# — O-Club —

Dalia Mahmoud

March 2022

| Name | Date | Reason for change | Version |
|---|---|---|---|
| O-Club SRS 1.0 | 06/03/2022 | The initial documentation | 1.0 |

## Contents

# 1 Introduction

## 1.1 Purpose of this document

This document is meant to delineate the web application's features, and also to guide to the developers and to be a software validation document for the client. This document will discuss O-Project's Affiliate Program, O-Club.

This document is intended for developers, product owners, project managers, marketing staff, users, testers, and documentation writers

Generally, Affiliate marketing works by spreading the responsibilities of product marketing and creation across parties, it manages to leverage the abilities of a variety of individuals for a more effective marketing strategy while providing affiliates with a share of the profit. To make this work, three different parties must be involved:

1. Seller and product creators. (O-Project)
2. The affiliate or advertiser. (The User)
3. The consumer. (The Client)

## 1.2 Scope of this document

This program helps affiliates to earn a commission for marketing our products or services. The affiliate simply contacts us, then promotes the product and earns a piece of the profit from each sale they make including various benefits.

## 1.3 Overview

This system is to assist affiliates to sell O-Projects products and services which will allow them to receive lots of benefits. They can receive benefits from our clients like a Smart Gym membership or some L'Oreal products, etc.

# 2 General Description

The web application allows users to pick slot to contact O-Projects team, and to search through the products and services. The system will utilize the web as the best way to offer items to its consumers.

## 2.1 Product Functions

1. User shall be able to pick a slot to contact our team through Calendly.
2. User shall receive a generated username and password.
3. User can change their password.
4. User shall view available projects and services.
5. User shall be able to create a new brief.
6. User shall be able to navigate through dashboard which contains their acquired projects and briefs.
7. User shall be able to view account manager assigned.
8. User shall see a list of benefits.
9. User can redeem benefits.
10. User can report a problem.
11. User shall view all money transactions.
12. User shall be able to view all financial history.
13. Admin shall add, edit and delete certain data.
14. Super Admin shall add, edit and delete Admin.
15. Super Admin should be able to view users.
16. Super Admin shall have full authorization to change any data.
17. Admin should gives access to employees.
18. Admin shall approve the briefs.
19. Admin and Super Admin should be able to decline brief.
20. User shall be able to sign out.
21. User should be able to cancel his slot.
22. User should be able to write feedback for team and project.

## 2.2 Similar System Information

Click Bank is a similar system to the one we are going to build, as it has the products classified and the chosen product with data and information about it included. Our system will have an extra functionality which is the ability of the user to customize any product according to his needs.

**Click Bank Link**.

## 2.3 User Types

1. Users
2. Admin
3. Super Admin

## 2.4 User Problem Statement

Enhancing O-Projects marketing through affiliates by presenting our projects and services to the users and consumers.

## 2.5 User Objective

By using this system we should be able to present our products and services, and the affiliate should be able to view the existing products and contact the team.

## 2.6 General Constrains

Some of the general constraints are:

1. The Customer's data needs to be safe and securely stored.
2. The Customer's information needs to be confidential.
3. The system is going to require internet connection.

# 3 Functional Requirements

## 3.1 Description

| Function | Book a slot |
|---|---|
| ID | FR00 |
| Description | This function shall allow users to book a meeting with O-Projects team. |
| Type | Boolean |
| Input | Meeting Date and Time, User name and project summary. |
| Action | Opens Calendly user interface to allow user to book a slot date and time. |
| Output | User slot is sent to team and added to calendar. |
| Precondition | Customer chose date and time. |
| Postcondtion | Creating a new entry in the team calendar with user information. |
| Dependencies | |

| Function | Password |
|---|---|
| ID | FR01 |
| Description | This function allows customers to change their generated passwords to their accounts. |
| Type | Boolean |
| Input | Password |
| Action | Checks if the entered input is 8 or more characters. |
| Output | Customer's password is added to the database. |
| Precondition | Customer entered the password into the required text field. |
| Postcondtion | Adding the password to the row of this user in the database. |
| Dependencies | - |

| Function | Mobile Number |
|---|---|
| ID | FR02 |
| Description | This function allows customers to add their mobile numbers to their accounts. |
| Type | Boolean |
| Input | Mobile Number (string) |
| Action | Checks if the entered input is 11 integer numbers. |
| Output | Customer's number is added to the database. |
| Precondition | Customer entered the mobile number into the required text field. |
| Postcondtion | Adding the number to the row of this user in the database. |
| Dependencies | - |

