

# Table of Contents

## Contents

<b>Requirement Analysis .....</b>	5
<b>Project Objectives .....</b>	6
<b>Problem Statement .....</b>	6
<b>Scope of the Project .....</b>	7
<b>Project Vision .....</b>	7
<b>Business Constraints.....</b>	8
<b>Technical Constraints.....</b>	8
<b>Documentation .....</b>	9
<b>Use Case Analysis .....</b>	10
Mobile Application Use Case List Table .....	11
Mobile Use Case Diagram .....	13
Mobile Use Case Dependency Diagram .....	14
Desktop Application Use Case List Table .....	15
Desktop Use Case Dependency Diagram .....	18
Web Application Use Case List Table.....	19
Web Use Case Diagram .....	20
Web Use Case Dependency Diagram .....	21
Mobile Application Narratives, Forms, and Sequence Diagrams .....	22
Use Case: Login .....	22
use Case: Plan Workout .....	24
Use Case: Filter Moves.....	26
Use Case: Advised Workout.....	29
Use Case: Time Range.....	31
Use Case: Workout Content .....	32
Use Case: Current Workout.....	34
Use Case: Finish Workout.....	36
Use Case: Workout Content .....	38
Use Case: Previous Workouts .....	40
Use Case: Time Range .....	42
Use Case: Moves' Details .....	43

Use Case: Add to Current Workout .....	45
Use Case: Equipment Location.....	47
Use Case: Order .....	49
Use Case: View Items.....	51
Use Case: Filter Items .....	53
Use Case: Add Items .....	55
Use Case: View Order .....	57
Use Case: Clear Order .....	59
Use Case: Confirm Order.....	61
Use Case: PT Advertisements .....	63
Use Case: Contact PT .....	65
Use Case: Machine Help.....	68
Use Case: Scan QR Code.....	69
Use Case: Moves Preview.....	71
Use Case: Feedback .....	73
Use Case: Assign Workout.....	75
Use Case: Select Moves .....	77
Use Case: Send Report.....	79
Desktop Application Narratives, Forms, and Sequence Diagrams.....	81
Use Case: Log in .....	81
Use Case: View Members.....	83
Use Case: Add Members .....	85
Use Case: Edit Members.....	87
Use Case: Remove Members .....	89
Use Case: View Coaches.....	91
Use Case: Add Coaches .....	93
Use Case: Edit Coach .....	95
Use Case: Remove Coach .....	97
Use Case: View Machines.....	99
Use Case: Add Machines .....	101
Use Case: Edit Machines .....	103
Use Case: Remove Machines .....	105
Use Case: View Workouts .....	107

Use Case: Add Workouts .....	109
Use Case: Edit Workouts .....	111
Use Case: Remove Workouts.....	113
Use Case: View Menu .....	115
Use Case: Add Items .....	117
Use Case: Remove Items .....	119
Use Case: Edit Menu.....	121
Use Case: View Bartenders.....	123
Use Case: Edit Bartenders .....	125
Use Case: Add Bartender .....	127
Use Case: Remove Bartender .....	129
Use Case: View Reports .....	131
Use Case: View Orders.....	133
Use Case: Confirm Order.....	135
Use Case: Order Ready.....	137
Use Case: Order Served .....	139
Use Case: Order Unserved.....	141
Use Case: Order Denied .....	143
Use Case: View Menu .....	145
Use Case: Add Items .....	147
Use Case: Remove Items .....	149
Use Case: Edit Menu.....	151
Use Case: Login .....	153
Web Application Narratives, Forms, and Sequence Diagrams.....	155
Use Case: Navigate .....	155
Use Case: Register .....	157
Use Case: Confirm Email .....	159
Use Case: Login .....	161
Use Case: Forgot Password .....	163
Use Case: Workout History.....	166
Use Case: Contact.....	168
Use Case: View Profile .....	170
Use Case: Edit Profile .....	172

<b>Data Modeling .....</b>	174
Data Context Diagram .....	175
Conceptual Data Diagram .....	176
Physical Data Diagram .....	177
<b>Process Modeling.....</b>	178
Mobile Application Event Table .....	179
Desktop Application Event Table .....	182
Web Application Event Table .....	186
Desktop Application Decomposition Diagram .....	189
Web Application Decomposition Diagram .....	190
Mobile Single Logical and Physical DFDs.....	191
Actor: Member .....	191
Actor: Coach .....	200
Desktop Single Logical and Physical DFDs .....	203
Actor: Admin.....	203
Actor: Bartender .....	217
Web Single Logical and Physical DFDs .....	220
Actor: Member .....	220
Actor: Guest.....	225

# **Requirement Analysis**

## **Project Objectives**

This project will develop new business processes and supporting information system processes and services to support the strategic vision for myGym member services. It is anticipated that the resulting system will provide for highly integrated processes and services that cross many internal business functions and reach out directly to customers. It is anticipated that this project will result in at least one of the following (listed in order of expected likelihood).

1. Develop an in-house information system that aims to further integrate more tech into the gym industry, keeping it as parallel with the tech industry as possible.
2. Create a modern interface that users can access to enhance their experience at the gymnasium, be it pre-workout, during the workout, or post-workout.
3. Provide the facility with a reliable tool with a focus on functionality, that facilitates the process of managing its various aspects.

## **Problem Statement**

Over the past decade, fitness and the trend of bodybuilding have been on the rise, the fitness industry expanded rapidly and is still expanding, however, the integration of new technology has been kept rather basic, so the gap between both industries kept on growing. The human dependence on technology is immense, hence the fitness industry can greatly benefit from implementing more tech. The following specific problems were internally discussed by the members of the project team:

1. Current available systems only focus on the management side of things, without adding anything to the user experience.
2. Currently-used systems have limited and basic functions, they are centered around members' registration and subscriptions' renewal.
3. Multiple employees are assigned to keep track of subscriptions, and that's due to the fact that these are tracked on either papers or a poorly-designed system that's not easy to use.
4. The need for plenty of material resources and storage space, for example: member files and the storage drawers and rooms needed.
5. Difficulty of communication between the gym members and the coaches, where the member has to talk to the coach multiple times per session to get the workout right.
6. Members fail to keep track of their workout routines, thus wasting both their time and the coaches.
7. Gym members find difficulty in describing and remembering their previous workout or exercise, resulting in some confusion on the coach's side.
8. Members are not aware of the coaches' backgrounds, resulting in him/her not knowing which coach to seek for the goal they're trying to achieve.

## Scope of the Project

This cross-functional project will support or impact the following business functions and external parties:

1. Subscriptions
2. Information storing (Employee databases)
3. Accounts payable
4. Member services
5. External parties:
  - Prospective members
  - Members
  - Contracted Gym Owners
  - Gym Employees
  - Independent Personal Trainers

It is recognized that the project scope may need to be refined over the course of the project. The project scope should be defined as explicitly as possible in the first phase of the project. Any significant deviation of functionality, or timetable must be reported promptly to the appropriate supervisor. That supervisor must promptly request and facilitate a scope change consensus meeting.

## Project Vision

The strategic plan for this project is to fulfill the following:

1. Introduce Mobile and Web applications which keep the members in mind and improve their overall gym experience.
2. Add new innovative and modern features to the system, such as identifying machines and how to use them through QR codes, workout planning, pre-ordering from the bar, and so on...
3. Implement an advanced yet simple-to-use management system that can be operated by a few people with barely any knowledge in computers.
4. Digitalizing most of the data needed to run the gymnasium, eliminating the need for most of the physical resources, thus saving time and money in the long run.
5. By including information about each of the machines and their respective workouts, members will not need to come up to the gym coaches as often for help.
6. Members will have access to features that help them keep track of their daily workouts, thus making it easier for them to see their progress. This also helps the coaches point the members in the right direction simply by looking at the members' previous workouts.
7. By tracking the workout on each user's account, coaches will no longer misinterpret what members are trying to describe, as all moves and workouts are professionally labeled on the system in a way that coaches will be able to understand quickly.

8. All of the relevant employees' backgrounds will be available to the members through the system, giving them enough information to know who is best suited to help them accomplish their goals.

## **Business Constraints**

1. The initial version of the system must be operational within three months. The need for subsequent versions will be discussed later on.
2. The database must be updated overtime as to include all the data that was present prior to the implementation of myGym system.
3. The system must be maintained and monitored for the duration of the contract post-installment.
4. A level of risk tolerance must be established to prevent the loss of data.
5. The need to establish a contract with the owners of the gyms before implementing the system that will be slightly customized based on their facility.
6. There is no set budget for this project, the developers are free to spend as much as they see fit within reason as long it ends up operational.

## **Technical Constraints**

The new system must conform to the following information technology architectural standards:

1. The current messaging architecture is based on Outlook clients (for e-mail) running on a Microsoft Windows 10-based Exchange Server.
2. This project will require the development of one or more enterprise databases. The database server standard is MySQL Server running on 000webhost's free servers as an initial testing environment in addition for Firebase's Firestore database.
3. The project will require the development of a website, two android mobile applications, and two desktop applications. The application development environments chosen are Visual Studio Code, and Android Studio. The development team has also approved the use of Edraw Max and Lucidchart for the pre-planning of the project.
4. Internet servers will be implemented using 000webhost's free server-hosting service.
5. Internally, the development process will be through personal workstations running windows 10 OS, all including Mozilla Firefox browsers.
6. Externally, for members, any solution developed must run equally well on Microsoft Edge, Google Chrome, Safari, or Mozilla Firefox mobile or desktop browsers.
7. External hard drives will also be used to store backups for the system.
8. Local workstations' storages will be used as backups in addition to Google's drive, that's where the development team will share and keep track of the work.
9. Communication applications such as Discord, Gmail, and WhatsApp will be used for team communication in the times the members of the team cannot meet.

## **Documentation**

The following guidelines should be used for communications:

1. The project team will hold weekly status meetings, chaired by the project supervisor.
2. Team members will utilize electronic mail, dialogue, and written completion criteria on a regular basis as vehicles for project communication.
3. The following directory folders and cloud drives shall be used to store this charter and all subsequent documentation and work-in-progress components:

Google Drive:

[https://drive.google.com/drive/folders/1OoUGOZozUmPnCGF9YLquqGX2EFum2Dc-  
?usp=sharing](https://drive.google.com/drive/folders/1OoUGOZozUmPnCGF9YLquqGX2EFum2Dc-?usp=sharing)

Local Storage:

D:\Senior Project\ ...

# Use Case Analysis

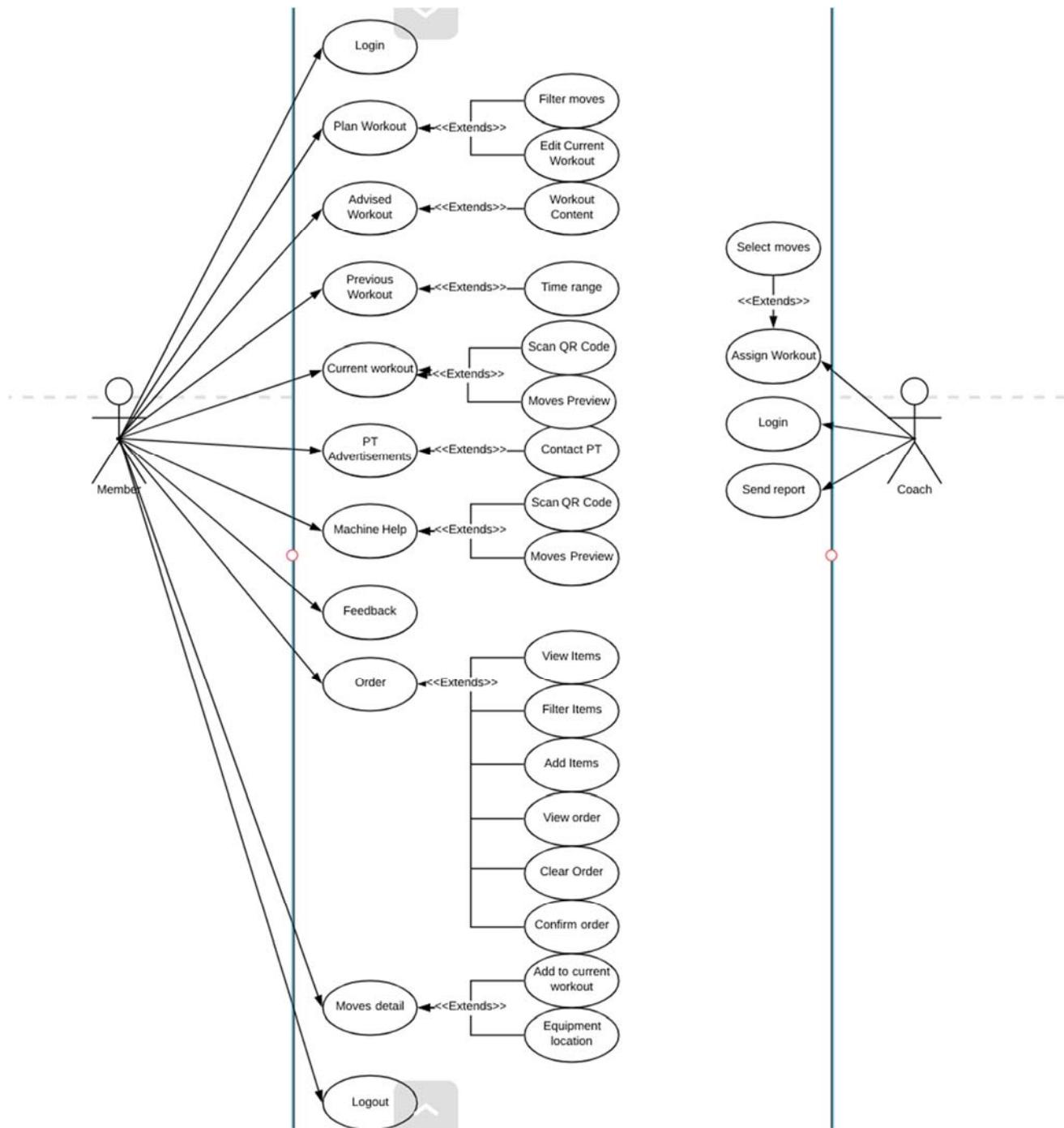
All diagrams were made using Lucidchart.

## Mobile Application Use Case List Table

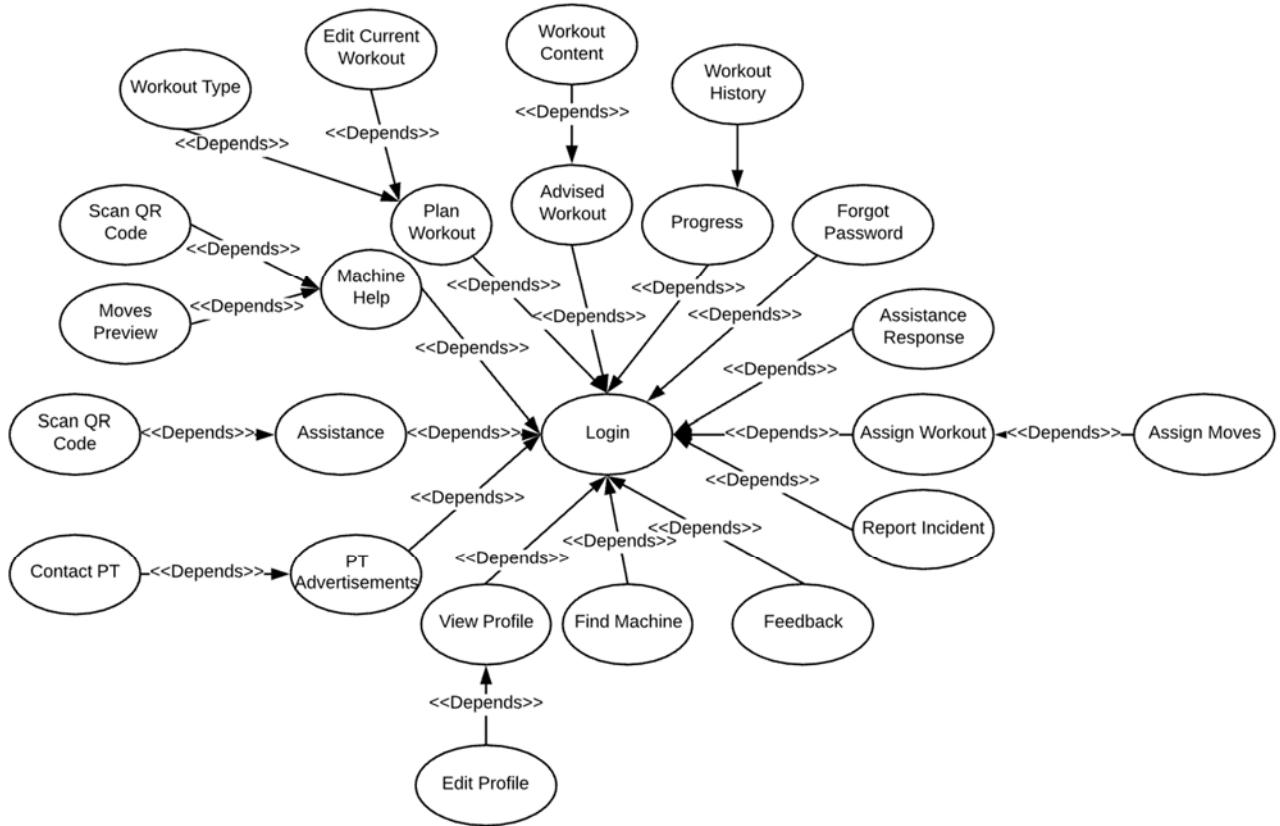
Actors	Case Name	Case Description	Case ID
Member	Login	Access the member system by entering valid credentials	UMM001
	Plan Workout	Create a new personal workout	UMM002
	-Filter Moves	-Filter available moves	-UMM003
	-Edit Current Workout	-Edit your in-progress workout	-UMM004
	Advised Workout	View the workout advised by the coach	UMM005
	-Time Range	-Display workouts in time range	-UMM006
	-Workout Content	-Details of each of the workout's moves	-UMM007
	Current Workout	View workout in progress	UMM008
	-Finish Workout	-Mark current workout as finished	-UMM009
	-Workout Content	-Details of each of the workout's moves	-UMM010
	Previous Workout	Check workouts you've previously completed	UMM011
	-Time Range	-Display workouts in time range	-UMM012
	Moves' Details	Check a move's details	UMM013
	-Add to Current Workout	-Add the move to current workout	-UMM014
	-Equipment Location	-Show move's equipment's location	-UMM015
	Order	Order from the bar	UMM016
	-View Items	-Search for items by name	-UMM017
	-Filter Items	-View the bar's menu	-UMM018
	-Add Items	-Add items to order	-UMM019
	-View Order	-View chosen order	-UMM020
	-Clear Order	-Reset order	-UMM021
	-Confirm Order	-Send order	-UMM022
	PT Advertisements	Personal trainers' listings section	UMM023
	-Contact PT	-Show ways to contact listed trainers	-UMM024
	Machine Help	Show more information about a machine	UMM025
	-Scan QR Code	-Scan the QR code on the machine	-UMM026
	-Moves Preview	-View listed moves	-UMM027
	Feedback	Submit your personal feedback	UMM028
	Logout	Logout	UMM029

Actors	Case Name	Case Description	Case ID
Coach	Login	Access the coach system by entering valid credentials	UMC001
	Assign Workout -Select Moves	Make a workout for a member -Choose included moves	UMC002 -UMC003
	Send Report	Report incidents and machine malfunctions	UMC004
	Logout	Logout	UMC005

## Mobile Use Case Diagram



# Mobile Use Case Dependency Diagram

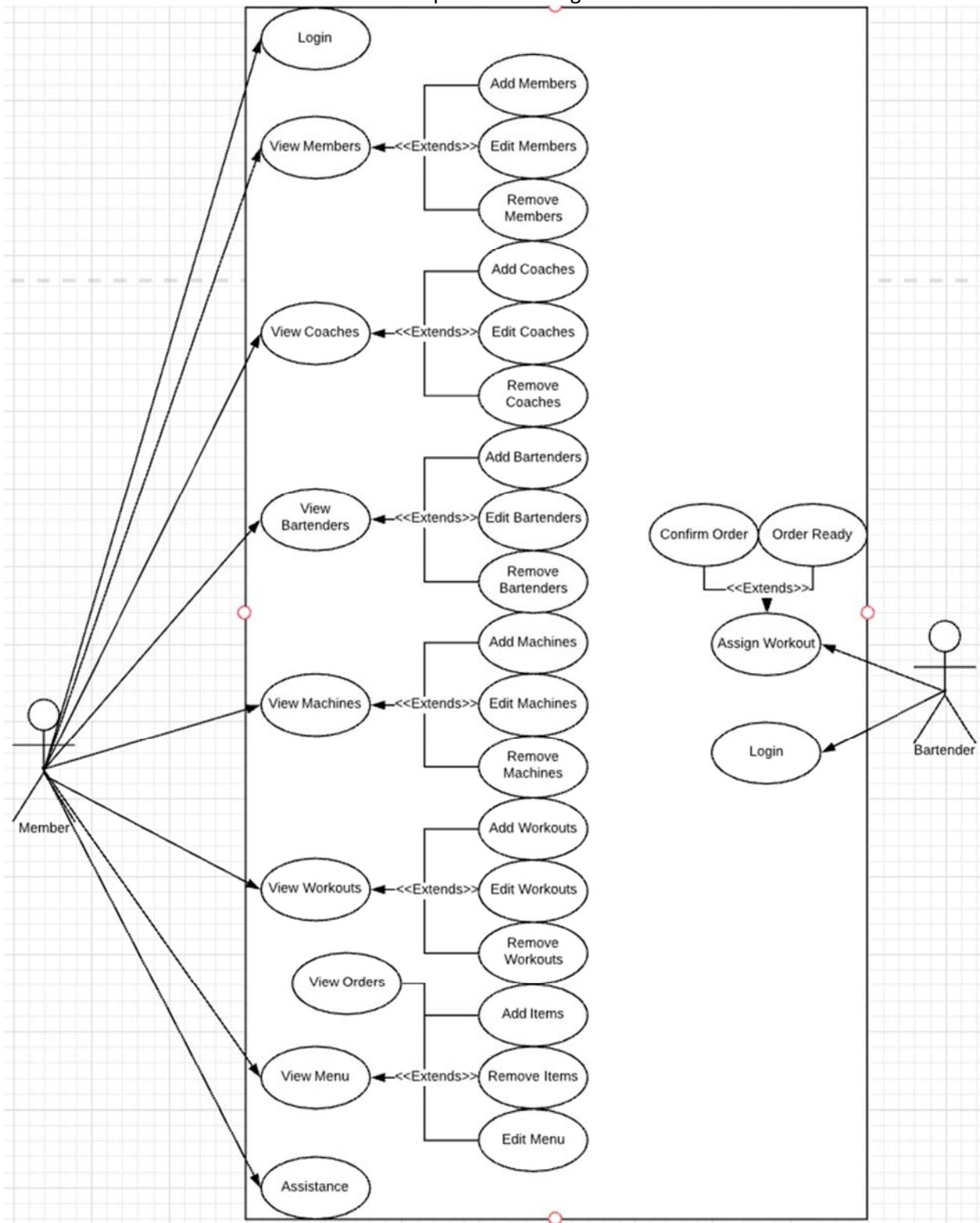


Desktop Application Use Case List Table

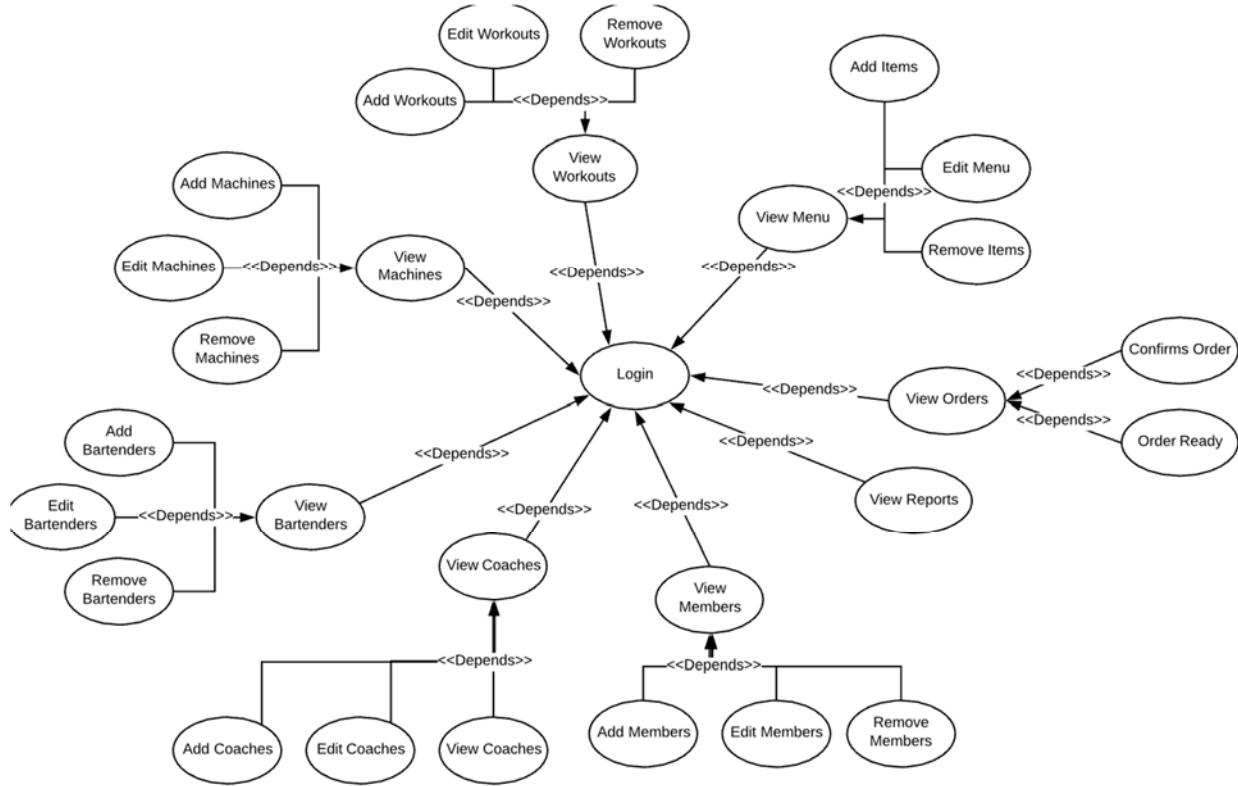
Actors	Case Name	Case Description	Case ID
Admin	Login	Access the admin system by entering valid credentials	UDA001
	View Members -Add Members -Edit Members -Remove Members	View all the members on the system -Add new members to the system -Edit members' data -Remove members from the system	UDA002 -UDA003 -UDA004 -UDA005
	View Coaches -Add Coaches -Edit Coaches -Remove Coaches	View all the coaches on the system -Add new coaches to the system -Edit coaches' data -Remove coaches from the system	UDA006 -UDA007 -UDA008 -UDA009
	View Bartenders -Add Bartenders -Edit Bartenders -Remove Bartenders	View all the bartenders on the system -Add new bartenders to the system -Edit bartenders' data -Remove bartenders from the system	UDA010 -UDA011 -UDA012 -UDA013
	View Machines -Add Machines -Edit Machines -Remove Machines	View all the machines on the system -Add new machines to the system -Edit machines' data -Remove machines from the system	UDA014 -UDA015 -UDA016 -UDA017
	View Workouts -Add Workouts -Edit Workouts -Remove Workouts	View all the workouts on the system -Add new workouts to the system -Edit workouts' details -Remove workouts from the system	UDA018 -UDA019 -UDA020 -UDA021
	View Moves -Add Moves -Edit Moves -Remove Moves	View all the moves on the system -Add new moves to the system -Edit moves' details -Remove moves from the system	UDA022 -UDA023 -UDA024 -UDA025
	View Menu -Add Items -Remove Items -Edit Menu -View Orders	View the items on the menu -Add new items to the menu -Remove items from the menu -Edit items on the menu -View previous orders	UDA026 -UDA027 -UDA028 -UDA029 -UDA030
	View ads -Add ads -Edit ads -Remove ads	View all the ads on the system -Add new ads to the system -Edit ads' details -Remove ads from the system	UDA031 -UDA032 -UDA033 -UDA034
	View Reports	View members' feedback and coaches' reports	-UDA035

Actors	Case Name	Case Description	Case ID
Bartender	Login	Access the Bartender system by entering valid credentials	UDB001
	View Orders -Confirm Order -Order Ready -Order Served -Order Unserved -Order Denied	View received orders -Confirm order availability -Notify member that the order is ready -Mark Order as served -Mark order as unserved -Reject Order	UDB002 -UDB003 -UDB004 -UDB005 -UDB006 -UDB007
	View Menu -Add Items -Remove Items -Edit Menu	View the items on the menu -Add new items to the menu -Remove items from the menu -Edit items on the menu	UDB008 -UDB009 -UDB010 -UDB011

Desktop Use Case Diagram



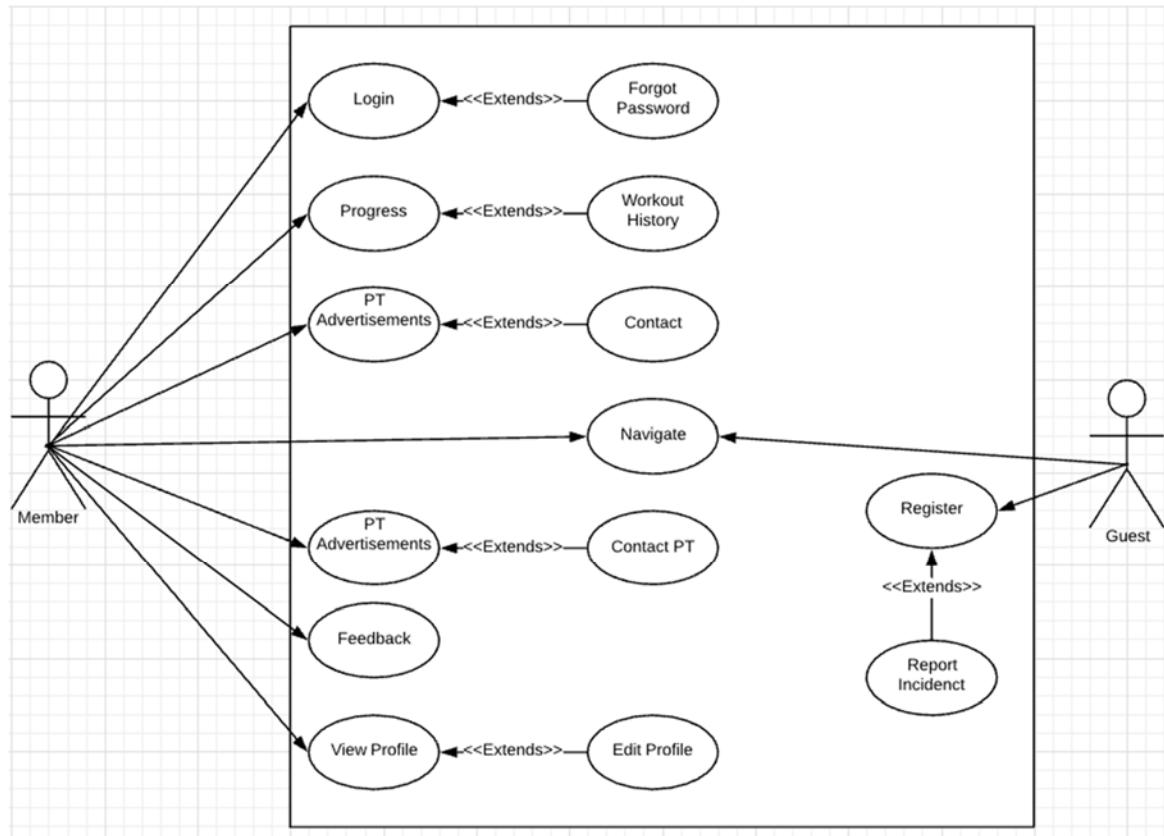
## Desktop Use Case Dependency Diagram



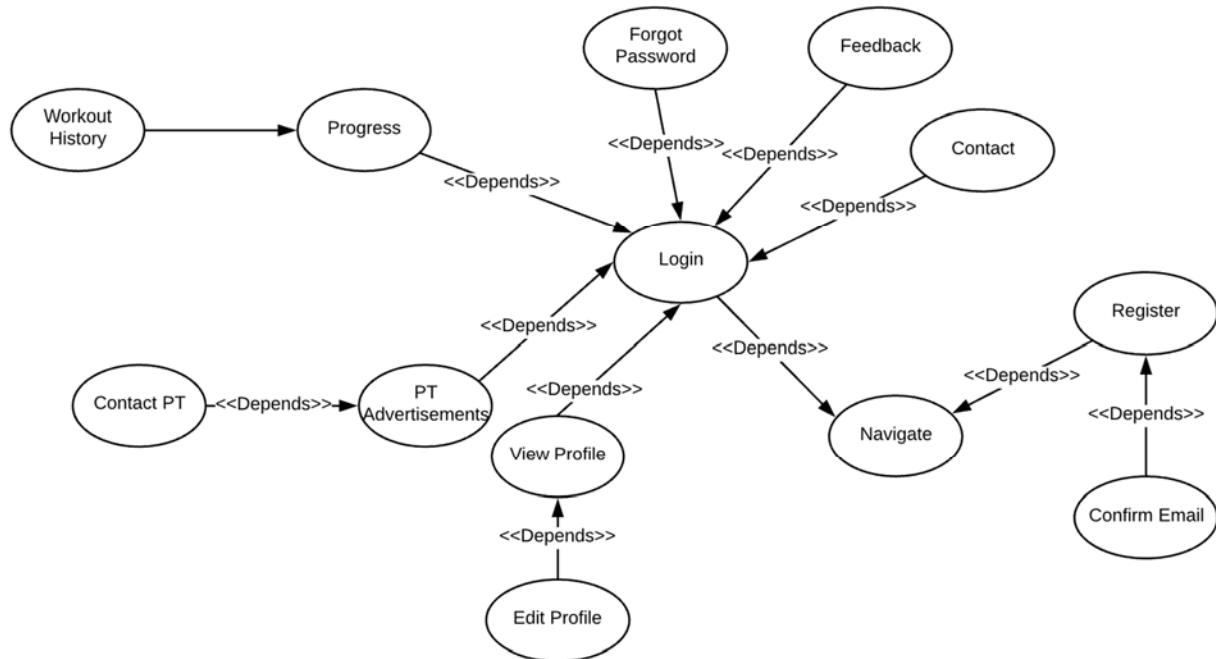
## Web Application Use Case List Table

Actors	Case Name	Case Description	Case ID
Guest	Navigate	Navigate the different sections of the website	UWG001
	Register -Confirm Email	Apply to become a member in the gym -Confirm that the email entered is real	UWG002 -UWG003
	Contact	Send a message to the admin	UWG004
Member	Navigate	Navigate the different sections of the website	UWM001
	Login -Forgot Password	Access the member system by entering valid credentials -Request a change of password	UWM002 -UWM003
	Workout History	View all of your previous workouts	UWM004
	Feedback	Submit your personal feedback	UWM005
	View Profile -Edit Profile	View your profile -Edit certain details in your profile	UWM006 -UWM007
	Logout	Logout	UWM008

## Web Use Case Diagram



## Web Use Case Dependency Diagram



# Mobile Application Narratives, Forms, and Sequence Diagrams

## Use Case: Login

Use-Case Name:	Login	
Use-Case ID:	UMM001 / UMC001	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"><li>● Member</li><li>● Coach</li></ul>	
Description:	<p>The use case describes the event of the actor entering his email and password, the system authenticates the data entered, then the actor gains access to his/her respective account. Based on the entered email, the system will grant access to the member or the coach.</p>	
Precondition:	<p>The proper account must exist in the database beforehand. In the member's case, he/she must register and be confirmed by the admin.</p>	
Trigger:	<p>This use case is triggered upon clicking the "Login" url.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"><li>● Step 1: The actor enters his account email.</li><li>● Step 2: The actor enters his password.</li></ul>	<ul style="list-style-type: none"><li>● Step 3: The system verifies that an account with this email exists.</li><li>● Step 4: The system checks whether the email belongs to an admin or a member</li><li>● Step 5: The system verifies that the password matches the account linked to the email.</li><li>● Step 6: The system logs the actor into his account.</li></ul>
Alternate Courses:	<ul style="list-style-type: none"><li>● Alt-Step 3: The actor entered an email with no account linked to it; the system informs the actor and asks him/her to change the email or to register.</li><li>● Alt-Step 4: The actor entered the wrong password; the system informs the actor and asks him/her to re-enter the password.</li></ul>	
Conclusion:	<p>This use case concludes when the system is successfully logged into the system and redirected to the home page.</p>	
Post condition:	<p>The actor gains access to his account and the session is created.</p>	





Email

Password

**LOGIN**



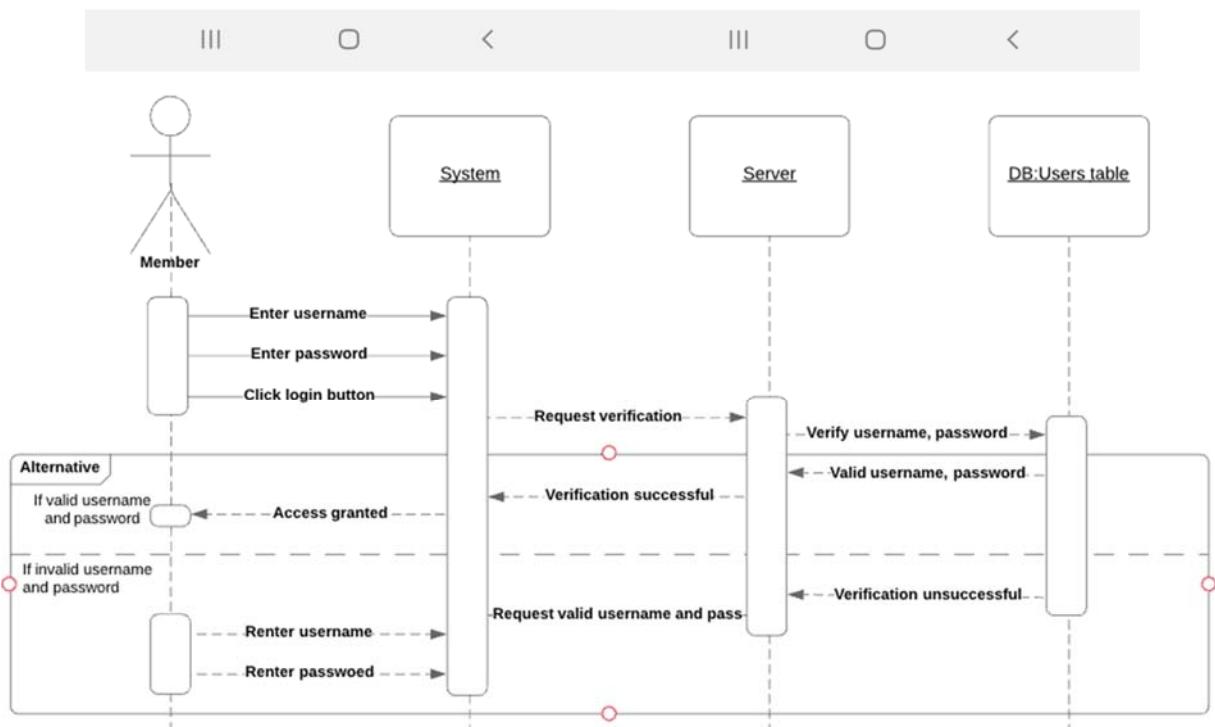
Email

Password

**LOGIN**

Your account is not activated.

Users only need to login once, then the device remembers their state through Firebase authentication, unless of course, they logout manually.



## use Case: Plan Workout

Use-Case Name:	Plan Workout	
Use-Case ID:	UMM002	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Member</li> </ul>	
Description:	<p>The use case describes the event of the actor deciding to plan or edit his/her own workout, the member is then taken to a new activity to choose the type of his workout.</p>	
Precondition:	<p>The member must be logged.</p>	
Trigger:	<p>This use case is triggered upon clicking the “Workouts” option from the navigation drawer.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: The actor chooses to plan a new workout.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system opens an activity with the available workout options.</li> </ul>
Alternate Courses:	<p>There are no alternate courses for this use case.</p>	
Conclusion:	<p>This use case concludes when the activity with the available options opens.</p>	
Post condition:	<p>The workout is now able to move to the next step of making a workout.</p>	

Search by Name, Equipment, or Muscle

Ball

Barbell

Ball Leg Curl

Ab Crunch Machine

Band Hip Adductions

Barbell Full Squat

Exercise Ball Pull-In

Barbell Glute Bridge

Clean and Press

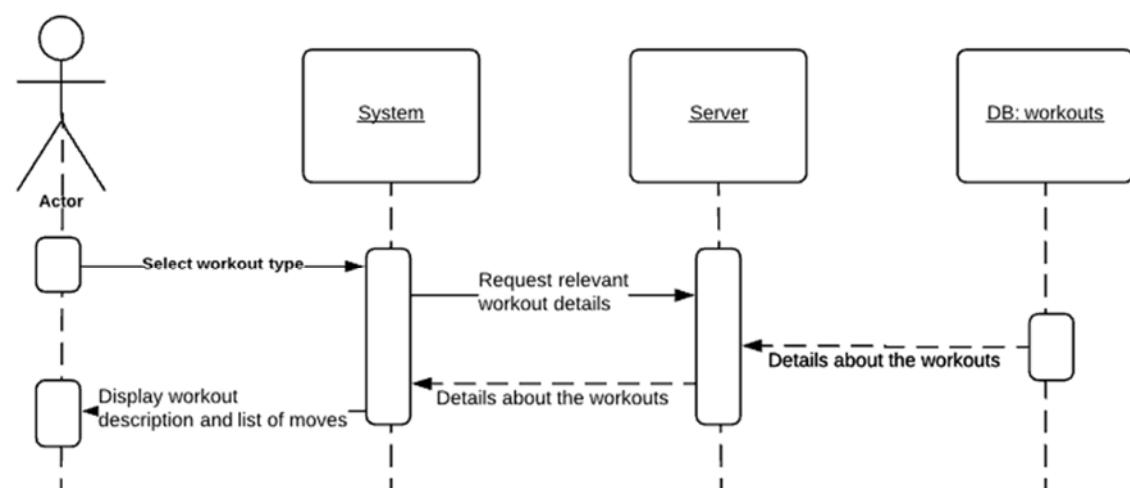
Clean Deadlift

Clean from Blocks

Current Assigned Previous Customize

Current Assigned Previous Customize

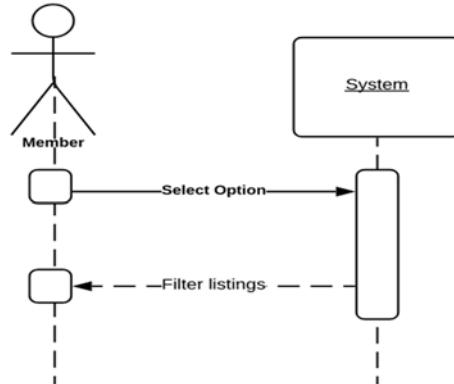
Current Assigned Previous Customize



## Use Case: Filter Moves

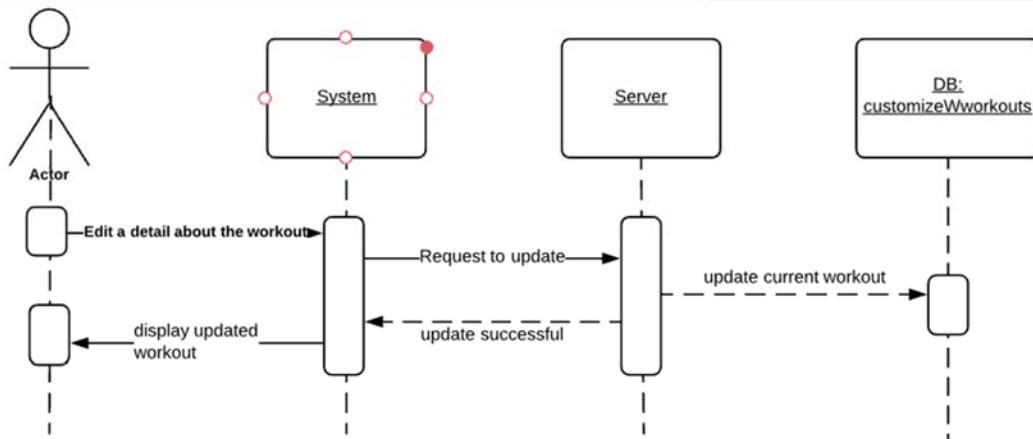
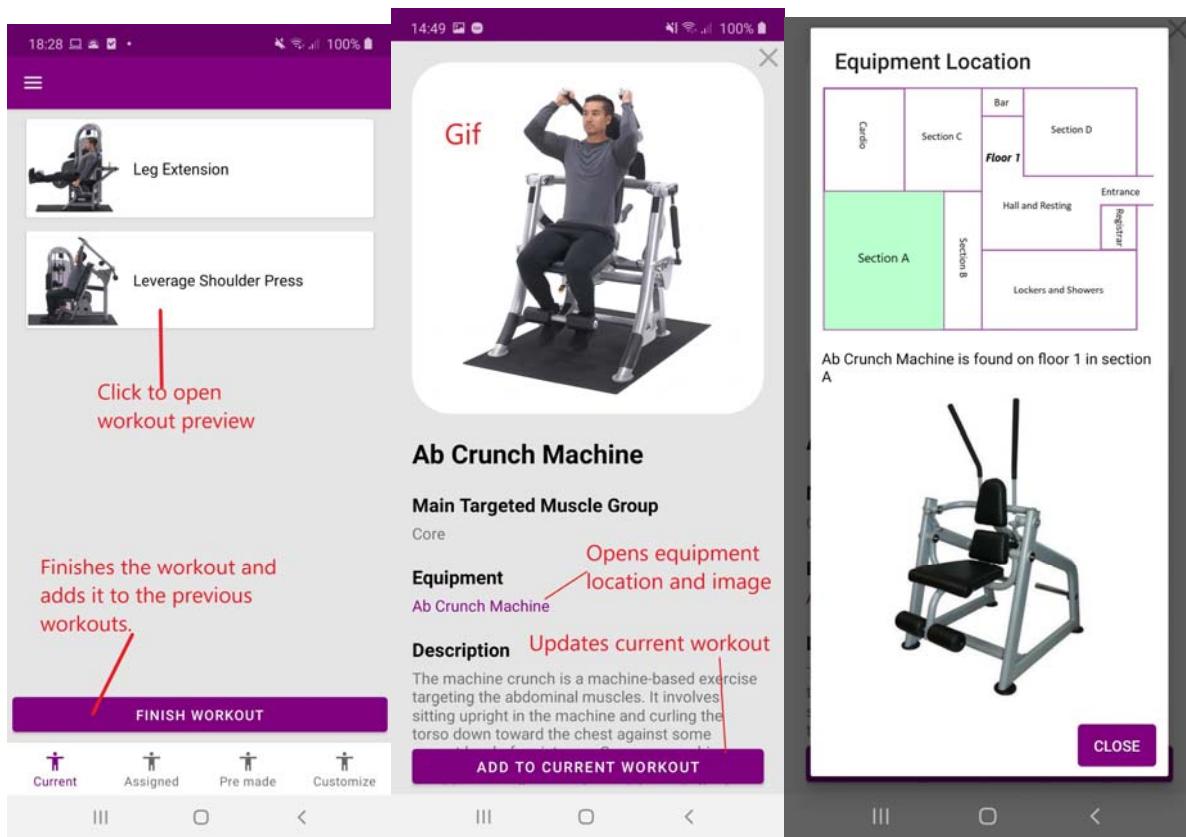
Use-Case Name:	Filter Moves	
Use-Case ID:	UMM003	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Member</li> </ul>	
Description:	<p>The use case describes the event of an actor typing inside the filter bar, the system will filter out all the options that do not match the typed text.</p>	
Precondition:	<p>The member must be logged in and viewing the list of moves.</p>	
Trigger:	<p>This use case is triggered upon typing inside the filter bar.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: The actor types inside the bar.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system filters out any irrelevant moves.</li> </ul>
Alternate Courses:	<p>There are no Alt-Steps for this use case.</p>	
Conclusion:	<p>This use case concludes when the actor clears out the filter bar.</p>	
Post condition:	<p>The actor can now learn how to perform specific moves and complete a workout correctly.</p>	

Preview Image included in use case UMM002



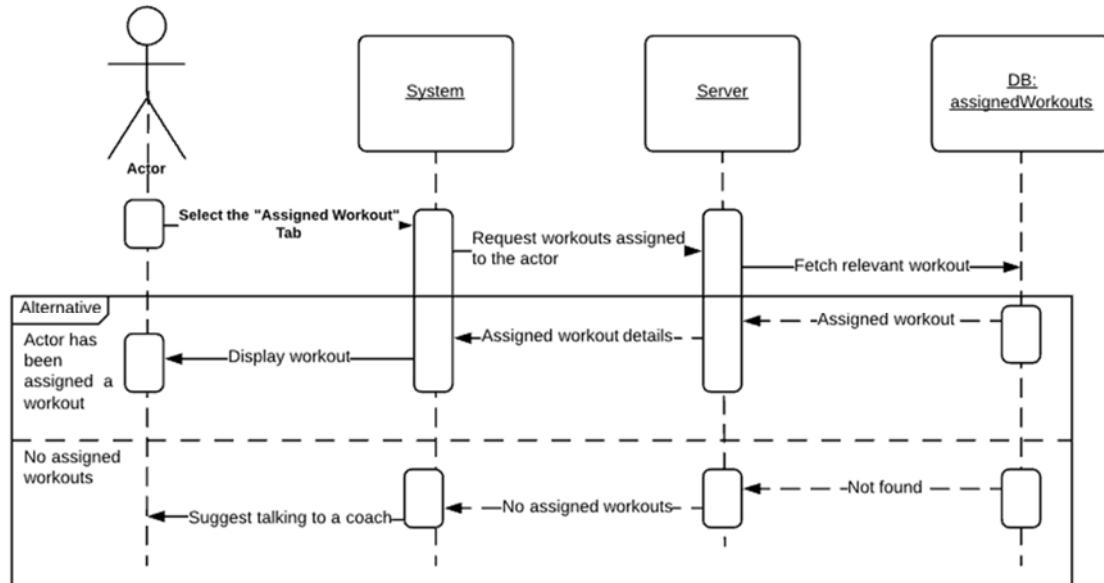
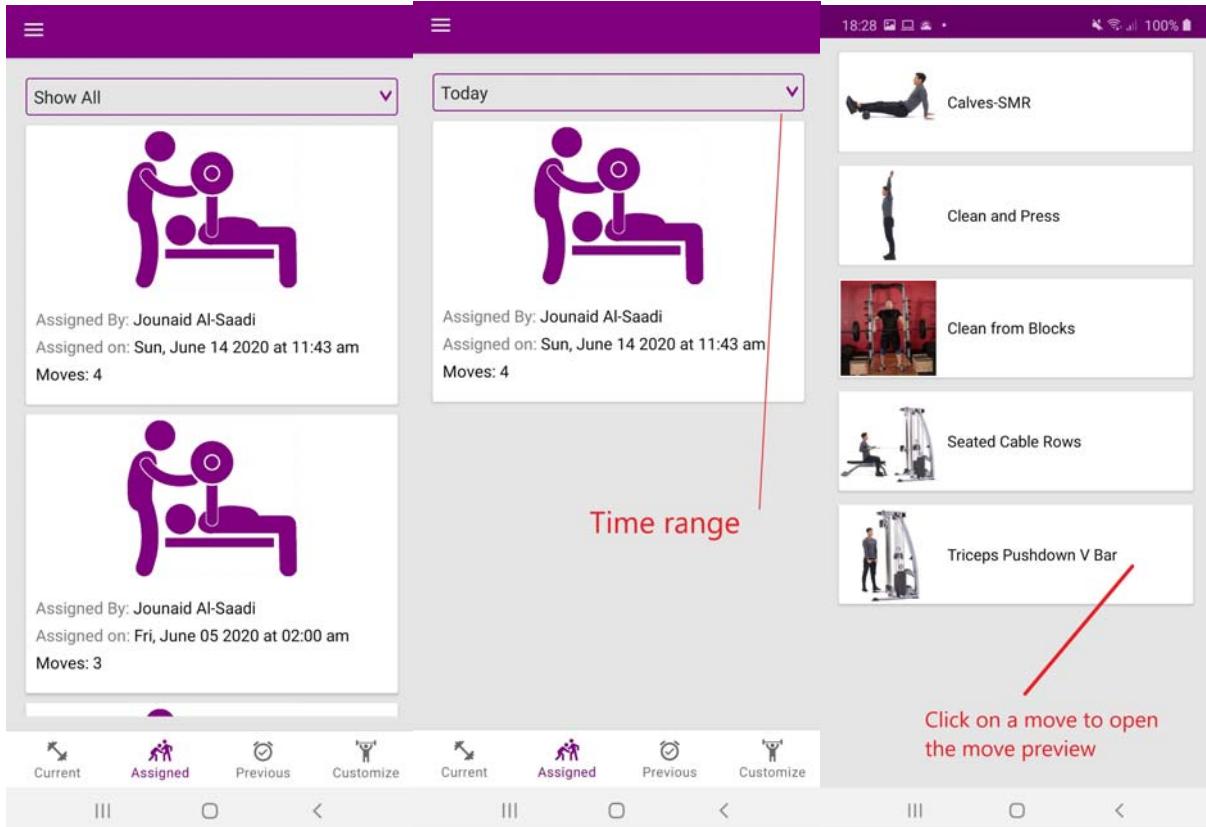
### Use Case: Edit Current Workout

Use-Case Name:	Edit Current Workout	
Use-Case ID:	UMM004	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Member</li> </ul>	
Description:	<p>The use case describes the event of an actor deciding to edit the workout he is currently following.</p>	
Precondition:	<p>The member must be logged in and have a current workout selected.</p>	
Trigger:	<p>This use case is triggered upon viewing the “current workout” tab.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: The actor chooses to view the current workout.</li> <li>● Step 3: The actor adds/removes parts of the workout.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system opens an activity showing the current workout.</li> <li>● Step 4: The current workout is updated on the system.</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● Alt-Step 2: The actor doesn't have a current workout active; the system displays text indicating that.</li> </ul>	
Conclusion:	<p>This use case concludes when the actor leaves the current workout activity.</p>	
Post condition:	<p>The actor has updated his current workout to his likings.</p>	



## Use Case: Advised Workout

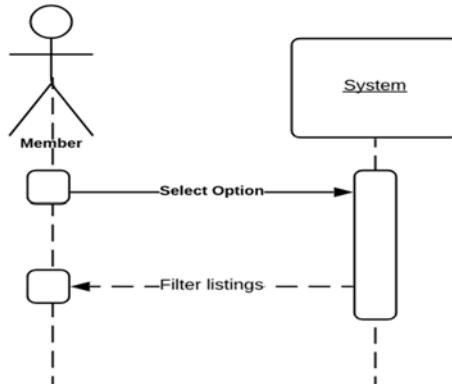
Use-Case Name:	Advised Workout	
Use-Case ID:	UMM005	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Member</li> </ul>	
Description:	<p>The use case describes the event of an actor viewing the workouts advised by a coach, the actor can then view the full details of the workout.</p>	
Precondition:	<p>The member must be logged in and advised by a coach.</p>	
Trigger:	<p>This use case is triggered upon clicking the “Advised Workouts” option from the navigation drawer.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: The actor chooses to view the advised workouts.</li> <li>● Step 3: The actor clicks on the workout.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system opens an activity showing the advised workout.</li> <li>● Step 4: The system creates a new activity showing the details of the workout.</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● Alt-Step 2: The actor hasn't been advised yet; the system will show a text indicating that.</li> </ul>	
Conclusion:	<p>This use case concludes when the actor leaves the activities related to the advised workout.</p>	
Post condition:	<p>The actor can now follow the workouts advised by the coaches.</p>	



## Use Case: Time Range

Use-Case Name:	Time Range	
Use-Case ID:	UMM006	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>• Member</li> </ul>	
Description:	<p>The use case describes the event of an actor selecting a specific time range, the system will compare the current date with the date the workout was assigned and only show the workouts in the selected time range.</p>	
Precondition:	<p>The member must be logged in and viewing the list of assigned workouts.</p>	
Trigger:	<p>This use case is triggered upon selecting a time period.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>• Step 1: The actor selects a time option.</li> </ul>	<ul style="list-style-type: none"> <li>• Step 2: The system filters out any workout not in the selected range.</li> </ul>
Alternate Courses:	<p>There are no Alt-Steps for this use case.</p>	
Conclusion:	<p>This use case concludes when the actor leaves the activities related to the advised workout.</p>	
Post condition:	<p>The actor can now learn how to perform specific moves and complete a workout correctly.</p>	

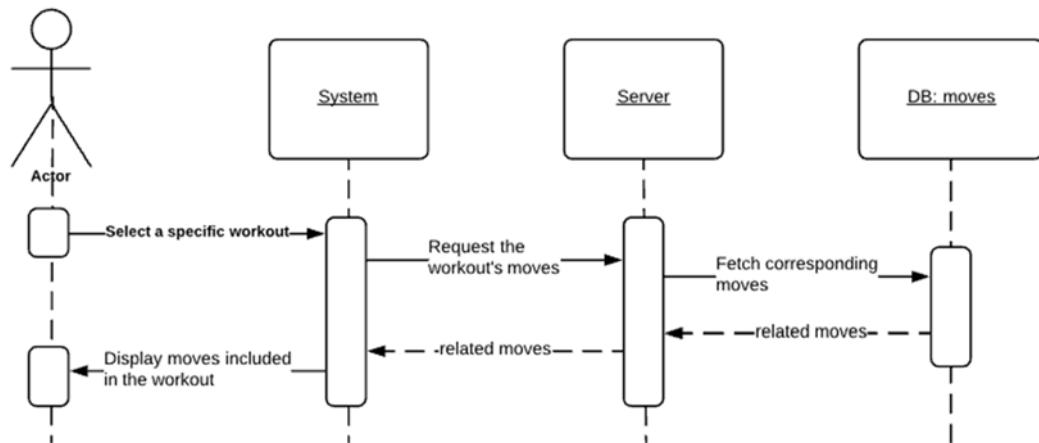
Preview Image included in use case UMM005



## Use Case: Workout Content

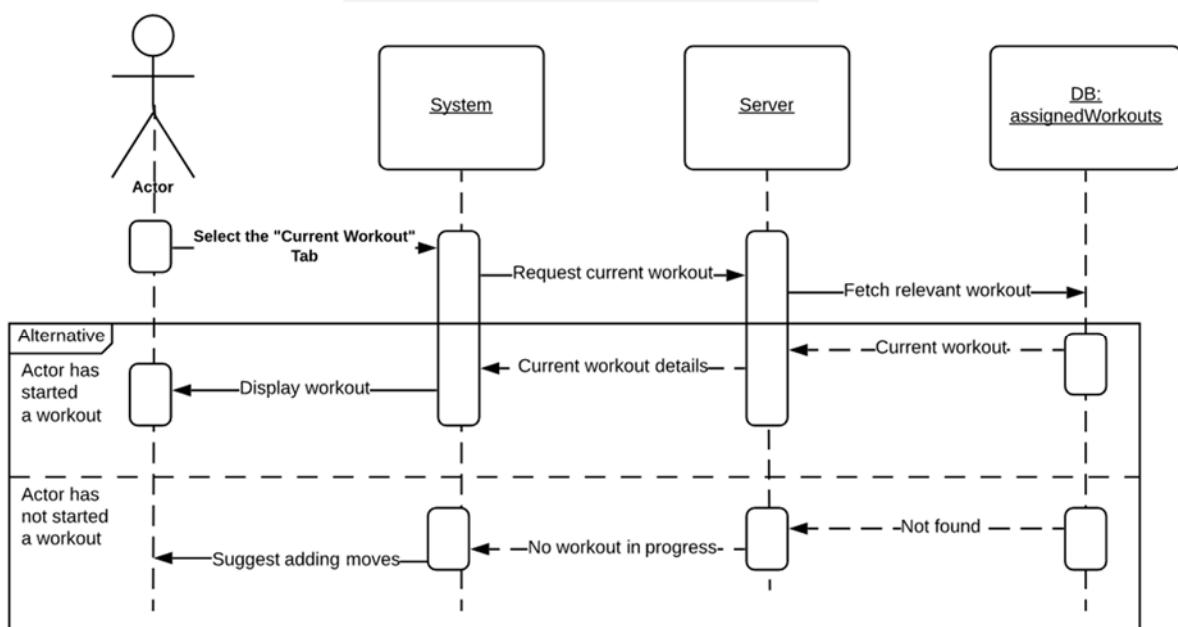
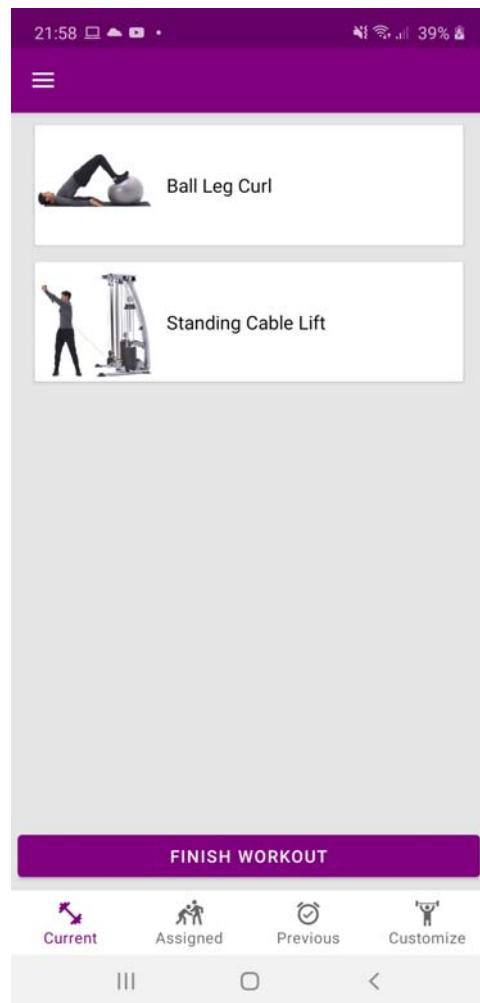
Use-Case Name:	Workout Content	
Use-Case ID:	UMM007	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Member</li> </ul>	
Description:	<p>The use case describes the event of an actor viewing a workout's details and moves, the system will show a list of all possible moves and describe them, it will also provide some visual explanations.</p>	
Precondition:	<p>The member must be logged in and viewing the list of workouts.</p>	
Trigger:	<p>This use case is triggered upon clicking on a listed workout.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: The actor chooses to view a specific workout.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system opens an activity showing the list of moves included in that workout with description and explanation.</li> </ul>
Alternate Courses:	<p>There are no Alt-Steps for this use case.</p>	
Conclusion:	<p>This use case concludes when the actor leaves the activity related to the workout he was viewing.</p>	
Post condition:	<p>The actor can now learn how to perform specific moves and complete a workout correctly.</p>	

## Preview Image included in use case UMM005



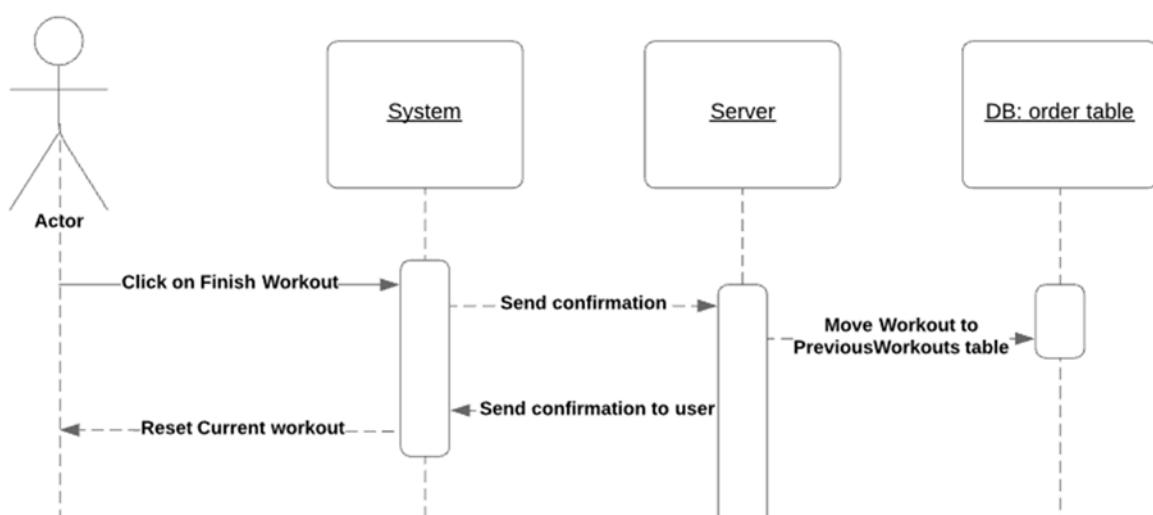
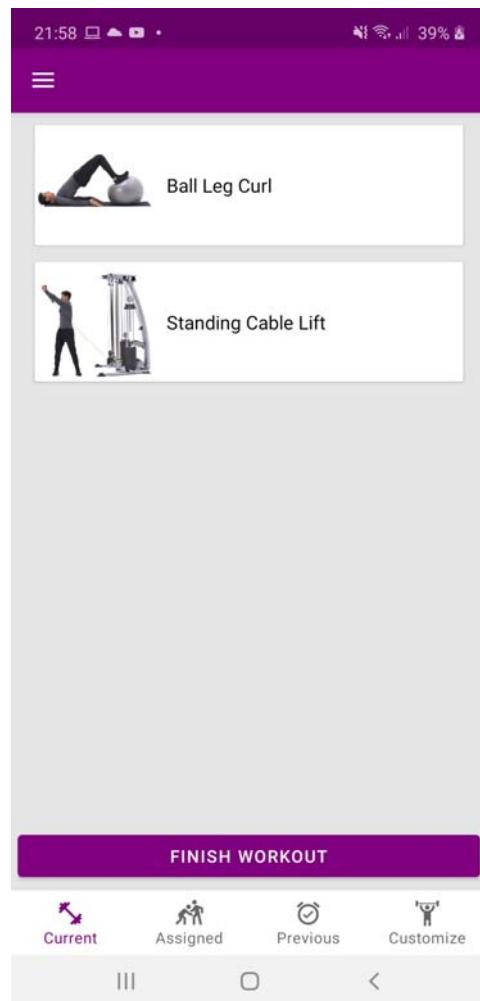
## Use Case: Current Workout

