

Workout Day

Gym Management System

Final Report

CPSC 471 Group Project

Group 68

Junyi Li(30113375)
Sipeng He(30113342)
Yihan Wang(30102919)



Abstract

Our goal is to design a comprehensive gym management system that groups the information in an organized way and coordinates the actions of customers, branch managers and the executive manager from the upper level. Multiple features that the operator of a fitness business commonly requires are considered and implemented in our system. We pick React as our front-end framework library and Node.js as our tool to build APIs to achieve an efficient design. We build our database with MongoDB so that the data can be easily stored and fetched. We believe our system has user-friendly interfaces and can help everyone involved in the industry achieve their goals more easily.

Contents

1. Introduction.....	4
2. Project Design.....	5
1) Customer	5
2) Branch Manager	7
3) Executive Manager.....	11
4) EER Diagram	11
3. Project Implementation.....	13
1) Relational Model Diagram	13
2) Implementation Details	15
4. Postman Documentation	22
5. User Guide	23
1) For Customer.....	23
2) For Branch Manager.....	29
3) For Executive Manager.....	38
6. System Requirements.....	43

1. Introduction

Nowadays, many people begin to work out in the gym to pursue a healthy and balanced lifestyle, which has led to the prosperous expansion of the fitness industry. Many big brands run multiple branches nationwide or even internationally and are still expanding at an astonishing speed.

Meanwhile, organizing and making good use of the increasing amount of information related to branches, customers, and products becomes an important issue to be addressed.

Our system implements role-based authentication and authorization so that different participants in the industry can have access to different services. The system provides services that can be convenient for customers, such as ordering products provided by the gym and making an appointment with a coach. The manager of a branch gym can use our system to manage customer orders, staff, and facilities. An executive manager can access information of all the branches owned by the corporation. Therefore, by making good use of the data, our system caters to the need of fitness corporations by helping them to do efficient management and please their customers with convenient features.

2. Project Design

Three different types of users are involved in our system. They are customers, branch managers and executive managers.

1) Customer

Feature: Customer register and login

Customers can register with email, password and other essential information. After the registration is done successfully, the customer will be redirected to the login page. Upon successful login with email and password, the customer will be given a customer token stored in the local storage. They can use the token to access other services provided by the system.

APIs used:

-Post to /customer

Save a customer's information to the database.

-Post to /auth

Login and get the customer JSON web token.

Feature: Customer orders products and services

A customer can browse the products and services provided by the gym, add items to the shopping cart and place an order that includes multiple products in various quantities.

Products provided by the gym are categorized into three types, which are courses, goods and meals. Some services, such as locker rental and membership renewal, can also be regarded as a product. The product's name, price, and type are displayed on the producing browsing section on the shopping page.

Customers can choose the quantity that they need when adding an item to the shopping cart. When the customer adds the same product multiple times, the quantity will be accumulated.

In the shopping cart, the customer can remove an item or add and reduce its quantity. The customer can either finalize the order or clear the shopping cart.

APIs used:

-Get to /product

Retrieve all the products provided by the shop to display

-Post to /order

Save the order placed by the customer to the database.

Feature: Customer manages his or her order

Customers can see their orders on the order page. The products included in the order and their totals are displayed. The customer can choose to cancel an order that hasn't been fulfilled. When the customer attempts to cancel an order that has been fulfilled, the system will deny that request.

APIs used:

-Get to /order/customer/:id

Retrieve all the orders placed by the specific customer.

-Delete to /order/:id

Delete the order specified by the id.

Feature: Customer updates fitness profile

The customer can update his or her fitness file by entering the latest weight, height, BMI and BFP. The time of the newest update will also be recorded and displayed.

APIs used:

-Get to /customer/:id

Retrieve the fitness profile of the specific user.

-Patch to /customer/profile/:id

Update the fitness profile.

Feature: Customer make appointments

By picking a date from the calendar, the customer can get all the available coaches and corresponding time slots scheduled for that day. The customer can make a new appointment, view existing appointments and choose to cancel them.

APIs used:

-Get to /appointment/:date/:id

Get all occupied slots for a specified coach on a specified date.

-Get to /appointment/:id

Get the appointments made by the specified customer.

-Post to /appointment

Save the new appointment made by the customer to the database.

-Delete to /appointment/:id

To delete a specified appointment.

2) Branch Manager

Feature: Branch manager login

The branch manager can log in with his or her unique credentials. Upon successful login, a manager token will be given and stored in the local storage, which can be used to access all the branch management features.

APIs used:

-Post to /branch_manager

Branch manager login and get the manager JSON web token.

Feature: Branch manager manages customer information

The branch manager can see the information of all the customers registered in the system and add, delete and edit any entry of it.

APIs used:

-Get to /customer

Get all customer information.

-Get to /customer/:id

Get the information of a specific customer.

-Put to /customer/:id

Modify the information of a specific customer.

-Delete to /customer/:id

Delete the information of the specified customer.

Feature: Branch manager manages product information

The branch manager can view the information of all the products provided by the gym. A specific product can be easily looked up by id or name using the search bars. Products can also be filtered by their type and stock status. New entries of an item can be added as well.

APIs used:

-Get to /product

Retrieve the data of all the products.

-Get to /product/:id

Retrieve the data of a specific product.

-Post to /product

Create a new product entry.

-Put to /product/:id

Modify the information of a specific product.

-Delete to /product/:id

Delete a product entry from the database.

Feature: Branch manager manages facilities

Two types of facilities are recorded in the system. One is the lockers, and the other one is the common facilities like treadmills. As for each locker, the branch manager can decide whether it is for temporary usage or fixed-term rental. When a customer rents a locker, the manager can record the renter's information.

For other common facilities, the branch manager can add a new entry or delete an existing one. He or she can also edit the information of these facilities, like their current value, condition and so on.

APIs used:

-Get to /facility

Retrieve the data of all facilities.

-Get to /facility/common

Retrieve the data of all common facilities.

-Get to /facility/locker

Retrieve the data of all lockers.

-Post to /facility/common

Add a new entry of common facility.

-Post to /facility/locker

Add a new entry of locker.

-Put to /facility/common/:id

Modify the information of a common facility.

-Put to /facility/locker/:id

Modify the information of a specified locker.

-Delete to /facility/common/:id

Delete the information of a common facility from the database.

-Delete to /facility/locker/:id

Delete the information of a locker from the database.

Feature: Branch manager manages staff

The branch manager can view the information of all the staff who works in the gym branch. Each staff entry can be modified and deleted, and adding a new entry is also allowed.

APIs used:

-Get to /branch_staff

Retrieve all the data of staff.

-Get to /branch_staff/:id

Retrieve the data of a specific branch staff.

-Get to /branch_staff/coach

Retrieve the data of all the staff that are coaches.

-Post to /branch_staff

Add a new branch staff entry to the database.

-Put to /branch_staff/:id

Modify the information of a specific staff.

-Delete to /branch_staff/:id

Delete the information of a specific staff from the database.

Feature: Branch manager manages orders

The branch manager can view all the orders placed by the customers.

When an order is fulfilled, the branch manager can change the fulfillment status of the order from unfulfilled to fulfilled. The manager can also delete an order upon the customer's request.

APIs used:

-Get to /order

Retrieve all the orders.

-Get to /order/:id

Retrieve the information of a specific order.

-Delete to /order/:id

Delete a specific order from the database.

-Patch to /order/:id

Change the fulfillment status.

Feature: Branch manager manages appointments

The branch manager can view all the appointments made by the customers and choose to delete or edit any specific entry.

APIs used:

-Get to /appointment

Retrieve all the appointments in the database.

-Put to /appointment/:id

Modify an appointment made by customer.

-Delete to /appointment/:id

Delete an appointment made by customer from the database.

3) Executive Manager

Feature: Executive manager login

The executive manager can log in with their unique username and password. Upon successful login, a token will be returned and stored locally. It can be used to access the information of all the branch information stored in the system.

-Post to /executiveLogin

Login as an executive manager.

Feature: Executive manager manages branches

The executive manager can view the information of all the branches, like name, location and so on. Our system enables him or her to add new branch entries or modify/delete existing ones.

-Get to /branch

Retrieve all the data of branches stored in the database.

-Get to /branch/:id

Retrieve the data of a specific branch stored in the database.

-Post to /branch

Add a new branch entry to the database.

-Put to /branch/:id

Modify the information of a specific branch.

