
Software Requirements Specification

for

Feastfreedom.com

Version 1.0 approved

Prepared by: Thien Le, Ashutosh Gautam, Saritha K, Bo Mao

November,27,2018

Table of Contents

Introduction	4
Purpose	4
Document Conventions	4
Intended Audience and Reading Suggestions	4
Product Scope	4
References	4
Overall Description	5
Product Perspective	5
Product Functions	6
User Classes and Characteristics	7
Use-case model	7
Operating Environment	8
Design and Implementation Constraints	8
User Documentation	8
Assumptions and Dependencies	8
External Interface Requirements	9
User Interfaces	9
Hardware Interfaces	15
Software Interfaces	15
Communications Interfaces	15
System Features	16
System Feature 1	16
Other Nonfunctional Requirements	18
Performance Requirements	18
Safety Requirements	18
Security Requirements	19
Software Quality Attributes	19
Business Rules	19
Other Requirements	20

Revision History

Name	Date	Reason For Changes	Version
Bo Mao	11/27/2018	Section 1,3,5	v1.0
Thien Le	11/27/2018	Section 2,3	v1.0
Saritha K	11/27/2018	Section 2,4	v1.0
Ashutosh Gautam	11/27/2018	Section 5,6	v1.0

1. Introduction

1.1 Purpose

The purpose of this documents is to give a detail description of the requirements of the 'Feast Freedom'(FD) software. This document is intended to provide a background and functional specification for a product which can be used as a platform, such as Uber, Grubhub etc.

1.2 Document Conventions

This document is normal text style writing in standard English with Italic and also with blue text color and standard 11 font size in Arial font.

1.3 Intended Audience and Reading Suggestions

This project is a prototype for the food providing service and it is restricted within the Python training Patch. So the intended audience of this document are supervisors, trainer, developer. This project is useful for the homemade food provider and as well as to regular customer.

1.4 Product Scope

This 'feastfreedom' platform is a non-profit product, designed to benefit the kitchen providers and regular users. This designing allows the kitchen providers to post their menu to 'feastfreedom.com' website and the working days, working hours, the type of the food, which could be much easier, flexible, and convenient for regular users to take orders on the website. In this designing platform, it not allows one kitchen provider manage two or more kitchens under his/her account which registered already. Furthermore, this designing is not realizing the payment process.

1.5 References

This SRS refers to the project assignment issued by SummitWorks Inc. 'Python training batch' on November, 26, 2018.

2. Overall Description

2.1 Product Perspective

This product manages following information of a food providers:

- *Service Provider Information*
It included the service provider name, email, password
- *Kitchen information*
It contained working days, working time, image, and kitchen menu
- *Menu items information*
It included item name, item type such as veg/non-veg, and price
- *Customer information*
It included first name, last name, email, password, and two security questions and answers