Use-Case Name:	Current Workout	
Use-Case ID:	UMM008	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Member</li> </ul>	
Description:	<p>The use case describes the event of an actor viewing the workout he/she currently has in progress, the actor can then view the full details of the workout.</p>	
Precondition:	<p>The member must be logged in and have added moves to his/her workout.</p>	
Trigger:	<p>This use case is triggered upon clicking the “Current” option from the bottom menu.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: The actor chooses to view the current workout.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system opens an activity showing the current workout.</li> <li>● Step 3: The system shows the details of the workout.</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● Alt-Step 2: The actor hasn't added a move to current workout yet; the system will show a text indicating that.</li> </ul>	
Conclusion:	<p>This use case concludes when the actor leaves the activities related to the current workout.</p>	
Post condition:	<p>The actor can now view his/her current workout.</p>	



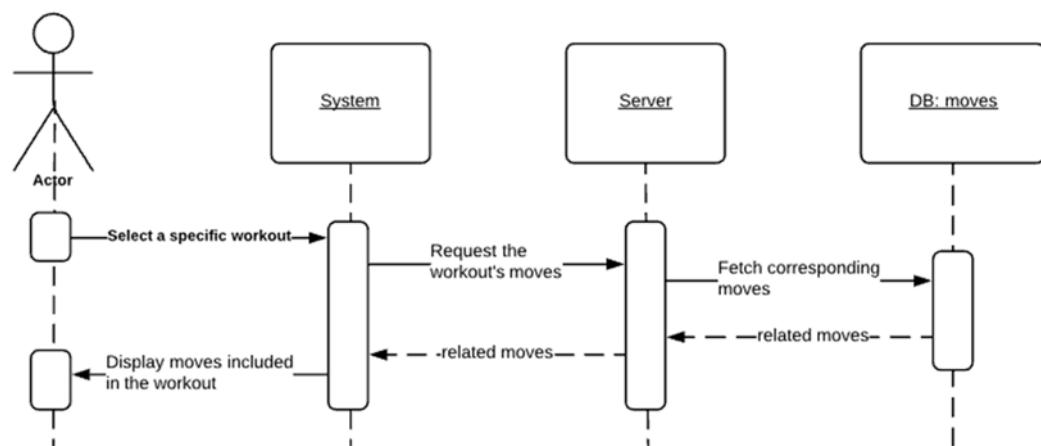
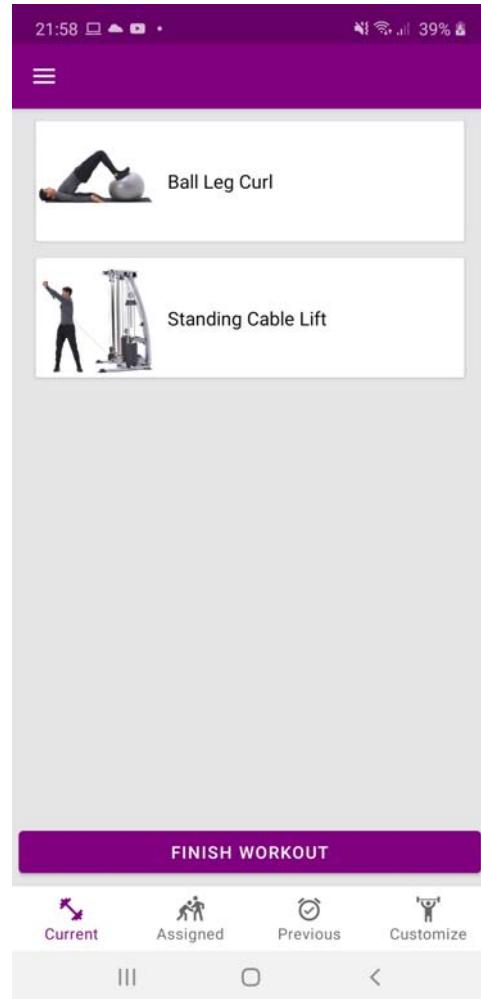
## Use Case: Finish Workout

Use-Case Name:	Finish Workout	
Use-Case ID:	UMM009	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Member</li> </ul>	
Description:	<p>The use case describes the event of an actor deciding to finish the current workout, the system will reset the current workout and save the current workout in the previous workouts activity.</p>	
Precondition:	<p>The member must be logged in and viewing the current workout.</p>	
Trigger:	<p>This use case is triggered upon clicking the “Finish Workout” button.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: The actor clicks “Finish Workout”.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system resets the current workout and updates the database.</li> </ul>
Alternate Courses:	<p>There are no Alt-Steps for this use case.</p>	
Conclusion:	<p>This use case concludes when the current workout is saved in the “Previous Workouts” tab.</p>	
Post condition:	<p>The actor can now start a new workout.</p>	



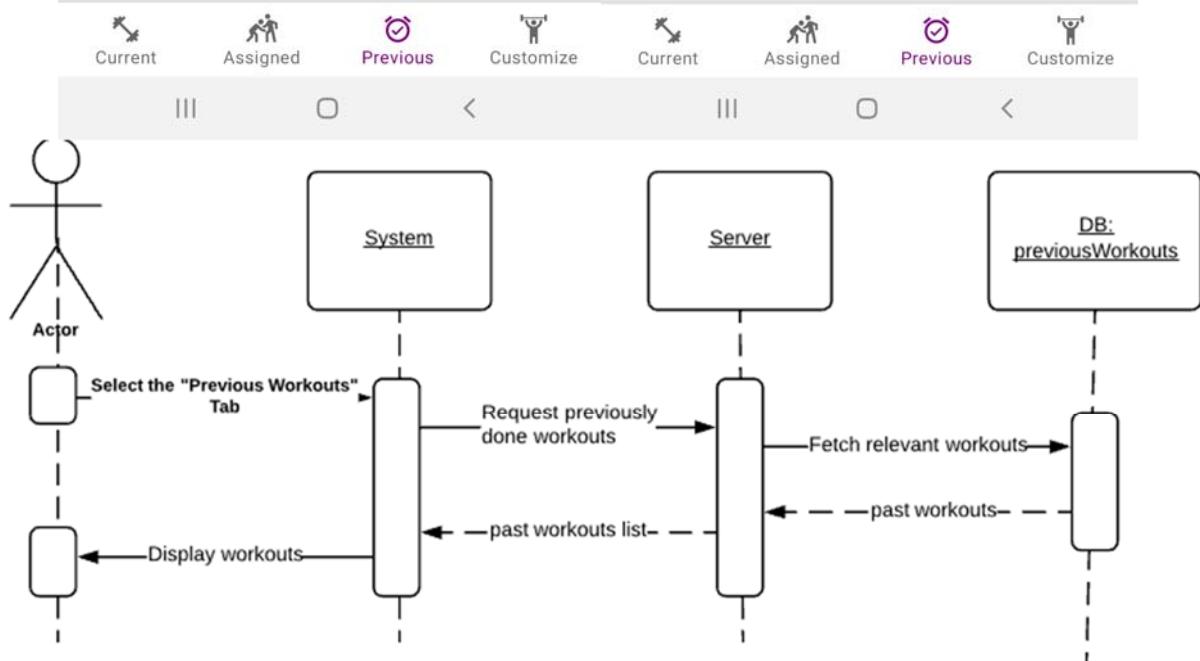
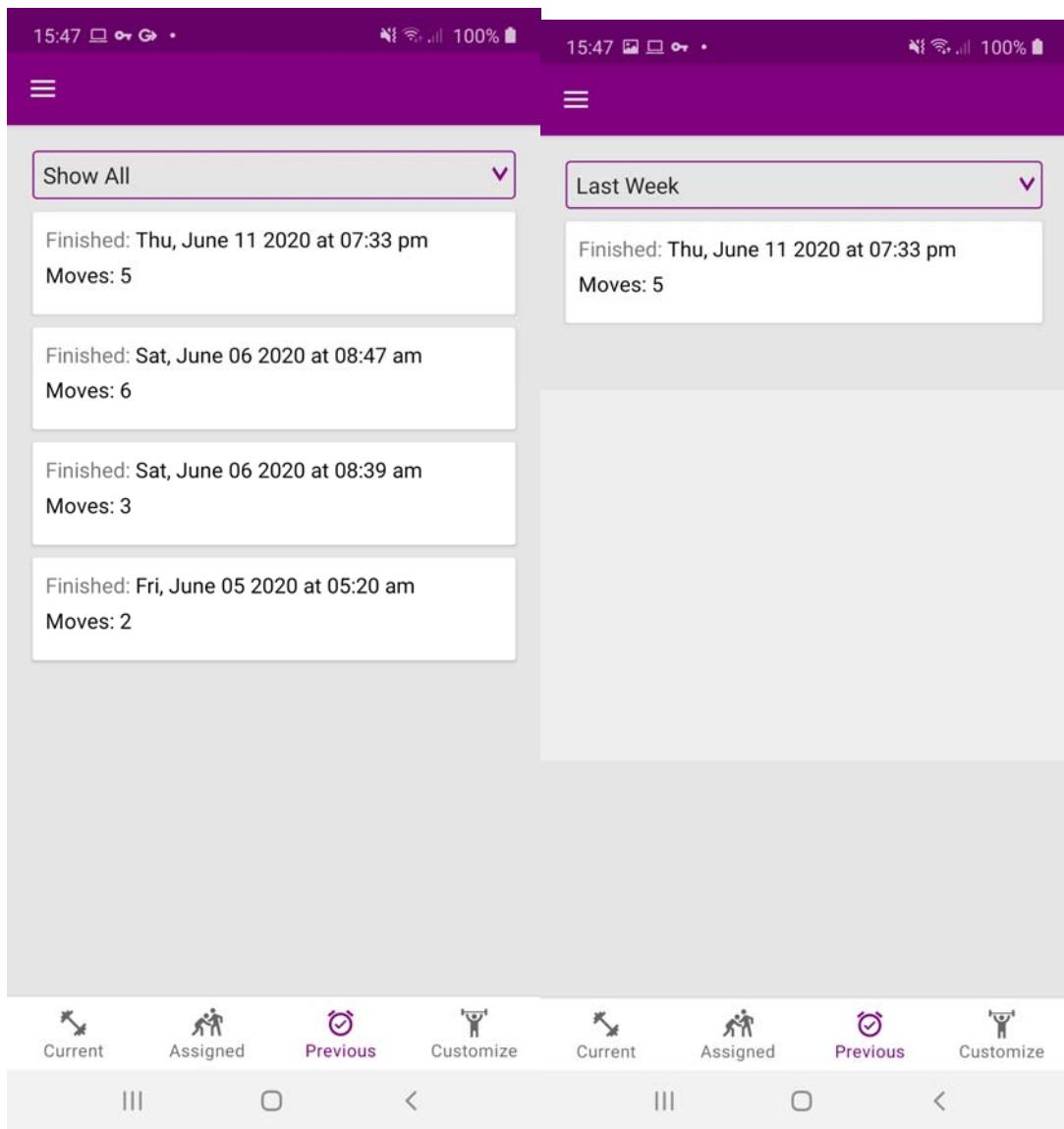
## Use Case: Workout Content

Use-Case Name:	Workout Content	
Use-Case ID:	UMM010	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Member</li> </ul>	
Description:	<p>The use case describes the event of an actor viewing a workout's details and moves, the system will show a list of all possible moves and describe them, it will also provide some visual explanations.</p>	
Precondition:	<p>The member must be logged in and viewing the list of workouts.</p>	
Trigger:	<p>This use case is triggered upon clicking on a listed workout.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: The actor chooses to view a specific workout.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system opens an activity showing the list of moves included in that workout with description and explanation.</li> </ul>
Alternate Courses:	<p>There are no Alt-Steps for this use case.</p>	
Conclusion:	<p>This use case concludes when the actor leaves the activity related to the workout he was viewing.</p>	
Post condition:	<p>The actor can now learn how to perform specific moves and complete a workout correctly.</p>	



## Use Case: Previous Workouts

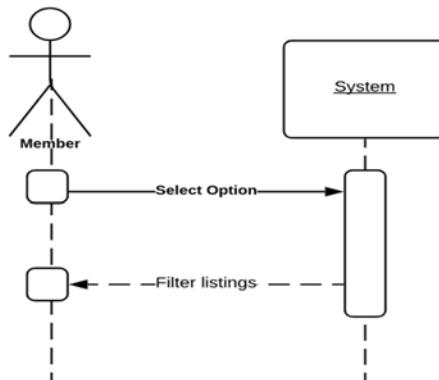
Use-Case Name:	Previous Workouts	
Use-Case ID:	UMM011	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Member</li> </ul>	
Description:	<p>The use case describes the event of an actor viewing all the workouts he has performed in previous sessions.</p>	
Precondition:	<p>The member must be logged in and navigating the “Progress” tab.</p>	
Trigger:	<p>This use case is triggered upon clicking the “Previous Workouts” option from the navigation options in the progress tab.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: The actor chooses to view his workout history.</li> <li>● Step 3: The actor clicks on the workout.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system opens an activity showing all of his workout history.</li> <li>● Step 4: The system creates a new activity showing the details of the workout.</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● Alt-Step 2: The actor hasn't completed a workout yet; the system will show a text indicating that.</li> </ul>	
Conclusion:	<p>This use case concludes when the actor leaves the “Workout History” activity.</p>	
Post condition:	<p>The actor can now go back and check his previous workouts and plan his next ones based on them.</p>	



## Use Case: Time Range

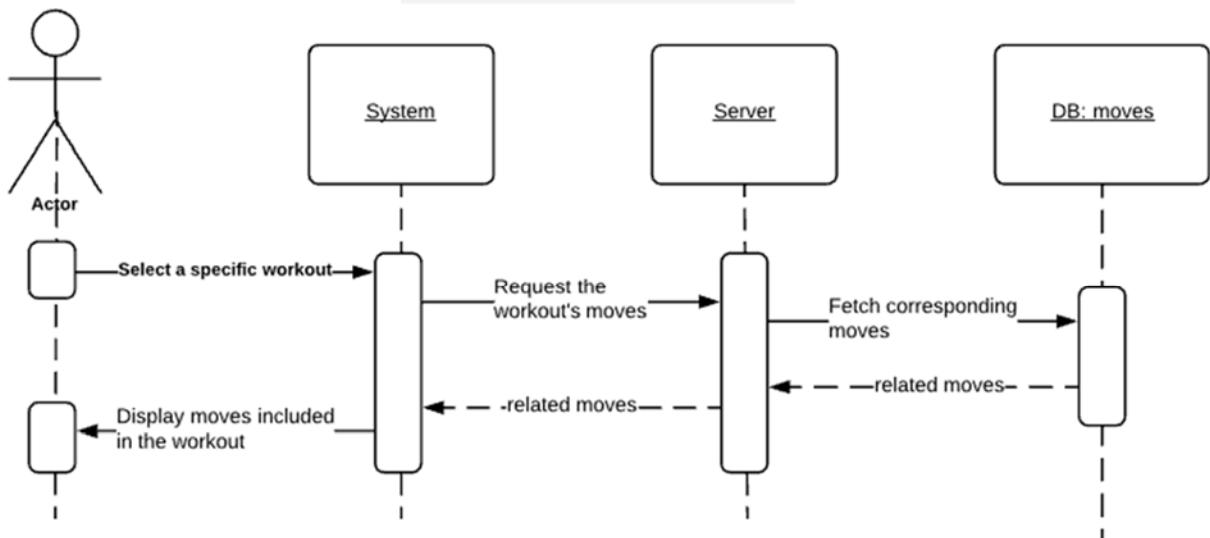
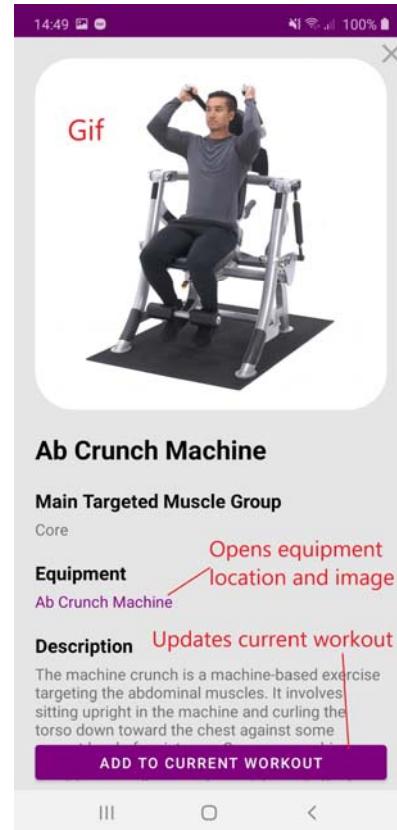
Use-Case Name:	Time Range	
Use-Case ID:	UMM012	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Member</li> </ul>	
Description:	<p>The use case describes the event of an actor selecting a specific time range, the system will compare the current date with the date the workout was assigned and only show the workouts in the selected time range.</p>	
Precondition:	<p>The member must be logged in and viewing the list of assigned workouts.</p>	
Trigger:	<p>This use case is triggered upon selecting a time period.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: The actor selects a time option.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system filters out any workout not in the selected range.</li> </ul>
Alternate Courses:	<p>There are no Alt-Steps for this use case.</p>	
Conclusion:	<p>This use case concludes when the actor leaves the activities related to the advised workout.</p>	
Post condition:	<p>The actor can now learn how to perform specific moves and complete a workout correctly.</p>	

Preview Image included in use case UMM011



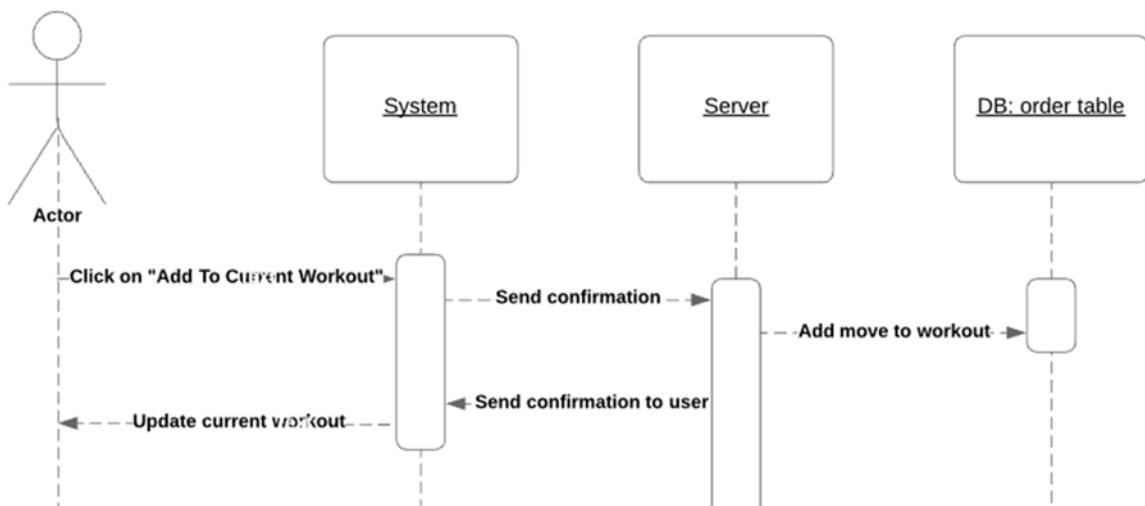
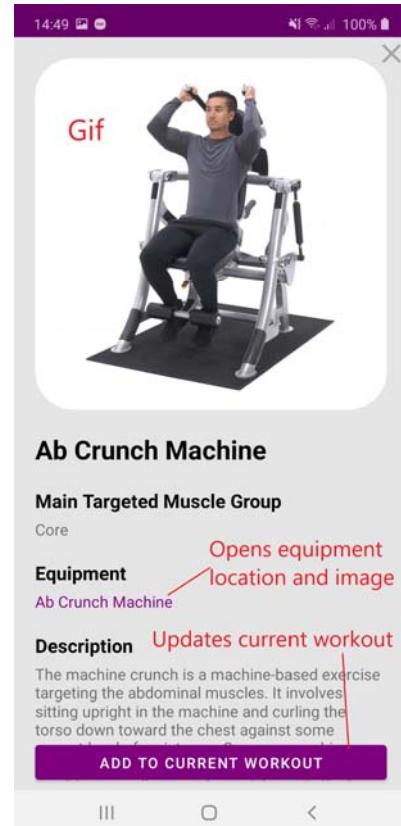
## Use Case: Moves' Details

Use-Case Name:	Moves' Details	
Use-Case ID:	UMM013	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Member</li> </ul>	
Description:	<p>The use case describes the event of an actor clicking on a specific move to view its details, the system will view general info about the move.</p>	
Precondition:	<p>The member must be logged in and viewing a list of moves.</p>	
Trigger:	<p>This use case is triggered upon clicking on one of the shown possible moves.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: The actor chooses to view one of the moves.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system expands the listing to show more information about the move.</li> </ul>
Alternate Courses:	<p>There are no Alt-Steps for this use case.</p>	
Conclusion:	<p>This use case concludes when the actor leaves the activity that lists the possible moves.</p>	
Post condition:	<p>The actor now knows precisely how to perform a move.</p>	



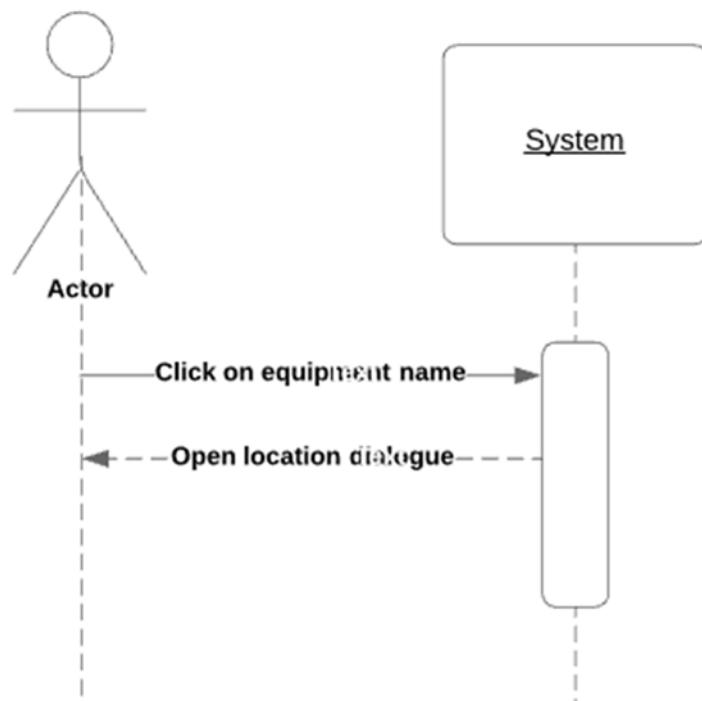
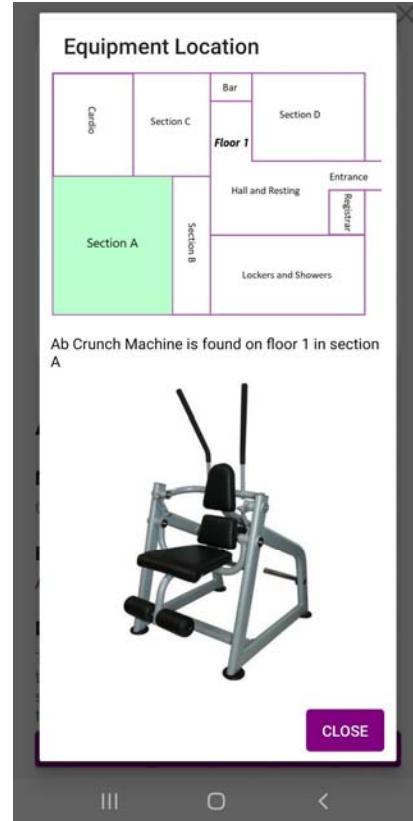
## Use Case: Add to Current Workout

Use-Case Name:	Add to Current Workout	
Use-Case ID:	UMM014	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Member</li> </ul>	
Description:	<p>The use case describes the event of an actor Adding a specific move to the currently in progress workout.</p>	
Precondition:	<p>The member must be logged in and viewing a specific move.</p>	
Trigger:	<p>This use case is triggered upon clicking on the “Add to Current Workout” button.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: The actor clicks “Add To Current Workout”.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system updates the current workout and adds the new move.</li> </ul>
Alternate Courses:	<p>There are no Alt-Steps for this use case.</p>	
Conclusion:	<p>This use case concludes when the move is added to the current workout.</p>	
Post condition:	<p>The actor now sees the move in his/her current workout.</p>	



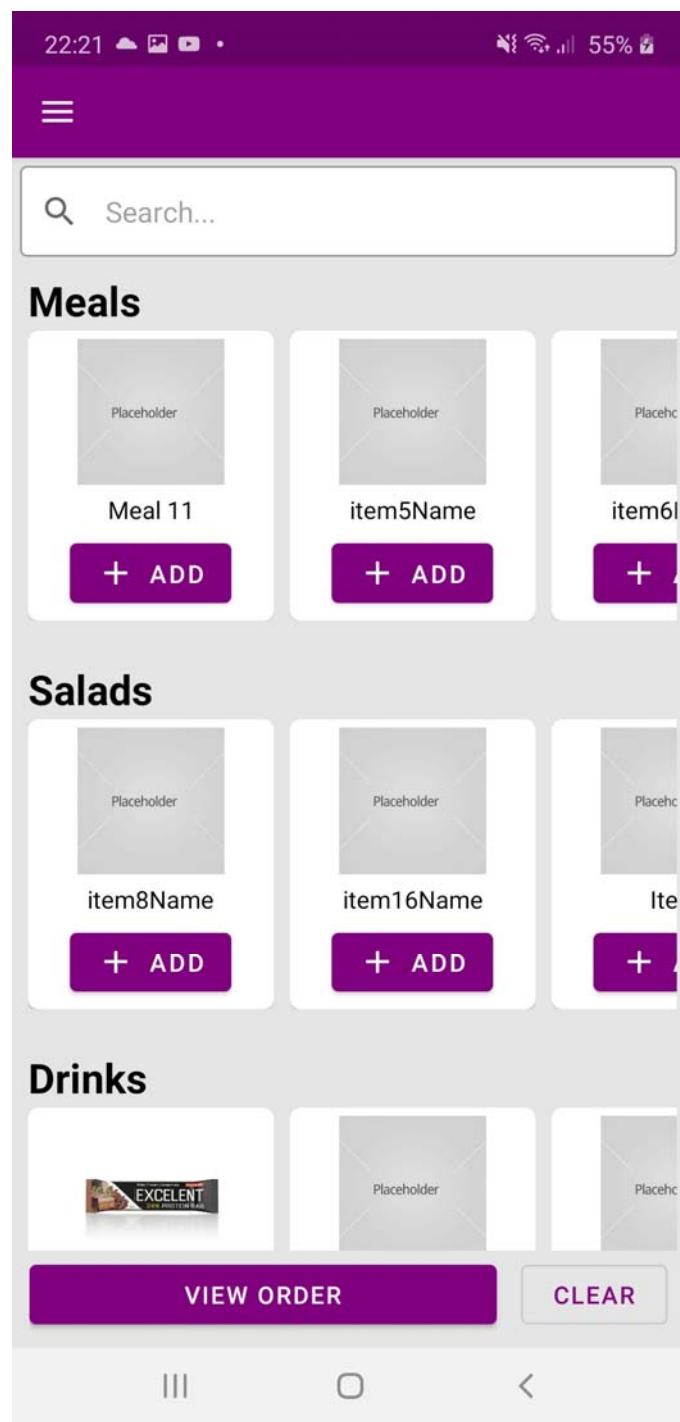
## Use Case: Equipment Location

Use-Case Name:	Equipment Location	
Use-Case ID:	UMM015	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Member</li> </ul>	
Description:	<p>The use case describes the event of an actor viewing the location and equipment used for a specific move.</p>	
Precondition:	<p>The member must be logged in and viewing a move.</p>	
Trigger:	<p>This use case is triggered upon clicking on the name of the equipment required for a specific move.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: The actor clicks on the equipment's name.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system shows a dialogue showing the image of the equipment and the location.</li> </ul>
Alternate Courses:	<p>There are no Alt-Steps for this use case.</p>	
Conclusion:	<p>This use case concludes when the actor closes the location dialogue.</p>	
Post condition:	<p>The actor now knows where he/she can perform the move they were previewing.</p>	



## Use Case: Order

Use-Case Name:	Moves Preview	
Use-Case ID:	UMM016	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Member</li> </ul>	
Description:	<p>The use case describes the event entering the Order tab to start the ordering from the bar process.</p>	
Precondition:	<p>The member must be logged in.</p>	
Trigger:	<p>This use case is triggered upon clicking the “Order” option in the navigation drawer.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: The actor chooses to order.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system opens the order activity.</li> </ul>
Alternate Courses:	<p>There are no Alt-Steps for this use case.</p>	
Conclusion:	<p>This use case concludes when the actor leaves the order activity.</p>	
Post condition:	<p>The actor has initiated the ordering process.</p>	



## Use Case: View Items

Use-Case Name:	View Items	
Use-Case ID:	UMM017	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Member</li> </ul>	
Description:	<p>The use case describes the event of an actor viewing the specific items that are currently available on the menu.</p>	
Precondition:	<p>The member must be logged in and in the order tab.</p>	
Trigger:	<p>This use case is triggered upon clicking on one of the shown items.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: The actor chooses to view one of the items.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system expands the listing to show more information about the item.</li> </ul>
Alternate Courses:	<p>There are no Alt-Steps for this use case.</p>	
Conclusion:	<p>This use case concludes when the actor leaves the activity that lists the available items.</p>	
Post condition:	<p>The actor now knows the nutrition facts and price of the items.</p>	

22:30

61%



**item3Name** 5.4\$

Preparation: 9 minutes

**Nutritional Facts**

Protein: 12g  
Sugar: 4g  
Fats: 8g  
Carbohydrates: 22g  
Calories: 174kcal

**Description**

item3Description

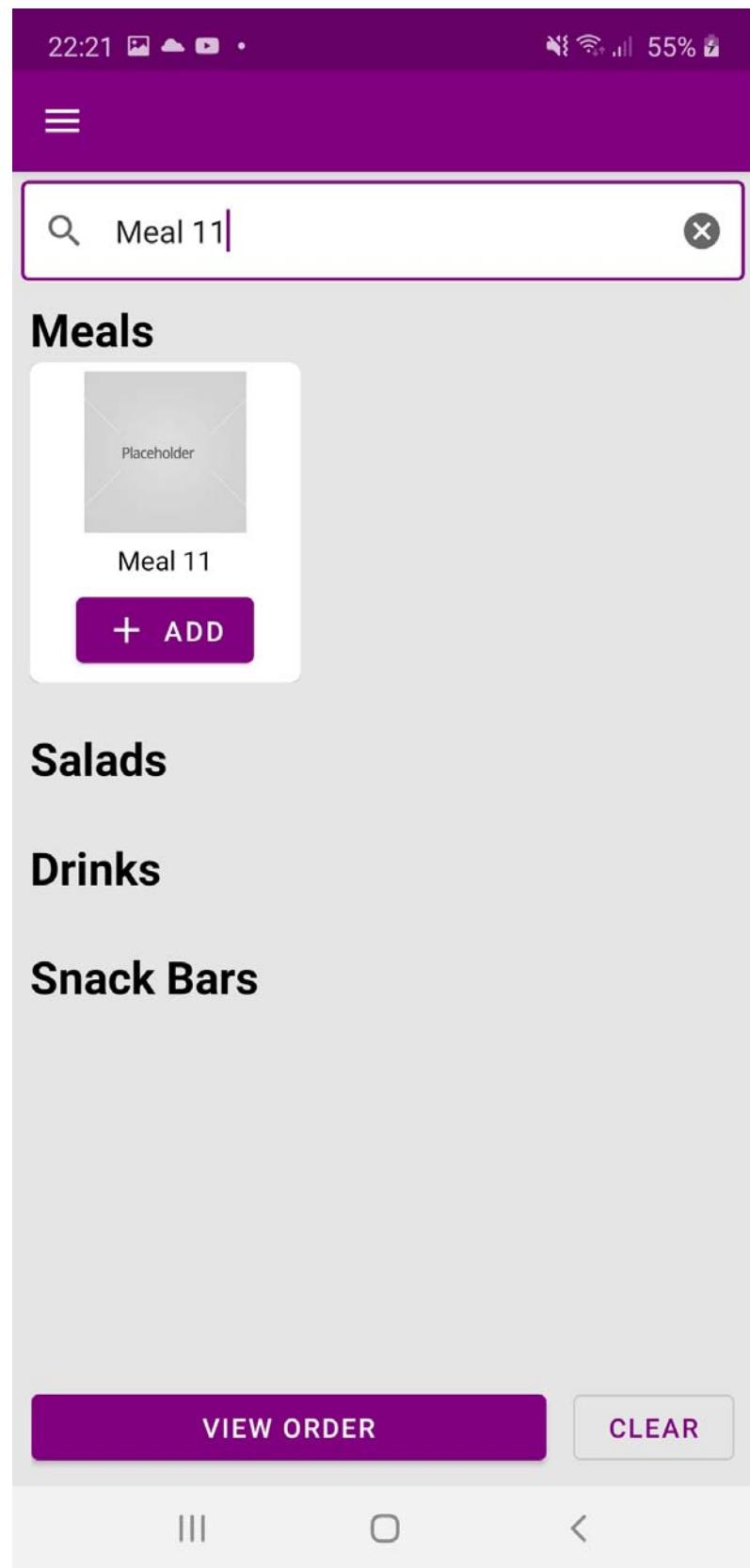
|||

□

<

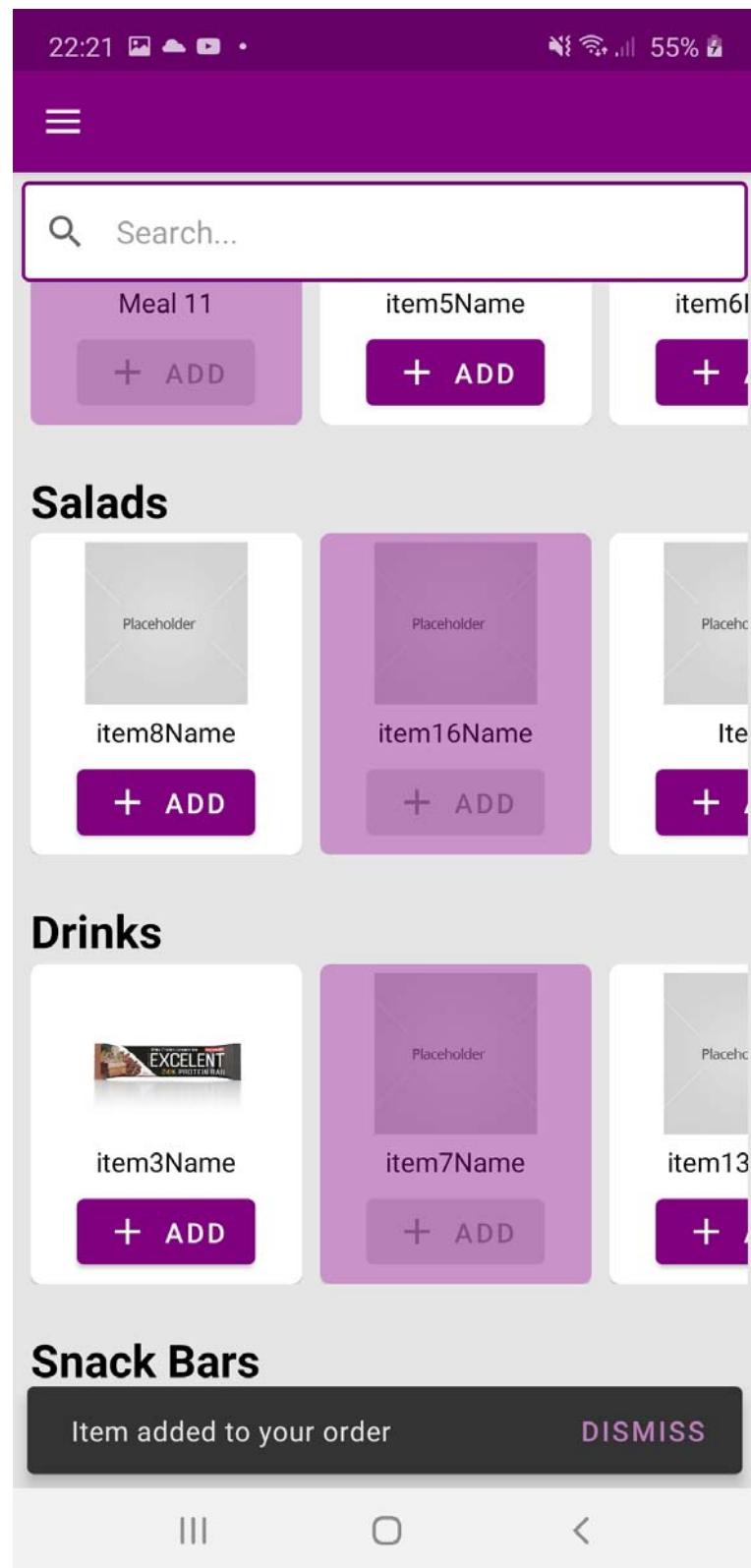
## Use Case: Filter Items

Use-Case Name:	Filter Items	
Use-Case ID:	UMM018	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Member</li> </ul>	
Description:	<p>The use case describes the event of an actor typing inside the filter bar, the system will filter out all the options that do not match the typed text.</p>	
Precondition:	<p>The member must be logged in and viewing the list of items.</p>	
Trigger:	<p>This use case is triggered upon typing inside the filter bar.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: The actor types inside the bar.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system filters out any irrelevant items.</li> </ul>
Alternate Courses:	<p>There are no Alt-Steps for this use case.</p>	
Conclusion:	<p>This use case concludes when the actor clears out the filter bar.</p>	
Post condition:	<p>The actor can now look for a specific item.</p>	



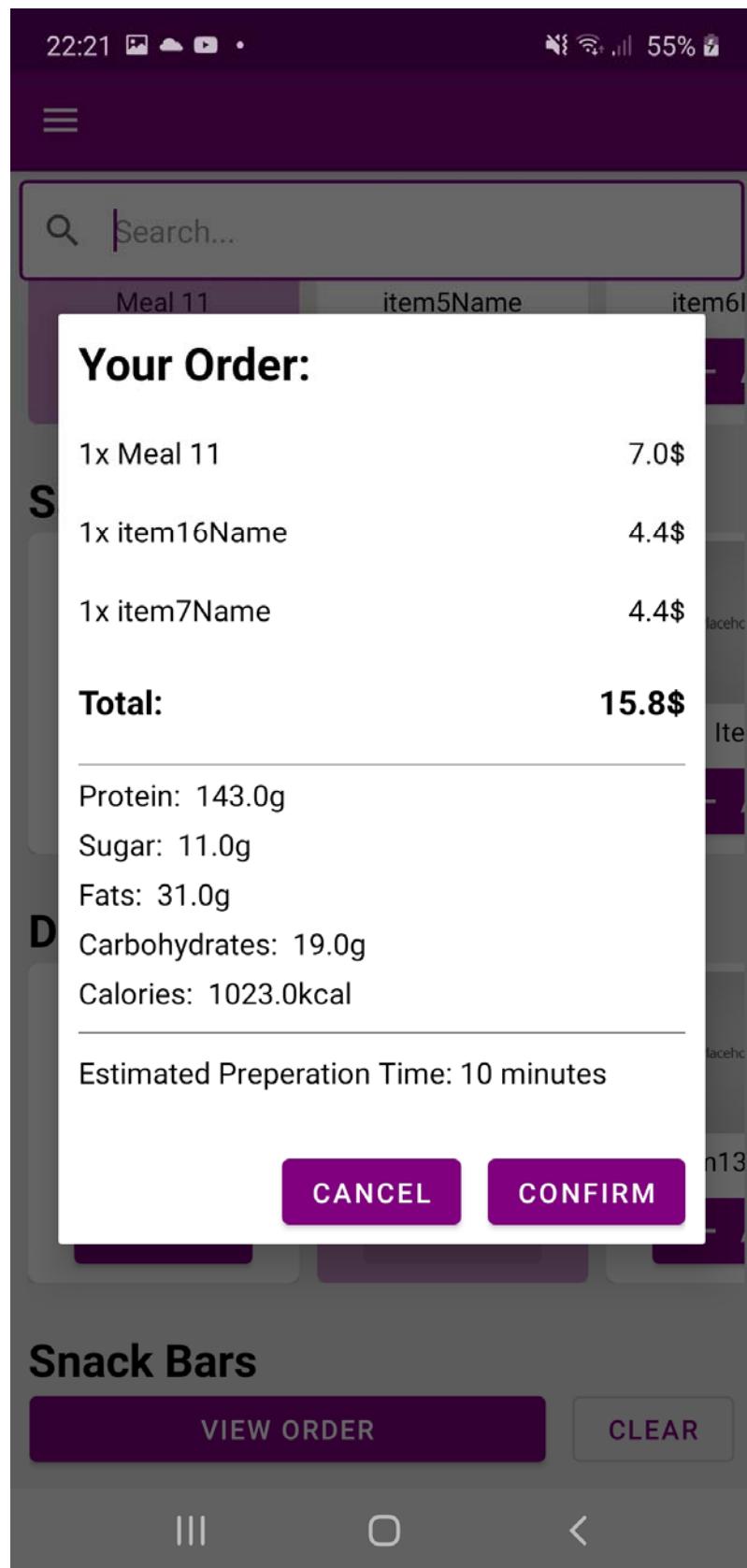
## Use Case: Add Items

Use-Case Name:	Add Items	
Use-Case ID:	UMM019	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Member</li> </ul>	
Description:	The use case describes the event of an actor Adding a specific item to the current order.	
Precondition:	The member must be logged in and viewing the order tab.	
Trigger:	This use case is triggered upon clicking on the “Add” button.	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: The actor clicks “Add”.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system updates the current order and adds the item..</li> </ul>
Alternate Courses:	There are no Alt-Steps for this use case.	
Conclusion:	This use case concludes when the item is added to the current order.	
Post condition:	The actor now add items to his/her order.	



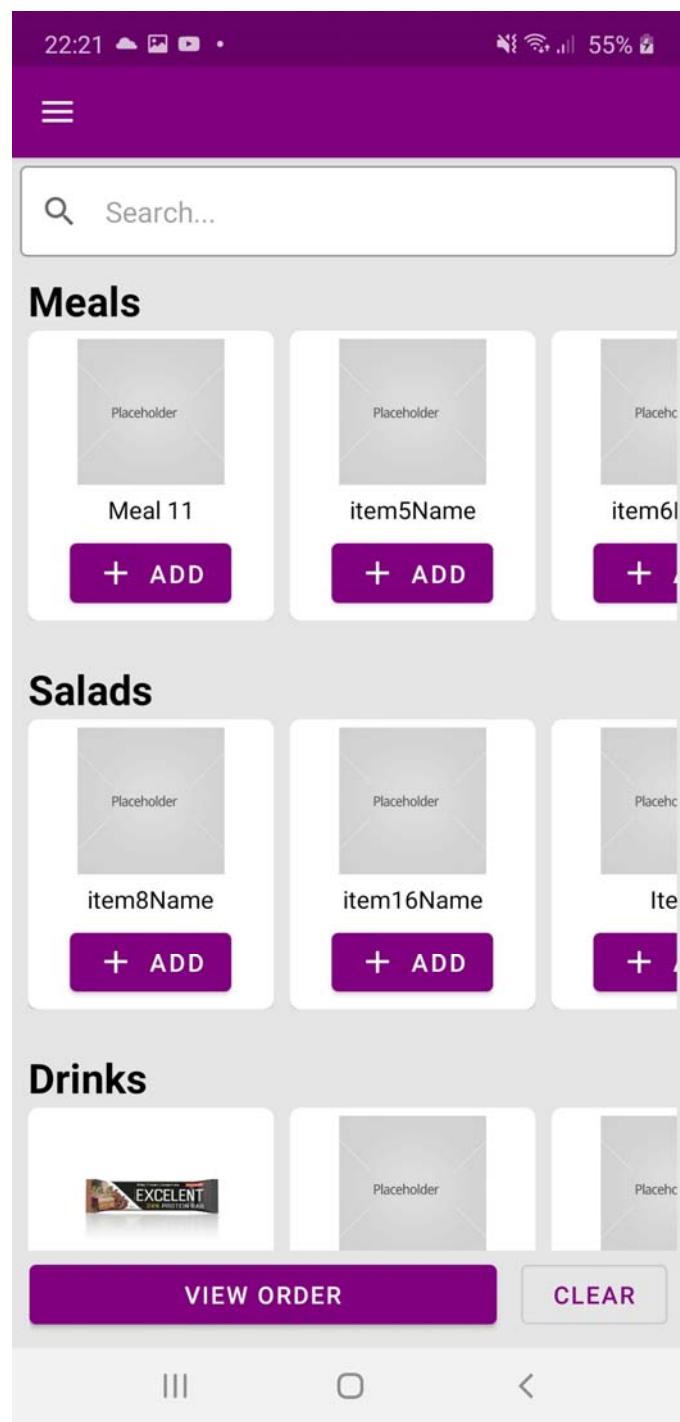
## Use Case: View Order

Use-Case Name:	View Order	
Use-Case ID:	UMM020	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Member</li> </ul>	
Description:	The use case describes the event of an actor viewing the current order to see its details.	
Precondition:	The member must be logged in and viewing the order tab.	
Trigger:	This use case is triggered upon clicking the “View Order” button.	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: The actor chooses to view the current order.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system shows a dialogue with the information about the current order.</li> </ul>
Alternate Courses:	There are no Alt-Steps for this use case.	
Conclusion:	This use case concludes when the actor closes the information dialogue.	
Post condition:	The actor now knows the details of his/her current order.	



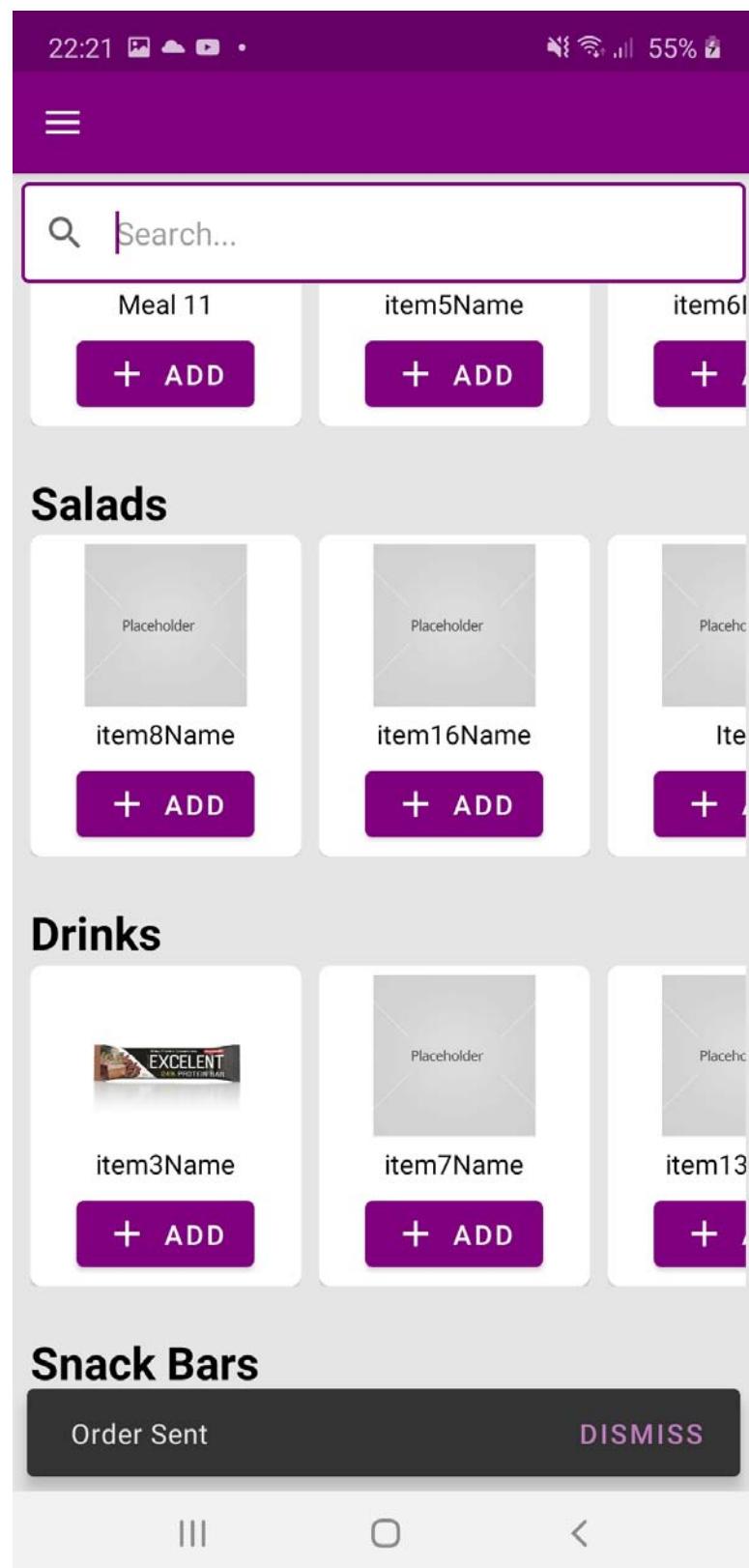
## Use Case: Clear Order

Use-Case Name:	Moves Preview	
Use-Case ID:	UMM021	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Member</li> </ul>	
Description:	<p>The use case describes the event of an actor deciding to clear the current order, the system will reset the current order.</p>	
Precondition:	<p>The member must be logged in and viewing the orders tab.</p>	
Trigger:	<p>This use case is triggered upon clicking the “Clear” button.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: The actor clicks “Clear”.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system resets the current order.</li> </ul>
Alternate Courses:	<p>There are no Alt-Steps for this use case.</p>	
Conclusion:	<p>This use case concludes when the current order is cleared.</p>	
Post condition:	<p>The actor can now start a new order.</p>	



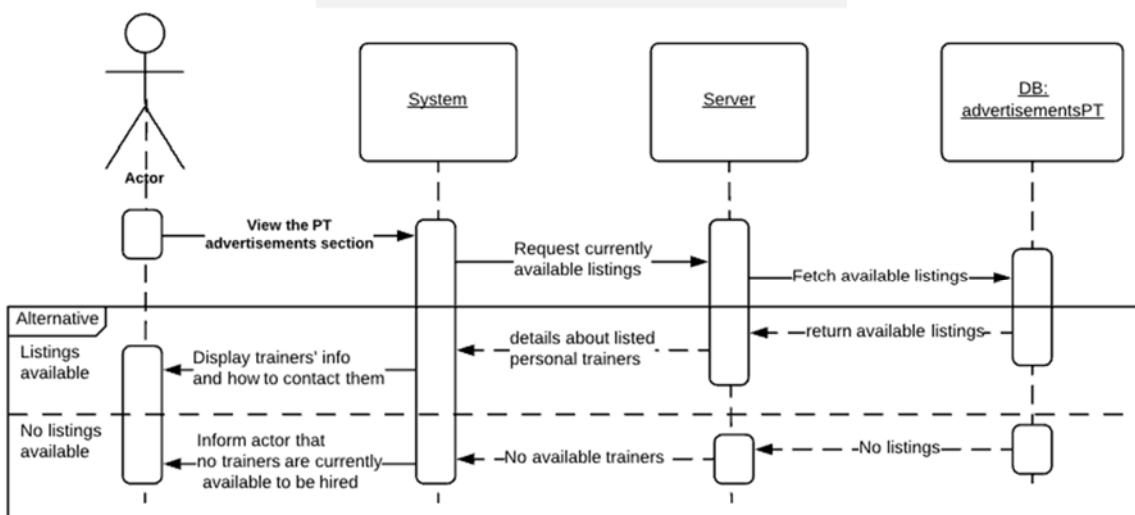
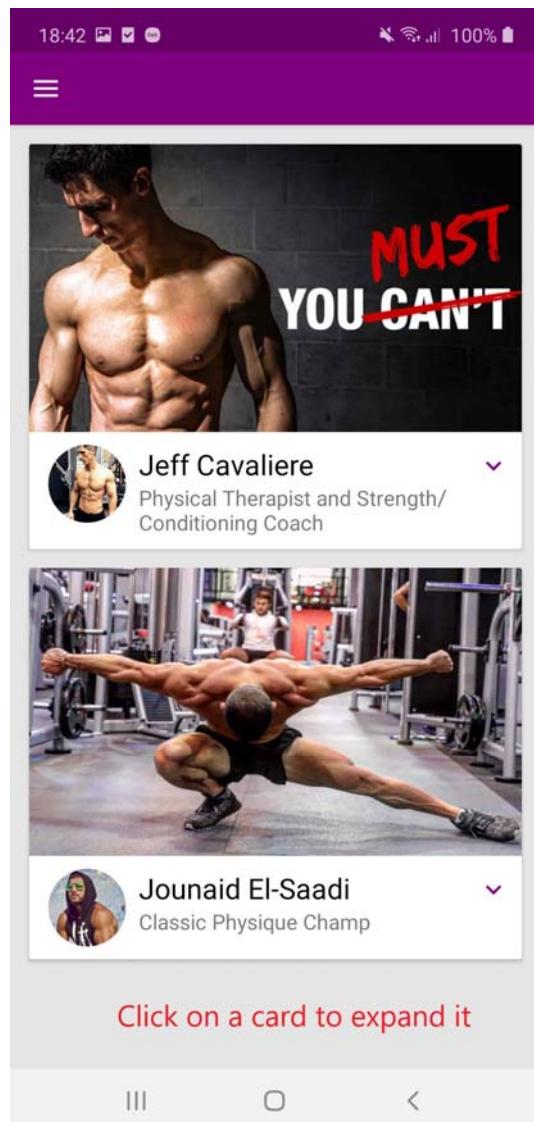
## Use Case: Confirm Order

Use-Case Name:	Moves Preview	
Use-Case ID:	UMM022	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Member</li> </ul>	
Description:	<p>The use case describes the event of an actor deciding to finalize the order and send it to the bartender.</p>	
Precondition:	<p>The member must have chosen at least one option from the menu.</p>	
Trigger:	<p>This use case is triggered upon clicking on the confirm in the order dialogue.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: The actor chooses to confirm the order..</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system sends the order to the bartender.</li> </ul>
Alternate Courses:	<p>There are no Alt-Steps for this use case.</p>	
Conclusion:	<p>This use case concludes when the order is sent to the bartender.</p>	
Post condition:	<p>The bartender can now see the member's order and prepare it.</p>	



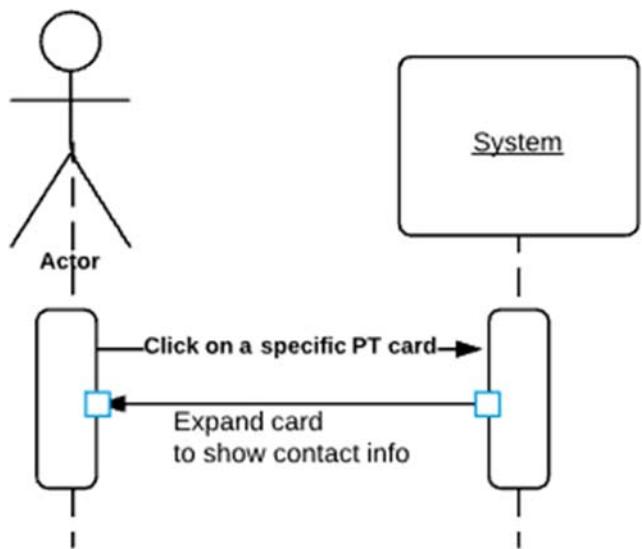
## Use Case: PT Advertisements

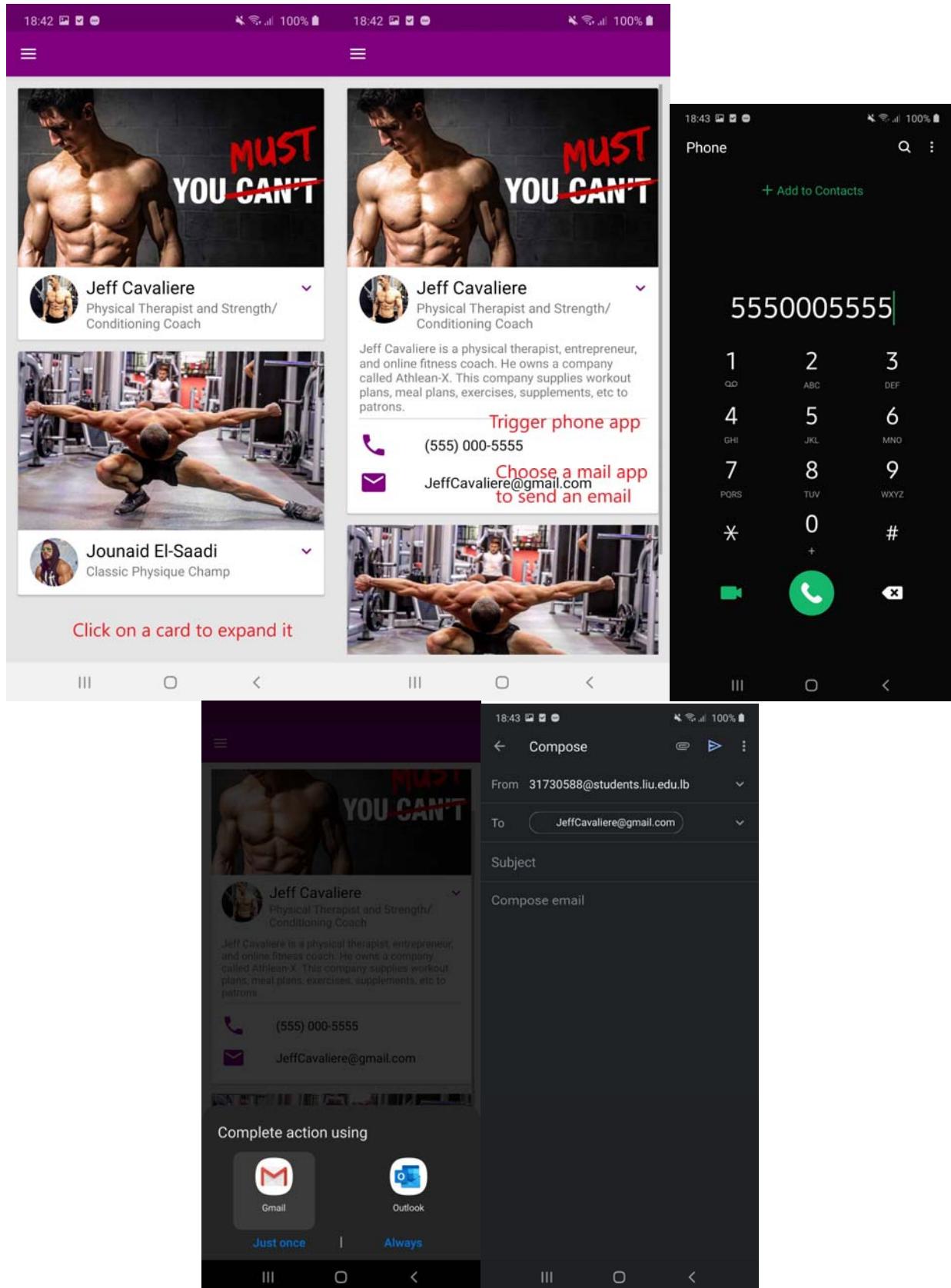
Use-Case Name:	PT Advertisements	
Use-Case ID:	UMM023	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Member</li> </ul>	
Description:	<p>The use case describes the event of an actor viewing the available advertisements of personal trainers on the system, the system will show a list of the personal trainers and a small description about them.</p>	
Precondition:	<p>The member must be logged in.</p>	
Trigger:	<p>This use case is triggered upon clicking the “Hire a Trainer!” option from the navigation drawer.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● The actor clicks on the “Hire a Trainer!” tab.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system opens an activity showing a list of all available personal trainers.</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● Alt-Step 2: There are currently no available personal trainers; the system shows a text to indicate that.</li> </ul>	
Conclusion:	<p>This use case concludes when the actor leaves the Personal Trainers activity.</p>	
Post condition:	<p>The actor can now check on the personal trainers and see if any of them can help him.</p>	



## Use Case: Contact PT

Use-Case Name:	Contact PT	
Use-Case ID:	UMM024	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Member</li> </ul>	
Description:	<p>The use case describes the event of an actor choosing a specific personal trainer and wanting to view his contact information, the system will display the personal trainer's email and phone number.</p>	
Precondition:	The member must be logged in.	
Trigger:	This use case is triggered upon clicking on a PT listing.	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● The actor clicks on a specific PT listing.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system expands the listing to show the contact information.</li> </ul>
Alternate Courses:	There are no Alt-Steps for this use case.	
Conclusion:	This use case concludes when the actor leaves the Personal Trainers activity.	
Post condition:	The actor now has the required information to contact the trainer he chose.	





