

[Foodies]

Design Document



May 15, 2019

Design Document

Project Name: Foodies

Project Manager: Aya Hamdy

Synopsis: Design Document

Version: V1.12

Issue Date: 05-15-2019

Prepared by: Marwa El-Sheshtawy

Authorized by: Foodies Team

* **Revision History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Issue Date** | **Changes** | **Author** | **Reviewed By** |
| 1.0 | 06-05-2019 | Initial version | Marwa El-sheshtawy |  |
| 1.1 | 07-05-2019 | Add Architecture Diagram | Aya Hamdy |  |
| 1.2 | 08-05-2019 | Add Use Case Scenarios | Marwa El-sheshtawy | Mariam Nesiem |
| 1.3 | 10-05-2019 | Add Data Flow diagram | Dina Ibrahim | Aya Hamdy |
| 1.4 | 10-05-2019 | Add HLDS Description | Aya Hamdy |  |
| 1.5 | 10-05-2019 | Add new section for class diagram | Aya Hamdy. |  |
| 1.6 | 10-05-2019 | Add sequence Diagram | Aya Hamdy. |  |
| 1.7 | 11-05-2019 | Add wireframes captions | Maryam M. |  |
| 1.8 | 11-05-2019 | Edit user sequence diagram and new alternative solutions | Aya Hamdy |  |
| 1.9 | 11-05-2019 | Add ERD Diagram | Hossam Galal |  |
| 1.10 | 12-05-2019 | Edit Use Case Diagram and Use Case Scenarios | Marwa El-Sheshtawy |  |
| 1.11 | 13-05-2019 | Edit wireframes | Maryam M. |  |
| 1.12 | 15-05-2019 | Update sequence diagram | Aya Hamdy |  |

* **Table of content**

[1 Introduction 6](#_Toc8643787)

[1.1 Document Purpose 6](#_Toc8643788)

[1.2 Project Overview 6](#_Toc8643789)

[1.3 Document Scope 6](#_Toc8643790)

[1.4 Document Audience 6](#_Toc8643791)

[2 System Architectural Design 7](#_Toc8643792)

[2.1 High Level System Architecture 7](#_Toc8643793)

[2.2 Use Case Diagram 8](#_Toc8643794)

[2.2.1 Use Case Scenarios: 9](#_Toc8643795)

[2.2.1.1 Register Use Case Scenario: 9](#_Toc8643796)

[2.2.1.2 Login Use Case Scenario: 10](#_Toc8643797)

[2.2.1.3 View Loyalty points Use Case Scenario: 12](#_Toc8643798)

[2.2.1.4 View Profile Page Use Case Scenario: 13](#_Toc8643799)

[2.2.1.5 View restaurant information Use Case Scenario: 14](#_Toc8643800)

[2.2.1.6 Make Order Use Case Scenario: 15](#_Toc8643801)

[2.2.1.7 Browse List of restaurants Use Case Scenario: 17](#_Toc8643802)

[2.2.1.8 Search restaurant Use Case Scenario: 18](#_Toc8643803)

[2.2.1.9 Delete User Use Case Scenario: 19](#_Toc8643804)

[2.2.1.10 Add Restaurant Use Case Scenario: 21](#_Toc8643805)

[2.2.1.11 Log out Use Case Scenario: 22](#_Toc8643806)

[2.3 Wireframes 23](#_Toc8643807)

[2.3.1 Login Page 23](#_Toc8643808)

[2.3.2 Create Account Page 24](#_Toc8643809)

[2.3.3 Homepage 25](#_Toc8643810)

[2.3.4 Offers&Promotions Page 26](#_Toc8643811)

[2.3.5 Menu Page 27](#_Toc8643812)

[2.3.6 User Account Page 28](#_Toc8643813)

[2.3.7 Admin Panel – Delete User Page 29](#_Toc8643814)

[2.3.8 Admin Panel – Add Menu Page 30](#_Toc8643815)

[2.3.9 Admin Profile 31](#_Toc8643816)

[2.4 Data Flow Diagram: 32](#_Toc8643817)

[2.5 Class Diagram: 33](#_Toc8643818)

[2.6 Sequence Diagram: 34](#_Toc8643819)

[2.6.1 Admin Panel – Delete User’s Profile 34](#_Toc8643820)

[2.6.2 Admin Panel – Add New Restraunt 35](#_Toc8643821)

[2.6.3 User Sequence Diagram 35](#_Toc8643822)

[2.7 ERD Diagram: 37](#_Toc8643823)

* **Table of figures**

[Figure 1 Foodies Sys Architecture 6](#_Toc8668551)

[Figure 2 Foodies\_Sys\_UseCase 7](#_Toc8668552)

[Figure 3: Foodies\_Sys\_LoginPage 22](#_Toc8668553)

[Figure 4: Foodies\_Sys\_CreateAccountPage 23](#_Toc8668554)

[Figure 5: Foodies\_Sys\_Homepage 24](#_Toc8668555)

[Figure 6: Foodies\_Sys\_Offers&PromosPage 25](#_Toc8668556)

[Figure 7: Foodies\_Sys\_MenuPage 26](#_Toc8668557)

[Figure 8: Foodies\_Sys\_UserAccountPage 27](#_Toc8668558)

[Figure 9: Foodies\_Sys\_Admin\_DeleteUserPage 28](#_Toc8668559)

[Figure 10: Foodies\_Sys\_Admin\_AddMenuPage 29](#_Toc8668560)

[Figure 11: Foodies\_Sys\_AdminProfile 30](#_Toc8668561)

[Figure 12 Foodies\_Sys\_DFD 31](#_Toc8668562)

[Figure 13 Foodies\_Sys\_ClassDiagram 32](#_Toc8668563)

[Figure 14 Foodies\_seq\_Admin\_delete 33](#_Toc8668564)

[Figure 15 Foodies\_sys\_seq\_admin\_addrestaurant 34](#_Toc8668565)

[Figure 16Foodies\_seq\_User 35](#_Toc8668566)

# Introduction

The Software Design Document to provide documentation which will be used to aid in software development by providing the details for how the software should be built. Within the Software Design Document are narrative and graphical documentation of the software design for the project including use case models, sequence diagrams, object behavior models, and other supporting requirement information.