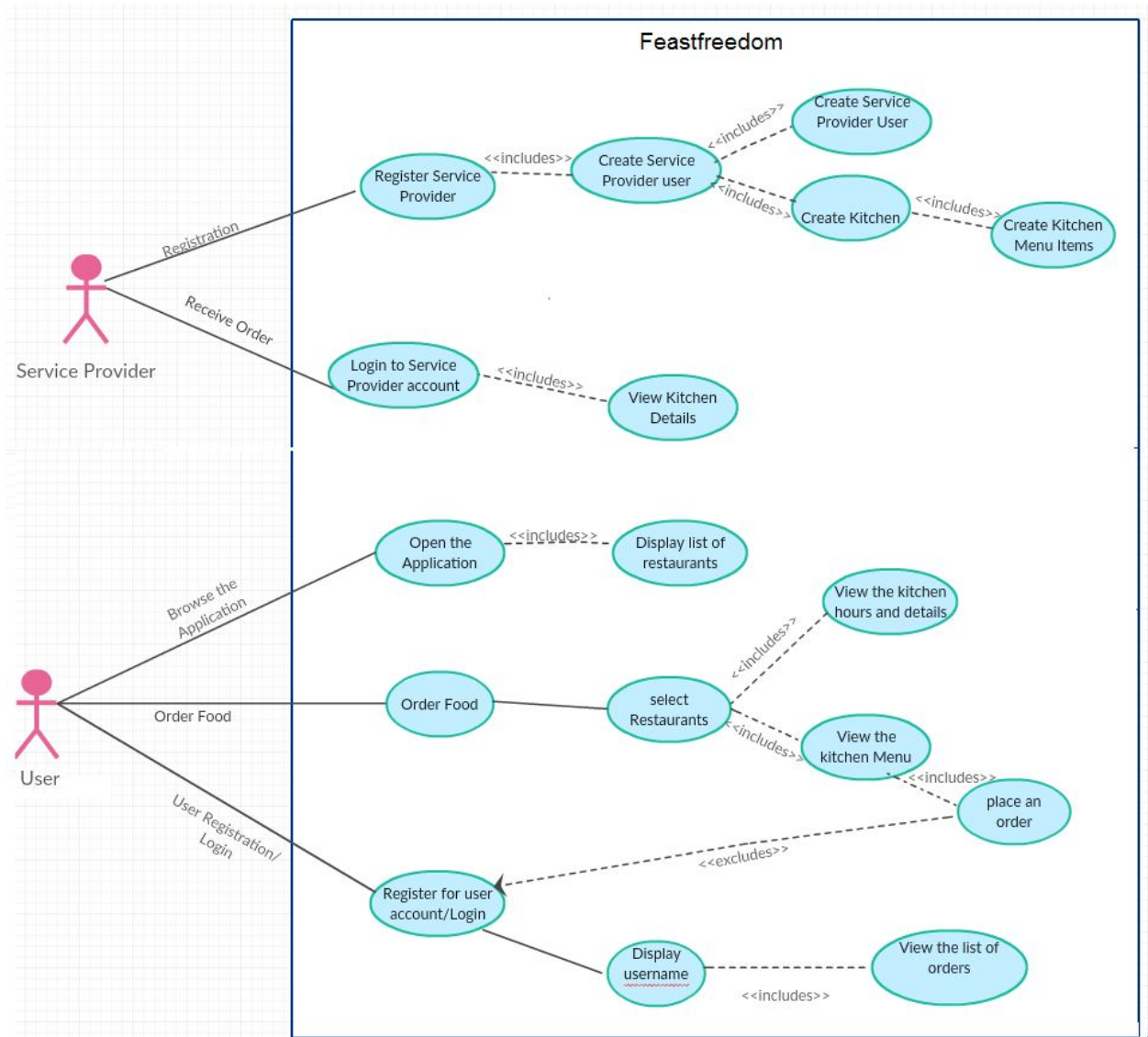


Figure 1 -- General use-case diagram for feastfreedom web application

2.2 Product Functions

The main purposes of Feastfreedom is to make a connection between consumer and homemade service service provider. The Feastfreedom will provided following functions:

- Service Provider Registration and adding kitchen
- Consumer/Regular User Registration
- Consumer can order item from kitchen
- Display total price of an order
- Edit the cart
- Sending email for Service Provider and user every order

2.3 User Classes and Characteristics

There are two user classes in this product, namely, service provider user and regular user. Service provider user is the user that can create kitchen and editing menu items. Regular user is able to view the detail of kitchen and menu items. Also, regular user can order food that listed in the kitchen details.

2.4 Use-case model

2.4.1 Service Provider User:

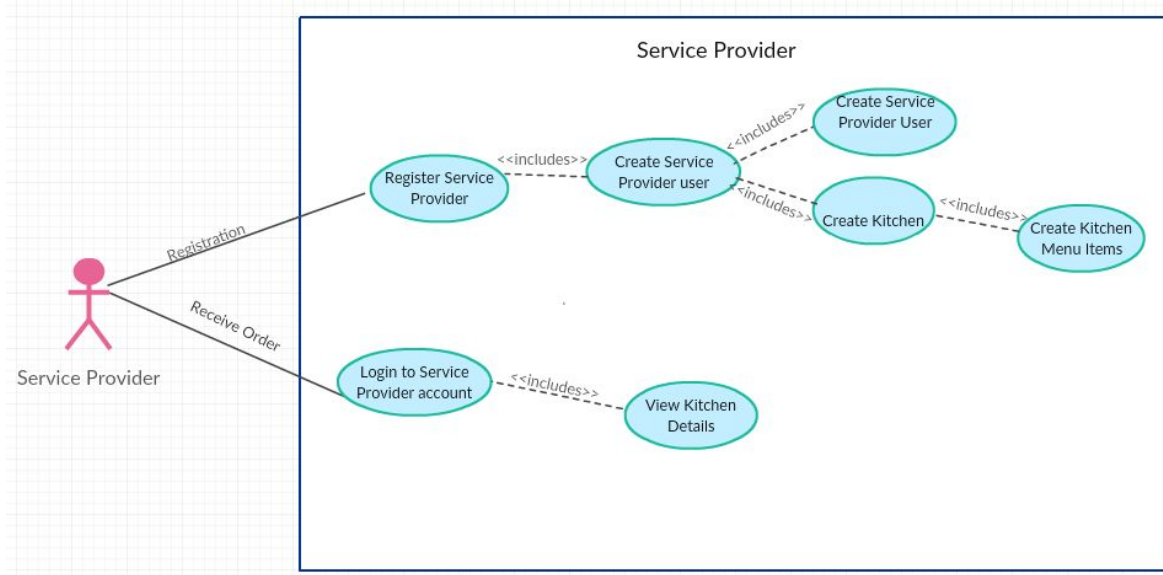


Figure 2 -- Service provider use-case diagram

Figure 2 show user use-case diagram. It describes the different use cases that can be executed by service provider users on feastfreedom web application.

