



Addis Ababa Institute of Technology  
Department of IT/SW Eng.

**Yet[Lebla]**  
(version 1.0)

" Find the best food with cheaper price!"

## **Software Requirements Specification**

### Team Members

1. Tibebeselasie Mehari
2. Robel Ephraim
3. Natnael Sisay
4. Yosef Worku

Supervisor Name Mr. Natnael Argaw

## Revision History

Date	Description	Author	Comments
March 23	Version 1	Yet[Lebla]	Draft

## Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Signature	Printed Name	Title	Date

## Table of Contents

<b>DOCUMENT APPROVAL.....</b>	<b>II</b>
<b>LIST OF FIGURES.....</b>	<b>IV</b>
<b>DEFINITIONS, ACRONYMS, AND ABBREVIATIONS.....</b>	<b>ERROR! BOOKMARK NOT DEFINED.</b>
<b>DECLARATION .....</b>	<b>ERROR! BOOKMARK NOT DEFINED.</b>
<b>1. INTRODUCTION .....</b>	<b>ERROR! BOOKMARK NOT DEFINED.</b>
1.1 BACKGROUND .....	1
1.2 OBJECTIVE.....	<b>ERROR! BOOKMARK NOT DEFINED.</b>
1.3 SCOPE OF THE PROJECT .....	2
1.3 LIMITATIONS .....	2
<b>2. REQUIREMENTS .....</b>	<b>3</b>
2.1 FUNCTIONAL REQUIREMENTS .....	4
2.2 NON - FUNCTIONAL REQUIREMENTS.....	5
<b>3. SYSTEM MODELS .....</b>	<b>6</b>
<b>4. USAGE SCENARIOS .....</b>	<b>9</b>
<b>5. UI SAMPLES .....</b>	<b>27</b>
<b>4. OTHER REQUIREMENTS.....</b>	<b>33</b>
<b>4. CHANGE MANAGEMENT PROCESS .....</b>	<b>34</b>
<b>REFERENCES.....</b>	<b>40</b>

## List of figures

Figure 1 System Landing page (Home Page).....	28
Figure 2 System Landing page (Home Page) .....	28
Figure 3 Menus Item List .....	29
Figure 4 Page Footer.....	29
Figure 5 Food List.....	30
Figure 6 Delicious List .....	30
Figure 7 Menu List... ..	30
Figure 8 Sign Up.....	31
Figure 9 Login .....	31

## *Acronyms*

- **SRS**                      **S**oftware **R**equirements **S**pecification
- **AAiT**                     **A**ddis **A**baba **I**nstitute of **T**echnology
- **FR**                        **F**unctional **R**equirement
- **QA**                        **Q**uality **A**tttribute
- **UC**                        **U**se **C**ase
- **OOP**                     **O**bject **O**riented **P**rogramming
- **API**                     **A**pplication **P**rogramming **I**nterface
- **JS**                        **J**ava**S**cript
- **HTML**                  **H**yper **T**ext **M**arkup **L**anguage
- **CSS**                     **C**ascading **S**tylesheet
- **JSON**                  **J**ava**S**cript **O**bject **N**otation
- **OS**                        **O**perating **S**ystem
- **UI**                        **U**ser **I**nterface
- **URL**                    **U**niform **R**esource **L**ocator
- **HTTP**                  **H**yper **T**ext **T**ransfer **P**rotocol

## DECLARATION

We declare that this SRS document reflects the idea of our team in our own words. If there are words or any other form of information included from other sources, we assure that it has been mentioned in the Reference section of the SRS document. We also declare that our team have prepared this document under the principle of AAiT.

Our team has not made any effort of cheating or illegal action that may violate any rules and regulations of the campus. All statements in this document are written as clear as possible to avoid any ambiguity and all statements are not falsified or untrue as well.

**Yet[lebla]** team

Date: 16-03-2018

## 1. Introduction

### 1.1 Background

Human beings rely on their communicational skills, memory and intelligence to find new places to eat that offers high quality food with cheap price. Unfortunately, eating out at random places to find quality food is not only costly but also tiresome, boring and most importantly time-consuming. Not only that, users might assume they found some place that is the cheapest of them all, effectively stopping them from exploring more places that do serve similar places for even cheaper prices.

For these reason, this project aims at developing a web-based system that would make the process of finding such places easier and more efficient. Additionally, only foods that are allowed to be consumed by the users (according to their belief, customs & traditions) and places that serve them will be available to each user, ensuring users preferences.

### 1.2 Objectives

As mentioned in the above section of the document(Introduction), people encounter many problems when they rely only on their judgment and social skills. To mention some of these ordeals they face:

- ✓ *It is hard to keep a balanced diet*
- ✓ *It is easy to waste time looking for foods in this hectic world.*
- ✓ *People do need to memorize places with their food menu in order to pick places.*
- ✓ *It is often hard to know (or guess) whether a certain place do serve a food that doesn't contradict with one's beliefs and norms.*

- ✓ *It is very time consuming and boring thing to do.*

These are just some examples of the problems one could face when finding a good place is done manually.

This system aims at mitigating those problems by:

- ✓ Offering a simple and good UI (User Interface)
- ✓ Taking user's belief, preferences and geographical coordination into consideration.
- ✓ Allowing users to access the system using their hand-held android devices.

**Target Audience:** This document is intended for any individual user, developer, or program tester who needs to understand the basic features of the program.

### 1.3 Scope of the project

The purpose of this document is **outlining** the requirements of this system. A web-based software system built on a **recommendation** model which assists users in picking and sharing places to eat that offers **quality food** with **cheap prices**.

## I. System Administrators

### Description:

Systems Administrators is a group with the most privileged powers in the system.

### Roles (what they can do):

- ✓ Warn users
- ✓ Broadcast a message to all users
- ✓ Ban users



- ✓ Delete data
- ✓ Manage (add, revoke) privileges for other users.

## **II. Content Moderators**

### **Description:**

Content Moderator is second privileged users group in the system. Their main role is to verify the authenticity of the data in the system.

### Roles (what they can do):

- ✓ Handle any invalid data (incl. spam) reports from users
- ✓ Report users, that need to be banned (or warned), to System Administrators
- ✓ Report invalid or spam data to System Administrators.
- ✓ Verify the authenticity of data provided from "place owners" group.

## **III. Place Owners**

### **Description:**

Users in this group are owners of places (restaurants, cafes, hotels etc...)

### Roles (what they can do):

- ✓ Manage the place's menu.
- ✓ Manage basic information (like phone number, location, branches, etc..)

