



ADDIS ABABA UNIVERSITY

ADDIS ABABA INSTITUTE OF TECHNOLOGY

Center of Information Technology and Scientific Computing

Department of Software Engineering

Software engineering II

Software Requirement Specification

Prepared By:

- | | |
|--------------------|-------------|
| 1. Aman Bereket | ATR/9348/08 |
| 2. Biya Girma | ATR/7547/08 |
| 3. Estifanos sisay | NSR/9401/08 |
| 4. Hena Fufa | ATR/3750/08 |
| 5. Hermella Frew | ATR/1689/08 |
| 6. Mihret Tamene | ATR/3534/08 |
| 7. Oromia Godanna | ATR/6053/08 |
| 8. Yohannes Fassil | ATR/4122/08 |

Submitted to: Mr. Natnael Argaw

Date: March 23/2018

Addis Ababa

Ethiopia

Software Requirement Specifications

(SRS)

Revision History

Date	Description	Author	Comments
March 23	Version 1	Team members	Draft
April 8	Version 2	Team members	First revision
April 23	Version 3	Team members	Final revision

Document Approval

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

Signature	Printed Name	Title	Date
	Natnael Argaw	Instructor, ITSE	

Table of Contents

Revision History.....	i
Document Approval.....	i
List of Tables.....	ii
List of Figures.....	ii
Definitions, Acronyms, and Abbreviations.....	iii
Declaration.....	iv
1. Introduction.....	1
1.1 Purpose.....	1
1.2 Scope.....	1
1.3 Overview.....	2
2. General Description.....	3
2.1 Product perspective.....	3
2.2 Product functions.....	3
2.3 users characteristics.....	3
2.4 general constraints.....	4
2.5 Assumptions and dependencies.....	4
3. Specific Requirements.....	5
3.1 External Interface Requirements.....	5
3.1.1 User Interfaces.....	5
3.1.2 User Interface Prototype.....	8
3.1.3 Hardware Interfaces.....	12
3.1.4 Software Interfaces.....	12
3.1.5 Communication Interfaces.....	12
3.2 Functional Requirements.....	13
3.2.1 FR-01 Registration.....	13
3.2.2 FR-02 Login.....	13
3.2.3 FR-03 Customize Profile.....	14
3.2.4 FR-04 Choose mood.....	15
3.2.5 FR-05 Suggestion.....	15

3.2.6	FR-06 Rate.....	16
3.2.7	FR-07 Set Profile Picture.....	16
3.2.8	FR-08 Reset password.....	17
3.2.9	FR-09 Add/Remove photo.....	17
3.2.10	FR-10 Change Password.....	18
3.2.11	FR-11 View Place profile.....	19
3.2.12	FR-12 View on map.....	19
3.2.13	FR-13 Logout.....	20
3.3	Uses Cases.....	21
3.3.1	UC-01 Customize profile.....	22
3.3.2	UC-02 Add Profile Picture.....	22
3.3.3	UC-03 Specify mood.....	23
3.3.4	UC-04 request suggestion.....	24
3.3.5	UC-05 Rate.....	25
3.3.6	UC-06 Remove/add photo.....	25
3.3.7	UC-07 Sign up.....	26
3.3.8	UC-08 Sign in.....	27
3.3.9	UC-09 Change Password.....	27
3.3.10	UC-10 Reset Password.....	28
3.3.11	UC-11 Logout	29
3.4	NON-FUNCTIONAL requirements.....	29
3.4.1	Performance.....	29
3.4.2	Reliability.....	30
3.4.3	Availability.....	30
3.4.4	Security.....	30
3.4.5	Maintainability.....	30
3.4.6	Portability.....	30
3.5	Inverse Requirements.....	31
3.6	Design constraint.....	31
3.7	Other Requirements.....	31
3.7.1	Training-related Requirements.....	31
4.	Change Management Process.....	32
	References.....	33

List of Tables

None

List of Figures

Figure 3.1.1.1 Logical UI flow diagram

Figure 3.1.1.2.1 Login

Figure 3.1.1.2.2 Sign up (Individual)

Figure 3.1.1.2.3 Sign up (Place)

Figure 3.1.1.2.4 Home Screen (Individual)

Figure 3.1.1.2.5 Home Screen (Place)

Figure 3.1.1.2.6 Suggestion Screen

Figure 3.1.1.2.7 View Map

Figure 3.1.1.2.8 Customize Profile

Figure 3.1.1.2.9 View Profile

Figure 3.3 Use case diagrams

Definitions, Acronyms, and Abbreviations

Mood: A state or quality of feeling at a particular time.

Place vibe: A distinctive emotional atmosphere offered by the place.

Individual: A type of user which wants to visit places providing his current mood.

Place: Another type of user which wants to advertise his/her place by specifying a certain place vibe.

SRS: A description of a software system to be developed.

Google Maps: An application which is developed by Google and offers a map services.

IOS: An operating system developed by Apple Inc.

Android: Open source operating system made for mobile phones.

GUI: Graphical user interface.

TCP/IP: Transmission Control Protocol/ Internet Protocol.

ER: Entity relationship diagram, used for showing logical characteristics of a database.

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

1. Introduction

1.1 PURPOSE

The purpose of this SRS is to provide a general overview and also a very detailed examination of what the system (Vibe) should offer its users and the constraints under which it operates.

The document first begins by giving a general overview of the application which should be suitable for anyone without technical knowledge of the system to read and understand it. The document then continues examining the requirements in detail giving those with some technical knowledge to be able to read it and implement or test or maintain it.

This SRS documentation is intended to be proposed to any potential user for their approval and to give a brief description of the application as a reference for developing the first version of the system for the development team, to identify what it must do and the basic requirements it must fulfill.

1.2 SCOPE

By the end of development a web based application, Vibe, will be produced with the following characteristics:

I. The application will allow a certain registered place to advertise itself in its own words which it thinks best describes the place's 'Vibe' or feeling.

II. The application will allow any registered user to be able to find places based on their input of current mood or feeling.

III. The application will allow any registered place or user to upload picture including profile pictures.

IV. The application (if available) will display the places location in a map.

V. The application will allow any registered user to rate the places they visited.

VI. The application will enable the user to search for any registered places to view their profile, the service they provide and their location.

VII. In the contrast the application will not allow any type of communication between any two users, i.e. direct communication among individual users and between individuals and the registered places.

Benefits of the system:

- Provides a means for places to advertise themselves as products, by giving a list of services they provide and associating the emotional response they expect from their customers.
- Enables users to find new places based on their current mood.

The primary goal of the system is to create an easy way for users to navigate what's around them based on their current mood and places to attract customers based on what they offer and think represents the places vibe.

1.3 OVERVIEW

With the above simple understanding of the application it is possible to continue to define more of the requirements of the system. In the next section, General Description, the requirements of the application from a more general and easy perspective, it tries to make the specific requirements in section 3 easier to understand and familiarizes readers with the system informally.

To achieve the above goal section 2 contains subsections such as **product perspective**, to familiarize the reader with the system by comparing this application with similar existing systems, **product functions**, to provide an overall summary of the functions this system provides, **user characteristics**, this subsection highlights the characteristics of expected end users of the application, and finally the **general constraints** subsection provides insights to the constraints put on the developer whilst making the application.

Section 3, Specific Requirements, then explores those requirements outlined in section 2 briefly.

2. General description

2.1 PRODUCT PERSPECTIVE

The application provided below tends to have few similarities:

I. **Google maps**: This system provides navigation of countries, cities, searching of place and more, which makes it similar with this system even though Google maps offers a very large scope of exploration.