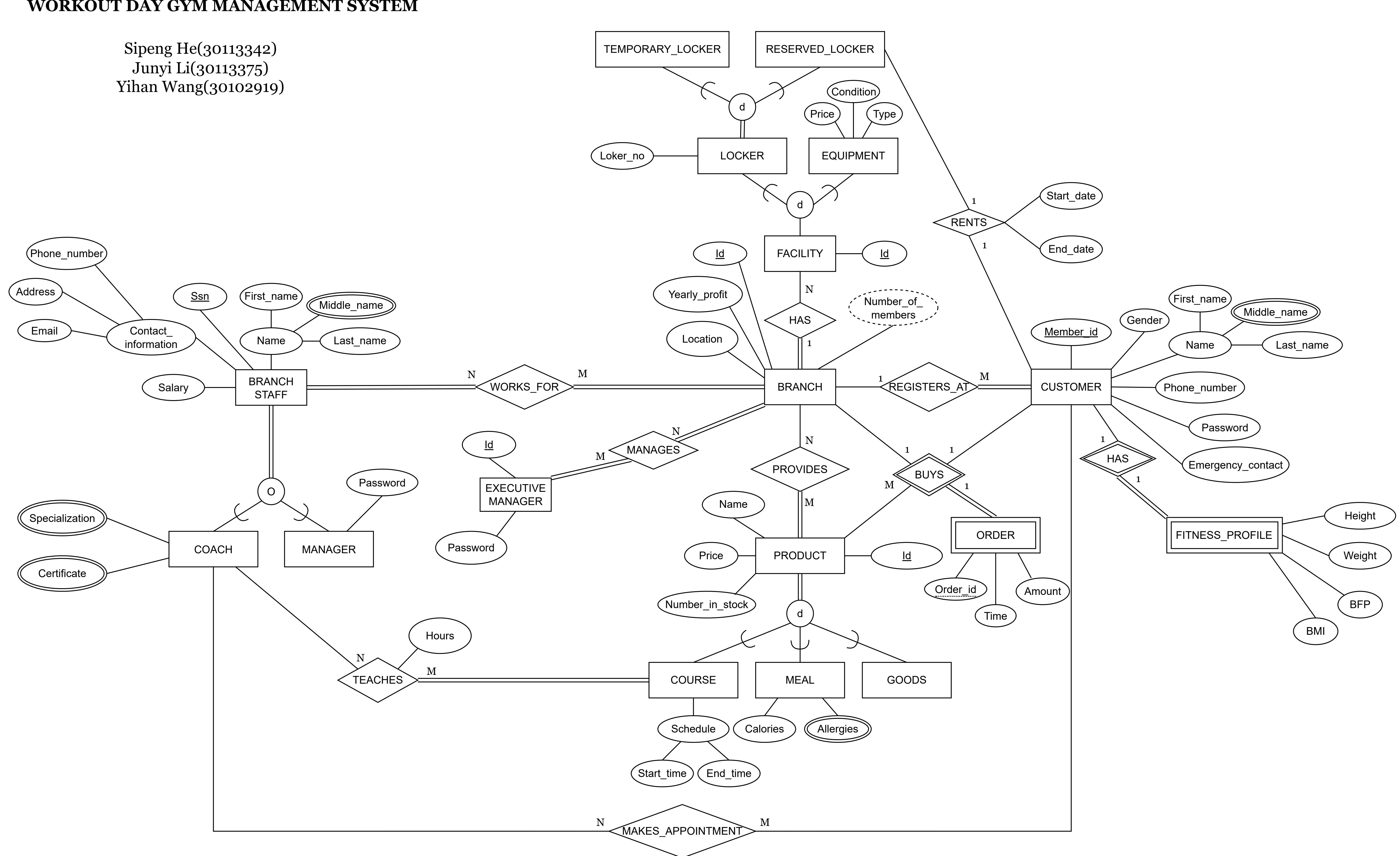
-Delete to /branch/:id

Delete the information of a specific branch from the database.

4) EER Diagram (on the following page)

WORKOUT DAY GYM MANAGEMENT SYSTEM

Sipeng He(30113342)
 Junyi Li(30113375)
 Yihan Wang(30102919)



3. Project Implementation

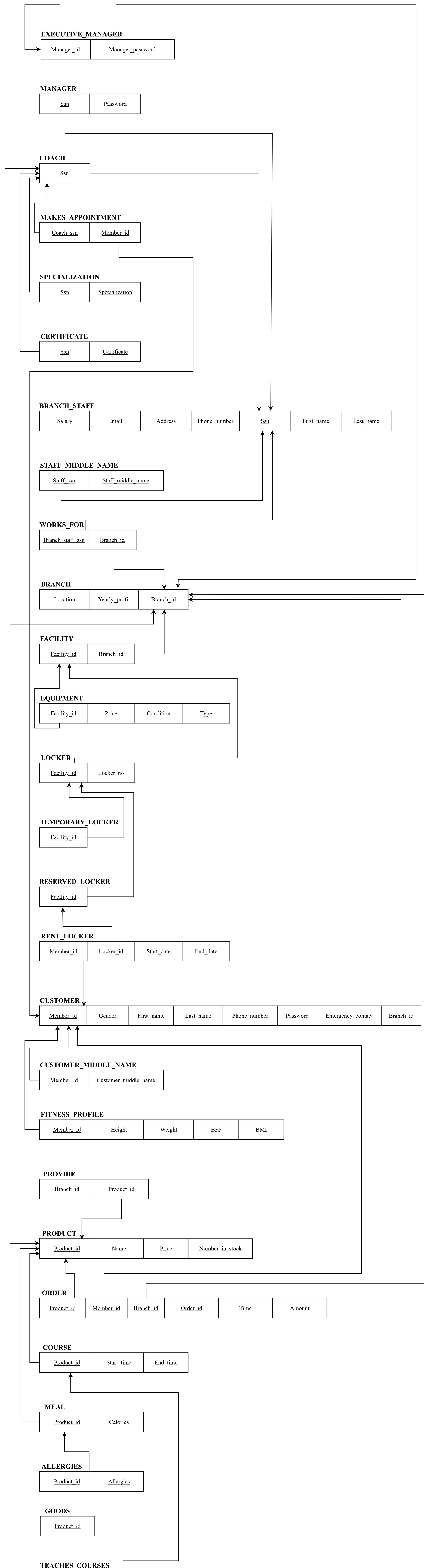
1) Relational Model Diagram (on the following page)

We followed the standard algorithm of transforming the EER diagram into an RM diagram. No special decisions that are worth mentioning are made during this process.

Workout Day Gym Management System

Relational Model

Yihan Wang(30102919)
Junyi Li(30113375)
Sipeng He(30113342)



2) Implementation Details

The DBMS we choose is MongoDB. Since it is a document-oriented, NoSQL database service, we provide details of the model schema we used to retrieve data from the database instead of SQL queries.

(1) Customer

This schema models a customer of the fitness corporation, and is used to retrieve customer-related information from the database so that the branch manager can view and modify the entries. It is also used for customer authentication purposes. Since the customer fitness profile is included as part of the customer information, this model is used to get and update the fitness profile.

Schema:

```
const customerSchema = new mongoose.Schema({  
    gender: {  
        type: String,  
        required: true,  
    },  
    firstName: {  
        type: String,  
        required: true,  
    },  
    midName: {  
        type: String,  
        required: false,  
    },  
    lastName: {  
        type: String,  
        required: true,  
    },  
    phoneNumber: {  
        type: String,  
        required: true,  
    },  
    email: {  
        type: String,  
        required: true,  
    },
```

```

password: {
  type: String,
  required: true,
},
emergencyContact: {
  name: { type: String, required: false },
  phoneNumber: { type: String, required: false },
},
fitnessProfile: {
  height: { type: Number, required: false },
  weight: { type: Number, required: false },
  BFP: { type: Number, required: false },
  BMI: { type: Number, required: false },
  lastUpdateDate: { type: Date, required: false },
},
});

```

(2) Branch manager

This schema models a branch manager with a username and password to log in to the management system. It is used to retrieve login credential data stored in the database for branch manager authentication.

Schema:

```

const branchManagerSchema = new mongoose.Schema({
  username: {
    type: String,
    required: true
  },
  password: {
    type: String,
    required: true
  }
});

```

(3) Executive manager

This schema models an executive manager with a username and password as login credentials.