## Document Purpose

The purpose of the Software Design Document is to provide a description of the design of a system fully enough to allow for software development to proceed with an understanding of what is to be built and how it is expected to build. The Software Design Document provides information necessary to provide description of the details for the software and system to be built.

## 1.2 Project Overview

**Foodies** is a web app that will provide a good source of the nearby restaurants so the users can discover, browse all the restaurants near him or perform a keyword search and order their food from mouthwatering photo-driven menus.

## 1.3 Document Scope

This Software Design Document is for a base level system which will work as a proof of concept for the use of building a system that provides a base level of functionality

## 1.4 Document Audience

* Development team
* Testing team
* Project Manager
* Product owner

# System Architectural Design

## 2.1- High Level System Architecture

**Description:**

Foodies HLSD shows the components of the system and interaction between it and users of this system.



Figure 1 Foodies Sys Architecture

## 2.2- Use Case Diagram

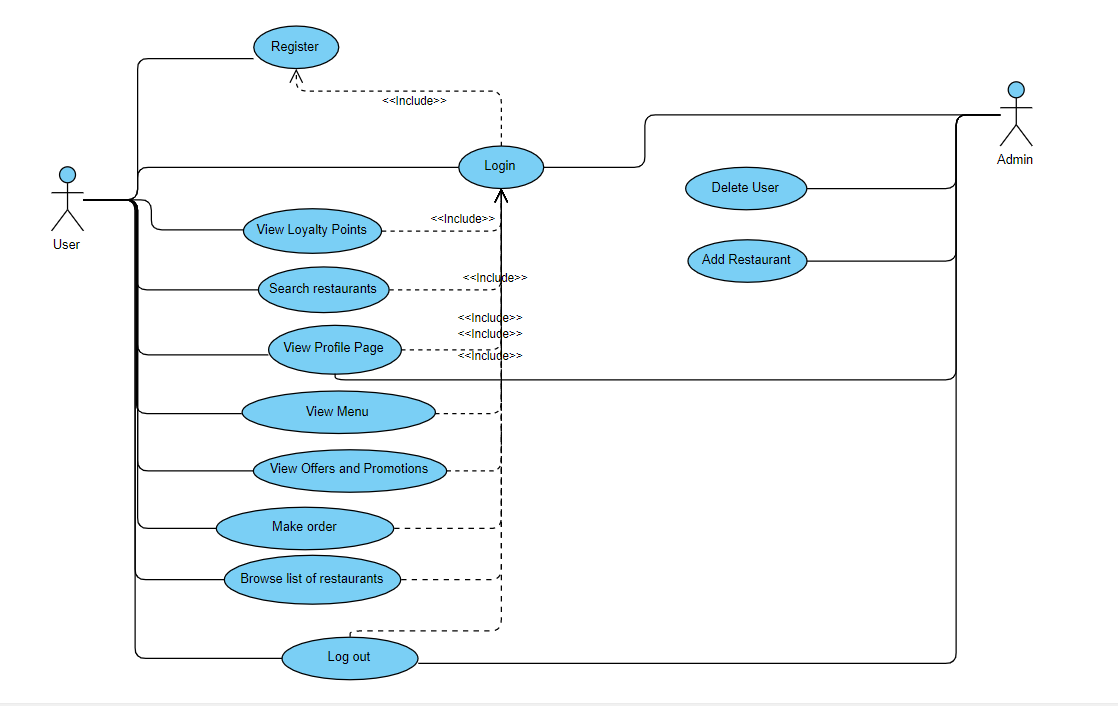


Figure 2 Foodies\_Sys\_UseCase

### 2.2.1 Use Case Scenarios:

#### 2.2.1.1 Register Use Case Scenario:

|  |  |
| --- | --- |
| **Foodies\_Sys\_UseCase\_001** | UseCase\_ID |
| Register | Scenario Name |
| User | Actors |
| A New user of the System creates an account by entering his personal information to create an account. | Brief Description |
| None | Pre-condition |
| |  |  | | --- | --- | | **System Response** | **Actor** | | * 1. The System is successfully create user’s account.   2. The System is failed to create account. | 1- If user entered successful information and is returned to the Home page as a Logged In User.  2- User is unable to Register for one or more reason and is returned to the Register page again. |  | | Post condition |
| |  |  | | --- | --- | | **System Response** | **Actor** | | * 1. The system prompts the user for a user Id and password or register new account.   2. The system prompts user for registration mandatory information (user Id, Email and password, gender, location).   3. System verifies information and creates account.   4. The system navigate from register to Home page | 1. New user select option to register, so system create an account. 2. New User fill-in the required Mandatory information (user Id, Email and password, gender, location). 3. Submit register button. 4. Return to Home page. | | Steps |
| |  |  |  | | --- | --- | --- | | **System Response** | **Actor** | **Actor** | | 1.1-System displays information with appropriate message to correct invalid information.  2.1-System verifies user’s information and navigate to HomePage. | | 1- The User enter invalid information (User Id /Password or Email).  2- The User  Re-enter information.  3- Click Submit. | | Alternative conditions |

Table 1: Register Use Case Scenario.