## **IV. Customers**

### **Description:**

This group is the least privileged group and where all normal users are. When registering to the system, user will be put into this users group.

#### Roles (what they can do):

- ✓ Search for foods and places.
- ✓ Share foods information.
- ✓ Report spams and invalid data to content moderators.

### **1.4. Limitations**

Since the system is web-based, an **Internet** connection is necessary. The speed the user experiences greatly depends on the **Internet** connection speed between the back-end (server) and the front-end (users browser).

## **2. Requirements**

### **2.1 Functional Requirements**

- 2.1.1.1** Users share foods by providing information about where, when and for how much they had a certain food along with pictures.
- 2.1.1.2** The system must take the user's religious belief into consideration when recommending foods.
- 2.1.1.3** Users can search for foods they are interested in.
- 2.1.1.4** Users can delete or edit foods within certain(fixed) time limit if invalid information was provided.
- 2.1.1.5** Authorized users can moderate food data provided by users and have the authority to delete those data as well as the user accounts if they are found to provide spam or incorrect data.
- 2.1.1.6** The system will read users GPS coordinates to yield nearest places first when recommending places.
- 2.1.1.7** The system provides directions on how to use the system.

### **2.2 Non-Functional Requirements (Quality Attributes)**

### 2.2.1 UI (User Interface)

The system should have user-friendly interface with ease in navigation in-between the major sections of the system by using menus backward navigation support. Additionally, descriptive and user-friendly messages for invalid inputs, errors, confirmation should be displayed by the system to let know the user know what exactly is going on.

### 2.2.2 Security

Only authorized users (who own legit account) can access the system. The system also assigns permissions or privileges to different user groups to ensure data integrity and accountability.

### 2.2.3 Miscellaneous

The system should be:

- ✓ Cross-browser compatible.
- ✓ Mobile-friendly (responsive)
- ✓ Able to offer an option to users to decrease too much imaging and graphics as it might be bad for metered connections (in terms of bandwidth consumption).

## 3. System Models

### 3.1 Use Case Model

#### 3.1.1 Actors

The following are actors identified in the System

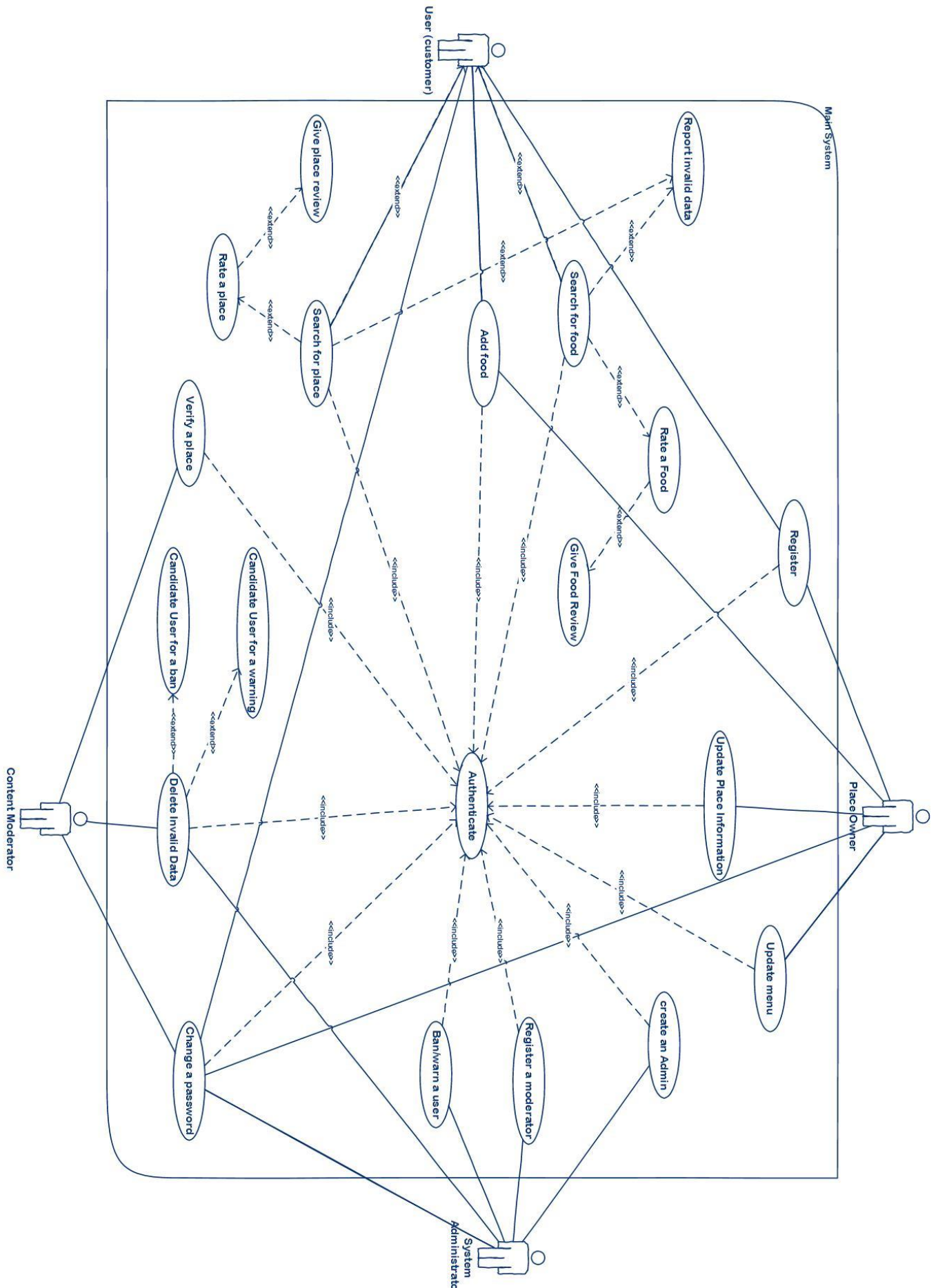
- ✓ **System Administrator:** represents a person, with the highest privilege in the system, that manages users and their privileges.
- ✓ **Content Moderator:** represents a person who checks data authenticity.
- ✓ **Place Owner:** represents a person who owns a business (where other users can eat at)
- ✓ **User:** represents normal end-user of the system.

### 3.1.2 Use Cases

The following are use cases identified in the system.

