.com, membership at gyms and fitness centers have nearly doubled since 2000. This rapid influx of new memberships has made these gyms and fitness centers very busy at peak periods and made it harder for members to find a good time to visit the gym and perform their desired workouts. As more and more gyms open up to accommodate and invite more members, it becomes harder to find and determine the best time to visit the gym and perform workouts.

After deliberating on all the issues we are presented with and multiple possible solutions, we have decided to approach this problem by incorporating IoT sensors in the workout equipment and using technology and database systems to help members understand and determine the best possible time to workout as well as provide workout recommendations based on what equipment is currently available at that location.

As gym members enter the gym, they currently are required to swipe or scan their membership card in order to be allowed entrance into the gym. Currently, the gym system does nothing with this data other than validate the membership. The GymBuddy app will use this data and calculate the occupancy rate and show GymBuddy users how busy the gym is currently.

There will be IoT sensors embedded into every piece of equipment in the gym (such as workout machines, dumbbell racks, bench press bars/benches, etc.) that will sense if the equipment is currently in use by a person or not. For example, if a person grabs a dumbbell from the rack, the sensor embedded in the rack will sense that the dumbbell is not there and will provide that information to the GymBuddy app.

The GymBuddy app will provide workout recommendations to the gym member based on their personal preferences detailed in the user profile as well as equipment availability. For example, if the user wants to do leg workouts that day, but there are no equipment currently available, the GymBuddy app will provide exercises that can be performed without that equipment and can provide estimated wait time for when the desired equipment will be available for use.

This report will delve into the details of only one gym location, but theoretically and ideally, this service can be implemented and expanded to multiple locations and other gyms in the region and more to better serve those in other locations.

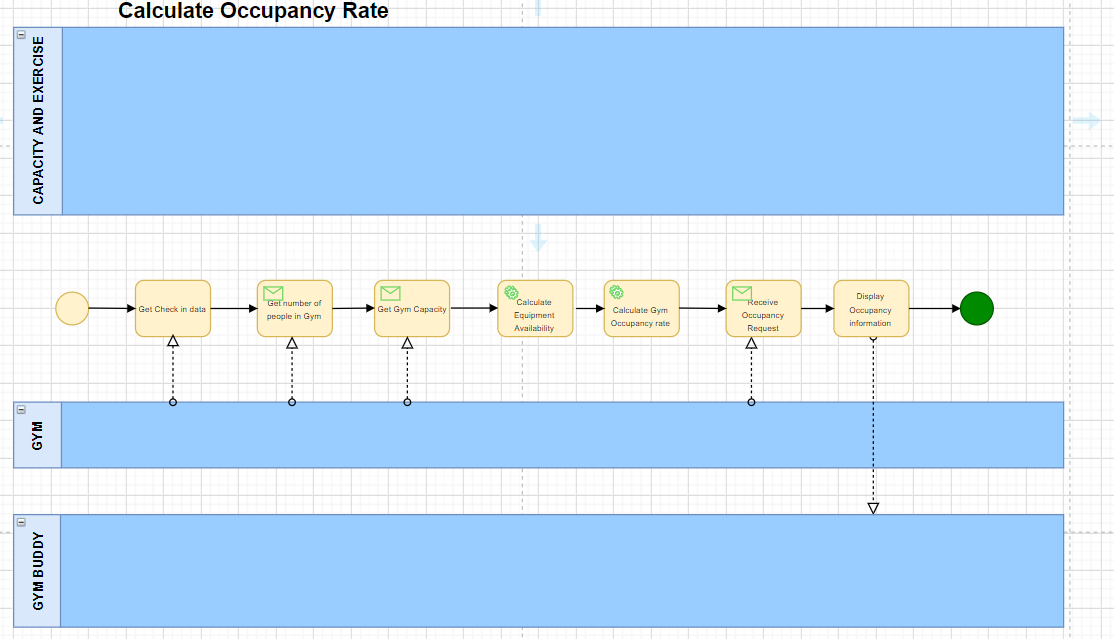
## 2. Problem Statement

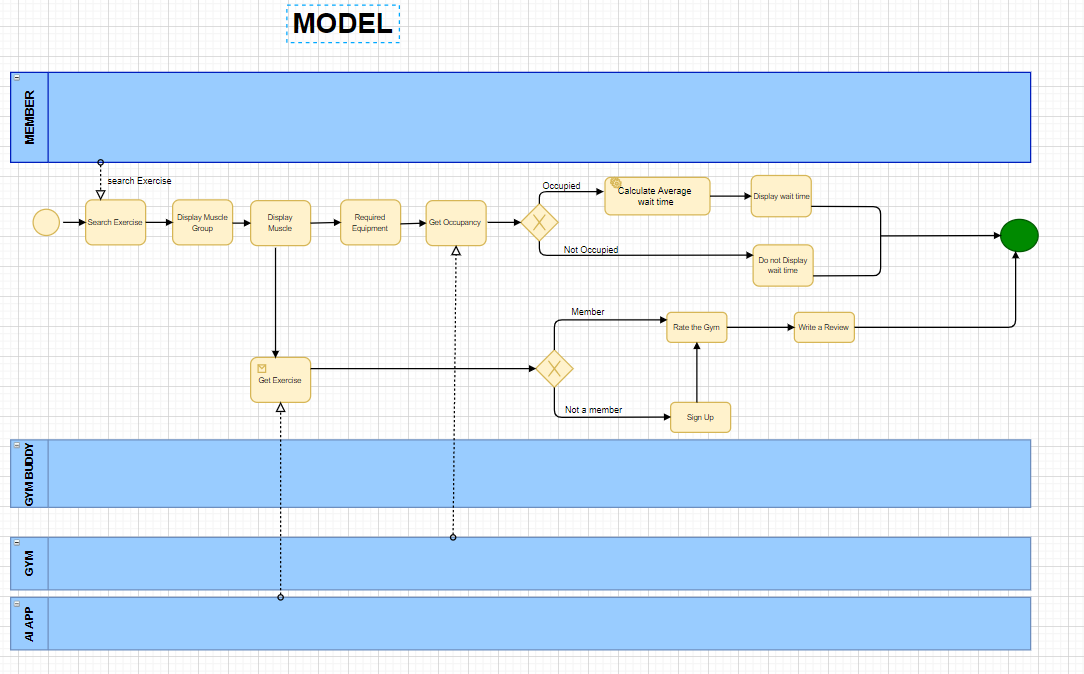
Before a gym member goes to the gym, they may have no idea how busy the gym is currently and would have to waste a lot of time and effort traveling to the location just to see that the gym is too busy. Currently, there is no system that displays real-time on-demand information that shows how busy a gym is or what equipment is available at that particular gym that is available for use.