Schema:

```

const executiveManagerSchema = new mongoose.Schema({
  username: { type: String, required: true },

```

```
password: { type: String, required: true },  
});
```

(4) Branch staff

This schema models a branch staff. Whether the staff is a coach or not is indicated by the “isCoach” attribute. The features for managing branch staff and making and managing appointments depend on this model.

Schema:

```
const branchStaffSchema = new mongoose.Schema({  
  ssn: {  
    type: String,  
    required: true  
  },  
  firstName: {  
    type: String,  
    required: true  
  },  
  middleName: {  
    type: String,  
    required: false  
  },  
  lastName: {  
    type: String,  
    required: true  
  },  
  phone: {  
    type: String,  
    required: true  
  },  
  address: {  
    type: String,  
    required: true  
  },  
  email: {  
    type: String,  
    required: true  
  },  
  salary: {  
    type: Number,  
    required: true  
  },  
  isCoach: {  
    type: Boolean,  
  }}, {  
  timestamps: true  
});
```

```

    required: true
  },
});

});
```

(5) Facility

This schema models the facilities in the gym. A specific facility can be a locker or other kind of equipment. We can also use this model to retrieve the rental status from the database for a rental locker.

Schema:

```

const facilitySchema = new mongoose.Schema({
  isEquipment: {
    type: Boolean,
    required: true
  },
  isTemporaryLocker: {
    type: Boolean,
    required: true
  },
  price: {
    type: Number,
    required: true
  },
  condition: {
    type: Number,
    required: true
  },
  type: {
    type: String,
    required: true
  },
  locker_number: {
    type: Number,
    required: false
  },
  rental: {
    renterId: {type: mongoose.Schema.Types.ObjectId, ref: 'Customer'},
    startDate: {type: Date, required: false},
    endDate: {type: Date, required: false}
  }
});
```

(6) Product

This schema models a product provided by the gym. It can be categorized into one of the courses, meals or other goods, indicated by the corresponding boolean values.

Schema:

```
const productSchema = new mongoose.Schema({
  InStock : {
    type: Number, required: true
  },
  price: {
    type: Number,
    required: true
  },
  name: {
    type: String,
    required: true
  },
  isCourse: {
    type: Boolean,
    required: true
  },
  isMeal: {
    type: Boolean,
    required: true
  },
  isGoods: {
    type: Boolean,
    required: true
  },
  startTime: {
    type: String,
    required: function() {return this.isCourse == true;},
    //required: false
  },
  endTime: {
    type: String,
    //required: false
    required: function() {return this.isCourse == true;},
  },
  courseCoachId:{
    type: mongoose.Schema.Types.ObjectId, ref: 'BranchStaff',
    //required: false
    required: function() {return this.isCourse == true;},
  },
});
```

```

allergies: {
  type: Array,
  //required: false
  required: function() {return this.isMeal == true;}
},
calories: {
  type: Number,
  //required: false
  required: function() {return this.isMeal == true;}
}
});

```

(7) Order

This schema models an order placed by the customer. It contains the information of both the customer who makes the purchase, and the products included. Whether the order is fulfilled or not is indicated by the isFulfilled boolean attribute.

Schema:

```

const orderSchema = new mongoose.Schema({
  products: [
    {
      name: { type: String, required: true },
      price: { type: Number, required: true },
      quantity: { type: Number, required: true },
    },
  ],
  createDate: { type: Date },
  customerId: { type: mongoose.Schema.Types.ObjectId, ref: "Customer" },
  isFulfilled: { type: Boolean },
});

```

(8) Appointment

This schema models an appointment made by the customer to be trained by a coach.

Schema:

```

const appointmentSchema = new mongoose.Schema({
  coachId: {
    type: mongoose.Schema.Types.ObjectId,

```

```
    required: true
  },
  date: {
    type: Number,
    required: true
  },
  slot: {
    type: Number,
    required: true
  },
  customerId: {
    type: mongoose.Types.ObjectId,
    required: true
  }
});
```

(9) Branch

This schema models a branch from the fitness corporation. It is used to retrieve data from all the branches for the executive manager to look at and make modifications.

Schema:

```
const branchSchema = new mongoose.Schema({
  name: {
    type: String,
    required: true,
  },
  location: {
    type: String,
    required: true,
  },
  numberOfMembers: {
    type: Number,
    required: true,
  },
  yearlyProfit: {
    type: Number,
    required: true,
  },
  managerId: {
    type: mongoose.Types.ObjectId,
    ref: "BranchStaff",
  }
});
```

4. Postman Documentation

Appointment API Collection:

<https://documenter.getpostman.com/view/20543500/Uyr5nyXW>

Branch API Collection:

<https://documenter.getpostman.com/view/20543500/Uyr5nyXd>

Branch Manager API Collection:

<https://documenter.getpostman.com/view/20543500/Uyr5nybu>

Branch Staff API Collection:

<https://documenter.getpostman.com/view/20543500/Uyr5nybx>

Customer API Collection:

<https://documenter.getpostman.com/view/20543500/Uyr5nyby>

Executive Manager API Collection:

<https://documenter.getpostman.com/view/20543500/Uyr5nyc1>

Facility API Collection:

<https://documenter.getpostman.com/view/20450033/Uyr5nyc2>

Order API Collection:

<https://documenter.getpostman.com/view/20450033/Uyr5nyc3>

Product API Collection:

<https://documenter.getpostman.com/view/20450033/Uyr5nyc4>

5. User guide

1) For customer

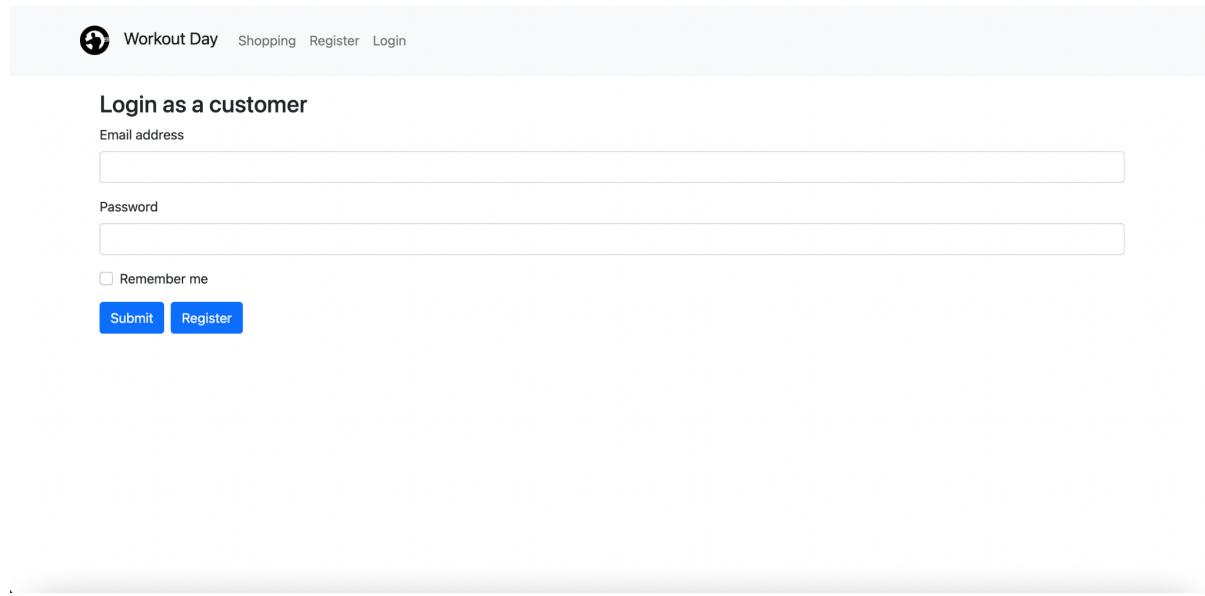
(1) Register

The customer can click on the “Register” button on the navbar to navigate to the register page. Upon successful registration, the customer will navigate to the login page automatically.

The screenshot shows a "Register" form. It includes fields for First name, Last name, Phone number, Gender, Email address, Password, and Confirm Password. Each field has a placeholder text and a corresponding text input box. Below the form is a blue "Register" button.