1. Authenticate User
2. Search for food
3. Search for place
4. Add Food
5. Rate a food
6. Rate a place
7. Give food reviews
8. Give place reviews
9. Report Invalid Data
10. Register
11. Update Menu
12. Update place information
13. Verify a place
14. Report user to System Admin
15. Register a Moderator
16. Create an Admin
17. Change a password
18. Delete Invalid data

19. Ban/Warn a user



#### 4. Usage Scenarios:

##### UC-01: Authenticate User

**Use case name:** UC-01: Authenticate User

**Primary Actors:** Ordinary User, Place Owners, Content Moderators, System Administrators

**Summary:** The system will authenticate the user.

**Preconditions:** The user has a valid account.

**Post-conditions:** The user is granted access to the system with his/her associated privilege.

**Trigger:** The user wants to use the system.

##### Main Success Scenario

1. The user navigates to the login page of the system.
2. The user provides his/her username and password.
3. The user clicks on "Login".
4. The system validates the entered information.
5. The system verifies the authenticity of the account.
6. The system grants the user an access to the system.
7. UC-01 ends.

##### Extensions

- 5a. Connection to the database server fails while trying to fetch the account.
- 5a1. Prompt the user to retry.

##### Alternative Paths

**Alternate Course A:** The entered information is incorrect.

A.4. The system verifies that the entered information is incorrect.

A.5. The system shows an error message indicating the error.

A.6. The system informs the user to re-enter the correct information.

A.7. The user enters the information again and click on "login".

A.8. Back to step 5 of the main success scenario

**Alternate Course B:** The entered information is invalid.

B.5. The system validates that the entered information and determine the information is invalid.

B.6. The system shows an error message indicating the error

B.7. The system informs the user to re-enter the correct information.

B.8. The user enters the information again and click on "login".

B.9. Back to step 4 of the main success scenario

#### UC-02: Search for food

**Use case name:** UC-02: Search for food

**Primary Actors:** Ordinary User

**Summary:** The Users finds a food s/he is looking for.

**Preconditions:** The user has authenticated.

**Post-conditions:** The user lands on food's detailed page.

**Trigger:** The user wants to view detailed information about a food.

#### **Main Success Scenario**

1. The user clicks on "look for a food".
2. The system displays a form to search for a food by name.
3. The user provides the required information and submit.
4. The system displays search results.
5. The user clicks on the desired food.
6. The system display a detailed information of the selected food.
7. UC-02 ends.

#### **Extensions**

- 4a. Connection to the database server fails while trying to fetch the results.
  - 4a1. Prompt the user to retry.

#### **Alternative Paths**

**Alternate Course A:** No results found with the name provided.



- A.4. The system informs the user that no results were found with keyword s/he provided.
- A.5. The system informs the user to try searching again with a different keyword.
- A.6. Back to step 3 of the main success scenario

**Alternate Course B:** The user filters and/or sort the results.

- B.5. The user alters the filter/sort form.
- B.6. The system automatically applies the changes to the search results.
- B.7. Back to step 5 of the main success scenario

#### UC-03: Search for place

**Use case name:** UC-03: Search for place

**Primary Actors:** Ordinary User

**Summary:** The Users finds a place s/he is looking for.

**Preconditions:** The user has authenticated.

**Post-conditions:** The user lands on place's detailed page.

**Trigger:** The user wants to view detailed information about a place.

#### **Main Success Scenario**

1. The user clicks on "look for a place".
2. The system displays a form to search for a place by name.
3. The user provides the required information and submit.
4. The system displays search results.
5. The user clicks on the desired food.
6. The system display a detailed information of the selected food.
7. UC-03 ends.

#### **Extensions**

- 4a. Connection to the database server fails while trying to fetch the results.
- 4a1. Prompt the user to retry.

#### **Alternative Paths**

**Alternate Course A:** No results found with the name provided.

A.4. The system informs the user that no results were found with information s/he provided.

A.5. The system informs the user to try searching again with a different information.

A.6. Back to step 3 of the main success scenario

**Alternate Course B:** The user filters and/or sort the results.

B.5. The user alters the filter/sort form.

B.6. The system automatically applies the changes to the search results.

B.7. Back to step 5 of the main success scenario

#### UC-04: Add Food

**Use case name:** UC-04: Add Food

**Primary Actors:** Ordinary User, Place Owner

**Summary:** The new Food gets added to the system.

**Preconditions:** The user has authenticated.

**Post-conditions:** The Food is added to the system.

**Trigger:** The user wants lands on "Add new Food" page through navigation or other mechanisms (by entering the URL, etc.).

#### **Main Success Scenario**

1. The user lands on "add new food" page.
2. The system displays a form.
3. The user provides the required information and click on "next".
4. The system verifies the data entered.
5. The User uploads a picture of the food and click on "next".
6. The User shares GPS locations by allowing the system to gather the information and click on "next".

7. The system displays list of food categories for the user to choose from.
8. The user selects the appropriate categories and click on "submit".
9. The system verifies the information provided.
10. The system displays confirmation message to the user.
11. UC-04 ends.

### Alternative Paths

**Alternate Course A:** The data entered was not in the correct format.

- A.4. The system verifies that the data format is not valid.
- A.5. The system informs the user about the problem by showing error messages.
- A.6. The system tells the user to try again.
- A.7. Back to step 3 of the main success scenario

**Alternate Course B:** The Users platform doesn't have a support for GPS access (or no permission is granted).

- B.6. The system shows a map and an input to type in.
- B.7. The user finds the place in the map by providing the name of the place.
- B.8. Back to step 7 of the main success scenario

**Alternate Course C:** The User selects contradicting categories together (or no category is selected).

- C.9. The system informs the user about the problem by showing error message(s).
- C.10. Back to step 8 of the main success scenario

### UC-05: Rate a food

**Use case name:** UC-05: Rate a food

**Primary Actors:** Ordinary User

**Summary:** The food gets rated.

**Preconditions:** The user has authenticated and is on food's detail page.

**Post-conditions:** The Food's rating gets updated.

**Trigger:** The user who is on "Food detail" page decided to rate the food (based on personal preferences).

**Main Success Scenario**

1. The system displays a control for rating (slider or stars, based on the browser support)
2. The user provide the information.
3. The rating for the food gets updated.
4. The system prompts the user to provide a review (optional)
5. UC-05 ends.