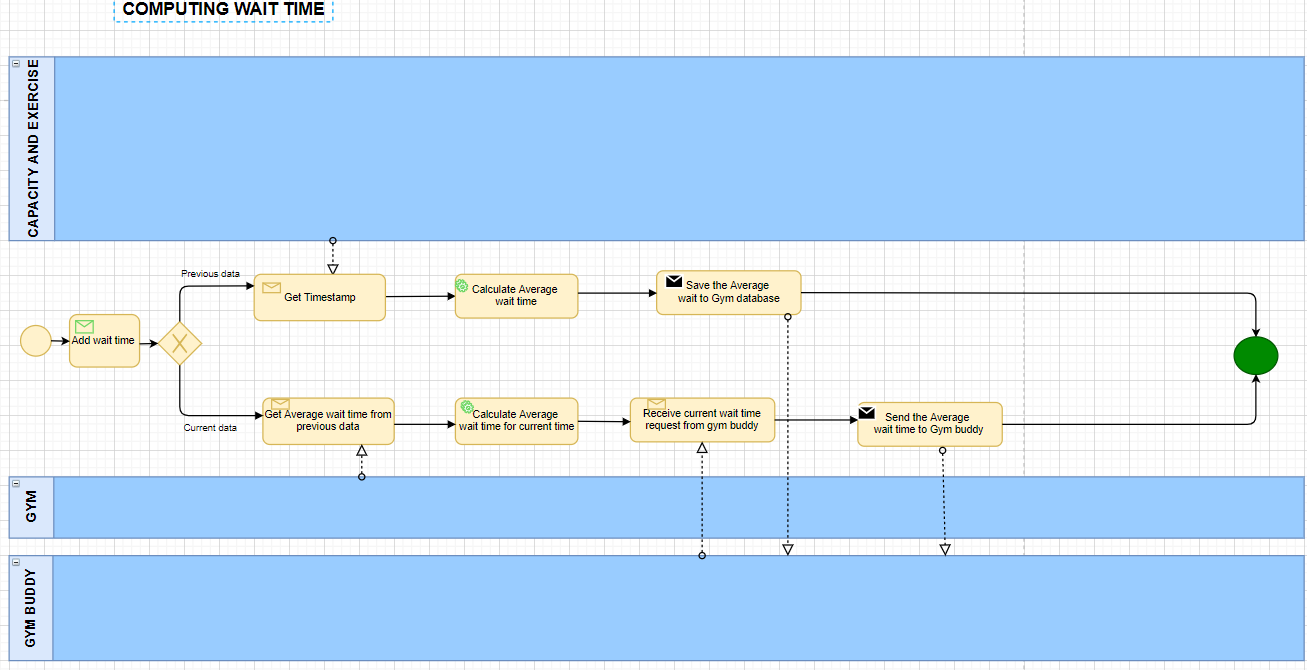
There are many different apps that are available on the App Store and Google Play Store that provide the user with very general fitness guides and tips. However, there is nothing currently available that is tailored to what equipment a gym has and what workout routines can be performed based on what equipment is currently available. Implementing a system that can provide the aforementioned services to a gym member would be very useful to both gym members and the gyms itself.