#### 2.2.1.2 Login Use Case Scenario:

|  |  |
| --- | --- |
| **Foodies\_Sys\_UseCase\_002** | UseCase\_ID |
| Login | Scenario Name |
| User/Admin | Actors |
| A user of the System can login using his valid credentials. | Brief Description |
| User/Admin must be registered before. | Pre-condition |
|  |  |
| |  |  |  | | --- | --- | --- | | **System Response** | **Actor** | | | 2.1- The system validates the entered user Id and password, making sure that the entered username is a valid username in the System, and that the entered password is the password associated to the entered user Id.  3.1- The System navigate to restaurants page if the user is normal user and direct to admin panel if an admin. | This use case starts when a system user is not logged in to the system and goes to the login page.   1. The Admin/User wants to log in. 2. Fill the user id and password. 3. Click Login Button. | | Steps |
| |  |  |  | | --- | --- | --- | | **System Response** | **Actor** | | | * 1. The system is successful.   2.1- The system fails. | | 1. If the user/Admin authenticated and redirected to next page. 2. If the user/Admin not authenticated | | Post –Condition |
| |  |  |  | | --- | --- | --- | | **System Response** | **Actor** | | | * 1. The system describes the reasons why the User authentication failed.   2. The system prompts the User to re-enter the valid information | | 1. If the user/Admin entered invalid user Id/password. | |  |

Table 2: Login Use Case Scenario.

#### 2.2.1.3 View Loyalty points Use Case Scenario:

|  |  |
| --- | --- |
| **Foodies\_Sys\_UseCase\_003** | UseCase\_ID |
| View Loyalty points | Scenario Name |
| User | Actors |
| A logged In user view his collected loyalty points. | Brief Description |
| Logged In to system with valid credentials | Pre-condition |
| |  |  | | --- | --- | | **System Response** | **Actor** | | 1.1 The system successfully display the loyalty points. | 1. User select profile page and view Loyalty points from his profile. | | Steps |
| The system display user’s loyalty points if available. | Post- Condition |
| None | Alternative Conditions |

Table 3: View Loyalty points Use Case Scenario.

#### 2.2.1.4 View Profile Page Use Case Scenario:

|  |  |
| --- | --- |
| **Foodies\_Sys\_UseCase\_004** | UseCase\_ID |
| View profile page | Scenario Name |
| User/admin | Actors |
| A logged In user view his profile page | Brief Description |
| Logged In to system with valid credentials | Pre-condition |
| |  |  | | --- | --- | | **System Response** | **Actor** | | 1.1 The system successfully display profile if user is authorized to view it. | 1. User select my profile. | | Steps |
| The system display user’s profile with user’s (User Id , Email , Photo, Loyalty Points). | Post- Condition |
| None | Alternative Conditions |

Table 4: View Profile Page Use Case Scenario.

#### 2.2.1.5 View restaurant information Use Case Scenario:

|  |  |
| --- | --- |
| **Foodies\_Sys\_UseCase\_005** | UseCase\_ID |
| View Restaurant Menu. | Scenario Name |
| User | Actors |
| A logged In user view restaurant Menu. | Brief Description |
| Logged In to system with valid credentials | Pre-condition |
| |  |  | | --- | --- | | **System Response** | **Actor** | | * 1. The system successfully display the restaurant’s information (Phone Number, Name, Menu,   Promotions if exit , etc.). | 1. User select restaurant from list of nearby restaurants to View its information (Phone Number, Name, Menu, promotions if exist, etc.). | | Steps |
| The system display restaurant’s information if available. | Post- Condition |
| None | Alternative Conditions |

Table 5: Restaurant information Use Case Scenario.

#### 2.2.1.6 Make Order Use Case Scenario:

|  |  |
| --- | --- |
| **Foodies\_Sys\_UseCase\_006** | UseCase\_ID |
| Make order | Scenario Name |
| User | Actors |
| A logged In user make order from specific nearby restaurant/ restaurant with offers and promotions. | Brief Description |
| Logged In to system with valid credentials | Pre-condition |
| |  |  | | --- | --- | | **System Response** | **Actor** | | * 1. The system successfully display restaurant’s phone number in a popup message. | 1. User select restaurant from list of nearby restaurants/ restaurants with offers and promotions to make an order. 2. Click confirm order. | | Steps |
| * The system display restaurant’s Phone Number and User’s Loyalty Points Increased By 1. | Post- Condition |
| |  |  | | --- | --- | | **System Response** | **Actor** | | * 1. The system redirect user to Restaurant Page. | 1. User select restaurant from list of nearby restaurants/ restaurants with offers and promotions to make an order. 2. Click cancel. | | Alternative Conditions (1) |
| |  |  | | --- | --- | | **System Response** | **Actor** | | * 1. The system shows pop up message “Congratulations this is your free order ” | 1. User select restaurant from list of nearby restaurants / restaurants with offers and promotions to make an order. 2. Click Make order. | | Alternative Conditions (2) |

Table 6: Make order Use Case Scenario.

#### 2.2.1.7 Browse List of restaurants Use Case Scenario:

|  |  |
| --- | --- |
| **Foodies\_Sys\_UseCase\_007** | UseCase\_ID |
| Browse list of restaurants | Scenario Name |
| User | Actors |
| A logged In user browse list of nearby restaurants by clicking on Next/Previous button in each page. | Brief Description |
| Logged In to system with valid credentials | Pre-condition |
| |  |  | | --- | --- | | **System Response** | **Actor** | | * 1. The system successfully display list of nearby restaurant each 5 restaurants in a page.   2.1 The System display Next/Previous Page. | 1. User agreed to access his location 2. User click Next/Previous button to navigate between restaurants. | | Steps |
| The system display Next/Previous page based on the user’s choice. | Post- Condition |
| None | Alternative Conditions |

Table 7: Browse list of restaurants Use Case Scenario.

#### 2.2.1.8 Search restaurant Use Case Scenario:

|  |  |
| --- | --- |
| **Foodies\_Sys\_UseCase\_008** | UseCase\_ID |
| Search restaurant | Scenario Name |
| User | Actors |
| A logged In user search for a specific restaurant. | Brief Description |
| Logged In to system with valid credentials | Pre-condition |
| |  |  | | --- | --- | | **System Response** | **Actor** | | 1.1 The system successfully retrieve all restaurants that match that name. | 1. User write restaurant name (Substring of restaurant name) in a search box field. 2. Click search. | | Steps |
| The system display restaurants with same name. | Post- Condition |
| None | Alternative Conditions |

Table 8: Search restaurant Use Case Scenario.