II. **YELP**-Yelp mobile app, which publish crowd-sourced reviews about local businesses, as well as the online reservation service Yelp Reservations. The company also trains small businesses in how to respond to reviews, hosts social events for reviewers, and provides data about businesses, including health inspection scores.

2.2 PRODUCT FUNCTIONS

- Suggests a place to visit in based on the feeling they provide
- The application will let a registered user rate a place.
- The application will allow places to promote their services to appeal variety of customers through pictures and list of moods they elevate to meet customers state of conscious.

2.3 USERS CHARACTERISTICS

The users of the application can roughly be divided into two: the **Individual user** and **place user**.

- Both end users need to register in order to get or provide services.
- The individual user has general characteristics that resemble most social media users; where registration is required to find places he/she might enjoy.
- The place user is generally assumed to have organizational characteristics. The page being their canvas where they can paint a picture of their place through marketing and advertisement skills.

Structure of users resembles client-server design where individual user requests for a suggestion (as a client) and the place user is suggestion based on signed up mood.

2.4 GENERAL CONSTRAINTS

- Since the software interacts with users' personal it's critical it doesn't leak or expose that sensitive information to unauthorized person. Thus developers should be aware of security threat.
- To improve User experience, developers should be cautious about performance and response time of the application.
- Connection with maps to provide exact location for user.

2.5 ASSUMPTIONS AND DEPENDENCIES

- Assumes the user can express their current feeling through an emoji or an adjective.
- Assumes user has had exposure on how to use the internet or different social Medias.
- Assumes signed up place can be accessed through maps
- Since the system shall retrieve the current location of the user, it assumes the user has consistent internet connection.
- It is generally assumed that the Users computer is able to give information about its current location.

3. Specific Requirements

In the above section, we have tried to give the general overview of our entire system including what requirements it has. Now we will give the specific and detailed requirements which will be the guidelines to the design, implementation and testing phase. These requirements are specifically written to aid the developer in developing the system and the testers to test the system.

Here is the list of things covered in this section:

1. Requirements concerning External interfaces (i.e. UI, hardware and software Interfaces).
2. The functional and non-functional requirements of the application will be briefly dealt with.
3. Use cases will be illustrated.
4. We will briefly discuss our design constraints.
5. The requirements for the database needed will be discussed.

3.1 EXTERNAL INTERFACE REQUIREMENTS

This section provides a detailed description of all inputs into and outputs from the system. It also gives a description of the hardware, software and communication interfaces and provides basic prototypes of the user interface.

3.1.1 User Interfaces

Link to the signup/register page should be provided for a first time users of the application. The application shall offer a sign up screen to its users. The signup page screens shall provide the necessary tools to successfully register a *place* and *individuals* respectively on the system. After successfully registered a user should be directed to the login page. The registration step (Screen) shall provide a way to accept the following inputs:

From *Individuals*:

- i. Username , Password and email

From *Places*:

- i. Username and Password
- ii. Current location
- iii. Contact information (Phone number and email)
- iv. Associated Place vibe

If the user is not a first time user, he/she should be able to login. The login screen shall offer its users to insert their credentials (Username and Password) to login. After successful login, the system shall present the user with the appropriate home screen based on the type of user he/she is.

There shall be a home screen for both users after successful login. The home page is responsible for displaying a picture gallery, the total rank (given by people) and the profile and contact information for the place. The home page for the individual user should include their profile information, a list of previously suggested places, and an input to insert the emotion they are experiencing to request place suggestion.

The users shall be able to customize their profile by:

- i. Providing their preferred profile picture.
- ii. Change their current password.
- iii. *Individuals*, by providing their current mood.
- iv. *Places*, by providing their picture gallery content.
- v. *Places*, by removing pictures from the gallery.
- vi. *Places*, by providing their current place vibe.
- vii. *Places*, by modifying their current location

The GUI shall be able to display the suggested places in a map. There shall be an interface for the user to rate and the system **may** allow users to give comments about the *place* he was suggested after visiting it. An *Individual* shall be able to see the profile of the suggested places, view their picture gallery, view the user ratings and **might** see the comments associated with the place. The system **may** allow users to retrieve forgotten passwords through a forgotten password feature. Users who previously have experience using Google maps and any of the social media apps like what's app, Messenger, or Telegram (on android or IOS) will not find it hard to use.

Each registered place should have a profile page which should be available for individual's users when the place got suggested. It should include the whole profile of the place with the name, location, services and location.

There **may** be a screen for when the user forgot their passwords. It should enable the users to re-set a new password. After the password is successfully changed the user should be able to login with the new password.

3.1.1.1 Logical UI flow Diagram

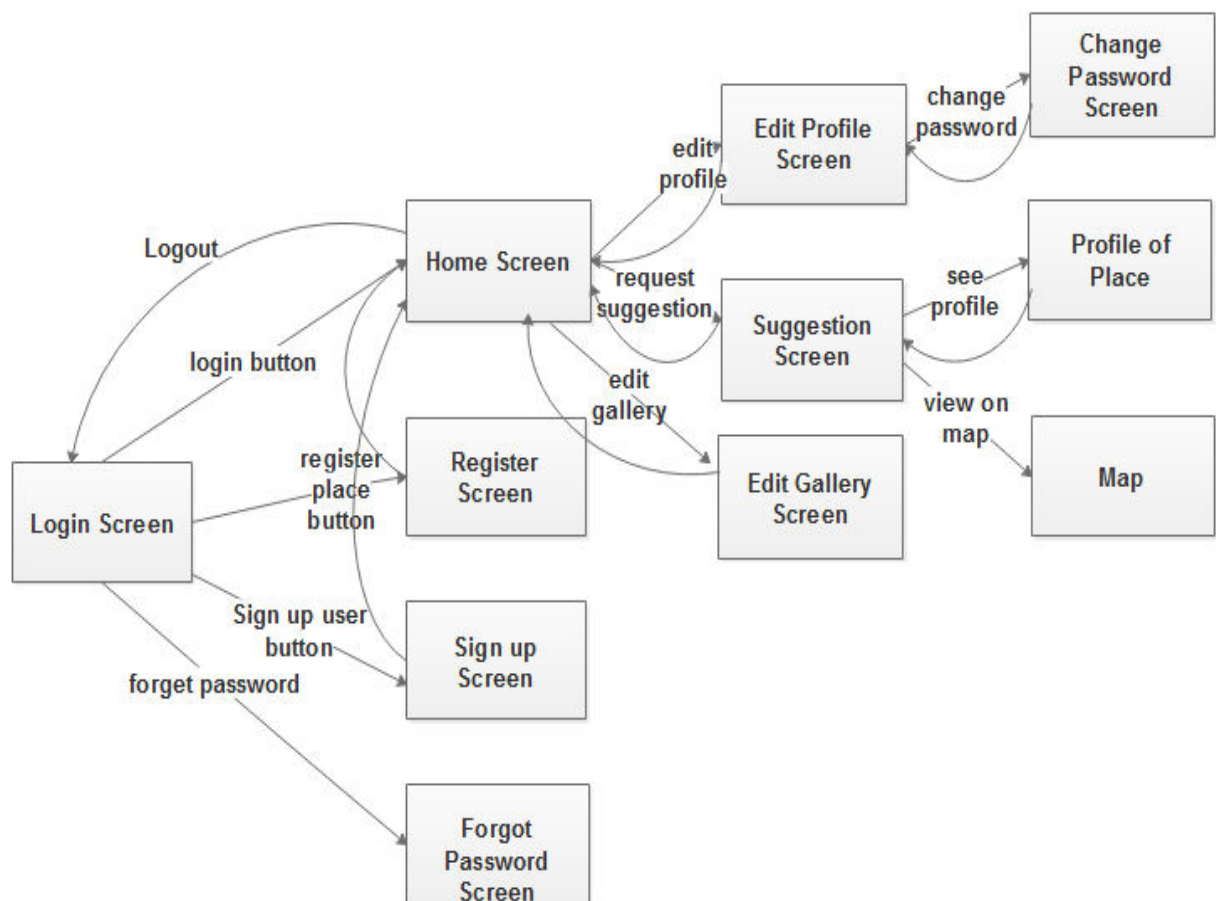
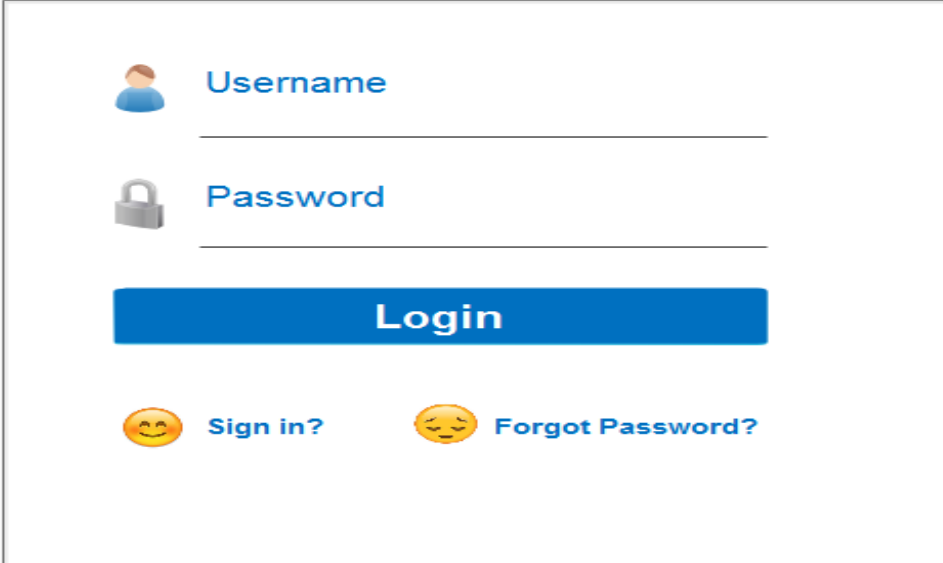