**Alternative Paths**

**Alternate Course A:** The data entered was not in the range of the rating.

- A.3. The system verifies that the data format is not valid.
- A.4. The system informs the user about the problem by showing error message(s).
- A.5. The system tells the user to try again.
- A.6. Back to step 2 of the main success scenario

**Alternate Course B:** The User decided to give a review.

- B.4. To UC-7: Give food Review

**UC-06: Rate a place**

**Use case name:** UC-06: Rate a place

**Primary Actors:** Ordinary User

**Summary:** The place gets rated.

**Preconditions:** The user has authenticated and is on place's detail page.

**Post-conditions:** The Place's rating gets updated.

**Trigger:** The user who is on "Place detail" page decided to rate the place (based on their service and prices).

**Main Success Scenario**

1. The system displays a control for both service and prices rating (slider or stars, based on the browser support)
2. The user provide the information.
3. The rating for the place gets updated.
4. The system prompts the user to provide a review (optional)
5. UC-06 ends.

**Alternative Paths**

**Alternate Course A:** The data entered was not in the range of the rating.

- A.3. The system verifies that the data format is not valid.
- A.4. The system informs the user about the problem by showing error message(s).
- A.5. The system tells the user to try again.
- A.6. Back to step 2 of the main success scenario

**Alternate Course B:** The User decided to give a review.

- B.4. To UC-8: Give place Review

**UC-07: Give food reviews**

**Use case name:** UC-07: Give food reviews

**Primary Actors:** Ordinary User

**Summary:** The food gets a new review.

**Preconditions:** The user has authenticated and is on food's detail page.

**Post-conditions:** The food's reviews get updated (the new review gets included).

**Trigger:** The user who is on "Food's detail" page decided to leave a review about the food (based on personal preference or experience).

**Main Success Scenario**

1. The system displays a form.

2. The user provide the information prompted and click on "Give Review".
3. The system displays a confirmation message to user.
4. UC-07 ends.

#### UC-08: Give place reviews

**Use case name:** UC-08: Give place reviews

**Primary Actors:** Ordinary User

**Summary:** The place receives a new review.

**Preconditions:** The user has authenticated and is on the "Place's Detail Page".

**Post-conditions:** The place's reviews get updated (the new review gets added).

**Trigger:** The user who is on "Place's detail" page decided to give a comment about the place (for other users to see).

**Main Success Scenario**

1. The system displays a form.
2. The user provide the information prompted and click on "Give Review".
3. The system displays a confirmation message to user.
4. UC-08 ends.

#### UC-09: Report Invalid Data

**Use case name:** UC-09: Report Invalid Data

**Primary Actors:** Ordinary User

**Summary:** The data gets reported to privileged users to further investigate.

**Preconditions:** The user has authenticated and is either on place's or food's detail page.

**Post-conditions:** The problem will be reported for authorized persons to see.

**Trigger:** The user who is either on “Place’s detail” or “Food’s Detail” page clicks on “Report” button that’s placed on either the food itself or on each review.

**Main Success Scenario**

1. The system displays a form for the user to fill.
2. The user provide all the necessary information and click “submit”.
3. The system responds with confirmation prompt.
4. The user confirms.
5. The system informs the user that the report was successfully filed.
5. UC-09 ends.

**Alternative Paths**

**Alternate Course A:** Some (or all) of the required information were missing.

- A.3. The system informs the user about the problem by showing error message(s).
- A.4. The system tells the user to try again.
- A.5. Back to step 2 of the main success scenario

**Alternate Course B:** The user didn’t confirm.

- B.5. The system notifies the user that the operation is canceled.
- B.6. To Step 1 of the main success scenario

UC-10: Register

**Use case name:** UC-10: Register

**Primary Actors:** Ordinary User, Place owner

**Summary:** The user gets registered to the system.

**Preconditions:** None

**Post-conditions:** The user/place owner is registered on the system.

**Trigger:** The user lands on the register page through navigation or by clicking “create a new account” link from the login page.

**Main Success Scenario**

1. The system displays an option between registering as an ordinary user or as a place owner.
2. The user selects "ordinary user".
3. The user provide all the necessary information and click on "register".
4. The system responds with confirmation prompt.
5. UC-10 ends.

### Alternative Paths

**Alternate Course A:** User selects a place owner.

- A.3. The user provides all the necessary information and click on "register".
- A.4. The system notifies the user that it is registered and that it will be verified once the content moderators confirm it).
- A.5. Back to step 5 of the main success scenario

### UC-11: Update Menu

**Use case name:** UC-11: Update Menu

**Primary Actors:** Place owner

**Summary:** The menu for the place gets updated.

**Preconditions:** The user must be authenticated.

**Post-conditions:** The place's menu gets updated.

**Trigger:** The User clicks on "Edit/Update Menu" link on the place's detail page of the place.

### Main Success Scenario

1. The system displays a menu with "add", "edit", "delete" options.
2. The user selects "add".
3. The system shows a form to add a new menu.
4. The user provide all the necessary information and click on "add".
5. The system process the data and validate it.
6. The user uploads photo of the menu.



7. The system prompts the user (place owner) to confirm the operation.
8. The user confirms.
9. The system update the menu information of the place.
10. The system notifies the user about the success of the operation.
11. UC-11 ends.

### **Alternative Paths**

**Alternate Course A:** User selects "edit menu".

- A.2. The user chose "edit".
- A.3. The system responds with list of menus available for the place.
- A.4. The user selects a menu.
- A.5. The system displays a form to edit the selected menu.
- A.6. The user provides all the necessary information and click "Edit".
- A.7. The system processes (and validates) the user input.
- A.8. Back to step 7 of the main success scenario

**Alternate Course B:** User selects "delete menu".

- B.2. The user chose "delete".
- B.3. The system responds with list of menus available for the place the user owns.
- B.4. The user (place owner) selects a menu.
- B.5. Back to step 7 of the main success scenario

**Alternate Course C:** User didn't confirm the operation.

- C.8. The user cancels the confirmation.
- C.3. The system responds with list of menus available for the place the user owns.
- C.4. The user (place owner) selects a menu.
- C.5. Back to step 7 of the main success scenario

## **UC-12: Update Place Information**

**Use case name:** UC-12: Update Place Information

**Primary Actors:** Place owner

**Summary:** The basic information about the place gets changed.

**Preconditions:** The user must be authenticated.

**Post-conditions:** The place's information gets updated.

**Trigger:** The User (place owner) clicks on "Edit Basic Information" link on the place's detail page of the place.

**Main Success Scenario**

1. The system displays a form with pre-existing data for the user to edit.
2. The user edits the form.
3. The user submits the form.
4. The system validates the form.
5. The system prompts the user (place owner) to confirm the operation.
6. The user confirms.
7. The system updates the place's information.
8. The system shows a confirmation message back to the user indicating the operation was successful.
9. The user gets navigated to the "place's detail page"
10. UC-12 ends.

**Alternative Paths**

**Alternate Course A:** User cancels the form.

A.6. The user decided not to edit the information after all (cancels the operation).