2.4.2 Regular User:

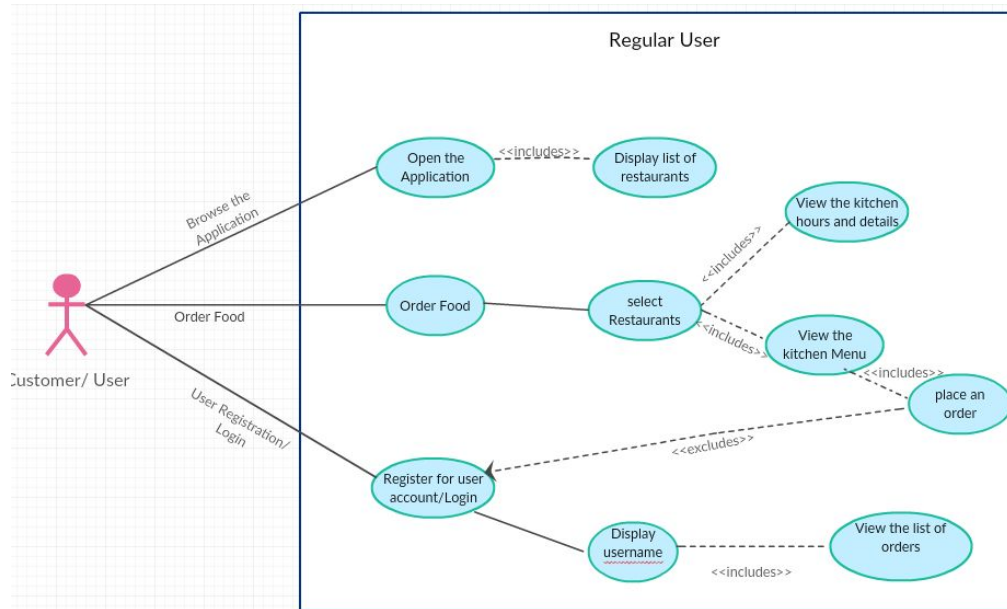


Figure 3 -- Regular user use-case diagram

Figure 3 shows regular user use-case diagram. It describes the different use cases that a regular users can be executed on feastfreedom web application.

2.5 Operating Environment

The project is setup and execute in Python virtual environment which can be run either on Windows system or MacOS.

2.6 Design and Implementation Constraints

Since the project is implemented in django framework / Python, the constraints of this project is the constraints of Python. Currently, the recommendation for system requirement as follows:

- Processors: Intel® Core™ i5 processor 4300M at 2.60 GHz or 2.59 GHz (1 socket, 2 cores, 2 threads per core), 8 GB of DRAM
- Disk space: 2 to 3 GB
- Operating systems: Windows® 10, macOS*, and Linux*

2.7 User Documentation

N/A

2.8 Assumptions and Dependencies

- <https://software.intel.com/en-us/distribution-for-python/system-requirements>
- Requirement Document: BrightRace & SummitWorks Pilot Project 1: FeastFreedom.com

3. External Interface Requirements

3.1 User Interfaces

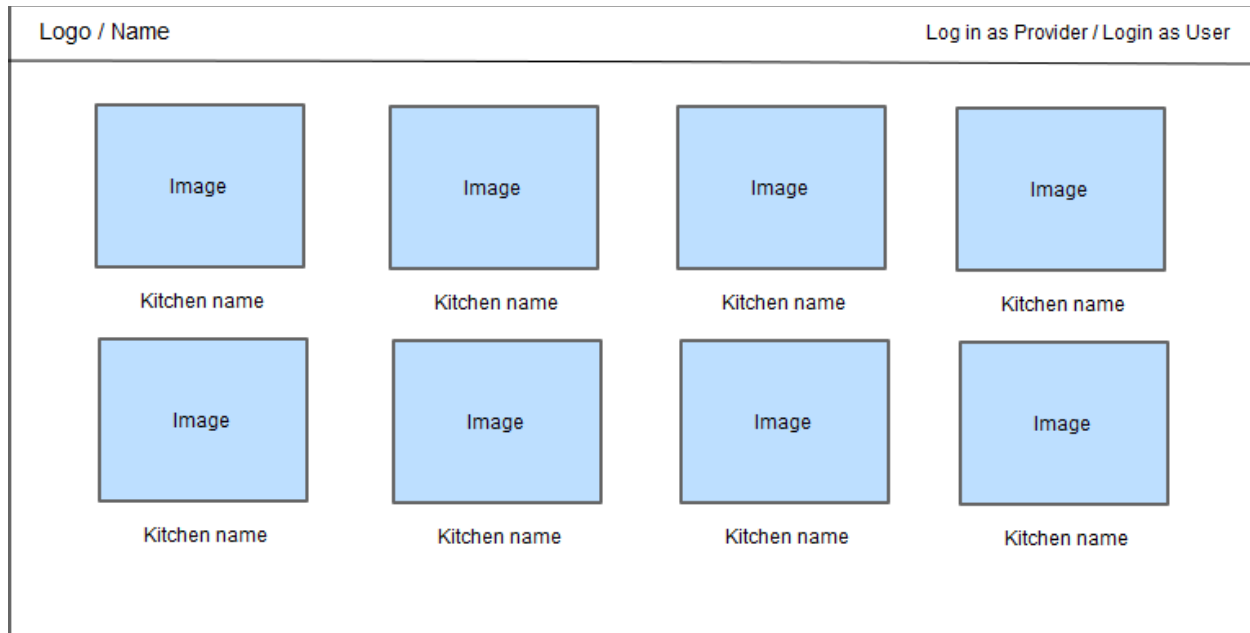
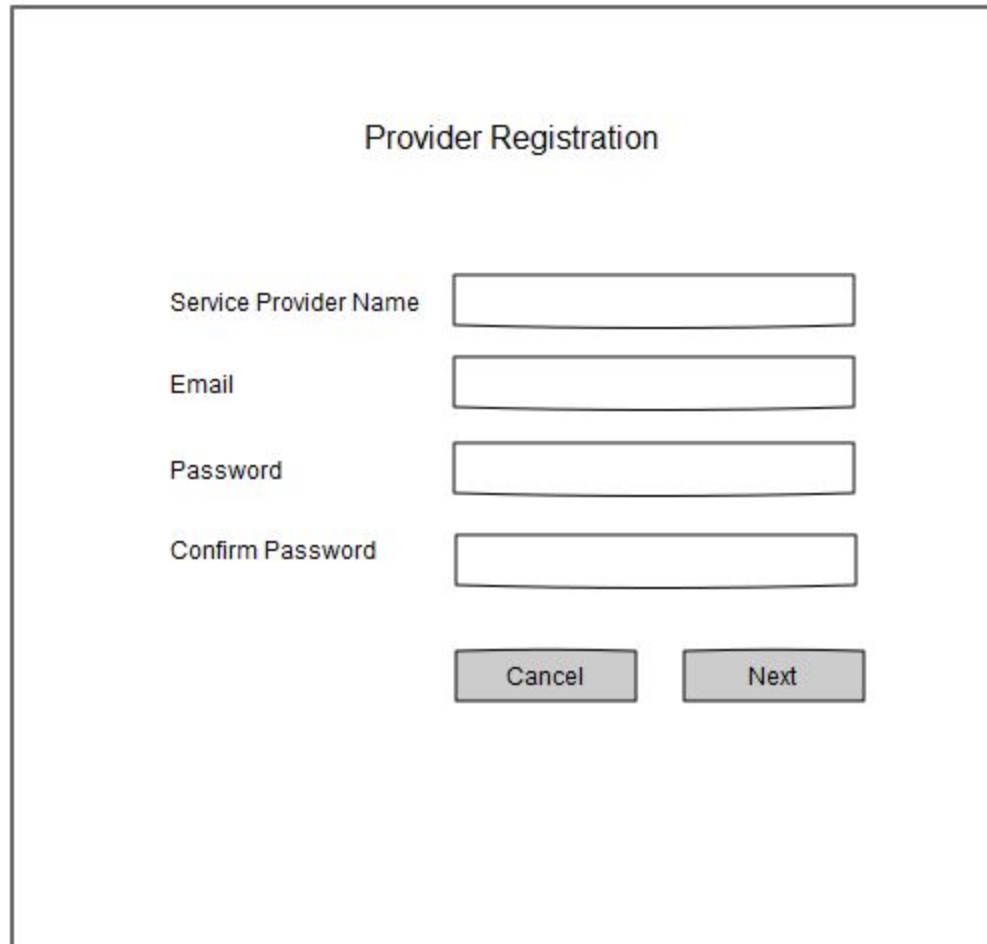