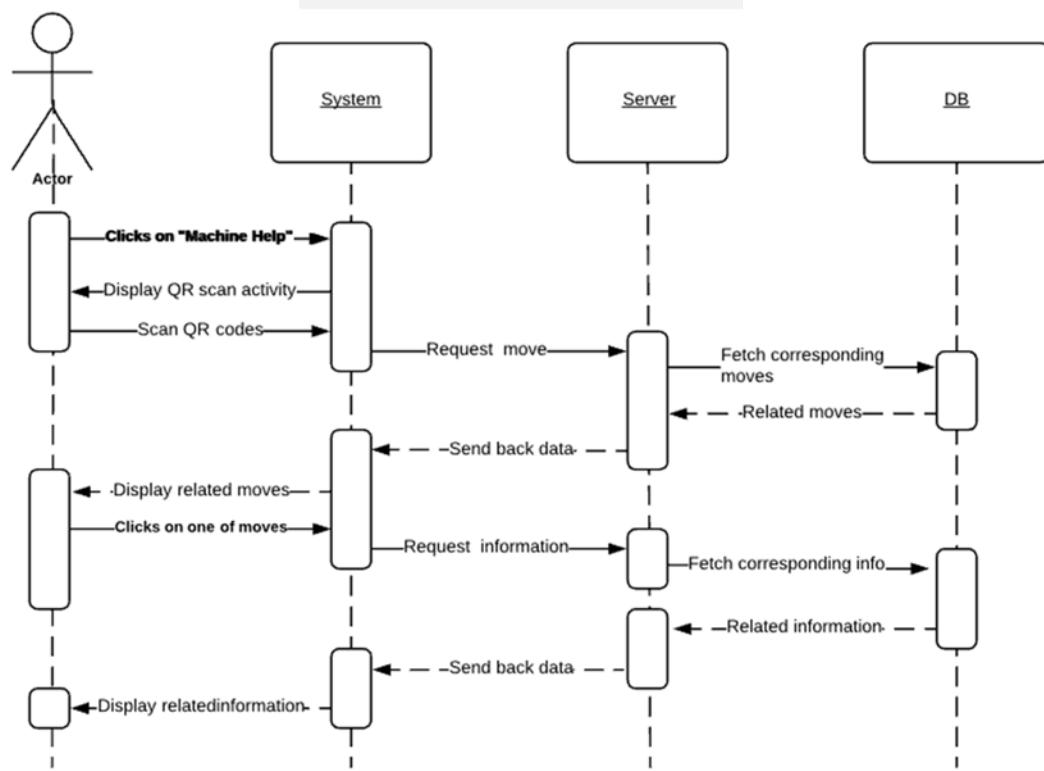
## Use Case: Machine Help

Use-Case Name:	Machine Help	
Use-Case ID:	UMM025	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Member</li> </ul>	
Description:	<p>The use case describes the event of an actor needing help operating a machine, the actor can then use this feature to scan a QR code to request information from the system.</p>	
Precondition:	<p>The member must be logged in.</p>	
Trigger:	<p>This use case is triggered upon clicking the “Machine Help” option from the navigation drawer.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● The actor clicks on the “Machine Help” tab.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system directs the user to the QR code scanning activity.</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● There are no Alt-Steps for this use case.</li> </ul>	
Conclusion:	<p>This use case concludes when the actor leaves the Machine Help activity.</p>	
Post condition:	<p>The actor can now scan the machine to get the required information.</p>	

Preview and Sequence Diagram are included with use case below

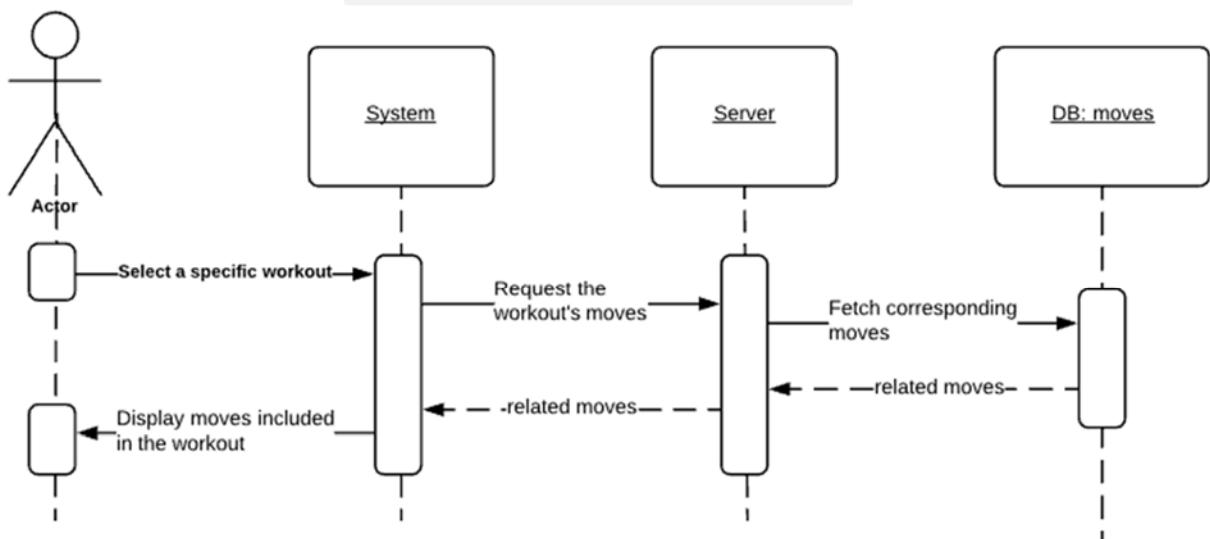
## Use Case: Scan QR Code

Use-Case Name:	Scan QR Code	
Use-Case ID:	UMM026	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Member</li> </ul>	
	<p>The use case describes the event of an actor deciding to scan the QR code on the machines to pull up information, the system will return details about the information and a list of possible moves that can be done on it.</p>	
Precondition:	<p>The member must be logged in and have a functional camera.</p>	
Trigger:	<p>This use case is triggered upon scanning a QR code.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: The actor scans a QR code.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system opens an activity showing the information about the machine and possible moves.</li> </ul>
Alternate Courses:	<p>There are no Alt-Steps for this use case.</p>	
Conclusion:	<p>This use case concludes when the actor leaves the activity that was created for the scanned machine.</p>	
Post condition:	<p>The actor now knows how to operate the machine.</p>	



## Use Case: Moves Preview

Use-Case Name:	Moves Preview	
Use-Case ID:	UMM027	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Member</li> </ul>	
Description:	<p>The use case describes the event of an actor viewing the specific moves that were pulled due to scanning a QR code.</p>	
Precondition:	<p>The member must be logged in and have pre scanned a QR code.</p>	
Trigger:	<p>This use case is triggered upon clicking on one of the shown possible moves.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: The actor chooses to view one of the moves.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system expands the listing to show more information about the move.</li> </ul>
Alternate Courses:	<p>There are no Alt-Steps for this use case.</p>	
Conclusion:	<p>This use case concludes when the actor leaves the activity that lists the possible moves.</p>	
Post condition:	<p>The actor now knows precisely how to perform a move.</p>	



## Use Case: Feedback

Use-Case Name:	Feedback	
Use-Case ID:	UMM028	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Member</li> </ul>	
Description:	<p>The use case describes the event of an actor deciding to send some feedback and suggestions regarding the gym.</p>	
Precondition:	<p>The member must be logged in.</p>	
Trigger:	<p>This use case is triggered upon clicking the “Feedback” option from the navigation drawer.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: The actor types in the subject of the feedback with its content and clicks send.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system stores the feedback and marks it as “unread” for later reviewing by the admin.</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● Alt-Step 1: The actor exceeded the maximum length of the feedback message; the system grays out the “Send” button and shows a warning.</li> </ul>	
Conclusion:	<p>This use case concludes when the feedback message is successfully stored in the database.</p>	
Post condition:	<p>The members can now share their ideas and suggestions on how to improve the gym.</p>	



### Message Us:

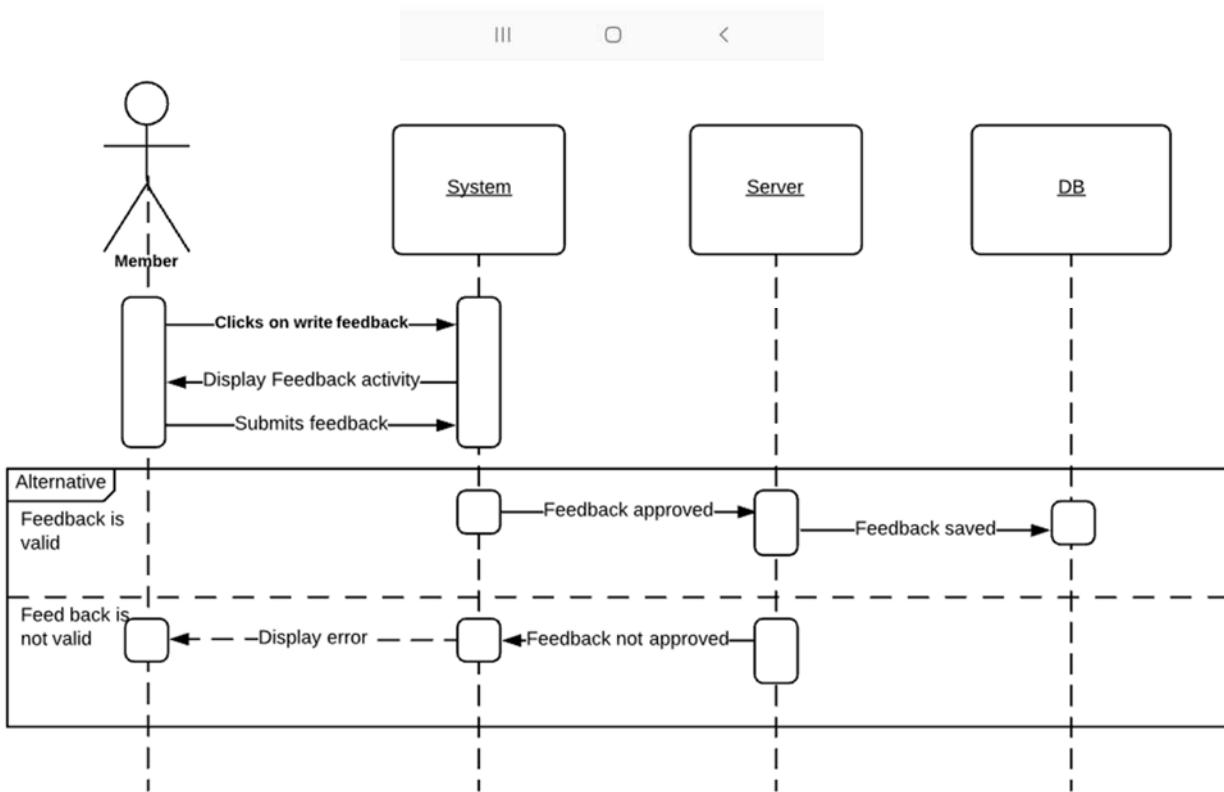
Subject..

Feedback Content...

**SEND FEEDBACK**

### Call Us:

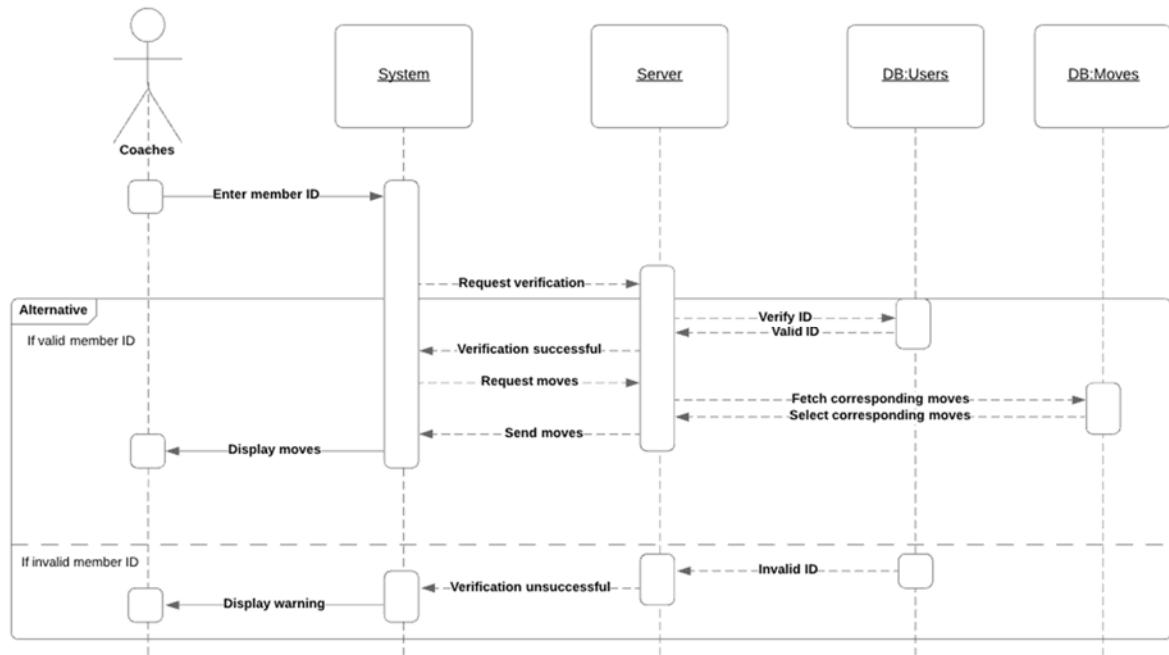
📞 (961) 71-100100



## Use Case: Assign Workout

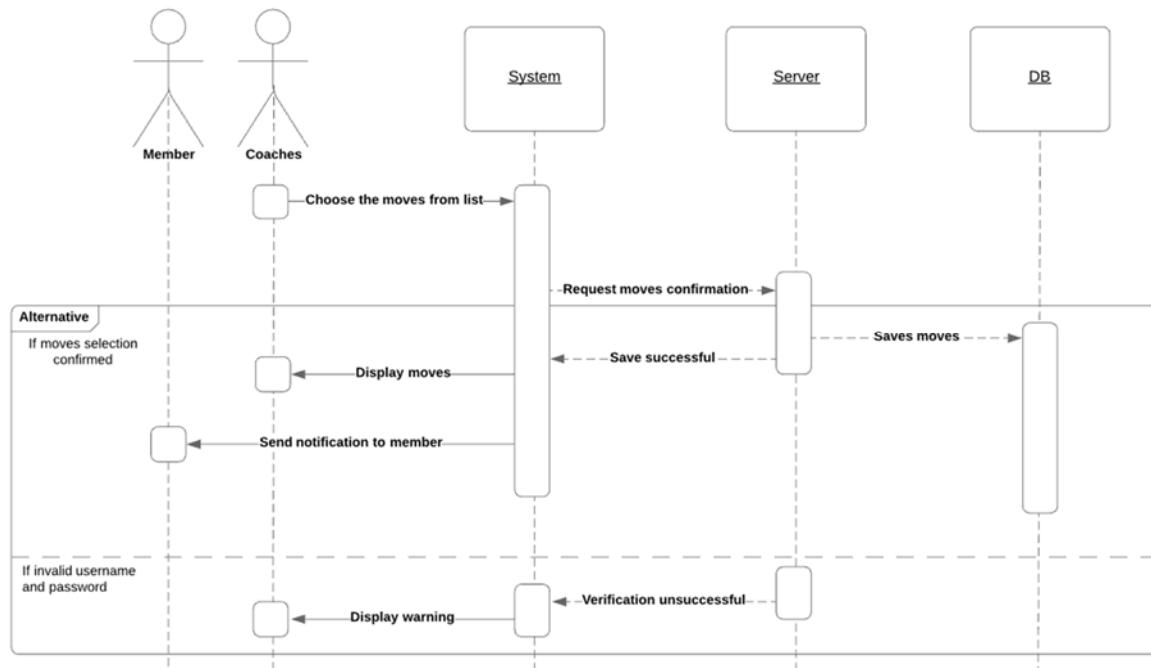
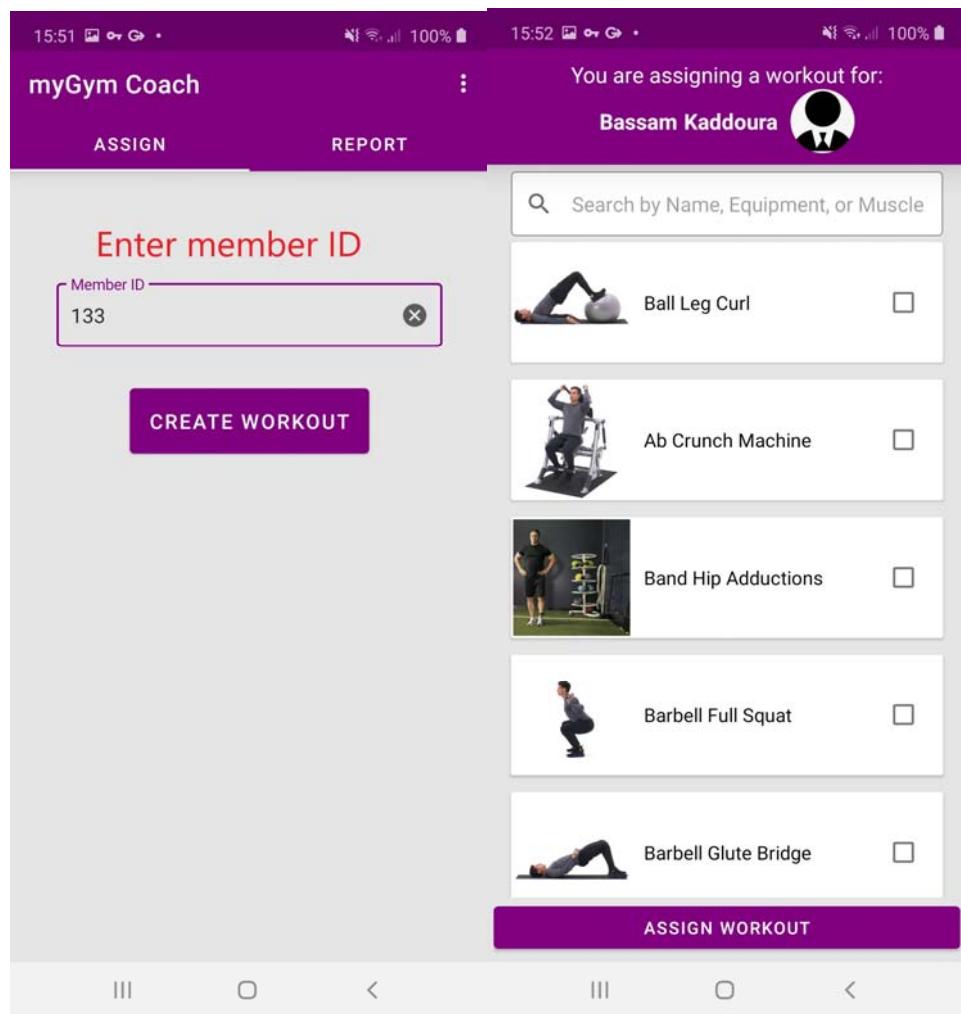
Use-Case Name:	Assign Workout	
Use-Case ID:	UMC002	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>• Coach</li> </ul>	
Description:	<p>The use case describes the event of an actor assigning a specific workout to a specific member, this will notify the member.</p>	
Precondition:	<p>The coach must be logged in and the entered member ID must be valid.</p>	
Trigger	<p>This use case is triggered upon clicking the “Assign Workout”.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>• Step 1: The actor enters the member's ID.</li> </ul>	<ul style="list-style-type: none"> <li>• Step 2: The system verifies that the ID exists in the database.</li> <li>• Step 3: The system lists all the possible moves to assign.</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>• Alt-Step 2: The entered ID is not valid; the system will warn the actor.</li> </ul>	
Conclusion:	<p>This use case concludes when the coach successfully assigns a workout notifying the member.</p>	
Post condition:	<p>The coach can now easily advise members with proper workouts.</p>	

Preview with the below use case



## Use Case: Select Moves

Use-Case Name:	Select Moves	
Use-Case ID:	UMC003	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Coach</li> </ul>	
Description:	<p>The use case describes the event of an actor viewing the workouts advised by a coach, the actor can then view the full details of the workout.</p>	
Precondition:	<p>The coach must be logged in and have entered a valid member ID.</p>	
Trigger	<p>This use case is triggered by entering a valid ID in use case UMC004.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: The actor chooses the moves from the list and confirms.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system confirms that at least one option has been chosen</li> <li>● Step 3: The system stores the workout in the database.</li> <li>● Step 4: The system notifies the relevant member.</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● Alt-Step 2: The actor didn't choose any option; the system displays a warning.</li> </ul>	
Conclusion:	<p>This use case concludes when the coach successfully assigns a workout notifying the member.</p>	
Post condition:	<p>The assigned workout is now stored in the database and can be viewed the member.</p>	



## Use Case: Send Report

Use-Case Name:	Send Report	
Use-Case ID:	UMC004	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Coach</li> </ul>	
Description:	<p>This use case describes the event of a coach reporting a specific incident or machine malfunction around the gym, the system will store the incident's details in the database.</p>	
Precondition:	<p>The coach must be logged in.</p>	
Trigger	<p>This use case is triggered upon entering the “Report Incident” tab.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: The actor chooses to create a new report.</li> <li>● Step 3: The actor fills all fields and submits.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system opens an activity showing the report form for the actor to fill.</li> <li>● Step 4: The system checks that all required fields have been filled.</li> <li>● Step 5: The system stores the report in the database.</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● Alt-Step 4: The actor left a required field empty; the system will display a warning.</li> </ul>	
Conclusion:	<p>This use case concludes when the report is successfully sent and stored in the database.</p>	
Post condition:	<p>The coach can report any incident or malfunction in the gym to let the relevant people know.</p>	

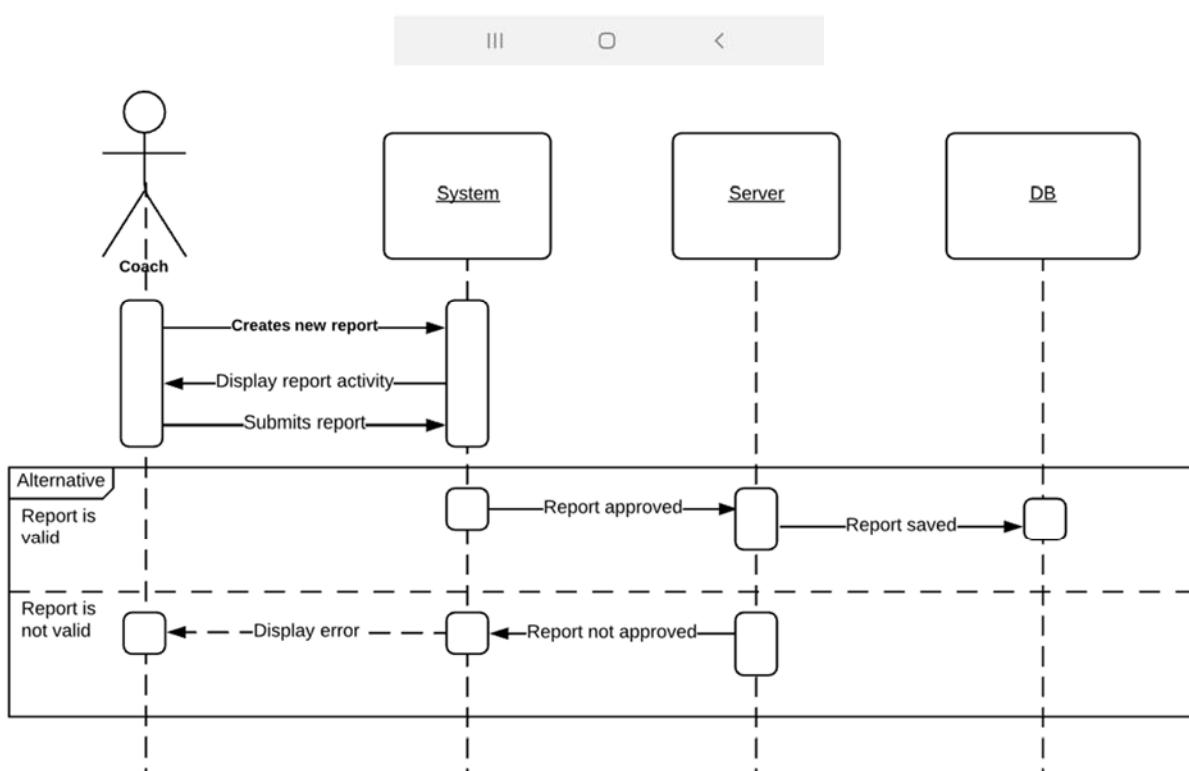


Subject..

---

Report Content...

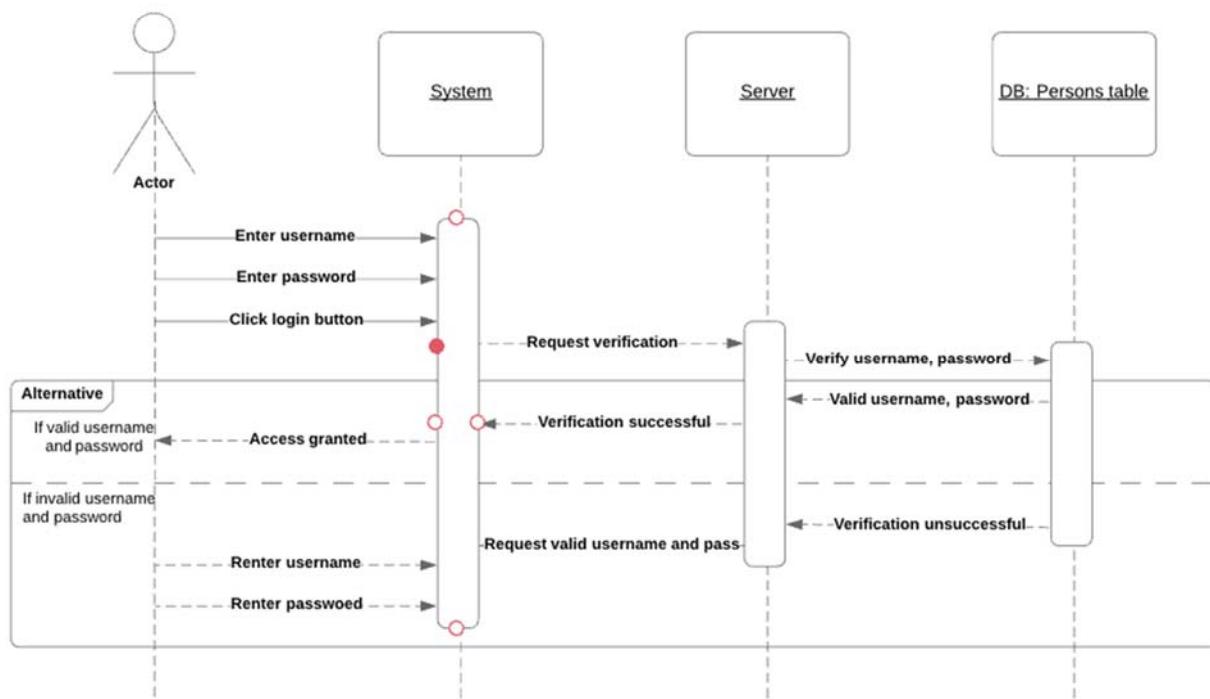
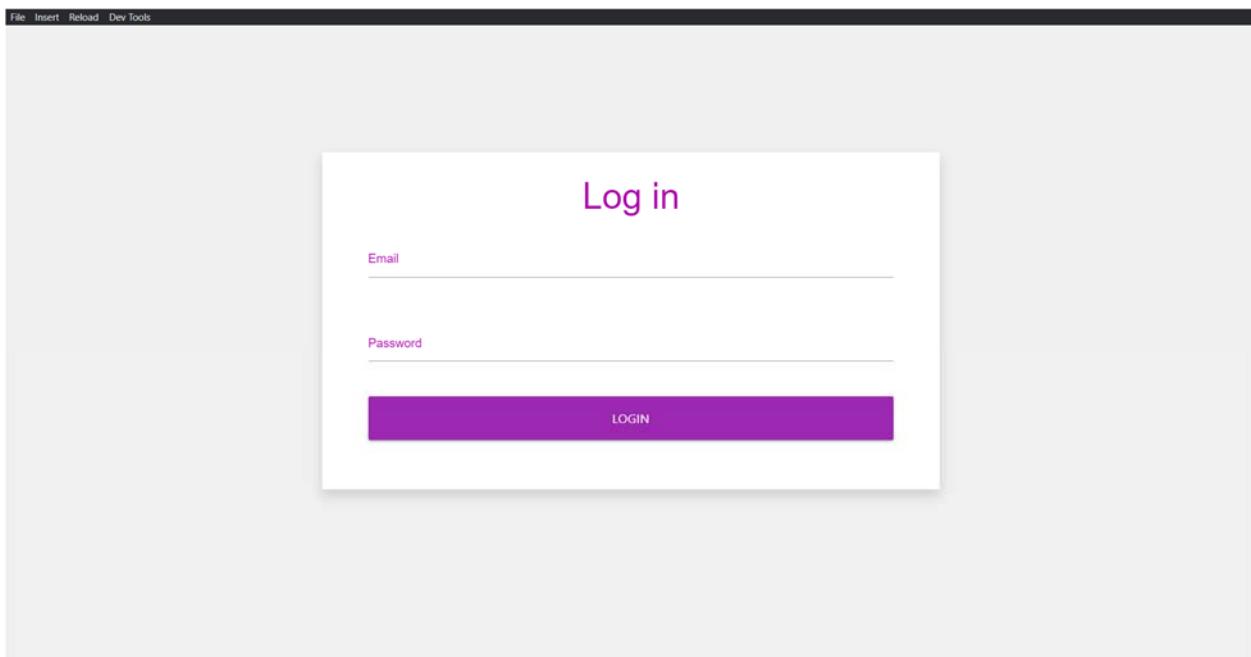
**SEND REPORT**



# Desktop Application Narratives, Forms, and Sequence Diagrams

## Use Case: Log in

Use-Case Name:	Login	
Use-Case ID:	UDA001	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Admin</li> </ul>	
Description:	<p>Admin user enters his username and password, the system authenticates the data entered, then the user gains access to his admin account with admin privileges</p>	
Precondition:	<p>To login as admin you have to be a registered admin.</p>	
Trigger	<p>This use case is triggered upon launching Admin's desktop application</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: The actor enters his username.</li> <li>● Step 2: The actor enters his password.</li> <li>● Step 3: The actor clicks on the "Login" button.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 4: The system verifies that an account with this username exists.</li> <li>● Step 5: The system verifies that the password matches the account linked to the username.</li> <li>● Step 6: The system logs the actor into his account.</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● Alt-Step 4: An account with that username doesn't exist; the system displays a warning.</li> <li>● Alt-Step 5: The entered password is wrong; the system displays a warning.</li> </ul>	
Conclusion:	<p>This use case concludes when the actor is logged into the system.</p>	
Post condition:	<p>The actor can now perform admin-level actions.</p>	

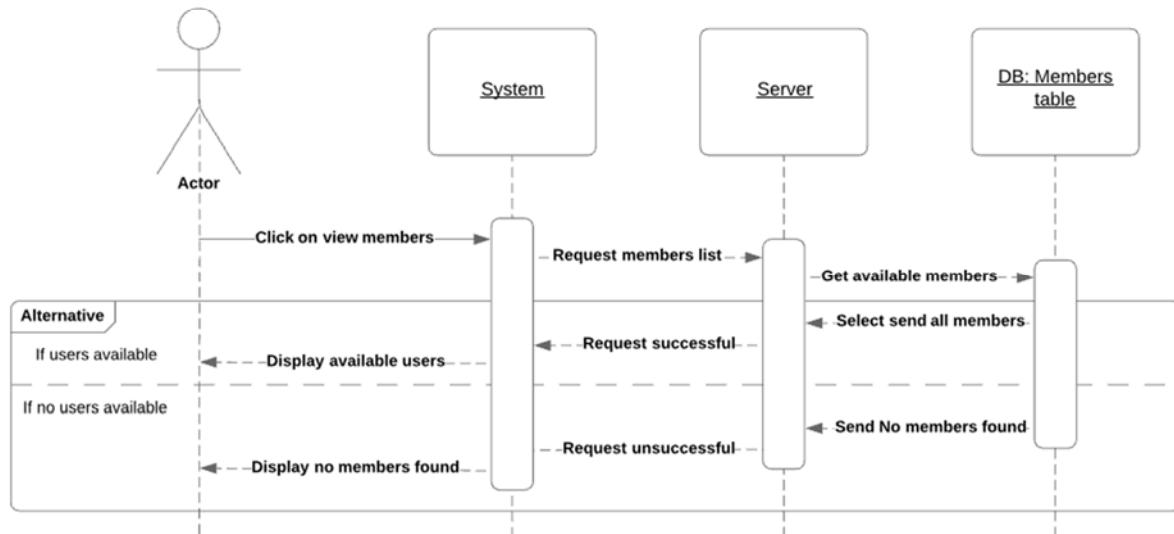


## Use Case: View Members

Use-Case Name:	View Members	
Use-Case ID:	UDA002	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Admin</li> </ul>	
Description:	<p>An admin user can view all the members on the system and their relative details, the admin can also activate and deactivate the account and monitor subscription expiration dates.</p>	
Precondition:	<p>The actor has to be logged in as an admin.</p>	
Trigger	<p>Click on “View members” button on the admin panel</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: Click on the “View members” button in the admin panel.</li> <li>● Step 3: Admin can scroll between all member’s records and view detailed profile and log information</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: System will show the members’ records and logs in a table opened in a new window.</li> </ul>
Alternate Courses:	<p>There are no Alt-Steps for this use case.</p>	
Conclusion:	<p>This use case concludes when all members are listed along with their records.</p>	
Post condition:	<p>The actor is now able to go through all the members and their details.</p>	

File Insert Reload Dev Tools

MEMBERS		COACHES		BARTENDERS		WORKOUTS		MOVES		MACHINES		MENU		PT ADS		MESSAGES	
ID	Email	First	Last	Password	Phone	Address	Image	Activity	Expire	Joined	Actions						
<b>Search in a specific column</b>																	
11	makdah21@gmail.com	Mohammed	Mohammed	MobPsycho100	(961) 39393	-		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
58	Work@Here.Why	This	Not1234	BbbbBBB123	(961) 111111	-		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>							
59	31730588@students.liu.se	Kaddoura	Kaddoura	BsmKdd11	(961) 111111	-		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>							
88	Bassam-bla-bla@hotmail.com	Bassam	Kaddoura	Bassam1234	(1242) 82201000	-		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>							
89	Bassamblablabla@gmail.com	Bassam	Kaddoura	Bassam1234	(1242) 82201000	-		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>							
ID	Email	First	Last	Password	Phone	Address	Image	Activity	Expire	Joined	Actions						
1 - 5 / 9 (9)		5	Records per page	1	Page number												



## Use Case: Add Members

Use-Case Name:	Add Members	
Use-Case ID:	UDA003	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Admin</li> </ul>	
Description:	<p>Admin users can add new members to the database through a form in the admin panel, the admin enters all the needed information about the member and then triggers the form.</p>	
Precondition:	<p>To view members the actor has to be logged in as an admin.</p>	
Trigger	<p>Click on the “Add member” button.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: Click on the “Add member” button in the admin panel.</li> <li>● Step 3: fill the required fields in the forms.</li> <li>● Step 4: Click on the “Add” button.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system will open a new window with the form of the required information to be filled.</li> <li>● Step 5: The system saves new records in the database.</li> <li>● Step 6: System shows a message “Member added successfully”.</li> <li>● Step 7: System adds the new member to the member list so it can be seen by the admin in UDA002</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● Alt-Step 5: If entered data is incorrect, system requests admin to reenter.</li> <li>● Alt-Step 6: If the addition of the member was unsuccessful, the system will return and display an error.</li> </ul>	
Conclusion:	<p>This use case concludes when the member is successfully added to the system</p>	
Post condition:	<p>The admin was able to update the database with a new member record.</p>	

File Insert Reload Dev Tools

New Record Ctrl+Shift+N COACHES BARTENDERS WORKOUTS MOVES MACHINES MENU

ID	Email	First	Last	Password	Phone	Address	Image
11	ma						
59	31						
88	Bar						
89	Bar						
90	BS						

Choose "New Record" or click Ctrl+Shift+N to trigger modal based on the currently active tab

ID	Email	First
1	5	1 - 5 / 7 (7) > > 5 < <

First Name \_\_\_\_\_ Last Name \_\_\_\_\_

Email \_\_\_\_\_

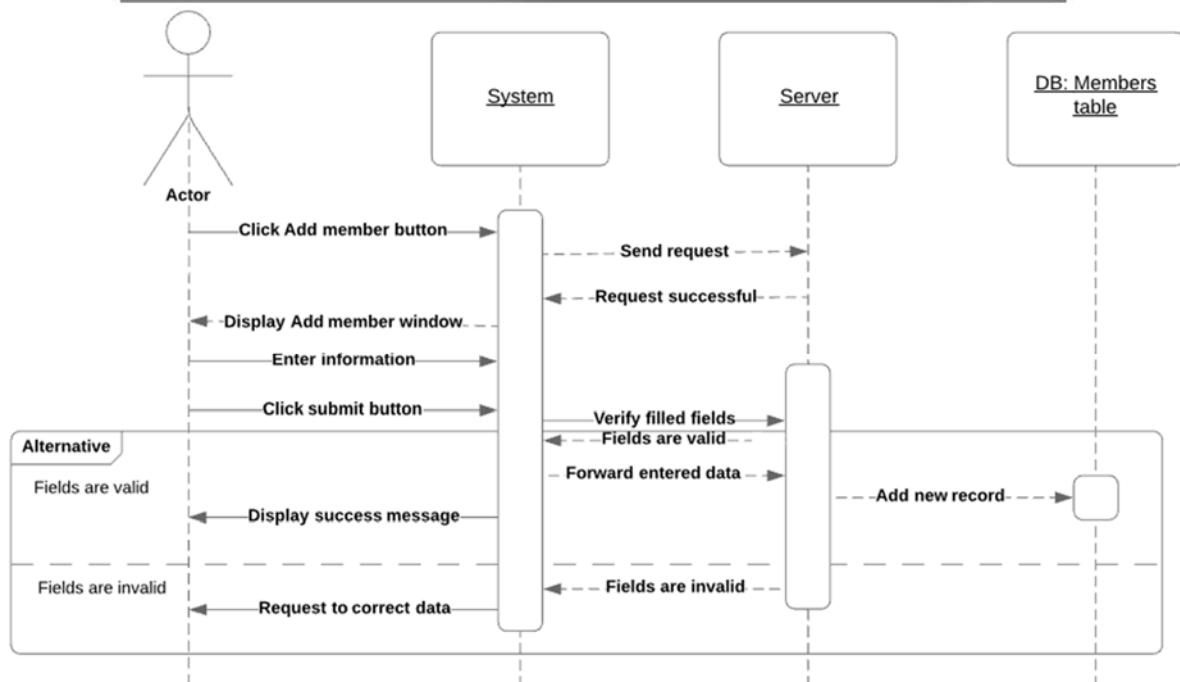
Password \_\_\_\_\_

Country: LB (961) \_\_\_\_\_ Telephone: \_\_\_\_\_

Address \_\_\_\_\_

Subscription Plan:

1 month    3 months    12 months  
45\$                  120\$                  400\$



## Use Case: Edit Members

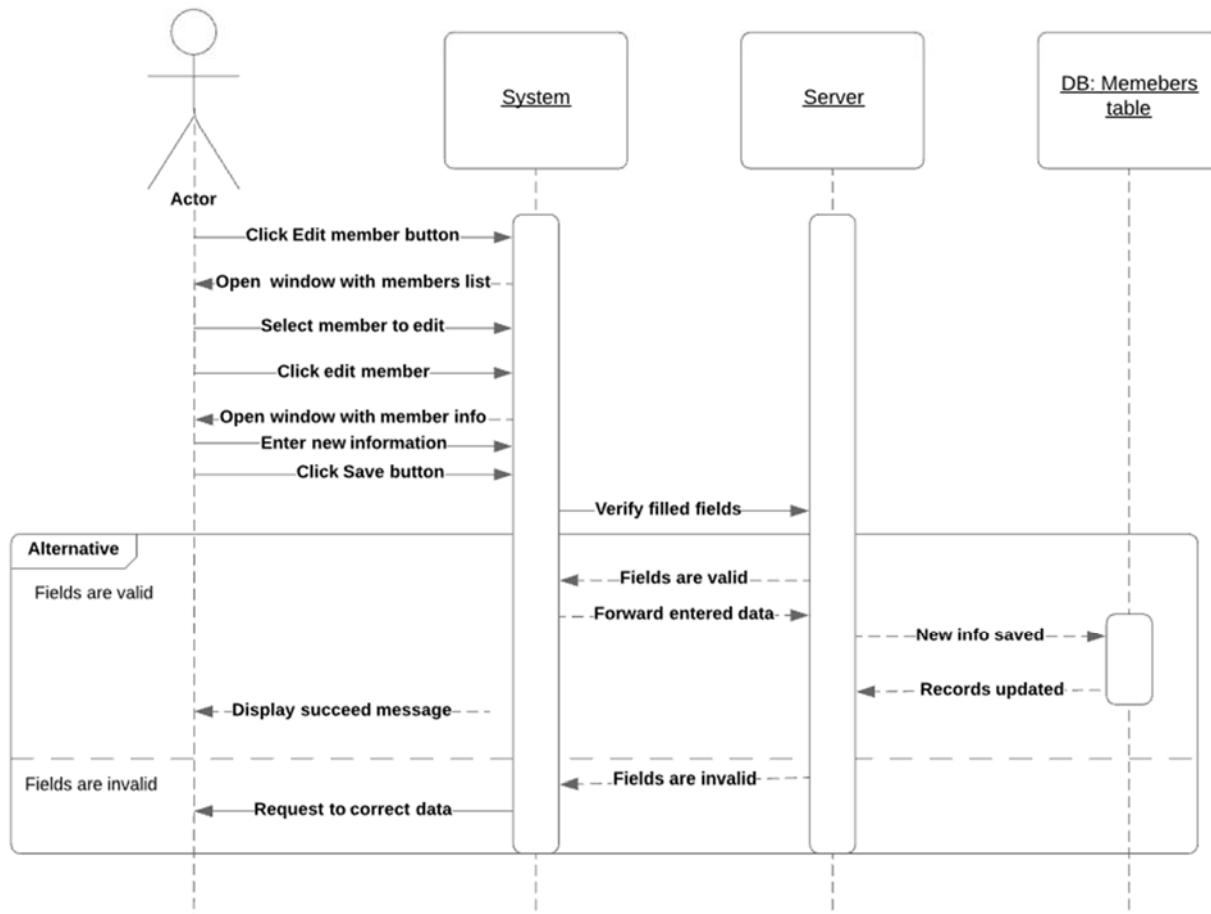
Use-Case Name:	Edit Members	
Use-Case ID:	UDA004	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Admin</li> </ul>	
Description:	Admin users can edit members' details and information in the database, including their emails, names, addresses, phone numbers, etc..	
Precondition:	To edit members, the actor has to be logged in as an admin.	
Trigger	Click on "Edit members" button	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: Click on the "Edit" button beside a member.</li> <li>● Step 3: Admin edits the information to be changed</li> <li>● Step 4: Click the "Save" button.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: System opens a new window with a form filled with all coaches' information.</li> <li>● Step 5: System verifies and saves the changes, then shows an "Update Successful" message.</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● Alt-Step 8: The editing was unsuccessful; system asks the actor to retry.</li> </ul>	
Conclusion:	This use case concludes when a member's record is successfully updated.	
Post condition:	The admin was able to edit the necessary information and details for any member.	

File Insert Reload Dev Tools

MEMBERS	COACHES	BARTENDERS	WORKOUTS	MOVES	MACHINES	MENU	PT ADS	MESSAGES			
ID Email		First	Last	Password	Phone	Address	Image	Activity	Expire	Joined	Actions
11 makdah21@gmail.com	Mohammad	Mohammed	MobPsycho100	(961) 7139253			<input checked="" type="checkbox"/>				
88 Bassam-bla-bla@hotmail.com	Bassam	Kaddoura	BsmKdd11	(961) 111111			<input type="checkbox"/>				
89 Bassamblabla@gmail.com	Bassam	Kaddoura	Bassam1234	(1242) 82201000			<input type="checkbox"/>				
ID Email		First	Last	Password	Phone	Address	Image	Activity	Expire	Joined	Actions

1 - 5 / 9 (9) < > 5 ⌂ 1 ⌂

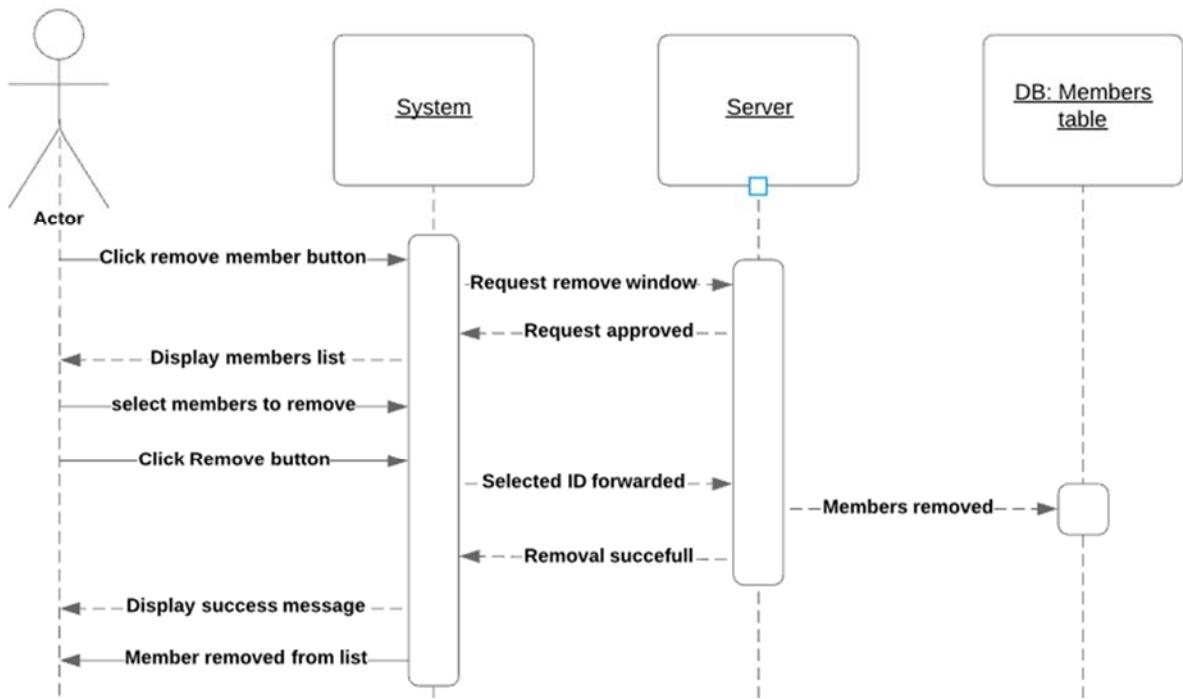
**Double click on a field to make it editable**



## Use Case: Remove Members

Use-Case Name:	Remove Members	
Use-Case ID:	UDA005	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Admin</li> </ul>	
Description:	<p>The admin has the ability to remove member records from the database (this should be done in rare cases).</p>	
Precondition:	<p>To remove members, the actor has to be logged in as an admin.</p>	
Trigger	<p>Click on the “Remove member” button beside a member’s record.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: Click On “Remove member” button in the admin panel.</li> <li>● Step 3: The actor confirms.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system will open a confirmation dialogue.</li> <li>● Step 4: The system removes the selected record from the database.</li> <li>● Step 5 : System show the message “Member removed successfully”.</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● Alt-Step 3: The actor cancels the process and doesn’t confirm; the action doesn’t proceed.</li> </ul>	
Conclusion:	<p>This use case concludes when the record is successfully removed from the database.</p>	
Post condition:	<p>The admin removed the specific record and the database is updated.</p>	

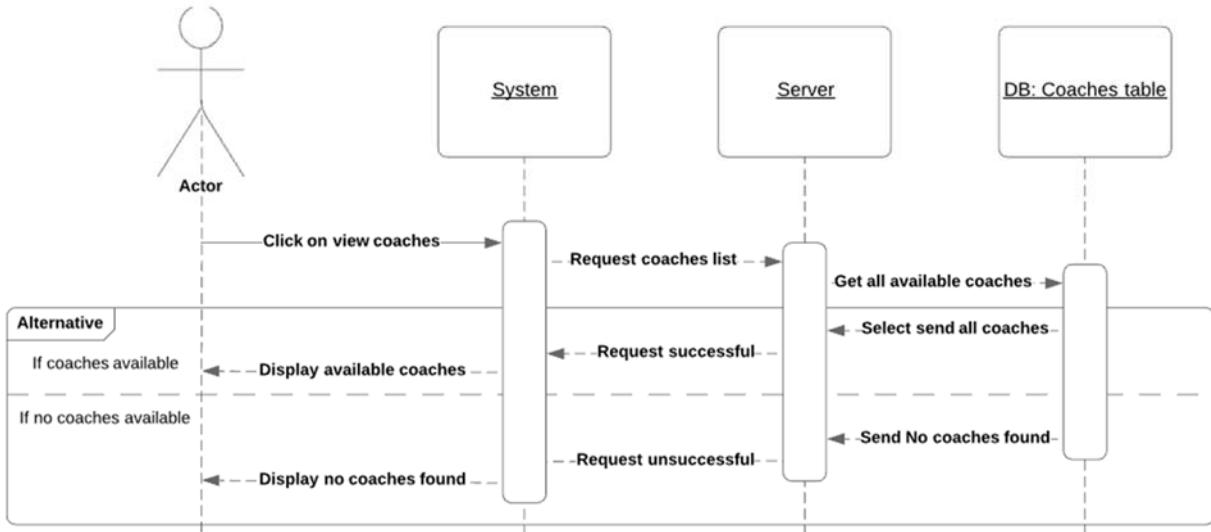
## Preview Image included in use case UDA003



## Use Case: View Coaches

Use-Case Name:	View Coaches	
Use-Case ID:	UDA006	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Admin</li> </ul>	
Description:	<p>Admin users can look through coaches details and view their activity, such as the reports they have submitted and the assistance requests they have responded to.</p>	
Precondition:	<p>The actor has to be logged in as an admin.</p>	
Trigger	<p>Click on the “View coaches” button on the admin panel.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: Click on the “View coaches” button in the admin panel.</li> <li>● Step 3: Admin can scroll between all coaches records and view detailed profile and log information.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system will show coaches' records and logs in a table that opens a new window.</li> </ul>
Alternate Courses:	<p>There are no Alt-Steps for this use case.</p>	
Conclusion:	<p>This use case concludes when all available coaches are listed along with their records.</p>	
Post condition:	<p>The admin is able to view all the coaches along with their information.</p>	

MEMBERS	COACHES	BARTENDERS	WORKOUTS	MOVES	MACHINES	MENU	PT ADS	MESSAGES				
ID	Email	First	Last	Password	Phone	Address	Image	Expertise	Description	Salary	Hired	Actions
2	coach@coach.coach	coachFName	coachLName	coach1234	123456789			Weight Loss		2940	2020-03-03	
4	Coach1@gmail.com	Coach1FName	Coach1LName	(961) 11111111	Coach1Password			Expertise 1		2500	2020-04-25	
6	Coach3@gmail.com	Coach3FName	Edited	(961) 11111111	Coach3Password			Expertise 3		2500	2020-04-02	
7	Coach4@gmail.com	Coach4FName	Coach4LName	(961) 11111111	Coach4Password			Expertise 4		2500	2020-04-20	
9	Coach6@gmail.com	Coach6FName	Coach6LName	(961) 11111111	Coach6Password			Expertise 6		2500	2020-04-22	



## Use Case: Add Coaches

Use-Case Name:	Add Coaches	
Use-Case ID:	UDA007	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Admin</li> </ul>	
Description:	<p>Admin users can add new coaches to the database through a form in the admin panel, the admin enters all the needed information about the coach and then triggers the form.</p>	
Precondition:	<p>The actor has to be logged in as an admin.</p>	
Trigger	<p>Click on the “Add coach” button.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: Click on the “Add coach” button in the admin panel.</li> <li>● Step 3: Fill the required fields in the forms.</li> <li>● Step 4: Clicks on the “Add” button.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system opens a new window with a form of the required information to be filled.</li> <li>● Step 5: System saves new records in the database.</li> <li>● Step 6: System shows a message “Coach added successfully”.</li> <li>● Step 7: System adds the new coach to the members list so it can be seen by the admin UDA006</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● Alt-Step 5: If entered data is incorrect; the system doesn't save the record</li> <li>● Alt-Step 6: The addition was unsuccessful; a message is displayed accordingly.</li> </ul>	
Conclusion:	<p>The use case concludes when the coach is successfully added.</p>	
Post condition:	<p>The admin was able to add a new coach and update the database.</p>	

coach@coach.coach	coachFName	coachLN
coach1@gmail.com	Coach1FName	Coach1LN
coach3@gmail.com	Coach3FName	Edited
coach4@gmail.com	Coach4FName	Coach4LN
coach6@gmail.com	Coach6FName	Coach6LN
mail	First	Last

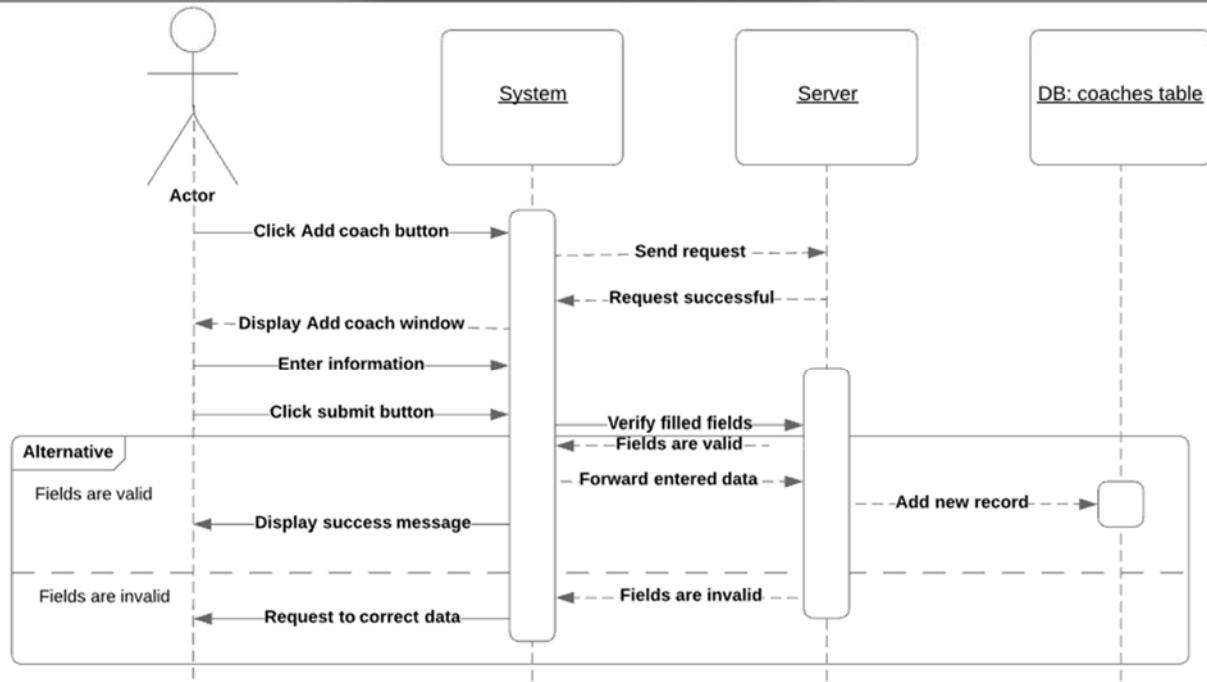
1 - 5 / 9 (9) > >> 5 < << 1

Expertise				
Country	LB (961) ▾	Telephone		
Salary				
Description				
Address				

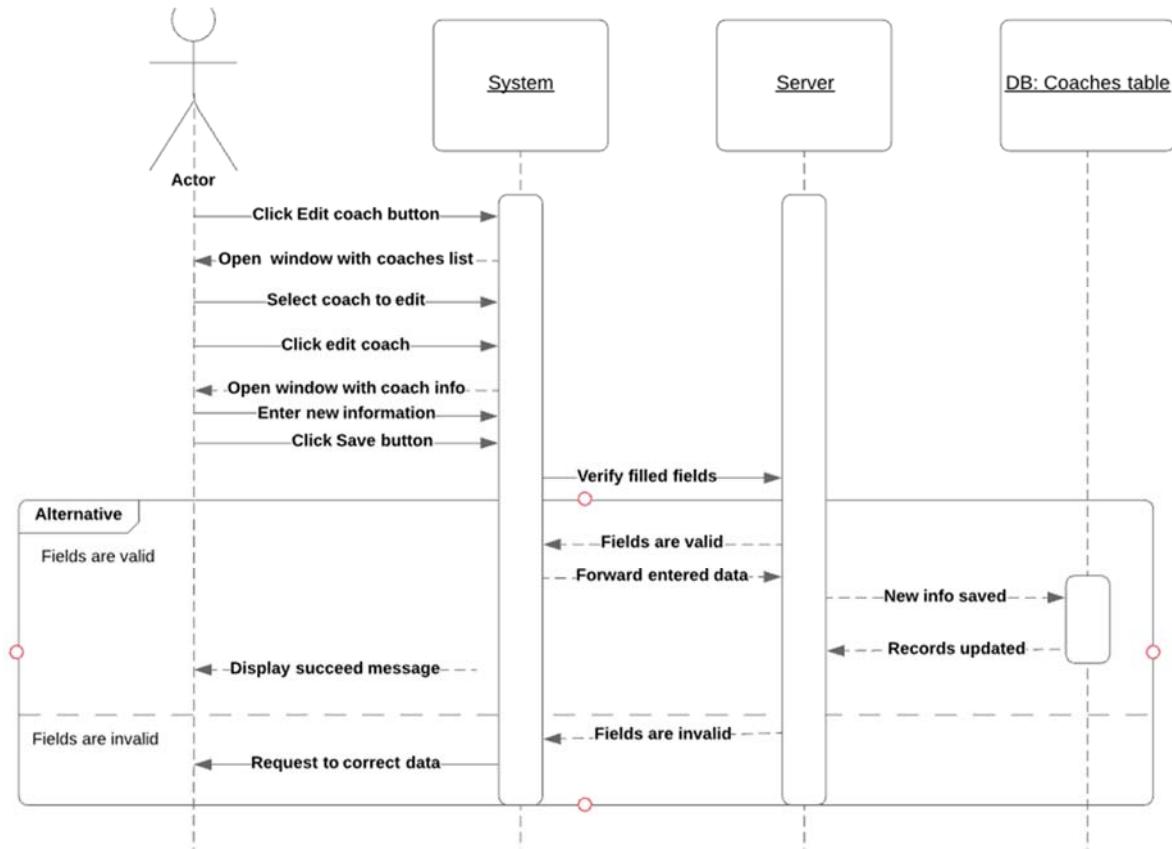
Weight Loss	2940	2020-03-03	Actions
Expertise 1	2500	2020-04-25	trash
Expertise 3	2500	2020-04-02	trash
Expertise 4	2500	2020-04-20	trash
Expertise 6	2500	2020-04-22	trash



## Use Case: Edit Coach

Use-Case Name:	Edit Coach	
Use-Case ID:	UDA008	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Admin</li> </ul>	
Description:	Admin users can edit members' details and information in the database, including their emails, expertise, names, addresses, phone numbers, etc..	
Precondition:	The actor has to be logged in as an admin.	
Trigger	Click on the “Edit” button beside the coach.	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: Click on the “Edit” button beside a coach.</li> <li>● Step 3: Admin edits the information to be changed</li> <li>● Step 4: Click the “Save” button.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: System opens a new window with a form filled with all coaches' information.</li> <li>● Step 5: System verifies and saves the changes, then shows an “Update Successful” message.</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● Alt-Step 5: The update was unsuccessful; the system displays a message accordingly.</li> </ul>	
Conclusion:	This use case concludes when the coach record is successfully updated.	
Post condition:	The admin was able to edit the record and update the database.	

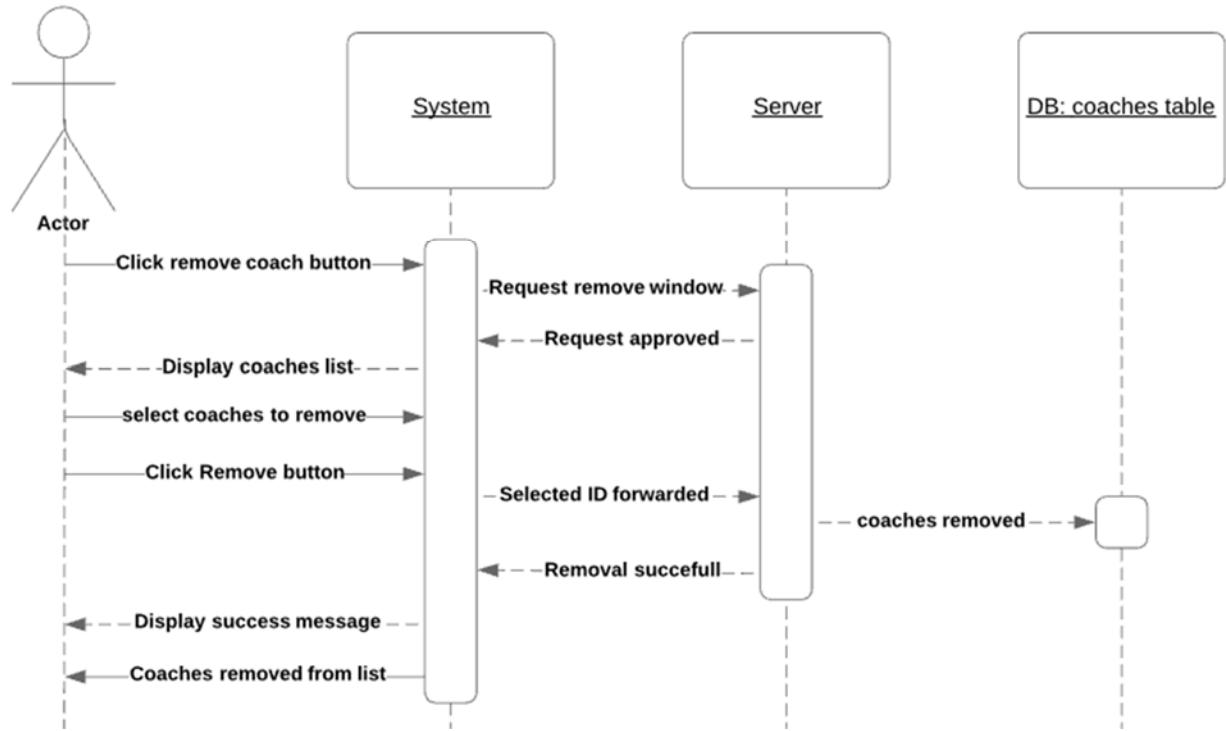
## Preview Image included in use case UDA006



## Use Case: Remove Coach

Use-Case Name:	Remove Coach	
Use-Case ID:	UDA009	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Admin</li> </ul>	
Description:	<p>The admin has the ability to remove coach records from the database (this should be done in rare cases).</p>	
Precondition:	<p>The actor has to be logged in as an admin.</p>	
Trigger	<p>Click on the “Remove” button beside a coach.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: Click On “Remove” button in the admin panel.</li> <li>● Step 3: The actor confirms</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system will open a confirmation dialogue.</li> <li>● Step 4: The system removes the selected record from the database.</li> <li>● Step 5 : System shows the message “coach removed successfully”.</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● Alt-Step 3: The actor cancels the process and doesn't confirm; the action doesn't proceed.</li> </ul>	
Conclusion:	<p>This use case concludes when the record is successfully removed from the database.</p>	
Post condition:	<p>The admin removed the specific record and the database is updated.</p>	

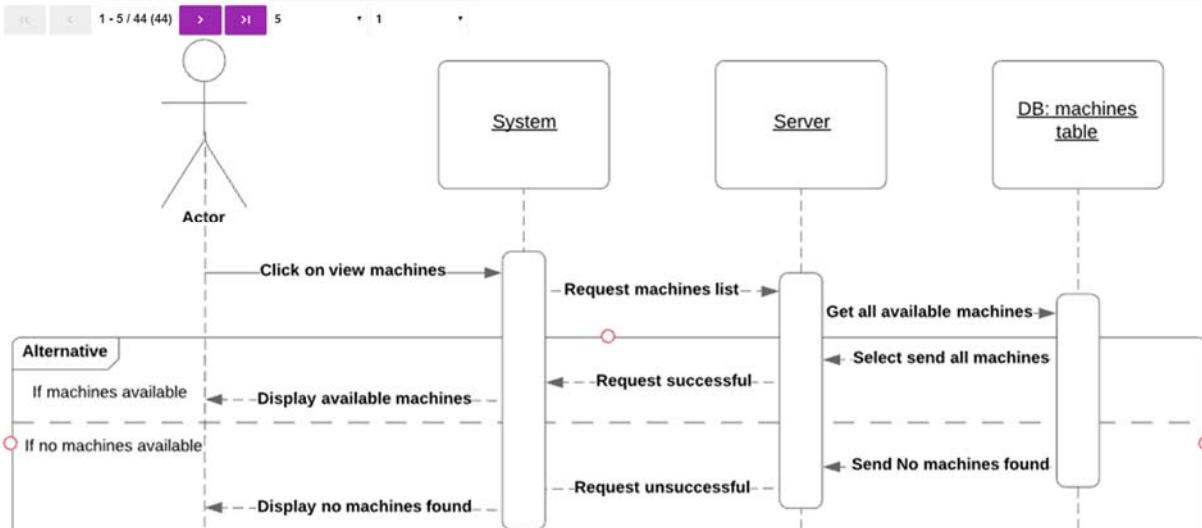
## Preview Image included in use case UDA006



## Use Case: View Machines

Use-Case Name:	View Machines	
Use-Case ID:	UDA014	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Admin</li> </ul>	
Description:	Admins can look through all the machines and equipment available at the gym, they can view images, names, QRCodes, locations, etc...	
Precondition:	The actor has to be logged in as an admin.	
Trigger	Click on the “Machines “ button on the admin panel.	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: Click on the “View machines” button in the admin panel.</li> <li>● Step 3: Admin can scroll between all available machines, view detailed profile and log information.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system will show all machines opened new windows.</li> </ul>
Alternate Courses:	There are no Alt-Steps for this use case.	
Conclusion:	This use case concludes when all available machines are shown along with their information.	
Post condition:	The admin was able to view all the available machines and equipment.	