A.7. The system notifies the user that the operation was canceled successfully.

A.8. Back to step 9 of the main success scenario

**Alternate Course B:** The system detected invalid data.

B.5. The system informs the problems with error messages.

B.6. Back to step 1 of the main success scenario

UC-13: Verify a place

**Use case name:** UC-13: Verify a place

**Primary Actors:** Content Moderator

**Summary:** Content moderators verify the authenticity of a place (business).

**Preconditions:** The user must be authenticated.

**Post-conditions:** A place gets verified (is now a trusted source).

**Trigger:** The Content Moderators navigate to "Verify Registrations" page through either navigation provided by the system or using URL.

**Main Success Scenario**

1. The systems list all recent place registration attempts to the content moderator.
2. The user (moderator) selects one of the places.
3. The user clicks on "Verify this" link.
5. The system prompts the content moderator to confirm the operation.
6. The moderator confirms.
7. The system makes the place verified.
8. The system shows a confirmation message back to the moderator indicating the operation was successful.
9. The asks the moderator to continue verifying.
10. The moderator agrees to continue.
11. Repeat Step 1

**Alternative Paths**

**Alternate Course A:** Content moderator decides to cancel the operation.

A.7. The moderator cancels the operation.

A.8. Back to step 9 of the main success scenario

**Alternate Course B:** The Content moderator want has finished operating and doesn't want to continue.

B.5. The system navigates the user to the "Home Page"

B.6. End of UC-13

#### UC-14: Report user to System Administrator

**Use case name:** UC-14: Report user to System Administrator

**Primary Actors:** Content Moderator

**Summary:** Content moderators report users that don't follow the system's policies to System Administrators to ban/warn from the system.

**Preconditions:** The user must be authenticated.

**Post-conditions:** A User gets a banned from the system.

**Trigger:** The content moderator clicks on "Report user" link in "Reports" page of the system.

##### Main Success Scenario

1. The system prompts the content moderator to confirm the operation.
2. The moderator confirms.
3. The system puts the victim in a black list (effectively banning it from accessing the system).
4. The system shows a confirmation message back to the moderator to indicate the success of the operation.
5. UC-14 ends.

##### Alternative Paths

**Alternate Course A:** Content moderator doesn't confirm the operation.

A.2. The moderator cancels the operation.

A.3 The system notifies the user that the operation is canceled successfully.

A.4. UC-14 ends.

#### UC-15: Register a Moderator

**Use case name:** UC-15: Register a Moderator

**Primary Actors:** System Administrator

**Summary:** Creates a new content moderator for the system.

**Preconditions:** The user must be authenticated.

**Post-conditions:** A new content moderator gets added to the system.

**Trigger:** The user lands on "Register a new moderator" page of the system (through navigation or direct URL).

**Main Success Scenario**

1. System displays a form for the system administrator to fill in.
2. The System Administrator fills in the form & submits.
3. The system prompts for confirmation.
4. The Administrator confirms.
5. The system registers the new moderator.
6. The system shows a success message back to the Administrator of the system.
7. UC-15 ends.

**Alternative Paths**

**Alternate Course A:** Actor doesn't confirm the operation.

- A.2. The Actor cancels the operation.
- A.3 The system notifies the user that the operation is canceled successfully.
- A.4 The Administrator gets navigated to "Home page".
- A.5. UC-15 ends.

UC-16: Register an Admin

**Use case name:** UC-16: Register an Admin

**Primary Actors:** System Administrator

**Summary:** A new System Administrator gets assigned for the system.

**Preconditions:** The user must be authenticated.

**Post-conditions:** A new System Administrator is added to the system.

**Trigger:** The user lands on "Register a new Admin" page of the system (through navigation or direct URL).

**Main Success Scenario**

1. System displays a form for the system administrator to fill in.
2. The System Administrator fills in the form & submits.
3. The system prompts for confirmation.
4. The Administrator confirms.

5. The system registers the new moderator.
6. The system shows a confirmation message back to the moderator to indicate the success of the operation.
7. UC-16 ends.

### Alternative Paths

**Alternate Course A:** The System Administrator doesn't confirm.

- A.2. The Administrator cancels the operation.
- A.3 The system notifies the user that the operation is canceled successfully.
- A.4 The Administrator gets navigated to "Home page".
- A.5. UC-16 ends.

## UC-17: Change a password

**Use case name:** UC-17: Change a password

**Primary Actors:** Ordinary User, Content Moderator, Place Owner, System Administrator

**Summary:** The user changes his/her password.

**Preconditions:** The user must be authenticated.

**Post-conditions:** The user's password gets updated.

**Trigger:** The user lands on "Change a password" page of the system (through navigation or direct URL).

### Main Success Scenario

1. System displays a form with old, new, and confirmation input.
2. The Primary Actor fills in the form & click "submit".
3. The system prompts for confirmation.
4. User confirms.
5. The system updates the user's password.
6. The system shows a confirmation message to indicate the success of the operation.
7. UC-17 ends.

### Alternative Paths

**Alternate Course A:** Primary Actor doesn't confirm the operation.

- A.4. The User (Actor) cancels the operation.
- A.5 The system notifies the user that the operation is canceled successfully.
- A.6. Back to Step 1 of UC-17.

**Alternate Course B:** The two passwords provided by the User don't match.

- A.3. The system informs the user about the problem through error messages.
- A.4. Back to Step 1 of UC-17.

#### UC-18: Delete Invalid Data

**Use case name:** UC-18: Delete Invalid Data

**Primary Actors:** Content Moderator, System Administrator

**Summary:** The Primary Actor deletes the invalid data.

**Preconditions:** The user must be authenticated.

**Post-conditions:** The selected data gets deleted.

**Trigger:** The user clicks on "Delete Invalid Data" link on "Reports" page.

#### **Main Success Scenario**

1. The system prompts for confirmation.
2. User confirms.
3. The system gives an option to "Report the User".
4. UC-18 ends.

#### **Alternative Paths**

**Alternate Course A:** Primary Actor doesn't confirm the operation.

- A.2. The User (Actor) cancels the operation.
- A.3 The system notifies the user that the operation is canceled successfully.
- A.4. The user gets navigated to "Reports" page.
- A.5. UC-18 Ends.

**Alternate Course B:** The User want to report the user.

B.4. The Primary Actor click on "Report the User".

B.5. Triggers UC-18.