Figure 4 -- Main Home Page

In Figure 4, both users and providers will see this page, for first-time users, they need register before they can login.

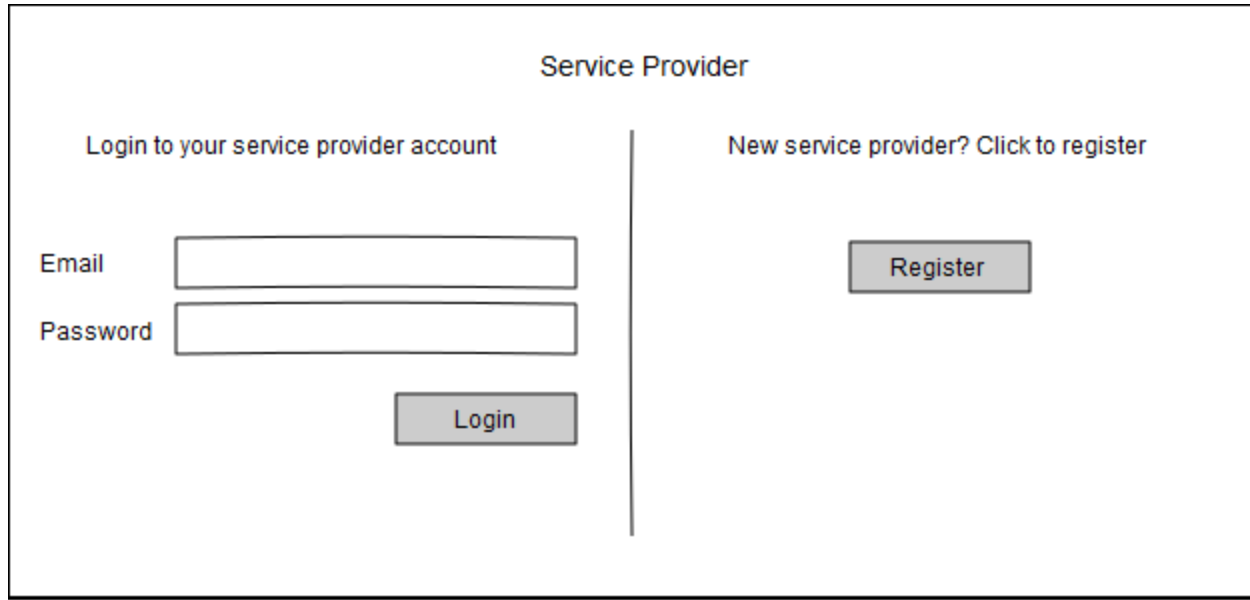


The image shows a web form titled "Provider Registration". It contains four text input fields arranged vertically, each preceded by a label: "Service Provider Name", "Email", "Password", and "Confirm Password". Below these fields are two buttons: "Cancel" and "Next". The entire form is enclosed in a rectangular border.