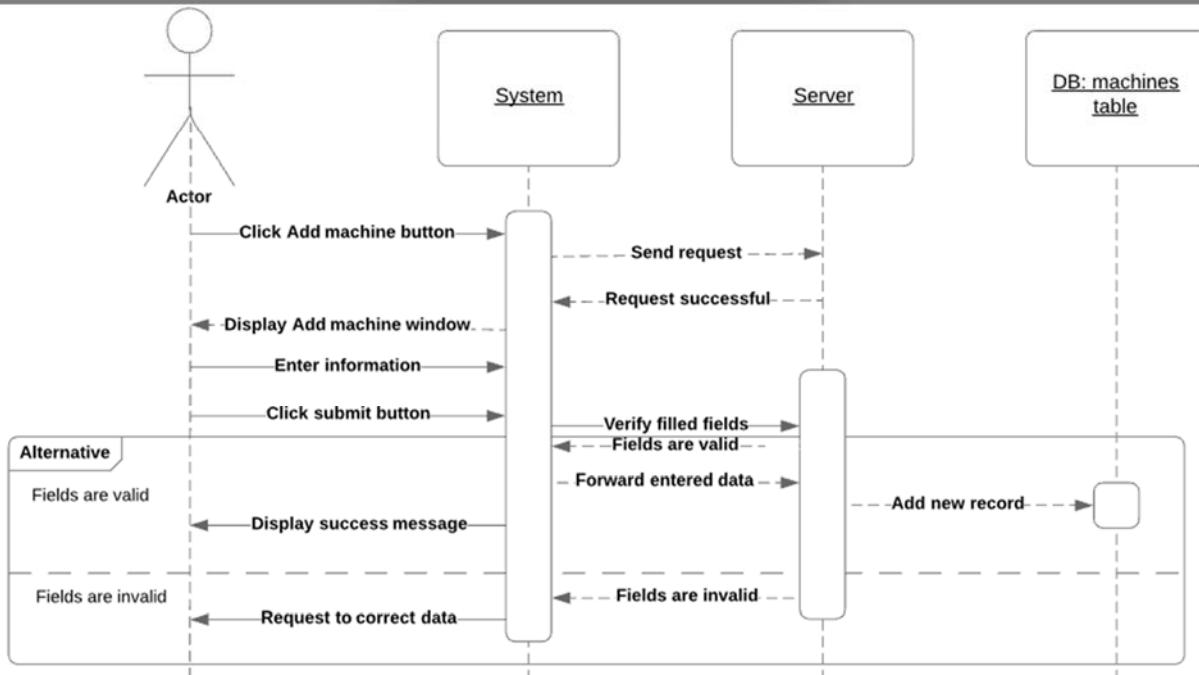
Gym Management System								
MEMBERS	COACHES	BARTENDERS	WORKOUTS	MOVES	MACHINES	MENU	PT ADS	MESSAGES
ID	Name			QR Code		Image	Floor	Section
43	Ab Crunch Machine			QRAbCrunch		1	A	
46	Bioep Curl Machine			QRBicepCurl		1	A	
8	Barbell			QRBarbell		1	B	
14	Dumbbells			QRDumbbells		1	B	
13	Dip Bars			QRDipBars		1	B	



## Use Case: Add Machines

Use-Case Name:	Add Machines	
Use-Case ID:	UDA015	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Admin</li> </ul>	
Description:	<p>Admin users can add new machines to the database through a form in the admin panel, the admin enters all the needed information about the machine and then triggers the form.</p>	
Precondition:	<p>The actor has to be logged in as an admin.</p>	
Trigger	<p>Click on the “New machine” button.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: Click on the “New machine” button in the admin panel.</li> <li>● Step 3: Fill the required fields in the forms.</li> <li>● Step 4: Clicks on the “Add machine” button.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system opens a new window with a form of the required information to be filled.</li> <li>● Step 5: System saves new records in the database.</li> <li>● Step 6: System shows a message “Machine added successfully”.</li> <li>● Step 7: System adds the new machine to the machines list so it can be seen by the admin UDA014</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● Alt-Step 5: If entered data is incorrect, system requests admin to reenter.</li> <li>● Alt-Step 6: If the addition of the member was unsuccessful, the system will return and display an error.</li> </ul>	
Conclusion:	<p>This use case concludes when the admin has successfully registered a machine.</p>	
Post condition:	<p>The admin was able to add a new machine to the database.</p>	

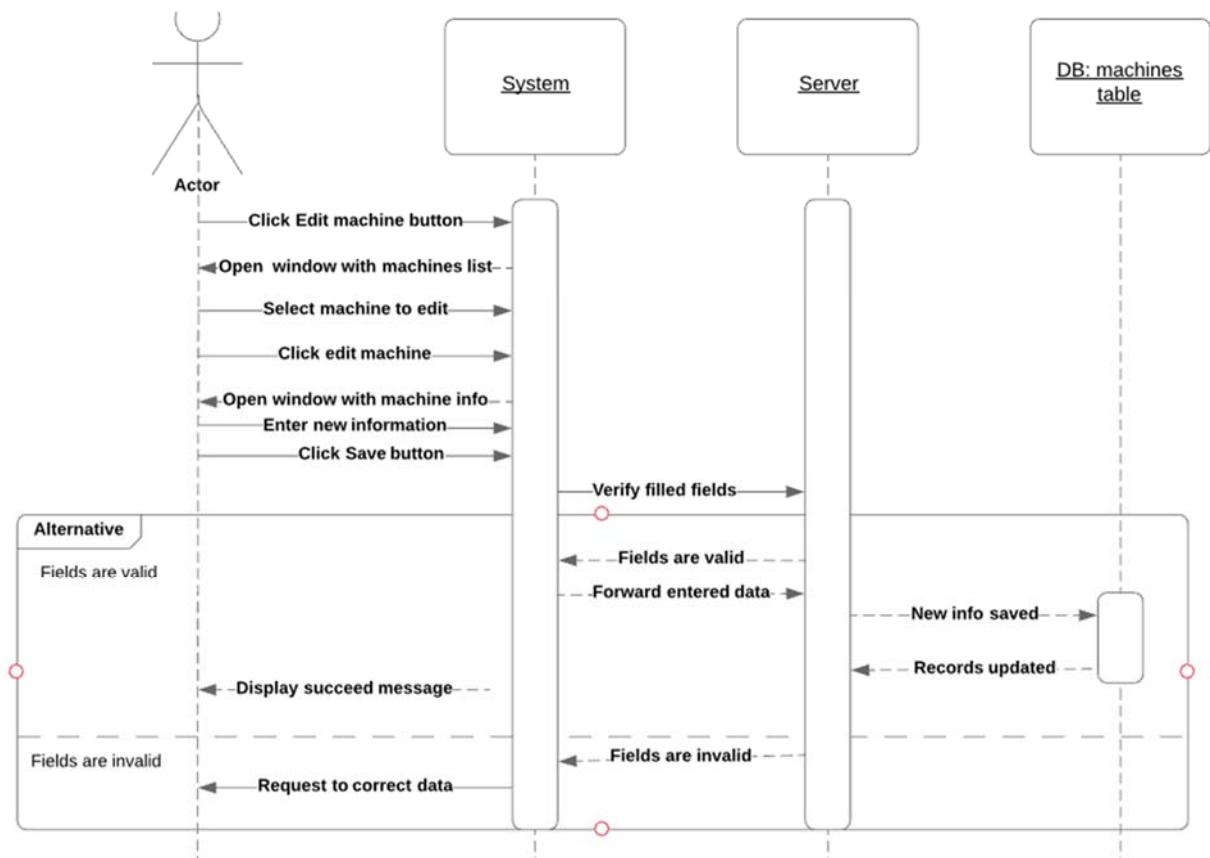
ID	Name	QR Code	Image	Floor	Section	Actions
43	Ab Crunch Machine	<input type="text" value="Machine Name"/>		1	A	
46	Bicep Curl Machine	<input type="text" value="Machine QR Code"/>		1	A	
8	Barbell	<input type="text" value="Floor"/>		1	B	
14	Dumbbells	<input type="text" value="Section"/>		1	B	
13	Dip Bars			1	B	
ID Name		<b>INSERT ✓</b>				
		1 - 5 / 44 (44) < > >> 5 + 1				



## Use Case: Edit Machines

Use-Case Name:	Edit machine	
Use-Case ID:	UDA016	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Admin</li> </ul>	
Description:	Admin users can edit machines' details and information in the database, including their names, allocated moves, locations, etc..	
Precondition:	The actor has to be logged in as an admin.	
Trigger	Click on the “Edit machine” Button.	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: Click on the “Edit” button beside a machine.</li> <li>● Step 3: Admin edits the information to be changed</li> <li>● Step 4: Click the “Save” button.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: System opens a new window with a form filled with all machine’s information.</li> <li>● Step 5: System verifies and saves the changes, then shows an “Update Successful” message.</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● Alt-Step 5: The update was unsuccessful; the system displays a message accordingly.</li> </ul>	
Conclusion:	This use case concludes when the machine record is successfully updated.	
Post condition:	The admin was able to edit the record and update the database.	

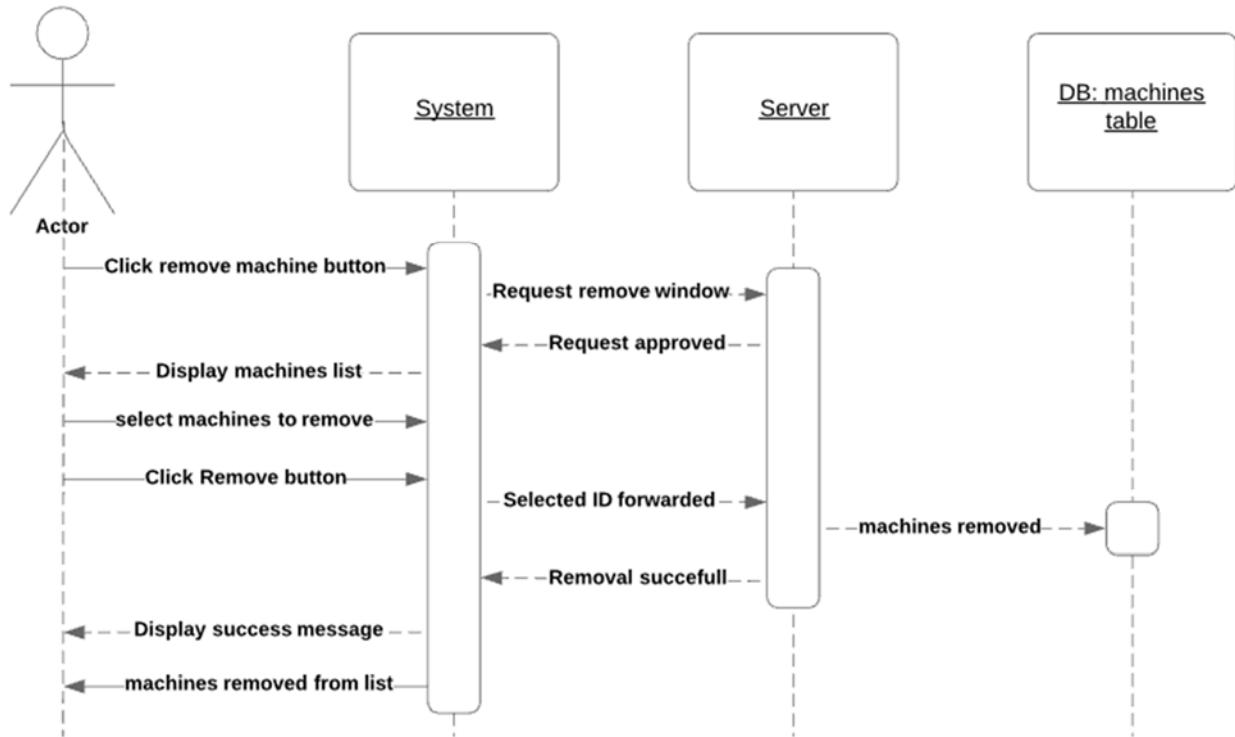
## Preview Image included in use case UDA015



## Use Case: Remove Machines

Use-Case Name:	Remove Machine	
Use-Case ID:	UDA017	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Admin</li> </ul>	
Description:	<p>The admin has the ability to remove machines from the database (this should be done in rare cases).</p>	
Precondition:	<p>The actor has to be logged in as an admin.</p>	
Trigger	<p>Click on the “Remove” button beside a machine.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: Click On “Remove” button in the admin panel.</li> <li>● Step 3: The actor confirms</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system will open a confirmation dialogue.</li> <li>● Step 4: The system removes the selected record from the database.</li> <li>● Step 5 : System shows the message “coach removed successfully”.</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● Alt-Step 3: The actor cancels the process and doesn't confirm; the action doesn't proceed.</li> </ul>	
Conclusion:	<p>This use case concludes when the record is successfully removed from the database.</p>	
Post condition:	<p>The admin removed the specific record and the database is updated.</p>	

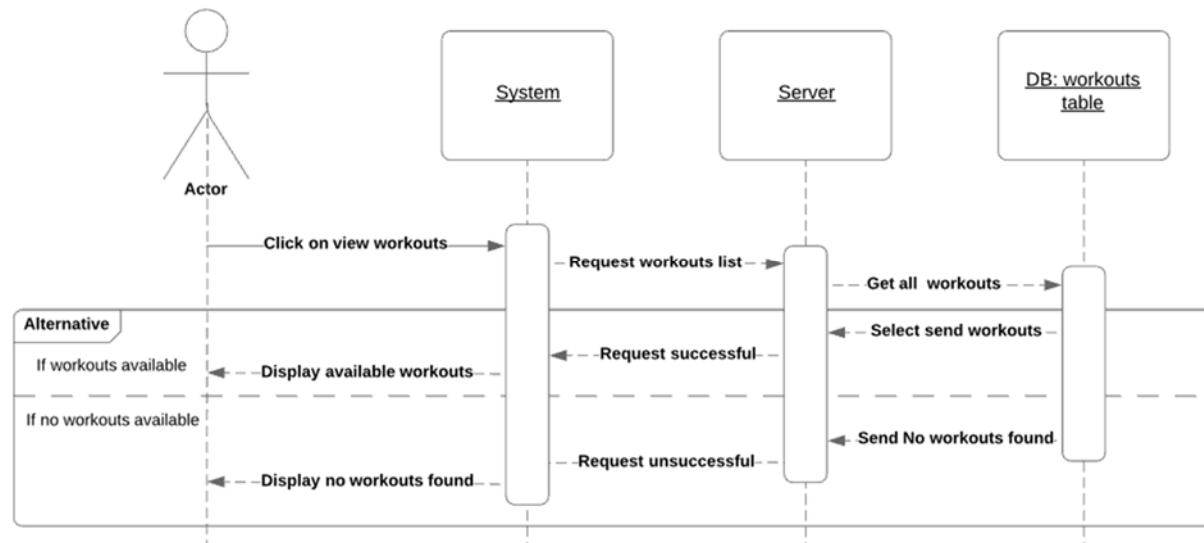
## Preview Image included in use case UDA015



## Use Case: View Workouts

Use-Case Name:	View Workouts	
Use-Case ID:	UDA018	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Admin</li> </ul>	
Description:	The admin user can view all the workout plans available at the gym.	
Precondition:	The actor has to be logged in as an admin.	
Trigger	Click on the "Workouts" button in the admin panel.	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: Click on "View workouts" button in the admin panel.</li> <li>● Step 3: Admin can scroll between all available workouts and view detailed information.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system will show all workouts' information opened in a new window.</li> </ul>
Alternate Courses:	There are no Alt-Steps for this use case.	
Conclusion:	This use case concludes when all available workouts are shown along with their information.	
Post condition:	The admin was able to view all the available workouts along with their information.	

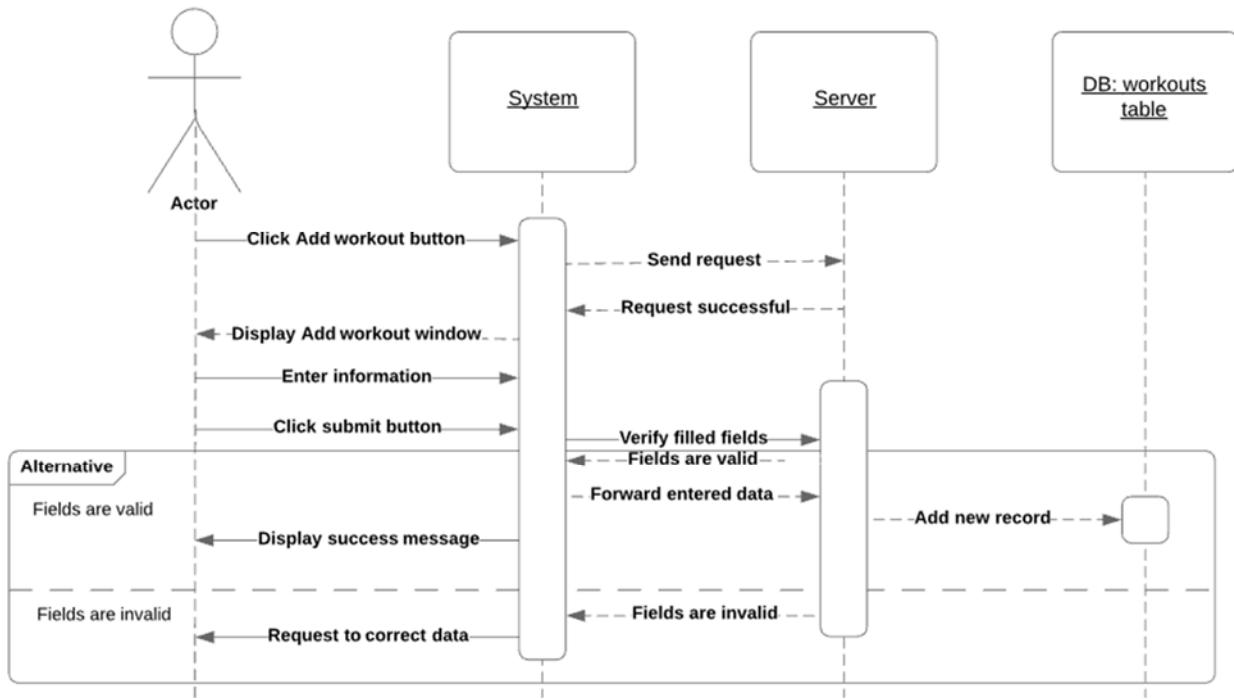
CURRENT WORKOUTS		PREVIOUS WORKOUTS		ASSIGNED WORKOUTS	
ID	Attached Workout	ID	Attached Workout	Member ID	Actions
30	90	149			
31	59	151			
ID	Attached Workout	ID	Attached Workout	Member ID	Actions
5	1 - 2 / 2 (2)	5	1		
ID	Member ID	Attached Workout		Time Finished	Actions
29	11	21		2020-04-21 00:58:43	
31	59	127		2020-05-07 16:50:43	
32	90	142		2020-06-05 05:20:31	
33	90	143		2020-06-06 08:39:36	
34	90	144		2020-06-06 08:47:04	
35	90	147		2020-06-11 19:33:33	
36	59	129		2020-06-14 15:31:53	
ID	Member ID	Attached Workout		Time Finished	Actions
1	7 / 7 (7)	40	1		



## Use Case: Add Workouts

Use-Case Name:	Add Workouts	
Use-Case ID:	UDA019	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Admin</li> </ul>	
Description:	<p>Admin users can add a new workout to the database through a form in the admin panel, the admin enters all the needed information about the workout and their included moves and then triggers the form.</p>	
Precondition:	The actor has to be logged in as an admin.	
Trigger	Click on the “New workout” button.	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: Click on the “New workout” button in the admin panel.</li> <li>● Step 3: Fill the required fields in the forms.</li> <li>● Step 4: Clicks on the “Add workout” button.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system opens a new window with a form of the required information to be filled.</li> <li>● Step 5: System saves new records in the database.</li> <li>● Step 6: System shows a message “Workout added successfully”.</li> <li>● Step 7: System adds the new workout to the workouts list so it can be seen by the admin UDA018</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● Alt-Step 5: If entered data is incorrect, system requests admin to reenter.</li> <li>● Alt-Step 6: If the addition of the member was unsuccessful, the system will return and display an error.</li> </ul>	
Conclusion:	This use case concludes when the actor has successfully added a new workout.	
Post condition:	The admin was able to add a new machine to the database.	

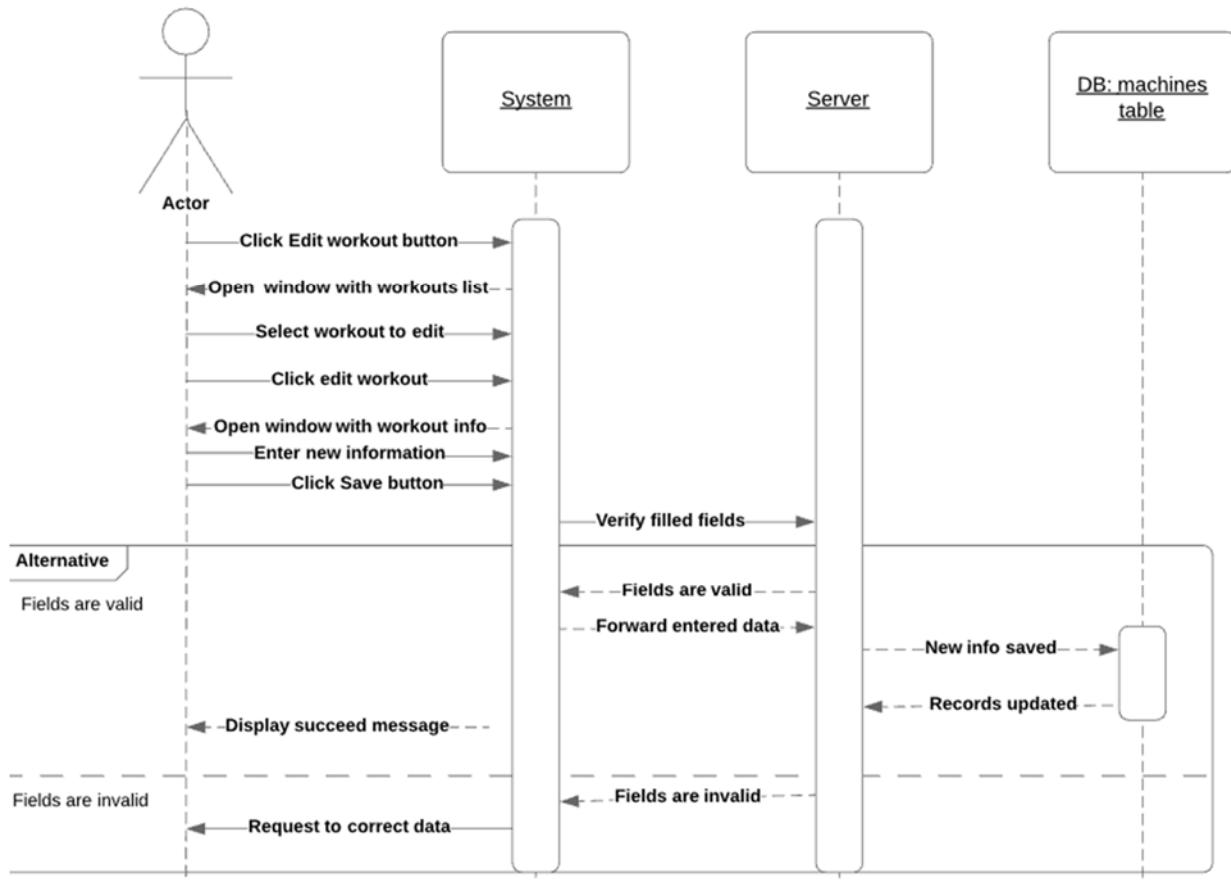
## Preview Image included in use case UDA018



## Use Case: Edit Workouts

Use-Case Name:	Edit Workouts	
Use-Case ID:	UMM020	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Admin</li> </ul>	
Description:	Admin users can edit workouts' details and information in the database, including their names, included moves, types, etc..	
Precondition:	The actor has to be logged in as an admin.	
Trigger	Click on the “Edit workout” Button.	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: Click on the “Edit workout” button in the admin panel.</li> <li>● Step 3: Admin edits the information to be changed</li> <li>● Step 4: Click “Save” button</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: System opens a new window with a form filled with all workout information.</li> <li>● Step 5: System saves all changes to the records in the database and show “Update Successful”</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● Alt-Step 5: The update was unsuccessful; the system displays a message accordingly.</li> </ul>	
Conclusion:	This use case concludes when the admin has successfully updated the workout.	
Post condition:	The admin was able to edit the record and update the database.	

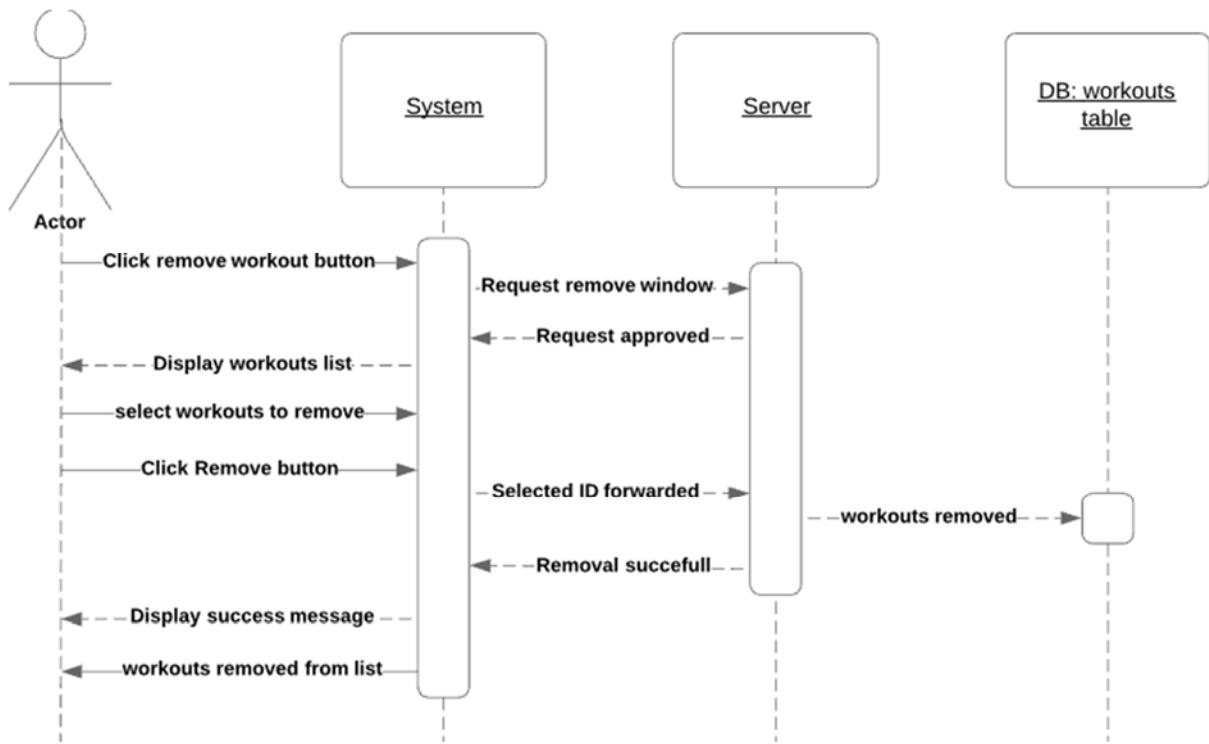
## Preview Image included in use case UDA018



## Use Case: Remove Workouts

Use-Case Name:	Remove Workouts	
Use-Case ID:	UDA021	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Admin</li> </ul>	
Description:	<p>The admin has the ability to remove workouts from the database (this should be done in rare cases).</p>	
Precondition:	<p>The actor has to be logged in as an admin.</p>	
Trigger	<p>Click on “Remove workout” button</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: Click On “Remove” button in the admin panel.</li> <li>● Step 3: The actor confirms</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system will open a confirmation dialogue.</li> <li>● Step 4: The system removes the selected record from the database.</li> <li>● Step 5 : System shows the message “workout removed successfully”.</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● Alt-Step 3: The actor cancels the process and doesn't confirm; the action doesn't proceed.</li> </ul>	
Conclusion:	<p>This use case concludes when the admin has successfully removed the workout.</p>	
Post condition:	<p>The admin removed the specific record and the database is updated.</p>	

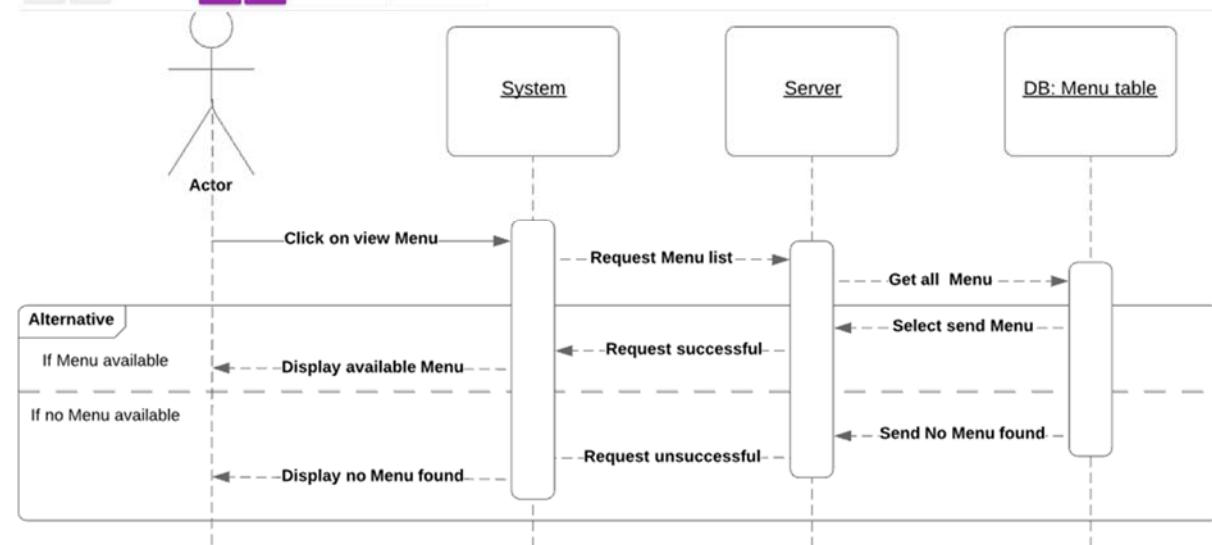
## Preview Image included in use case UDA018



## Use Case: View Menu

Use-Case Name:	View Menu	
Use-Case ID:	UDA022	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Admin</li> </ul>	
Description:	Admins can look through all the items in the menu at the gym, they can view images, names, nutrition information, availability, etc...	
Precondition:	The actor has to be logged in as an admin.	
Trigger	Click on the “Menu” button on the admin panel.	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: Click on the “Menu” button in the admin panel.</li> <li>● Step 3: Admin can preview the menu.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system will show the available menu offered</li> </ul>
Alternate Courses:	There are no Alt-Steps for this use case.	
Conclusion:	This use case concludes when the available menu is previewed.	
Post condition:	The admin was able to view all the available menu items along with their information.	

Dashboard Overview											
Members		Coaches		Bartenders		Workouts		Moves		Machines	
ID	Name	Image	Description	Calories	Fats	Protein	Carbohydrates	Sugar	Price	Available	Actions
1	Meal 11		Item 11	350	14	55	11	3	7.00	1	<span style="color:red;">Delete</span>
2	Snackbar 1		Item 12	250	4	9	25	0	7.00	1	<span style="color:red;">Delete</span>
3	item3Name		Item 13	174	8	12	22	4	5.40	1	<span style="color:red;">Delete</span>
7	item4Name		Item 14	444	4	44	4	4	4.40	2	<span style="color:red;">Delete</span>
8	item5Name		Item 15	555	5	55	5	5	5.50	5	<span style="color:red;">Delete</span>



## Use Case: Add Items

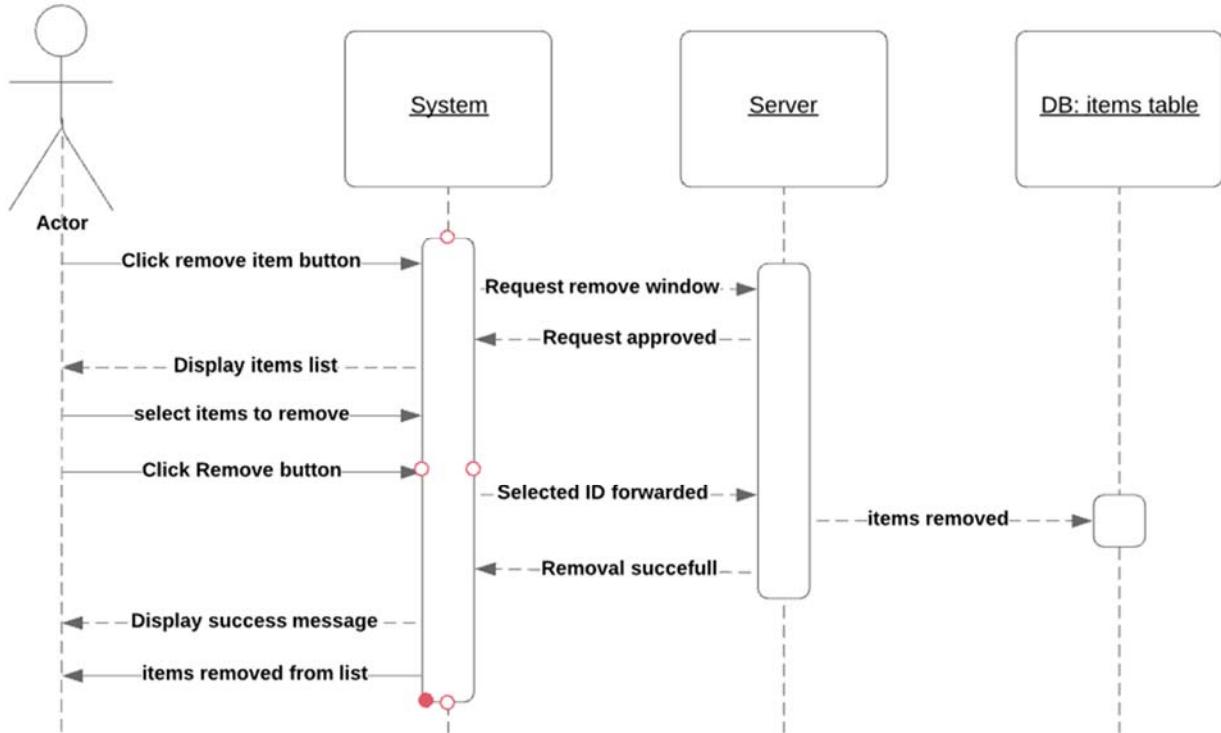
Use-Case Name:	Add Items	
Use-Case ID:	UDA023	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Admin</li> </ul>	
Description:	<p>Admin users can add new machines to the database through a form in the admin panel, the admin enters all the needed information about the machine and then triggers the form.</p>	
Precondition:	<p>The actor has to be logged in as an admin.</p>	
Trigger	<p>Click on the “New item” button.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: Click on the “New Item” button in the admin panel.</li> <li>● Step 3: Fill the required fields in the forms.</li> <li>● Step 4: Clicks on the “Add Item” button.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system opens a new window with a form of the required information to be filled.</li> <li>● Step 5: System saves new records in the database.</li> <li>● Step 6: System shows a message “Item added successfully”.</li> <li>● Step 7: System adds the new workout to the workouts list so it can be seen by the admin UDA018</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● Alt-Step 5: If entered data is incorrect, system requests admin to reenter.</li> <li>● Alt-Step 6: If the addition of the member was unsuccessful, the system will return and display an error.</li> </ul>	
Conclusion:	<p>This use case concludes when the actor has successfully added a new item.</p>	
Post condition:	<p>The admin was able to add a new menu item to the database.</p>	



## Use Case: Remove Items

Use-Case Name:	Remove item	
Use-Case ID:	UDA024	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>• Admin</li> </ul>	
Description:	The admin has the ability to remove items from the menu(this should be done in rare cases).	
Precondition:	The actor has to be logged in as an admin.	
Trigger	Click on “Remove Item” button	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>• Step 1: Click On “Remove” button in the admin panel.</li> <li>• Step 3: The actor confirms</li> </ul>	<ul style="list-style-type: none"> <li>• Step 2: The system will open a confirmation dialogue.</li> <li>• Step 4: The system removes the selected record from the database.</li> <li>• Step 5 : System shows the message “item removed successfully”.</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>• Alt-Step 3: The actor cancels the process and doesn't confirm; the action doesn't proceed.</li> </ul>	
Conclusion:	This use case concludes when the admin has successfully removed an item.	
Post condition:	The admin was able to remove the item from the system.	

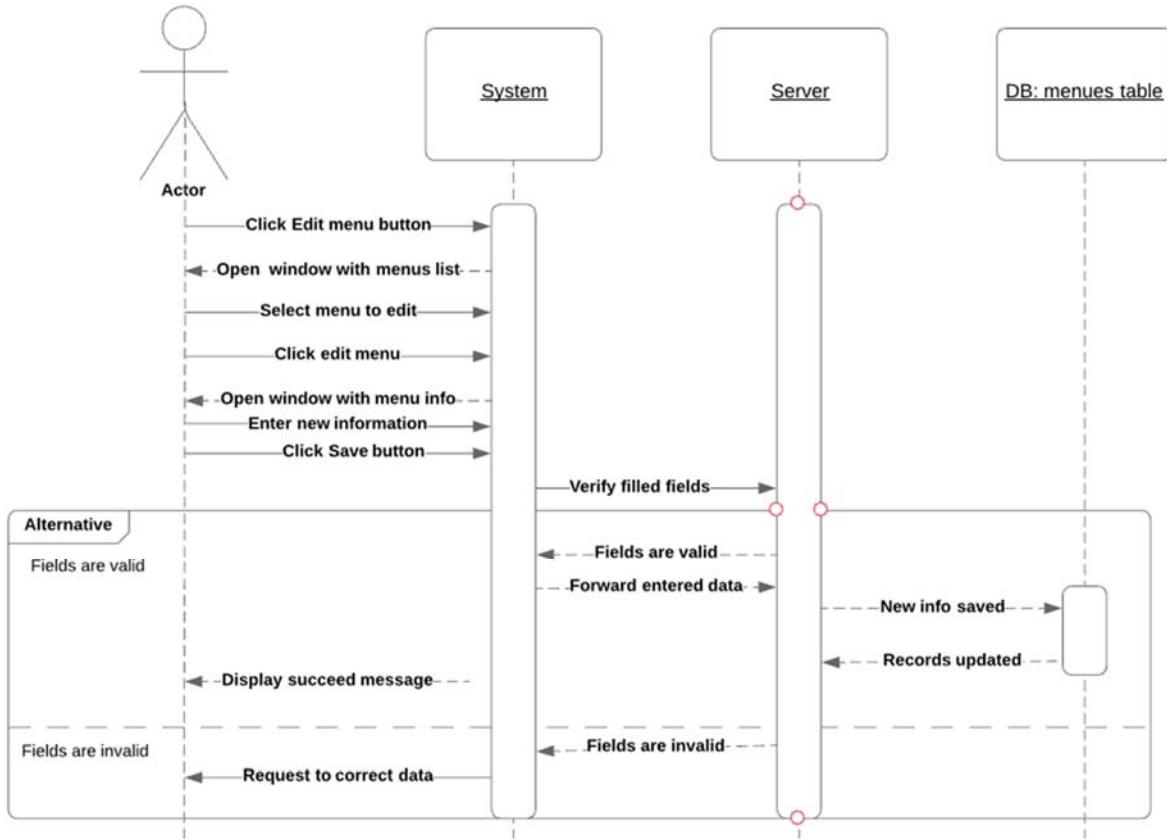
## Preview Image included in use case UDA022



## Use Case: Edit Menu

Use-Case Name:	Edit Menu	
Use-Case ID:	UDA025	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Admin</li> <li>● Bartender</li> </ul>	
Description:	Admin users can edit menu items' details and information in the database, including their names, prices, nutritional facts, etc..	
Precondition:	The actor has to be logged in as an admin or bartender.	
Trigger	Click on the "Edit" Button.	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: Click on the "Edit" button beside an item.</li> <li>● Step 3: Admin edits the information to be changed</li> <li>● Step 4: Click the "Save" button.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: System opens a new window with a form filled with all coaches' information.</li> <li>● Step 5: System verifies and saves the changes, then shows an "Update Successful" message.</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● Alt-Step 8: The update was unsuccessful; the system displays a message accordingly.</li> </ul>	
Conclusion:	This use case concludes when the admin has successfully updated the menu.	
Post condition:	The admin was able to edit the record and update the database.	

## Preview Image included in use case UDA022



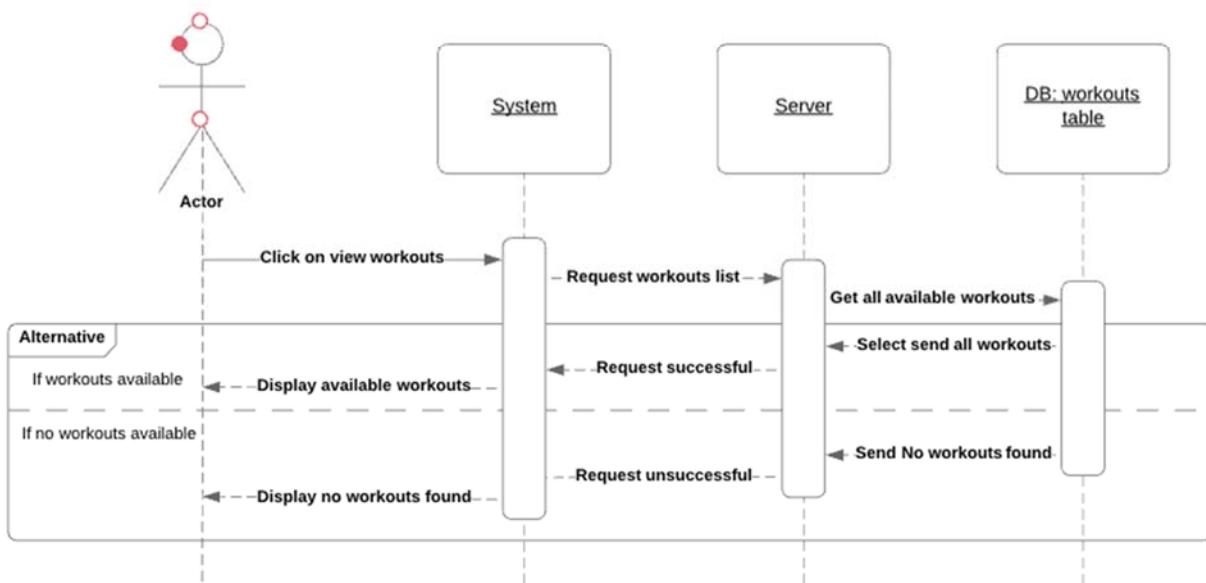
## Use Case: View Bartenders

Use-Case Name:	View bartenders	
Use-Case ID:	UDA010	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Admin</li> </ul>	
Description:	<p>Admin users can look through bartenders' details and view their activity, such as the orders they have served.</p>	
Precondition:	<p>The actor has to be logged in as an admin.</p>	
Trigger	<p>Click on the “View bartenders” button on the admin panel.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: Click on “View bartenders” button in the admin panel.</li> <li>● Step 3: Admin can scroll between all bartenders records and view detailed profile and log information.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system will show all bartenders' records and logs in a table opened in a new window.</li> </ul>
Alternate Courses:	<p>There are no Alt-Step for this use case.</p>	
Conclusion:	<p>This use case concludes when all available bartenders are listed along with their records.</p>	
Post condition:	<p>The admin was able to view all the available bartenders along with their information.</p>	

File Insert Reload Dev Tools

MEMBERS	COACHES	BARTENDERS	WORKOUTS	MOVES	MACHINES	MENU	PT ADS	MESSAGES		
ID	Email		First	Last	Password	Phone	Address	Image	Salary	Actions
3	bartender@bartender.bartender		barfName	EditedLast	bartender1234	123456789			2430	
5	Bart@Email.com		Bart1	Bart	Bart1234	(961) 1111111			3330	
7	Role@Role.Role		Role	Role	Role1234	(961) 3223232			2202	
8	Bart@Bart.Bart		Bart	Bart	Bart1234	(961) 111111			202020	
9	Work@lam.Sure		This	Will	OfThis1234	(961) 11111			1111	

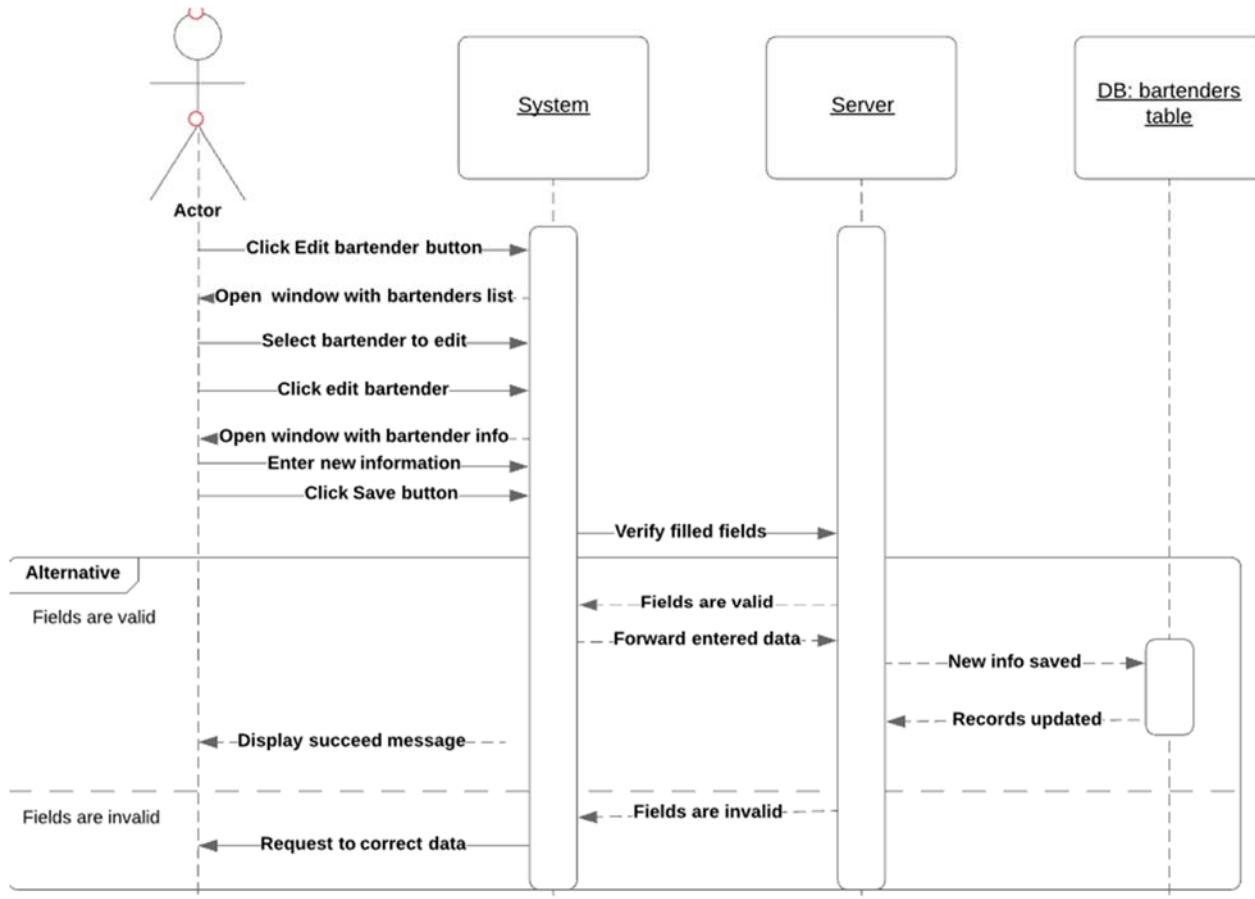
1 - 5 / 6 (6) > >> 5 ▼ 1 ▲



## Use Case: Edit Bartenders

Use-Case Name:	Edit Bartender	
Use-Case ID:	UDA012	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Admin</li> </ul>	
Description:	<p>Admin users can edit bartenders' details and information in the database, including their names, addresses, phone numbers, etc..</p>	
Precondition:	<p>The actor has to be logged in as an admin.</p>	
Trigger	<p>Click on the “Edit Bartender” Button.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: Click on the “Edit Bartender” button in the admin panel.</li> <li>● Step 3: Admin edits the information to be changed</li> <li>● Step 4: Click “Save” button</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: System opens a new window with a form filled with all workout information.</li> <li>● Step 5: System saves all changes to the records in the database and show “Update Successful”</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● Alt-Step 5: The update was unsuccessful; the system displays a message accordingly.</li> </ul>	
Conclusion:	<p>This use case concludes when the admin has successfully updated the bartender's information.</p>	
Post condition:	<p>The admin was able to edit the record and update the database.</p>	

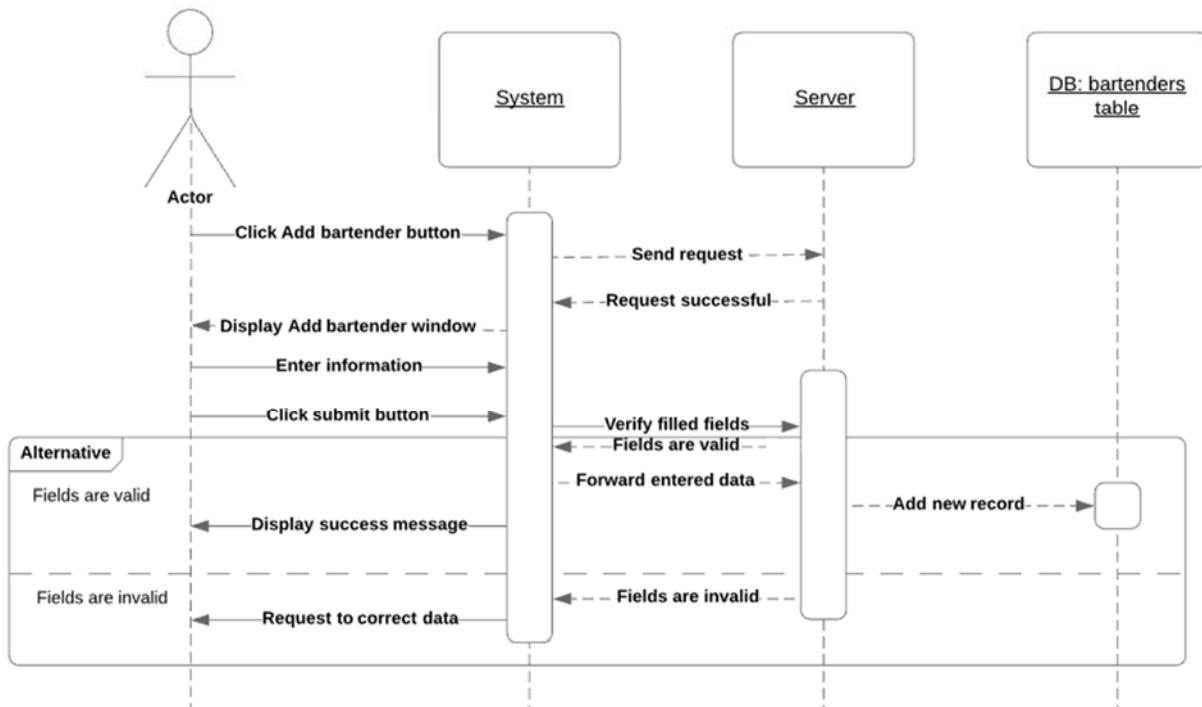
## Preview Image included in use case UDA010



## Use Case: Add Bartender

Use-Case Name:	Add Bartender	
Use-Case ID:	UDA011	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Admin</li> </ul>	
Description:	<p>Admin users can add new bartenders to the database through a form in the admin panel, the admin enters all the needed information about the coach and then triggers the form.</p>	
Precondition:	<p>The actor has to be logged in as an admin.</p>	
Trigger	<p>Click on the “Add coach” button.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: Click on the “Add Bartender” button in the admin panel.</li> <li>● Step 3: Fill the required fields in the forms.</li> <li>● Step 4: Clicks on the “Add” button.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system opens a new window with a form of the required information to be filled.</li> <li>● Step 5: System saves new records in the database.</li> <li>● Step 6: System shows a message “Bartender added successfully”.</li> <li>● Step 7: System adds the new coach to the members list so it can be seen by the admin UDA006</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● Alt-Step 5: If entered data is incorrect; the system doesn't save the record</li> <li>● Alt-Step 6: The addition was unsuccessful; a message is displayed accordingly.</li> </ul>	
Conclusion:	<p>The use case concludes when the bartender is successfully added.</p>	
Post condition:	<p>The admin was able to add a new bartender and update the database.</p>	

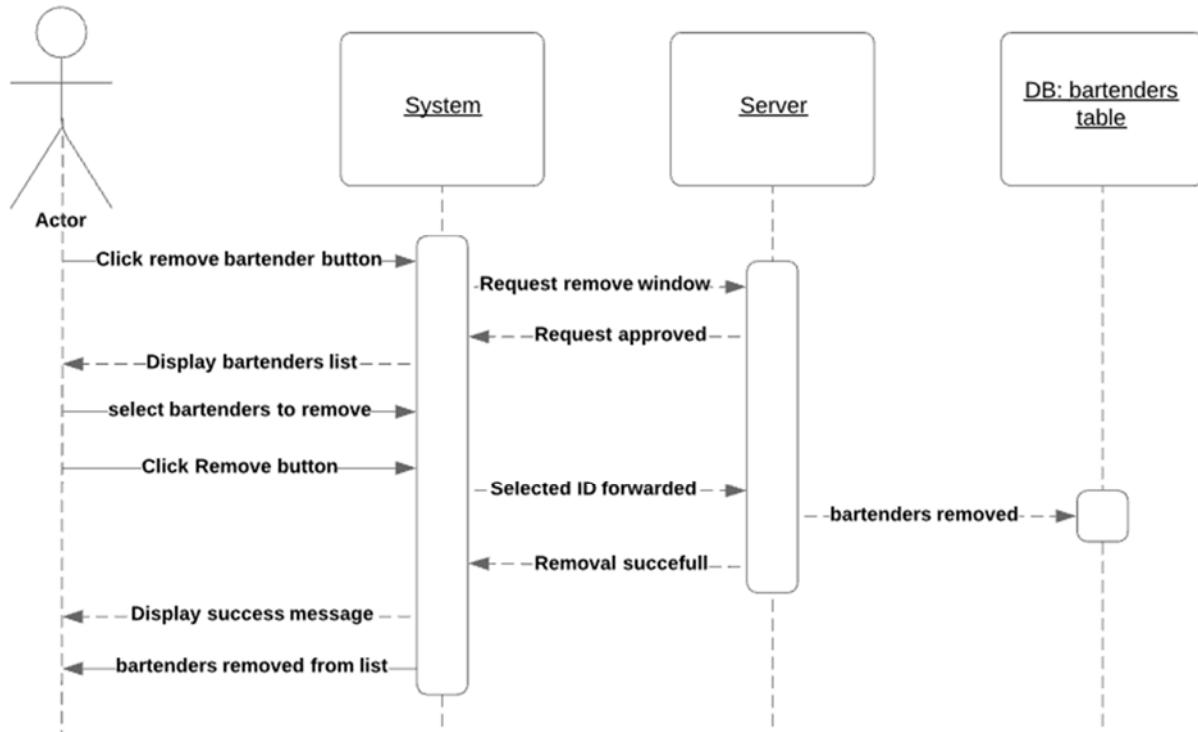
First	Last	Password	Phone	Address	Image	Salary
barfName		First Name	Last Name			
Bart1						
Role		Email				
Bart						
This		Password				
First						
5	1	LB (961) ▾	Telephone			
		Salary				
		Address				



## Use Case: Remove Bartender

Use-Case Name:	Remove Bartender	
Use-Case ID:	UDA013	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Admin</li> </ul>	
Description:	<p>The admin has the ability to remove bartender records from the database (this should be done in rare cases).</p>	
Precondition:	<p>The actor has to be logged in as an admin.</p>	
Trigger	<p>Click on the “Remove” button beside a bartender.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: Click On “Remove” button in the admin panel.</li> <li>● Step 3: The actor confirms</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system will open a confirmation dialogue.</li> <li>● Step 4: The system removes the selected record from the database.</li> <li>● Step 5 : System shows the message “bartender removed successfully”.</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● Alt-Step 3: The actor cancels the process and doesn't confirm; the action doesn't proceed.</li> </ul>	
Conclusion:	<p>This use case concludes when the record is successfully removed from the database.</p>	
Post condition:	<p>The admin removed the specific record and the database is updated.</p>	

## Preview Image included in use case UDA022



## Use Case: View Reports

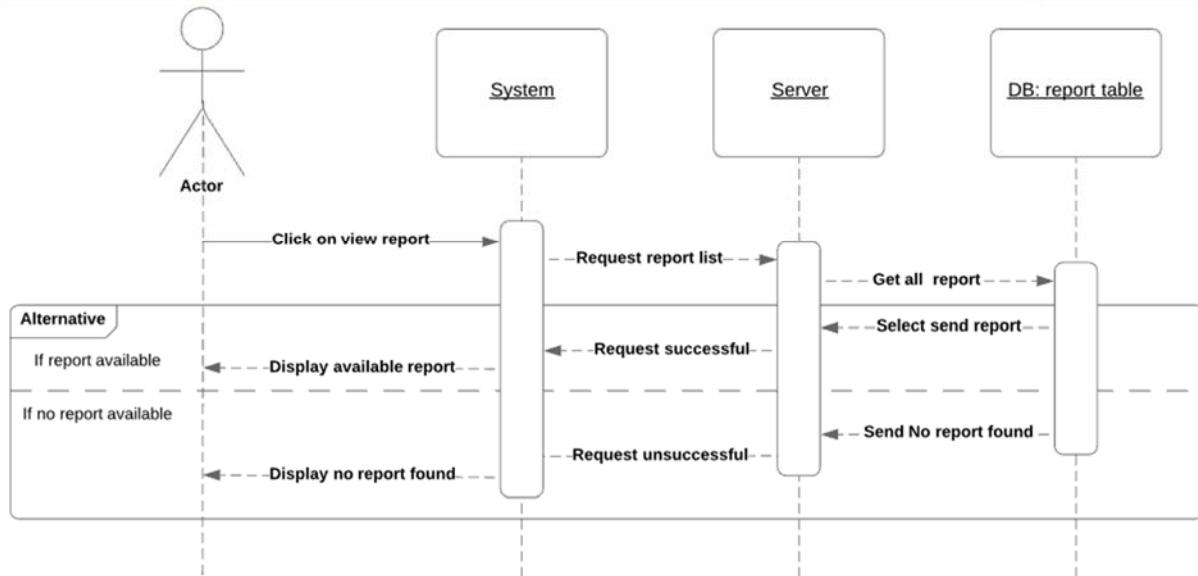
Use-Case Name:	View Reports	
Use-Case ID:	UDA027	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Admin</li> </ul>	
Description:	<p>Admin users can preview the reports, feedback, and contact messages sent by the members and coaches.</p>	
Precondition:	<p>The actor has to be logged in as an admin.</p>	
Trigger	<p>Click on the "View Reports" button on the admin panel.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: Click on the "View Reports" button in the admin panel.</li> <li>● Step 3: Admin can preview the Reports.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system will show the available Reports</li> </ul>
Alternate Courses:	<p>There are no Alt-Step for this use case.</p>	
Conclusion:	<p>This use case concludes when the available reports are previewed.</p>	
Post condition:	<p>The admin was able to view all the available reports along with their information.</p>	

File Insert Reload Dev Tools

MEMBERS	COACHES	BARTENDERS	WORKOUTS	MOVES	MACHINES	MENU	PT ADS	MESSAGES
ID	Message	From		Type	Time			Actions
30	✉	Jounaid@Coach.com		Coach	2020-06-04 04:24:14			trash
31	✉	Test@Test.Test		Guest	2020-06-04 04:37:42			trash
32	✉	Jounaid@Coach.com		Coach	2020-06-04 05:15:18			trash
33	✉	Jounaid@Coach.com		Coach	2020-06-04 05:15:22			trash
34	✉	Jounaid@Coach.com		Coach	2020-06-04 05:16:50			trash
ID	Message	From		Type	Time			Actions
30	✉	Jounaid@Coach.com		Coach	2020-06-04 04:24:14			trash
31	✉	Test@Test.Test		Guest	2020-06-04 04:37:42			trash
32	✉	Jounaid@Coach.com		Coach	2020-06-04 05:15:18			trash
33	✉	Jounaid@Coach.com		Coach	2020-06-04 05:15:22			trash
34	✉	Jounaid@Coach.com		Coach	2020-06-04 05:16:50			trash
35	✉	Jounaid@Coach.com		Coach	2020-06-04 05:18:04			trash
36	✉	BSMKDD@outlook.com		Member	2020-06-05 03:46:07			trash
37	✉	BSMKDD@outlook.com		Member	2020-06-14 08:25:29			trash

1 - 5 / 7 (7) ⏪ ⏹ 5 ⏵ 1 ⏶

30	✉	Jounaid@Coach.com	Coach	2020-06-04 04:24:14	trash
31	✉	Test@Test.Test	Guest	2020-06-04 04:37:42	trash
32	✉	Jounaid@Coach.com	Coach	2020-06-04 05:15:18	trash
33	✉	Jounaid@Coach.com	Coach	2020-06-04 05:15:22	trash
34	✉	Jounaid@Coach.com	Coach	2020-06-04 05:16:50	trash
35	✉	Jounaid@Coach.com	Coach	2020-06-04 05:18:04	trash
36	✉	BSMKDD@outlook.com	Member	2020-06-05 03:46:07	trash
37	✉	BSMKDD@outlook.com	Member	2020-06-14 08:25:29	trash



## Use Case: View Orders

Use-Case Name:	View Orders	
Use-Case ID:	UDB002	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Bartender</li> </ul>	
Description:	Bartender users can look through the orders and view their details, such as price, and time of serving.	
Precondition:	The actor has to be logged in as a Bartender.	
Trigger	Click on “Orders” button on the bartender panel	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: Click on “Orders” button in the admin panel</li> <li>● Step 3: Admin can preview the orders.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system shows the active orders at the bar.</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● Alt-Step 2: if no orders available the system will show a “No orders at this time” message</li> </ul>	
Conclusion:	This use case is concluded when orders are shown.	
Post condition:	The bartender was able to view all the current orders.	

File Insert Dev Tools

ORDERS MENU

By: Bassam#100

11 minutes

Item	Quantity	Price
Meal 1	1	7.0\$
item1Name	1	4.4\$
item7Name	1	4.4\$
item4Name	1	4.4\$
Snackbar 1	2	14.0\$

**20.2\$**

REJECT X CONFIRM ✓

By: Bassam#133

10 minutes

Item	Quantity	Price
item5Name	1	5.5\$
item16Name	1	4.4\$
item7Name	1	4.4\$
item4Name	1	4.4\$
Snackbar 1	2	14.0\$

**32.7\$**

REJECT X CONFIRM ✓

A new order has been placed...  
Snowload

By: Bassam#133

10 minutes

Item	Quantity	Price
item5Name	1	5.5\$
item16Name	1	4.4\$
item7Name	1	4.4\$
item4Name	1	4.4\$
Snackbar 1	2	14.0\$

**32.7\$**

REJECT X READY ✓

By: Bassam#133

10 minutes

Item	Quantity	Price
item5Name	1	5.5\$
item16Name	1	4.4\$
item7Name	1	4.4\$
item4Name	1	4.4\$
Snackbar 1	2	14.0\$

**32.7\$**

UNSERVED X SERVED ✓

Clicking the confirm button turns it to a "Ready button"

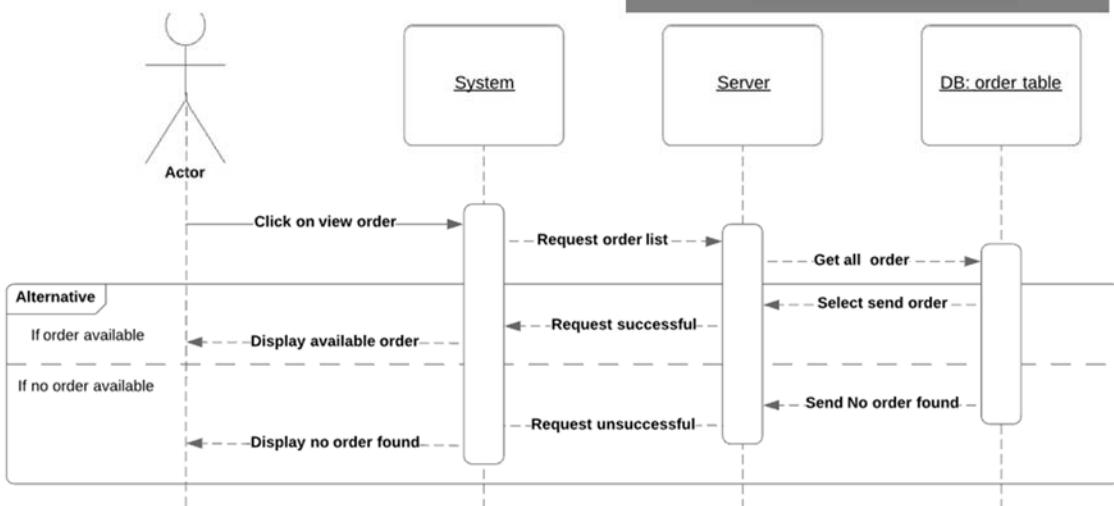
Clicking on the serve button saves the order and hides it from the live panel

Clicking the "Reject button shows a reason modal.

Type the reason for rejecting the order.

Reason:

REJECT X



## Use Case: Confirm Order

Use-Case Name:	Confirm Order	
Use-Case ID:	UDB003	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Bartender</li> </ul>	
Description:	Bartender users can confirm incoming orders to let users know that their order can be prepared.	
Precondition:	The actor has to be logged in as a bartender.	
Trigger	Click on the “Confirm Order” button on the bartender panel.	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: Click on “Confirm Order” button near the incoming order.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system registers the order as confirmed and sends a notification to the member.</li> <li>● Step 3: System moves order to “Being prepared “ list on the screen</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● Alt-Step 2: Order was not confirmed; notification sent to the member to let them know.</li> </ul>	
Conclusion:	This use case is concluded when orders are confirmed.	
Post condition:	The bartender was able to confirm and let the member know that their order is being prepared.	

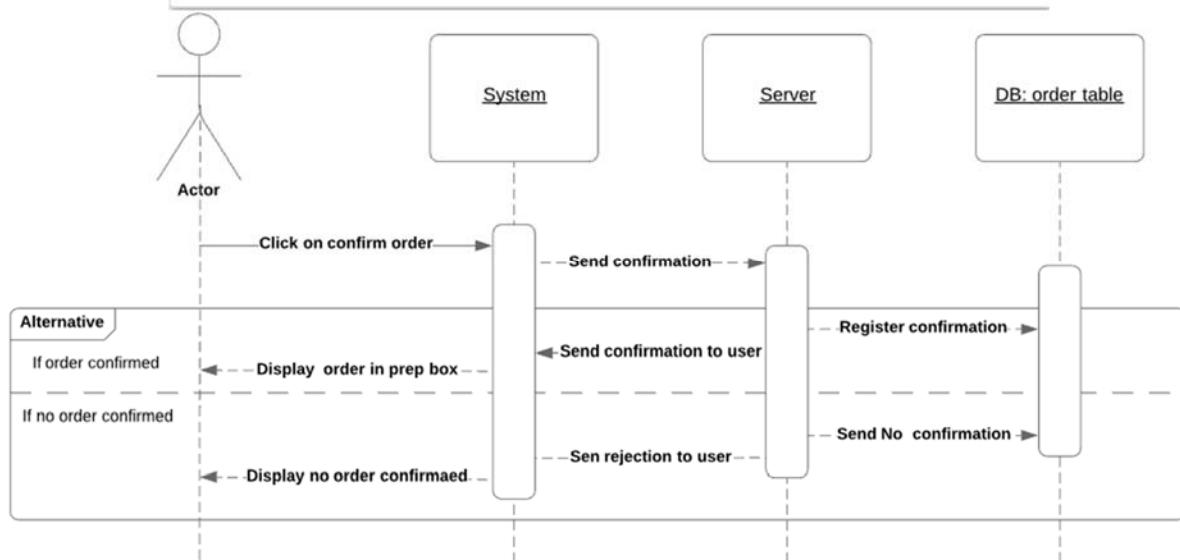
# By: Bassam#100

11 minutes

Item	Quantity	Price
Meal 1	1	7.0\$
item11Name	1	4.4\$
item7Name	1	4.4\$
item4Name	1	4.4\$
		<b>20.2\$</b>

REJECT ✕

CONFIRM ✓



## Use Case: Order Ready

Use-Case Name:	Order Ready	
Use-Case ID:	UDB004	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Bartender</li> </ul>	
Description:	Bartender user sends a ready notification to the member for him to pick it up.	
Precondition:	The actor has to be logged in as a bartender.	
Trigger	Click on “Order ready” Button near the order being prepared	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: Click on the “Order ready” button near the order being prepared.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system registers the order as ready and sends a notification to the member</li> <li>● Step 3: System removes order from being prepared list on the screen</li> <li>● Step 4: System registers order as done in the database</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● NA</li> </ul>	
Conclusion:	This use case is concluded when the notification is sent to the member.	
Post condition:	The order was added to the database as done.	

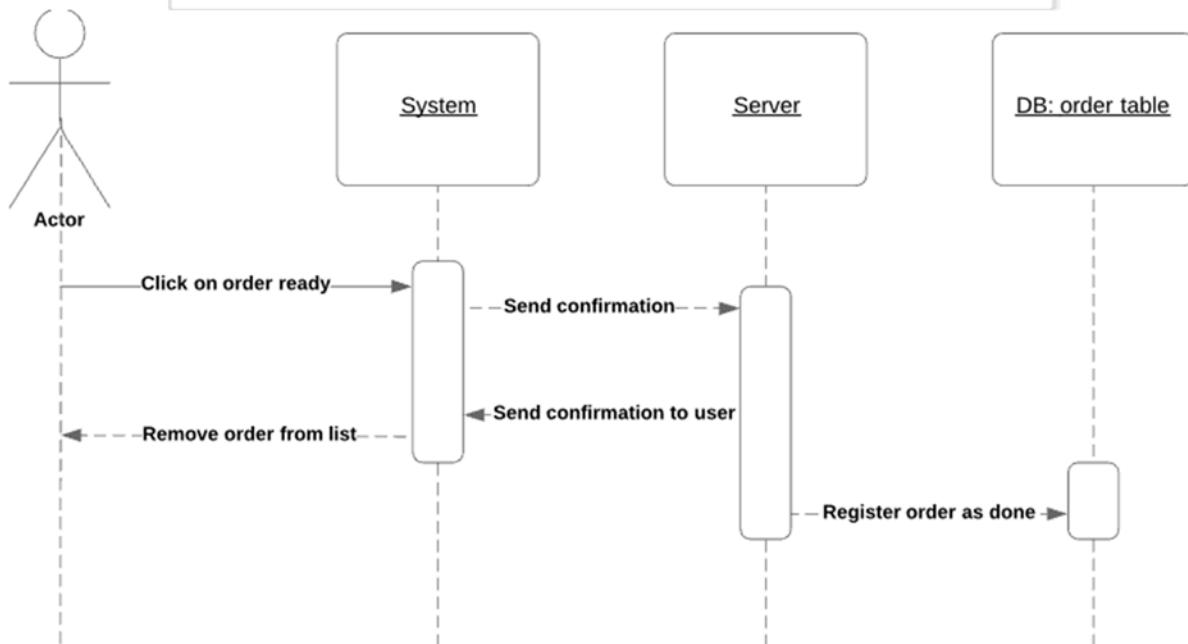
# By: Bassam#100

11 minutes

Item	Quantity	Price
Meal 1	1	7.0\$
item11Name	1	4.4\$
item7Name	1	4.4\$
item4Name	1	4.4\$
		<b>20.2\$</b>

REJECT X

READY ✓



## Use Case: Order Served

Use-Case Name:	Order Served	
Use-Case ID:	UDB005	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Bartender</li> </ul>	
Description:	Bartender marks the order as served to update the database.	
Precondition:	The actor has to be logged in as a bartender.	
Trigger	Click on “Served” Button near the order being prepared	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: Click on the “Served” button near the order being prepared.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system registers the order as served and updates the database.</li> <li>● Step 3: System removes order from being prepared list on the screen</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● NA</li> </ul>	
Conclusion:	This use case is concluded when the order is removed from the live orders tab.	
Post condition:	The order was added to the database as served.	

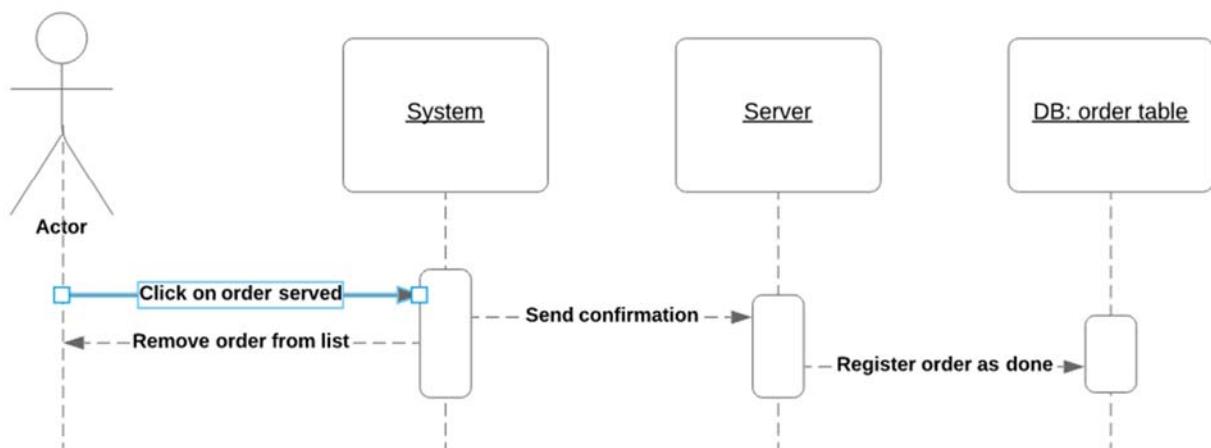
# By: Bassam#100

11 minutes

Item	Quantity	Price
Meal 1	1	7.0\$
item11Name	1	4.4\$
item7Name	1	4.4\$
item4Name	1	4.4\$
		<b>20.2\$</b>

UNSERVED ✕

SERVED ✓



## Use Case: Order Unserved

Use-Case Name:	Order Served	
Use-Case ID:	UDB006	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Bartender</li> </ul>	
Description:	Bartender marks the order as unserved to update the database.	
Precondition:	The actor has to be logged in as a bartender.	
Trigger	Click on “Unserved” Button near the order being prepared	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: Click on the “Unserved” button near the order being prepared.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system registers the order as unserved and updates the database.</li> <li>● Step 3: System removes order from being prepared list on the screen</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● NA</li> </ul>	
Conclusion:	This use case is concluded when the order is removed from the live orders tab.	
Post condition:	The order was added to the database as unserved.	

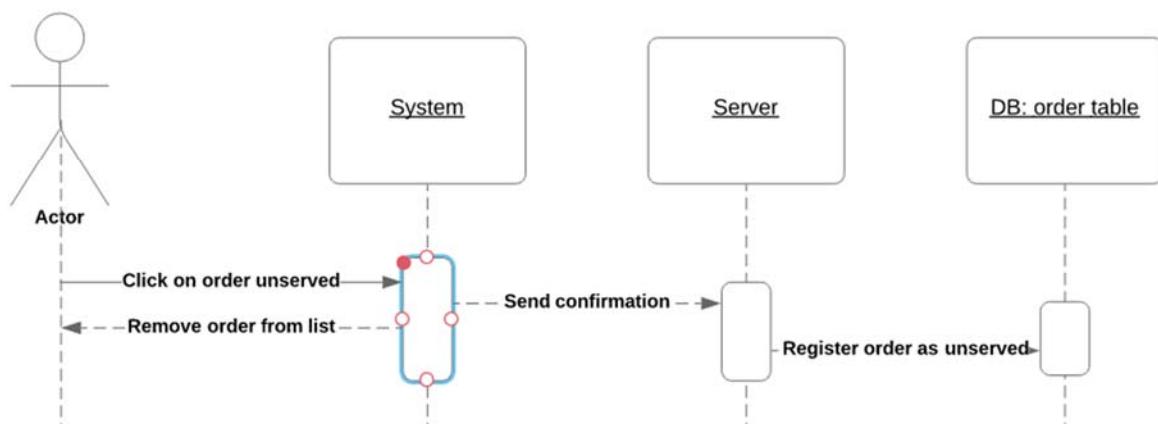
# By: Bassam#100

11 minutes

Item	Quantity	Price
Meal 1	1	7.0\$
item11Name	1	4.4\$
item7Name	1	4.4\$
item4Name	1	4.4\$
		<b>20.2\$</b>

UNSERVED X

SERVED ✓



## Use Case: Order Denied

Use-Case Name:	Order Denied	
Use-Case ID:	UDB007	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Bartender</li> </ul>	
Description:	Bartender user denies the order and adds the reason for denying the order, the system will send a notification to inform the member.	
Precondition:	The actor has to be logged in as a bartender.	
Trigger	Click on “Reject” Button near the order being prepared.	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: Click on the “Reject” button near the order being prepared.</li> <li>● Step 3: User enters the reason and confirms.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system pops a modal and prompts the user to enter a reason.</li> <li>● Step 4: System removes order from being prepared list on the screen;</li> <li>● Step 5: System sends notification and updates the database.</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● NA</li> </ul>	
Conclusion:	This use case is concluded when the notification is sent to the member.	
Post condition:	The order was added to the database as denied.	

By: Bassam#100

11 minutes

Item	Quantity	Price
Meal 1	1	7.0\$
item1Name	1	4.4\$
item7Name	1	4.4\$
item4Name	1	4.4\$
		<b>20.2\$</b>

REJECT ✕

CONFIRM ✓

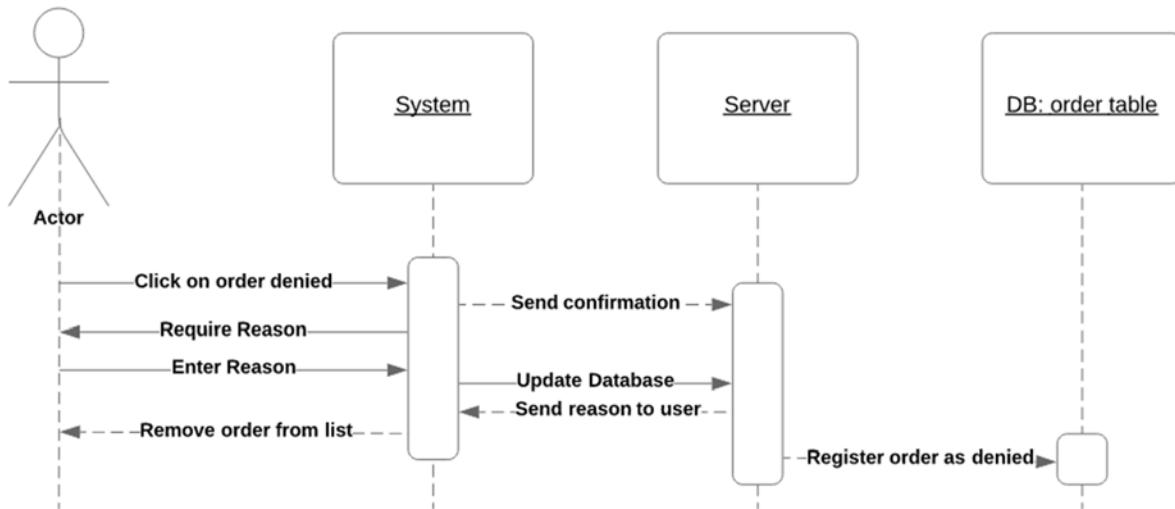
Meal 1	1	4.4\$
item1Name	1	4.4\$
item7Name	1	4.4\$
item4Name	1	4.4\$
		<b>20.2\$</b>

**X** **CONFIRM ✓**

Type the reason for rejecting the order.

Reason

**REJECT ✕**



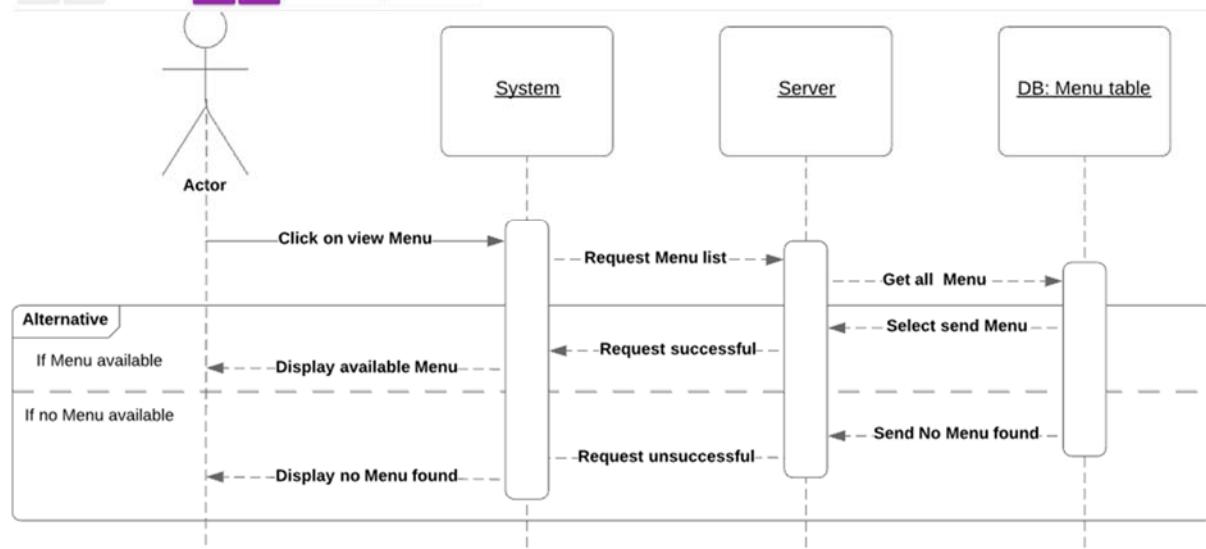
## Use Case: View Menu

Use-Case Name:	View Menu	
Use-Case ID:	UDB008	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Bartender</li> </ul>	
Description:	Admins can look through all the items in the menu at the gym, they can view images, names, nutrition information, availability, etc...	
Precondition:	The actor has to be logged in as an admin.	
Trigger	Click on the “Menu” button on the admin panel.	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: Click on the “Menu” button in the admin panel.</li> <li>● Step 3: Admin can preview the menu.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system will show the available menu offered</li> </ul>
Alternate Courses:	There are no Alt-Steps for this use case.	
Conclusion:	This use case concludes when the available menu is previewed.	
Post condition:	The admin was able to view all the available menu items along with their information.	

File Insert Reload Dev Tools

MEMBERS	COACHES	BARTENDERS	WORKOUTS	MOVES	MACHINES	MENU	PT ADS	MESSAGES			
ID	Name	Image	Description	Calories	Fats	Protein	Carbohydrates	Sugar	Price	Available	Actions
1	Meal 11		•	350	14	55	11	3	7.00	1	
2	Snackbar 1		•	250	4	9	25	0	7.00	1	
3	item3Name		•	174	8	12	22	4	5.40	1	
7	item4Name		•	444	4	44	4	4	4.40	2	
8	item5Name		•	555	5	55	5	5	5.50	5	

1 - 5 / 17 (17) 5 ⏪ 1 ⏩



## Use Case: Add Items

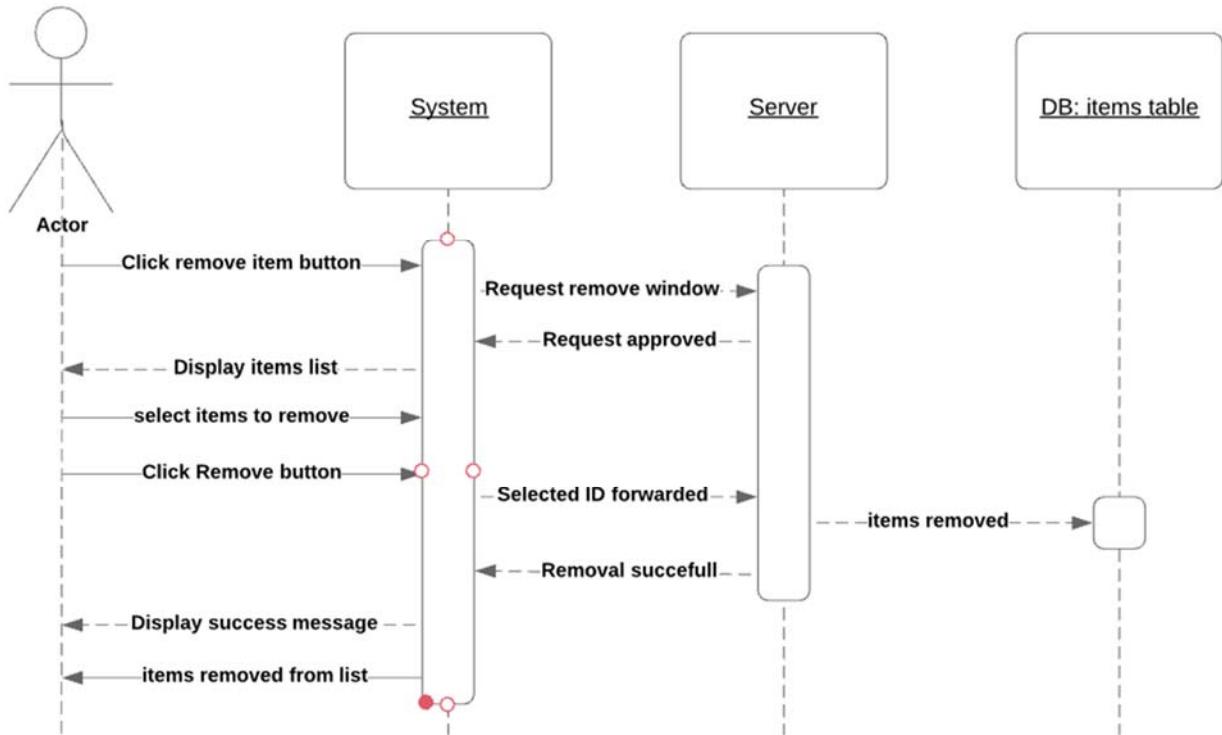
Use-Case Name:	Add Items	
Use-Case ID:	UDB009	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Bartender</li> </ul>	
Description:	<p>Bartender users can add new items to the database through a form in the bartender panel, the bartender enters all the needed information about the item and then triggers the form.</p>	
Precondition:	<p>The actor has to be logged in as an bartender.</p>	
Trigger	<p>Click on the “New item” button.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: Click on the “New Item” button in the admin panel.</li> <li>● Step 3: Fill the required fields in the forms.</li> <li>● Step 4: Clicks on the “Add Item” button.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system opens a new window with a form of the required information to be filled.</li> <li>● Step 5: System saves new records in the database.</li> <li>● Step 6: System shows a message “Workout added successfully”.</li> <li>● Step 7: System adds the new workout to the workouts list so it can be seen by the admin UDA018</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● Alt-Step 5: If entered data is incorrect, system requests admin to reenter.</li> <li>● Alt-Step 6: If the addition of the member was unsuccessful, the system will return and display an error.</li> </ul>	
Conclusion:	<p>This use case concludes when the actor has successfully added a new item.</p>	
Post condition:	<p>The admin was able to add a new menu item to the database.</p>	



## Use Case: Remove Items

Use-Case Name:	Remove item	
Use-Case ID:	UDB010	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Bartender</li> </ul>	
Description:	<p>The bartender has the ability to remove items from the menu(this should be done in rare cases).</p>	
Precondition:	<p>The actor has to be logged in as an admin.</p>	
Trigger	<p>Click on “Remove Item” button</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: Click On “Remove” button in the admin panel.</li> <li>● Step 3: The actor confirms</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system will open a confirmation dialogue.</li> <li>● Step 4: The system removes the selected record from the database.</li> <li>● Step 5 : System shows the message “item removed successfully”.</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● Alt-Step 3: The actor cancels the process and doesn't confirm; the action doesn't proceed.</li> </ul>	
Conclusion:	<p>This use case concludes when the bartender has successfully removed an item.</p>	
Post condition:	<p>The bartender was able to remove the item from the system.</p>	

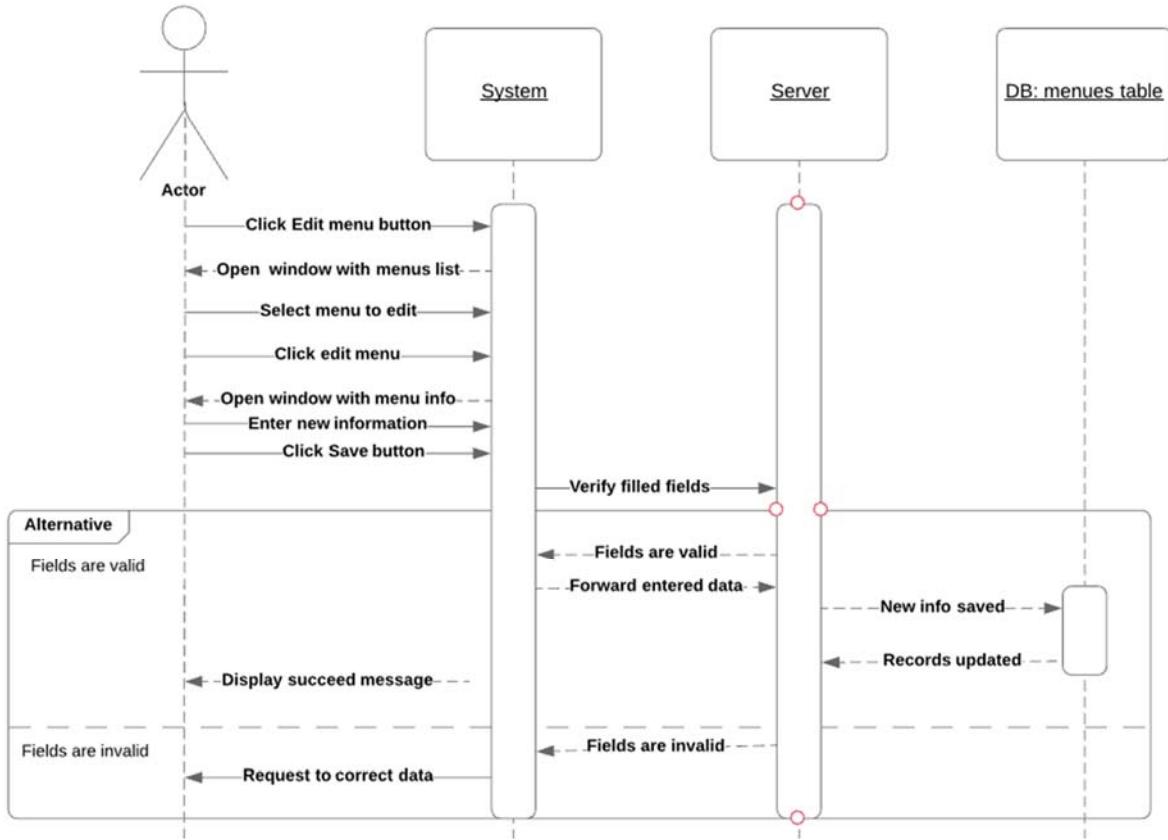
## Preview Image included in use case UDA008



## Use Case: Edit Menu

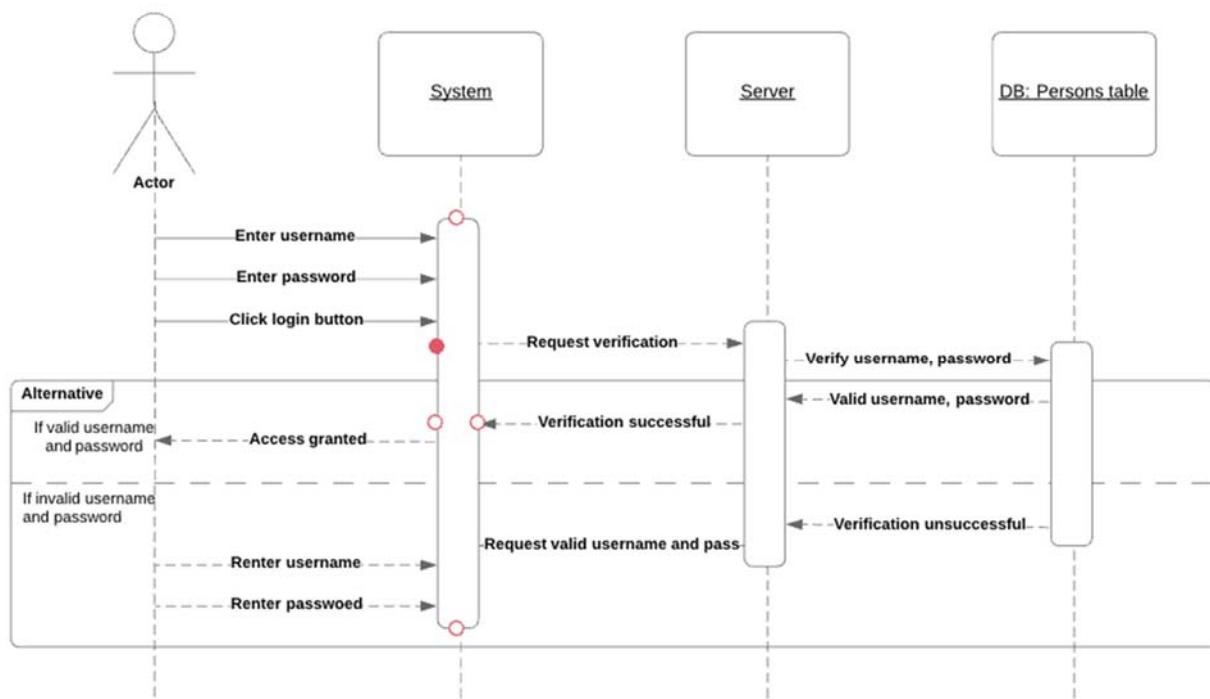
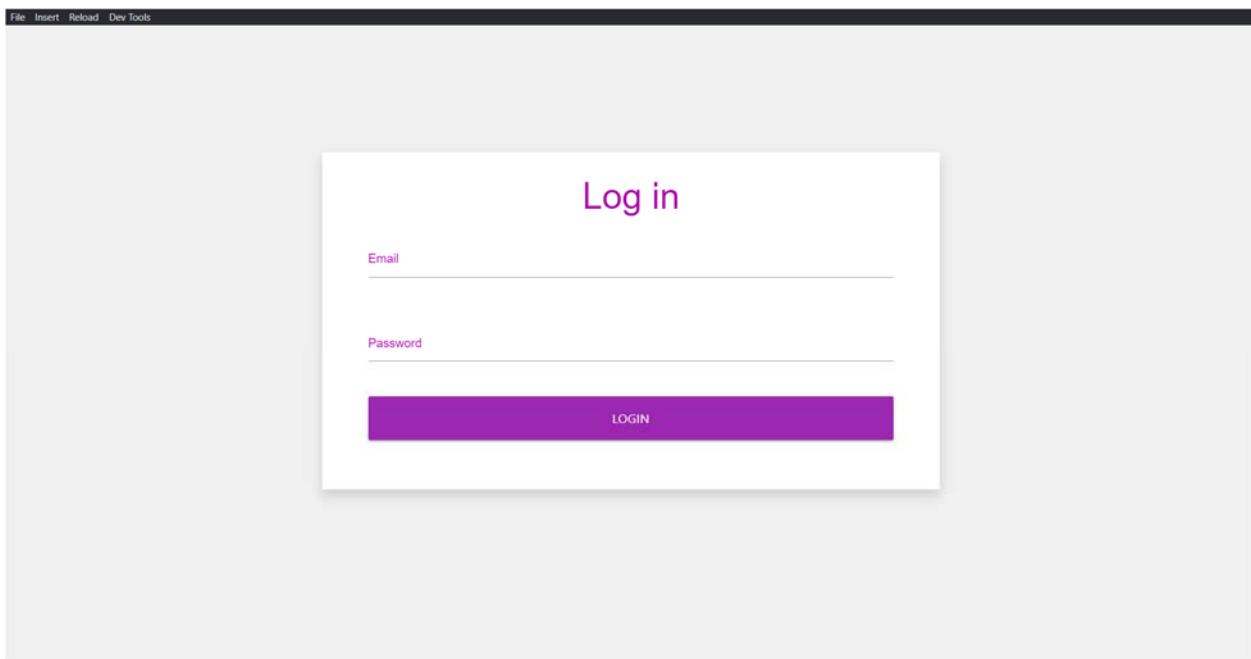
Use-Case Name:	Edit Menu	
Use-Case ID:	UDB011	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Admin</li> <li>● Bartender</li> </ul>	
Description:	Admin users can edit menu items' details and information in the database, including their names, prices, nutritional facts, etc..	
Precondition:	The actor has to be logged in as an admin or bartender.	
Trigger	Click on the "Edit" Button.	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: Click on the "Edit" button beside an item.</li> <li>● Step 3: Admin edits the information to be changed</li> <li>● Step 4: Click the "Save" button.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: System opens a new window with a form filled with all coaches' information.</li> <li>● Step 5: System verifies and saves the changes, then shows an "Update Successful" message.</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● Alt-Step 8: The update was unsuccessful; the system displays a message accordingly.</li> </ul>	
Conclusion:	This use case concludes when the admin has successfully updated the menu.	
Post condition:	The admin was able to edit the record and update the database.	

## Preview Image included in use case UDA008



## Use Case: Login

Use-Case Name:	Login	
Use-Case ID:	UDB001	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Bartender</li> </ul>	
Description:	Bartender user enters his username and password the system authenticates the data entered then the user gains access to the bartender panel.	
Precondition:	To login as bartender you have to be a registered as Bartender	
Trigger	This use case is triggered upon launching Bartender desktop application	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: The actor enters his username.</li> <li>● Step 2: The actor enters his password.</li> <li>● Step 3: The actor clicks on the “Login” button.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 4: The system verifies that an account with this username exists.</li> <li>● Step 5: The system checks whether the username belongs to an admin.</li> <li>● Step 6: The system verifies that the password matches the account linked to the username.</li> <li>● Step 7: The system logs the actor into his account.</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● Alt-Step 4: The actor entered an email with no account linked to it; the system informs the actor and asks him/her to change the email or to register.</li> <li>● Alt-Step 5: The actor entered the wrong password; the system informs the actor and asks him/her to re-enter the password.</li> </ul>	
Conclusion:	This use case concludes when the system is successfully logged into the system and redirected to the bartender panel.	
Post condition:	The actor gains access to his account and the session is created.	

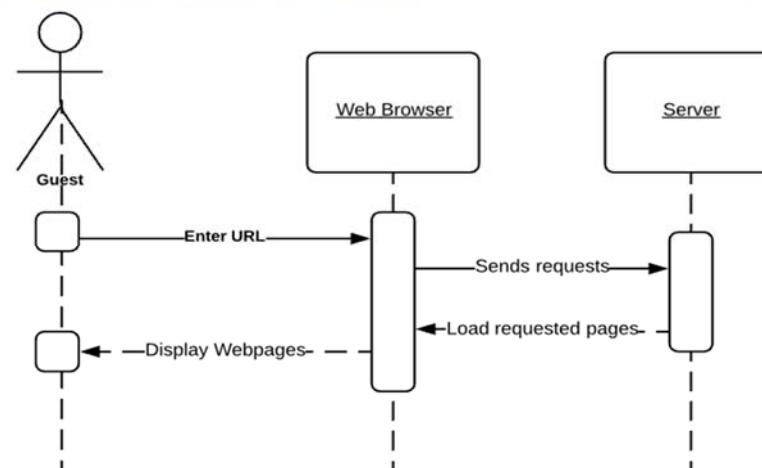


# Web Application Narratives, Forms, and Sequence Diagrams

## Use Case: Navigate

Use-Case Name:	Navigate	
Use-Case ID:	UWG001 / UWM001	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Guest</li> <li>● Member</li> </ul>	
Description:	<p>This use case describes the event of a guest/member just viewing the homepage and information of the website without making any changes that communicates with the database, the guest can view certain subscription plans and services offered by the gym. A member will have access to more features.</p>	
Precondition:	<ul style="list-style-type: none"> <li>● There is no precondition for the guest.</li> <li>● The member must be logged in for the extra features.</li> </ul>	
Trigger	This use case is triggered upon visiting the website.	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: The actor enters the website and is now viewing the home page.</li> <li>● Step 2: The actor clicks on the different sections of the website.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 3: The System directs him/her to the relevant page.</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● Alt-Step 3: The actor is not logged in and tries to access a locked section, the system instead prompts him/her to register or login to access the section.</li> </ul>	
Conclusion:	The navigation event is always active and concludes only when the actor leaves the system and his/her session is closed.	
Post condition:	The actor can view various information about the gym/	

A large banner image at the top of the page shows a man in a gym setting, performing a side plank exercise with dumbbells. The text "Gym Name" is visible at the top center of the image. Below the image, the text "Why put off feeling good?" is displayed in a large, white, sans-serif font.



## Use Case: Register

Use-Case Name:	Register	
Use-Case ID:	UWG002	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Guest</li> </ul>	
Description:	<p>This use case describes the event of a guest choosing to make an account on the system, this event will prompt the actor to enter his/her information (email, name, phone number, address, etc....), thus applying to become a member. Once it's verified that his/her email is not already on the system and that the password meets the requirements, the account will be pending on the system waiting payment and admin activation.</p>	
Precondition:	<p>The actor must have an email that is not already registered on the system.</p>	
Trigger	<p>This use case is triggered upon clicking the “Register” button.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 2: The actor enters an account email.</li> <li>● Step 4: The actor enters his desired account password.</li> <li>● Step 6: The actor chooses whether to enter the rest of the non-required information.</li> <li>● Step 8: Clicks on the “Register” button.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 1: The system directs the actor to the registration page.</li> <li>● Step 3: The system verifies that the email is valid.</li> <li>● Step 5: The system verifies that the entered password meets all the required password conditions (more than 8 characters, at least 1 digit, at least 1 capital letter, etc...)</li> <li>● Step 7: System sends a confirmation email to the actor.</li> <li>● Step 9: System creates the pending account.</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● Alt-Step 3: The email is invalid; the system warns the actor and asks him/her to enter a new one.</li> <li>● Alt-Step 5: The password does not match the required conditions; the system warns the actor and asks him/her to change it.</li> </ul>	
Conclusion:	<p>This use case concludes when the actor clicks on the “Register” button at the end of the form.</p>	
Post condition:	<p>The system creates a pending account for the actor.</p>	

# Sign Up

First Name \_\_\_\_\_ Last Name \_\_\_\_\_

Email \_\_\_\_\_

Password \_\_\_\_\_ Confirm password \_\_\_\_\_

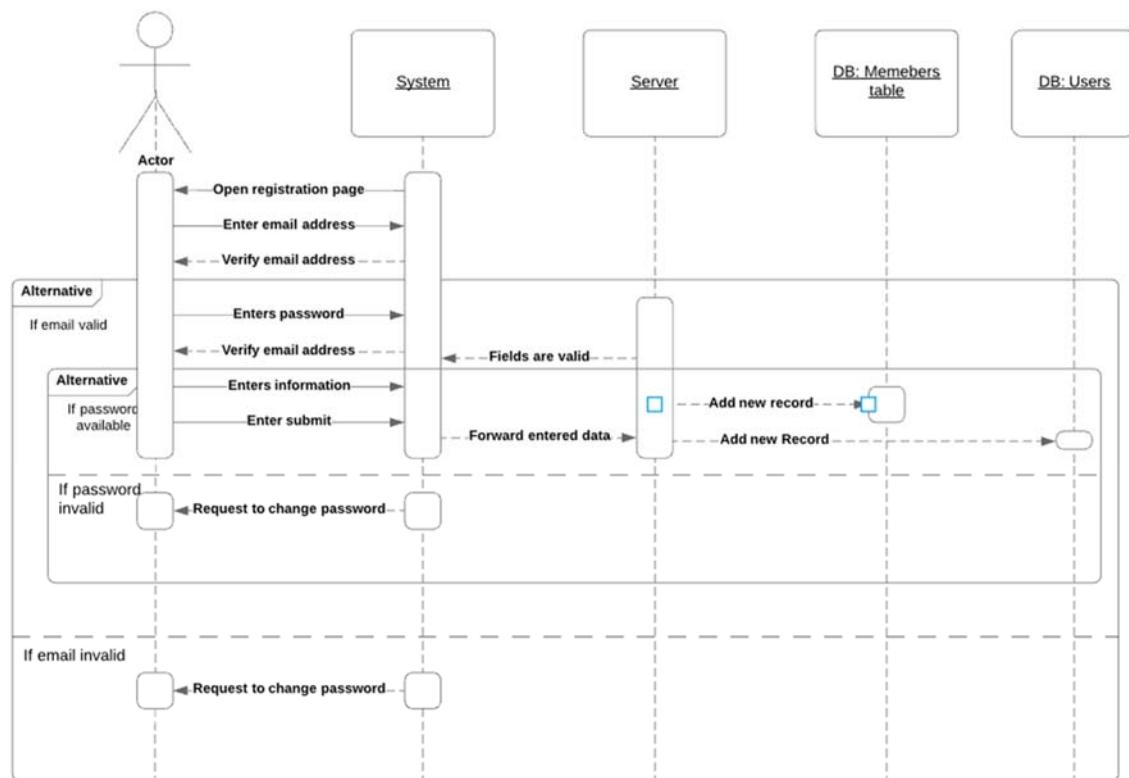
Country: LB (961) Telephone: \_\_\_\_\_

Address: \_\_\_\_\_

Subscription Plan:

<input type="radio"/> 1 month 45\$	<input type="radio"/> 3 months 120\$	<input type="radio"/> 12 months 400\$
---------------------------------------	---	--

**REGISTER ✓**



## Use Case: Confirm Email

Use-Case Name:	Confirm Email	
Use-Case ID:	UMM003	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Guest</li> </ul>	
Description:	<p>The use case describes the event of an actor that has already registered and is required to confirm his email.</p>	
Precondition:	<p>The actor must have already applied for an account.</p>	
Trigger:	<p>This use case is triggered upon finishing the registration process.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: The actor clicks on the link sent to his/her email.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system verifies the email and redirects the actor to the home page.</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● There are no Alt-Steps for this use case.</li> </ul>	
Conclusion:	<p>This use case concludes when the account is successfully verified.</p>	
Post condition:	<p>The actor's account is now verified and only requires payment to become activated.</p>	

Square star noreply New Verify your email for myGym - Hello, Follow this link to verify your email address. [https://gym-senior-2020.firebaseio.com/\\_auth/action?mode=verifyEmail&oobCode=](https://gym-senior-2020.firebaseio.com/_auth/action?mode=verifyEmail&oobCode=) 12:17 AM

Verify your email for myGym [Inbox](#)

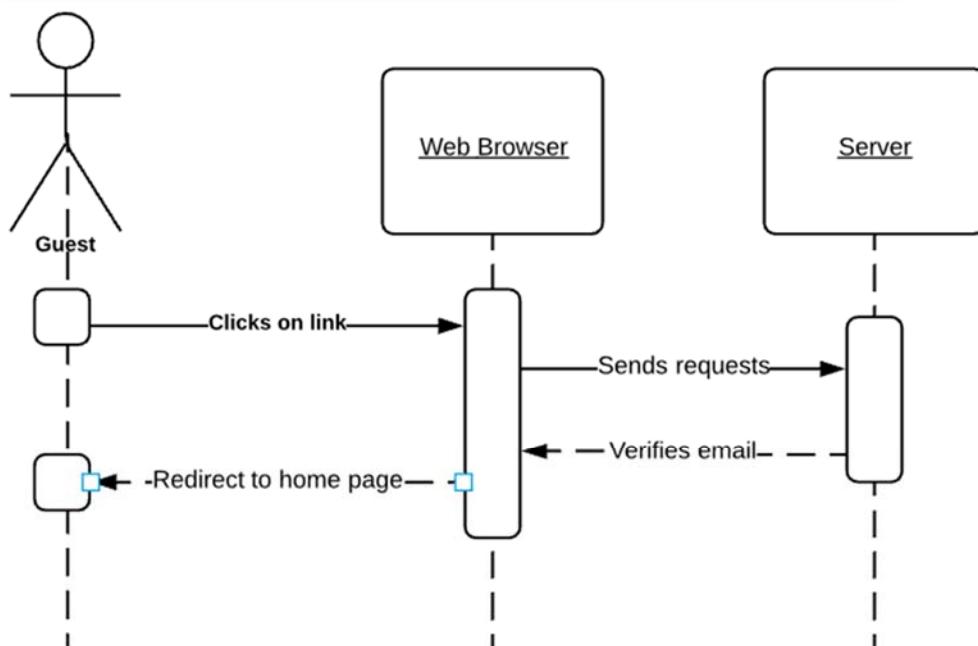
 noreply@mygym-97497.firebaseio.com to me 12:17 AM (0 minutes ago) [Star](#) [Reply](#) [Forward](#) [Print](#) [Report](#)

Hello,  
Follow this link to verify your email address.  
[https://gym-senior-2020.firebaseio.com/\\_auth/action?mode=verifyEmail&oobCode=bc0fFIRWzqX8Q8eYvac8oRe9HICd018HOPyalVZwAAEytLG1\\_o&aoKey=AIZaSyBWk2mYsYIV3bxSBZGMKy5KohLJocTzq&lang=en](https://gym-senior-2020.firebaseio.com/_auth/action?mode=verifyEmail&oobCode=bc0fFIRWzqX8Q8eYvac8oRe9HICd018HOPyalVZwAAEytLG1_o&aoKey=AIZaSyBWk2mYsYIV3bxSBZGMKy5KohLJocTzq&lang=en)  
If you didn't ask to verify this address, you can ignore this email.

Thanks,  
Your myGym team

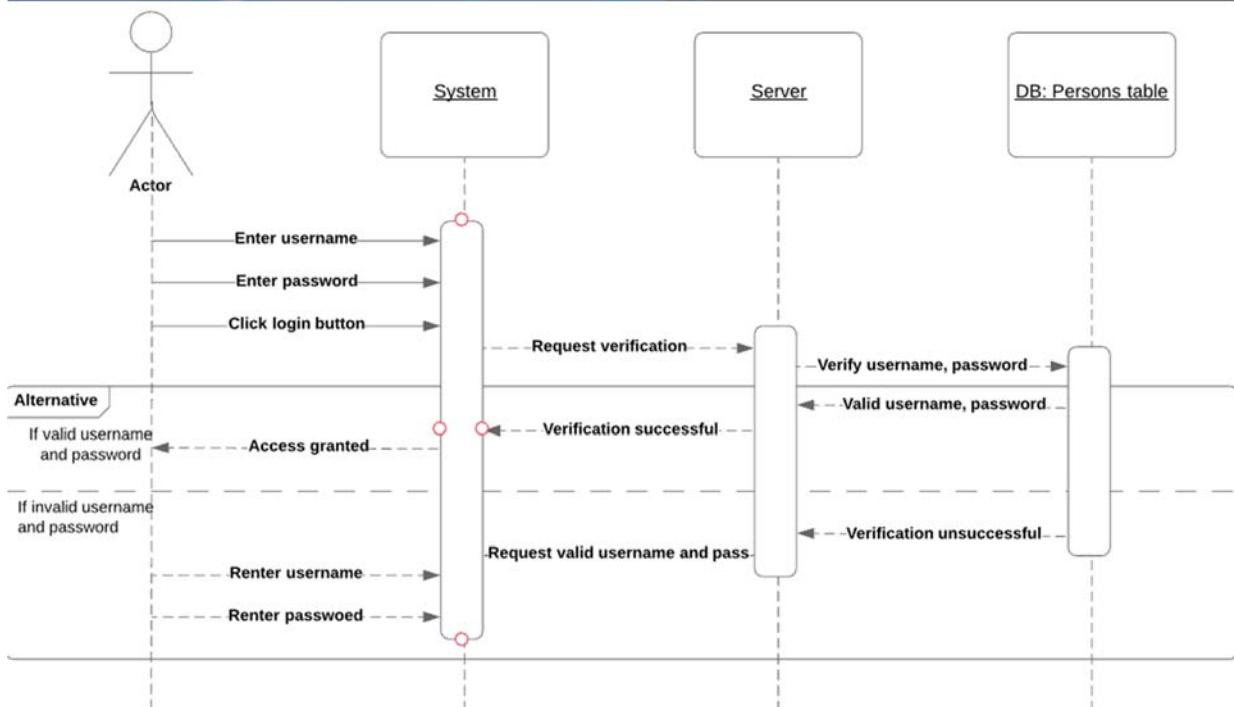
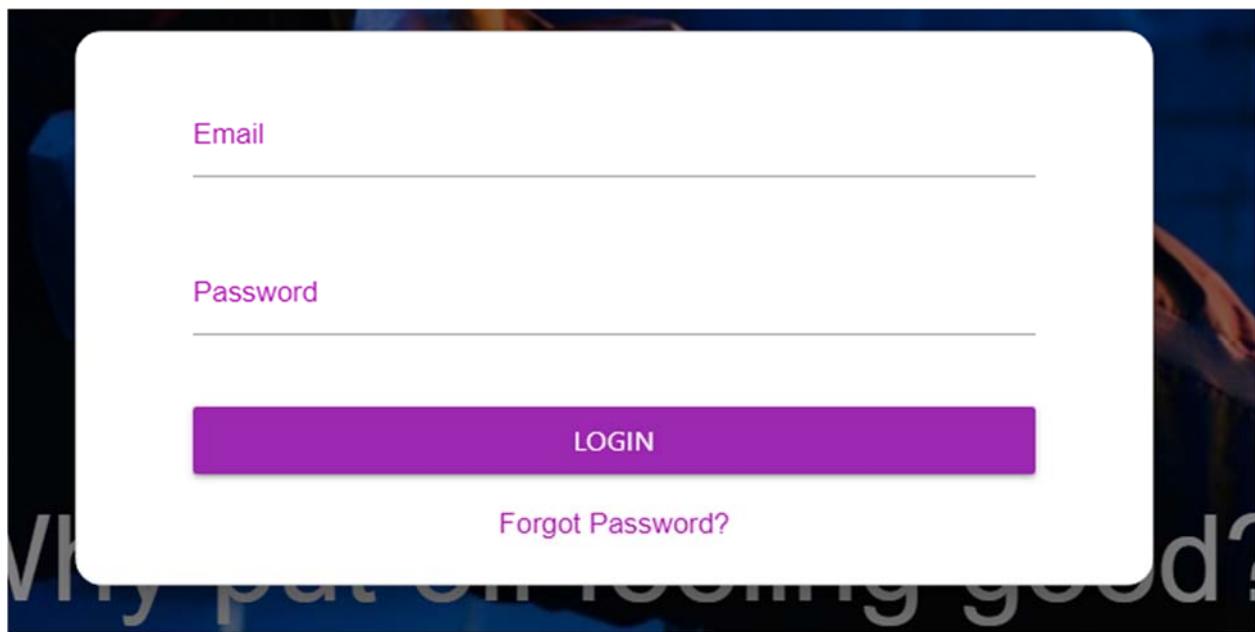
**Your email has been verified**

You can now sign in with your new account



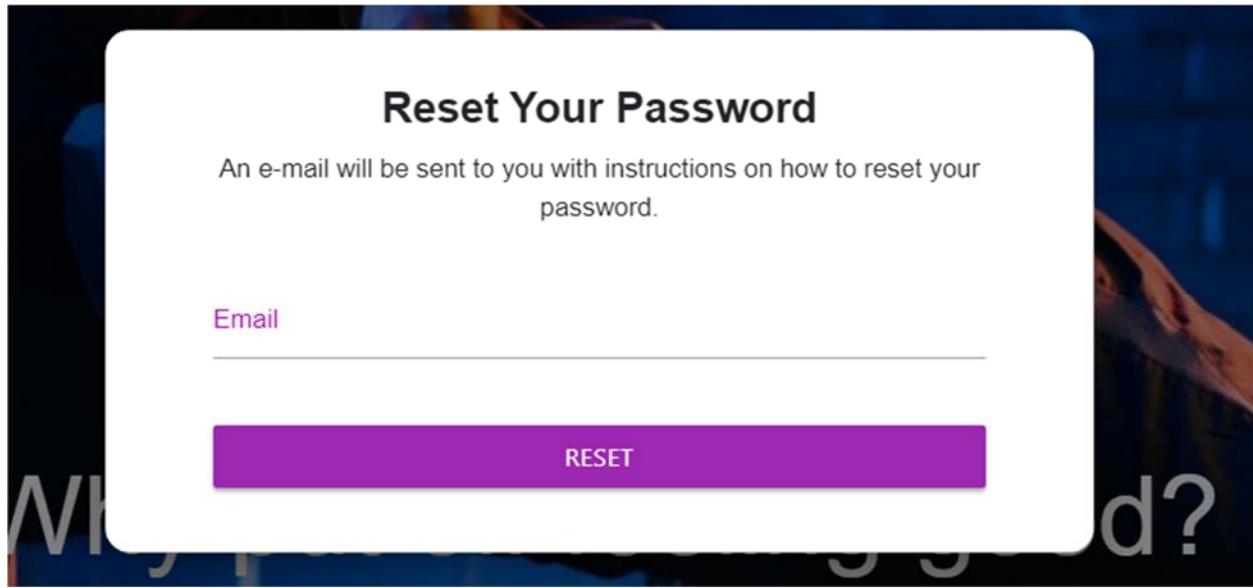
## Use Case: Login

Use-Case Name:	Login	
Use-Case ID:	UWM002	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Member</li> </ul>	
Description:	<p>The use case describes the event of the actor entering his email and password, the system authenticates the data entered, then the actor gains access to his/her respective account.</p>	
Precondition:	<p>The proper account must exist in the database beforehand. In the member's case, he/she must register and be confirmed by the admin.</p>	
Trigger:	<p>This use case is triggered upon clicking the "Login" url.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: The actor enters his account email.</li> <li>● Step 2: The actor enters his password.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 3: The system verifies that an account with this email exists.</li> <li>● Step 4: The system verifies that the password matches the account linked to the email.</li> <li>● Step 5: The system logs the actor into his account.</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● Alt-Step 3: The actor entered an email with no account linked to it; the system informs the actor and asks him/her to change the email or to register.</li> <li>● Alt-Step 4: The actor entered the wrong password; the system informs the actor and asks him/her to re-enter the password.</li> </ul>	
Conclusion:	<p>This use case concludes when the system is successfully logged into the system and redirected to the home page.</p>	
Post condition:	<p>The actor gains access to his account and the session is created.</p>	



## Use Case: Forgot Password

Use-Case Name:	Forgot Password	
Use-Case ID:	UWM003	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Member</li> </ul>	
Description:	<p>The use case describes the event of the actor wanting to change the password to his account, the system allows that after confirmation of account ownership.</p>	
Precondition:	<p>The account connected to the entered email must exist beforehand.</p>	
Trigger:	<p>This use case is triggered upon clicking the “Forgot Password” button.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: The actor enters the email whose account's password he/she wishes to change.</li> <li>● Step 3: The actor clicks on the confirmation email.</li> <li>● Step 5: The actor enters a new password, and confirms.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system verifies that an account with this email exists and sends a confirmation email.</li> <li>● Step 4: The system directs the actor to page with a form to enter the new password.</li> <li>● Step 6: The system checks that the password is valid.</li> <li>● Step 7: The system updates the password and redirects the user to the login page.</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● Alt-Step 2: The actor entered an email with no account linked to it; the system informs the actor and asks him/her to reenter the email or to register.</li> <li>● Alt-Step 6: The actor entered an invalid password; the system informs the actor and asks him/her to enter a new password indicating the required format.</li> </ul>	
Conclusion:	<p>This use case concludes when the system accepts the entered password and redirects the user to the login page.</p>	
Post condition:	<p>The password of the relevant account is updated in the database.</p>	



myGym

**myGym Password Reset** - We have received a password reset request for the account associated with this email. The link to reset your password is below. If you did not make this req... 12:20 AM

myGym Password Reset [Spam X]

myGym@outlook.com via us-imm-node3a.000webhost.io 12:20 AM (0 minutes ago)

to me +

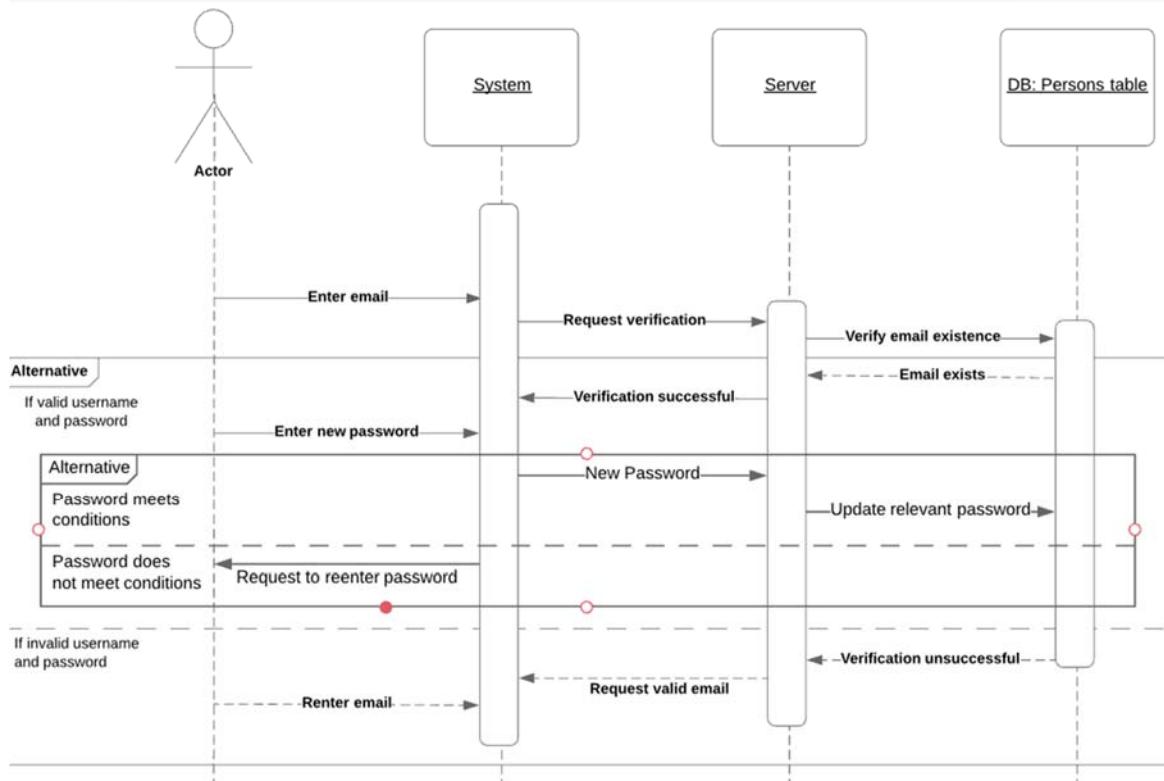
We have received a password reset request for the account associated with this email. The link to reset your password is below. If you did not make this request, you can ignore this email.  
The link expires in 30 minutes.  
<https://gym-senior-2020.000webhostapp.com/resetPassword.php?selector=de0e65cf85ab4317&validator=7605e2e39c881cb4965ccbf81d6104cd7d50ddead58c2af365366631a65e2cba>

## Enter your new password

Password

Confirm password

RESET



## Use Case: Workout History

Use-Case Name:	Workout History	
Use-Case ID:	UWM006	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Member</li> </ul>	
Description:	<p>The use case describes the event of an actor viewing all the workouts he has performed in previous sessions.</p>	
Precondition:	<p>The member must be logged in and navigating the “Progress” tab.</p>	
Trigger:	<p>This use case is triggered upon clicking the “Previous Workouts” option from the navigation options in the progress tab.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: The actor chooses to view his workout history.</li> <li>● Step 3: The actor clicks on the workout.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system opens an activity showing all of his workout history.</li> <li>● Step 4: The system creates a new activity showing the details of the workout.</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● Alt-Step 2: The actor hasn't completed a workout yet; the system will show a text indicating that.</li> </ul>	
Conclusion:	<p>This use case concludes when the actor leaves the “Workout History” activity.</p>	
Post condition:	<p>The actor can now go back and check his previous workouts and plan his next ones based on them.</p>	

Bassam Kaddoura  
MEMBER

Overview    Previous Workouts    Contact

2020-06-11 19:33:33  
Moves: 5

2020-06-06 08:47:04  
Moves: 6

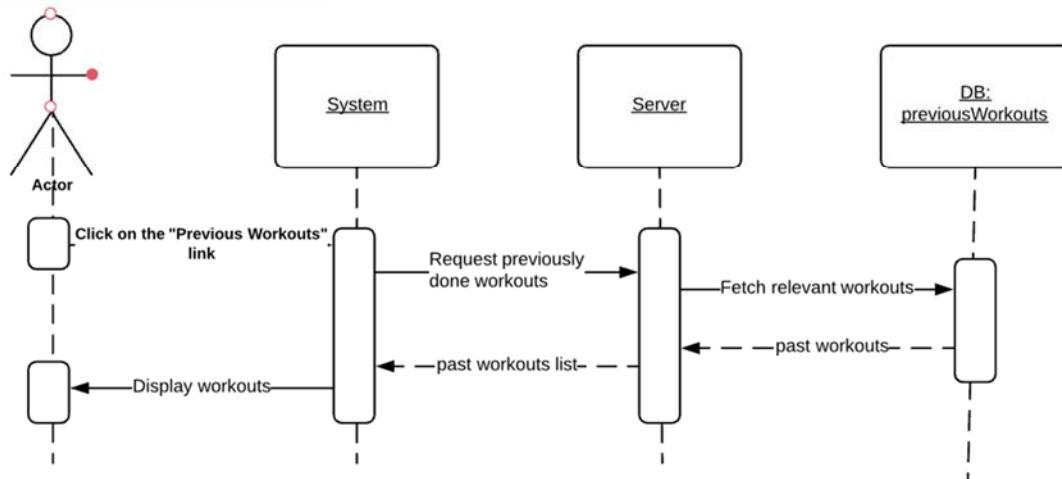
2020-06-06 08:39:36  
Moves: 3

Band Hip Adductions

Barbell Full Squat

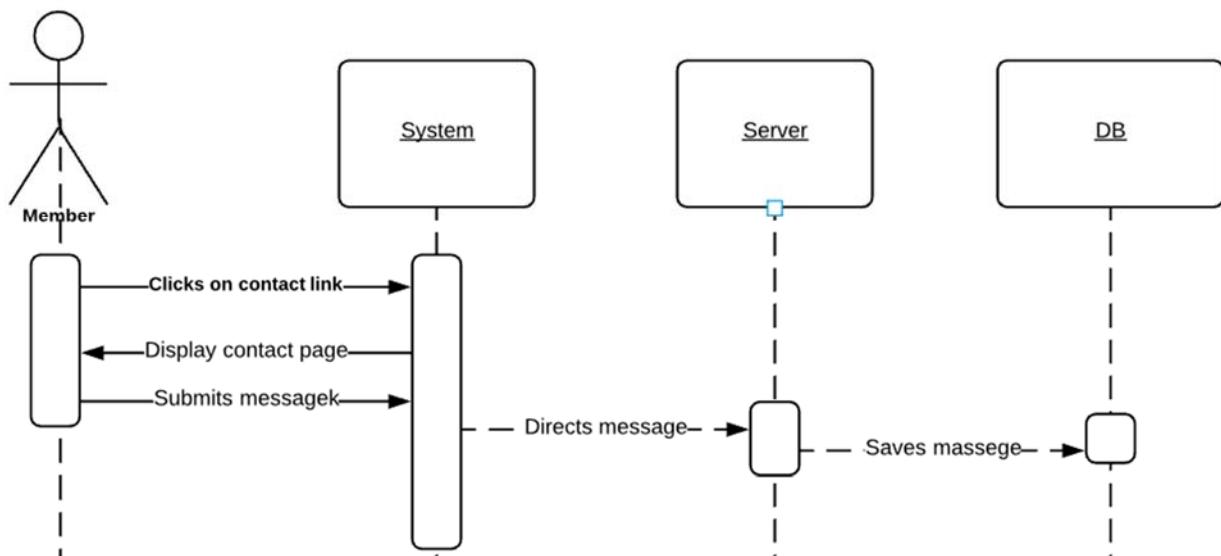
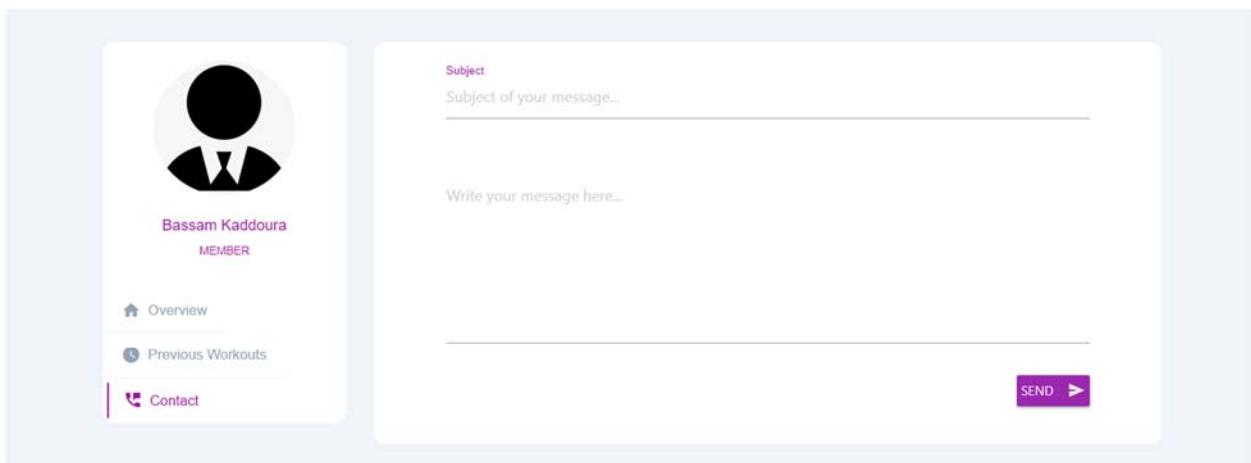
Calves-SMR

2020-06-05 05:20:31  
Moves: 2



## Use Case: Contact

Use-Case Name:	Contact	
Use-Case ID:	UWM010	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Member</li> <li>● Guest</li> </ul>	
Description:	<p>The use case describes an actor wanting to contact the gym, he will be presented with methods to do so, like emailing the gym, or calling by phone.</p>	
Precondition:	No preconditions for this use case.	
Trigger:	This use case is triggered upon clicking the “Contact” option from the navigation bar.	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: The actor chooses to send an email.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system opens a form for the actor to fill.</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● There are no Alt-Steps for this use case.</li> </ul>	
Conclusion:	This use case concludes upon successful initiation of the follow-up pages.	
Post condition:	The members now have the ability to contact the gym using the method they desire.	



## Use Case: View Profile

Use-Case Name:	View Profile	
Use-Case ID:	UWM011	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Member</li> </ul>	
Description:	The use case describes an actor wanting to view his own profile.	
Precondition:	The member must be logged in.	
Trigger:	This use case is triggered upon clicking the “Profile” option from the navigation bar.	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: The actor clicks on the “Profile” option.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system opens the actor’s profile page.</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● There are no Alt-Steps for this use case.</li> </ul>	
Conclusion:	This use case concludes upon leaving the profile page.	
Post condition:	The member now has access to his profile.	