Figure 3.1.1.1 logical UI flow diagram

3.1.1.2 User Interface Prototype

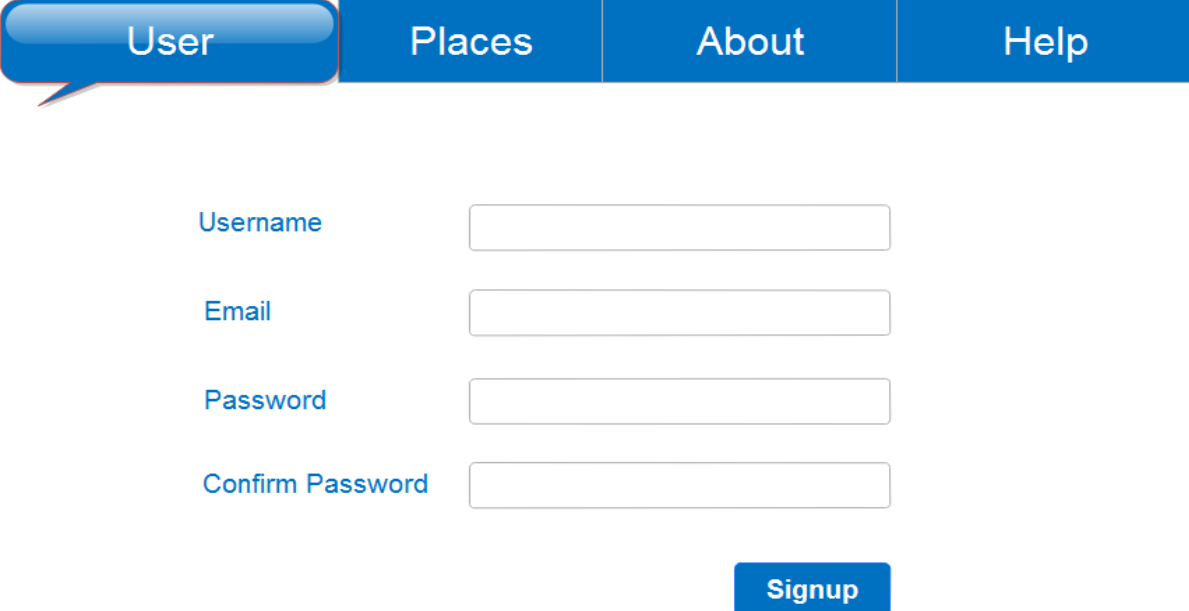
3.1.1.2.1 Login screen for users.



A login screen prototype with a white background and a thin grey border. It features a user icon (person) to the left of a text input field labeled "Username". Below this is a lock icon to the left of a text input field labeled "Password". A large blue button with the text "Login" is centered below the input fields. At the bottom, there are two links: "Sign in?" with a smiling face emoji and "Forgot Password?" with a sad face emoji.