#### UC-19: Ban/Warn a User

**Use case name:** UC-19: Ban/Warn a User

**Primary Actors:** System Administrator

**Summary:** The Administrator bans/warns a user from the system.

**Preconditions:** The user must be authenticated.

**Post-conditions:** The selected user gets either banned or warned.

**Trigger:** The user clicks on "Ban/Warn/Ignore" link on one of the reports from "Reports" page.

#### **Main Success Scenario**

1. The system gives an option among banning the user, warning the user and ignoring.
2. The User chooses to Ban.
3. The system bans the victim (reported user).
4. The system displays a success message.
5. UC-18 ends.

#### **Alternative Paths**

**Alternate Course A:** The User chooses to warn the user.

A.4. The User chooses to Warn.

A.5. The System warns the user (victim reported)

A.6. Back to step 4 of UC-18.

**Alternate Course B:** The User chooses to ignore the report.

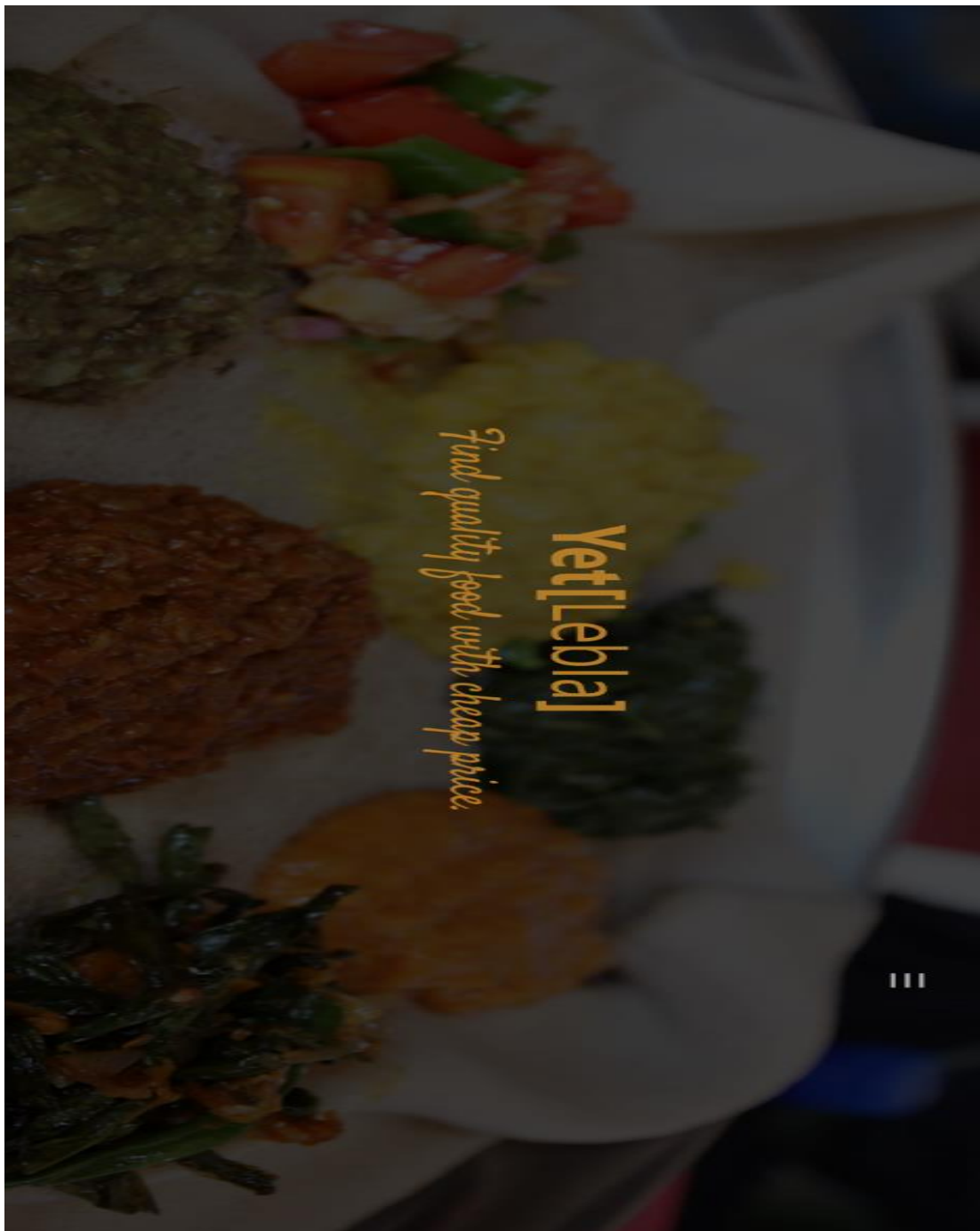
B.4. The User chooses to Ignore.