# Analysis

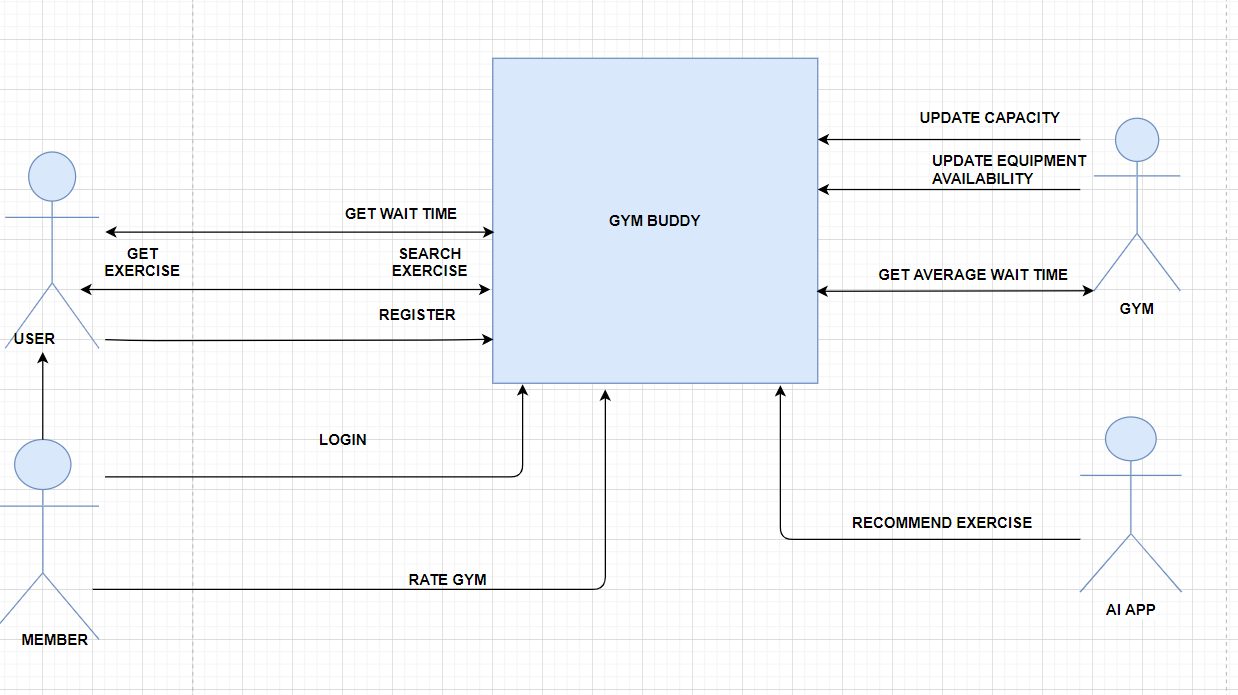
## 3. BPMN Diagrams







## 4. Context Diagram



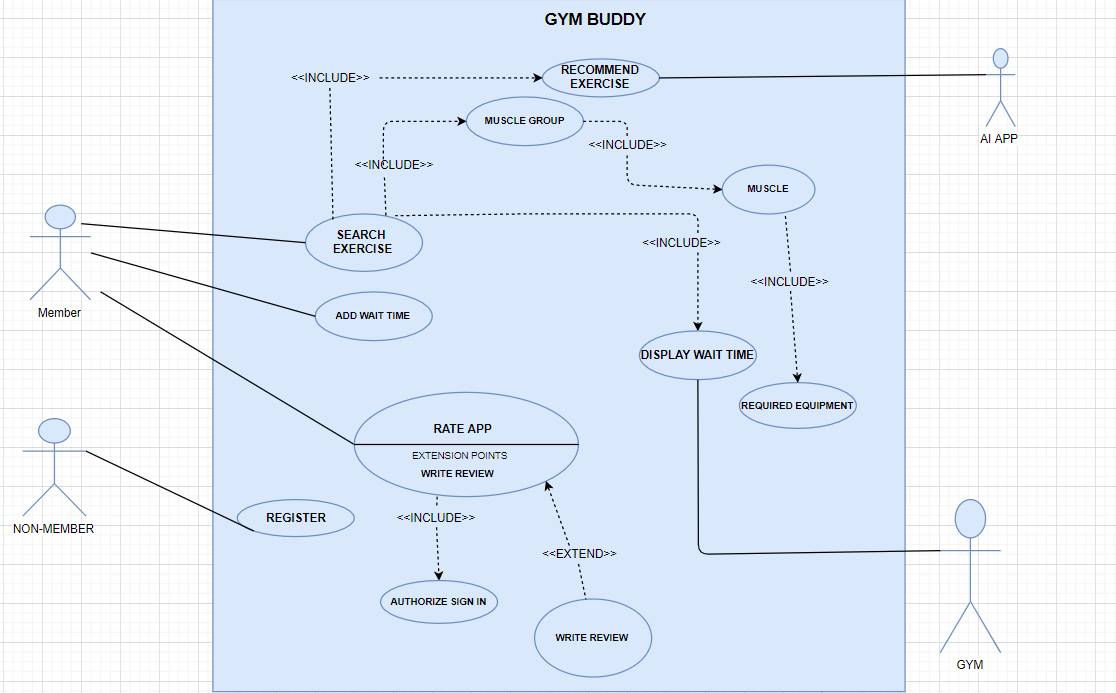
## 5. Process Model

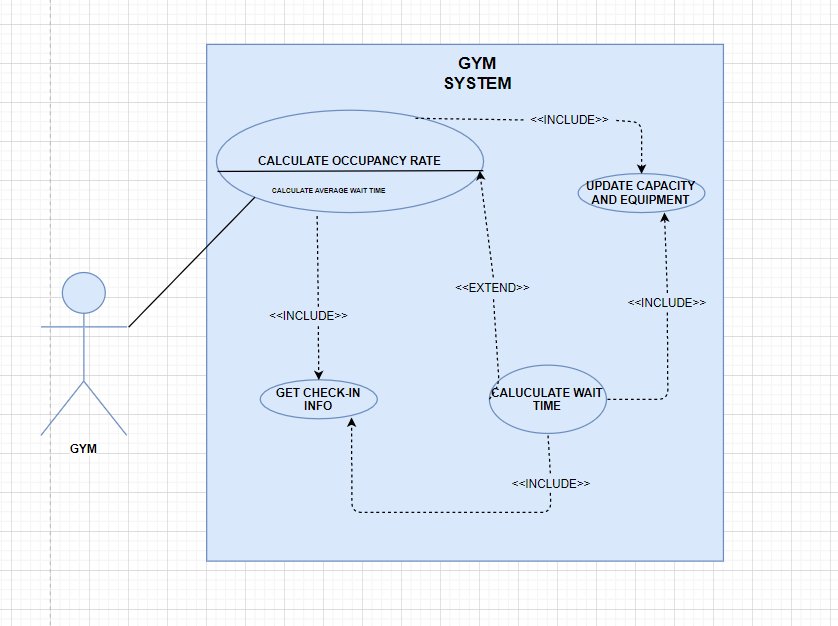
### Use Case Diagram

The below diagram represents all of the use cases for the GymBuddy app system. The actors are Gym Member, GymBuddy User, and Gym System.

The use cases involved are listed below.

1. Update Check-In Information
2. Calculate Occupancy Rate
3. Estimated Wait Time
4. Display Wait Time
5. Register as a GymBuddy Member
6. Authorize Log-In
7. Search Exercise
8. Rate App





### Use Case Descriptions

#### Use Case Description 1: Update Check-In Information

Primary Actor: Gym Member

Stakeholders: Gym

Brief Description: Updates Check-In information when the member checks in

Trigger: Customer swipes card and checks in

Normal Flow of Events

1. Member is ready to check-in and notifies the gym employee
2. Gym employee scans/swipes gym member’s membership card
3. Member checks into the gym
4. Gym employee updates the Gym system

Exception Flow:

2a. If the check-in fails, notify member

#### Use Case Description 2: Calculate Occupancy Rate

Primary Actor: Gym

Stakeholders: GymBuddy User

Brief Description: Computes occupancy rate for a system

Trigger: Receive request from GymBuddy user

Normal Flow of Events:

1. Retrieve Check-in data from Gym system
2. Retrieve Number of Member Swipes from Gym system
3. Retrieve Total Gym Capacity from Gym system
4. Retrieve Equipment Availability from Gym system
5. Calculate Occupancy Rate using swipe data, total capacity, and equipment availability
6. Send Occupancy data about how free the gym is and Equipment Availability to GymBuddy interface

Exception Flow:

5a. If the Occupancy Rate is 100%, then compute Average Wait Time use case

#### Use Case Description 3: Estimated Wait Time

Primary Actor: Gym

Stakeholders: GymBuddy User

Brief Description: Computes average wait time for a gym

Trigger: When Occupancy Rate is 100%

Normal Flow of Events:

1. Retrieve system time
2. Retrieve Equipment availability from Gym system based on retrieved time
3. Retrieve Number of Members Waiting in Queue data from queue database based on retrieved time
4. Calculate Estimated wait time based on Equipment availability and Number of Members Waiting and Retrieved Time
5. Send Estimated Wait Time information to GymBuddy system

#### Use Case Description 4: Display Wait Time

Primary Actor: GymBuddy

Stakeholder: Gym

Brief Description: Display the wait time for the selected gym

Trigger: Select a Wait Time from the menu

Normal Flow of Events:

1. The user selects a Wait time for the gym.
2. Retrieve the calculated wait time from the Gym system and display it in the app

#### Use Case Description 5: Register as a GymBuddy Member

Primary Actor: Gym Member

Stakeholder: GymBuddy

Brief Description: GymBuddy registration for Gym members

Trigger: When a user taps on the Sign-Up/Register button on the app

Normal Flow of Events:

1. User opens GymBuddy app
2. User taps the “Sign-Up/Register” button on the main splash screen
3. User inputs User Info which includes First Name, Last Name, Email Address, Password, and Gym Member ID
4. User Database file is updated
5. User continues on to Main Menu of the app

Exception Flow:

3a. If the user enters log-in details that already exist in the system, then direct user to log-in page.