Figure 3.1.1.2.1 Login screen

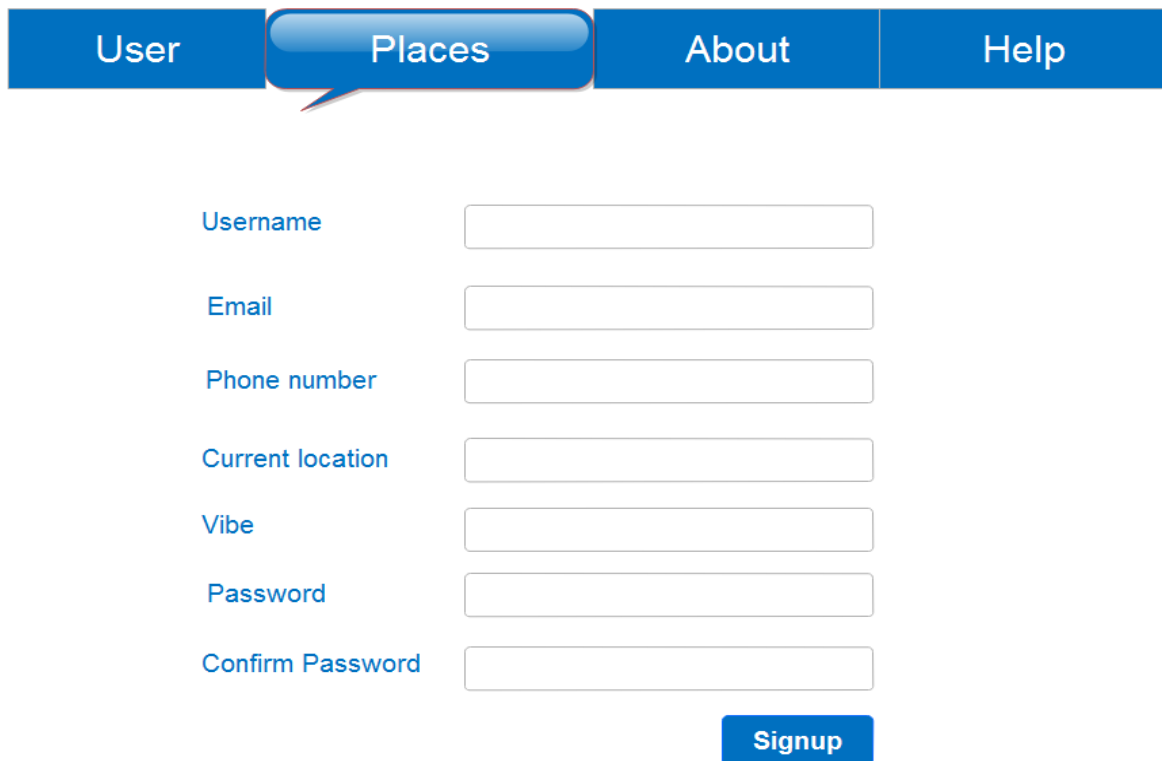
3.1.1.2.2 Signup screen for Individuals



A signup screen prototype. At the top is a horizontal navigation bar with four blue buttons: "User" (which has a speech bubble tail pointing down), "Places", "About", and "Help". Below the navigation bar are four text input fields, each preceded by a label: "Username", "Email", "Password", and "Confirm Password". A blue "Signup" button is positioned at the bottom right of the form.

Figure 3.1.1.2.2 signup (Individual) screen

3.1.1.2.3 Signup screen for Places.

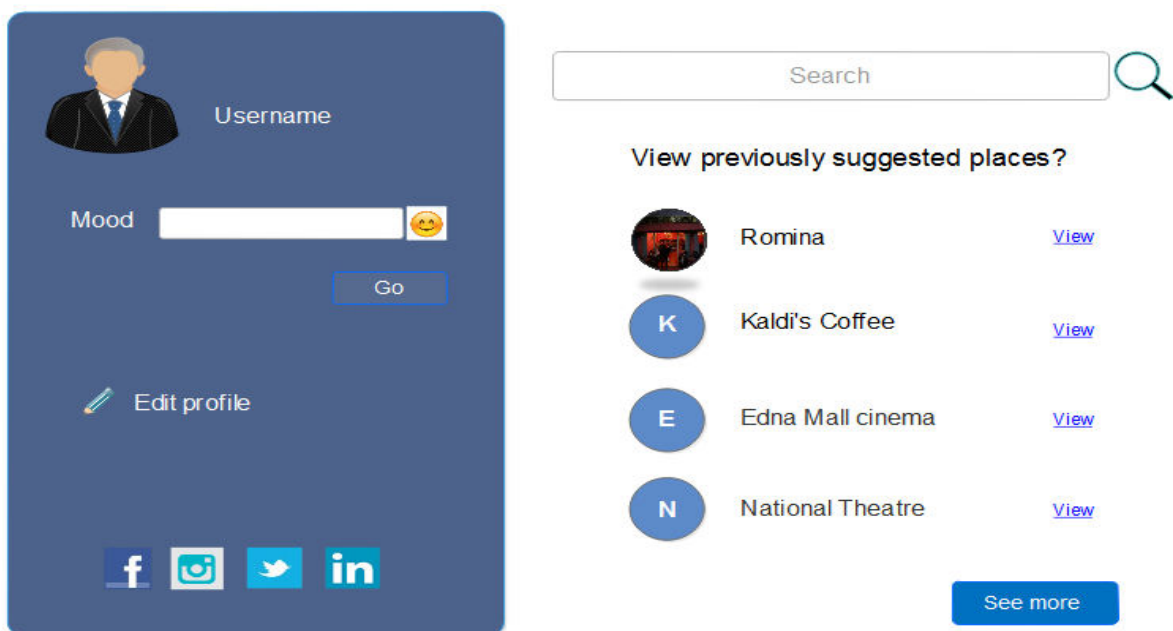


The image shows a navigation bar with four items: 'User', 'Places', 'About', and 'Help'. The 'Places' item is highlighted with a blue background and a white speech bubble. Below the navigation bar is a form with seven input fields: 'Username', 'Email', 'Phone number', 'Current location', 'Vibe', 'Password', and 'Confirm Password'. Each field is a simple white rectangle with a thin border. To the right of each label is its corresponding input field. At the bottom right of the form is a blue button with the text 'Signup' in white.








User	Places	About	Help
Username	<input type="text"/>		
Email	<input type="text"/>		
Phone number	<input type="text"/>		
Current location	<input type="text"/>		
Vibe	<input type="text"/>		
Password	<input type="text"/>		
Confirm Password	<input type="text"/>		
			Signup

Figure 3.1.1.2.3 signup (Place) screen

3.1.1.2.4 Home screen for Individuals.



The image shows a home screen for individuals. On the left is a dark blue profile card. It features a placeholder for a profile picture, the text 'Username', a 'Mood' section with a text input field and a smiley face icon, a 'Go' button, an 'Edit profile' link with a pencil icon, and social media icons for Facebook, Instagram, Twitter, and LinkedIn. On the right is a search bar with the text 'Search' and a magnifying glass icon. Below the search bar is the text 'View previously suggested places?'. This is followed by a list of four items, each with a circular icon, a name, and a 'View' link. The items are: Romina (with a red and black icon), Kaldi's Coffee (with a blue circle containing 'K'), Edna Mall cinema (with a blue circle containing 'E'), and National Theatre (with a blue circle containing 'N'). At the bottom right is a blue button with the text 'See more'.

	Username	
Mood	<input type="text"/>	
	Go	
	Edit profile	
		
		






<input type="text" value="Search"/>	
View previously suggested places?	
	Romina View
	Kaldi's Coffee View
	Edna Mall cinema View
	National Theatre View
	See more

Figure 3.1.1.2.4 Home screen (Individual)