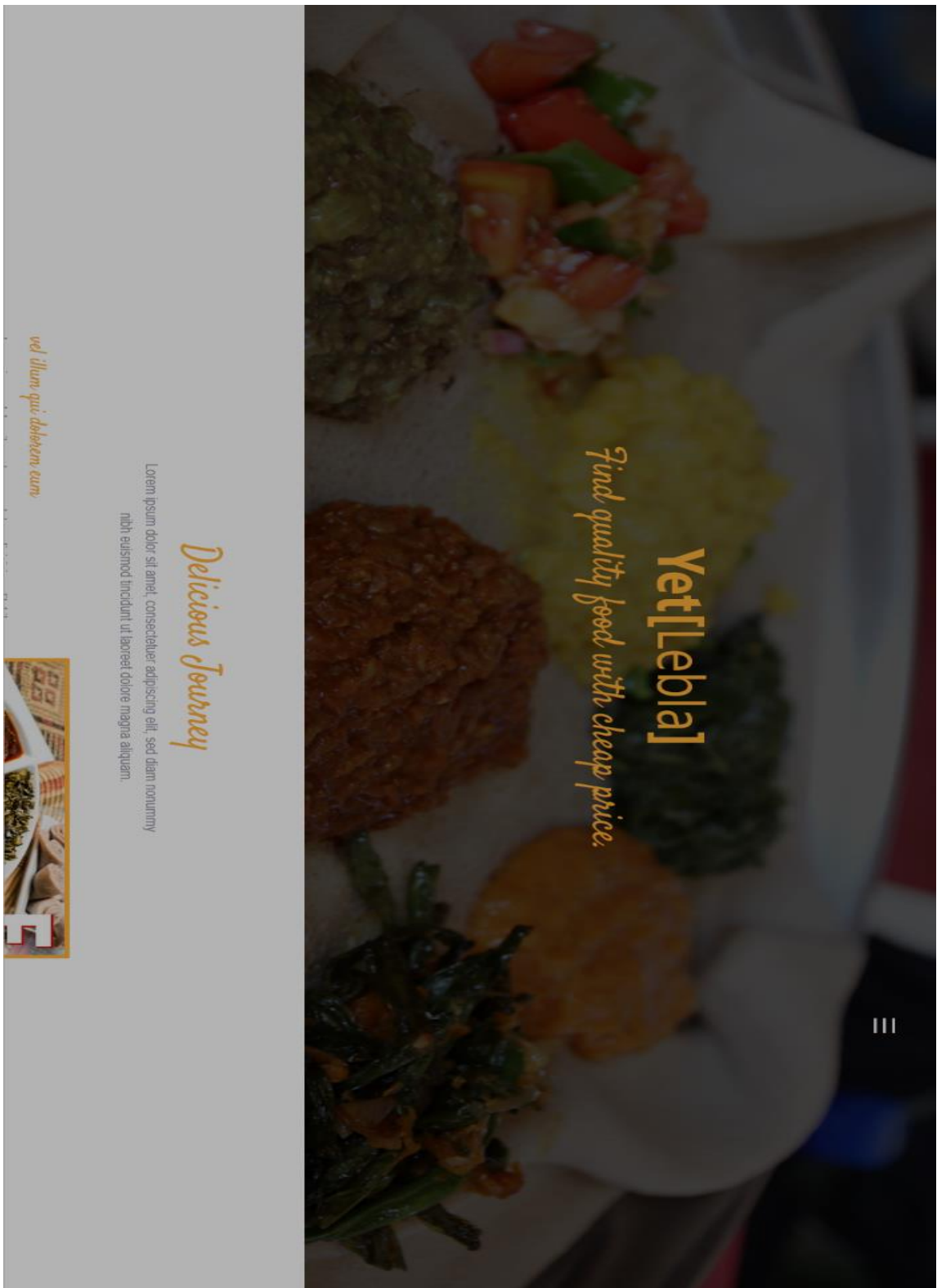
B.5. The system notifies the user that the report got ignored.

B.6. End of UC-18.



## UI Samples





Menu List

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam.

Show All
 Breakfast
 Lunch
 Dinner

FOOD ITEM NAME
 \$20.99
 FOOD ITEM NAME
 \$20.99
 *Ilque passu quidquam est qui debetur*

FOOD ITEM NAME
 \$20.99
 FOOD ITEM NAME
 \$20.99
 *Ilque passu quidquam est qui debetur*

FOOD ITEM NAME
 \$20.99
 FOOD ITEM NAME
 \$20.99
 *Ilque passu quidquam est qui debetur*

FOOD ITEM NAME
 \$20.99
 FOOD ITEM NAME
 \$20.99
 *Ilque passu quidquam est qui debetur*

FOOD ITEM NAME
 \$20.99
 FOOD ITEM NAME
 \$20.99
 *Ilque passu quidquam est qui debetur*

FOOD ITEM NAME
 \$20.99
 FOOD ITEM NAME
 \$20.99
 *Ilque passu quidquam est qui debetur*

FOOD ITEM NAME
 \$20.99
 FOOD ITEM NAME
 \$20.99
 *Ilque passu quidquam est qui debetur*

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy  
 nibh euismod tincidunt ut laoreet dolore magna aliquam.

Loem ipsam docti sit amet, consequetur adipiscing elit. Libero  
impedit. Invenire culpa vero accusamus in nostrum  
dignissimos modi, molestiae. Autem iusto esse necessitatibus  
ex corporis aenum quaeat volutates quibusdam dictal

-

Yet[Lebla]

Menu List

>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam.

Show All

Breakfast

Lunch

Dinner

FOOD ITEM NAME  
Illegible placeholder text

\$20.99

FOOD ITEM NAME  
Illegible placeholder text

\$20.99

FOOD ITEM NAME  
Illegible placeholder text

\$20.99

FOOD ITEM NAME  
Illegible placeholder text

\$20.99

FOOD ITEM NAME  
Illegible placeholder text

\$20.99

FOOD ITEM NAME  
Illegible placeholder text

\$20.99

FOOD ITEM NAME  
Illegible placeholder text

\$20.99

FOOD ITEM NAME  
Illegible placeholder text

\$20.99

FOOD ITEM NAME  
Illegible placeholder text

\$20.99

FOOD ITEM NAME  
Illegible placeholder text

\$20.99

FOOD ITEM NAME  
Illegible placeholder text

\$20.99

FOOD ITEM NAME  
Illegible placeholder text

\$20.99

Center Of Information Technology and Scientific Computing  
Page

31 |

Yet[Lebla]