Provider Registration	
Service Provider Name	<input type="text"/>
Email	<input type="text"/>
Password	<input type="text"/>
Confirm Password	<input type="text"/>
<input type="button" value="Cancel"/> <input type="button" value="Next"/>	

Figure 5 -- Service Provider Register Page

Figure 5 is the register page for users, they need fill in service provide name, email and passwords



The image shows a web page layout for a 'Service Provider' login. The page is divided into two main sections by a vertical line. The left section is titled 'Login to your service provider account' and contains two input fields labeled 'Email' and 'Password', followed by a 'Login' button. The right section is titled 'New service provider? Click to register' and contains a 'Register' button.

Service Provider

Login to your service provider account

Email

Password

Login

New service provider? Click to register

Register

Figure 6 -- Service Provider login page

In Figure 6 the service providers can login using their registered email and password. Or service providers can redirect to register page if they have not registered.

Kitchen Form

Workday

☐ Monday
☐ Tuesday
☐ Wednesday
☐ Thursday
☐ Friday
☐ Saturday
☐ Sunday

Working Time

Start Time

▼

End Time

▼

Image

Upload image

Upload

Kitchen Menu:

Item Name

Check for Veg

☐

Price

Save Menu

Add Menu

Cancel

Save

Figure 7 -- Kitchen form

In Figure 7, the service providers need to complete the kitchen form in order to finish their registration. Service providers need to provide kitchen information and kitchen menu details, none of them can be null.


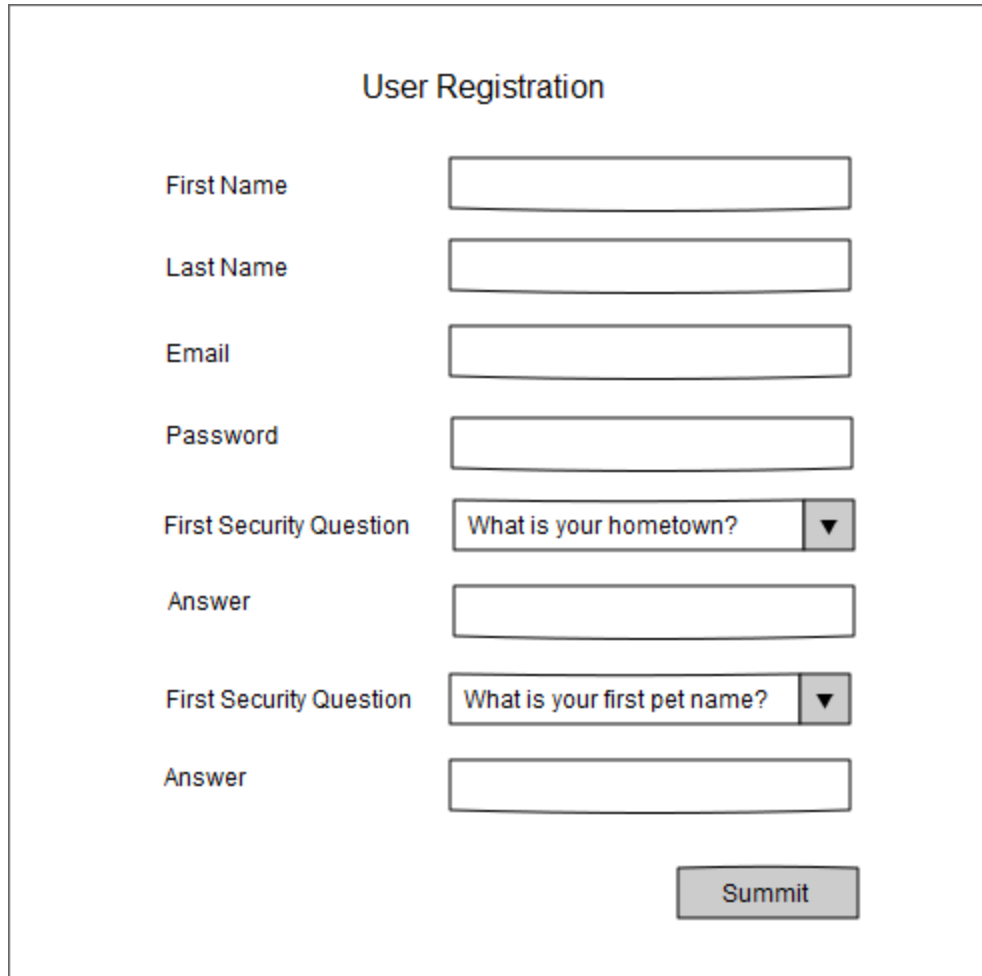
Logo / Name				User name
Kitchen name				
Orange Chicken	Vegan: no	Price: \$6.00	<input type="checkbox"/> Add to cart	
Red Chicken	Vegan: no	Price: \$6.00	<input type="checkbox"/> Add to cart	
Green Chicken	Vegan: yes	Price: \$6.00	<input type="checkbox"/> Add to cart	
Orange Salmon	Vegan: yes	Price: \$6.00	<input type="checkbox"/> Add to cart	
<input type="button" value="Add"/>				

Figure 8 -- Taking order page

In this figure, the users can choose any kitchen they love and click 'Add' button to add checked items into the cart.