(2) Login

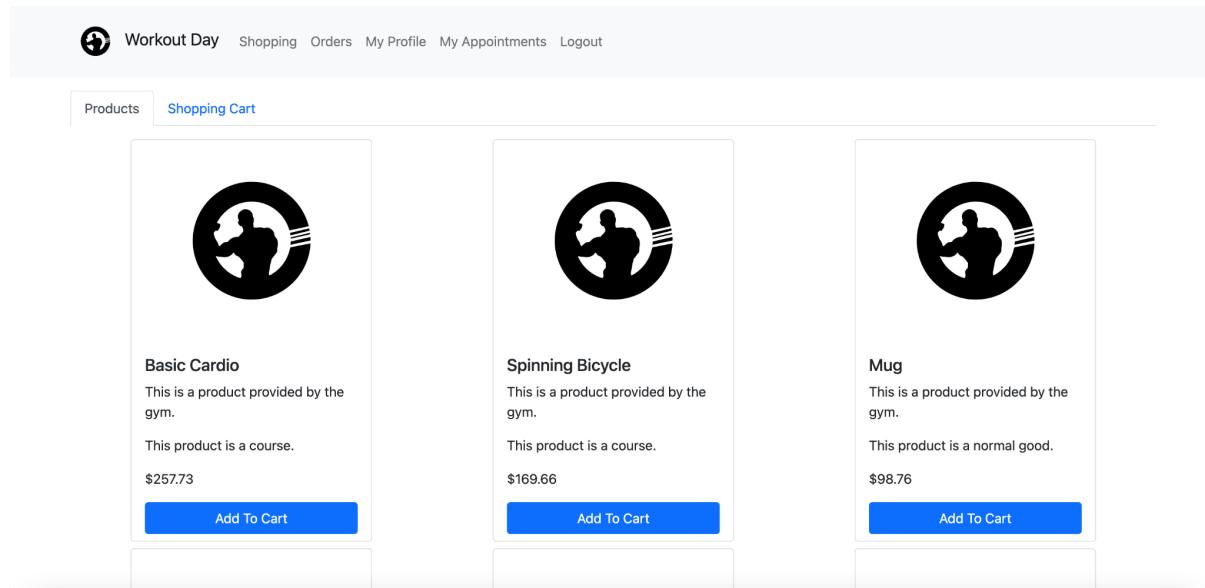
The customer can click on the "Login" button on the navbar to navigate to the Login Page. Fill in the customer registered Email address and password, then click "Submit". If the password or email address is invalid, an error message will pop out.



The screenshot shows a login form titled "Login as a customer". It includes fields for "Email address" and "Password", a "Remember me" checkbox, and "Submit" and "Register" buttons.

(3) Shopping

The customer can click on the “Shopping” button on the navbar to navigate to the Product and Shopping Cart page. Select a product then press “Add to Cart” to add the product to shopping cart.

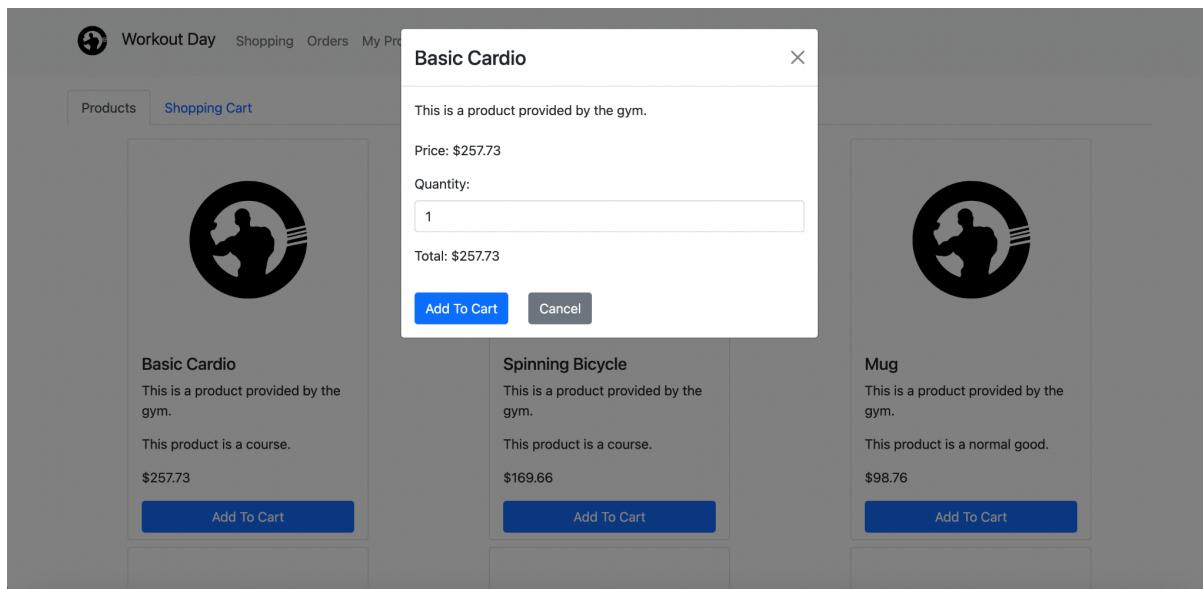


The screenshot shows a products page with three items: "Basic Cardio", "Spinning Bicycle", and "Mug". Each item has a thumbnail, a title, a description, a price, and an "Add To Cart" button.

Product	Description	Price	Action
Basic Cardio	This is a product provided by the gym. This product is a course.	\$257.73	Add To Cart
Spinning Bicycle	This is a product provided by the gym. This product is a course.	\$169.66	Add To Cart
Mug	This is a product provided by the gym. This product is a normal good.	\$98.76	Add To Cart

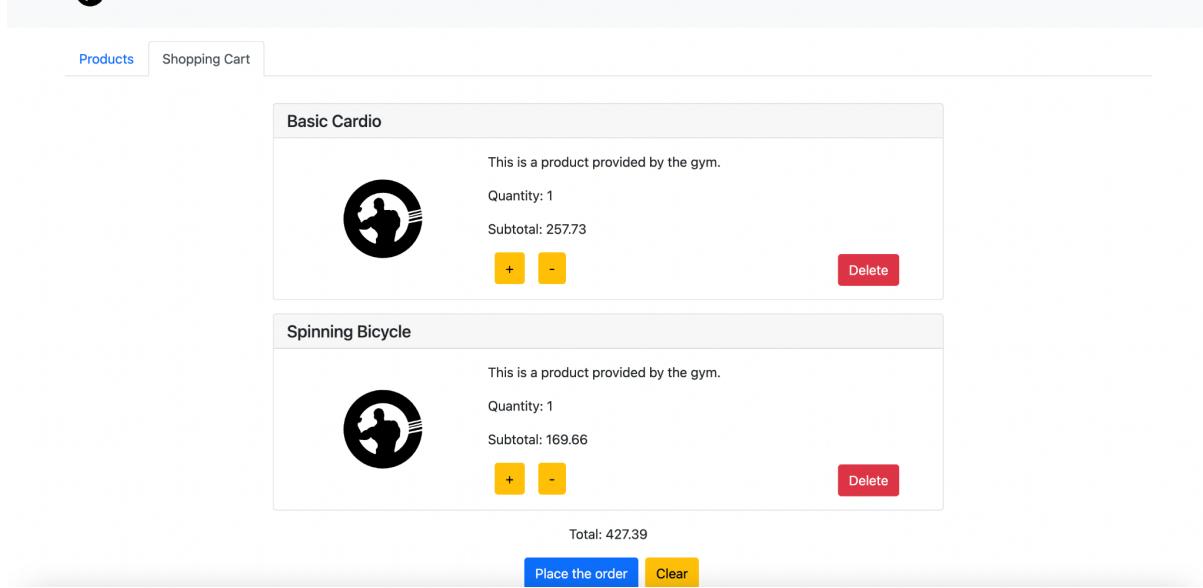
(4) Enter Quantity Number

Fill in the Quantity number and click the button “Add to Cart” to add numbers of products into the shopping cart, or click “Cancel” to cancel.