3.1.1.2.5 Home screen for Places.

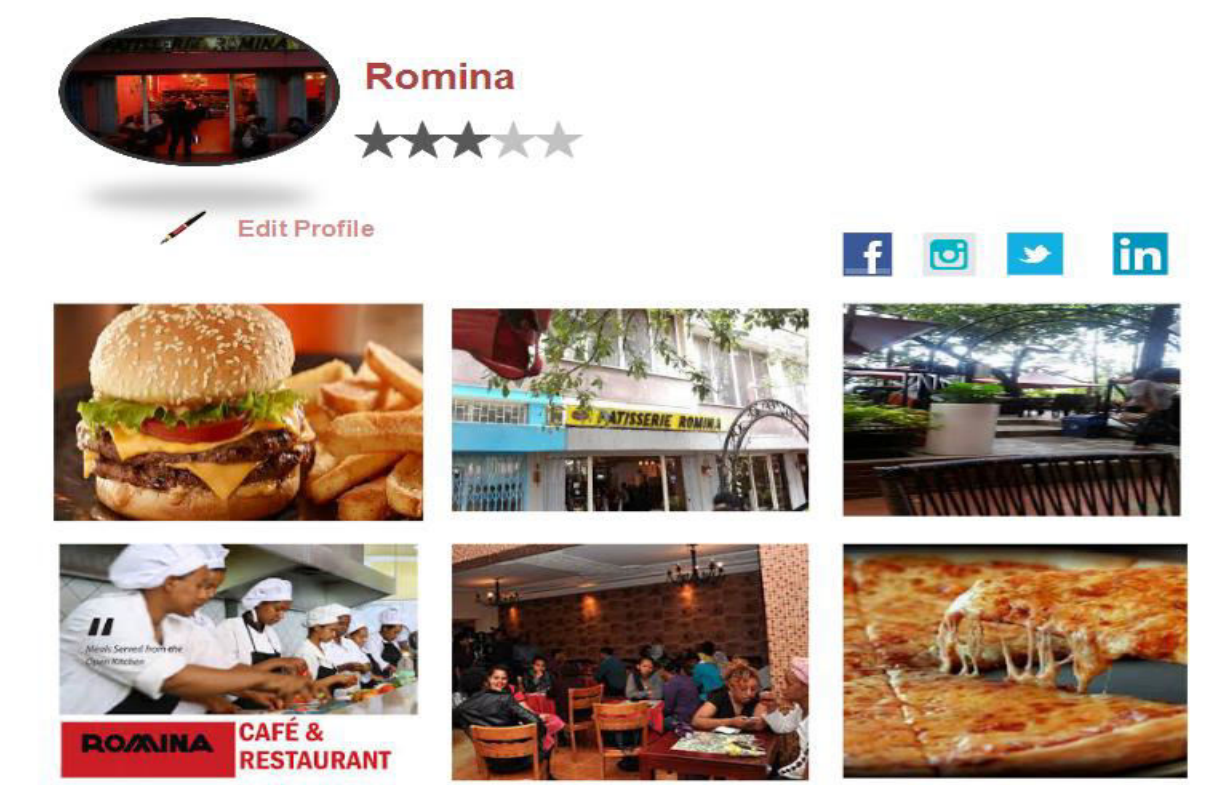
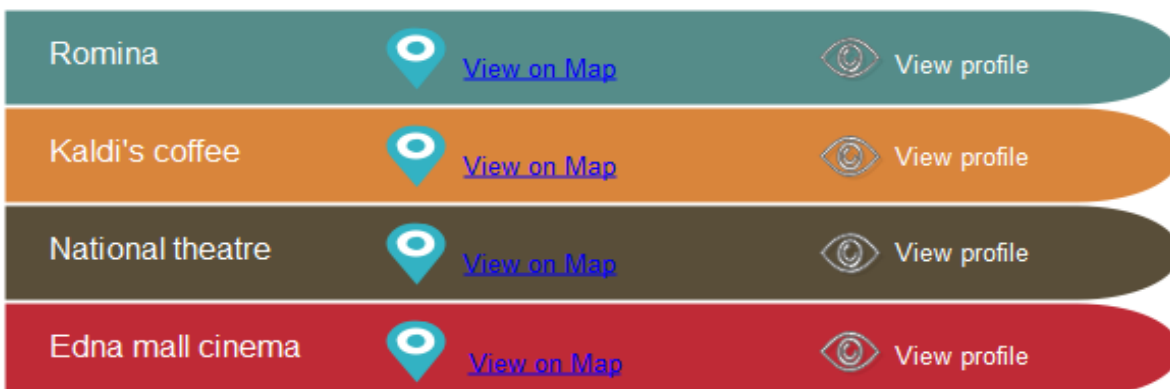


Figure 3.1.1.2.5 Home screen (Places)

3.1.1.2.6. Suggested Place

Suggested Places



[See more >>](#)

Figure 3.1.1.2.6 Suggested Places

3.1.1.2.7 View on map

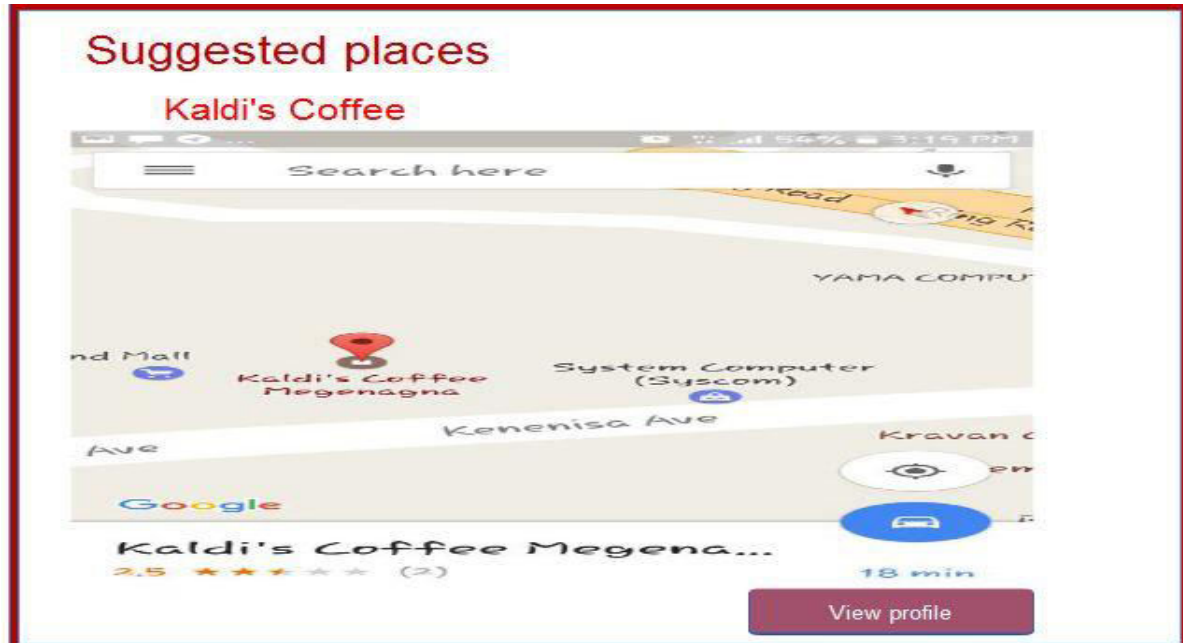




Figure 3.1.1.2.7 view on map

3.1.1.2.8 Customize Profile

Edit Profile



 [Browse](#)