Gym Name

Home Our Team About Contact Us

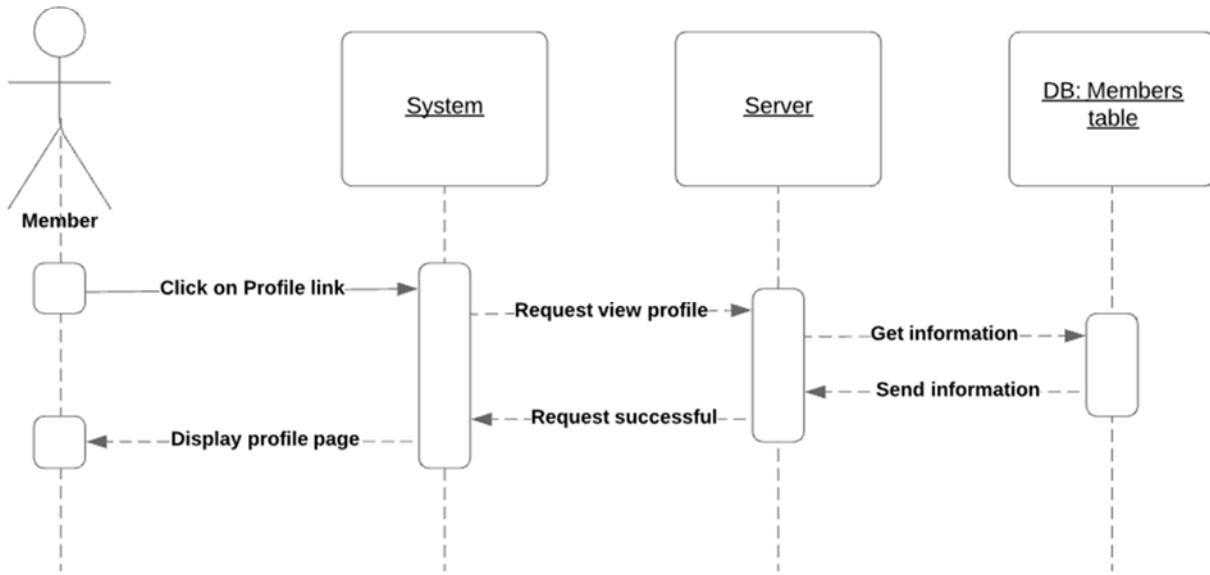
Bassam Kaddoura  
MEMBER

**Overview**

Previous Workouts

Contact

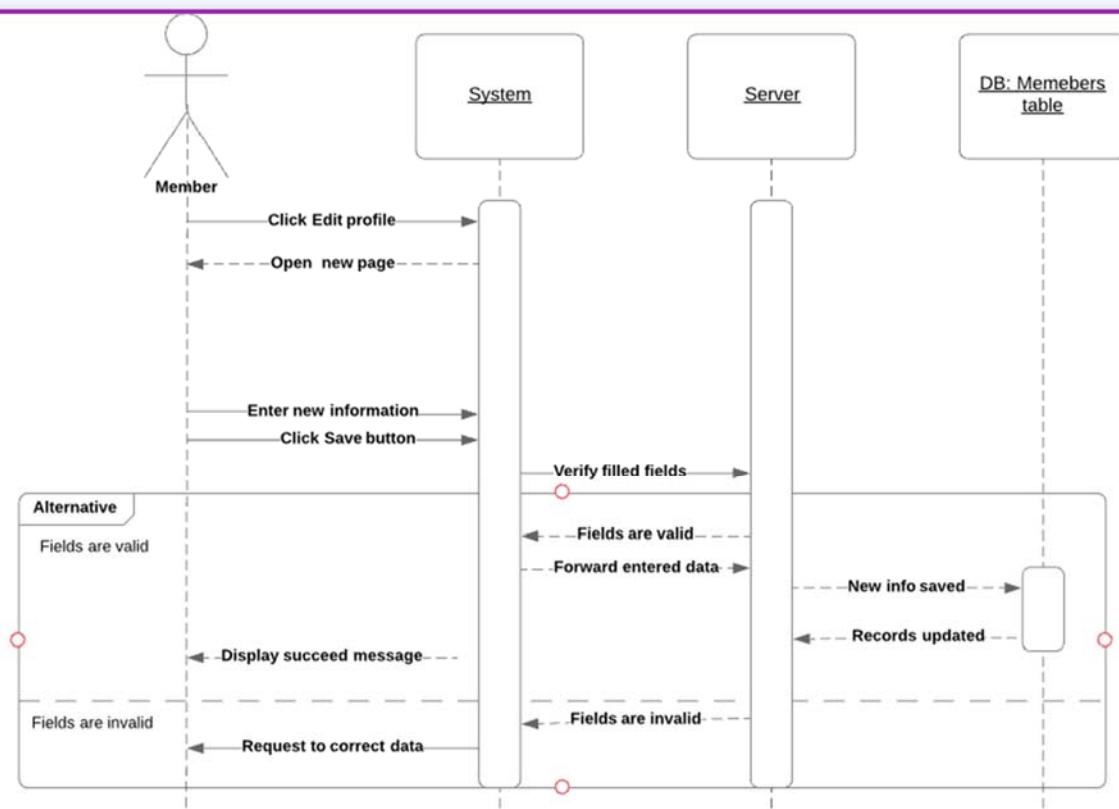
First Name	Bassam	Last Name	Kaddoura
Email	bsmkdd@outlook.com		
Country	LB (961)	Telephone	100029
Address		Address	
Subscription Plan:	120\$/3Months		
Expires:	27-06-2020		



## Use Case: Edit Profile

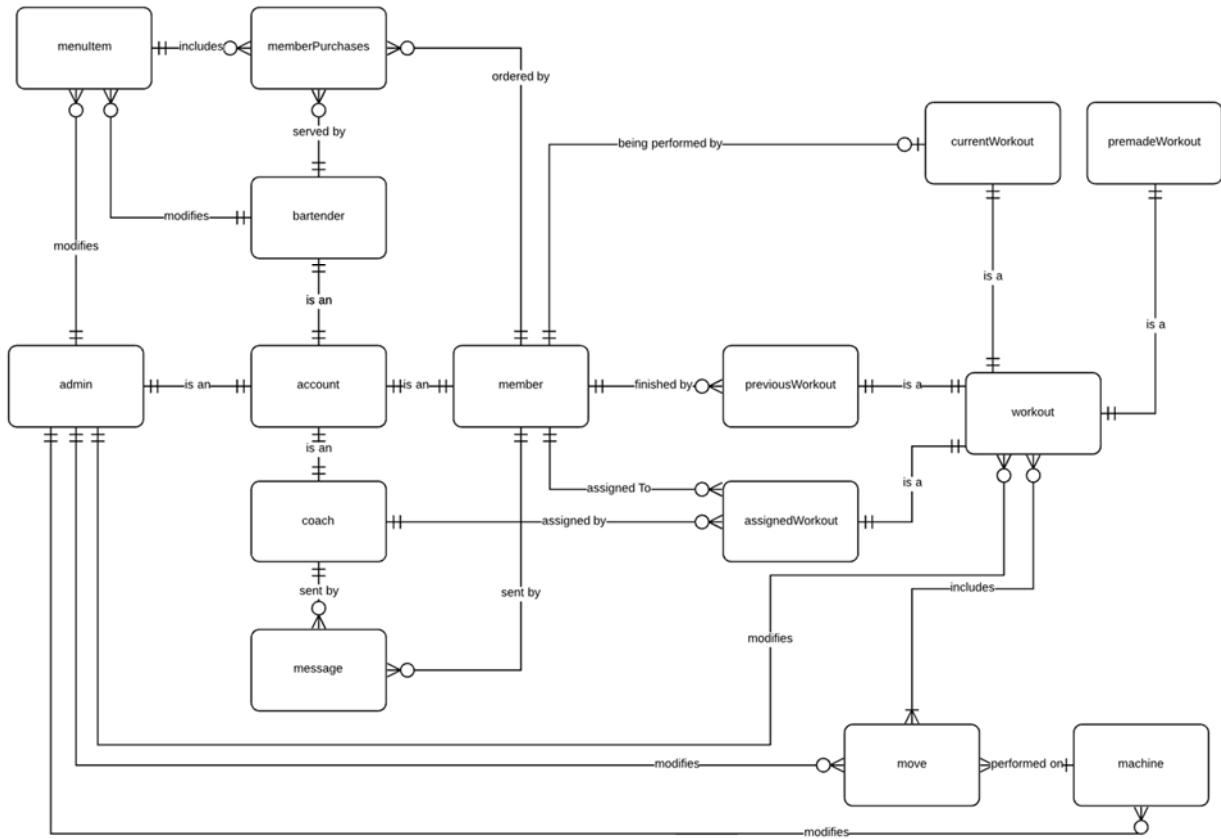
Use-Case Name:	Edit Profile	
Use-Case ID:	UWM012	
Priority:	High	
Participating Actors:	<ul style="list-style-type: none"> <li>● Member</li> </ul>	
Description:	<p>The use case describes the event of an actor editing specific details in his profile, this will cause his information on the database to change.</p>	
Precondition:	<p>The member must be logged in.</p>	
Trigger	<p>This use case is triggered upon clicking the edit icon beside a specific field.</p>	
Typical Course Events:	Actor Action	System Response
	<ul style="list-style-type: none"> <li>● Step 1: The actor edits a specific field in his profile.</li> </ul>	<ul style="list-style-type: none"> <li>● Step 2: The system verifies that the new data is valid.</li> <li>● Step 3: The system updates the data in the database.</li> <li>● Step 4: The system informs the member that his information has been updated.</li> </ul>
Alternate Courses:	<ul style="list-style-type: none"> <li>● Alt-Step 2: The entered data is not valid; the system displays a warning and asks the user to try again.</li> </ul>	
Conclusion:	<p>This use case concludes when the field that's being edited goes out of focus.</p>	
Post condition:	<p>The actor now has the ability to change various details about himself.</p>	

The screenshot shows a member profile page for Bassam Kaddoura. At the top, there's a navigation bar with links for Home, Our Team, About, Contact Us, and a user icon. Below the navigation is a header with the text "Gym Name". The main content area displays a profile picture of Bassam Kaddoura, his name, and his status as a MEMBER. There are three tabs: Overview, Previous Workouts, and Contact. The Overview tab is selected. On the right, there are several input fields: First Name (Bassam), Last Name (Kaddoura), Email (bsmkdd@outlook.com), Country (LB (961)), Telephone (100029), Address (Address), Subscription Plan (120\$/3Months), and Expires (27-06-2020). Each field has a small edit icon to its right.

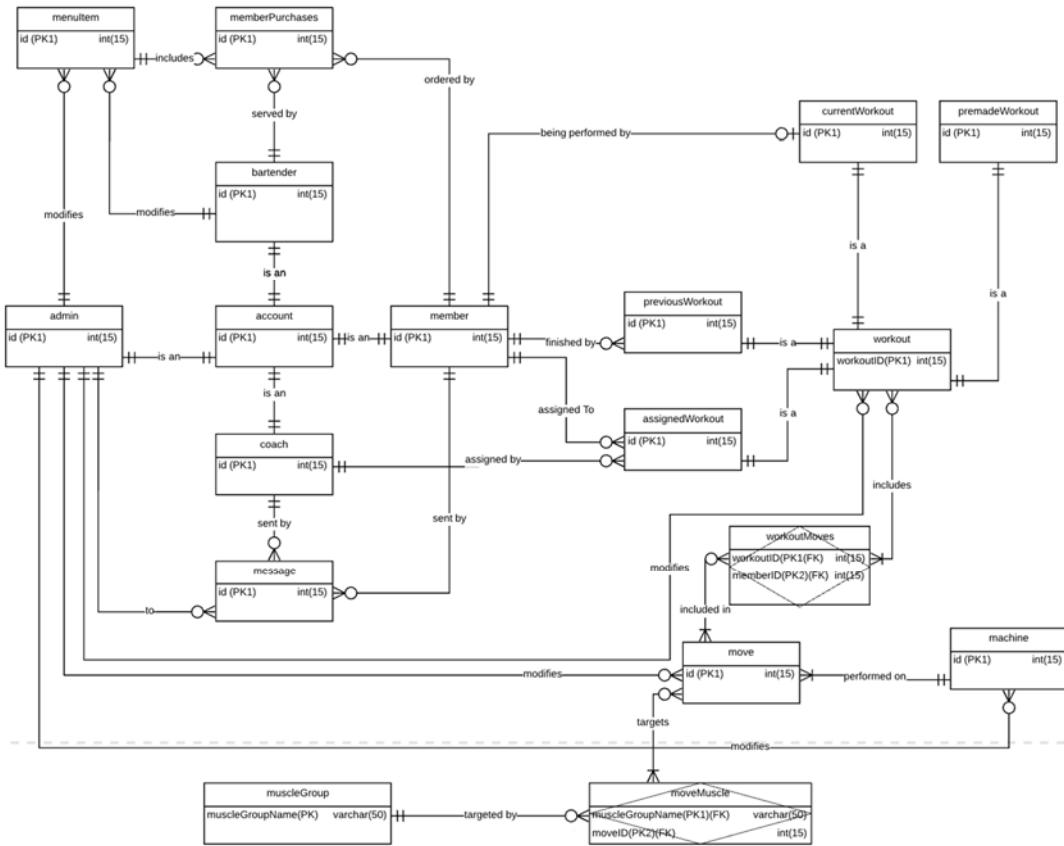


# **Data Modeling**

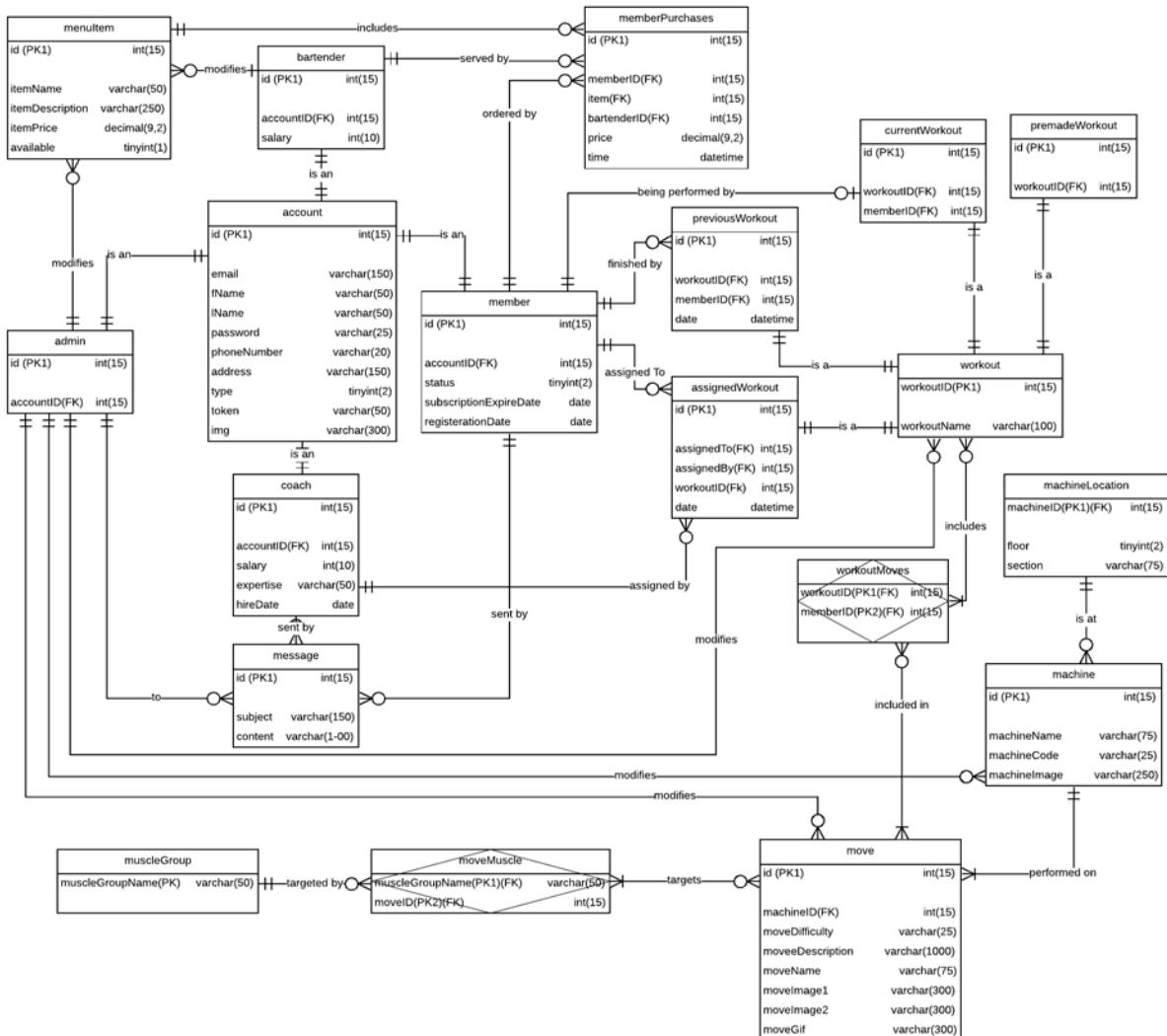
## Data Context Diagram



## Conceptual Data Diagram



# Physical Data Diagram



# **Process Modeling**

## Mobile Application Event Table

Actor	Use Case	Description	Actor Action	System Response
Member	Login	Access the member system by entering valid credentials	The actor tries to login by entering his email and password.	Based on the entered information, the system either displays an error or logs the member in.
	Forgot Password	Request a change of password	The actor requests a password change, he/she enters a new password.	The system validates the newly entered password, and then updates the database.
	Plan Workout	Create a new personal workout	The actor checks specific moves and adds them to the current workout.	The system checks that there are no duplicates and then updates the current workout.
	Workout Type	Choose between our premade workouts or make your own	The actor navigates to the workout type he/she desires.	The system shows the available workouts of that type.
	Edit Current Workout	Edit your in-progress workout	The actor chooses to add new moves to the workout or finish it.	The system updates the member's current workout accordingly.
	Advised Workout	View the workout advised by the coach	The actor views and clicks on his/her advised workout.	The system shows the requested information.
	Workout Content	Details of each of the workout's moves	The actor clicks on a specific premade/advised workout.	The system presents with the moves included in that workout.
	Progress	Check your overall progress	The actor chooses to check his progress tab.	The system fetches and shows his progress in all available aspects.
	Workout History	View all of your previous workouts	The actor navigates to his/her previous workouts tab.	The system shows all the previously-finished workouts.

Actor	Use Case	Description	Actor Action	System Response
Member	Find Machine	Find in which section a machine is located	The actor clicks on the machine's name in the move's preview.	The system shows a map scheme and a picture of the machine.
	PT Advertisements	Personal trainers' listings section	The actor navigates to the "Hire a PT" tab.	The system shows the current available listings.
	Contact PT	Show ways to contact listed trainers	The actor clicks on a PT listing.	The system expands the card to show the possible ways to contact the PT.
	Machine Help	Show more information about a machine	The actor navigates to the "Machine Help" tab.	The system asks the member to scan the QR code on the machine.
	Scan QR Code	Scan the QR code on the machine	The actor scans the QR code.	The system shows the moves that are performed on the scanned machine.
	Moves Preview	View listed moves	The actor clicks on a specific move.	The system opens a preview of the move and some details about it.
	Assistance	Request Assistance	The actor navigates to the "Assist Me" tab.	The system asks the member to scan the nearest QR code.
	Scan QR Code	Scan the nearest QR code	The actor scans the QR code.	The system sends a notification to the coach.
	Feedback	Submit your personal feedback	The actor inserts the details of the feedback.	The system stores the feedback so that the admin can read it later.
	Edit Profile	Edit certain details in your profile	The actor edits the details in the profile.	The system validates and updates the information.

Actor	Use Case	Description	Actor Action	System Response
Coach	Login	Access the coach system by entering valid credentials	The actor tries to login by entering his email and password.	Based on the entered information, the system either displays an error or logs the coach in.
	Assign Workout	Make a workout for a member	The actor enters the member's ID.	The system verifies the ID and lists possible moves.
	Assign Moves	Choose included moves	The actor chooses moves from the listed ones.	The system confirms that at least one move was selected and then notifies and assigns the workout.
	Assistance Response	Respond to an assistance request	The actor clicks on an assistance response.	The system sends a notification to the member.
	Report Incident	Report incidents and machine malfunctions	The actor enters the details of the incident/malfunction.	The system stores the report in the database,

## Desktop Application Event Table

Actor	Use Case	Description	Actor Action	System Response
Admin	Login	Access the admin system by entering valid credentials	The actor tries to login by entering his email and password.	Based on the entered information, the system either displays an error or logs the admin in.
	View Members	View all the members on the system	Navigate to the "Members" tab.	The system opens the tab and fills the table with the relevant information.
	Add Members	Add new members to the system	The actor fills the required details and confirms.	The system validates and confirms the details, then adds the new record to the system.
	Edit Members	Edit members' data	The actor clicks and changes the target field.	The system makes the field editable then updates the database with the newly entered data.
	Remove Members	Remove members from the system	The actor clicks on the remove button beside the record and confirms.	The system removes the record from the database.
	View Coaches	View all the coaches on the system	Navigate to the "Coaches" tab.	The system opens the tab and fills the table with the relevant information.
	Add Coaches	Add new coaches to the system	The actor fills the required details and confirms.	The system validates and confirms the details, then adds the new record to the system.
	Edit Coaches	Edit coaches' data	The actor clicks and changes the target field.	The system makes the field editable then updates the database with the newly entered data.
	Remove Coaches	Remove coaches from the system	The actor navigates to his/her previous workouts tab.	The system removes the record from the database.

Actor	Use Case	Description	Actor Action	System Response
Admin	View Bartenders	View all the bartenders on the system	Navigate to the “Bartenders” tab.	The system opens the tab and fills the table with the relevant information.
	Add Bartenders	Add new bartenders to the system	The actor fills the required details and confirms.	The system validates and confirms the details, then adds the new record to the system.
	Edit Bartenders	Edit bartenders' data	The actor clicks and changes the target field.	The system makes the field editable then updates the database with the newly entered data.
	Remove Bartenders	Remove bartenders from the system	The actor navigates to the workout type he/she desires.	The system removes the record from the database.
	View Machines	View all the machines on the system	Navigate to the “Machines” tab.	The system opens the tab and fills the table with the relevant information.
	Add Machines	Add new machines to the system	The actor fills the required details and confirms.	The system validates and confirms the details, then adds the new record to the system.
	Edit Machines	Edit machines' data	The actor clicks and changes the target field.	The system makes the field editable then updates the database with the newly entered data.
	Remove Machines	Remove machines from the system	The actor chooses to check his progress tab.	The system removes the record from the database.

Actor	Use Case	Description	Actor Action	System Response
Admin	View Workouts	View all the workouts on the system	Navigate to the “Workouts” tab.	The system opens the tab and fills the table with the relevant information.
	Add Workouts	Add new workouts to the system	The actor fills the required details and confirms.	The system validates and confirms the details, then adds the new record to the system.
	Edit Workouts	Edit workouts' details	The actor clicks and changes the target field.	The system makes the field editable then updates the database with the newly entered data.
	Remove Workouts	Remove workouts from the system	The actor navigates to the workout type he/she desires.	The system removes the record from the database.
	View Menu	View all the members on the system	Navigate to the “Menu” tab.	The system opens the tab and fills the table with the relevant information.
	Add Items	Add new items to the menu	The actor fills the required details and confirms.	The system validates and confirms the details, then adds the new record to the system.
	Remove Items	Remove items from the menu	The actor clicks on a specific premade/advised workout.	The system removes the record from the database.
	Edit Menu	Edit items on the menu	The actor clicks and changes the target field.	The system makes the field editable then updates the database with the newly entered data.
	View Orders	View previous orders	Navigate to the “Orders” tab.	The system shows all the previously-served orders.
	View Reports	View members' feedback and coaches' reports	Navigate to the “Reports” tab.	The system shows all the received reports.

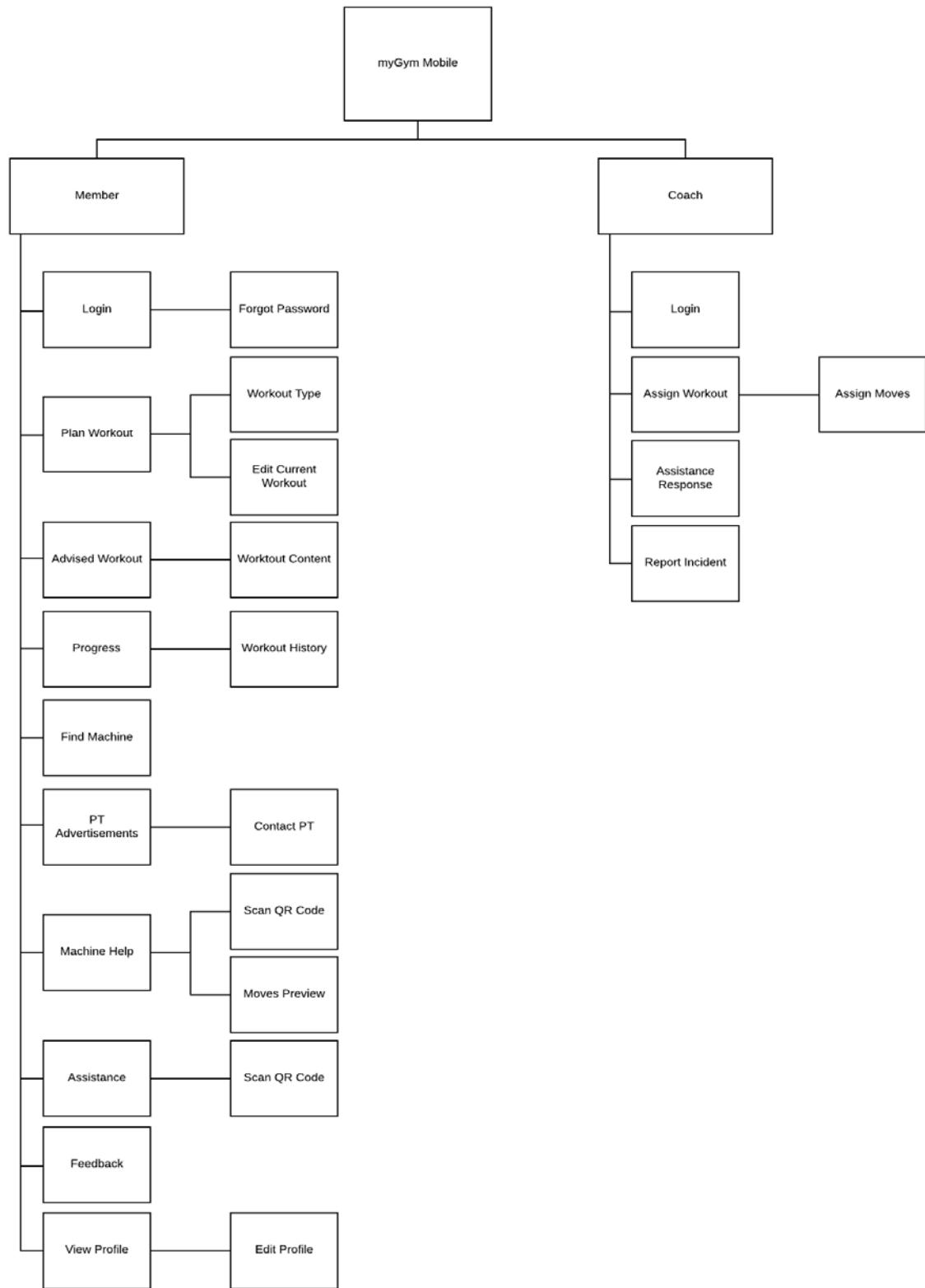
Actor	Use Case	Description	Actor Action	System Response
Bartender	Login	Access the Bartender system by entering valid credentials	The actor tries to login by entering his email and password.	Based on the entered information, the system either displays an error or logs the bartender in.
	View Orders	View received orders	The actor navigates to the “Orders” tab.	The system shows the current orders.
	Confirm Order	Confirm order availability	The actor confirms the availability of the order.	The system notifies the member.
	Order Ready	Notify member that the order is ready	The actor marks the order as ready.	The system stores the order and notifies the member,

## Web Application Event Table

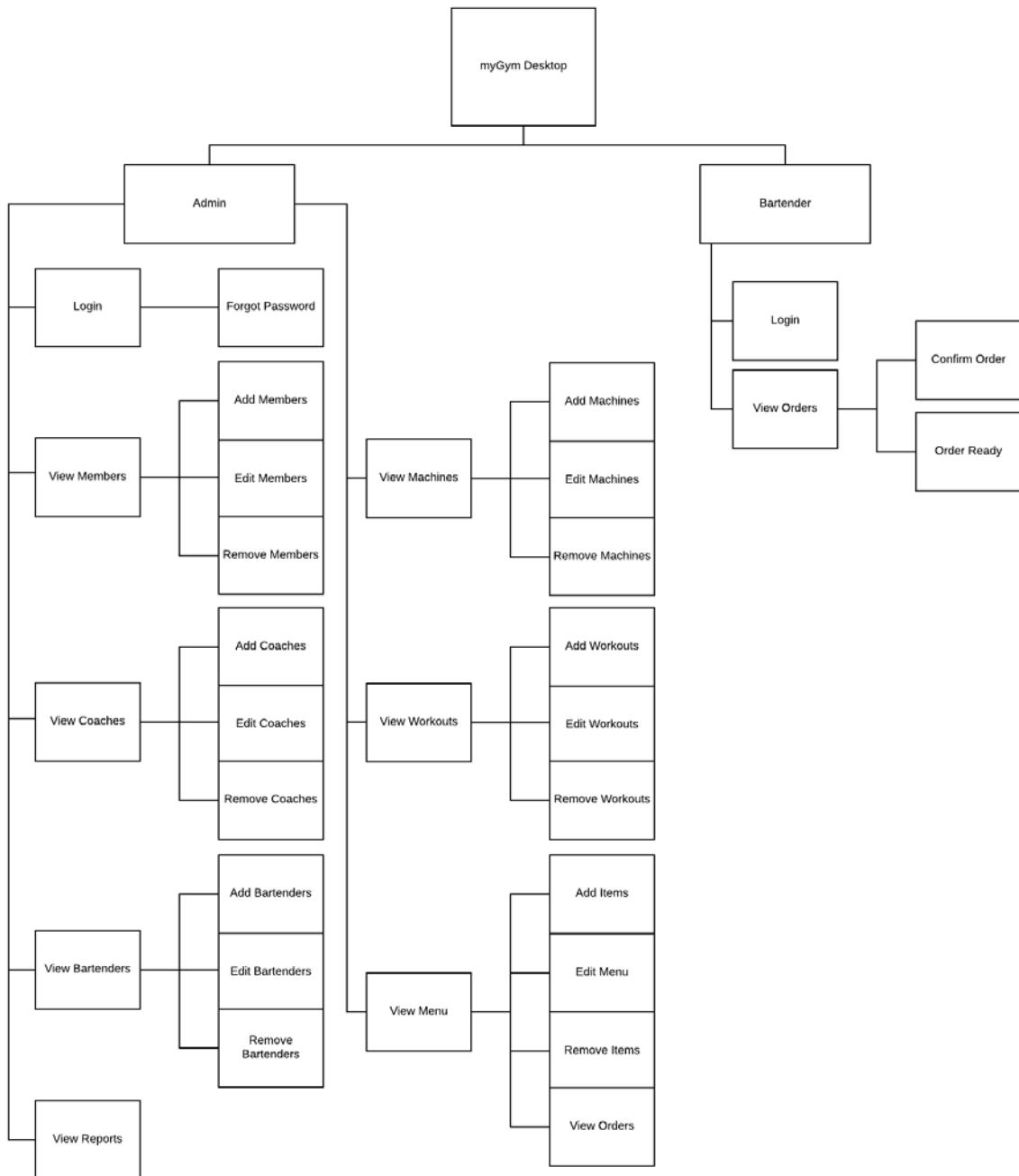
Actor	Use Case	Description	Actor Action	System Response
Member	Login	Access the member system by entering valid credentials	The actor tries to login by entering his email and password.	Based on the entered information, the system either displays an error or logs the member in.
	Forgot Password	Request a change of password	The actor requests a password change, he/she enters a new password.	The system validates the newly entered password, and then updates the database.
	Progress	Check your overall progress	The actor chooses to check his progress tab.	The system fetches and shows his progress in all available aspects.
	Workout History	View all of your previous workouts	The actor navigates to his/her previous workouts tab.	The system shows all the previously-finished workouts.
	PT Advertisements	Personal trainers' listings section	The actor navigates to the "Hire a PT" tab.	The system shows the current available listings.
	Contact PT	Show ways to contact listed trainers	The actor clicks on a PT listing.	The system expands the card to show the possible ways to contact the PT.
	Feedback	Submit your personal feedback	The actor inserts the details of the feedback.	The system stores the feedback so that the admin can read it later.
	View Profile	View your profile	The actor clicks on the "Profile" tab.	The system opens the actor's profile.
	Edit Profile	Edit certain details in your profile	The actor edits the details in the profile.	The system validates and updates the information.

Actor	Use Case	Description	Actor Action	System Response
Guest	Navigate	Navigate the different sections of the website	The actor navigates through different sections of the website.	The system fetches information relative to the selected section.
	Register	Apply to become a member in the gym	The actor applies to become a member by entering various information.	The system validates the entered information and creates a pending account.
	Confirm Email	Confirm that the email entered is real	The actor follows the instructions in the confirmation email.	The system marks the email of the account as real but keeps it as pending until paid.

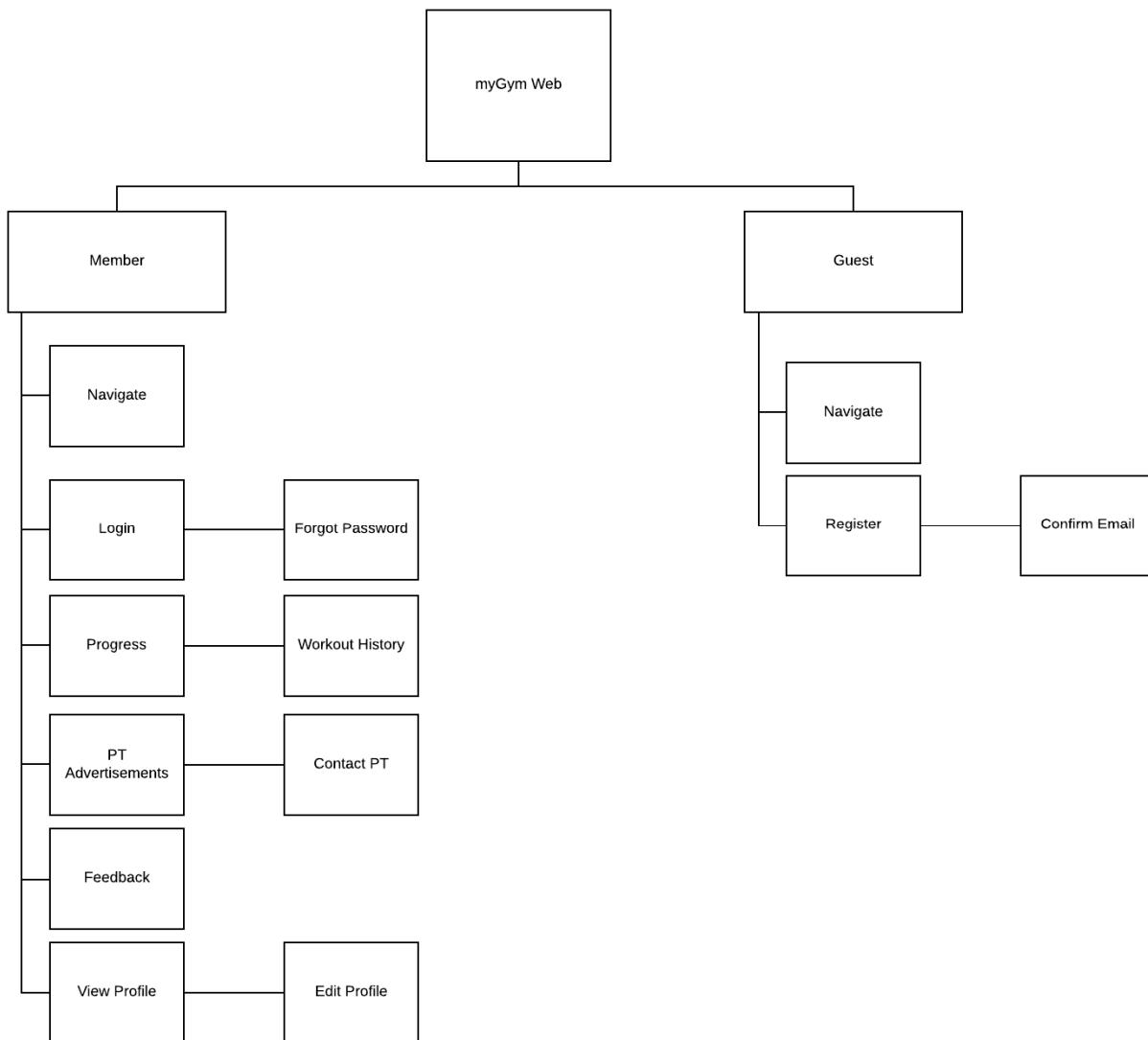
## Mobile Application Decomposition Diagram



## Desktop Application Decomposition Diagram



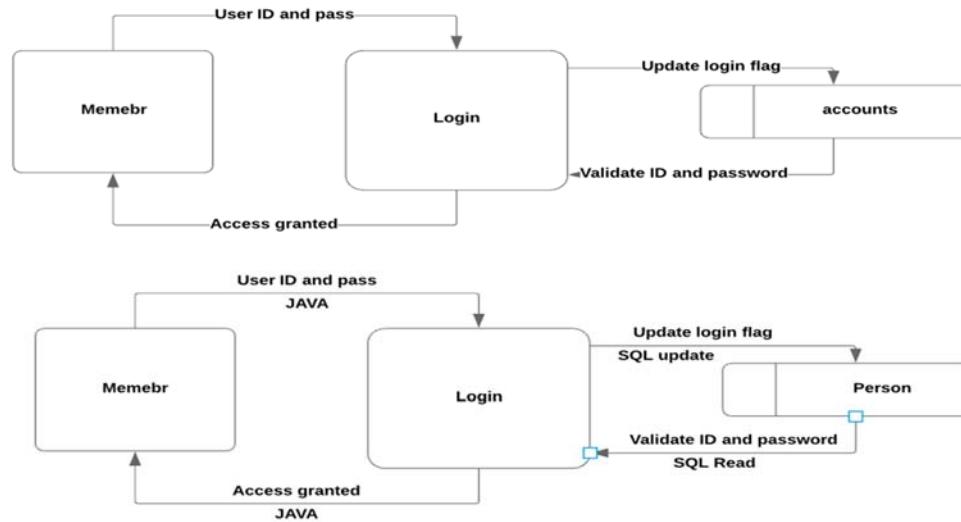
## Web Application Decomposition Diagram



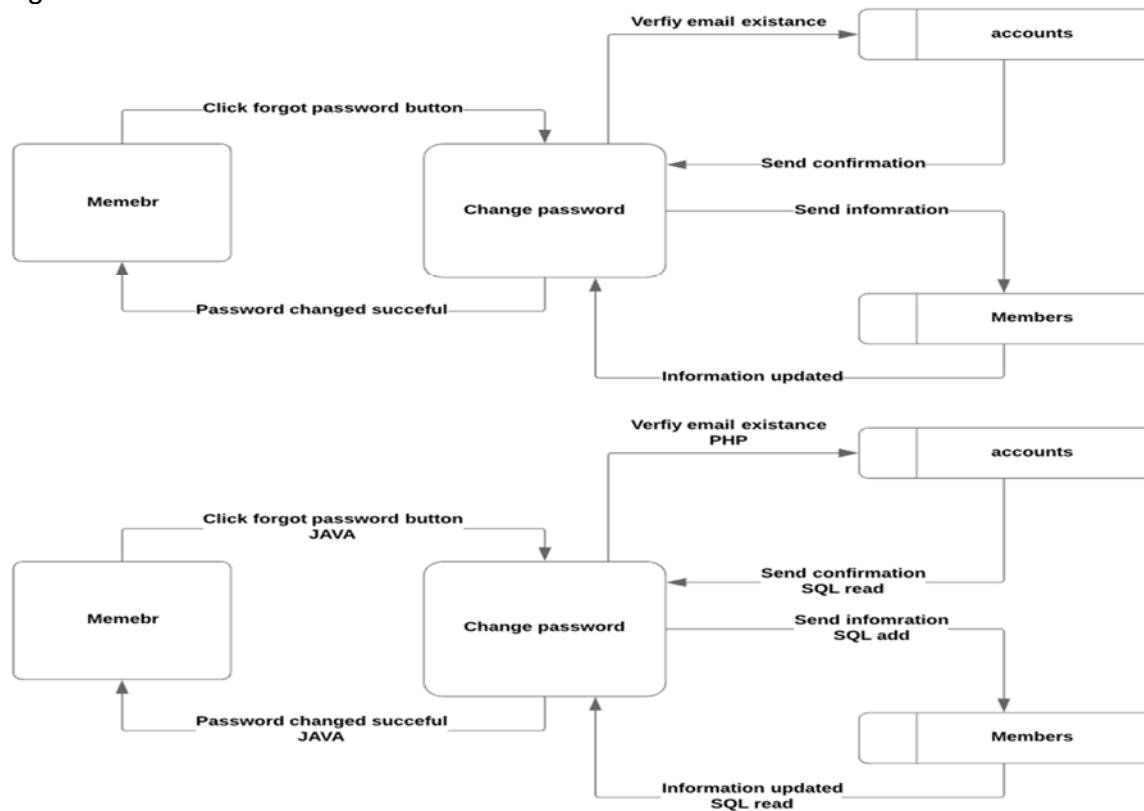
# Mobile Single Logical and Physical DFDs

Actor: Member

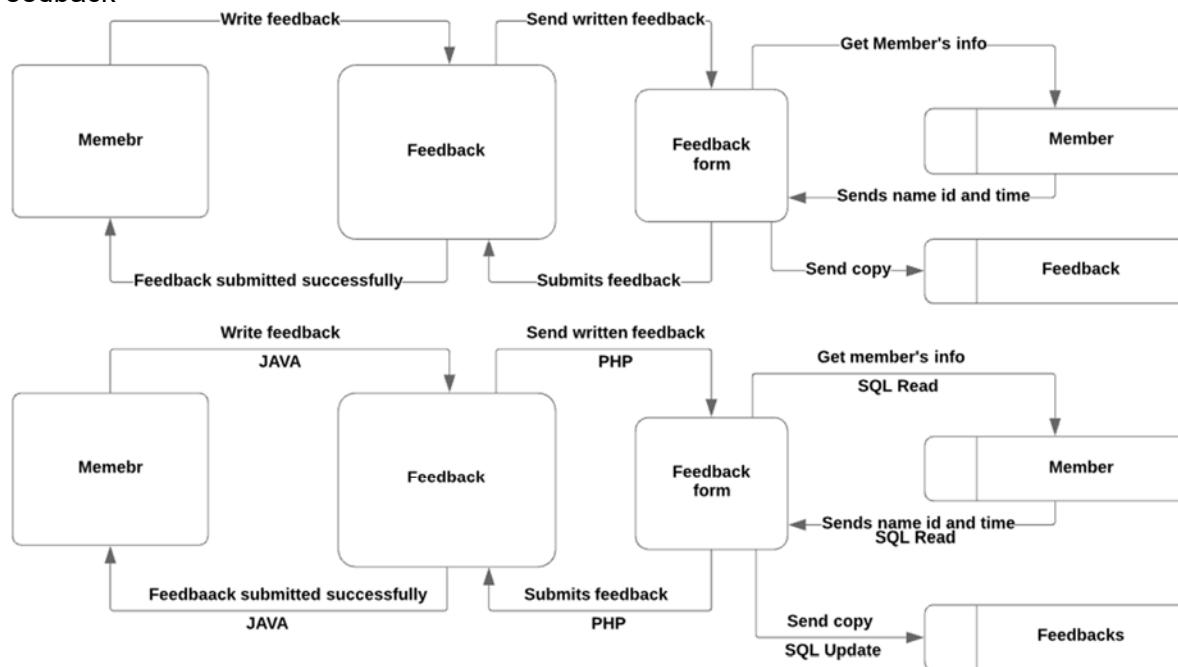
Login



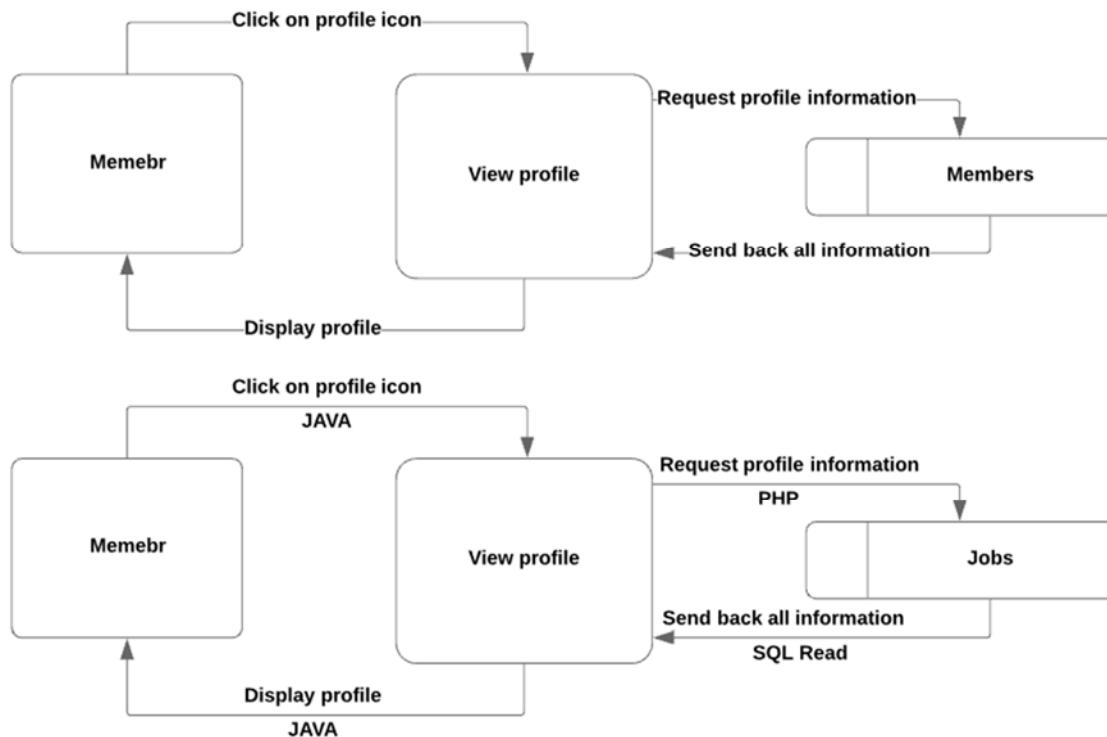
Forgot Password



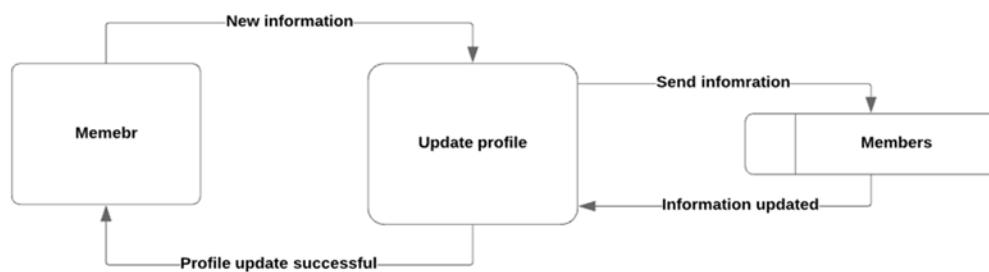
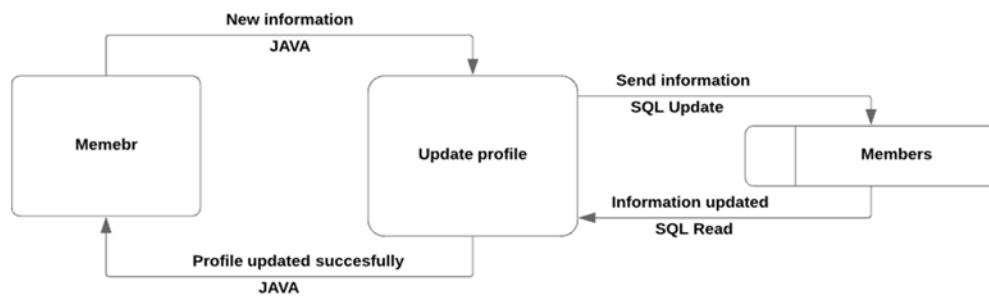
## Feedback



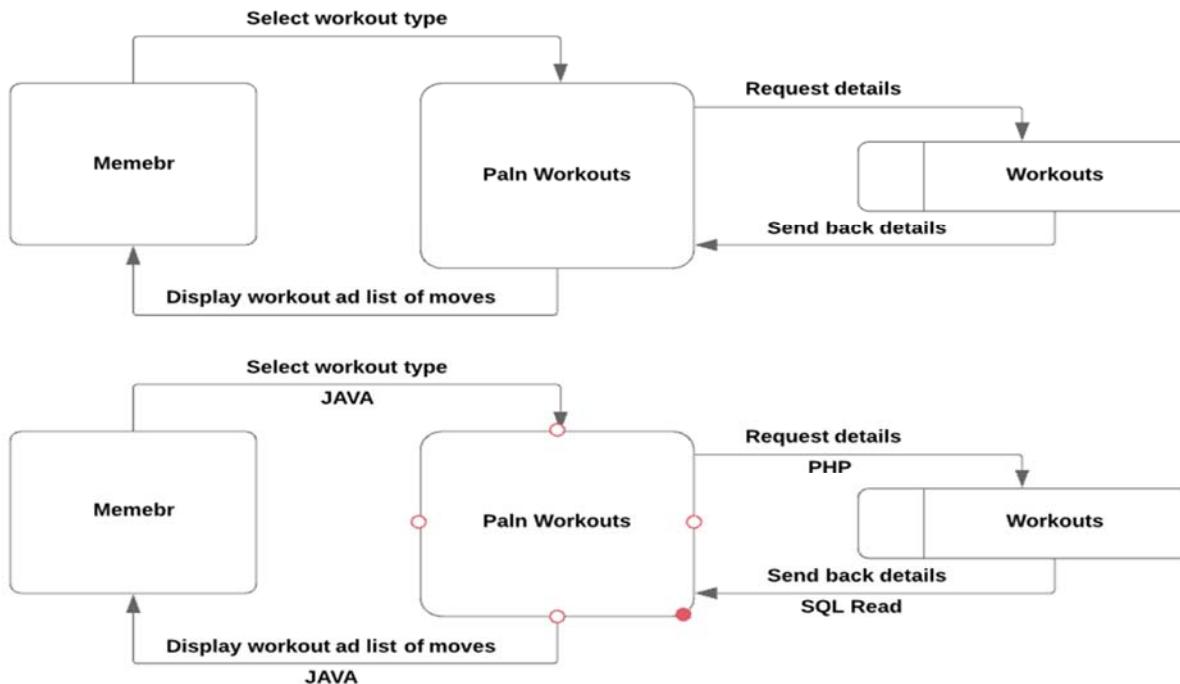
## View Profile



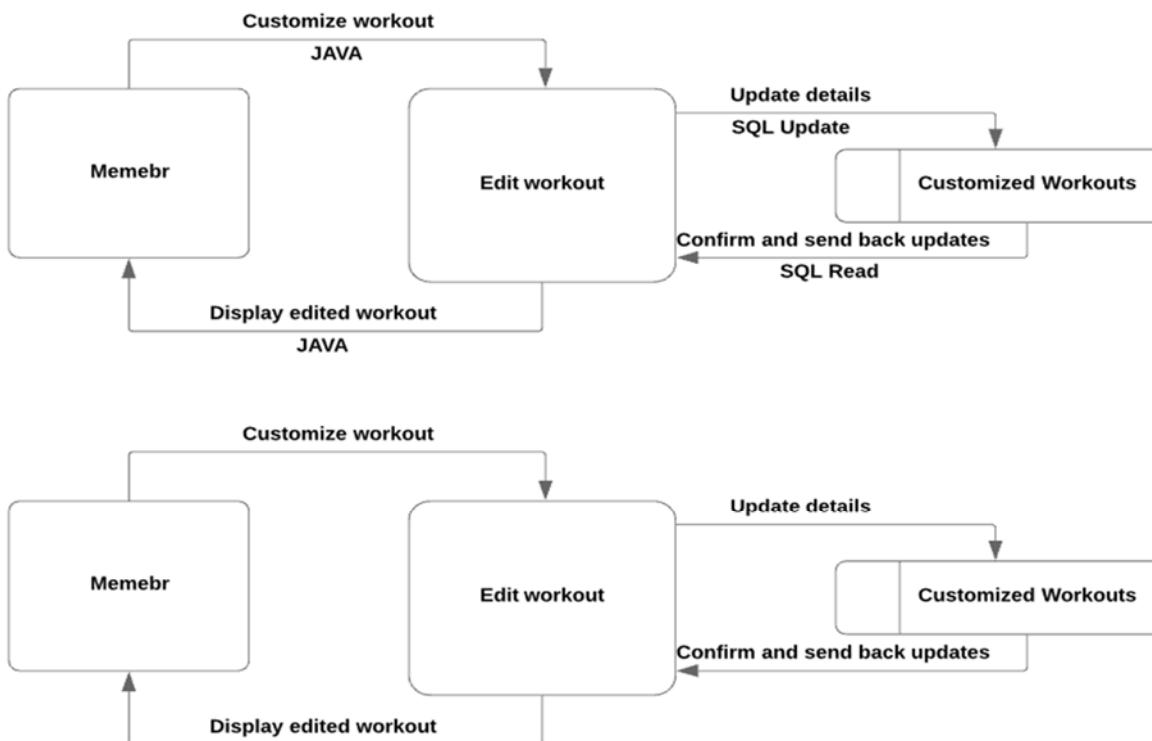
## Edit Profile



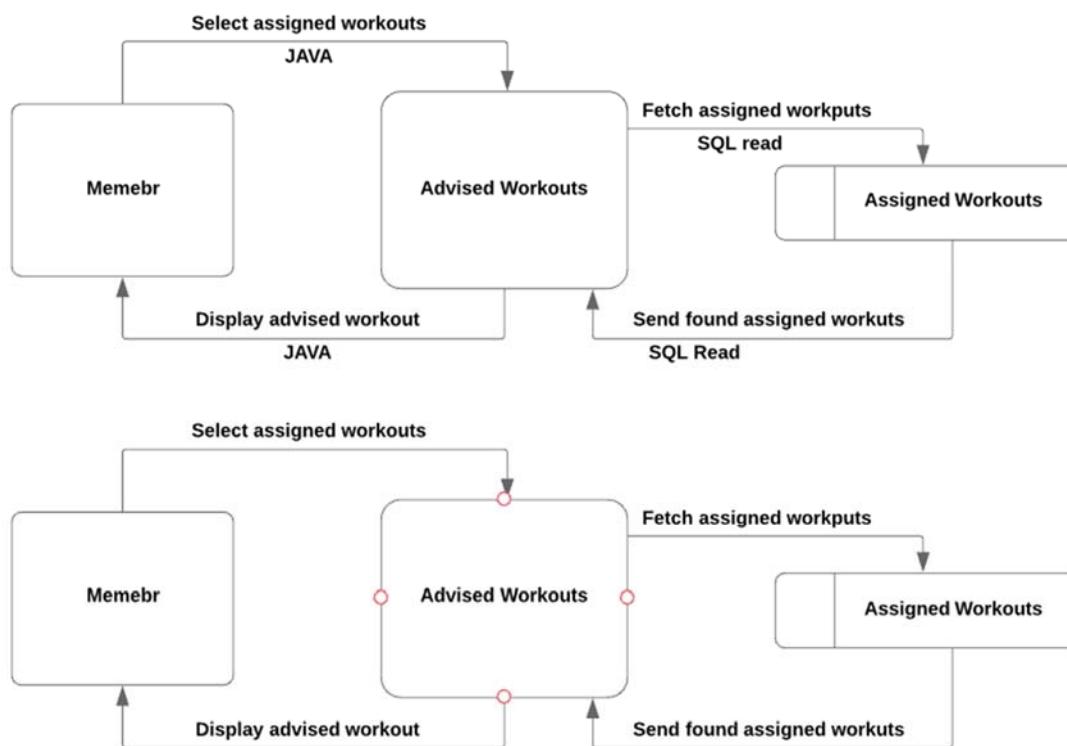
## Plan Workouts



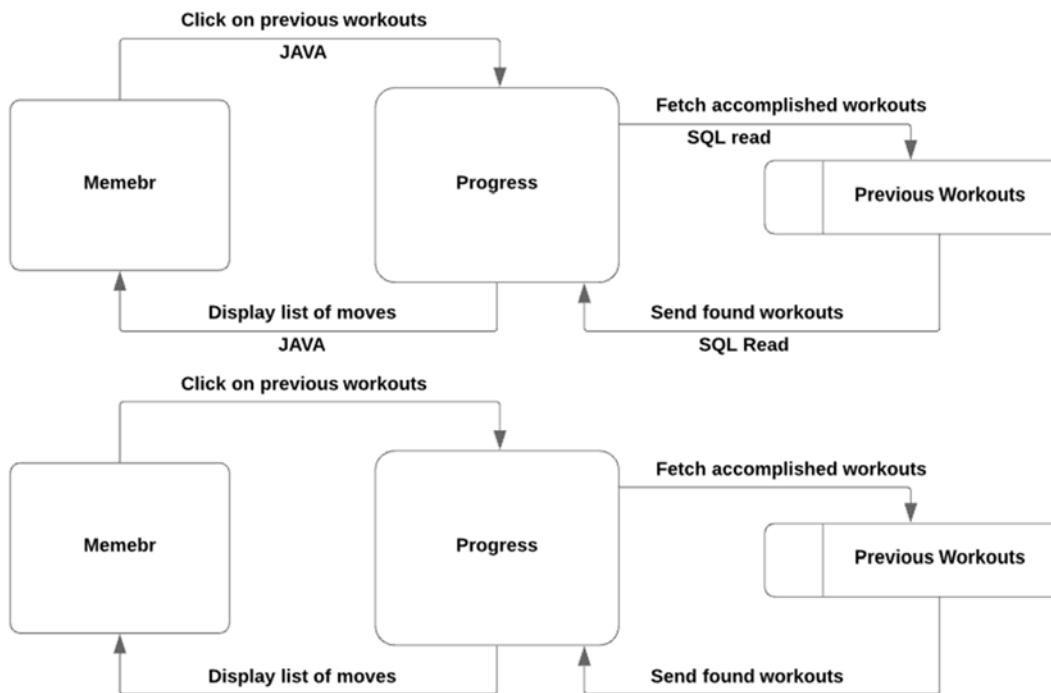
## Edit Workout



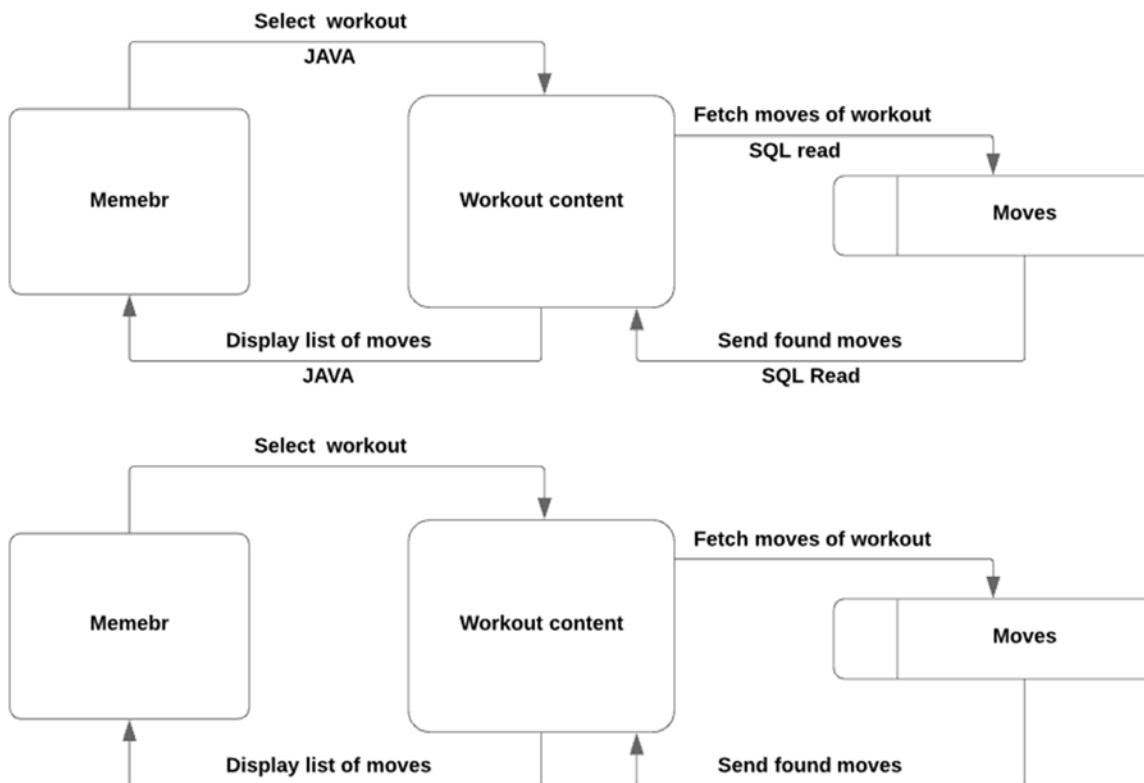
## Advised Workout



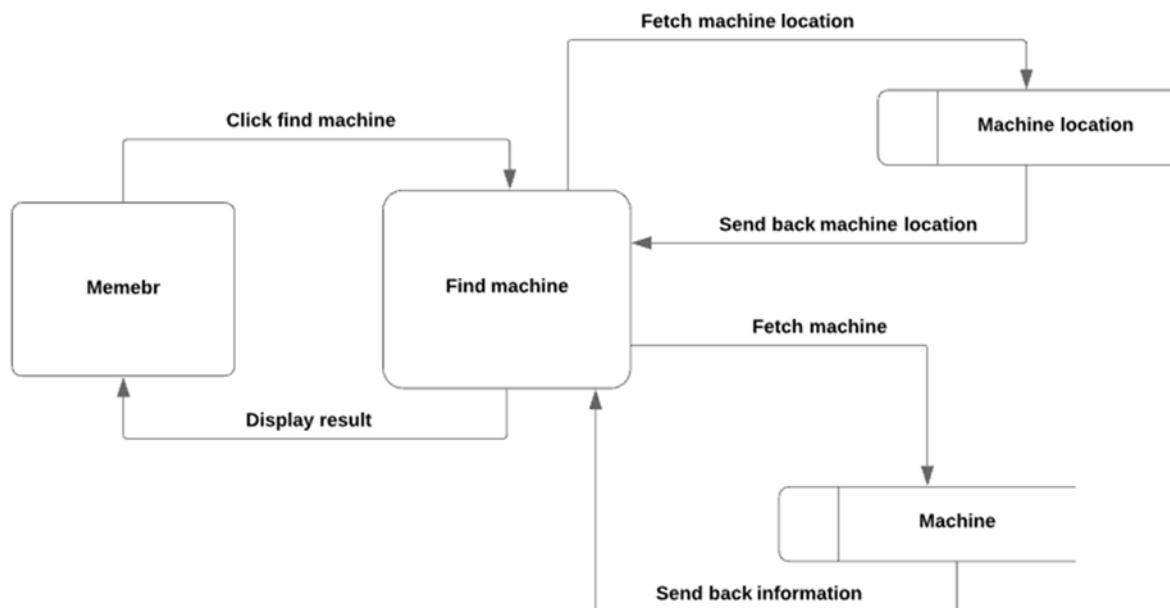
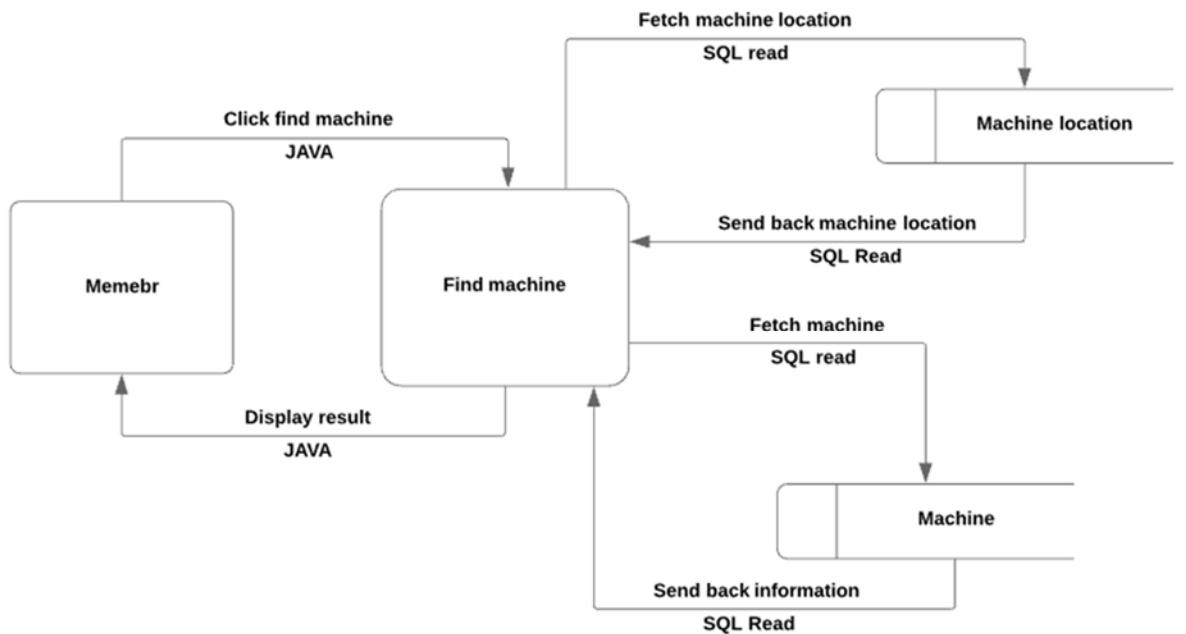
## Progress (Workout History)