Phone Numbers

+4438812456

Opening Hours of our Office

Monday to Friday 09:00 - 24:00

Friday and Sunday 08:00 - 03:00

Sign Up

Your Name

mm/dd/yyyy

Your Email

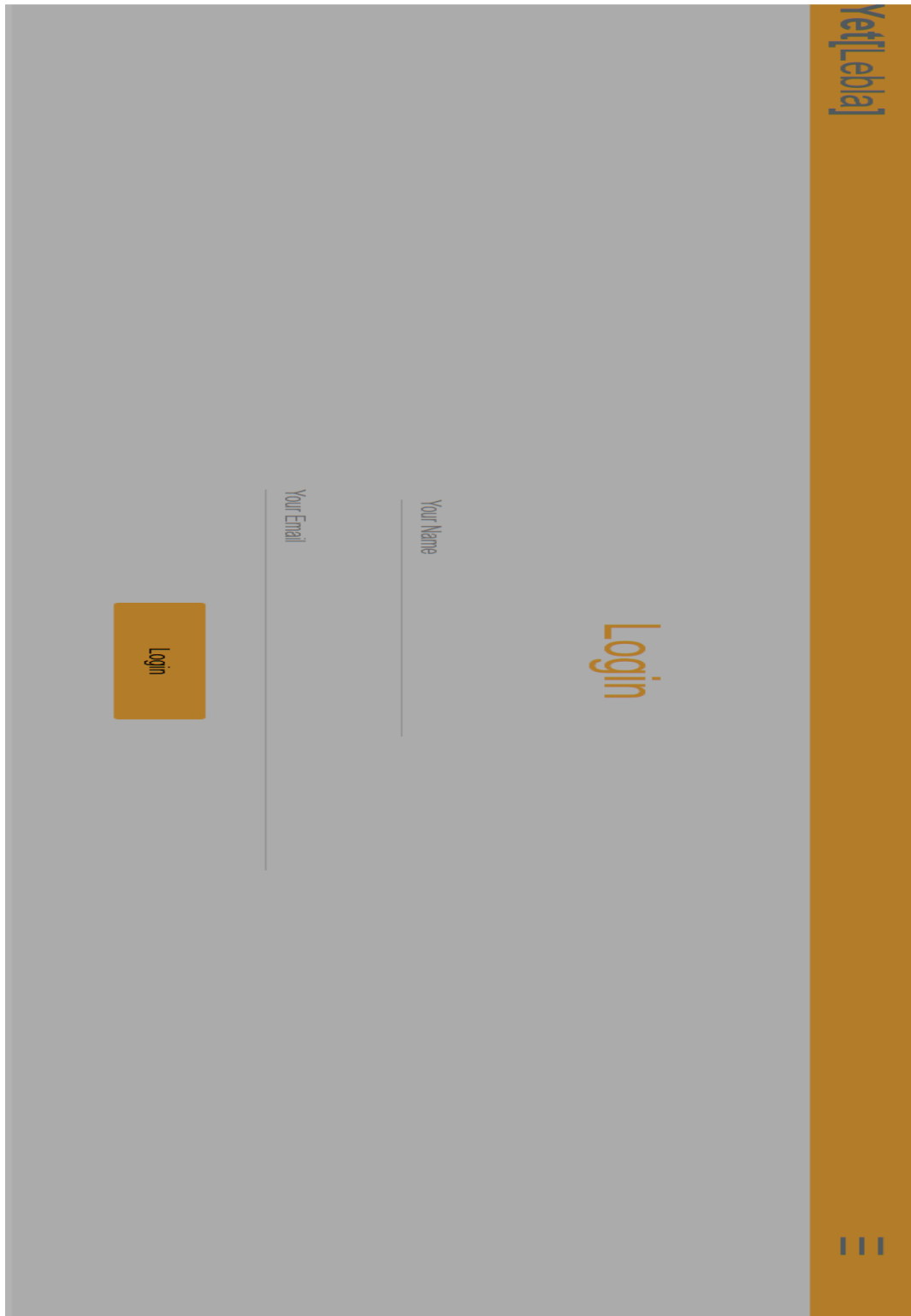
-- --

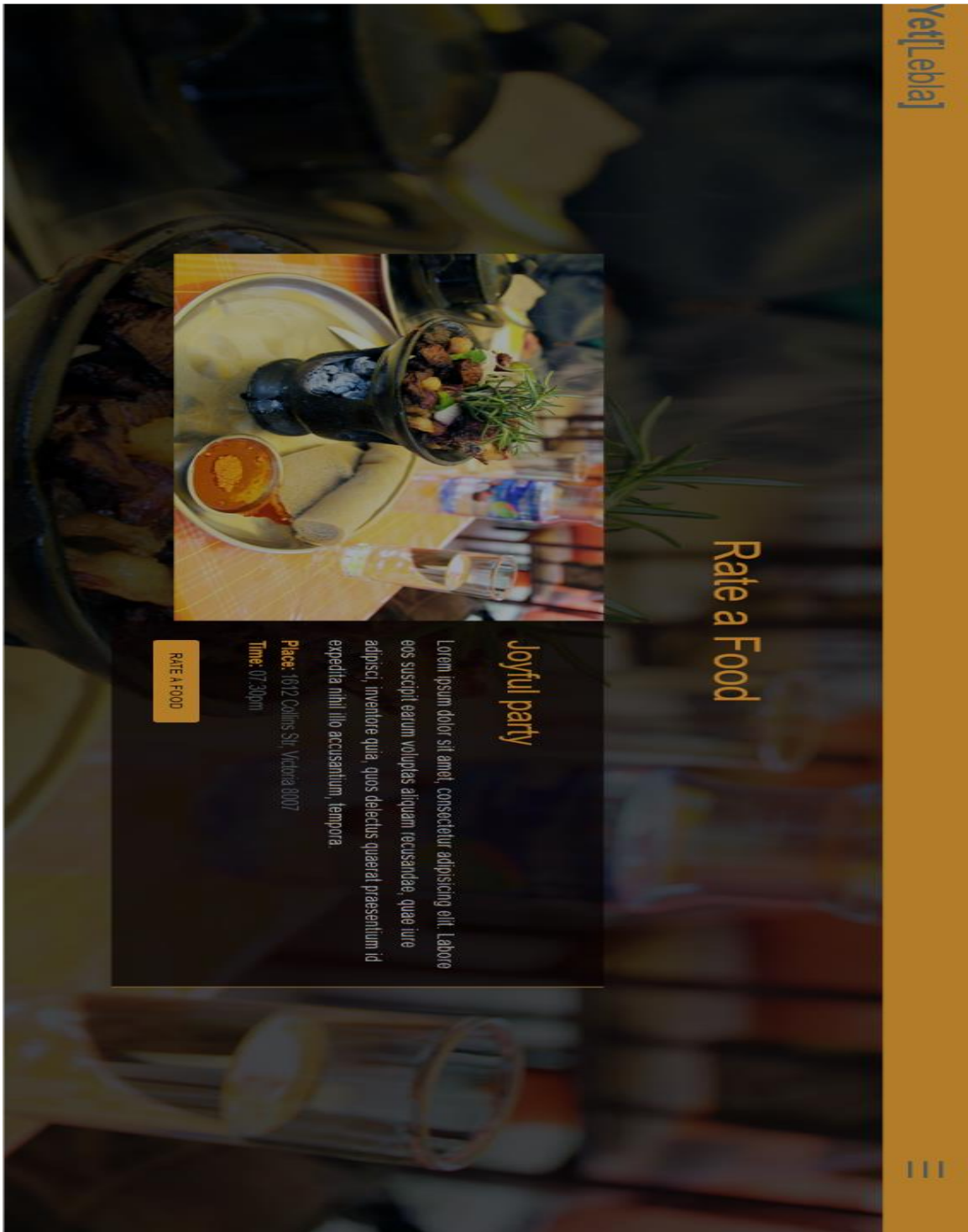
Phone

People

Message

Sign Up







## **5. Other Requirements**

### **Training-related Requirements**

No specific training is required to use this app.

## **6. Change Management Process**

During the course of implementing this project the SRS document can be edited or modified at any time by the consent of the development team. Accordingly, stakeholders of this project will be notified immediately via social media.

Any request to change or alter this SRS Document by a team member will be handled by voting on it by all team members and dealt upon accordingly.

**Websites**

<http://www.wikipedia.org/>

<http://www.appcrawler.com/web-apps/>

**Bibliography**

Ian Sommerville Software Engineering