#### 2.2.1.9 Delete User Use Case Scenario:

|  |  |
| --- | --- |
| **Foodies\_Sys\_UseCase\_009** | UseCase\_ID |
| Delete User | Scenario Name |
| User | Actors |
| An Admin of the system search for user’s account and delete. | Brief Description |
| Logged In to system with valid ADMIN credentials | Pre-condition |
| |  |  | | --- | --- | | **System Response** | **Actor** | | * 1. System displays a page with search box and Search/Delete buttons.   3.1 System retrieve user’s account that match this ID.  4.1 System Successfully delete the user’s account. | 1. Admin select delete user. 2. Admin fill-in search box with valid user id. 3. Click Search. 4. Click Delete. | | Steps |
| The system delete user’s account successfully. | Post- Condition |
| |  |  | | --- | --- | | **System Response** | **Actor** | | * 1. System displays a page with search box and Search/Delete buttons.   2.1 System display a message that “this user id not correct “.  4.1 System retrieve user’s account that match this ID.  5.1 System Successfully delete the user’s account.  4.1 System Successfully delete the user’s account. | 1. Admin select delete user. 2. Admin fill-in search box with invalid user id. 3. Admin Re-fill search box with valid User Id. 4. Click Search. 5. Click Delete. | | Alternative Conditions |

Table 9: Delete User Use Case Scenario.

#### 2.2.1.10 Add Restaurant Use Case Scenario:

|  |  |
| --- | --- |
| **Foodies\_Sys\_UseCase\_010** | UseCase\_ID |
| Add Restaurant | Scenario Name |
| Admin | Actors |
| A logged In Admin add new restaurant and its menu. | Brief Description |
| Logged In to system with valid ADMIN credentials | Pre-condition |
| |  |  | | --- | --- | | **System Response** | **Actor** | | 3.1 The system successfully add new restaurant. | 1. Admin write restaurant name in English or Franco , Phone , location and   Promotions if exit.   1. Enter items of Menu (Item by item) and its price. 2. Click Add. | | Steps |
| The system add new restaurant and its detailed menu. | Post- Condition |
| |  |  | | --- | --- | | **System Response** | **Actor** | | * 1. The system display message “ Restaurant already exist “   2. The System clear the search box.   4.1- The System verify  The new name.  4.2- System successfully add new restaurant. | 1. Admin write restaurant name. 2. Admin re-write invalid new restaurant name. 3. Enter items of Menu (Item by item) and its price. 4. Click Add. | | Alternative Conditions |

Table 10: Add Restaurant Use Case Scenario.

#### 2.2.1.11 Log out Use Case Scenario:

|  |  |
| --- | --- |
| **Foodies\_Sys\_UseCase\_007** | UseCase\_ID |
| Log out from the system | Scenario Name |
| User/Admin | Actors |
| A logged In user/Admin can logout from the system. | Brief Description |
| Logged In to system with valid credentials | Pre-condition |
| |  |  | | --- | --- | | **System Response** | **Actor** | | * 1. The system successfully logged out the user/admin from the system. | 1. User click on logout from navigation bar. | | Steps |
| The system logged user/admin out from the system. | Post- Condition |
| None | Alternative Conditions |

Table 11: Log out Use Case Scenario.

## Wireframes

### 2.3.1 Login Page

**ID:** Foodies\_SYS\_Wireframe\_001

**Description:**

This is the first page to appear when navigating to the website URL, any user with a registered account can log in

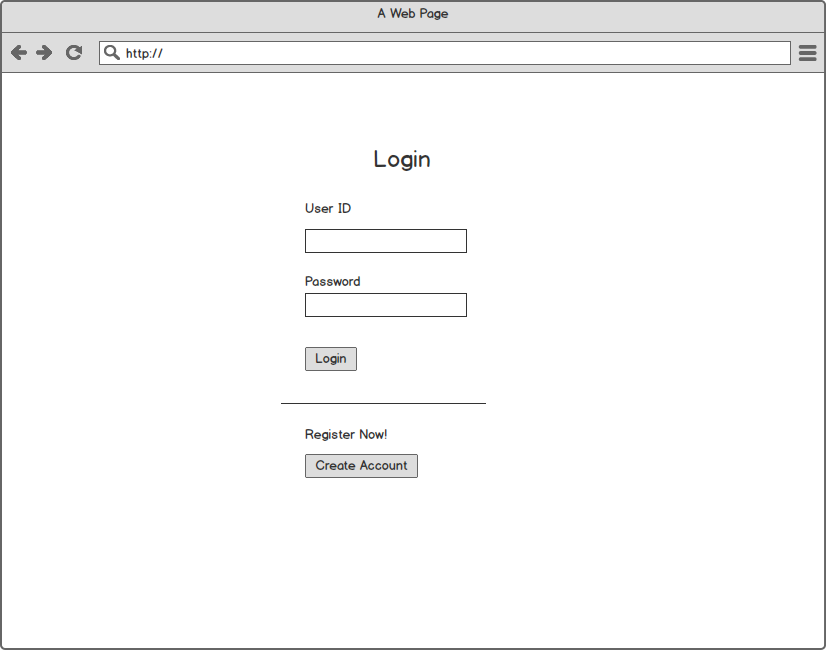


Figure 3: Foodies\_Sys\_LoginPage

### 2.3.2 Create Account Page

**ID:** Foodies\_SYS\_Wireframe\_002

**Description:**

User with no registered account can navigate to the “Create Account” Page and register with a User ID, an email, location, gender and a password.