| Function | Log In |
|---|---|
| ID | FR03 |
| Description | This function allows customers to login to their accounts. |
| Type | Boolean |
| Input | Email, Password |
| Action | Checks if the entered input is valid. |
| Output | Logged in successfully |
| Precondition | Customer entered the email and password into the required text field. |
| Postcondtion | The customer is logged in and do whatever activities they need. |
| Dependencies | FR01 |

| Function | Forget Password |
|---|---|
| ID | FR04 |
| Description | This function allows customers to change their passwords if they forgot it. |
| Type | Boolean |
| Input | Email |
| Action | Checks if the entered input is valid. |
| Output | Reset email is sent to change the password. |
| Precondition | Customer entered the email into the required text field. |
| Postcondtion | The customer can login with the new password. |
| Dependencies | FR01 |

| Function | Add Brief |
|---|---|
| ID | FR05 |
| Description | This function allows user to add new briefs to the system. |
| Type | Array |
| Input | Product |
| Action | New brief are added to the database |
| Output | Brief appears on the system |
| Precondition | - |
| Postcondtion | Database is updated with new brief |
| Dependencies | - |

| Function | Edit Brief |
|---|---|
| ID | FR06 |
| Description | This function allows users to edit any brief on their profile. |
| Type | Boolean |
| Input | Product object |
| Action | The users will chose the desired brief to edit and save the changes. |
| Output | If true then edited brief appear on the system |
| Precondition | search for the desired brief. |
| Postcondtion | Database is updated |
| Dependencies | |

| Function | Delete Brief |
|---|---|
| ID | FR07 |
| Description | This function allows users to delete any brief in their profile. |
| Type | Boolean |
| Input | Product object |
| Action | The user should choose the desired brief to delete and save the changes. |
| Output | Brief is successfully deleted notification |
| Preconditon | search for the desired brief. |
| Postcondtion | Brief is removed from database |
| Dependencies | |

| Function | Add Employee |
|---|---|
| ID | FR08 |
| Description | This function allows admin to add any employee to the system. |
| Type | Boolean |
| Input | Email, password,full name, mobile number |
| Action | Checks if the input is valid. |
| Output | Employee is successfully added. |
| Precondition | Admin entered the Email into the required text field. |
| Postcondtion | Creating a new row in the database table with new ID for the employee type. |
| Dependencies | |

| Function | Edit Employee |
|---|---|
| ID | FR09 |
| Description | This function allows admin to edit any employee in the system. |
| Type | Boolean |
| Input | Name, email |
| Action | Checks if the employee exist then edit it. |
| Output | Notification that employee is successfully edited. |
| Precondition | Admin entered the Email, full name into the required text fields. |
| Post-condition | Editing the info in the database. |
| Dependencies | |

| Function | Delete Admin |
|---|---|
| ID | FR10 |
| Description | This function allows admin to delete any employee in the system. |
| Type | Boolean |
| Input | Name, email |
| Action | Checks if the input exists the delete it from database. |
| Output | Notification that admin is successfully deleted. |
| Precondition | Super Admin chose admin to delete. |
| Post-condition | Deleting the admin from the database. |
| Dependencies | |

| Function | View Briefs |
|---|---|
| ID | FR11 |
| Description | This function allows all users to view any brief on their profile. |
| Type | Array |
| Input | - |
| Action | All the briefs on the user's profile appear and the user can search to find briefs. |
| Output | Brief is viewed |
| Precondition | - |
| Postcondtion | - |
| Dependencies | |

| Function | Report a Problem |
|---|---|
| ID | FR12 |
| Description | This function allows admins to view any feedback on the system. |
| Type | Boolean |
| Input | - |
| Action | Admin/Employee can view all problems |
| Output | Problem is viewed |
| Precondition | - |
| Postcondtion | - |
| Dependencies | |

| Function | View Feedback |
|---|---|
| ID | FR13 |
| Description | This function allows admins to view any feedback on the system. |
| Type | Boolean |
| Input | - |
| Action | Admin/Employee can view all feedback about any product from different users |
| Output | Feedback is viewed |
| Preconditon | - |
| Postcondtion | - |
| Dependencies | |

| Function | Redeem Benefit |
|---|---|
| ID | FR14 |
| Description | This function allow admin/employee to view any feedback on the system. |
| Type | Boolean |
| Input | - |
| Action | Benefit is assigned to user |
| Output | Benefit is decremented from account and assigned to user |
| Preconditon | - |
| Postcondtion | - |
| Dependencies | |