Figure 3.1.1.2.7 Customize Profile

3.1.1.2.9 View Profile

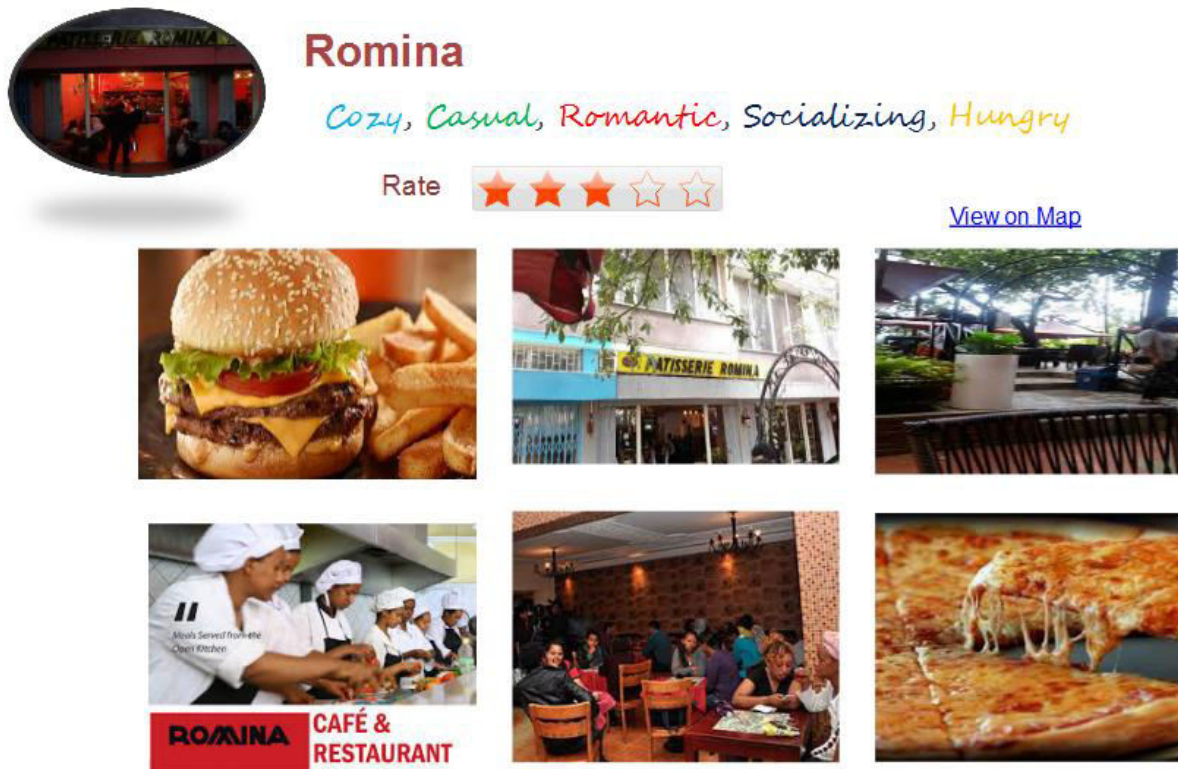


Figure 3.1.1.2.8 view Profile

3.1.2 Hardware Interfaces

The system has no specific hardware interface.

3.1.3 Software Interfaces

This system has no specific software interface.

3.1.4 Communication Interfaces

The system uses the Internet as a means of communication between the client app and the server, thus the system will be relying on the TCP/IP protocol.

3.2 FUNCTIONAL REQUIREMENTS

3.2.1 FR-01 Registration

3.2.1.1 Introduction

This functional requirement states: the application should allow users to register either as a *place* or *individual*, with the appropriate screen.

3.2.1.2 Input

Inputs required are: Username, Password and email address from both types of Users, and Phone number, current location and vibe for *Places*.

3.2.1.3 Processing

The user inputs mentioned above are sent to the server using the Internet where they are written to a database record.

3.2.1.4 Output

Upon successful registration, the user information will be written to database and notify the user or prompt the user to their home screen.

3.2.1.5 Error Handling

The application will notify the user of the problem that occurred. Actions he might take to resolve the problem are also suggested.

3.2.2 FR-02 Login

3.2.2.1 Introduction

The application shall provide Users with a means of authorization before logging in to their account.

3.2.2.2 Input

The application will accept Username and password as a form of input.

3.2.2.3 Processing

The application will send the accepted inputs and sends it over to the server where verification of the user credentials is assured.

3.2.2.4 Output

The application upon successful verification will prompt users to their home screen.

3.2.2.5 Error Handling

The application will notify the user of the problem that occurred. Actions he might take to resolve the problem are also suggested

3.2.3 FR-03 Customize Profile

3.2.3.1 Introduction

The application shall provide users with a profile which they can customize, based on the type of users they are.

3.2.3.2 Input

The application accepts a mood/vibe choice or a request to further modify the profile via add profile picture, change password, or add/remove pictures.

3.2.3.3 Processing

The application will send the accepted mood/vibe input and send it over to the server, or redirects user to the appropriate screen if further modifications is requested.

3.2.3.4 Output

The database will be updated with the new mood/vibe.

The profile screen of the user will be updated accordingly, or will be redirected to the appropriate screen if further modifications are requested.

3.2.3.5 Error Handling

The application will notify the user of the problem that occurred. Actions he might take to resolve the problem are also suggested.

3.2.4 FR-04 Choose mood/vibe

3.2.4.1 Introduction

The application shall offer its users multiple types of moods or *place vibes* to choose from.

3.2.4.2 Inputs

Request to choose moods or *place vibes*.

3.2.4.3 Processing

Determine the type of User it is and prepare a set of moods or *place vibes*.

3.2.4.4 Output

A list of moods or place vibes to choose from.

3.2.4.5 Error Handling

The application will notify the user of the problem that occurred. Actions he might take to resolve the problem are also suggested.

3.2.5 FR-05 Suggestion

3.2.5.1 Introduction

The application shall suggest a place based on *Individual* mood and *Place* vibe.

3.2.5.2 Inputs

Current *Individual* mood and place vibe entered.

3.2.5.3 Processing

After the user requests a place to be suggested, the client app sends current mood of the individual and to the server.

The server then tries to find best fit by comparing this data against several place vibes.

3.2.5.4 Output

Notifies *individual* of the found places and presents them with a link to the profile of the *Places* suggested.

3.2.5.5 Error Handling

The system will notify the user of the problem that occurred. Actions he might take to resolve the problem are also suggested.

3.2.6 FR-06 Rate

3.2.6.1 Introduction

Individuals shall be able to give feedback on a suggested *place* by rating them out five.

3.2.6.2 Input

The input is the amount of score the user gave the place on some defined scale.

3.2.6.3 Processing

After an individual has rated a *place*, the data is sent to a server, where the rating is used to calculate the overall rating of the place.

3.2.6.4 Output

The database and profile of the *place* will be updated following Rating.

3.2.6.5 Error handling

The application will notify the user of the problem that occurred. Actions he might take to resolve the problem are also suggested.

3.2.7 FR-07 Set Profile picture

3.2.7.1 Introduction

The application shall allow the User to specify a profile photo.

3.2.7.2 Inputs

The selected photo.

3.2.7.3 Processing

Sending of the photo to server and saving the photo in a database.

3.2.7.4 Output

Profile photo of the user will be changed.

3.2.7.5 Error Handling

The application will notify the user of the problem that occurred. Actions he might take to resolve the problem are also suggested.

3.2.8 FR-08 Reset password

3.2.8.1 Introduction

The application shall allow the User to reset his/her password.

3.2.8.2 Inputs

User's confirmation he still has access to the email address provided. The other is the new password desired.

3.2.8.3 Processing

Instructions to reset password are sent to the email address

3.2.8.4 Output

Resets the password with the new password provided.

3.2.8.5 Error Handling

The application will notify the personnel of the problem that occurred. Actions he might take to resolve the problem are also suggested.

3.2.9 FR-09 Add/remove photo

3.2.9.1 Introduction

The application shall allow the Place to add or remove a photo to/from its gallery.

3.2.9.2 Inputs

The photo selected to be removed or added.

3.2.9.3 Processing

Send the photo over to server or instruct the server to remove the selected photo based on the user's request.

3.2.9.4 Output

The selected photo is removed or added as instructed.

3.2.9.5 Error Handling

The application will notify the personnel of the problem that occurred. Actions he might take to resolve the problem are also suggested.

3.2.10 FR-10 Change password

3.2.10.1 Introduction

The application shall allow the User to change his current password.

3.2.10.2 Inputs

The current password and the new password.

3.2.10.3 Processing

The current password supplied is compared with the existing password on the server if it matches the current password is replaced with the new password.

3.2.10.4 Output

Change the user's current password and writes changes to the database.

3.2.10.5 Error Handling

The application will notify the personnel of the problem that occurred. Actions he might take to resolve the problem are also suggested.

3.2.11 FR-11 View Place Profile

3.2.11.1 Introduction

The application shall allow the User to view the profile of the place he/she got as a suggestion.

3.2.11.2 Inputs

The user name of the place he/she want to see the profile of.

3.2.11.3 Processing

The user name of the place is sent to server where its profile information is retrieved from the database.

3.2.11.4 Output

Uses the information provided from the server to display the desired place profile.