3b. If the user enters invalid details, then display “Details invalid, Registration Failed”

#### Use Case Description 6: Authorize Log-In

Primary Actor: Gym Member

Stakeholder: GymBuddy

Brief Description: When a member wants to use GymBuddy services, login details will be authorized

Trigger: When a gym member tries to log into GymBuddy

Normal Flow of Events:

1. User opens GymBuddy app
2. User taps the “Log-in” button on the main splash screen
3. User inputs Email Address and Password
4. Retrieve User Info from User Database file
5. If credentials match, then authorize log-in
6. User continues on to Main Menu of the app

Exception Flow:

3a. If the user enters log-in details that do not already exist in the system, then direct user to the sign-up/register page

3b. If the user enters incorrect or invalid log-in details, then display “Log-in failed”

#### Use Case Description 7: Search Exercise

Primary Actor: Member

Stakeholder: GymBuddy

Brief Description: Displays the Muscle Group

Trigger: The user clicks on Exercise

Normal flow of events:

1. The user clicks on Exercise in menu screen
2. Display the Muscle Group
3. User clicks on Muscle Group and muscle will be displayed
4. User selects the Muscle
5. Exercise for the Muscle is displayed.

#### Use Case Description 8: Rate Gym

Primary Actor: Customer

Stakeholder: GymBuddy

Brief Description: Provide a rating for the Gym

Trigger: The user clicks on Rate and Review

Normal flow of events:

1. User will Login
2. User clicks on Rate and Review
3. User will navigate to the desired gym in a list
4. User provides a Rating and Writes a Review.

### Data Dictionary

The data dictionary used is given below:

#### Use Case: Update checkout information

Checkout data = Member First Name + Member Last Name + Member Email ID + Member Phone Number + Checkout Date + Checkout Time

Checkout = checkout data

Gym= Gym name + Address + Capacity + Equipment Type + City + State + Zip Code

Use case Name: Calculate Occupancy Rate

Checkout data = Member First Name + Member Last Name + Member Email ID + Member Phone Number + Checkout Date + Checkout Time

Checkout Data = {Checkout}

Occupancy Rate = Number of people + Capacity + Equipment availability

Member Details = First Name + Last Name + Mobile Phone No + Home Phone No + Email ID

Gym= Gym name + Address + Capacity + Equipment Type + City + State + Zip Code

Equipment Availability = Capacity – (Number of people + Number of Equipment)

#### Use Case: Average Wait time

System time = Current time

Time slot = [Morning | Afternoon | Evening]

Checkout data = Member First Name + Member Last Name + Member Email ID + Member Phone Number + Checkout Date + Checkout Time

Capacity and Exercise = Exercise details + Exercise timestamp + No of people

Checkout Data = {Checkout}

Estimated wait time = Time slot + Total time + Exercise time + Checkout time + Capacity

#### Use Case: Display wait time

Wait time = Data Element

#### Use Case : Sign Up

First Name = Data element

Last Name = Data element

Email Address = Data element

Password = Data element

Zip Code = Data element

Birthday = Data element

Gender = Data element

User Info = First Name + Last Name + Email Address + Password + Zip Code + Birthday + Gender

#### Use Case : Authorize Login

Email Address: Data Element

Password: Data Element

User Info = First Name + Last Name + Email Address + Password + Zip Code + Birthday + Gender

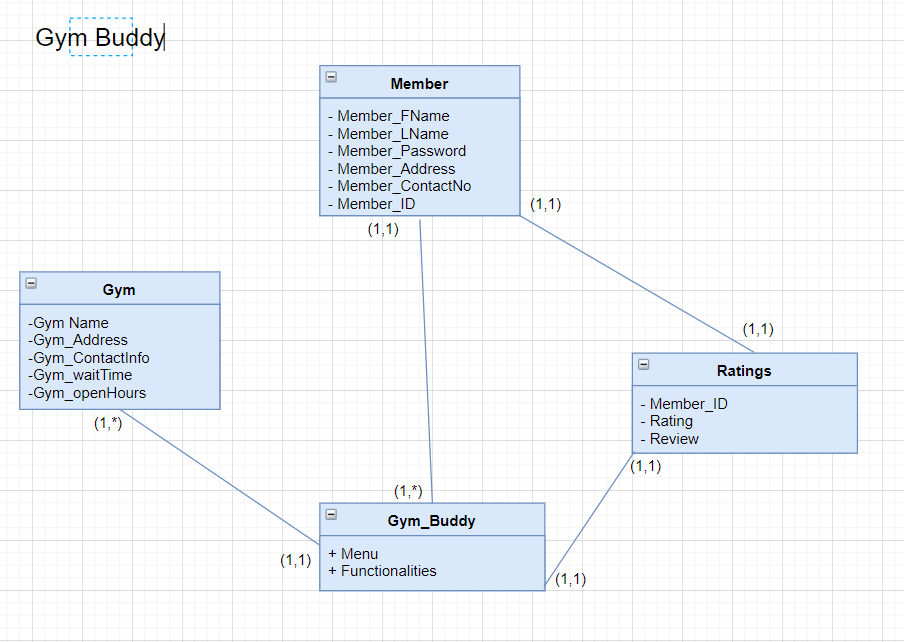
#### Use Case: Review Gym

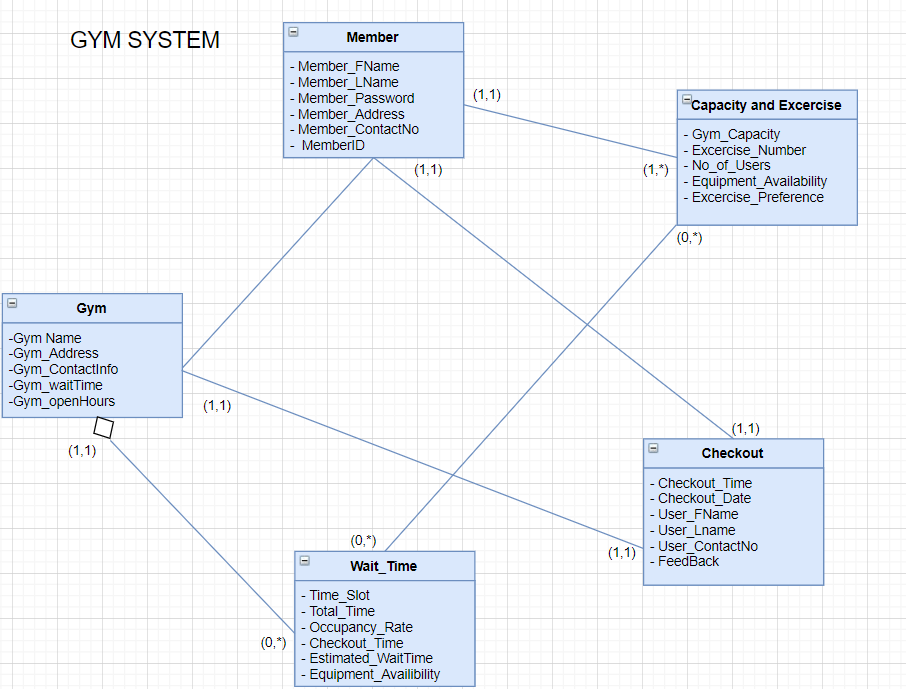
Review = Data Element

Gym Rating = [1 | 2 | 3 | 4 | 5]

Review Gym = Gym Rating + Review

## 6. Class Diagram (without Methods)





## 

## 

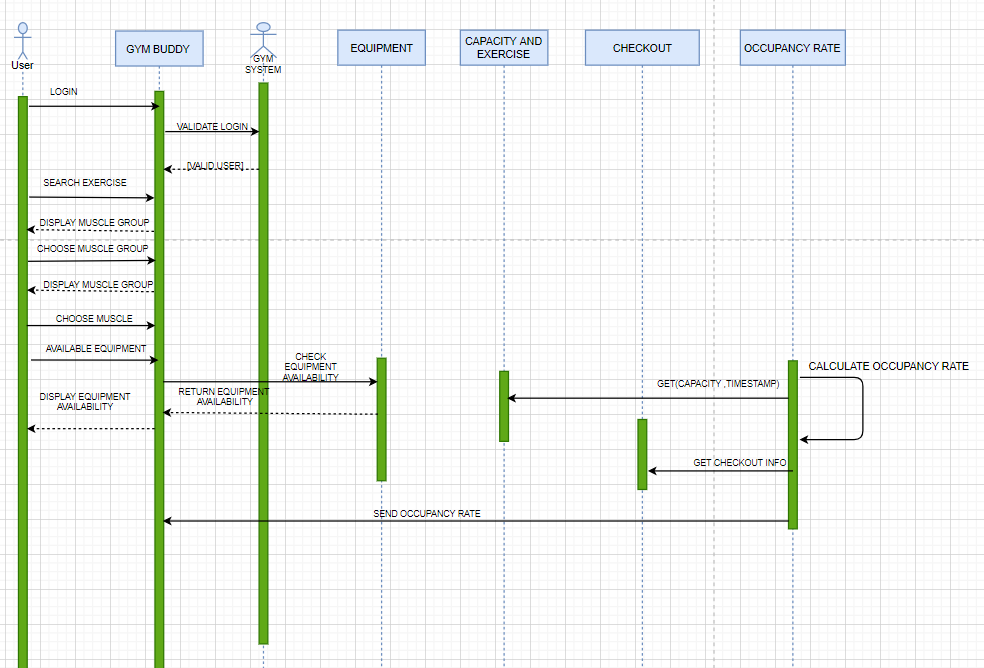
## 

## 

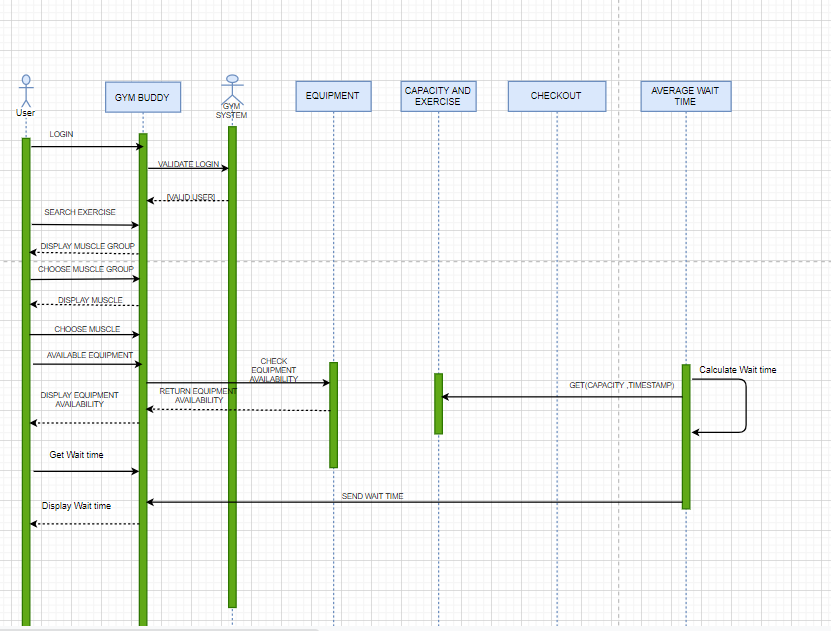
## 

## 7. Sequence Diagrams for Major Use Cases

**Occupancy Rate:**



**Average Wait Time**

****

## 

## 

## 

## 

## 

## 8. Functional Specifications for the proposed system

* The functionality will provide real-time, or as close to real-time as feasibly possible, on how busy the gym is currently, based on the total capacity of the gym, and how many people are currently checked in, compared to how many have checked out.
* The functionality will have the capability to provide real-time, or as close to real-time as feasibly possible, information on the average wait time when the desired equipment is currently occupied.
* The functionality will also provide an option for GymBuddy users to rate and review the desired gym in order to serve as a helpful tool for other GymBuddy users who are looking for a different gym without having to leave the app.
* The proposed functionality listed above will allow GymBuddy users to manage their time efficiently and give those users reliable information to better plan their workouts.

## 

## 

## 

## 

## 

## 

## 

## 9. Interface Design

## 

## 

## 

## 

## 

## 

## 

# 

# 

# 

# 

# 

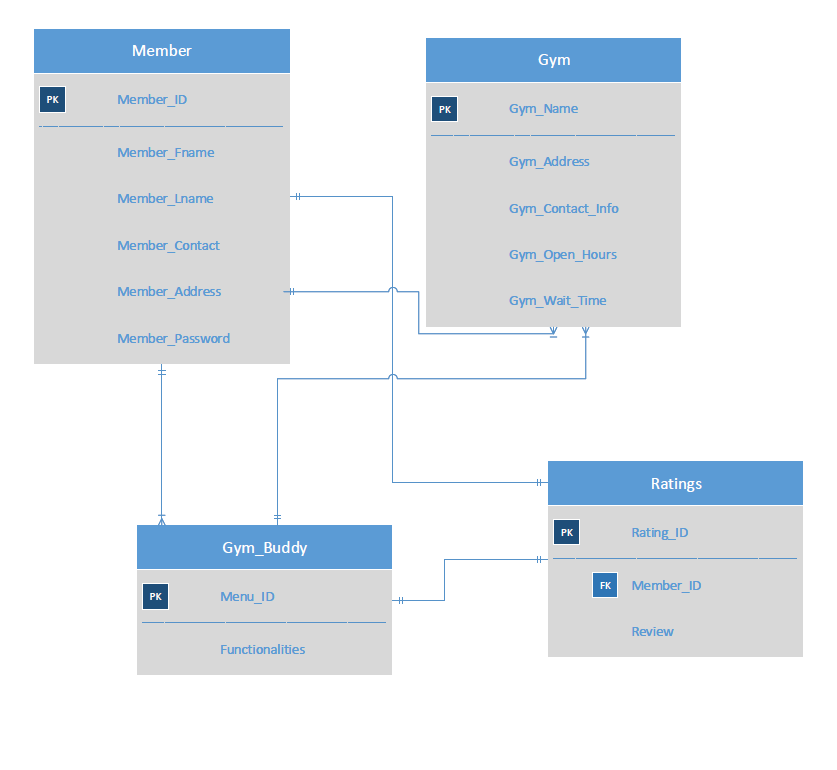
# 

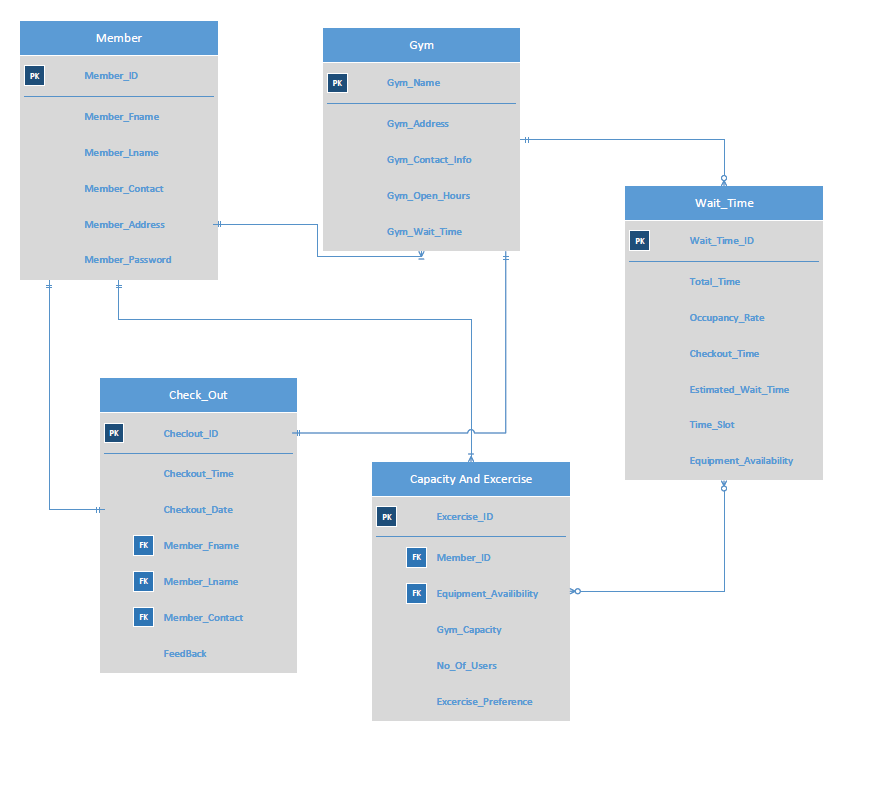
# 

# Design

## 10. Database Design

### Entity Relationship Diagram

****

****

### 

### 

### Database Constraints

#### Member Table Constraints

* Primary Key Constraints/Integrity Constraint on Primary Key ‘ MemberID’ and attribute ‘MemberFirstName’ should not be NULL.
* Uniqueness Constraint on Primary Key ‘MemberID’ and attribute ‘MemberEmailID‘ should be unique.
* Domain Specific / Context Specific Constraints on Attributes ’, ‘MemberEmailID’ and ‘MemberFirstName’ should not have null values.

#### Gym Table Constraints

* Primary Key Constraints/Integrity Constraint on Primary Key ‘GymName’ should not be NULL.
* Uniqueness Constraint on Primary Key ‘GymName’ should be unique.

#### Wait Timetable Constraints

* Primary Key Constraints/Integrity Constraint on Primary Key ‘WaittimeID’ should not be NULL.
* Uniqueness Constraint on Primary Key ‘WaitTimeID’ .
* Referential Integrity Constraint o ‘Gym Name’ which is foreign key in Wait timetable should exist as the primary key in Gym name table

#### Capacity and Exercise table Constraints

* Primary Key Constraints/Integrity Constraint on Primary Key ‘ExerciseID’ should not be NULL.
* Uniqueness Constraint on Primary Key ‘ExerciseID’ and should be unique.
* Domain Specific / Context Specific Constraints Attributes ‘NoOfPeople’ should not have null value.
* Referential Integrity Constraint o ‘MemberID’ which is foreign key in Capacity and Exercise table should exist as the primary key in customer table.

#### 

#### Member Table Constraints

* Primary Key Constraints/Integrity Constraint on Primary Key ‘Member ID’ and attribute ‘Member\_emailAddress’ should not be NULL.
* Uniqueness Constraint on Primary Key ‘Member ID’ and attribute ‘member\_emailAddress ‘should be unique.
* Domain Specific / Context Specific Constraints on Attributes ‘member\_FirstName’, ’member\_LastName’, ‘member\_emailAddress’ should not have null value.

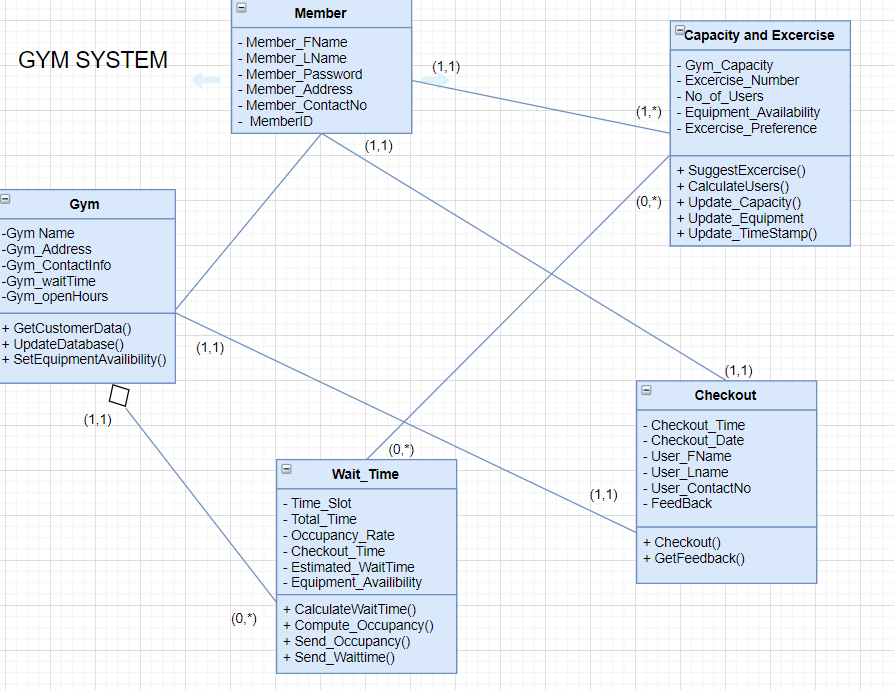
#### Gym\_Buddy Constraints

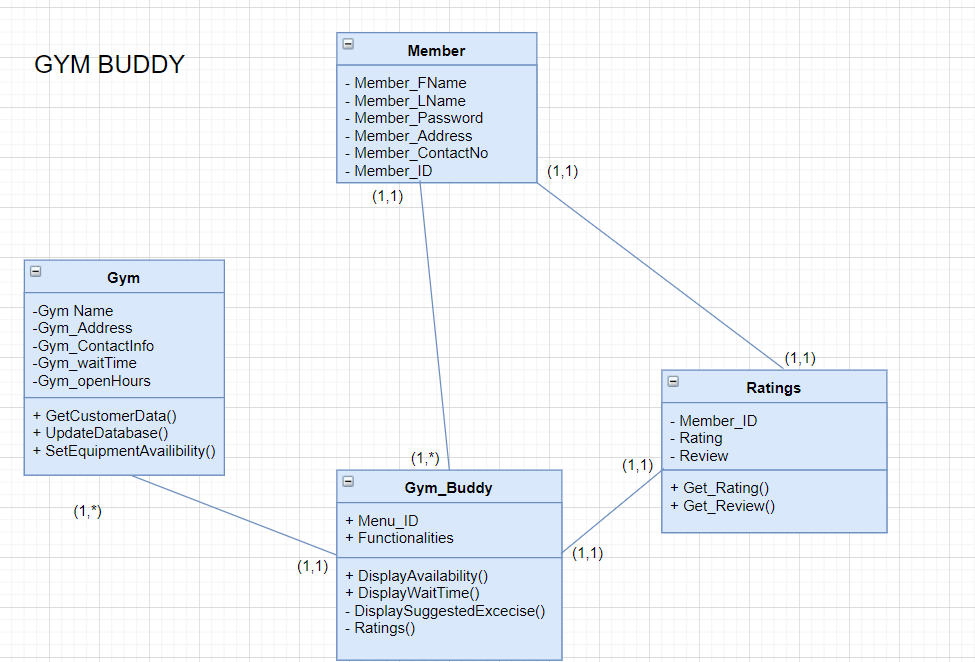
* Primary Key Constraints/Integrity Constraint on Primary Key ‘MenuID’ should not be NULL.
* Uniqueness Constraint on Primary Key Composite Primary Key ‘GymName’ and ‘MenuID’ should be unique.

#### Rating Constraints

* Primary Key Constraints/Integrity Constraint on Primary Key ‘MemberID’ and should not be NULL.
* Uniqueness Constraint on Primary Key ‘MemberID’ should be unique.
* Referential Integrity Constraint on ‘MemberID’ which is foreign key in Rating table should exist as the primary key in member table.

## 11. Complete Class Diagram (with Methods)





## 12. Software Design:

**Signature**

Method Name: Compute Occupancy Rate()

Class Name: Wait Time

ID: WaitTimeID

Clients (Consumers): Members, GymBuddy

Associated Use Cases: Calculate Occupancy Rate, Update Order Information, Update Checkout Information

Description of Responsibilities: Computes Occupancy Rate for Gym

Arguments Received: Table Occupancy, Capacity

Type of Value Returned: Occupancy Percentage

Pre-Conditions: Gym is not completely occupied

Post-Conditions: Sending the calculated Occupancy Rate to GymBuddy

**Logic:**

IF Gym is not FULL

DO (every 5 minutes)

FETCH Occupancy Capacity FROM Table CapacityandExercise

FETCH No of People in the Gym FROM Table checkout

CALCULATE Occupancy Rate = ((No of People in the Gym) /Occupancy Capacity) \* 100

SET the value of “Occupancy Rate” fields in WaitTime Table

UPDATE Occupancy Rate to GymBuddy at regular Intervals

**Signature**

Method Name: Estimated Wait Time()

Class Name: Wait Time

ID: WaitTimeID

Clients : Members, GymBuddy

Associated Use Cases: Calculate Occupancy Rate, Update Checkout Information

Description of Responsibilities: Computes Average Wait Time for Gym

Arguments Received: Time Slot, Time Stamp, Checkout TimeStamp, Total Time Stamp

Type of Value Returned: Calculated Average Wait Time

Pre-Conditions: Gym is completely occupied

Post-Conditions: Sending the calculated Wait Time to GymBuddy

**Logic:**

IF Gym is FULL

FETCH Time FROM System Date

SET Current Time Slot = Time. Timeslot

FETCH Checkout Data FROM Checkout Table

FETCH Equipment Availability Data FROM CapacityandExercise Table

FETCH Occupancy Capacity FROM Table CapacityandExercise

COMPUTE Average Wait Time = ( No of people\* Equipment availability/capacity )\*3600

SET the value of “Average Wait Time” field in WaitTime Table

UPDATE Average Wait Time to GymBuddy at regular Intervals

**Signature**

Method Name: Rate App()

Class Name: RateAPP

ID: Rating

Clients (Consumers): Members

Associated Use Cases: Authorize Sign-in, Write Review

Description of Responsibilities: Provide Rating

Arguments Received: MemberID, Rating No

Type of Value Returned: Rating No ( on a scale of 1 to 5)

Pre-Conditions: Member sign-in is required

Post-Conditions: Displaying the rating in the GymBuddy Menu

**Logic:**

FETCH Menu details FROM GymBuddy table

IF Request type = “RateApp”