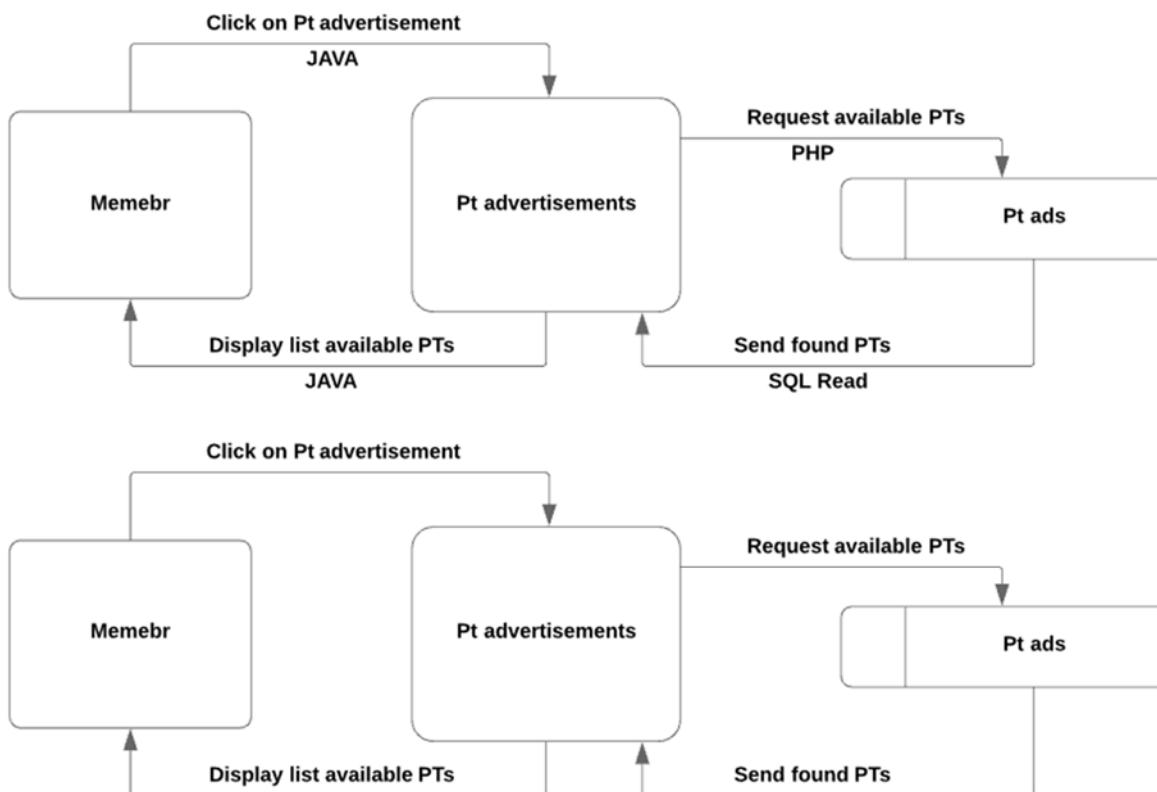
## Workout Content



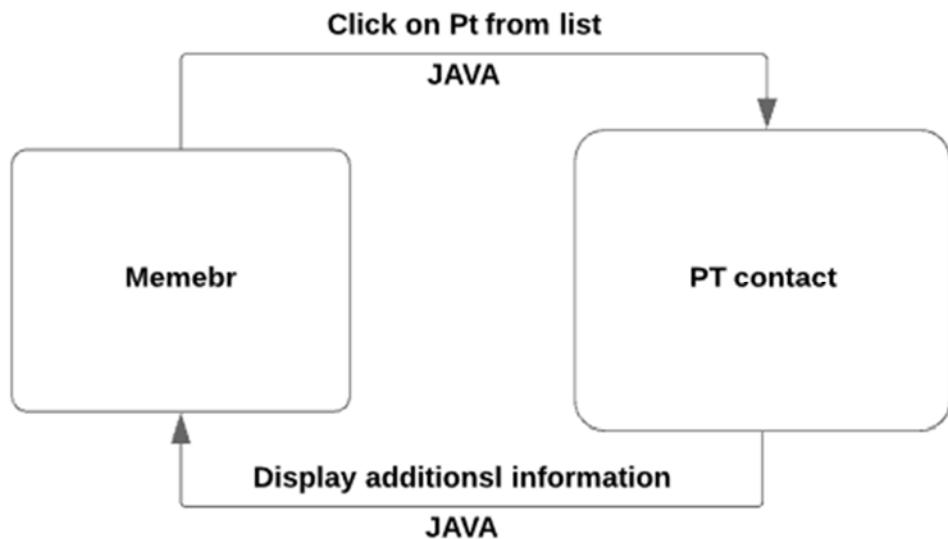
## Find Machine



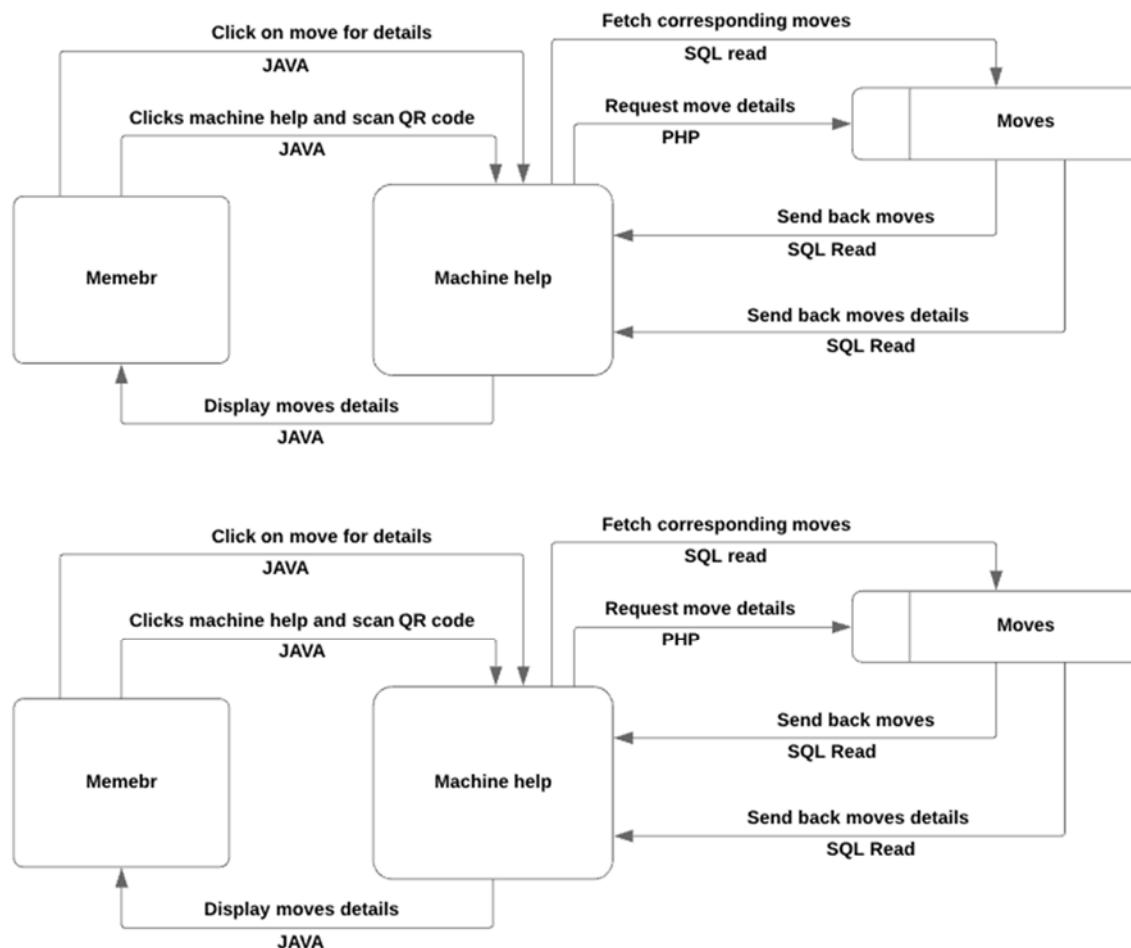
## PT Advertisements



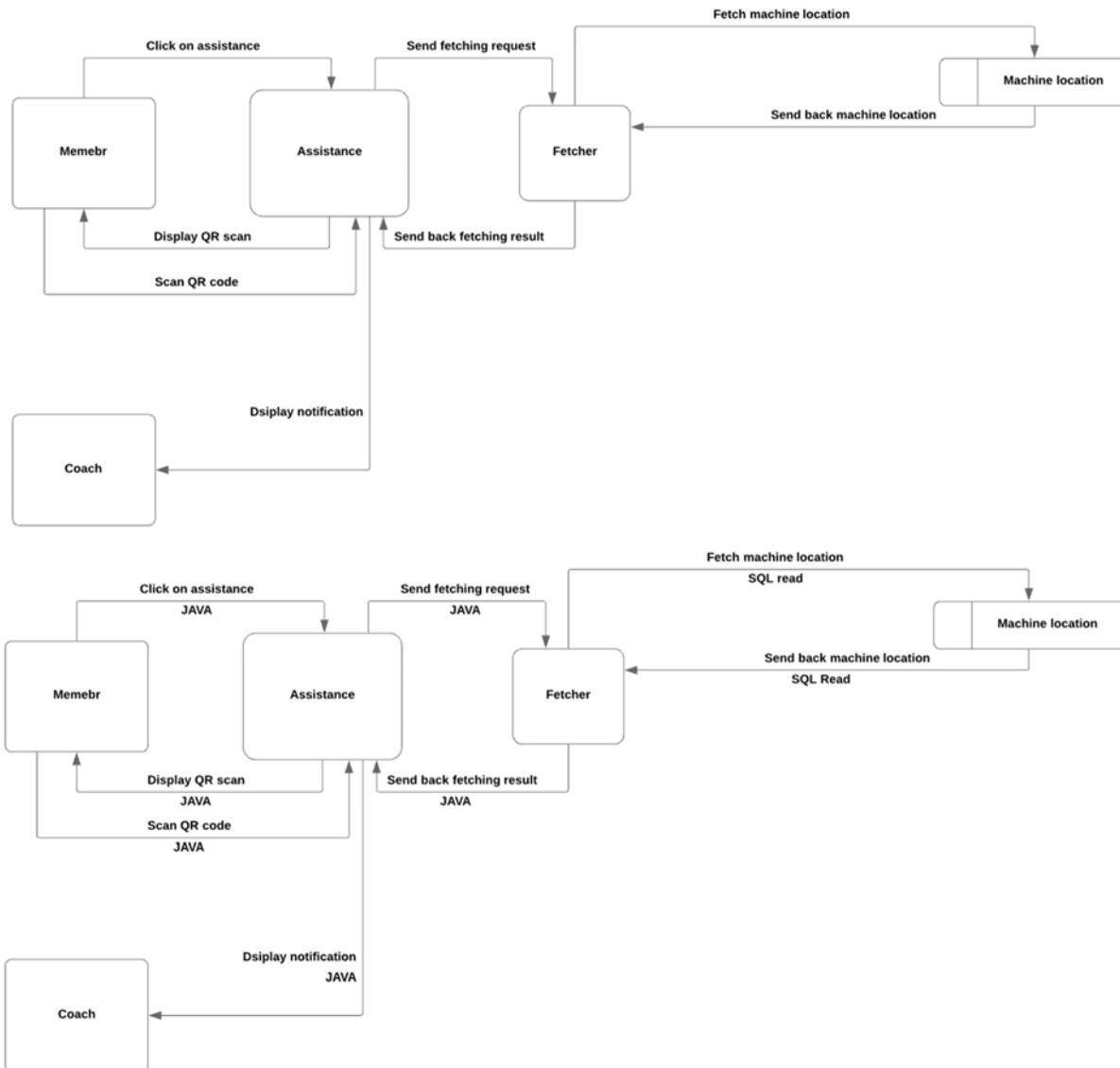
## Contact PT



## Machine Help

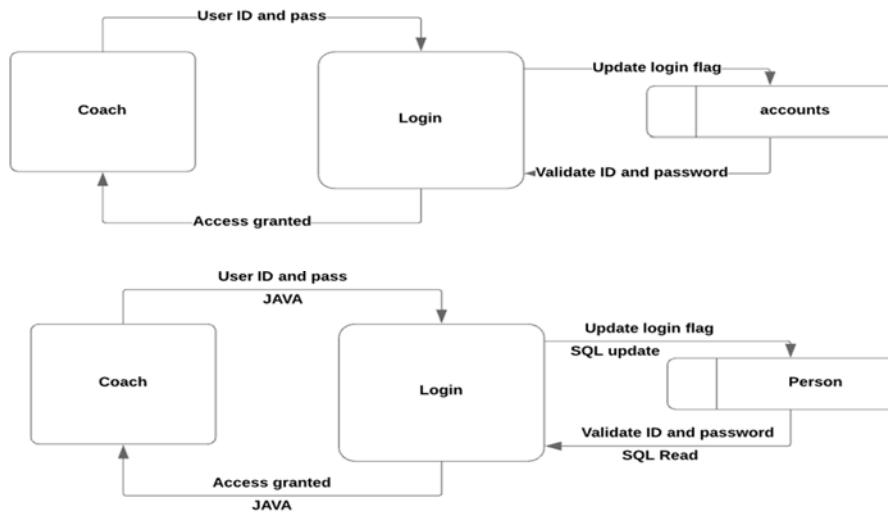


## Assistance

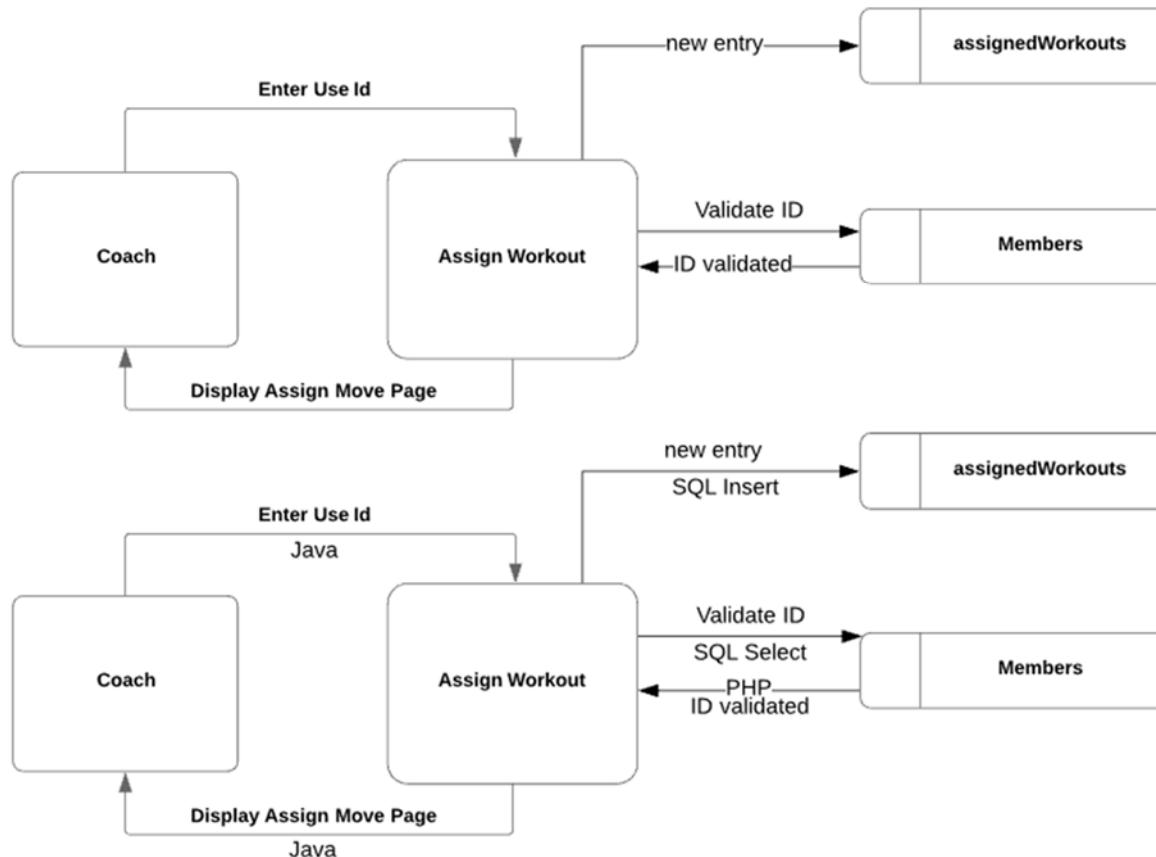


## Actor: Coach

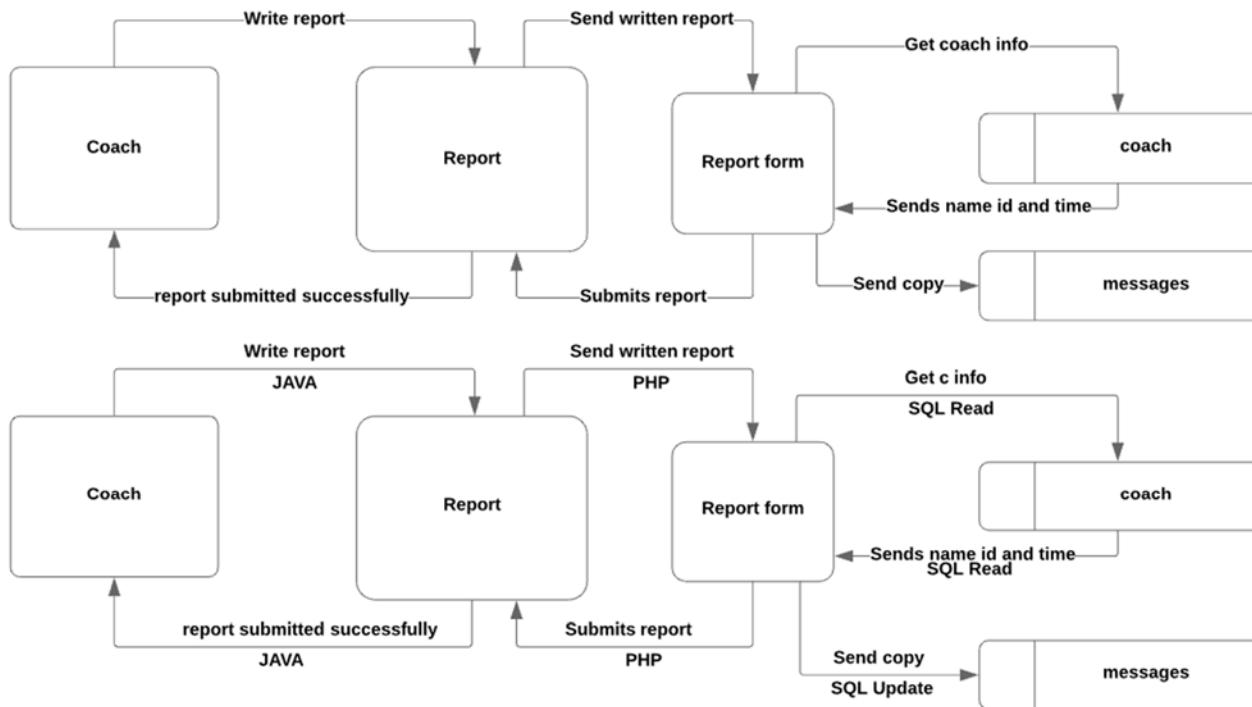
### Login



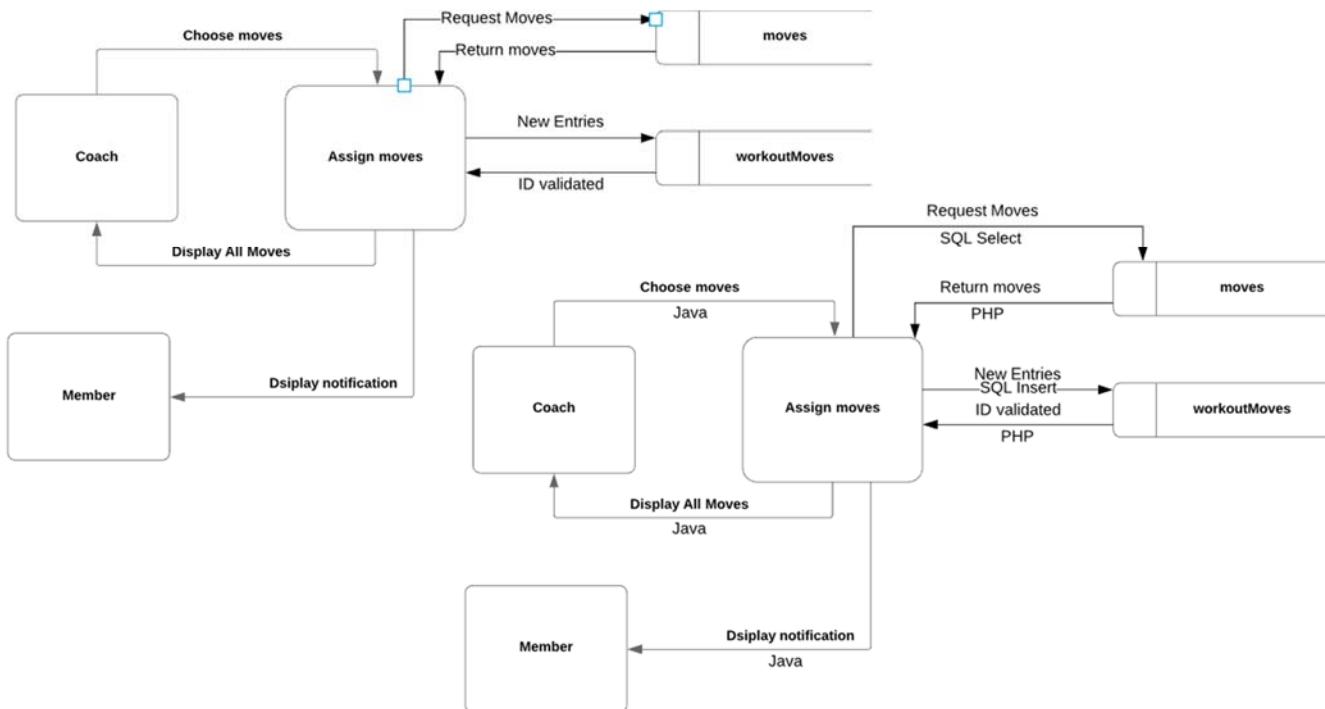
### Assign Workout



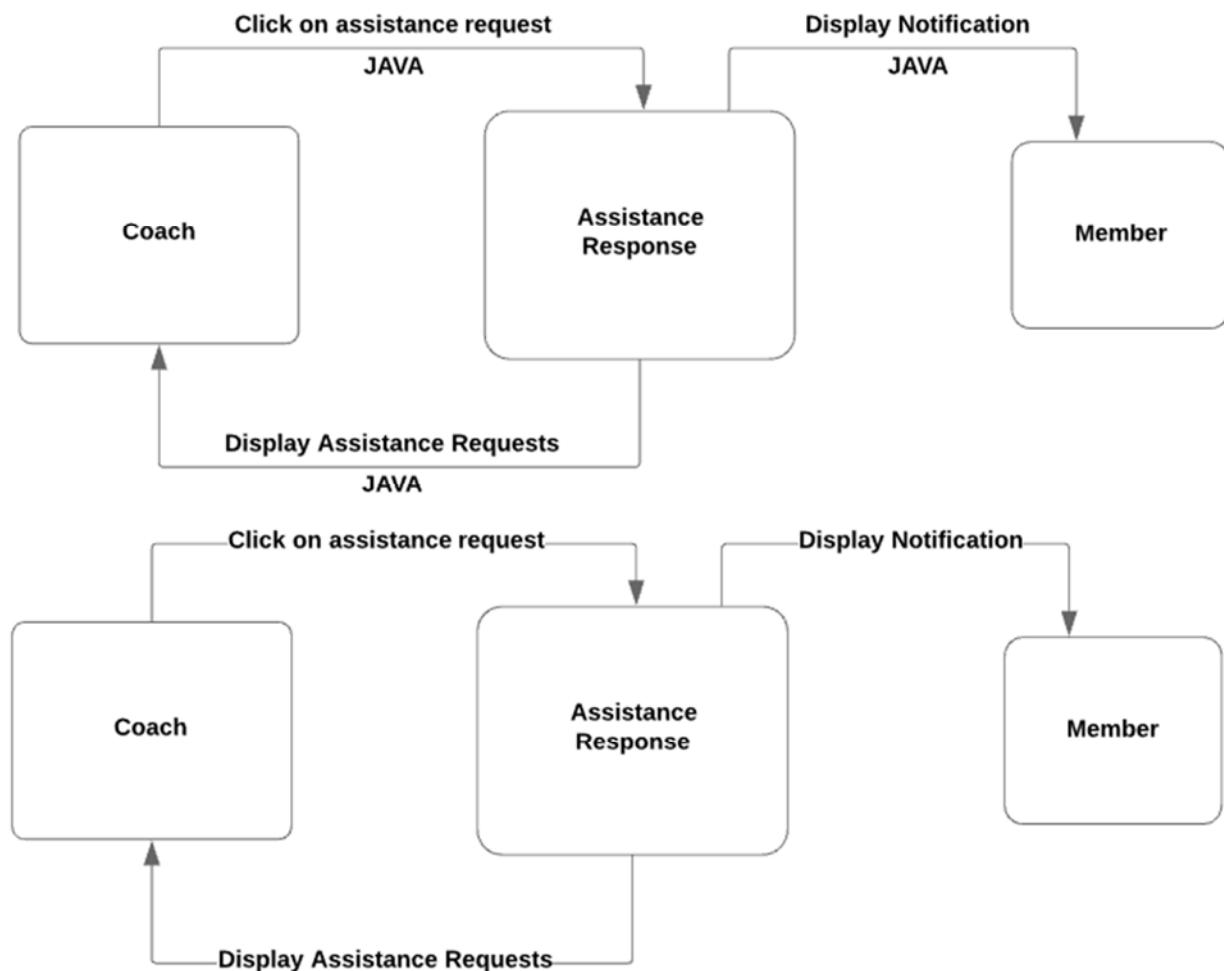
## Assign Moves



## Report



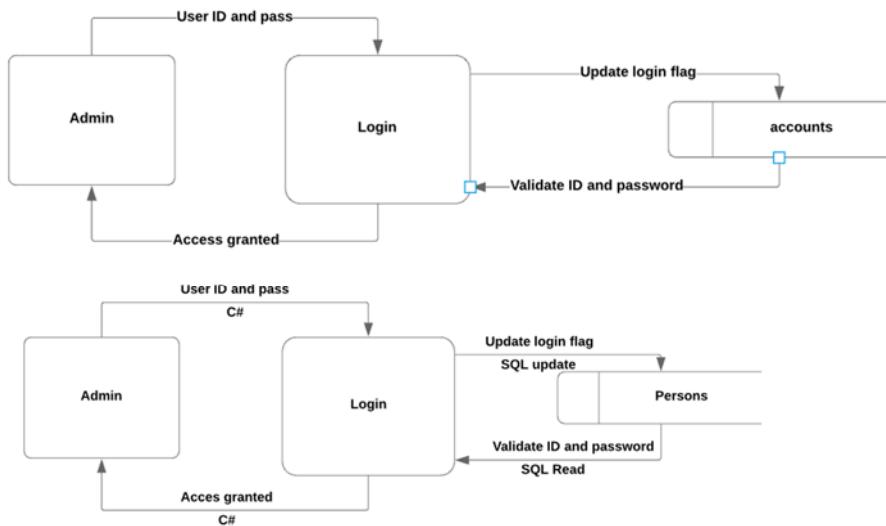
## Assistance Response



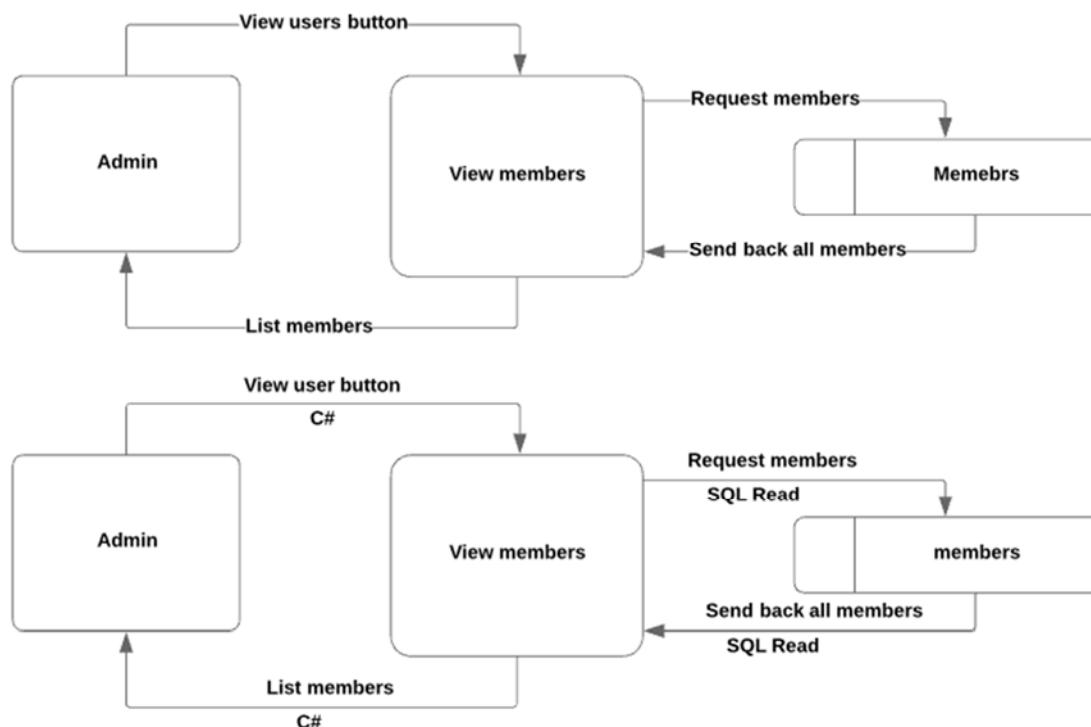
# Desktop Single Logical and Physical DFDs

Actor: Admin

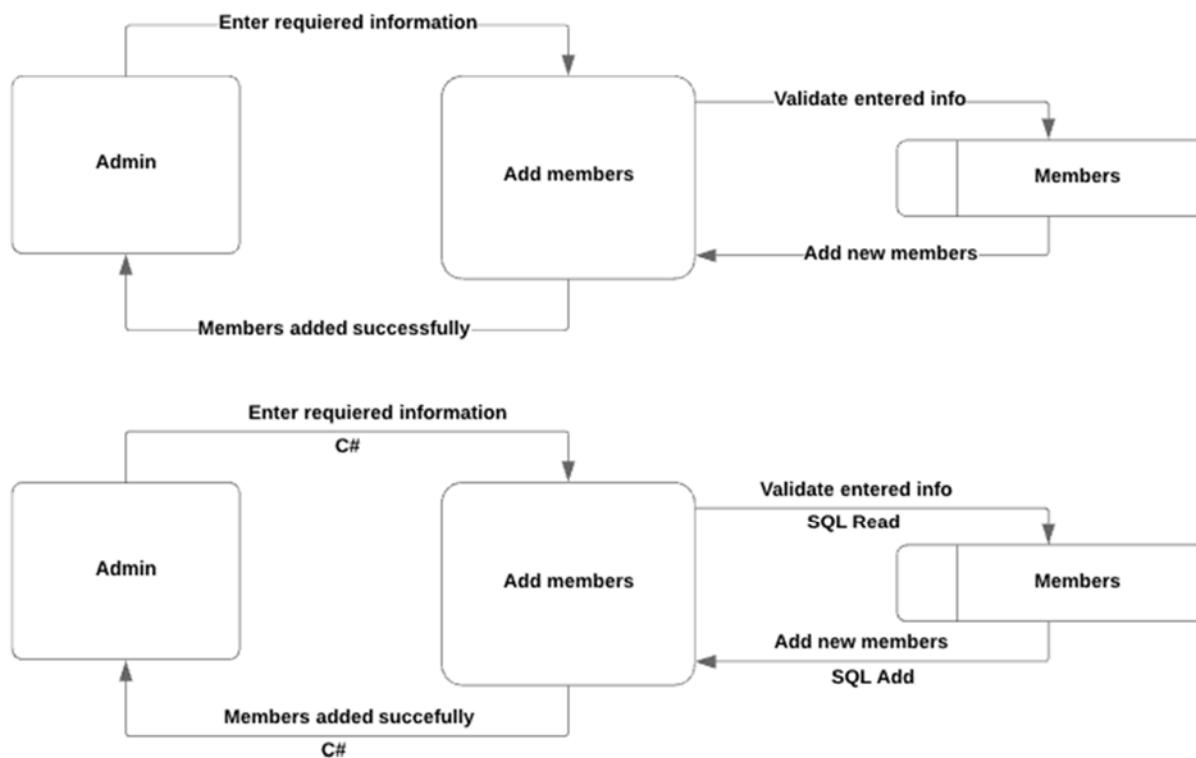
Login



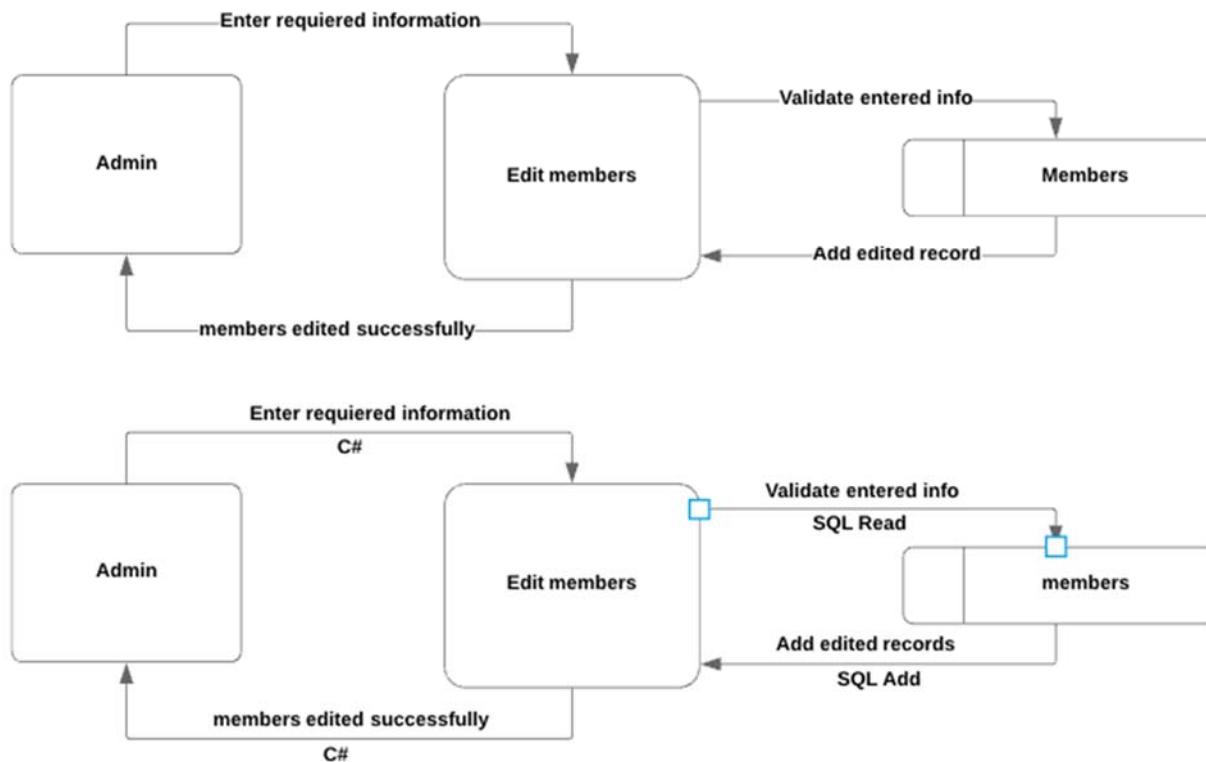
View Members



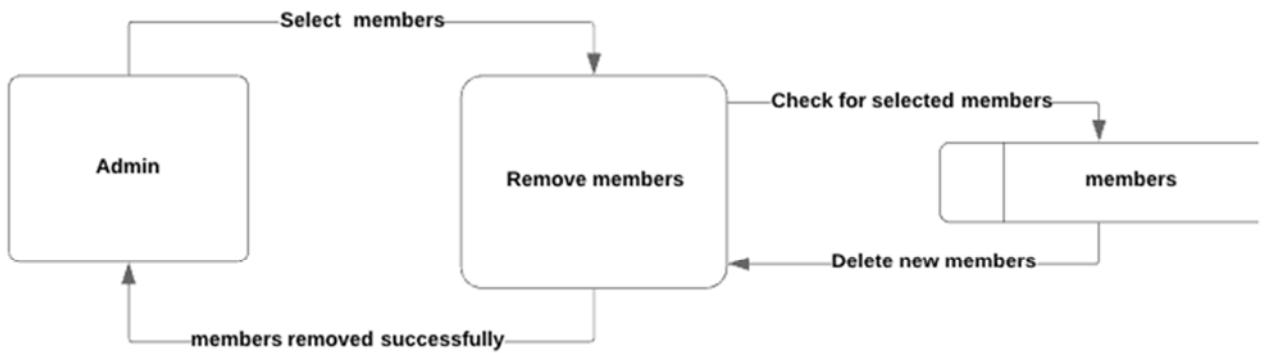
## Add Members



## Edit Members



## Remove Members



Select members

C#

Check for selected members

SQL Read

members

Delete new members

SQL Delete

Admin

Remove members

members removed successfully

C#

## View Coaches

View users button

Admin

View coach

Request coach

coaches

Send back all coach

List coach

View user button

C#

Admin

View coach

Request coach

SQL Read

coaches

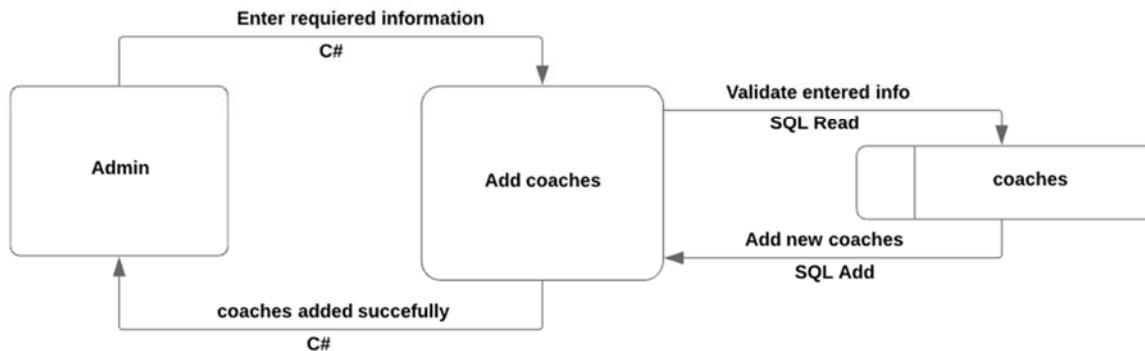
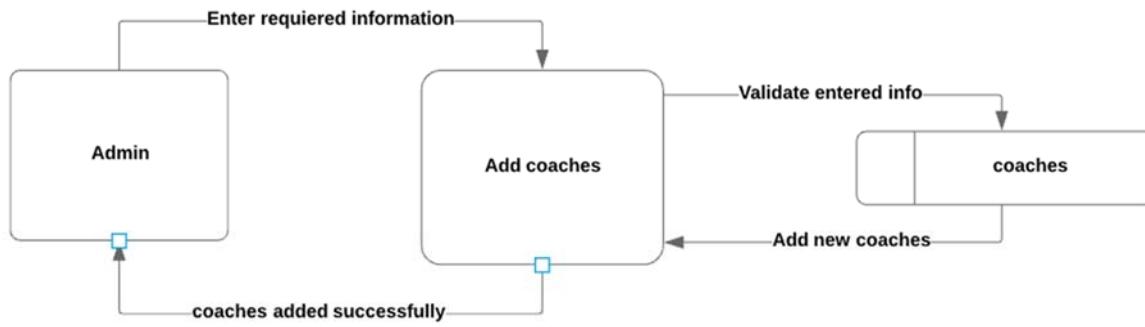
Send back all coach

SQL Read

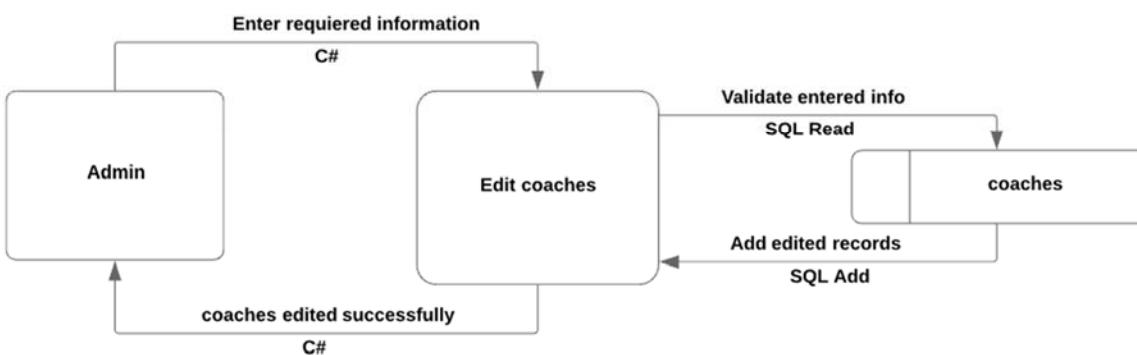
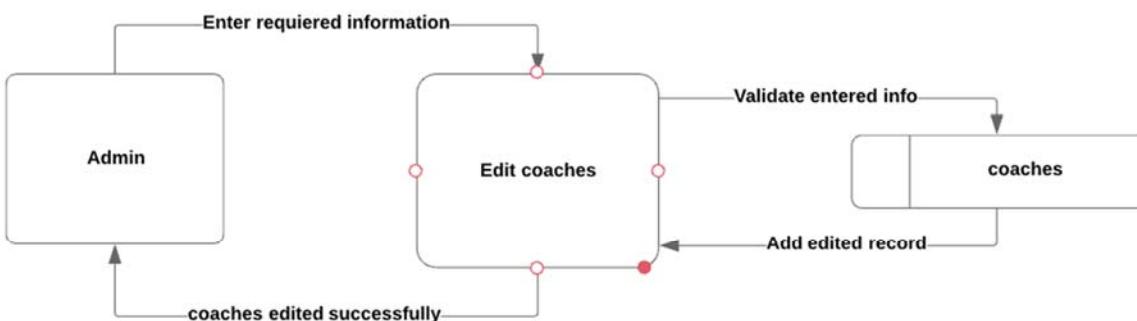
List coach

C#

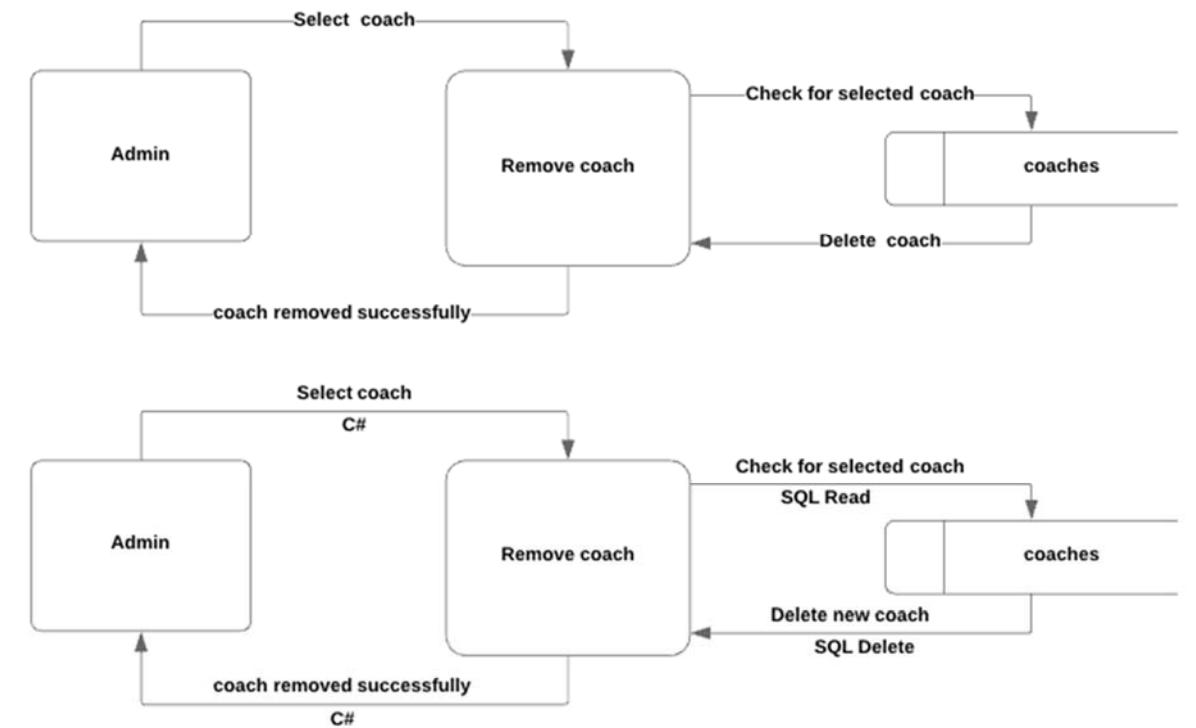
## Add Coaches



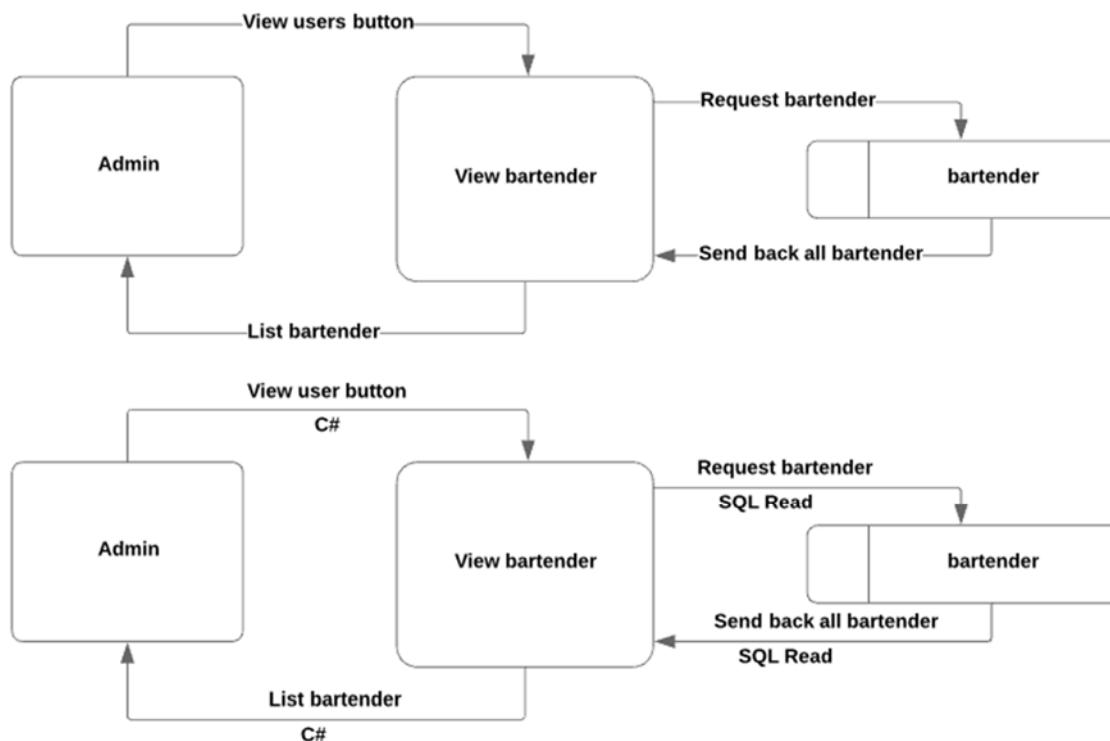
## Edit Coaches



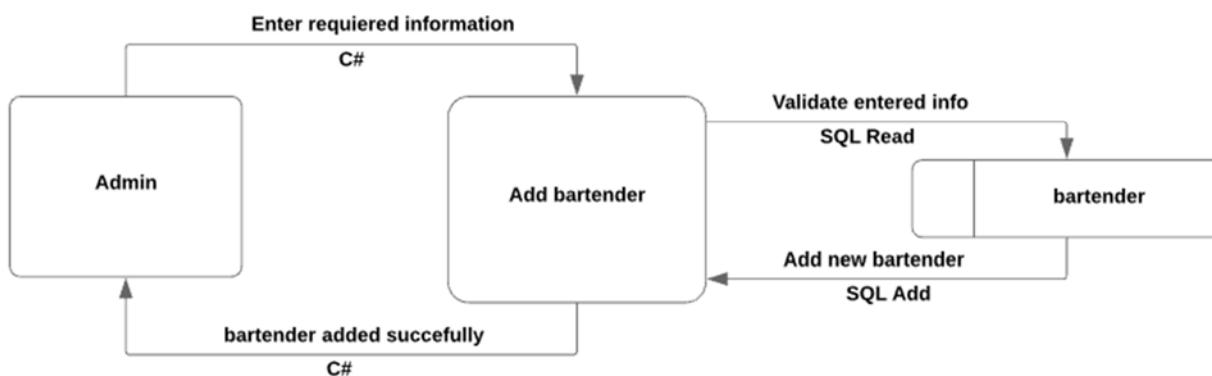
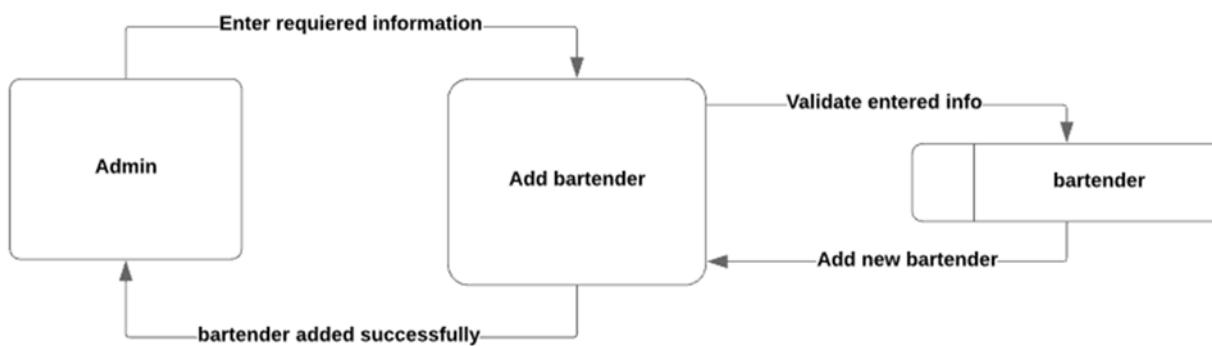
## Remove Coaches



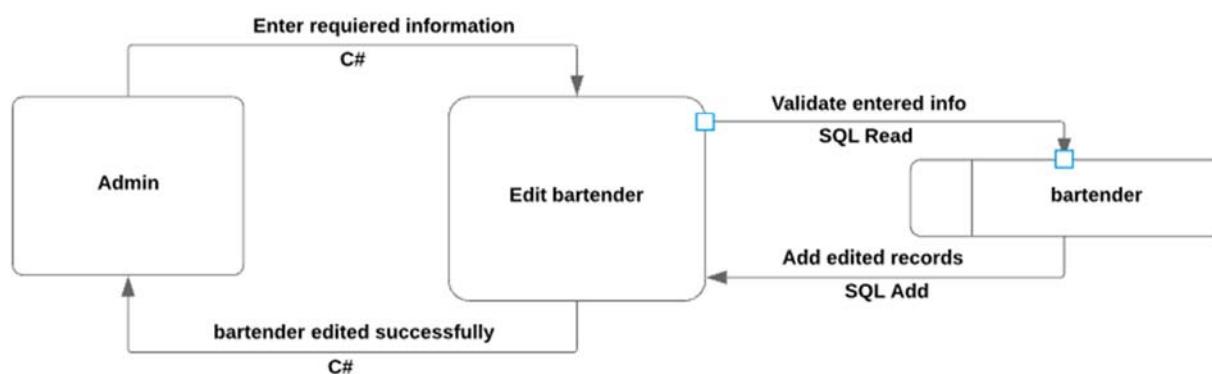
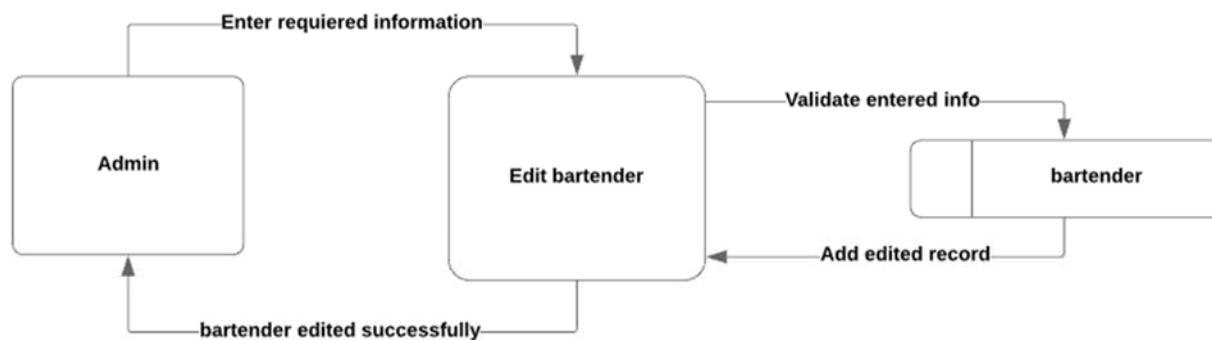
## View Bartenders



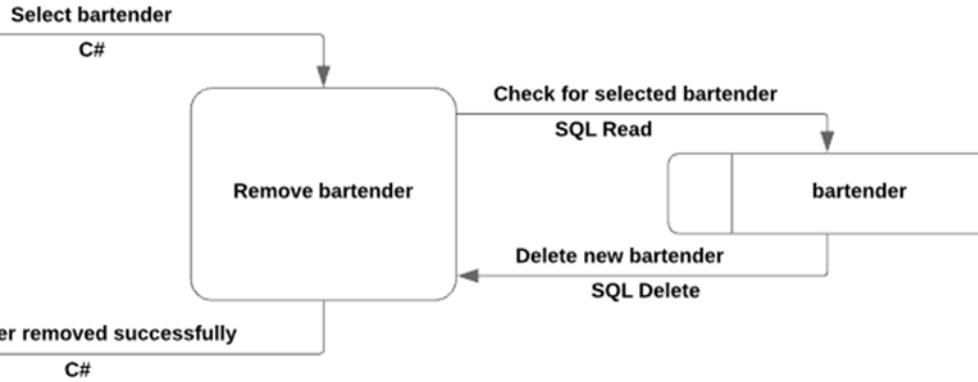
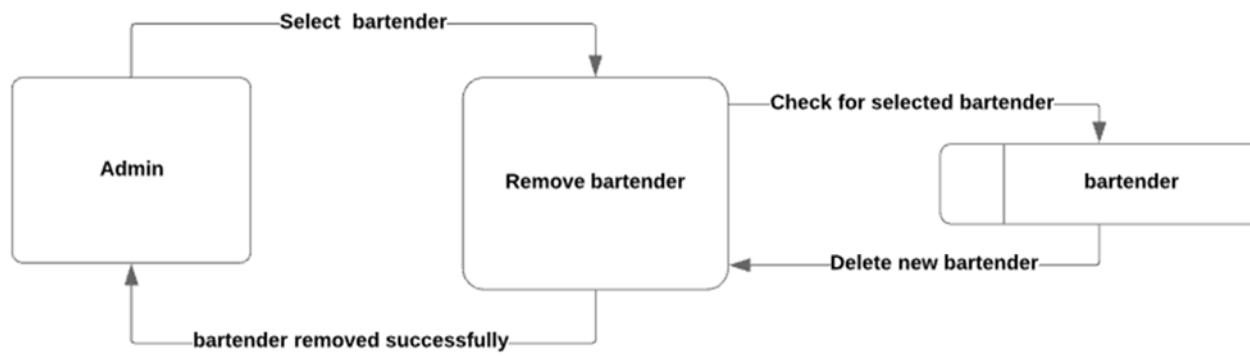
## Add Bartenders



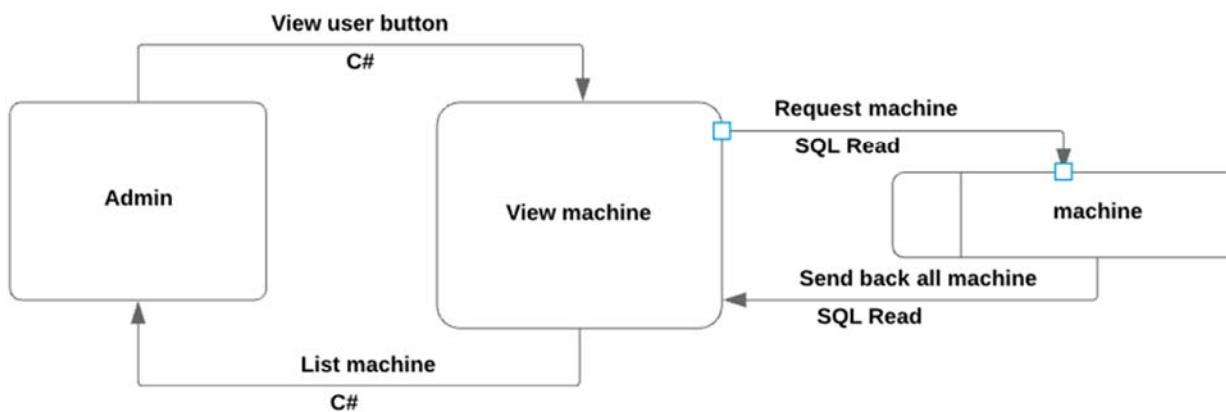
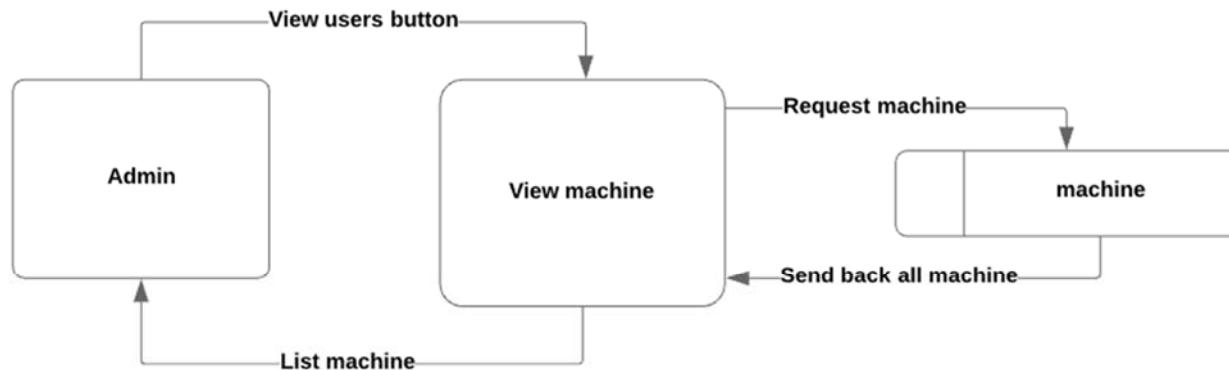
## Edit Bartenders



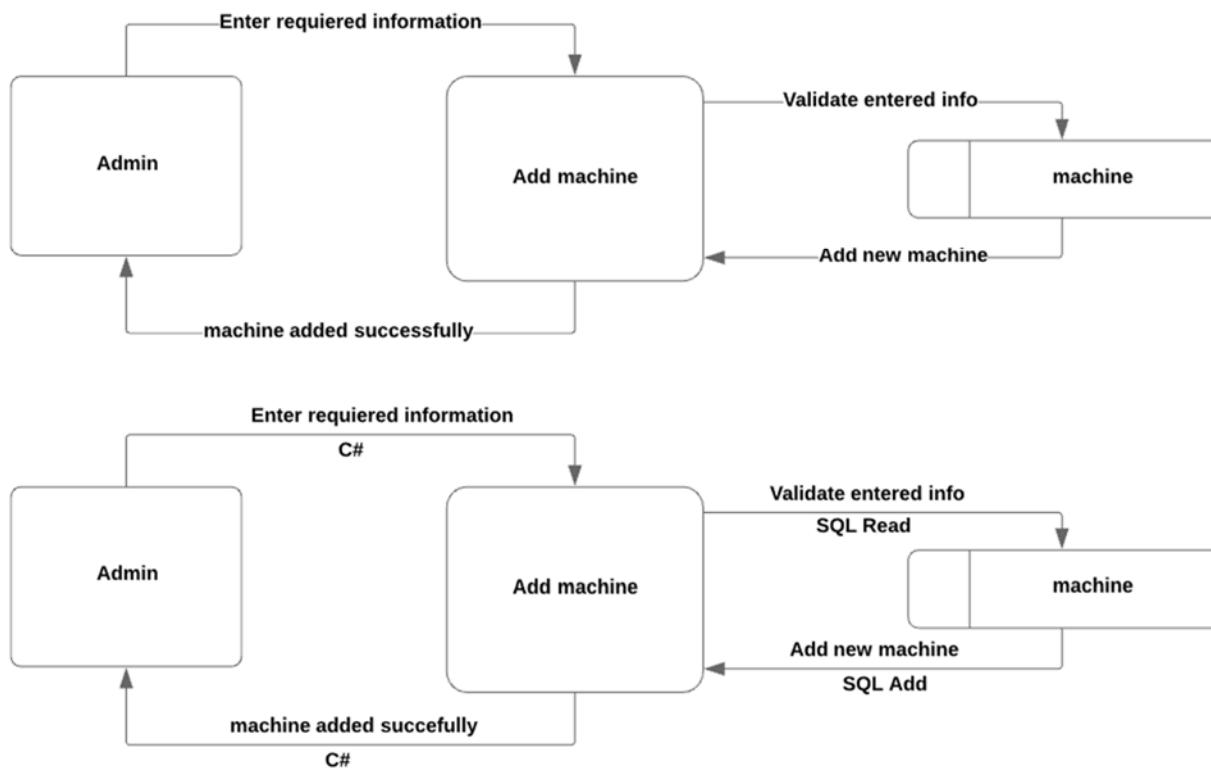
## Remove Bartenders



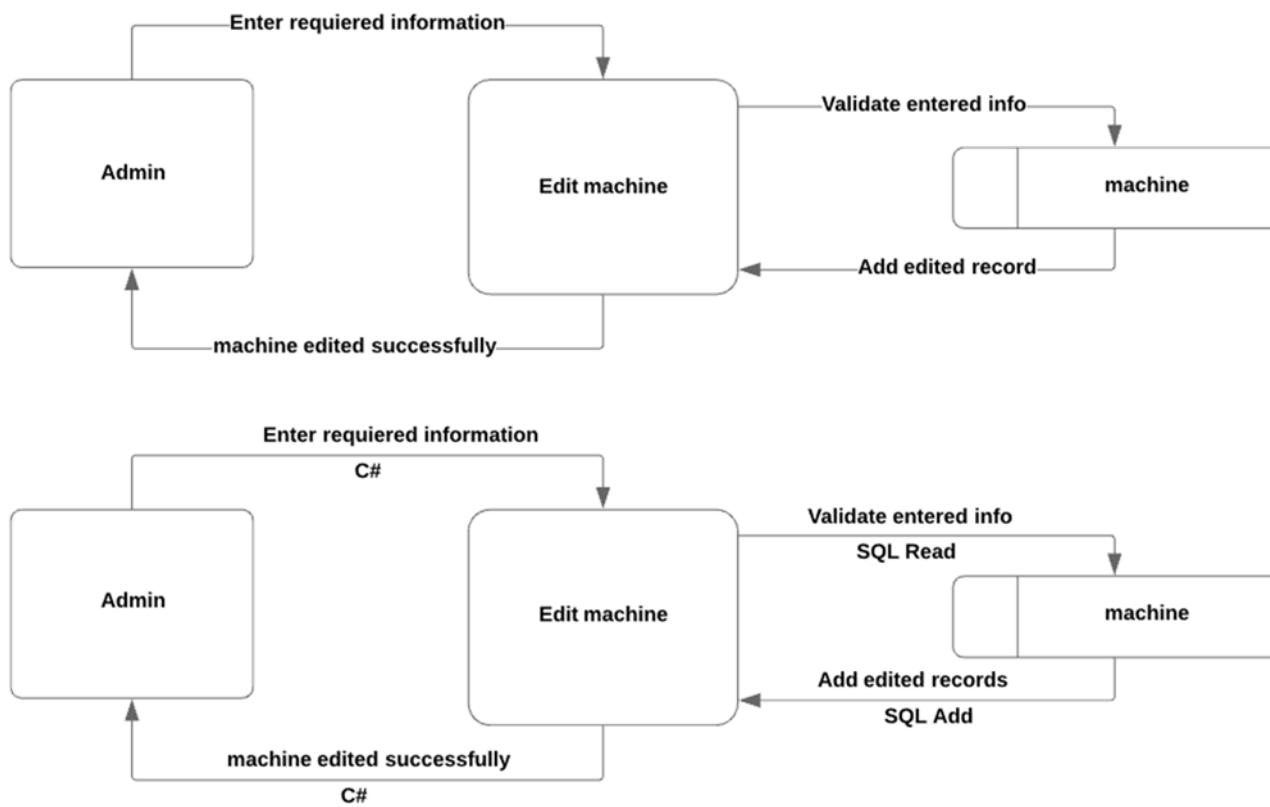
## View Machines



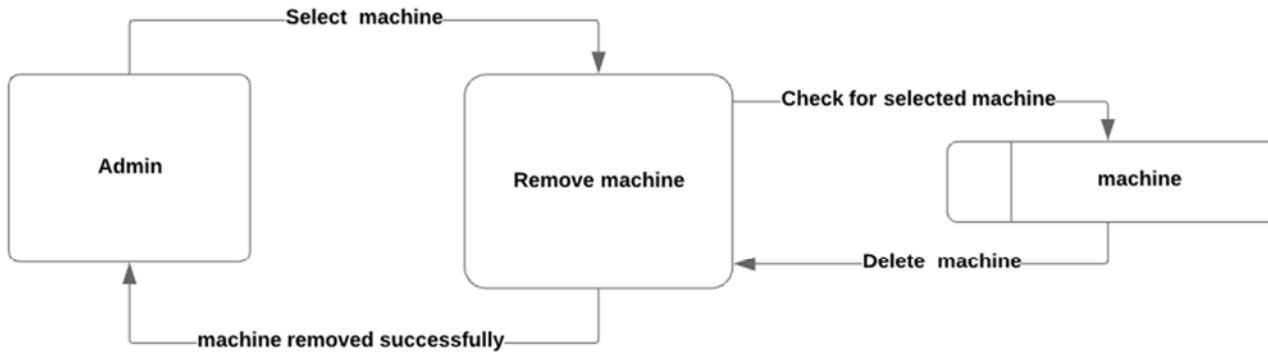
## Add Machines



## Edit Machines



## Remove Machines



Select machine

C#

Check for selected machine

SQL Read

machine

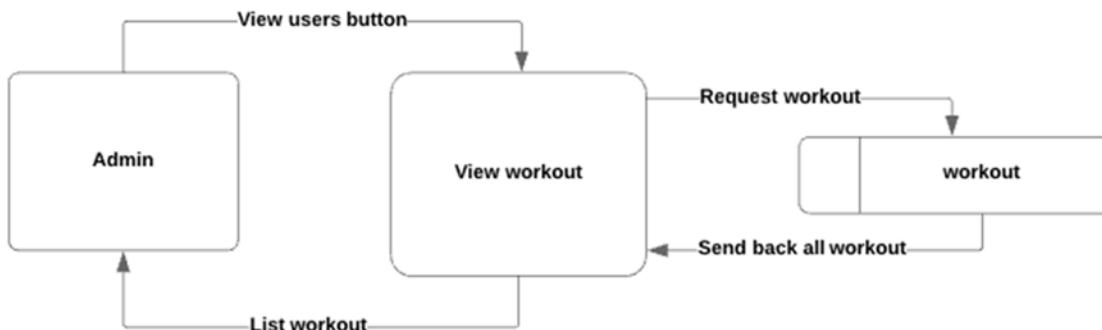
Delete machine

SQL Delete

machine removed successfully

C#

## View Workouts



View user button

C#

Request workout

SQL Read

workout

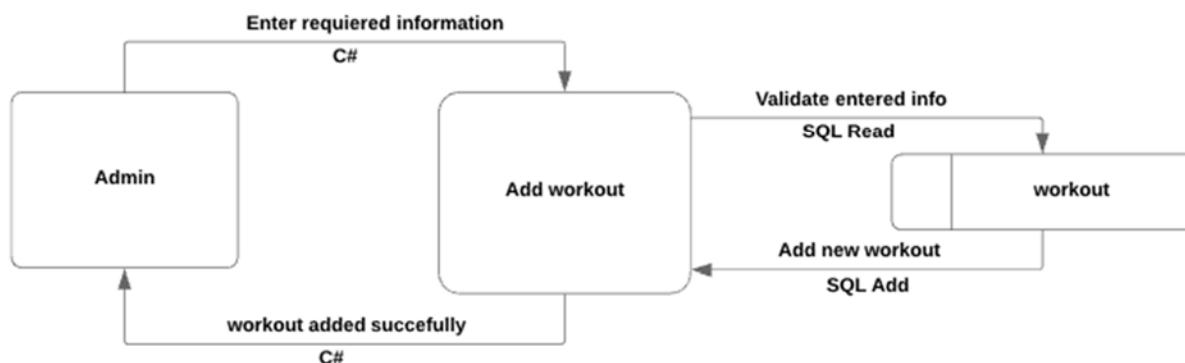
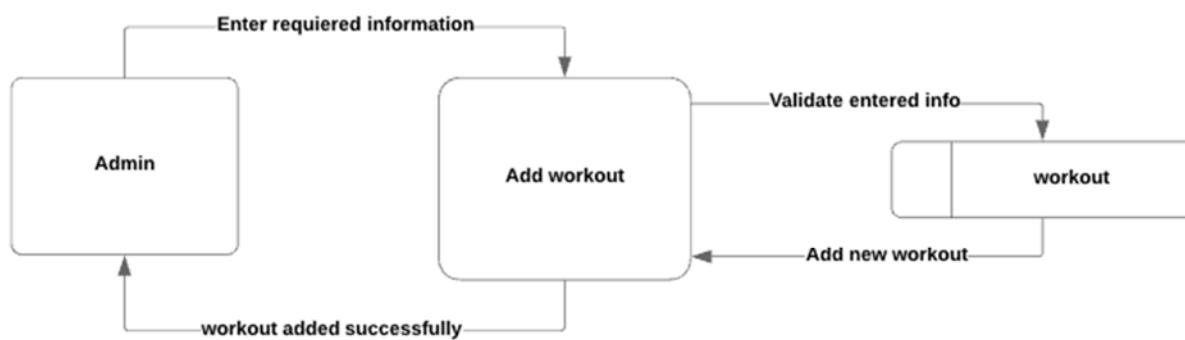
Send back all workout