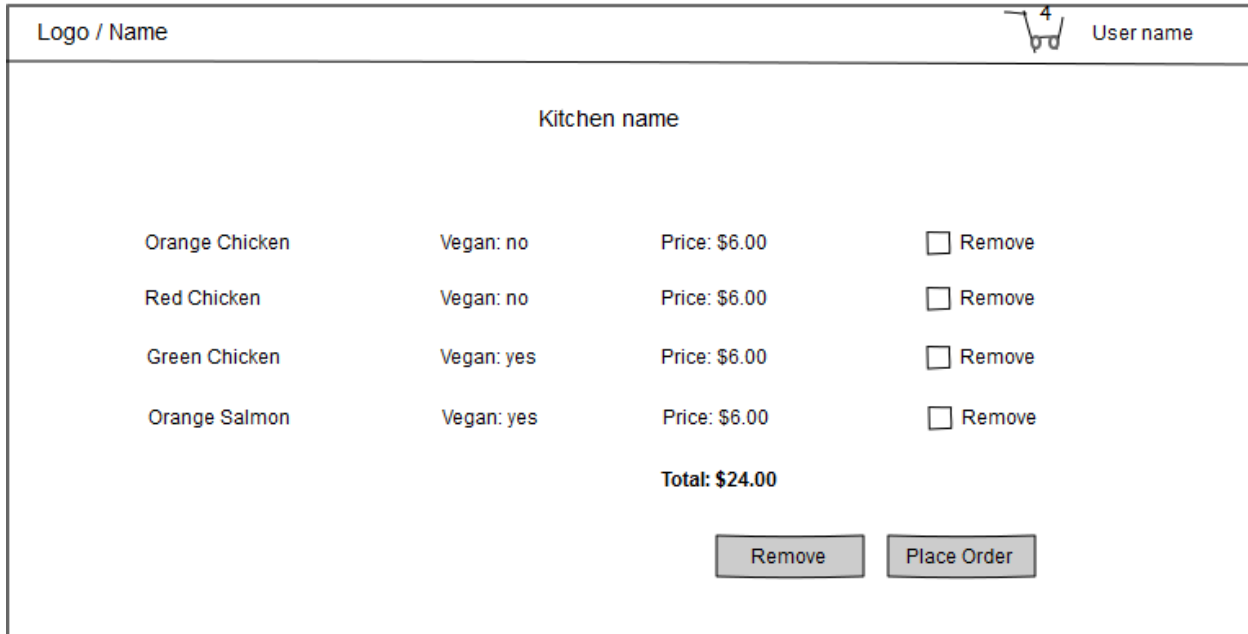


The form is titled "User Registration" and is enclosed in a rectangular border. It contains the following fields and controls:

- First Name:** A text input field.
- Last Name:** A text input field.
- Email:** A text input field.
- Password:** A text input field.
- First Security Question:** A dropdown menu with the selected option "What is your hometown?" and a downward arrow icon.
- Answer:** A text input field.
- First Security Question:** A second dropdown menu with the selected option "What is your first pet name?" and a downward arrow icon.
- Answer:** A text input field.
- Submit:** A button located at the bottom right of the form.

Figure 9 -- Regular User Register Page

In this figure, regular users register their information, they need fill in first name, last name, email, password and two security and passwords.



Logo / Name		User name	
Kitchen name			
Orange Chicken	Vegan: no	Price: \$6.00	<input type="checkbox"/> Remove
Red Chicken	Vegan: no	Price: \$6.00	<input type="checkbox"/> Remove
Green Chicken	Vegan: yes	Price: \$6.00	<input type="checkbox"/> Remove
Orange Salmon	Vegan: yes	Price: \$6.00	<input type="checkbox"/> Remove
Total: \$24.00			
		<input type="button" value="Remove"/>	<input type="button" value="Place Order"/>

Figure 9 -- Cart

Figure 4 is the shopping cart where the users can remove items and place order after finishing picking their food order process. Additionally, they will see the total pricing displayed on that page.

3.2 Hardware Interfaces

A browser which supports FastCgiModule, HTML & Javascript.

3.3 Software Interfaces

- Front-end software: django, html template
- Back-end software: django, sqlite

3.4 Communications Interfaces

This project supports all types of web browsers. The project uses HTTP services to receive the request and send the response. We are using simple registration forms, food ordering etc.

4. System Features

The feature of this project is best explained by its use case diagram (Page 6, Figure 1 and Page 7, figure 2).

The entire operation is contained within service provider and user/customer module, with Kitchen, menu and ordering being sub module within provider, which is accessed by the customer.