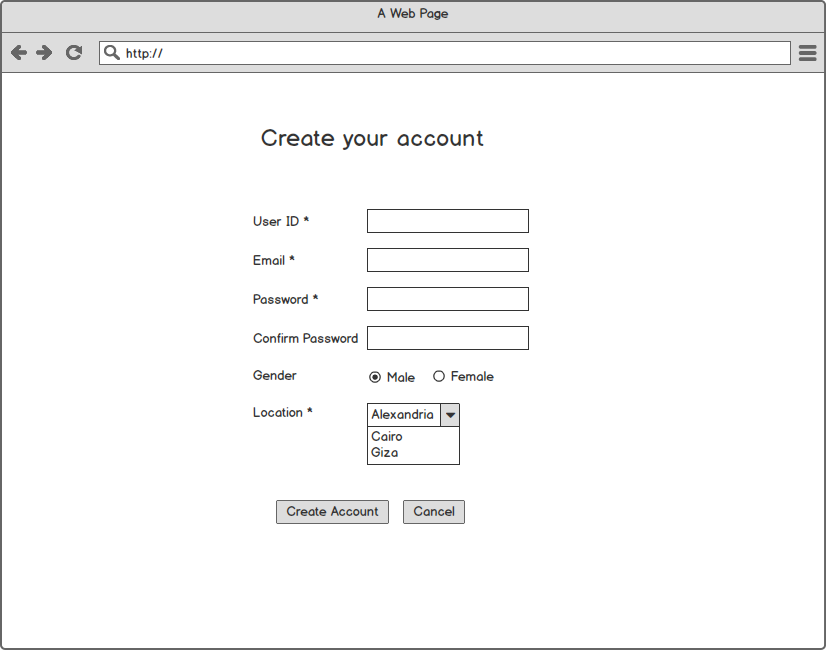


Figure 4: Foodies\_Sys\_CreateAccountPage

### 2.3.3 Homepage

**ID:** Foodies\_SYS\_Wireframe\_003

**Description:**

A logged-in user can view this page.

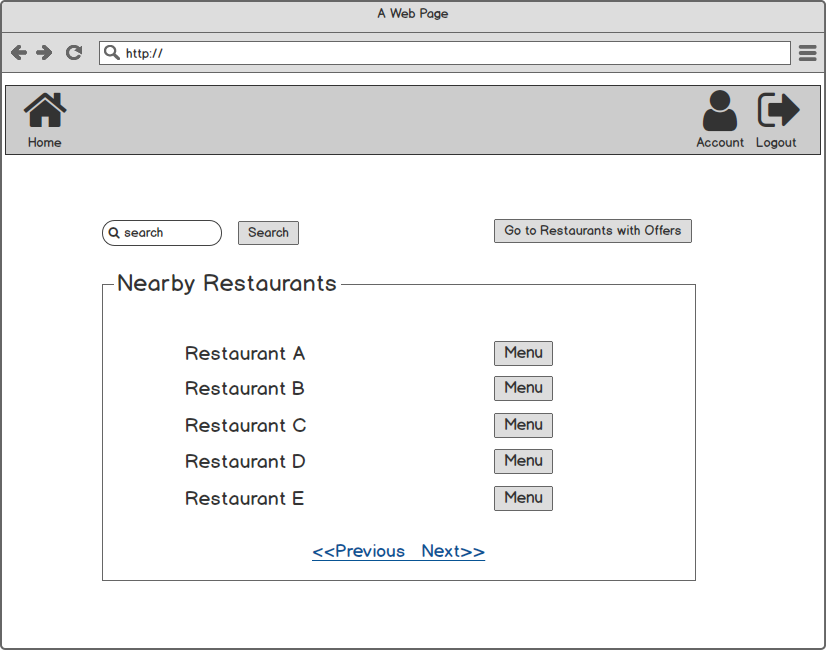


Figure 5: Foodies\_Sys\_Homepage

### 2.3.4 Offers&Promotions Page

**ID:** Foodies\_SYS\_Wireframe\_004

**Description:**

A logged-in user can view this page, a pop-up message with the restaurant phone number shall appear when the user click on “Order” Button.

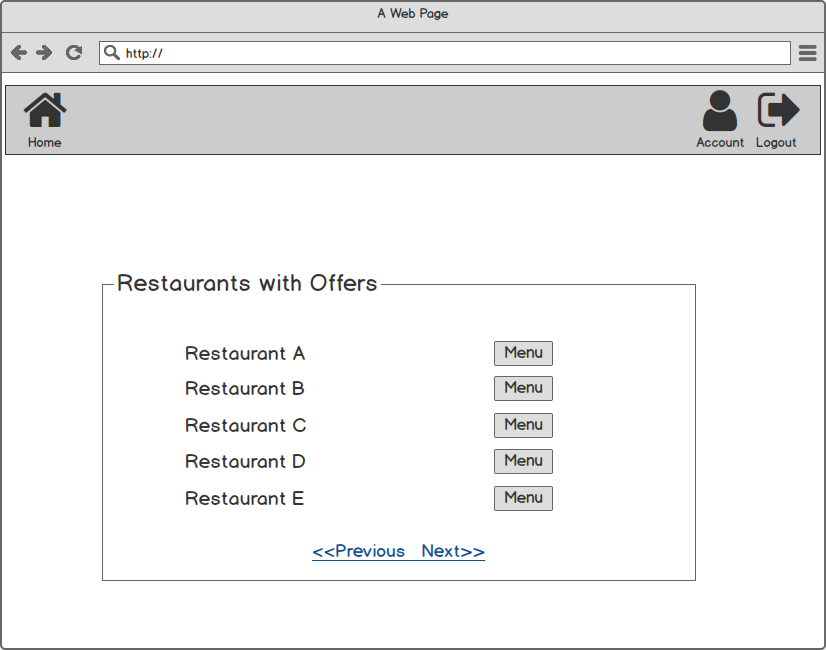


Figure 6: Foodies\_Sys\_Offers&PromosPage

### 2.3.5 Menu Page

**ID:** Foodies\_SYS\_Wireframe\_005

**Description:**

A logged-in user can view this page with the restaurant’s menu and set an order.