| Function | View Benefits |
|---|---|
| ID | FR15 |
| Description | This function allow admin/employee to view any feedback on the system. |
| Type | Boolean |
| Input | - |
| Action | Benefit is assigned to user |
| Output | Benefit is decremented from account and assigned to user |
| Preconditon | - |
| Postcondtion | - |
| Dependencies | |

| Function | View User Profile |
|---|---|
| ID | FR16 |
| Description | Admin will be able to see any user's public data on profile. |
| Type | Return Array |
| Input | User Object |
| Action | Check if user is signed up and available in database. |
| Output | Array of each customer data. |
| Pre-condition | The customer is signed up and has an account. |
| Post-condition | - |
| Dependencies | |

| Function | Delete User |
|---|---|
| ID | FR17 |
| Description | Allows the admin to delete users from database. |
| Type | Boolean |
| Input | ID of user to be removed. |
| Action | Check if user ID exists in database. |
| Output | Delete all the data of user from database and returns true or error message. |
| Pre-condition | The user is signed up and have an account. |
| Post-condition | Update database by removing the user ID row from all the database tables. |
| Dependencies | - |

| Function | View List of All Users |
|---|---|
| ID | FR18 |
| Description | Show all customers in the database |
| Type | Return Array |
| Input | Customer objects to be listed |
| Action | Check the database and return all customers in it |
| Output | Array by each customer data |
| Pre-condition | Customer exist |
| Post-condition | - |
| Dependencies | |

| Function | Confirm Brief |
|---|---|
| ID | FR19 |
| Description | Employee confirm the user brief |
| Type | Boolean |
| Input | - |
| Action | Employee check that the sent brief is doable |
| Output | True if the brief is doable and the order is approved, False if not approved. |
| Pre-condition | Customer Placed an order |
| Post-condition | Send message to the customer notifying him. |
| Dependencies | |

| Function | View Financial History |
|---|---|
| ID | FR20 |
| Description | Customer can see the history of his finances. |
| Type | array |
| Input | - |
| Action | Customer is able to list all his previous received money and transactions |
| Output | List of all his briefs and money |
| Pre-condition | Customer must have an account |
| Post-condition | - |
| Dependencies | |

# 4 Performance Requirements

The program needs to be fast and sensitive to obtain the data the user wants. The system is also expected to safeguard user data and information.

# 5 Design Constraint

An example of Constraints identified with the site's general intrigue is aesthetics, which allude to the look and feel of our identity. These imperatives concern fundamentally visual data; for instance, the photos or colors utilized.

# 6 Other non-functional attributes

## 6.1 Security

The system will provide security by encrypting all-important data that includes passwords, payment methods, Emails, addresses all of the data will be kept at the database. Also, our system won't allow unauthorized users to access any info.

## 6.2 Maintainability

The system should allow new features to be added easily by other developers as the system is documented.

## 6.3 Portability

The program is based on HTML and scripting language. So the end-user component is highly scalable and any device that uses any web browser will be able to use the system's features, including any hardware platform that is or will be available for potential use. This program is used by an end-user on any OS.

## 6.4 Availability

The system will be available at all times, ensuring that the user can access it through a web browser, restricted only by the server downtime on which the system is operating. In case of a hardware failure or malfeasance in the database, a replacement page should be displayed and database backups should also be retrieved from the server and saved by the administrator.

# 7 Operational Scenarios

1. Scenario 1: User wants to sell O-Project's products to a consumer. User starts viewing our products and services. User creates a brief for o-Project's team to view and then books a slot on Calendly to meet with

our team. After the call and arranging the deal, User can view his/her commission details through the application. User can benefit from the benefits list. User can view financial history of older briefs.

2. Scenario 2:

   An admin accesses the system, he is firstly required to log in. Then he checks if there are any unanswered briefs from users that he can confirm or decline. The Admin can start adding new products and editing old ones or deleting them from the system. The Admin also must confirm on all the briefs placed on the system and track them with the team. Finally, the admin can view feedback that are on the system about the products and services.

3. Scenario 3:

   The Super Admin accesses the system by logging in. The super admin can view everything on the system; he has more access rights than the admin. The super admin wants to add or remove an admin and he can do that easily using the super admin panel. He can also view all the users that have accounts on the system and their public data .