4.1 System Feature 1

4.1.1 Description and Priority

- *Registration: High priority (User and provider registration)*
- *Kitchen Description: High priority (Kitchen Description by provider, along with menu)*
- *Ordering: High priority (The main feature of this project)*
- *Password security: Medium priority (Since the project is based on local machine, and not over the internet)*
- *Order Canceling: Medium priority*
- *Scaling: least priority (Since the project will not have large number of users/providers)*
- *Performance: least priority (Performance features given by the system is more than enough, no extra measure is needed)*

4.1.2 Stimulus/Response Sequences

- *stimulus: users request to register*
- *response: system provides a form to the user and show successful page once finish register*
- *stimulus: providers request to register*
- *response: system provides a form to the provider and show successful page once finish register*
- *stimulus: users request to cancel the order due to different reasons*
- *response: system delete all the information for that order and send a message to users about order cancelation*
- *stimulus: providers request to delete some existing menu items*
- *response: system delete the request item in database and display the result in both providers and users webpage*

4.1.3 Functional Requirements

ID:FR1

Feature: Register an account in the FeastFreedom Application

In order to create an account a service provider should register

Scenario: Required information for registration

The service provide provider owner wants to create an account and the restaurant owner does not have an account

When the restaurant owner registers on the web-portal by providing Service provide name,email , password and confirm password then the restaurant owner should be able to apply for verification

Scenario: Confirmed registration

One the restaurant owner finished with the registration form then the restaurant owner should be able to log in

ID: FR2

Feature: The service provider log-in

In order to use the system the service provider should be logged in to the web-portal

Scenario: Successful log-in

When the service provider logs in with his/her account then the service provider should be logged in as a service provider

ID: FR3

Feature: Manage information

In order to manage information the service provider should be logged in to the web-portal

Scenario: Show fields for managing information

When the service provider wants to manage information the service provider should be able to manage information in a form

Scenario: Manage restaurant dishes

In order to have a list of dishes the service provider should be able to manage the dishes

Scenario: Add a new dish

The service provider should be log in , when the service provider creates a new dish Then the new dish should be added to the list of dishes

Scenario: Editing an existing dish

The service provider should be log in , when the service provider edits an existing dish then the dish should be updated in the list of dishes

Scenario: Delete a dish

The service provider should be log in when the service provider deletes a dish then the dish should be removed from the list of dishes

ID:FR4

Feature: Register an account in the FeastFreedom Application

In order to create an account a regular user should register

Scenario: Required information for registration

The regular user wants to create an account and the user does not have an account

When the user registers on the web-portal by providing first name, last name, email, password and two security questions & answers.

Scenario: Confirmed registration

One the regular user finished with the registration form then the user should be able to log in

ID: FR5

Feature: Order Food

In order to order the food the regular user should be logged in to the web-portal

Scenario: Show fields for ordering the food

When the regular user wants to order the food, the regular user should be logged in to the web-portal and then all the menu items should be viewable to user .

Scenario: Place an order

In order to place an order, he must add food items to "add to cart" then place an order. before he place an order either he must be register or should be logged in.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

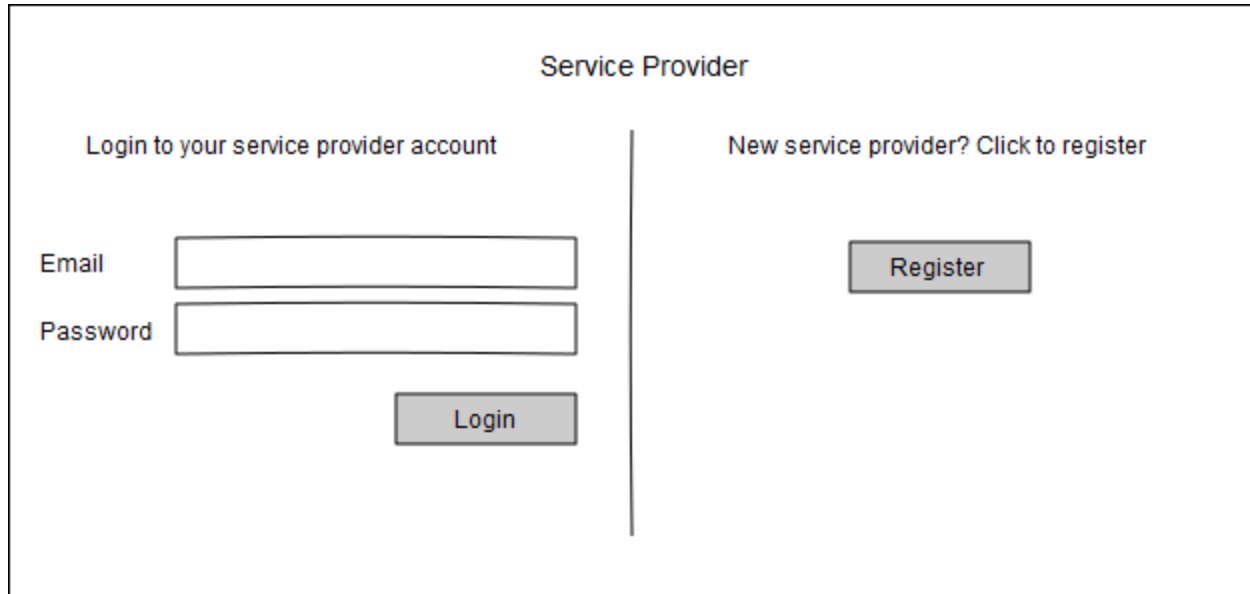
The software for the project does not have high scalability, and thus should be kept for limited number of users (below 50 users and/or providers). Apart from basic system features, no additional measure is taken to conserve time and memory. Since the project is not of large scale, time and memory are of least importance. As no transaction is involved, and only order confirmation is counted, the project does not suffer from precision or accuracy issues.