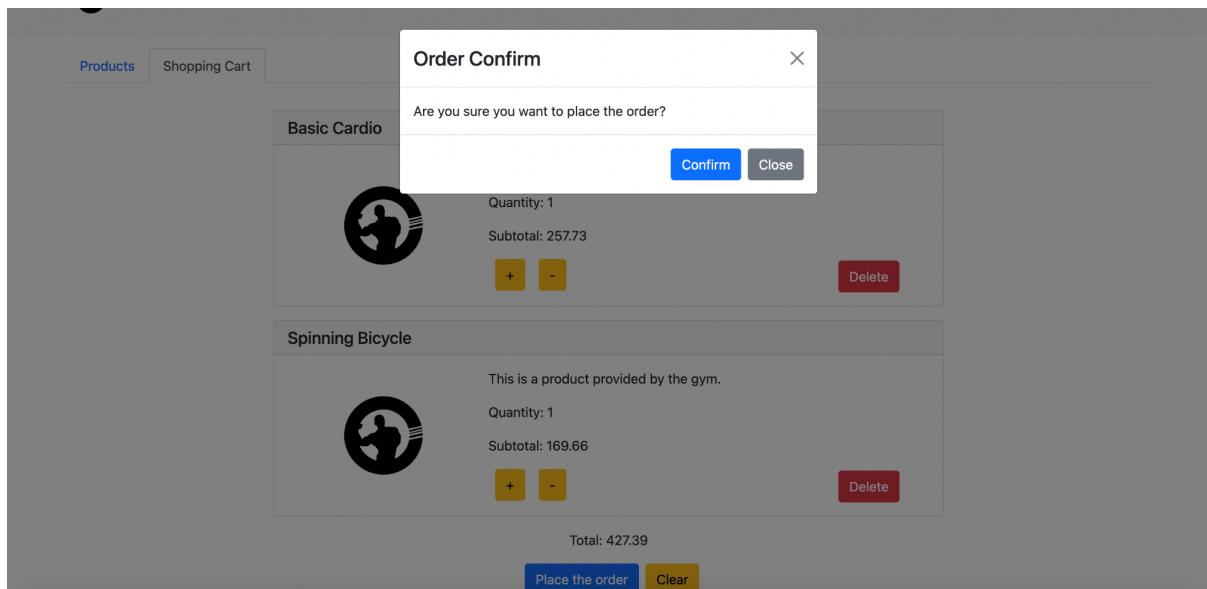
(5) Check Shopping Cart

Click on the “Shopping Cart” tab besides Products to check the current shopping cart. It will display all the products you added to the shopping cart with each quantity, subtotal price and total price.



(6) Place the Order

Click on the button “ Place the order ”at the bottom of the customer’s shopping cart to check out.



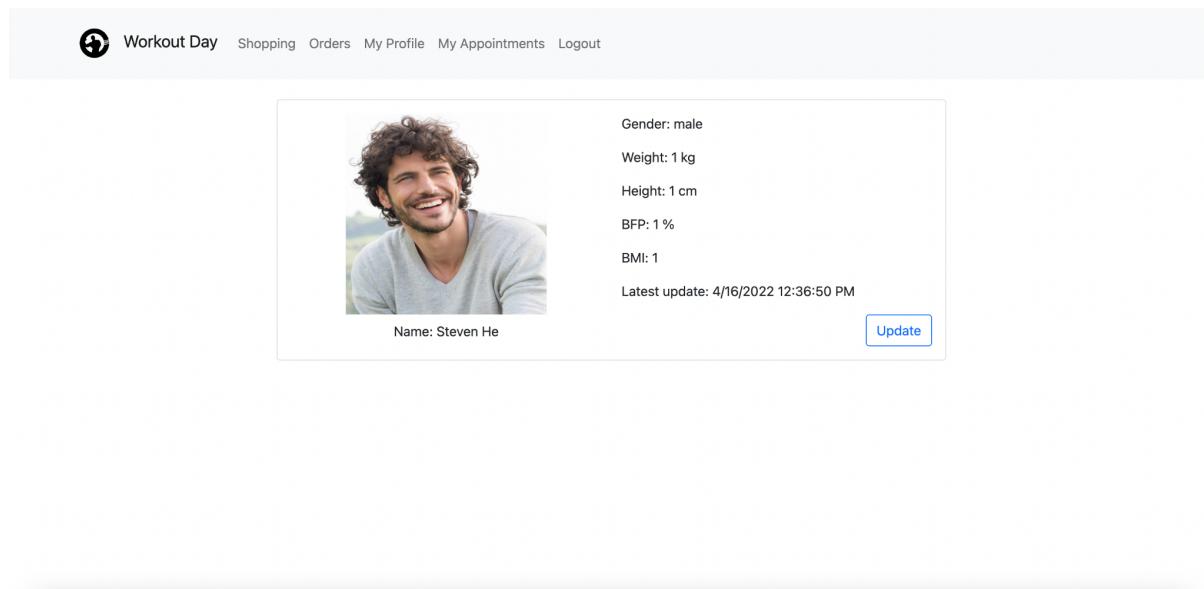
(7) Check Order Status

Click on the “Orders” button on the navbar to navigate to the Orders page. The customer can check orders’ status. The customer can cancel the order if the order is not fulfilled. Click on the button “Cancel the Order” to cancel the order.

Order Details
Basic Cardio *1 Spinning Bicycle *1 Total: 427.39 Date: 2022-04-16T17:18:13.737Z Cancel the Order
Basic Cardio *1 Spinning Bicycle *1 Mug *1 Total: 526.15 Date: 2022-04-16T18:41:29.604Z Cancel the Order

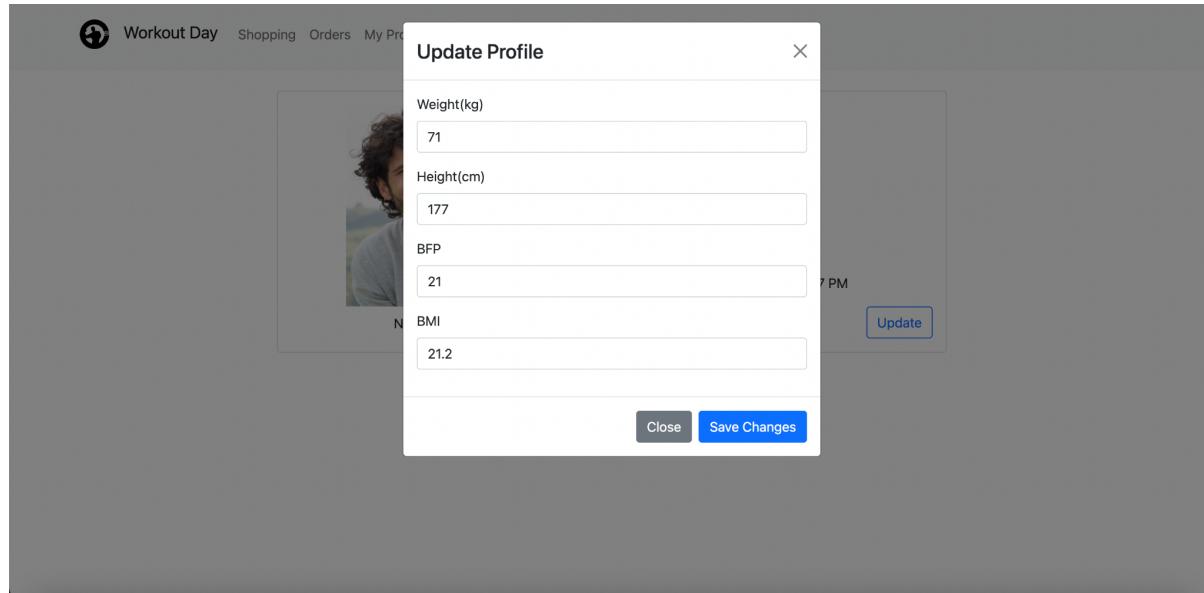
(8) Update Fitness Profile

Click on the “My profile” button on the navbar to navigate to the fitness profile page. Click the update button to update the fitness information.



(9) Update Profile Date

Fill in the Weight(kg), Height(cm), BFP and BMI, click the button “Save Changes” to update your fitness profile, or click the button “Close” to close the pop up window and cancel change.



(10) Make New Appointments

Click on the “My appointments” button on the navbar to navigate to the appointment page. Select the data, coach and timeslot, and press the “make new appointment” button to make a new appointment.

The screenshot shows a user interface for booking a workout appointment. At the top, there is a navigation bar with links: 'Workout Day' (highlighted), 'Shopping', 'Orders', 'My Profile', 'My Appointments', and 'Logout'. Below the navigation bar are three main sections: 'Date' (a calendar for April 2022 showing dates from 28 to 30), 'Coach' (a list of coaches: Daveen Shuter, Odilia Alten, Yena Choi, Huan Zhao, Anqi Liang), and 'Timeslot' (a list of time intervals from 8:00-9:00 to 16:00-17:00). A blue button labeled 'Make an appointment' is positioned between the Date and Coach sections. Below these sections is a header 'My Appointments'.