SQL Read

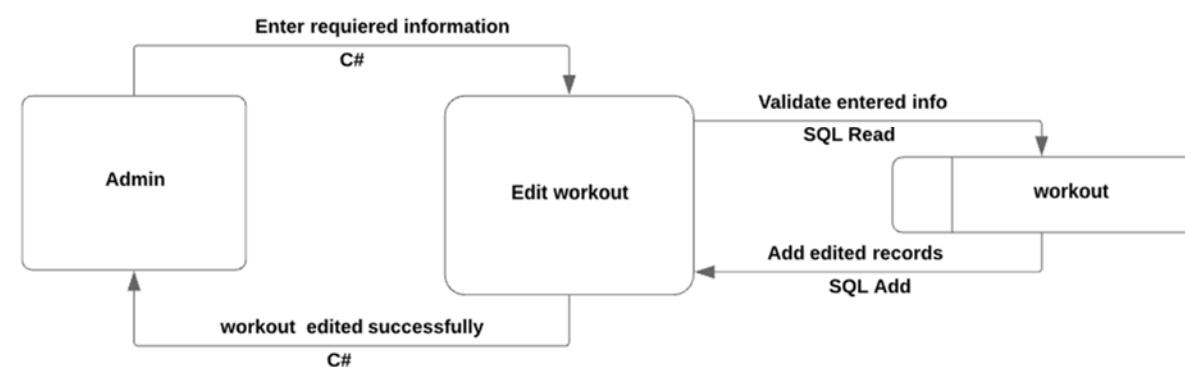
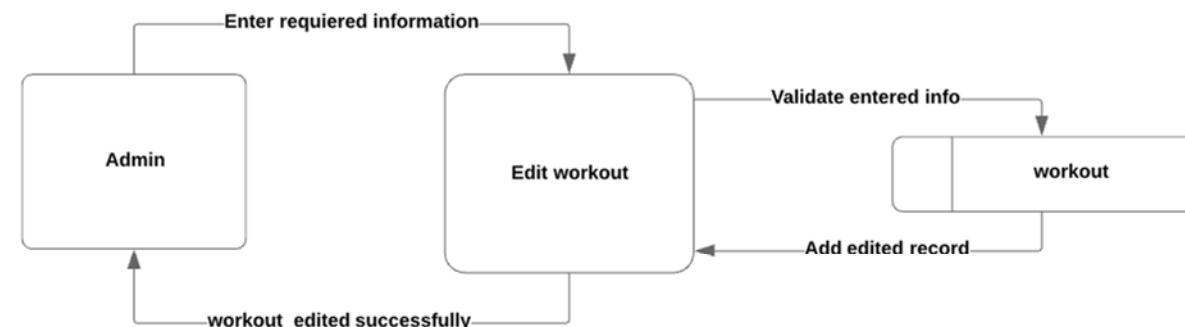
List workout

C#

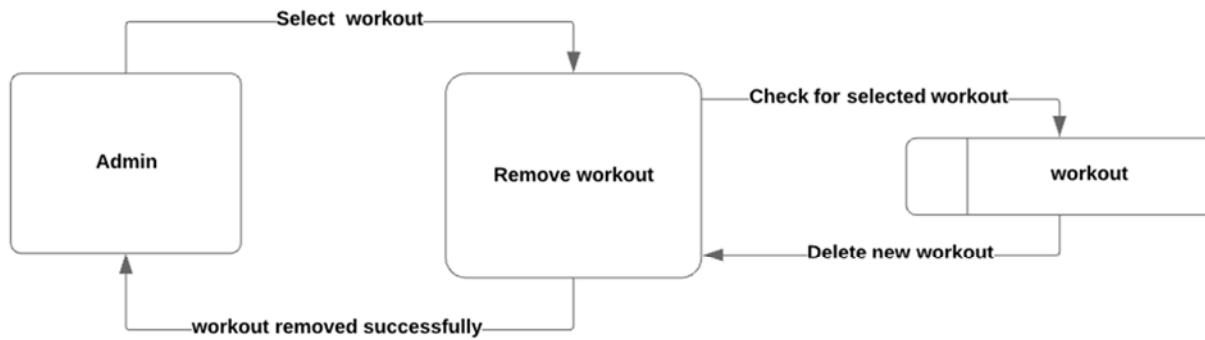
## Add Workouts



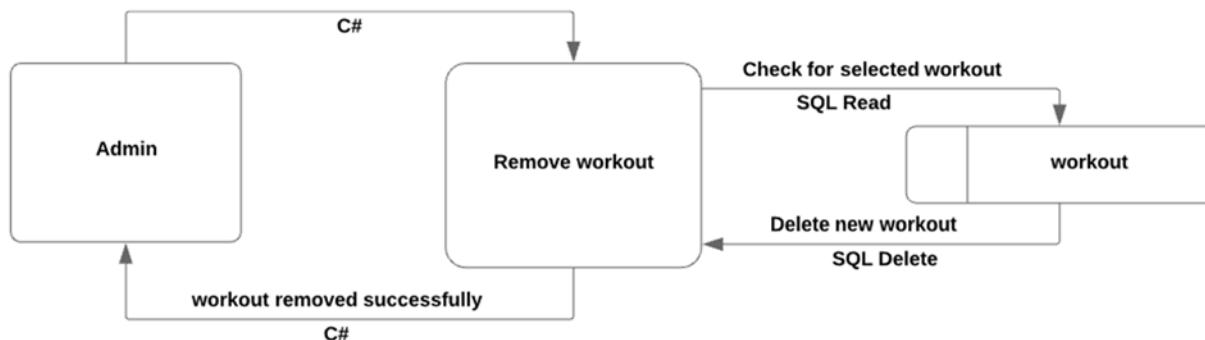
## Edit Workouts



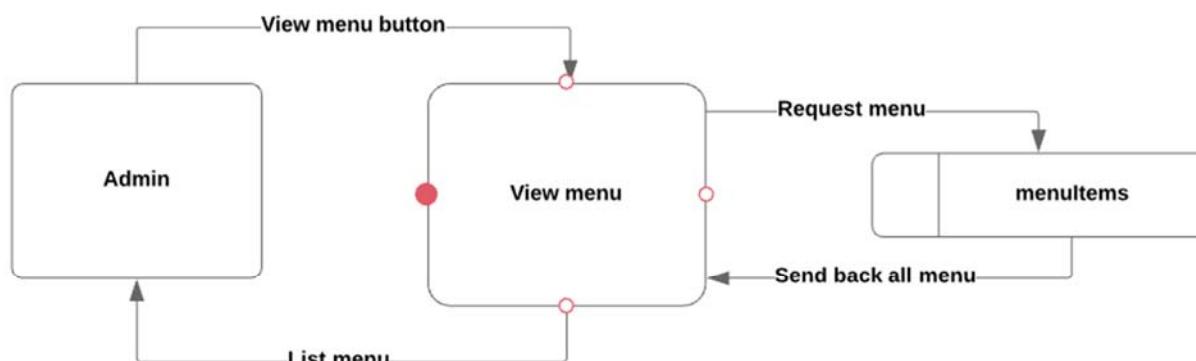
## Remove Workouts



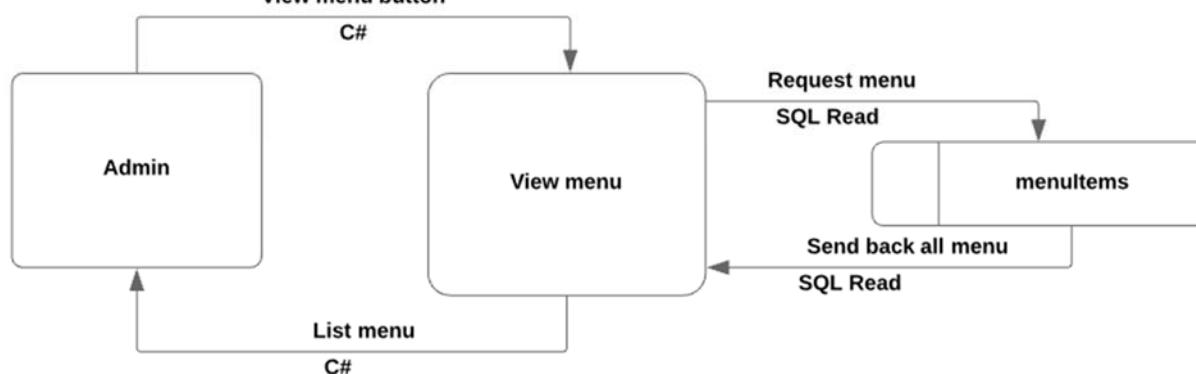
## Select workout



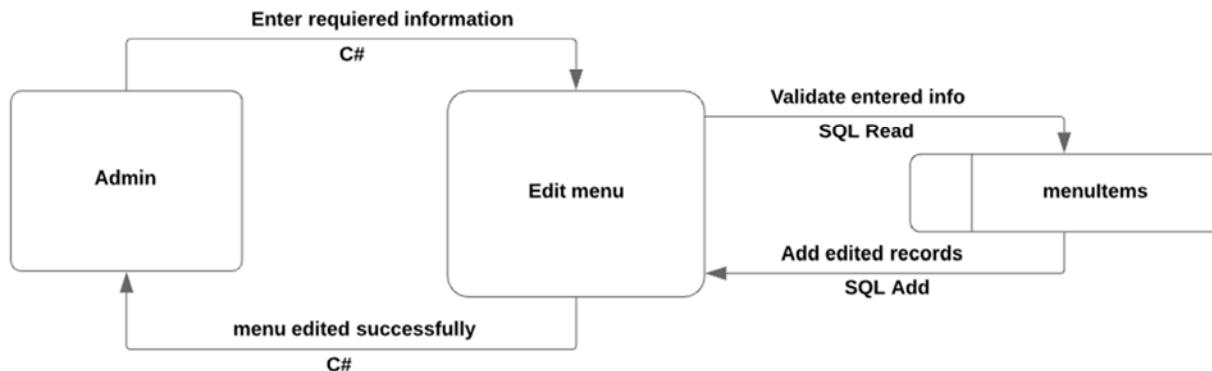
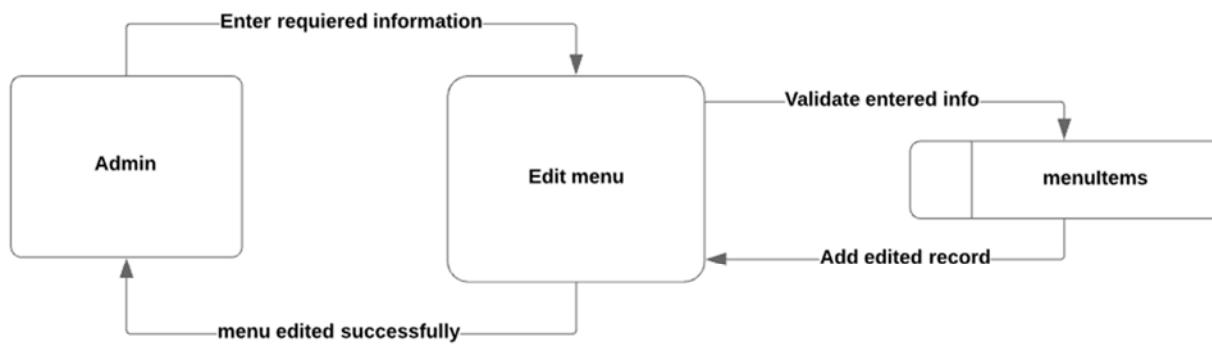
## View Menu



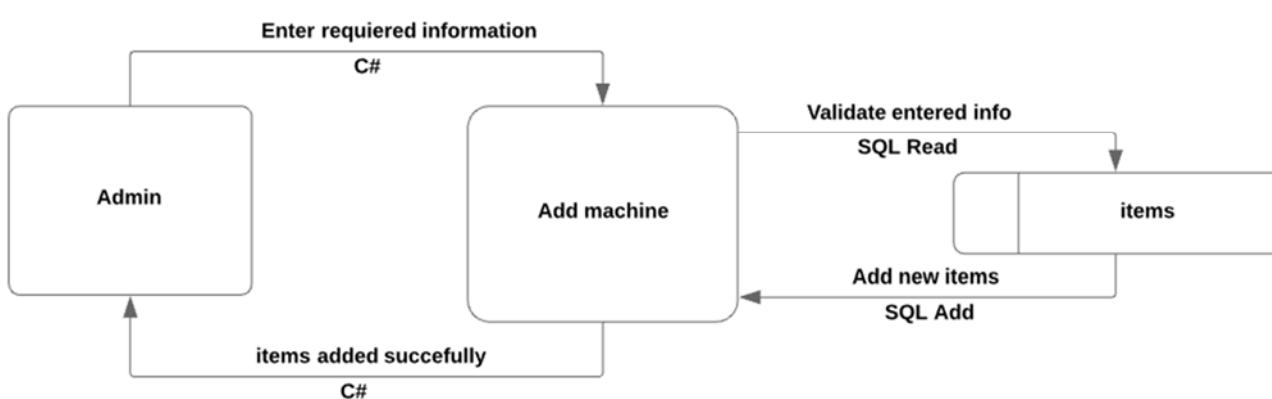
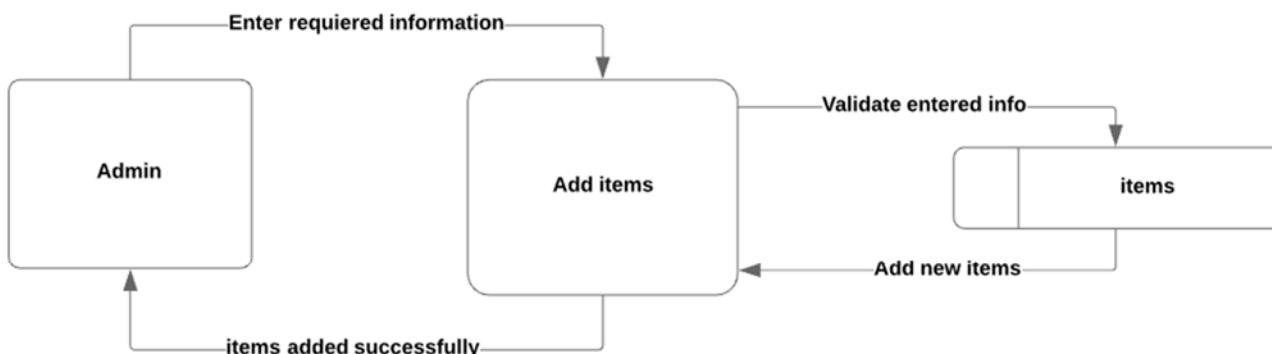
## View menu button



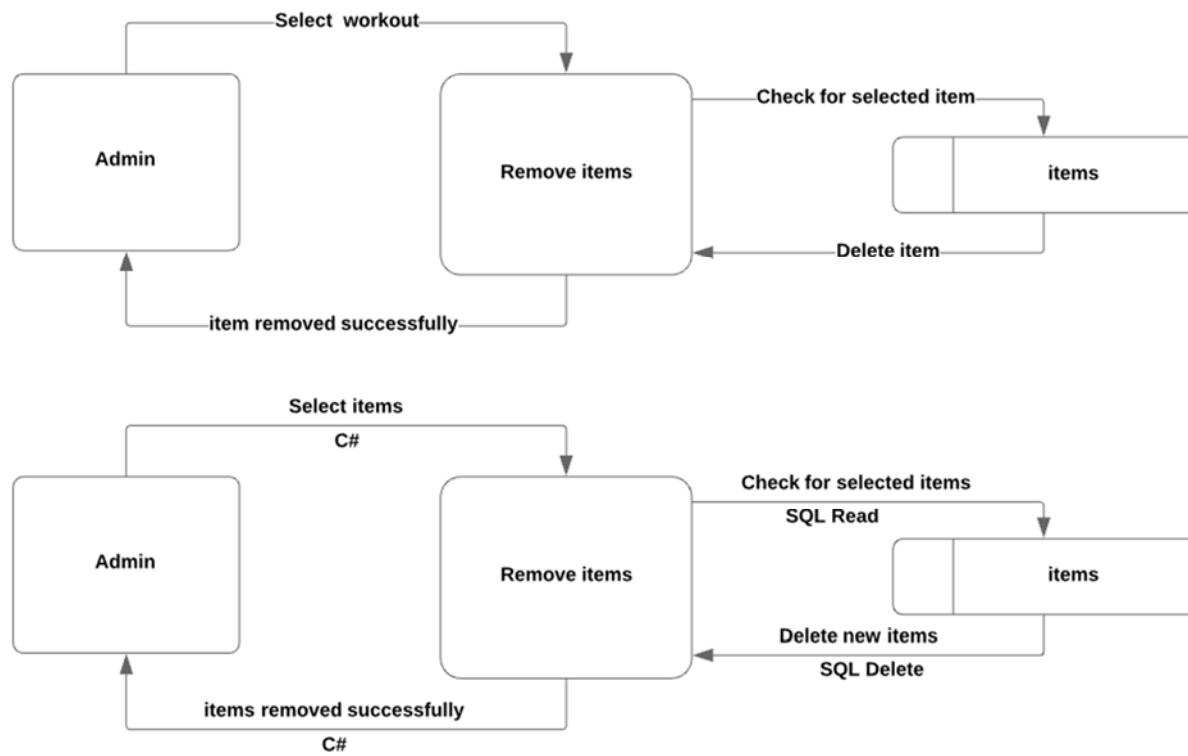
## Edit Menu



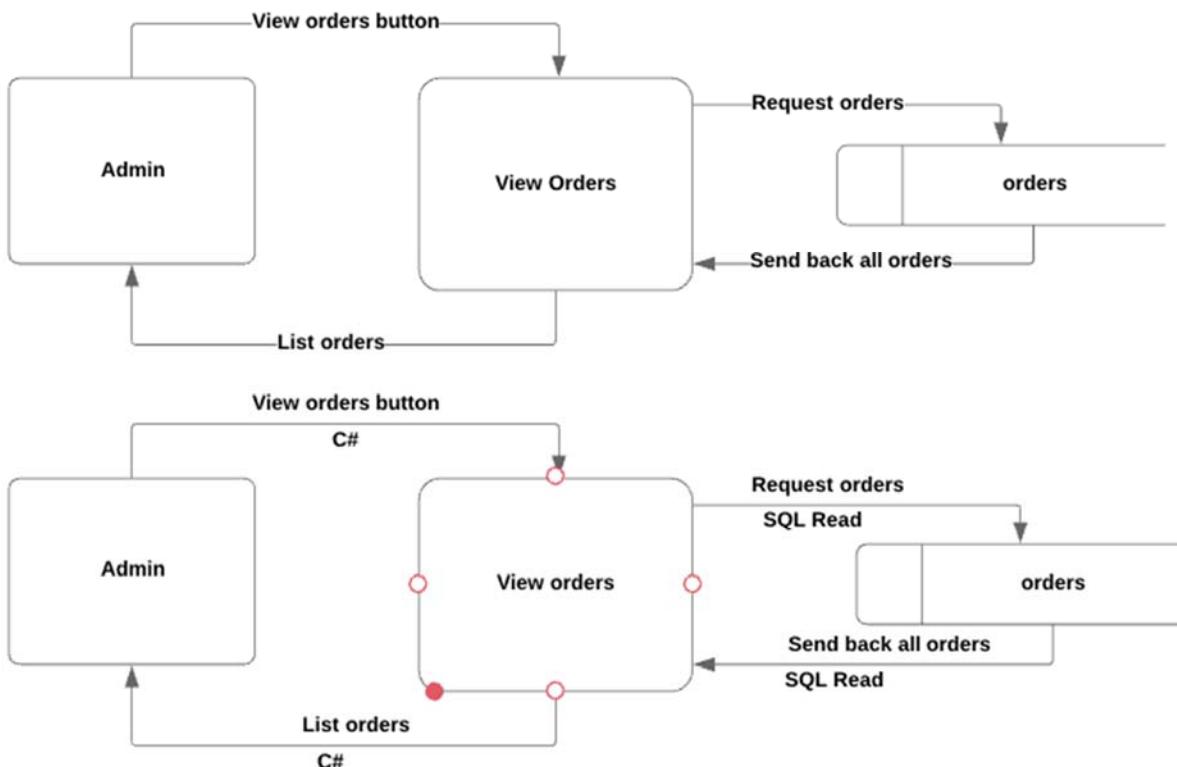
## Add Menu Items



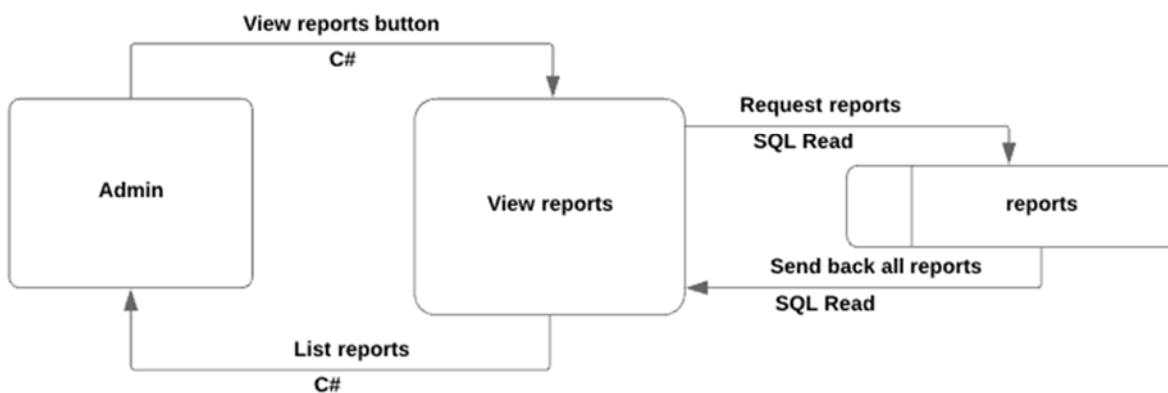
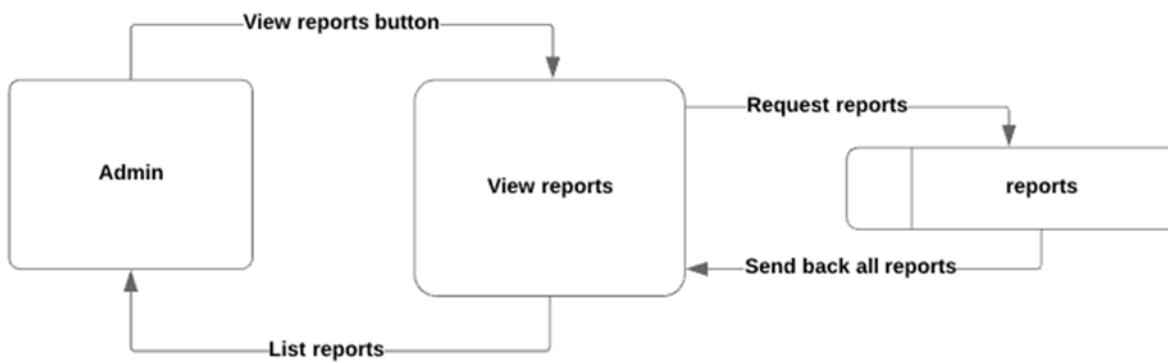
## Remove Menu Items



## View Orders

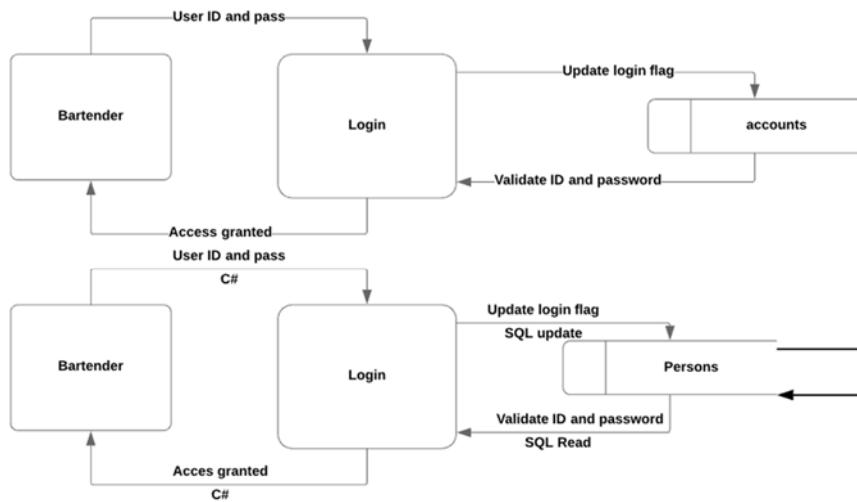


## View Report

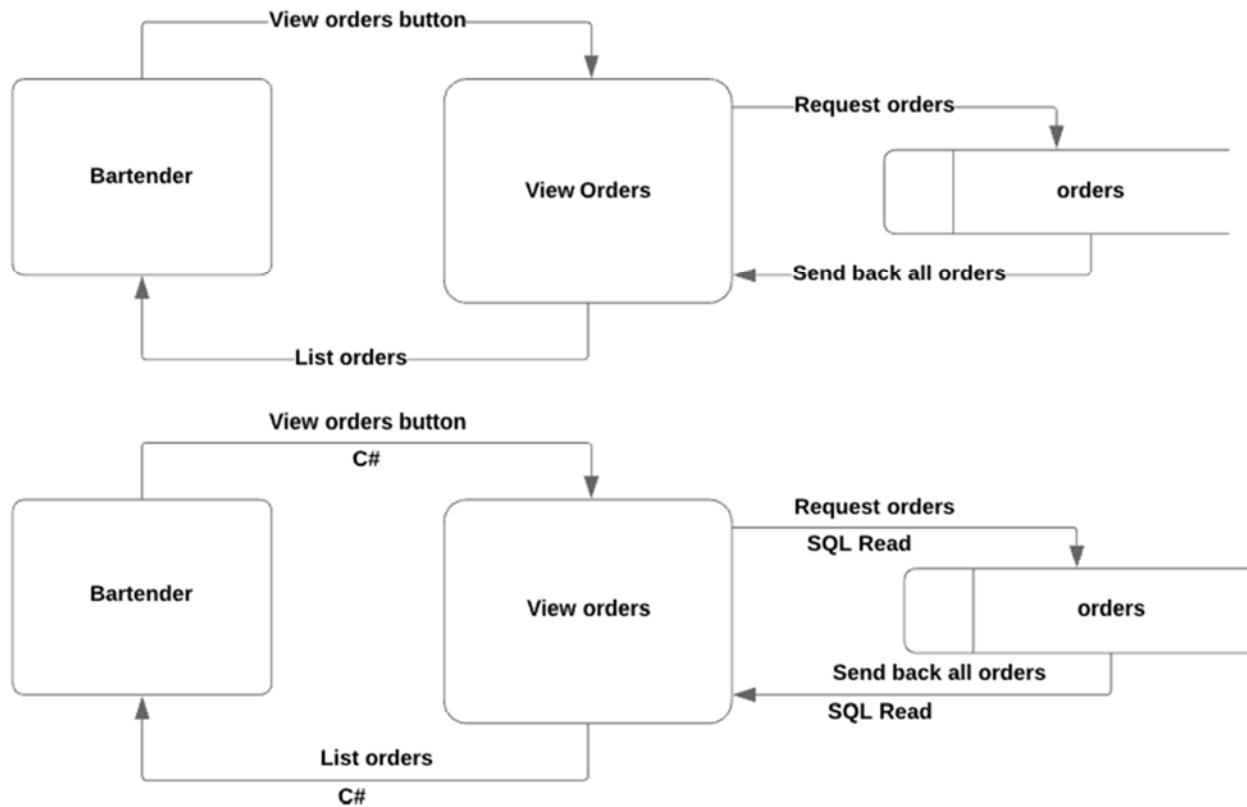


## Actor: Bartender

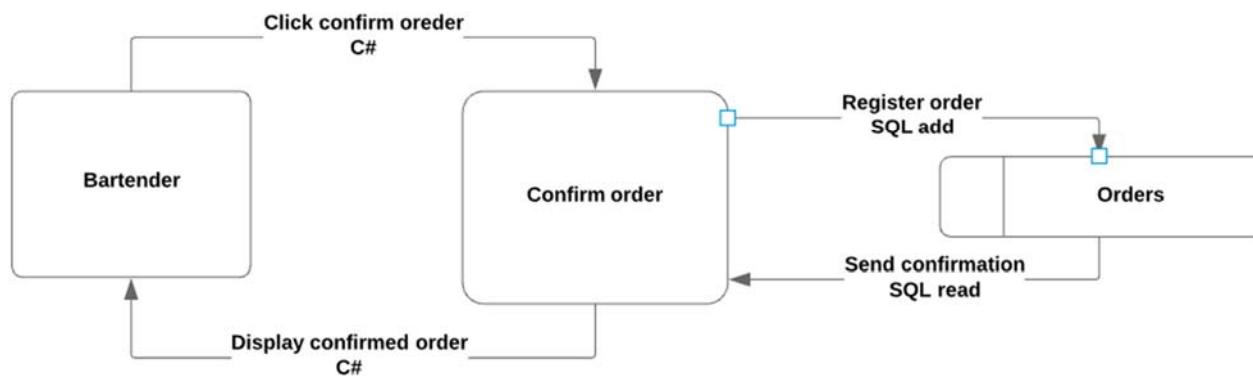
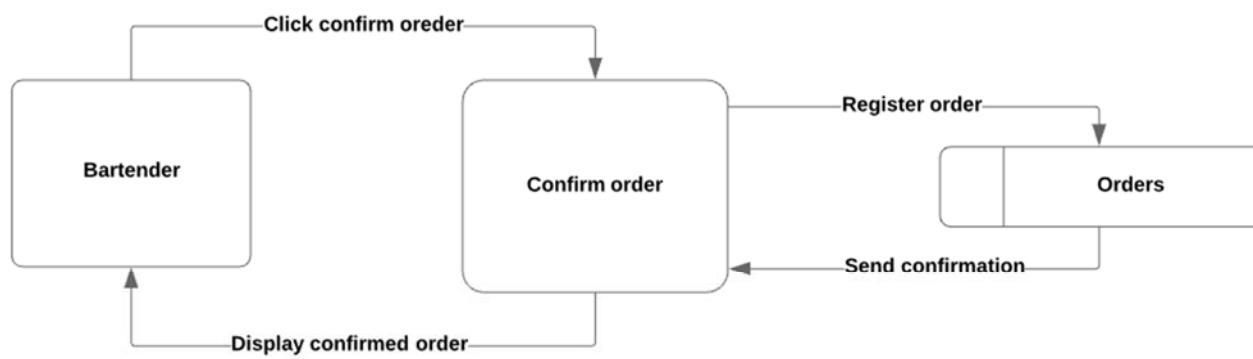
### Login



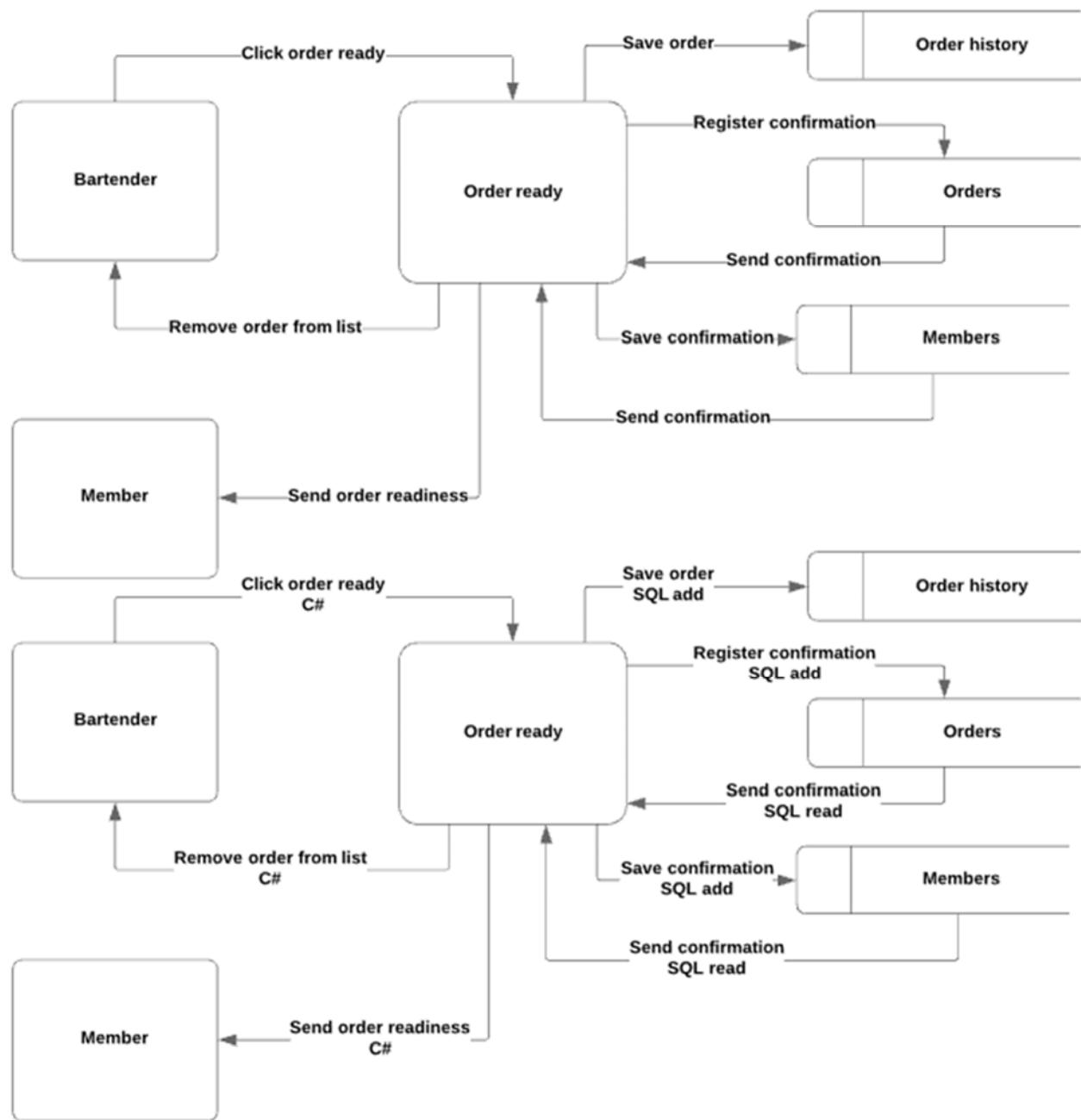
### View Orders



## Confirm Order



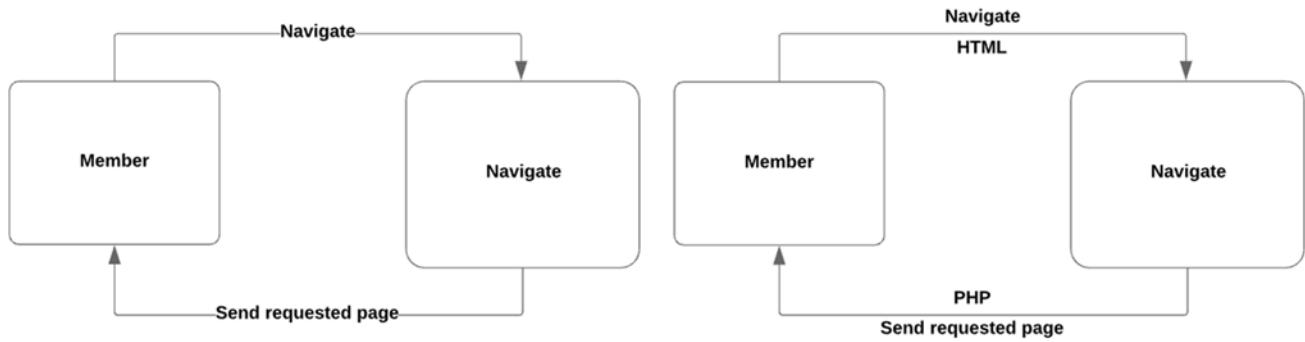
## Order Ready



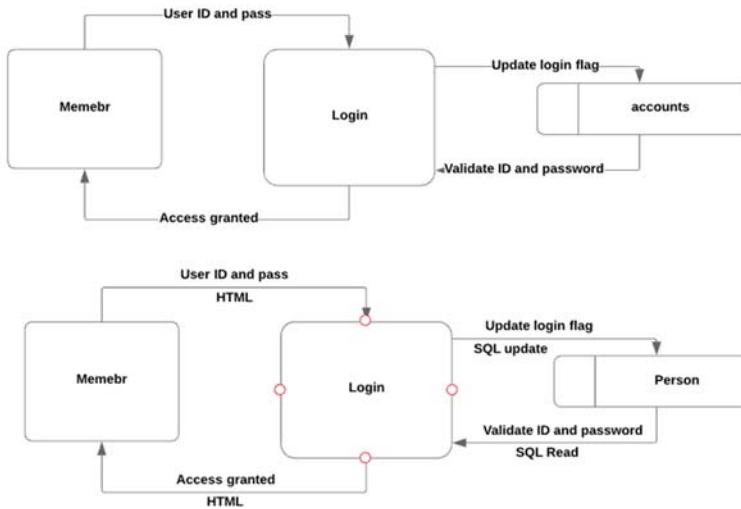
# Web Single Logical and Physical DFDs

Actor: Member

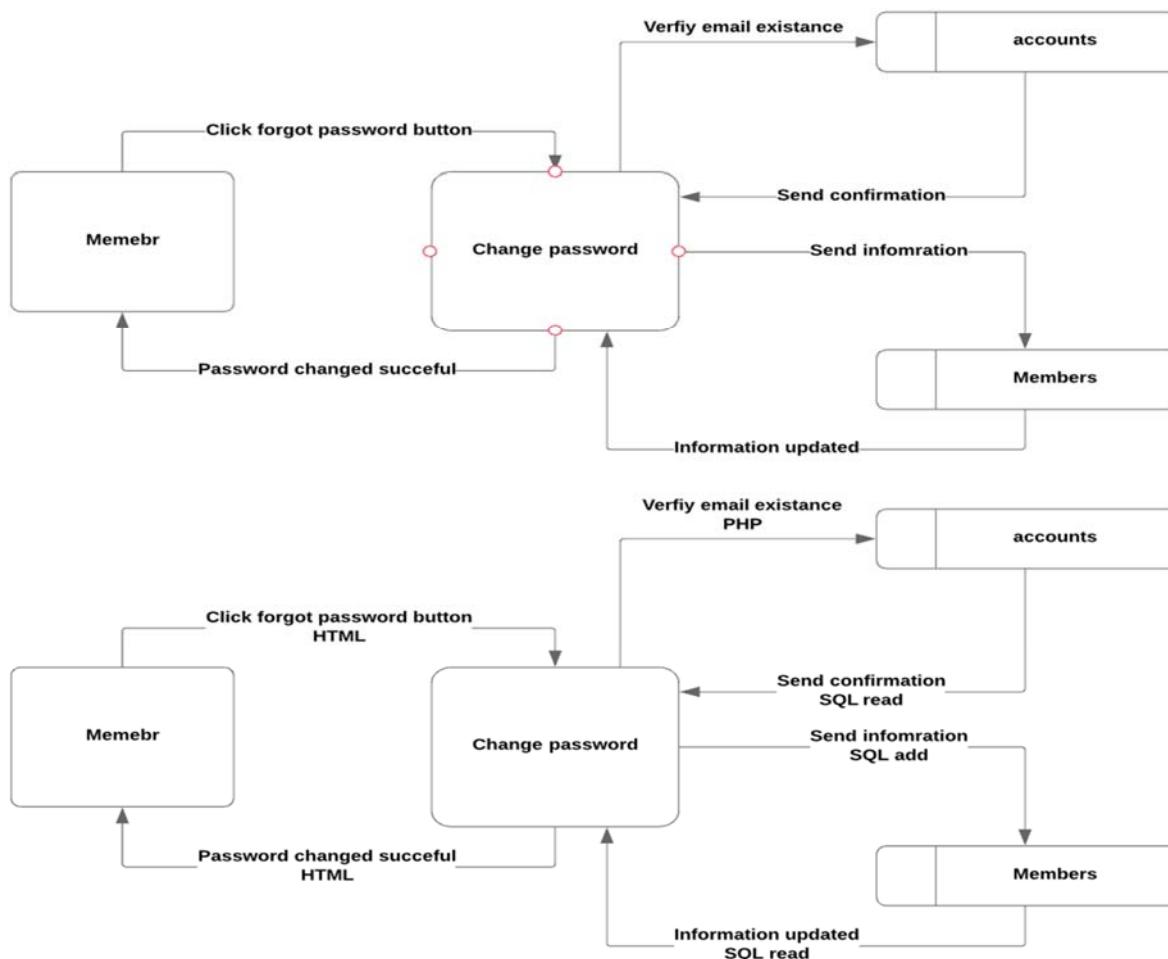
Navigate



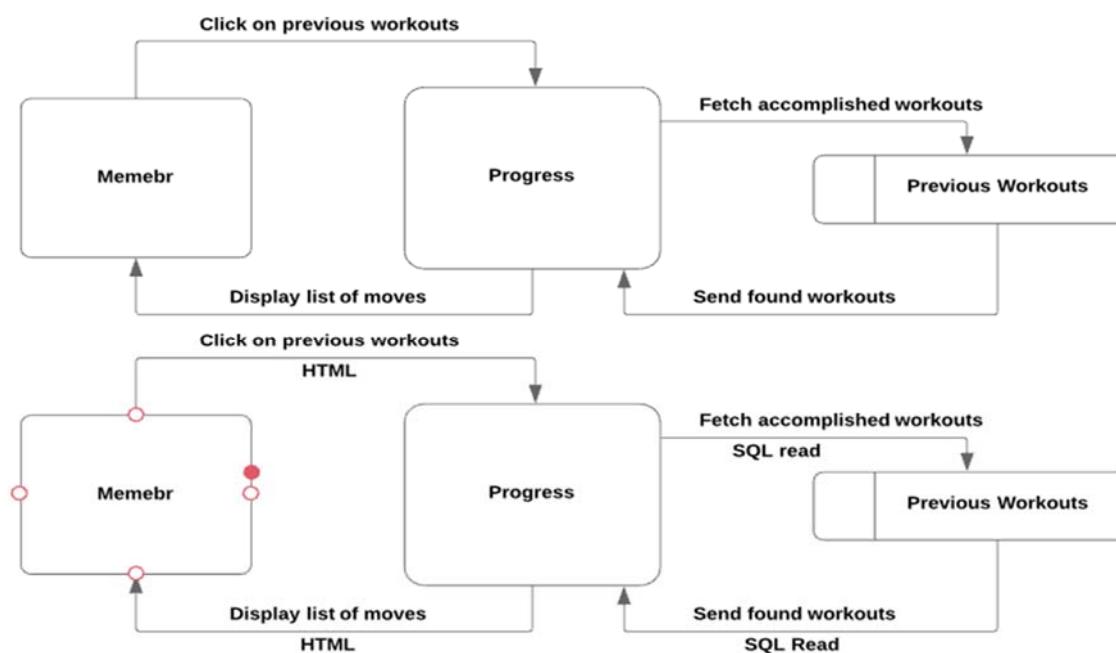
Login



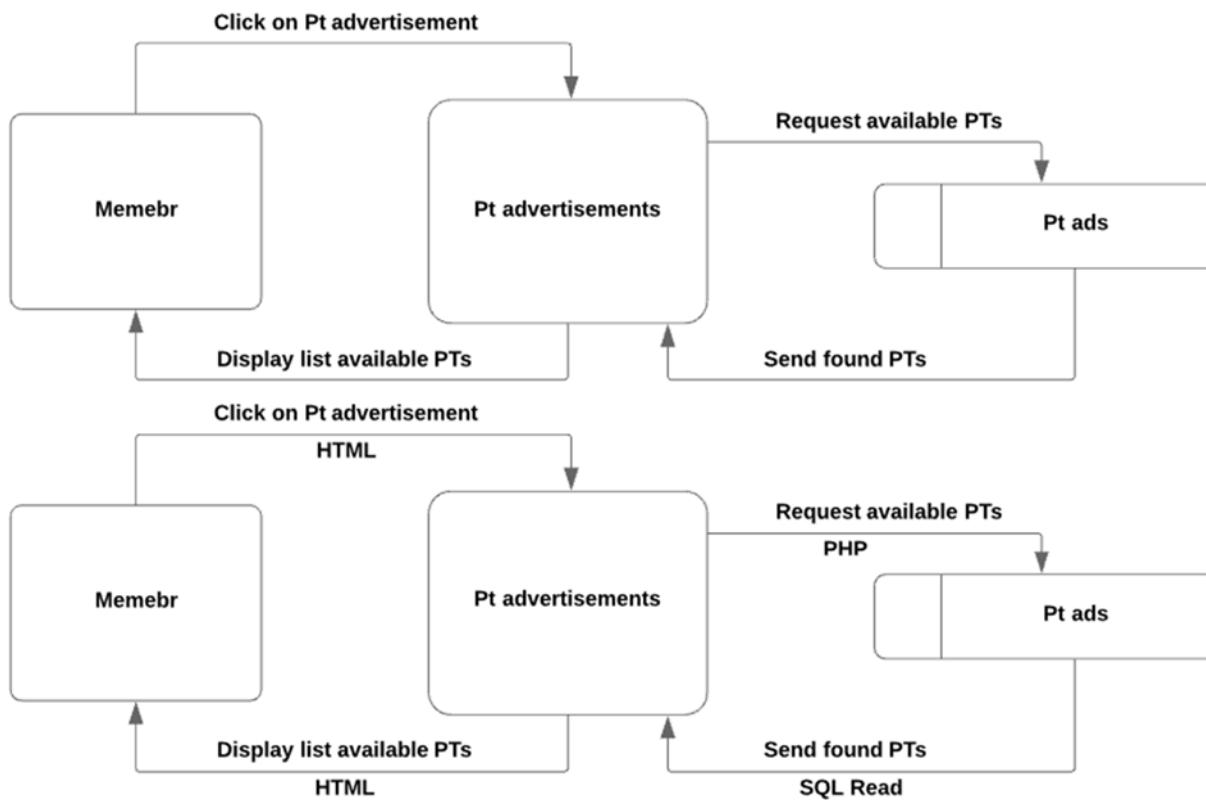
## Forgot Password



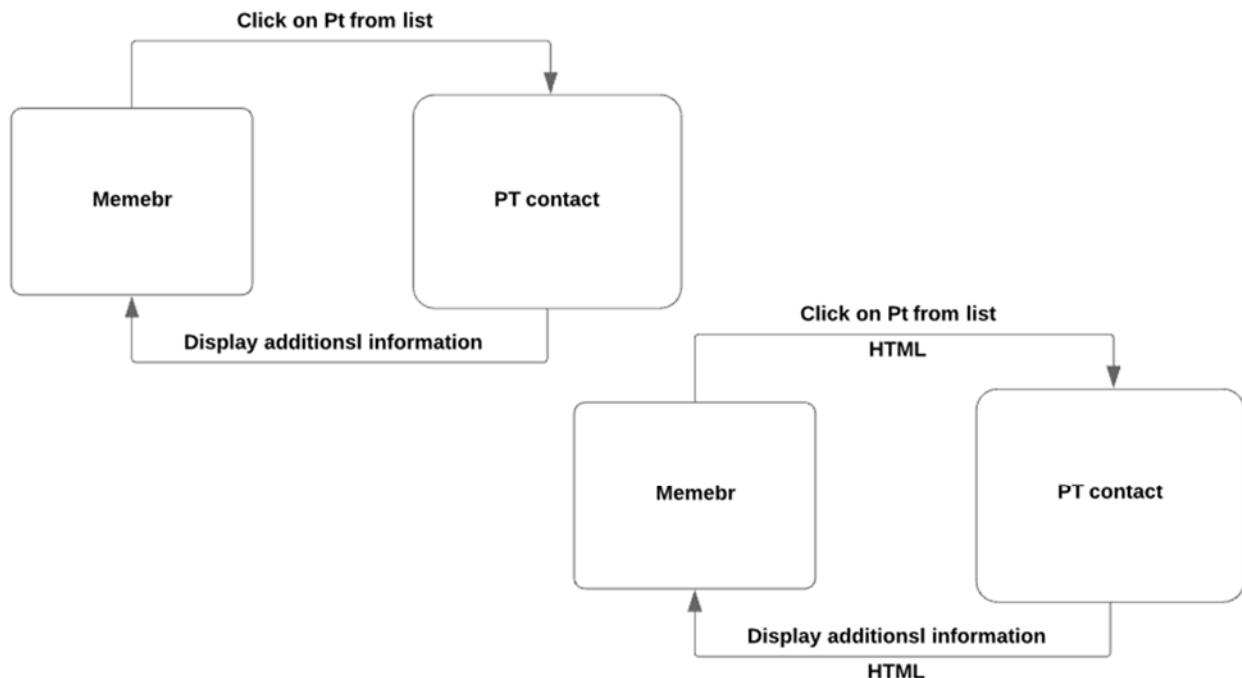
## Progress (Workout History)



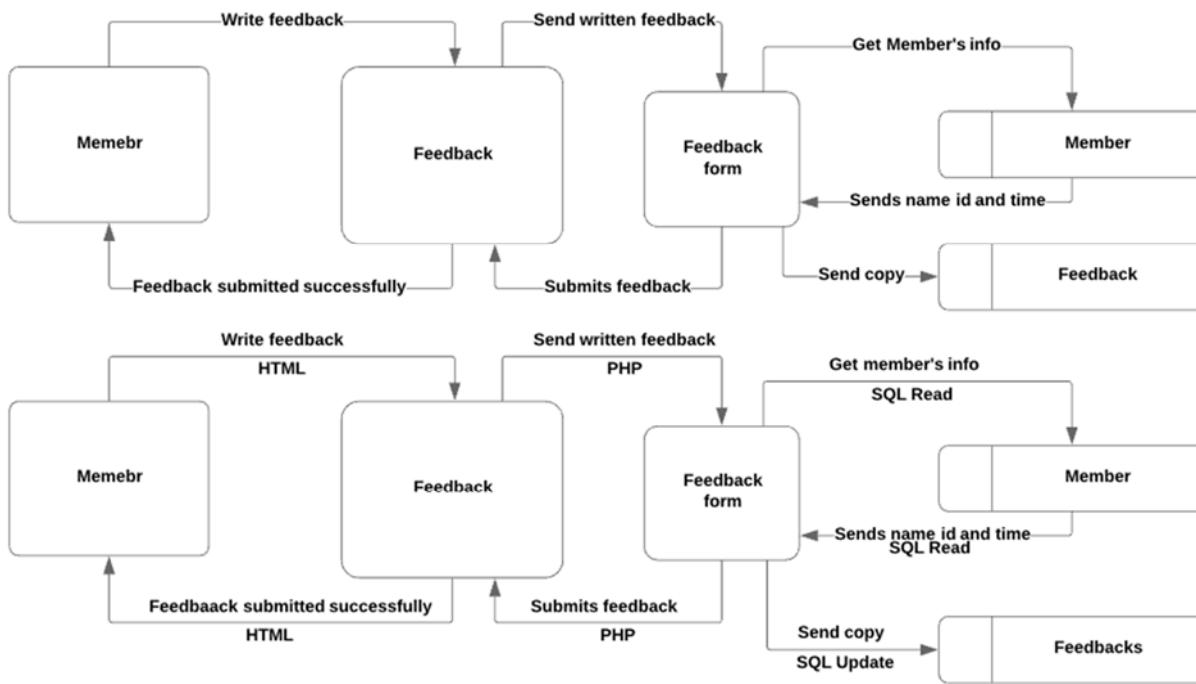
## PT Advertisements



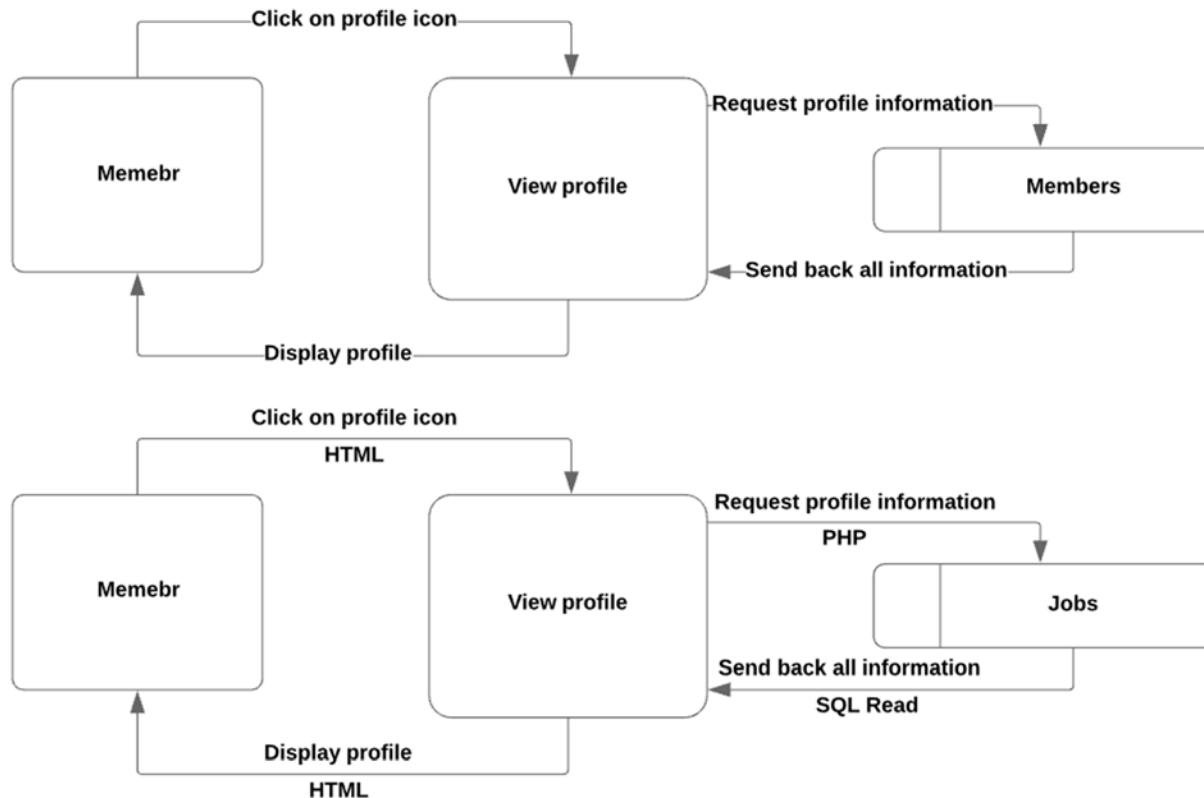
## Contact PT



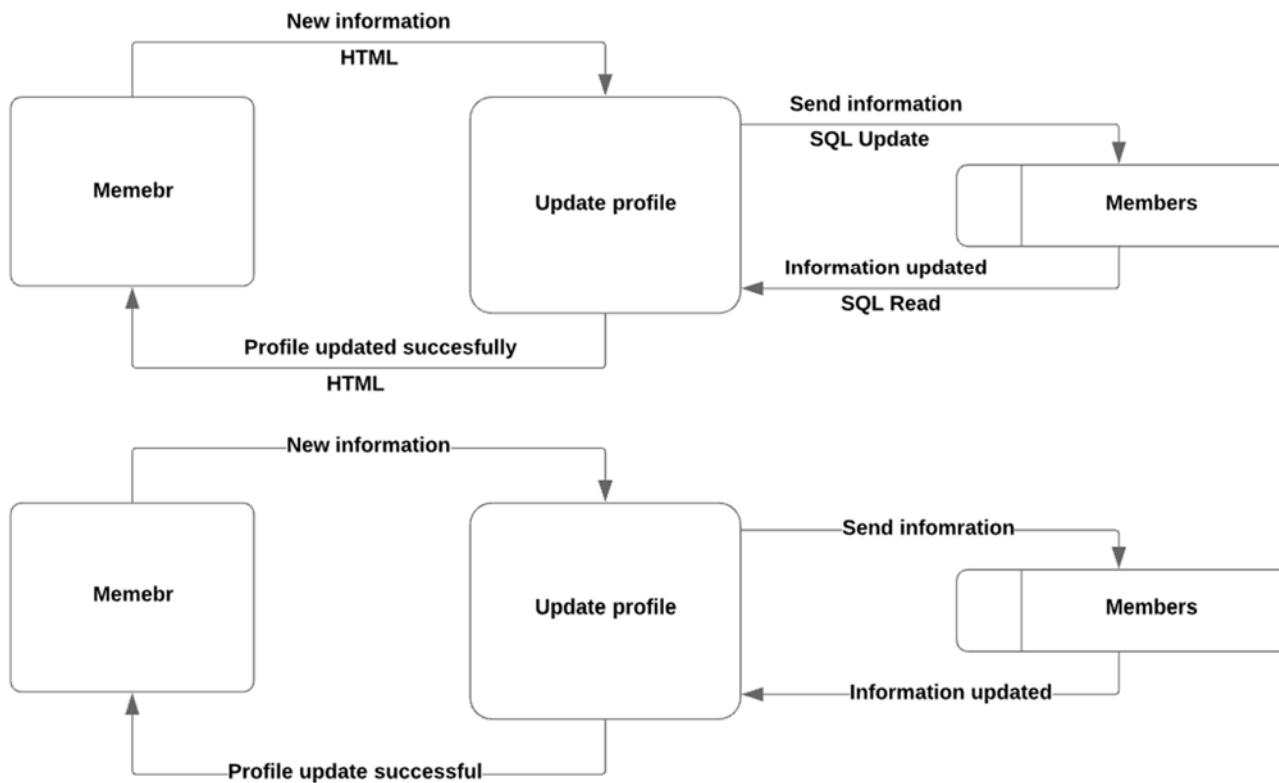
## Feedback



## View Profile

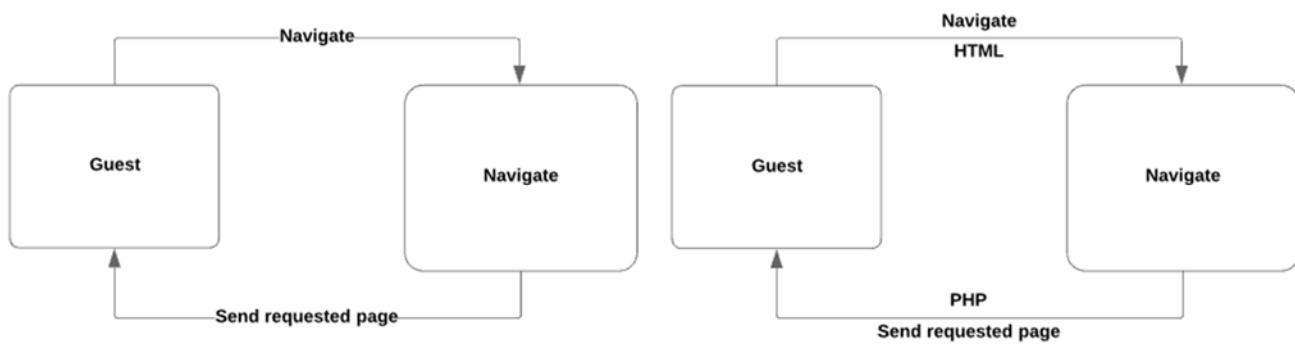


## Edit Profile

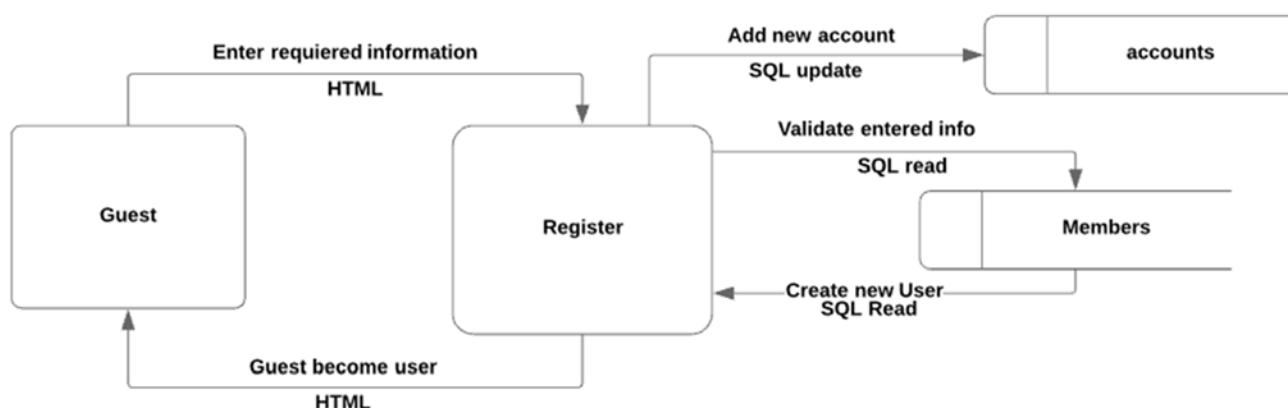
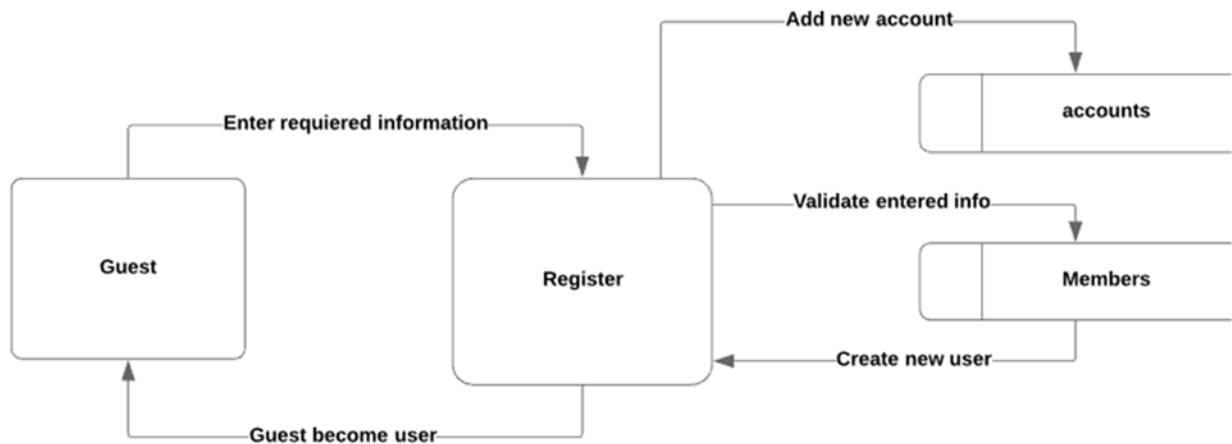


## Actor: Guest

### Navigate



### Register



## Email Confirmation