5.2 Safety Requirements

The provider and customer have personal information and password stored over the database, over which standard database security measures is applied. Private customer information is not shared with provider and vice versa. No additional security measure is applied. the project is not shared over the internet, and works in local machine, making accessibility constraint its best security feature.

5.3 Security Requirements

The first page of the project is like this:



The image shows a web form titled "Service Provider" at the top center. The form is divided into two sections by a vertical line. The left section is titled "Login to your service provider account" and contains two input fields labeled "Email" and "Password", followed by a "Login" button. The right section is titled "New service provider? Click to register" and contains a "Register" button.

Figure 10 -- Service Provider Login

Every user need to have their email and password to access their profile. In case of a forgotten password, the user has an option to answer a security question (for which they have submitted an answer while registering). If the information given is correct, the user receives an email with this password in it.

5.4 Software Quality Attributes

For the shopping cart, it's functionally designed to enable users to change the quantities of the products. In this case, users do not need to add a same product by clicking 'Add' button too many times, on the contrary, users can pick on item and change the quantity to the amount they want. Furthermore, the application allows automatically check the stocks, once the demand is out of the stocks from the providers side, an error will show up to request users reduce the specific shopping items.

5.5 Business Rules

Within the application, providers are the only members can access to edit the kitchen details, which regular users only have authority to view the kitchens and the menus. For the providers, they cannot shop through the application by login their providers' account.

6. Other Requirements

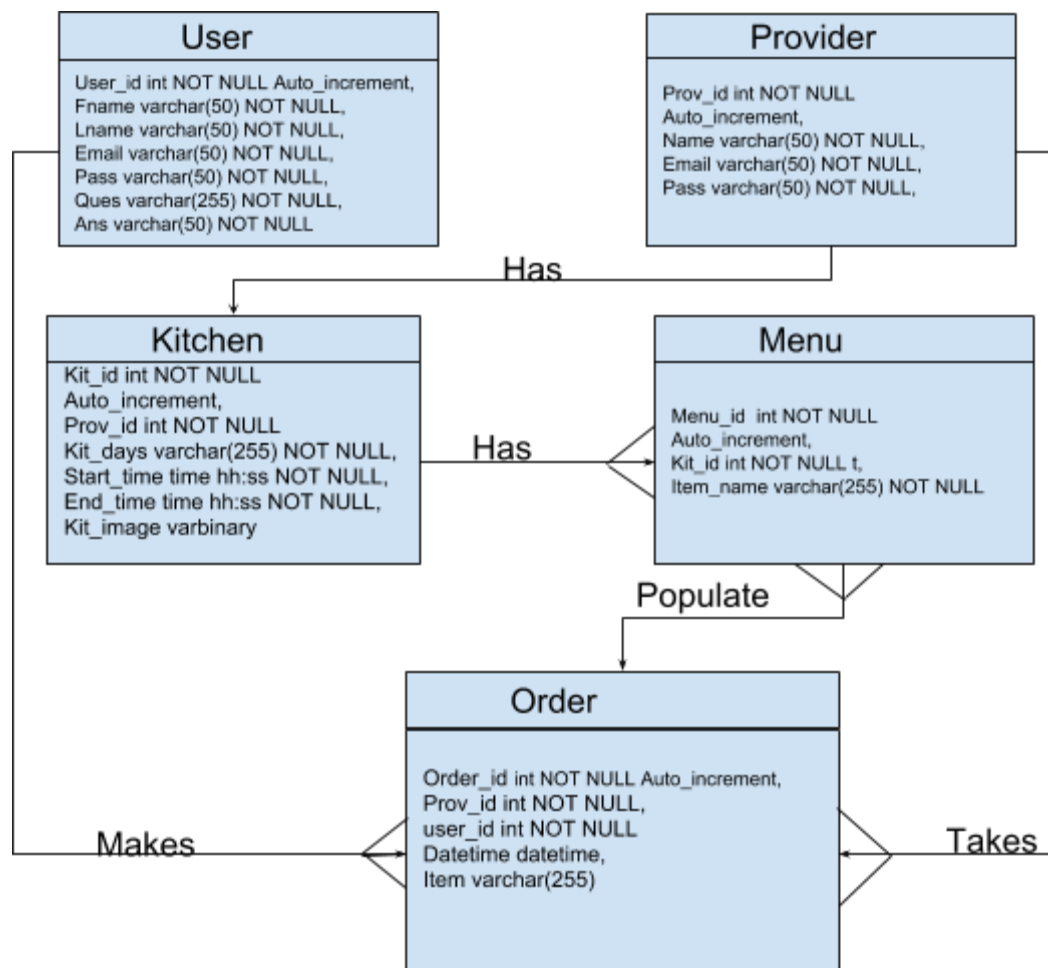
N/A

Appendix A: Glossary

- *FD*: The acronym for the feast freedom website.
- *SQL*: Structured Query Language, Language used to communicate with the database.
- *Python*: An interpreted high level programming language.
- *Django*: Python based free and open source web framework.
- *Operating Systems*: System software that manages computer hardware and software resources.

Appendix B: Analysis Models

Entity-Relationship Diagram to better analyse the flow of the project. and the relation between each module it contain



Appendix C: To Be Determined List

- *Django Template analysis: Checking each template used for this website for their compatibility, efficiency and consistency.*
- *PSRS Testing: Testing performance, scalability, reliability and security of the project to see how well it is standardized, how robust, and how safe to use it is.*