(11) Manage Appointments

Scroll down the appointment page to see the table that displays all the appointments that the user has made. Click the “Cancel” button at the right to delete an existing appointment.

This screenshot shows the 'My Appointments' page. It features a table listing three scheduled appointments. Each row contains the coach's name (Yena Choi, Anqi Liang, Huan Zhao), the date (Sat Apr 16 2022), the timeslot (13:00 - 14:00), and a red 'Cancel' button. Above the table is a blue 'Make an appointment' button.

Coach	Date	Timeslot	Action
Yena Choi	Sat Apr 16 2022	13:00 - 14:00	<button>Cancel</button>
Anqi Liang	Sat Apr 16 2022	13:00 - 14:00	<button>Cancel</button>
Huan Zhao	Sat Apr 16 2022	13:00 - 14:00	<button>Cancel</button>

(12) Logout

Press the “Logout” button on the navbar to log out as a customer and navigate back to the login page.

Login as a customer

Email address

Password

Remember me

Submit Register

2) For branch manager

(1) Login as a Branch Manager

Manually change the url to /branch to reach the branch management page.

Login using the following credential:

username: root

password: 123123

After successful login, the page will automatically change to the main management page.

The header navigation bar includes a globe icon, the text "Workout Day", and links for "Shopping", "Register", and "Login".

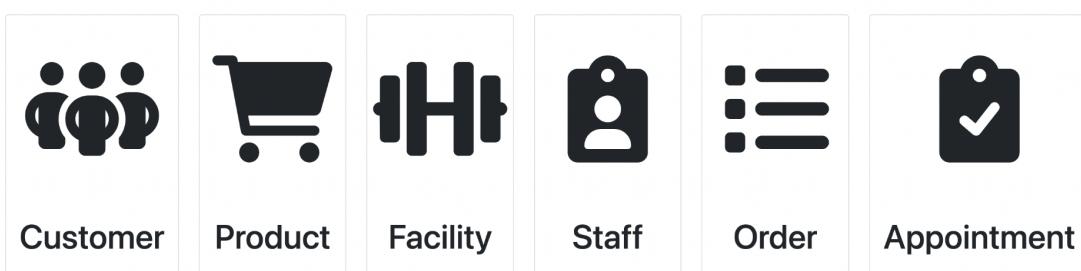
Branch Manager Login

Username

Password

Show Password

The header navigation bar includes a globe icon, the text "Workout Day", and links for "Shopping", "Register", and "Login".



[Logout](#)

(2) Manage Customer Information

Click the “Customer” button on the main page to reach the customer information management page.

Customer

ID	First Name	Last Name	Gender	Phone	Email	Emergency Contact	Emergency Contact Phone	Edit
6258d0c9199dab8fb729eb7b	adsfas	asdfasdf	Female	1231231234	lyne@gmail.com	afasdf	2321234232	<button>Edit</button>
6258d125e79c09bca96e7e5f	adsfadfs	dsafawdf	Male	2132324212	asdfasf@123.com	sadfa	1234212231	<button>Edit</button>
6259a473844c4b0f3f40fb2	Sipeng	He	Male	8221123212	hsp@gmail.com	Steve	2321124252	<button>Edit</button>
6259c14c3822121a2b398c62	Lorant	Bridgman	Male	2566599209	lbridgman0@opensource.org	Bev Boustead	3769773736	<button>Edit</button>
6259c1583822121a2b39955f	Valeda	Mouatt	Female	2533296098	vmouatt1@list-manage.com	Guthrie Wetherby	3659465761	<button>Edit</button>
6259c1583822121a2b399560	Kurt	D'Aulby	Male	6009442747	kdaulby2@google.fr	Kiersten Hache	8143920646	<button>Edit</button>
6259c1583822121a2b399561	Carolyn	Cust	Female	2807972417	ccust3@cdbaby.com	Sileas Fierro	8401828022	<button>Edit</button>
6259c1583822121a2b399563	Mylo	Spiniello	Bigender	4948392438	mspinello5@tuttocitta.it	Darb Rawls	3141513518	<button>Edit</button>
6259c1583822121a2b399564	Edlin	Tinner	Male	9487782416	etinner6@uol.com.br	Ilise Varker	5193435405	<button>Edit</button>
6259c1583822121a2b399565	Velvet	Hordell	Female	9704183361	vhordell7@dailymotion.com	Irwin Krates	2852144942	<button>Edit</button>

« 1 2 3 4 »

[Back](#) [Add](#)

(3) Edit Customer Information

Click on the edit button on the right of an entry to edit the customer information.

Customer

ID
6258d0c9199dab8fb729eb7b
6258d125e79c09bca96e7e5f
6259a473844c4b0f3f40fb2
6259c14c3822121a2b398c62
6259c1583822121a2b39955f
6259c1583822121a2b399560
6259c1583822121a2b399561
6259c1583822121a2b399563
6259c1583822121a2b399564
6259c1583822121a2b399565

« 1 2 3 4 »

[Back](#) [Add](#)

Edit a customer

First Name
adsfas

Last Name
asdfasdf

Gender
Female

Phone Number
1231231234

Email Address
lyne@gmail.com

Emergency Contact Name
afasdf

Emergency Contact Phone Number
2321234232

[Close](#) [Save changes](#) [Delete](#)

(4) Add New Customer Entry

Click the “Add” button on the customer management page to add a new customer entry.

The screenshot shows a 'Customer' management interface. On the left, a table lists customer IDs. A modal window titled 'Add a customer' is open, containing fields for First Name, Last Name, Gender, Phone Number, Email Address, Password, Emergency Contact Name, and Emergency Contact Phone Number. At the bottom of the modal are 'Close' and 'Add' buttons. To the right of the modal is a table titled 'Emergency Contact' with columns for 'Phone' and 'Edit'.

ID
6258d0c9199dab8fb729eb7b
6258d125e79c09bca96e7e5f
6259a473844c4b0f3f40fb2
6259c14c3822121a2b398c62
6259c1583822121a2b39955f
6259c1583822121a2b399560
6259c1583822121a2b399561
6259c1583822121a2b399563
6259c1583822121a2b399564
6259c1583822121a2b399565

« 1 2 3 4 »

[Back](#) [Add](#)

(5) Manage Product Information

Click the “Product” button on the main page to reach the product management page.

The screenshot shows a 'Product' management interface. On the left, two product cards are displayed with their details. On the right, there are search and filter options. The 'Type' filter dropdown is set to 'All', which is highlighted in blue. Other options include 'Meal', 'Course', and 'Goods'. There is also a 'Stock:' dropdown.

```
_id: 6254d36f4984178dc4ea59e2
InStock: 552
price: 257.73
name: Basic Cardio
isCourse: true
isMeal: false
isGoods: false
startTime: 12:00 AM
endTime: 10:50 AM
courseCoachId: 6254972d4984178dc4c7a6f4
allergies: null
calories: null
```



```
_id: 6254d36f4984178dc4ea59ee
InStock: 895
price: 169.66
name: Spinning Bicycle
isCourse: true
isMeal: false
isGoods: false
startTime: 5:30 AM
```

Filter

Type:

All

Meal

Course

Goods

Stock:

(6) Filters and Search Bars

Filters and search bars can be used to find specific products.

Search By Item ID

Search By Name

Filter

Type:

All

Meal

Course

Goods

Stock:

All

In stock

Out of stock

(7) Edit a product entry

Edit a product by clicking on the product card.

The screenshot shows a modal window titled "Edit a product". Inside the modal, there are input fields for Stock (552), Price (257.73), Name (Basic Cardio), Start Date (12:00 AM), End Date (10:50 AM), and Coach ID (6254972d4984178dc4c7a6f4). At the bottom of the modal are three buttons: "Close", "Save changes" (highlighted in blue), and "Delete". To the right of the modal, a sidebar displays a dropdown menu with options: All (selected), Meal, Course, and Goods. Below the sidebar, the word "Stock:" is visible.