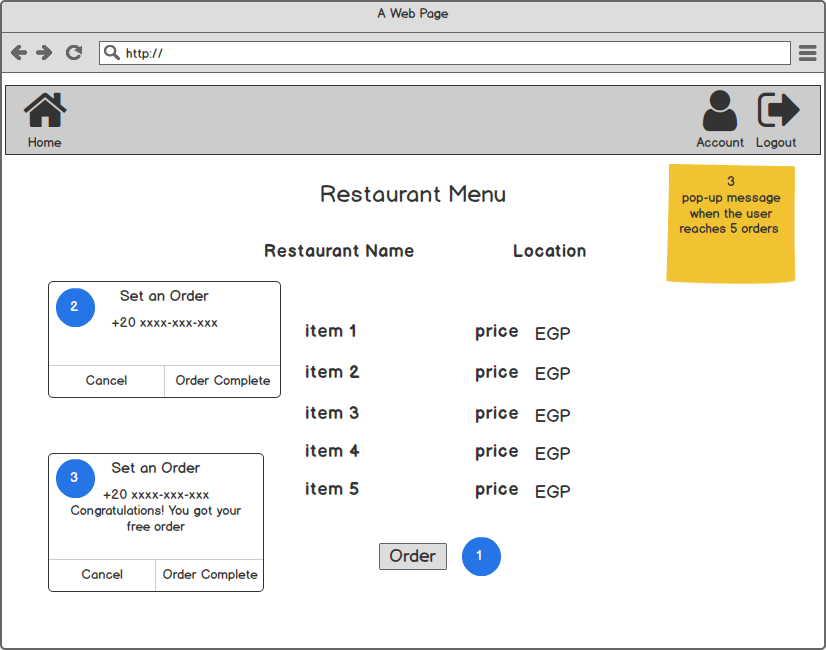


Figure 7: Foodies\_Sys\_MenuPage

### 2.3.6 User Account Page

**ID:** Foodies\_SYS\_Wireframe\_006

**Description:**

A logged-in user can view his profile showing his user ID, email, loyalty points and free orders if there’s any.

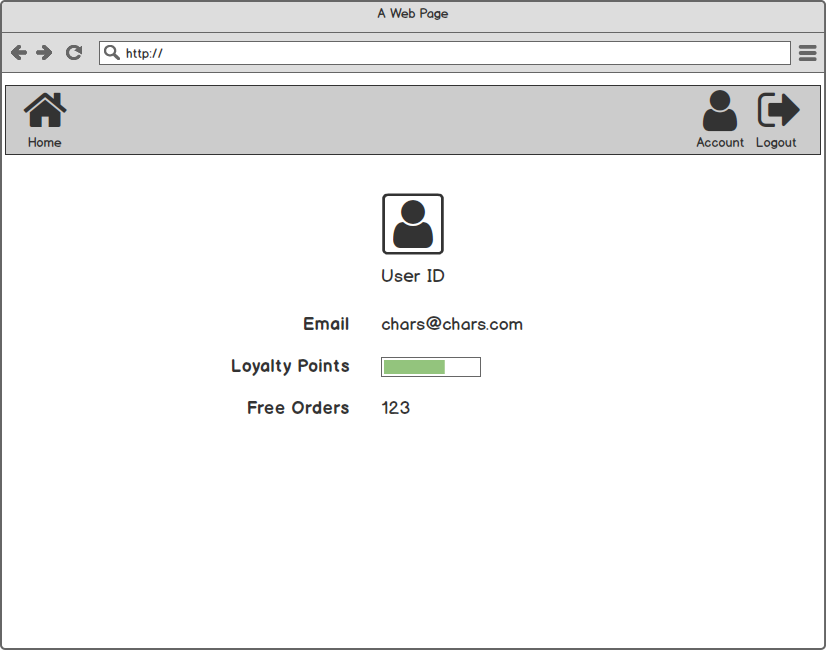


Figure 8: Foodies\_Sys\_UserAccountPage

### 2.3.7 Admin Panel – Delete User Page

**ID:** Foodies\_SYS\_Wireframe\_007

**Description:**

Admin can delete users by providing a user ID, click on “Search” then click on “Delete” Button

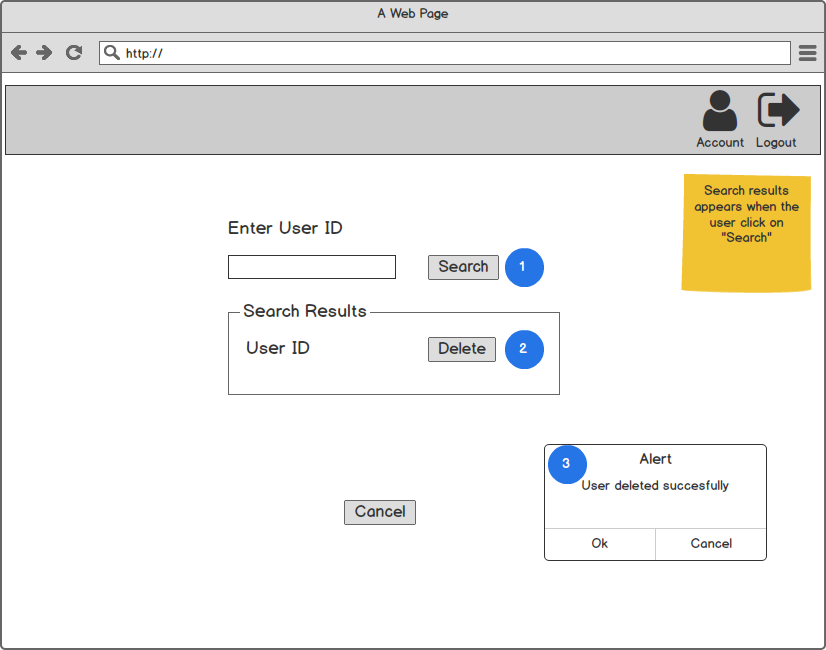


Figure 9: Foodies\_Sys\_Admin\_DeleteUserPage

### 2.3.8 Admin Panel – Add Menu Page

**ID:** Foodies\_SYS\_Wireframe\_008

**Description:**

Admin can view this page, add a new restaurant with its menu and its logo then submit it to the system.