THEN DISPLAY Menu with Rating Scale

SET Rating = Member Rating DISPLAY Member Authorization Screen

IF Member THEN

DISPLAY Sign-in Request form

ACCEPT Sign-in Request

PROCESS Sign-in Request

UPDATE GymBuddy Table with App Rating

ELSE

DISPLAY Sign-up Request form with a message “Should be a member to rate a dish”

ACCEPT sign-up request

PROCESS sign-up request

## 13. Minutes of project meetings

**Meeting 1: August 27, 2019 1pm-1:30pm**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attendees:** | Ahnaful Arephin, Vallabhan Kudlu Ramakrishnan, Bhavan Mehta | | |
| **Meeting Topic:** | Member Introduction and Project Topic Discussion | | |
| **Discussion Details:** | Met group members for the first time, introduced each other, discussed potential ideas for project topics | | |
| **Next Meeting:** | September 9, 2019 | | |
| **Items to be completed by Next Meeting:** | Do more research on potential project ideas to present to the professor in his office | | |

**Meeting 2: September 9, 2019 10pm-11pm**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attendees:** | Ahnaful Arephin, Vallabhan Kudlu Ramakrishnan, Bhavan Mehta | | |
| **Meeting Topic:** | Narrow down Project Ideas | | |
| **Discussion Details:** | Discussed project ideas and narrowed down to 3 topics to present to professor | | |
| **Next Meeting:** | September 10, 2019 | | |
| **Items to be completed by Next Meeting:** | Discuss Project Ideas with Professor and Finalize one Topic | | |

**Meeting 3: September 10, 2019 3:30pm-4pm**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attendees:** | Ahnaful Arephin, Vallabhan Kudlu Ramakrishnan, Bhavan Mehta, Professor Srinivasan Raghunathan | | |
| **Meeting Topic:** | Finalize Project Topic | | |
| **Discussion Details:** | Met with Professor and discussed our project ideas, finalized gym idea | | |
| **Next Meeting:** | September 18, 2019 | | |
| **Items to be completed by Next Meeting:** | Discuss core elements of gym idea | | |

**Meeting 4: September 18, 2019 7pm-8pm**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attendees:** | Ahnaful Arephin, Vallabhan Kudlu Ramakrishnan, Bhavan Mehta | | |
| **Meeting Topic:** | Develop Core Ideas for Gym App | | |
| **Discussion Details:** | Discussed potential functionalities for app, Need to meet with the professor for guidance on how to proceed | | |
| **Next Meeting:** | September 19, 2019 | | |
| **Items to be completed by Next Meeting:** | N/A | | |

**Meeting 5: September 19, 2019 10am-10:30am**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attendees:** | Ahnaful Arephin, Vallabhan Kudlu Ramakrishnan, Bhavan Mehta, Professor Srinivasan Raghunathan | | |
| **Meeting Topic:** | Project Core Elements Discussion | | |
| **Discussion Details:** | Met with Professor to discuss core elements of gym app and how to proceed, professor provided advice on what the app should do and how to draw the diagrams digitally | | |
| **Next Meeting:** | November 2, 2019 | | |
| **Items to be completed by Next Meeting:** | * Finalize Project Scope * Discuss Functional Requirements * Create Context and BPMN diagrams | | |

**Meeting 6: November 2, 2019 4:30pm-6:30pm**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attendees:** | Ahnaful Arephin, Vallabhan Kudlu Ramakrishnan, Bhavan Mehta | | |
| **Meeting Topic:** | Context and BPMN Diagrams | | |
| **Discussion Details:** | Discussed and prepared Context and BPMN diagrams | | |
| **Next Meeting:** | November 9, 2019 | | |
| **Items to be completed by Next Meeting:** | * Finalize Context and BPMN diagrams * Create ER Diagram * Create Use Case Diagrams | | |

**Meeting 7: November 9, 2019 1pm-1:30pm**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attendees:** | Ahnaful Arephin, Vallabhan Kudlu Ramakrishnan, Bhavan Mehta | | |
| **Meeting Topic:** | ER Diagram and Use Case Diagrams | | |
| **Discussion Details:** | Discussed and Prepared ER Diagrams and Use Case Diagrams | | |
| **Next Meeting:** | November 11, 2019 | | |
| **Items to be completed by Next Meeting:** | * Finalize ER Diagram * Finalize Use Case Diagram * Prepare Use Case Descriptions | | |

**Meeting 8: November 11, 2019 5pm-7:30pm**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attendees:** | Ahnaful Arephin, Vallabhan Kudlu Ramakrishnan, Bhavan Mehta | | |
| **Meeting Topic:** | Finalize all Diagrams and Prepare Use Cases | | |
| **Discussion Details:** | Discussed and Prepared Use Cases and Data Dictionary | | |
| **Next Meeting:** | November 19, 2019 | | |
| **Items to be completed by Next Meeting:** | Prepare methods and pseudo-code for Software Design section of report | | |

**Meeting 9: November 19, 2019 4pm-6:30pm**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attendees:** | Ahnaful Arephin, Vallabhan Kudlu Ramakrishnan, Bhavan Mehta | | |
| **Meeting Topic:** | Software Design | | |
| **Discussion Details:** | Prepared methods and pseudo-code | | |
| **Next Meeting:** | November 25, 2019 | | |
| **Items to be completed by Next Meeting:** | Interface Design and finalize report | | |

**Meeting 10: November 25, 2019 1pm-2:30pm**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attendees:** | Ahnaful Arephin, Vallabhan Kudlu Ramakrishnan, Bhavan Mehta | | |
| **Meeting Topic:** | Interface Design and Finalize Report | | |
| **Discussion Details:** | Created logo and app name, Created UI design, finalized most sections of report | | |
| **Next Meeting:** | December 5, 2019 | | |
| **Items to be completed by Next Meeting:** | Continue finalizing and proof-reading report, prepare and record presentation | | |

**Meeting 11: December 5, 2019 11am-1:30pm**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attendees:** | Ahnaful Arephin, Vallabhan Kudlu Ramakrishnan, Bhavan Mehta | | |
| **Meeting Topic:** | Finalize Report and Presentation | | |
| **Discussion Details:** | Proof-read report for any errors or inconsistencies, Created presentation and recorded voice lines, edited video and uploaded to YouTube | | |
| **Next Meeting:** | N/A | | |
| **Items to be completed by Next Meeting:** | Submit Report and Presentation Link by 11:59pm on December 6th, 2019 | | |

# References

* Gym Membership Statistics from Statista:
  + <https://www.statista.com/statistics/236123/us-fitness-center--health-club-memberships/>

• “Object-Oriented Systems Analysis and Design” by Jeff Hoffer , Joey George, and Joe Valacich, Pearson Prentice-Hall, Second Edition, 2006