3.2.11.5 Error Handling

The application will notify the personnel of the problem that occurred. Actions he might take to resolve the problem are also suggested.

3.2.12FR-12 View on map

3.2.12.1 Introduction

The application shall allow the User to view the location of the suggested place on a map.

3.2.12.2 Inputs

The user name of the place we need to retrieve the location of.

3.2.12.3 Processing

The location of the place is retrieved by sending the provided username and used to identify the location on the map

3.2.12.4 Output

The place location will be displayed on a map.

3.2.12.5 Error Handling

The application will notify the personnel of the problem that occurred. Actions he might take to resolve the problem are also suggested.

3.2.13FR-13 Logout

3.2.13.1 Introduction

The application shall allow the User to log out of their respected accounts.

3.2.13.2 Inputs

None.

3.2.13.3 Processing

The application will clear the user information and the token stored on the local storage.

3.2.13.4 Output

The user should be logged out successfully and is directed to the login page.

3.2.13.5 Error Handling

The application will notify the personnel of the problem that occurred. Actions he might take to resolve the problem are also suggested.

3.3 USES CASES

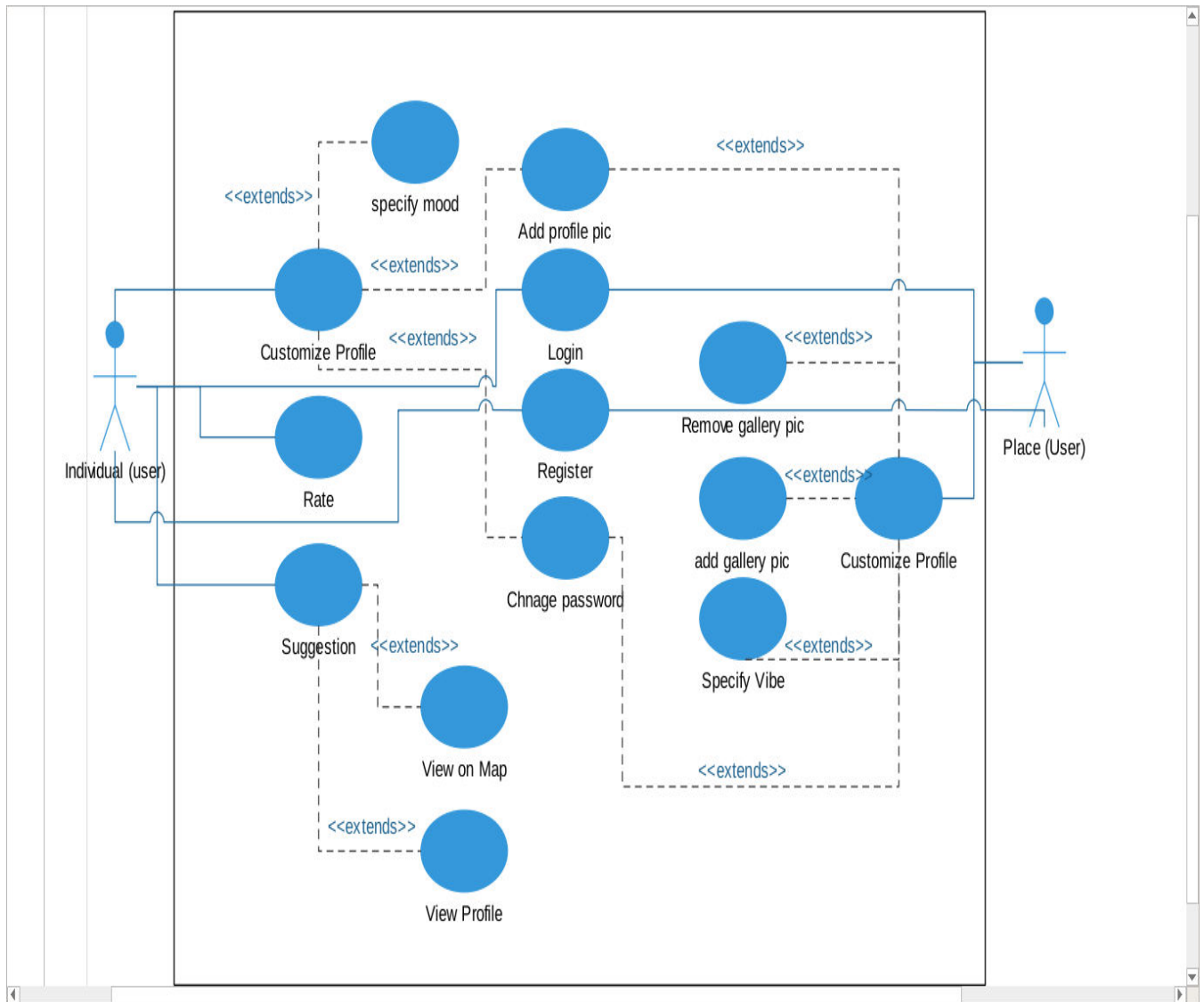


Figure 3.3.1 Use case diagram

3.3.1 UC-01 Customize profile

Primary actors: User (*Individual or Place*)

Goal: For user to customize their how their profile is.

Precondition: User must be logged in.

Success end: user customizes his/her profile.

Failure end: user can't customize his/her profile.

Trigger: user requests (Through the user interface) to edit their profile.

Main success scenario:

1. User inserts username and password at login screen.
2. System authenticates user and logs the user in.
3. System redirects user to home screen.
4. User requests to edit his/her profile.
5. System redirects to a screen where user can edit their profile.
6. User performs any of the following scenarios Add profile picture, Specify mood/vibe, Change password or remove/add picture.
7. System saves the changes made and updates their profile.

Extensions:

2a. Authentication fails

2a.1 System displays specific error message to user

3.3.2 UC-02 Add profile picture

Primary actor: both Individual and place.

Goal: to specify a picture to act as the profile photo.

Precondition: Users must be logged in, and at the customize profile screen.

Success end: Users add a new profile photo, and system displays it.

Failure end: No change will be made in the users profile photo.

Trigger: User requests to upload a profile photo, through the user interface.

Main Success Scenario:

1. User requests to upload a new profile photo.
2. System responds with a mechanism to choose a photo.
3. User selects the photo it wants.
4. System uploads the photo and notifies user of successful upload.

Extensions:

- 3a. User does not select a photo instead cancels.
 - 3a. 1 system notifies user exactly one photo must be chosen.
- 4a. System fails to upload photo.
 - 4a. 1 System notifies user of failure.
 - 4a. 2 system redirects user to the customize profile screen.

3.3.3 UC-03 Specify mood/vibe

Primary actor: both Individual and place.

Goal: to specify a mood or a vibe to describe the individual or the place.

Precondition: Users must be logged in, and at the customize profile screen.

Success end: Users add a new mood/vibe and the system displays it.

Failure end: No change will be made in the users' mood/vibe status.

Trigger: User requests to specify a new mood/vibe, through the user interface.

Main Success Scenario:

1. User requests to specify a new mood/vibe.
2. System lists the option available.
3. User selects the mood/vibe which it thinks fits it.
4. System saves the current mood/vibe status and redirects user to home screen.

Extensions:

4a. System fails to save current mood/vibe status.

4a. 1 System notifies user of failure.

4a. 2 system redirects user to the customize profile screen.

3.3.4 UC-04 Request Suggestions

Primary actors: User (*Individual only*)

Goal: User wanted a place suggestion based on his current mood.

Precondition: User must be logged in.