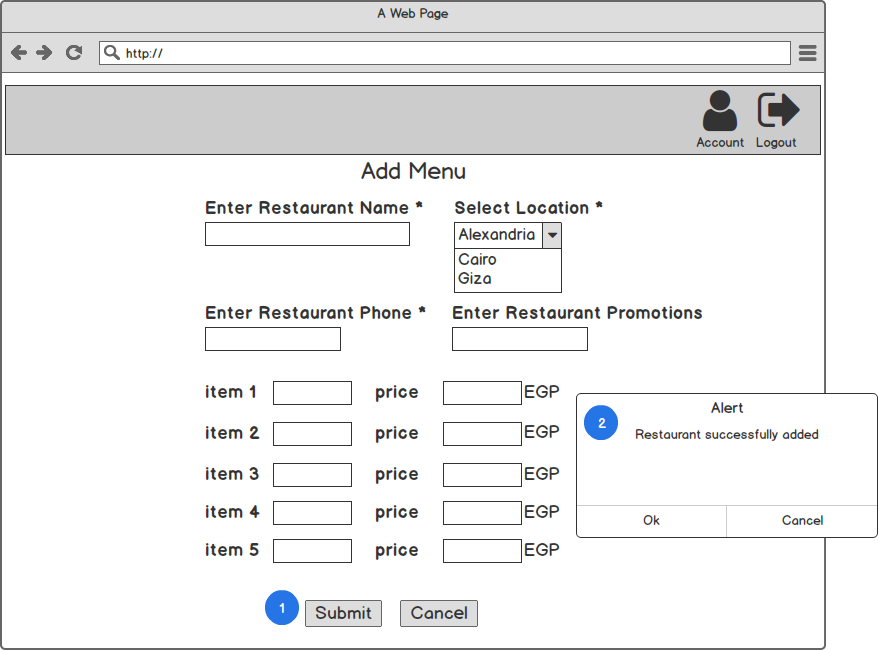


Figure 10: Foodies\_Sys\_Admin\_AddMenuPage

### 2.3.9 Admin Profile

**ID:** Foodies\_SYS\_Wireframe\_009

**Description:**

Admin can view this page, access “delete user page” and “add restaurant page”.

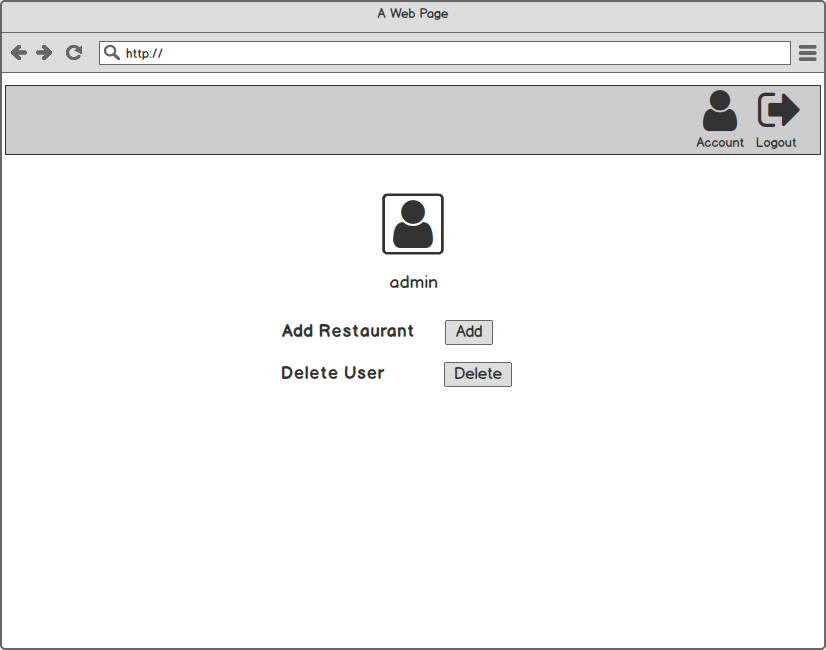


Figure 11: Foodies\_Sys\_AdminProfile

## 2.4 Data Flow Diagram:

**Description:**

Customer fill registration form the system check if the customer register before or not, after that customer login and system access customer location and shows nearby restaurants then customer can show menu or set an order and also customer can check promotions and offers before set an order.

Admin fill login form and admin also can Add menu and delete user but before delete user’s system check if this user existing to delete or not.