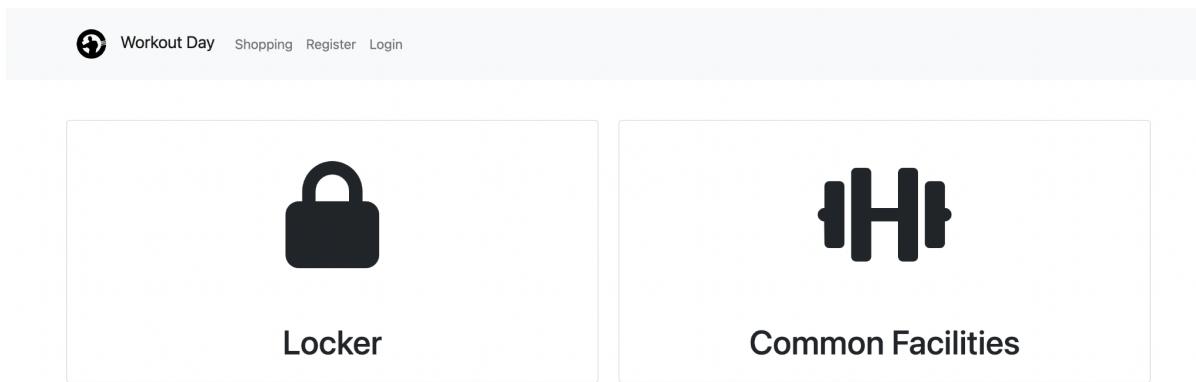
(8) Add a New Product

Click on the “Add” button at the bottom of the page to add a new product entry.

This screenshot is identical to the one above, showing the "Edit a product" modal with the same fields and button layout. The sidebar on the right also shows the "All" category selected in the dropdown menu.

(9) Manage the Facilities

Click on the “Facility” button on the main page to navigate to the facility management section.



(10) Mange Lockers

Click on the “Locker” button to enter the locker management page. Click the “Edit” button on the right to edit an entry, and click the add button at the bottom to add an entry.

Facility ID	Locker Number	Is Temporary Locker	Price	Condition	Renter ID	Start Date	End Date	Edit
6254b6ce4984178dc4da2299	985	No	198.31	1	6258fc12de0cdc83a2dcd28a	1970-01-01	1970-01-01	<button>Edit</button>
6254b6ce4984178dc4da2295	167	No	757.65	9	6254ad89fc13ae3195000092	2022-03-20	2022-03-20	<button>Edit</button>
6254b6ce4984178dc4da229c	192	No	92.74	2	6254ad89fc13ae3195000096	1970-01-01	1970-01-01	<button>Edit</button>
6254b6ce4984178dc4da2298	163	No	384.24	4	6254ad89fc13ae3195000096	2022-12-22	2022-12-22	<button>Edit</button>
6258fdecf6c84d939fa968f8	233	Yes	10	2				<button>Edit</button>
6258fe62f6c84d939fa9690c	223	Yes	20	3				<button>Edit</button>
6258ff38ae2484c7c071c68c	232	Yes	120	5				<button>Edit</button>
6258ffbbae2484c7c071c693	123	No	20	4	6258d0c9199dab8fb729eb7b	2010-01-03	2010-01-03	<button>Edit</button>
62599f46ebf5791a40c9c012	204	No	532.43	7	6254ad89fc13ae3195000096	1970-01-01	1970-01-01	<button>Edit</button>
6259c4c7ebf5791a40c9c01b	204	Yes	532.43	7				<button>Edit</button>

« 1 2 »

[Back](#) [Add](#)

(11) Manage Common Facilities

Click on the “Common Facility” button to enter the common facility management page. Click the “Edit” button on the right to edit an entry, and click the add button at the bottom to add an entry.

Common Facility

Facility ID	Name	Price	Condition	Edit
6254b6ce4984178dc4da229e	Treadmill	252.61	7	<button>Edit</button>
6254b6ce4984178dc4da22a0	Rowing machine	532.43	7	<button>Edit</button>
6254b6ce4984178dc4da22a6	Dumbbell set	749.45	2	<button>Edit</button>
6254b6ce4984178dc4da22a8	Ellipticals	727.93	9	<button>Edit</button>
6254b6ce4984178dc4da22a2	Stationary bicycle	98.79	6	<button>Edit</button>
6254b6ce4984178dc4da229f	Stationary bicycle	188.05	2	<button>Edit</button>
6254b6ce4984178dc4da22a3	Rowing machine	507.07	9	<button>Edit</button>
62578454a8d8597fb3a39141	vr/ar goggles	198.31	1	<button>Edit</button>
6258cebd8d4323d2dd84569b	ABCD	1230	3	<button>Edit</button>
6259dad4931d264d06f6bb19	Dumbbell	10	2	<button>Edit</button>

« 1 2 »

[Back](#) [Add](#)

(12) Manage Staff

Click the “Staff” button on the main page to navigate to the staff management page. Click the “Edit” button on the right to edit an entry, and click the add button at the bottom to add an entry.

Staff

ID	First Name	Last Name	SSN	Email	Phone	Address	Salary	Is Coach	Edit
6257329efc13ae0aff000002	Clementia	Malthouse	286-26-3121	bmalhouse2@sbwire.com	1232212231	34 Oak Valley Junction	1111	No	<button>Edit</button>
6257329efc13ae0aff000004	Daveen	Shuter	573-40-7956	cshuter4@photobucket.com	2151984277	533 Shopko Point	38031	Yes	<button>Edit</button>
6257329efc13ae0aff000005	Odilia	Alten	897-12-2255	balten5@timesonline.co.uk	6049598758	141 Rockefeller Park	3539.32	Yes	<button>Edit</button>
625934f66cbca8f7daa0fd85	Yena	Choi	789-44-4848	yenaChoi@yuehua.com	7783884700	45321 19 Ave	112	Yes	<button>Edit</button>
6259db66931d264d06f6bb1f	Huan	Zhao	123-23-1234	asdfa@gmail.com	1231231234	asdfasfasfads	2322	Yes	<button>Edit</button>
6259f8bb16a5b6030fba4f30	Anqi	Liang	436-16-5421	"391843294@sbwire.com	3313820491	11956 11 Ave, Burnaby, BC	3142	Yes	<button>Edit</button>

« 1 »

[Back](#) [Add](#)

(13) Manage Orders

Click the “Order” button on the main page to navigate to the order management page. Click the “Edit” button on the right to change the fulfillment status of an order.

Order ID	Date	Products	Customer ID	Fulfilled	Edit
6258e9143822121a2bad3897	1970-01-01	headband, Running Hat	625a7206a7897cc5f5dc6c86	false	Edit
6258e9143822121a2bad3898	1970-01-01	Pilates, HIIT Course, Low Carbs Meal	625a7206a7897cc5f5dc6c86	true	Edit
6258e9143822121a2bad3899	1970-01-01	Low Carbs Meal	6254be7bfc13ae31950000d6	true	Edit
6258e9143822121a2bad389a	1970-01-01	T-Shirt, Basic Cardio, Gluten-free Meal, Vegetarian Meal, AR/VR Cardio, Advanced HIIT	6254be7bfc13ae31950000d6	true	Edit
6258e9143822121a2bad389b	1970-01-01	Paleo Diet Meal, Jump Rope HIIT	6254be7bfc13ae31950000d6	false	Edit
6258f5b7bad98d1a7e7eb5de	2022-04-15	Paleo Diet Meal	6254be7bfc13ae31950000d6	false	Edit
6258fff36b3ab1bc8ff49057	2022-04-15	T-Shirt, Basic Cardio	6258d0c9199dab8fb729eb7b	false	Edit
625900e46b3ab1bc8ff49064	2022-04-15	T-Shirt, Basic Cardio, Gluten-free Meal	6258d0c9199dab8fb729eb7b	false	Edit
625900f86b3ab1bc8ff4906a	2022-04-15	T-Shirt, AR/VR Cardio	6258d0c9199dab8fb729eb7b	false	Edit
62596cf1f8a604b94389bd7a	2022-04-15	Basic Cardio, Spinning Bicycle, Running Hat	6258d0c9199dab8fb729eb7b	false	Edit

« [1](#) [2](#) »

[Back](#)

(14) Appointment Management

Click on the “Appointment” button on the main page to navigate to the appointment management page. Click the edit button on the right to edit or delete an existing appointment.

Appointment

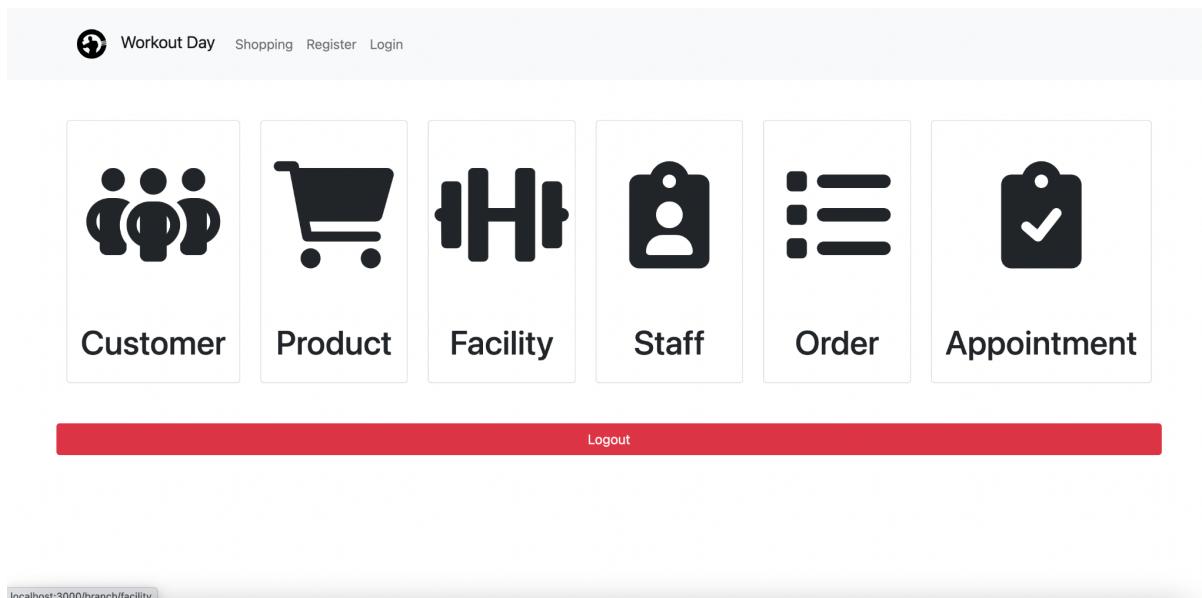
Appointment ID	Coach ID	Date	Slot	Customer ID	Edit
625a5e773822121a2b9f70c3	6259db66931d264d06f6bb1f	Apr 16 2022	14:00-15:00	6258d0c9199dab8fb729eb7b	Edit
625a88f4794c5f7393016e55	625934f66cbca8f7daa0fd85	Apr 16 2022	9:00-10:00	6258d0c9199dab8fb729eb7b	Edit
625a8922794c5f7393016e5c	6257329efc13ae0aff000005	Apr 16 2022	11:00-12:00	6258d0c9199dab8fb729eb7b	Edit
625a89c1794c5f7393016e65	625934f66cbca8f7daa0fd85	Apr 16 2022	11:00-12:00	6258d0c9199dab8fb729eb7b	Edit
625a89e2794c5f7393016e6f	6259db66931d264d06f6bb1f	Apr 16 2022	12:00-13:00	625a4e81324a52147c185c85	Edit
625a8a31794c5f7393016e7a	6257329efc13ae0aff000005	Apr 10 2022	9:00-10:00	625a4e81324a52147c185c85	Edit
625a8a40794c5f7393016e81	625934f66cbca8f7daa0fd85	Apr 17 2022	10:00-11:00	625a4e81324a52147c185c85	Edit
625a9376ae7ca3d3f2052514	625934f66cbca8f7daa0fd85	Apr 16 2022	10:00-11:00	625a4e81324a52147c185c85	Edit
625afc31924c6354e564578	625934f66cbca8f7daa0fd85	Apr 16 2022	13:00-14:00	625afa421924c6354e564547	Edit
625afc131924c6354e564588	6259f8bb16a5b6030fba4f30	Apr 16 2022	13:00-14:00	625afa421924c6354e564547	Edit

« [1](#) [2](#) »

[Back](#)

(15) Logout as Branch Manager

Press the “Logout” button on the main page to log out as a branch manager.



3) For Executive Manager

(1) Login as Executive Manager

Manually change the url to /executive to reach the executive management page. Login using the following credential:

username: executive

password: executive

After successful login, the page will automatically change to the executive management page.

 Workout Day Shopping Register Login
Login as executive manager

Username

Password

 Workout Day Shopping Register Login

Showing 5 branch(es).

Name	Yearly profit	Number of members	Location	Manager ID	Edit	Delete
Burnaby West	100010000	50	4728 Dawson St, Burnaby, BC	6257329efc13ae0aff000002	Edit	Delete
Confederation	300000	59	1234 Banff Trail NW, Calgary, AB	6257329efc13ae0aff00000f	Edit	Delete
Market Mall	1000000	1000	3022 Market Mall NW, Calgary, AB	6257329efc13ae0aff000002	Edit	Delete
Surrey	100000100	50	14954 90 Ave, Surrey, BC	6257329efc13ae0aff000002	Edit	Delete

[1](#) [2](#)

(2) Modify Branch Information

Click the button "Edit" corresponding to the branch you want to modify information. Fill in the data you want to change and click the button "Save Changes" to save changes, or click the button "Close" to quit the pop-up window and cancel changes.

Name	Manager ID
57329efc13ae0aff000002	Edit
57329efc13ae0aff00000f	Edit
57329efc13ae0aff000002	Edit
57329efc13ae0aff000002	Edit

(3) Add New Branch

Click the button “Add” at the bottom of the table of branches. Fill in Name, Location, Number of members, Yearly Profit and Manager Id. Then click the button “Save” to save data and create a new branch, or click the button “Close” to close the pop-up window and cancel to add a new branch.

Name	Manager ID
57329efc13ae0aff000002	Edit
57329efc13ae0aff00000f	Edit
57329efc13ae0aff000002	Edit
57329efc13ae0aff000002	Edit

(4) Delete Specific Branch

Click on the button “Delete” corresponding to the branch you want to delete. Then click on the button “Confirm” to confirm deletion, or click on the button “Close” to close the pop up window to cancel deletion.

Delete

(5) Sorting

By clicking the column header, the table will sort by the column you choose. \uparrow represent sorting in ascending order, \downarrow represent sorting in descending order.

Name \uparrow

Burnaby West

Confederation

Market Mall

Surrey

(6) Log out Executive Manager

Click on the button “Log out” to log out the executive manager. The user will be redirected to the executive manager login page automatically.

The screenshot shows a web page titled "Login as executive manager". At the top, there is a navigation bar with a logo, the text "Workout Day", and links for "Shopping", "Register", and "Login". Below the title, there are two input fields: one for "Username" and one for "Password", both represented by simple text input boxes. At the bottom of the form is a blue "Submit" button.

6. System Requirements

The following dependencies are needed for our front-end code:

```
"dependencies": {  
    "@fortawesome/fontawesome-svg-core": "^6.1.1",  
    "@fortawesome/free-regular-svg-icons": "^6.1.1",  
    "@fortawesome/free-solid-svg-icons": "^6.1.1",  
    "@fortawesome/react-fontawesome": "^0.1.18",  
    "@testing-library/jest-dom": "^5.16.2",  
    "@testing-library/react": "^12.1.4",  
    "@testing-library/user-event": "^13.5.0",  
    "axios": "^0.26.1",  
    "babel-plugin-macros": "^3.1.0",  
    "bootstrap": "^5.1.3",  
    "formik": "^2.2.9",  
    "lodash": "^4.17.21",  
    "react": "^17.0.2",  
    "react-bootstrap": "^2.2.1",  
    "react-calendar": "^3.7.0",  
    "react-dom": "^17.0.2",  
    "react-router-dom": "^6.2.2",  
    "react-scripts": "5.0.0",  
    "web-vitals": "^2.1.4",  
    "yup": "^0.32.11"  
}
```

Install these dependencies with command:

```
npm install @fortawesome/fontawesome-svg-core  
@fortawesome/free-regular-svg-icons @fortawesome/free-solid-svg-icons  
@fortawesome/react-fontawesome @testing-library/jest-dom  
@testing-library/react @testing-library/user-event axios  
babel-plugin-macros bootstrap formik lodash react react-bootstrap  
react-calendar react-dom react-router-dom react-scripts web-vitals yup
```

As for the back-end, we build the APIs using Node.js, and our data is stored on MongoDB Atlas.