Success end: System suggests a place to the user.

Failure end: No suggestions will be made.

Trigger: User requests for a place to be suggested.

Main Success Scenario:

1. User requests for a place to be suggested.
2. System reads the current mood and searches for a place.
3. System displays found places.
4. User chooses to view on map.
5. System redirects user to an interactive map

Extensions:

2a. User has not supplied his/her current mood.

2a.1 System notifies and redirects user to fill his current mood.

2a.2 User fills mood or returns to home screen

3a. No places found that meets the user's mood.

3a.1 System notifies then redirects the user to home screen.

Alternatives:

4a. User chooses to view profile of the place

4a.1 System redirects user to the profile of the place.

4a.2 User chooses to view on a map or goes back to home screen

3.3.5 UC-05 Rate

Primary actors: Users (*Individual*)

Goal: User wishes to rate a place.

Precondition: User must be logged in, the place must have been suggested previously.

Success end: Users rating will be visible on the places profile.

Failure end: No change will be made to the profile of the place.

Trigger: user chooses to rate a previously suggested place.

Main success Scenario:

1. User performs Request suggestions scenario.
2. User visits place and tries to rate.
3. System displays the users rating.

3.3.6 UC-06 remove/add picture

Primary actor: Places.

Goal: to add or remove a picture from the gallery of the place

Precondition: User must be logged in, and at the customize profile screen.

Success end: Users add or remove a picture to their gallery and the system displays it.

Failure end: No change will be made in the users' gallery.

Trigger: User requests to add or remove picture, through the user interface.

Main Success Scenario:

1. User requests to add picture
2. System responds with a mechanism to select photo to upload
(Add to the gallery).
3. User selects the photo it wants.
4. System saves the new photo in the gallery.

Extensions:

- 4a. System fails to save the new photo.
 - 4a. 1 System notifies user of failure.
 - 4a. 2 system redirects user to the customize profile screen.

Alternatives:

- 1a. User requests to remove a picture.
 - 1a.1 System responds with a mechanism to select photo to remove the photo from the gallery.
 - 1a.2 User selects the photo it wants.
 - 1a.3 System removes the photo from the gallery.

3.3.7 UC-07 Sign up

Primary actor: Users

Goal: to become a registered user of the system.

Precondition: No preconditions.

Success end: User is registered in the system

Failure end: User is not registered

Trigger: User requests to be registered

Main Success scenario:

- 1. User requests to sign up as an individual.
- 2. System redirects user to the appropriate sign up screen
- 3. User provides necessary information and proceeds
- 4. System updates database and prompts user to home screen

Extensions:

- 3a. User doesn't provide sufficient information
 - 3a.1 System notifies user to provide sufficient information

Alternatives:

1a. User requests to register a place

1a.1 Continue from step 2

3.3.8 UC-08 Sign in

Primary actor: Users

Goal: To login in to the application

Precondition: no precondition.

Success end: User logs in to the application

Failure end: User doesn't log in to the application

Trigger: User requests to log in.

Main success scenario:

1. User inserts his/her credentials and requests to sign in
2. System Authenticates User and logs him in.
3. System presents user with their home screen

Extensions:

1a. Users requests to sign in without providing credentials

1a.1 the application notifies user

2a. Authentication fails

2a.1 the application notifies user

3.3.9 UC-09 Change Password

Primary actors: Users

Goal: To change current password

Precondition: User is signed in

Success end: The password is changed

Failure end: Password is not changed

Trigger: User requests to change password

Main Success Scenario:

1. User Requests to change password.
2. System prompts user to a screen where he can change his/her password.
3. User fills the necessary information and submits the form.
4. System Changes password and updates database.

Extensions:

3a. User submits without filling necessary information

3a.1 System notifies user

3b. the entered current password doesn't match the current Password.

3b.1 System notifies User

3.3.10 UC-10 Reset Password

Primary actor: User

Goal: To reset a forgotten password

Precondition: User has access to the app

Success end: The password is reset to a new password

Failure end: Password is not changed

Trigger: User requests to reset password

Main Success Scenario:

1. User requests to reset password
2. System redirects user to a screen where he can reset password.
3. System displays option(s) to recover/reset his password.
4. User Chooses an option and proceeds
5. Based on User's Choice, System notifies user of instructions to follow
6. User Follows the steps and resets password.

Extensions:

3a. User didn't choose an option

3a. System redirects to step2

3.3.11 UC-11 Logout

Primary actors: Users

Goal: To logout from user account

Precondition: User is logged in.

Success end: The user logged out successfully.

Failure end: User still logged in.

Trigger: User requests to log out.

Main Success Scenario:

1. User Requests to logout
2. The application clears the data stored on the local storage
3. The application prompts to login screen.

Extensions:

3a. the system fails to logout correctly

3a.1 User still logged in.

3a.2 System redirects user to the home page.

3. 4 NON-FUNCTIONAL REQUIREMENTS

3.4.1 Performance

- **Description** - View loading time

0.1 second is about the limit for having the user feel that the system is reacting instantaneously, meaning that no special feedback is necessary except to display the result.

1.0 second is about the limit for the user's flow of thought to stay uninterrupted, even though the user will notice the delay. Normally, no special feedback is necessary during delays of more than 0.1 but less than 1.0 second, but the user does lose the feeling of operating directly on the data.

Measurements - The response times will be measured using tools located behind the firewall and in front of the web servers. The timer will measure the time from the request for a page to when the last bit required rendering the page is returned. Backend response times will be measured using the application server log files.

- Database response time should not be greater than 0.5 seconds per query.
- The server should handle multiple client requests with a response time of not more than 1 sec per client.

3.4.2 Reliability

- The system should keep its data integrity by reflecting any change made, to data the database.
- The system should backup data in case of failure.
- Any type of failure on the client's app should not cause loss or damage of client's data.

3.4.3 Availability

- The system shall not be down for more than 30 minutes a day.
- The system must be available all the time, but since that's not humanly possible we plan on getting to 99% or less of availability.

3.4.3 Security

- In signing up the system will check whether there is an existing account before it and will prompt failure.
- The system will check the user's identity in logging them
- The system shall encrypt passwords when storing them.

3.4.5 Maintainability

- The application should be flexible and be capable of any adjustment (to add new features in the future) that would be applied to it.
- The application should also be open to testing its several functions.

3.4.6 Portability

- It shouldn't take more than 2 days for a skilled software engineer to port the system to other platforms (Environments).

3.5 INVERSE REQUIREMENTS

The system will not offer any type of private chatting to individuals or places.

3.6 DESIGN CONSTRAINTS

As of this writing we are not aware of any design constraints.

3.7 OTHER REQUIREMENTS

Training-related Requirements

There will not be any training provided.

4. Change Management Process

Any requests to change the project scope and requirements shall be discussed by all the members of the team. A change will be made only when the majority of the team agrees on the change. In this case, the SRS document shall be updated by the team members in order to reflect the changes, and a date of change shall be noted in the file. If this change request is made by the client or anyone outside of the team, he or she will have to contact the team. If a change request is made by a team member, he or she can raise it during the weekly team meeting or contact other team members via email. During any of these requests, the team will assess the feasibility of the proposed changes considering the time constraints and structural constraints of the implemented modules and develop an implementation strategy. A change plan will be created for the implementation of the change. The team will then continue implementing the new requirements.

References

- Wikipedia – feasibility engineering
- Ian Somerville – software engineering 9th edition
- The section named “Change management process” taken from samples, specifically “AYATE ETHIOPIAN HOME REMEDIES”.