****

Figure 12 Foodies\_Sys\_DFD

## 2.5 Class Diagram:



Figure 13 Foodies\_Sys\_ClassDiagram

## 2.6 Sequence Diagram:

#### 2.6.1 Admin Panel – Delete User’s Profile



Figure 14 Foodies\_seq\_Admin\_delete

#### 2.6.2 Admin Panel – Add New Restraunt

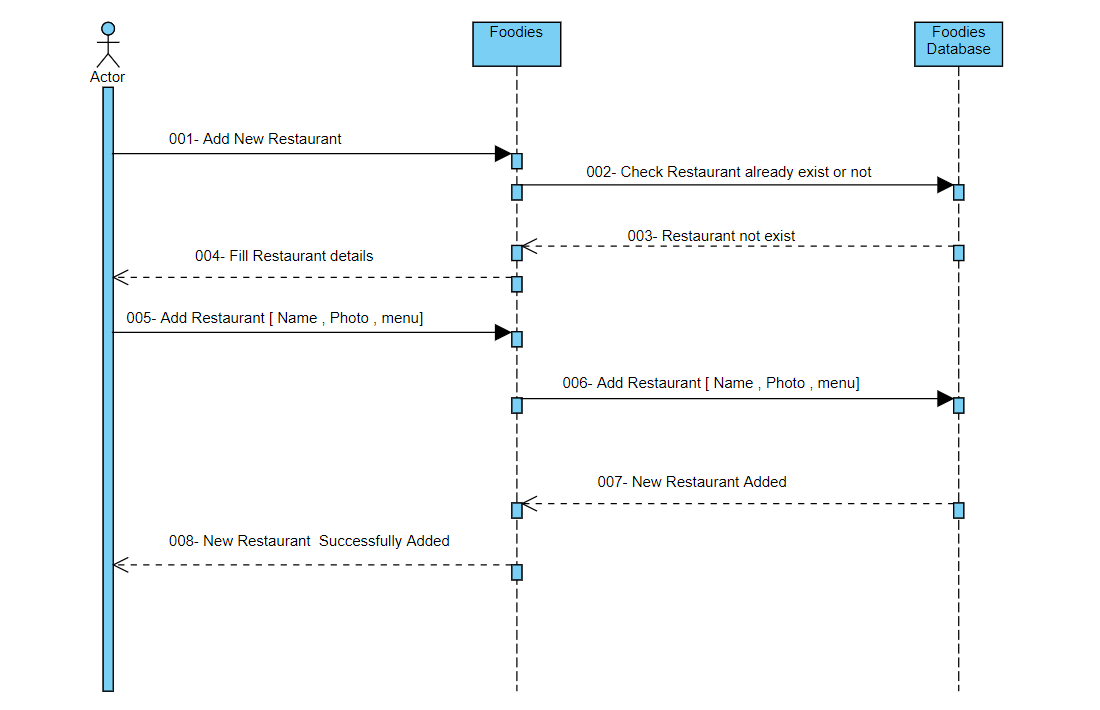


Figure 15 Foodies\_sys\_seq\_admin\_addrestaurant

#### 2.6.3 User Sequence Diagram

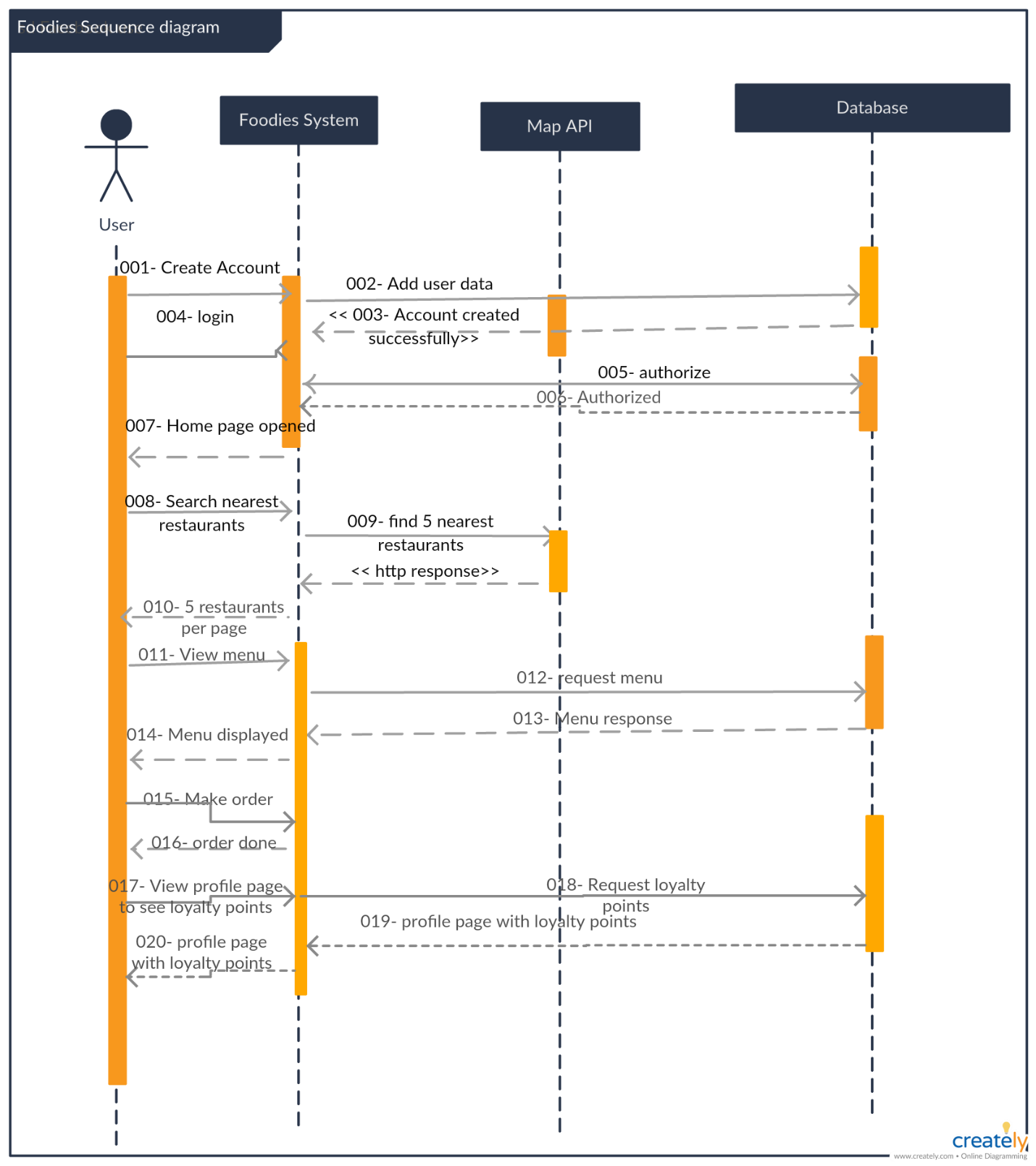


Figure 16Foodies\_seq\_User

## 2.7 ERD Diagram:

