## 

**Final Year Project Report**

**Food Point**

****

**Session**

**University of Management and Technology**

**C-II Johar Town Lahore Pakistan**

**Dedication**

This project work is dedicated to our teachers, who have been a constant source of support and encouragement during the challenges of graduate school and life. We are truly thankful for having such teachers in our life. This work is also dedicated to our parents, who have always loved us unconditionally and whose good examples have taught us to work hard for the things that we aspire to achieve.

## 

**Acknowledgment**

We would like to express our deepest gratitude to God for his guidance and everything He has done to us. We are also greatly indebted to our supervisor, Mr. Arslan Asif, who was of utmost importance to the completion of this endeavor. We have benefited a lot from his comments and suggestions, and for innumerable times, He puts us on the right track and gives us invaluable insight into how the study should be conducted. Without his advice and guidance, it would have been impossible for us to complete this work.

We must send our heartfelt thanks to our colleagues and friends who gave us unlimited support and encouragement. Special thanks are also due to the students who were involved in this project’s Preliminary Analysis. Without their cooperation, this Project could not have been initialized.

**Project Title:** Food Point

**Objective**

We expect our application to help people into eating better food according to their specified taste preference through advanced recommendation system and also helping them engage into food community to explore about new types of food.

**Undertaken by**

**Supervised by:**

**Starting Date:**

**Completion Date:**

**Tools Used:**

* Machine Learning
* Django Framework
* MySQL
* Visual Paradigm
* Visual Studio Code
* Jupyter Notebook

**Operating System:** Windows 10

**Abstract**

This Project is designed for a wide range of customers as well as restaurants. This work is conducted by using different machine learning algorithms such as Sentiment Analysis using Support Vector Machine into a web application. All work here is done by following using steps of SDLC.

This work is done to assist customers into ordering food that they actually enjoy by an automated recommendation system that trains on their personal taste preference and generates recommendations based on those preferences.

If customer still finds any further difficulty in choosing food he can engage in community within the application and create posts about his food adventures.

Revision Chart

| Version | Primary Author(s) | Description of Version | Date Completed |
| --- | --- | --- | --- |
| **Draft** |  | Initial draft created for distribution and review comments |  |
| **Preliminary** |  | Second draft incorporating initial review comments, distributed for final review |  |
| **Final** |  | First complete draft, which is placed under change control |  |
| **Revision 1** |  | Revised draft, revised according to the change control process and maintained under change control |  |
| **Revision 2** |  | Revised draft, revised according to the change control process and maintained under change control |  |

# Contents

[1. Contents 1](#_Toc52384369)

[List of Figures: 3](#_Toc52384370)

[Definitions and Acronyms 5](#_Toc52384371)

[Introduction 6](#_Toc52384372)

[Motivations 6](#_Toc52384373)

[1.1 Project Overview 6](#_Toc52384374)

[1.2 Problem Statement 6](#_Toc52384375)

[1.3 Objectives 7](#_Toc52384376)

[2. Domain Analysis 8](#_Toc52384377)

[2.1 Customer 8](#_Toc52384378)

[2.2 Stakeholders 8](#_Toc52384379)

[2.3 Affected Groups with social or economic impact 9](#_Toc52384380)

[2.4 Dependencies/ External Systems 9](#_Toc52384381)

[2.5 Reference Documents 9](#_Toc52384382)

[2.5.1 Related Projects 9](#_Toc52384383)

[3. Requirements analysis 11](#_Toc52384384)

[3.1 Requirements 11](#_Toc52384385)

[3.2 List of Actors 14](#_Toc52384386)

[3.3 List of use cases 15](#_Toc52384387)

[3.4 System use case diagram 16](#_Toc52384388)

[3.5 Use Cases 17](#_Toc52384389)

[3.6 User interfaces (mock screens) 35](#_Toc52384390)

[4. Data Flow Diagram 42](#_Toc52384391)

[5. System Design 43](#_Toc52384392)

[5.1 System Architecture Diagram 43](#_Toc52384393)

[5.2 Class Diagram 45](#_Toc52384394)

[5.3 Sequence Diagrams 46](#_Toc52384396)

[5.4 Collaboration Diagrams 47](#_Toc52384397)

[5.5 ERD 47](#_Toc52384398)

[6. ImplementatiOn details 48](#_Toc52384399)

[6.1 Development Setup 48](#_Toc52384400)

[6.2 Deployment setup 48](#_Toc52384401)

[6.3 Algorithms 48](#_Toc52384402)

[6.4 Constraints 49](#_Toc52384403)

[6.4.1 Assumptions 49](#_Toc52384404)

[6.4.2 System constraints 49](#_Toc52384405)

[6.4.3 Restrictions 49](#_Toc52384406)

[6.4.4 Limitations 49](#_Toc52384407)

[7. Testing 50](#_Toc52384408)

[7.1 Test Cases 50](#_Toc52384409)

[7.2 Code Snippets 62](#_Toc52384410)

[7.3 Traceability Matrix 65](#_Toc52384411)

[7.3.1 RID vs UCID (requirements vs use cases) 65](#_Toc52384412)

[7.3.2 PID vs TID (Prototype vs Test Case) 67](#_Toc52384413)

[7.3.3 RID vs TID (Requirements vs Test Case) 68](#_Toc52384414)

[8. Results/Output/Statistics 69](#_Toc52384415)

[Completion 69](#_Toc52384416)

[8.2 Accuracy 69](#_Toc52384417)

[8.3 Correctness 69](#_Toc52384418)

[9. Conclusion 70](#_Toc52384419)

[10. Future work 71](#_Toc52384420)

[11. Bibliography 72](#_Toc52384421)

[11.1 Books 72](#_Toc52384422)

[11.2 Articles 72](#_Toc52384423)

[11.3 Research papers 72](#_Toc52384424)

[11.4 Other References 73](#_Toc52384425)

[12. Appendix 74](#_Toc52384426)

[12.1 Glossary of terms 74](#_Toc52384427)

[12.2 Pre-requisites 75](#_Toc52384428)

## Definitions and Acronyms

Table 1: Table of acronyms and definitions

|  |  |
| --- | --- |
| **Acronym** | **Definition** |
| UMT | University of Management and Technology |
| POS | Point of Sale |
| UI | User Interface |
| UC | Use Case |
| P1, P2, etc.… | Prototype 1, Prototype 2, etc.… |

## Introduction

Our system is an online food recommendation-based application which will provide food ordering services and also help customers engage in growing food community within the application to let them experience new types of food.

This system will be beneficial for both the customer and restaurant since customer will be getting personalized food recommendations and restaurants will be highlighted which will match customer’s particular interest.

Without a Proper Recommender system, all restaurants will be displayed which can confuse customers into choosing a restaurant which do not match their taste and end up having a bad experience which leads to customer never using app again (we will avoid that).

Unlike tradition food ordering apps which rely on general ratings of restaurant to provide customer recommendations, our system will be trained on customer’s textual reviews, ratings and then generate recommendations according to customers specified taste/choice. Community will also have a large impact on an engaged customer’s choice.

## Motivations

Our Motivation behind this project is to assist the way customers order their food by providing them specified recommendations based on his/her taste preference. In most scenarios when customer orders food, if they don’t like it, they don’t order again from the application which we want to overcome. We also want to make our application into a social hub where different users of different taste gather and interact while evolving their taste preference as they

## Project Overview

## Problem Statement

Why should a system only be limited to ordering food? If a customer can engage in the community, read experience of others and share his own, he can expect better insight about a restaurant and so can others.

Mostly the problem persists is that an un-initiated user has no idea about the food quality a restaurant is serving.

Without a Proper Recommender system, all restaurants will be displayed which can confuse customers into choosing a restaurant which do not match their taste and end up having a bad experience.

## Objectives

We expect our application to meet the following criterion:

* + Restaurant can be successfully register and manage their menus
  + Customers can successfully register and place orders.
  + System will recommend restaurants based on customer’s ratings/feedback.
  + Customer can join community, share and learn experience related to food.

# Domain Analysis

## Customer

This system is being made keeping Both Customers and Restaurants themselves in mind. Since Customers (Local People) will be getting personalized food recommendations and restaurants will be highlighted which will match customer’s particular interest even more irrespective of its size and scale.

## Stakeholders

List of all stakeholders along with their roles in making of the system e.g.

Table 2: List of stakeholders

|  |  |
| --- | --- |
| **Stakeholder** | **Role in System** |
| Customer | Customers are responsible for searching Restaurants and choosing food items from individual restaurant’s menu. After Completion of their orders Customer are responsible and incentivized to rate for their order.  Customers are also able to Access Community and Posting there (In form of Text Descriptions and images and videos) about their food experience while also liking other posts and Commenting on them. |
| Restaurant | Restaurants will be able to Create profile (Restaurant’s Name, Location, Contact, and Timing) in our system and add different categories and Food items (Dishes).  Restaurants are also responsible for accepting order requests from customers and updating the status of their order respectively (Order Preparing, Order Dispatched, Order Delivered) |
| Admin | Admin are responsible to create restaurant’s account in our system and can also manage their profile and menus.  Admins can Access Customers Information and viewing their order history while also having Access to all restaurants Profile information. |

## Affected Groups with social or economic impact

* Customers:

This app will help customers by providing them ease of ordering and personalized recommendations based on their taste and ordering preferences and location. This will prevent customers from accidently ordering food items that they don’t like or are generally bad in quality.

* Restaurants**:**

This app will assist restaurants in being highlighted more for customers irrespective of their size and scale, helping them focusing more on their food quality and less on advertisement. This will prevent restaurants being discriminated only on based of general ratings.

## Dependencies/ External Systems

* **[FOR WEB]** OS (Windows 7 or Later) with Browser (Firefox, Chrome, Ms. Edge etc.) with Internet Connectivity.

## Reference Documents

### Related Projects

* **FOODPANDA**

Food panda is a German mobile food delivery marketplace operating in 40+ Countries. This service allows users to select from local restaurants and place orders via its mobile applications as well as its websites.

* **UBER EATS**

Uber Eats' parent company Uber was founded in 2009 by Garrett Camp and Travis Kalanick. The company made its Entrance into food delivery scene in august of 2014 with the launch of the UberEATS service. In 2015, that service was later renamed to UberEATS and the ordering software was released as its own application, separate from the app for Uber rides.

**Feature Comparison**

Table 3: Feature Comparison

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr No.** | **Comparison Feature** | **Foodpanda** | **Uber Eats** | **Remarks**  **(Significance in our system)** |
| **1** | Recommendation | Foodpanda recommends all the restaurants based on location | Uber Eats Recommend restaurants on feasibility of their vehicle routes. | Our system will apply proper recommend the restaurants according to the user feedback |
| **2** | Community Integration | Foodpanda does not support this feature | Uber Eats does not support this feature | Our system will integrate proper community feature where users will come and post about their experiences and have proper human interaction that can enhance their existing knowledge about a restaurant or a particular food item. |

# Requirements analysis

## Requirements

Table 4: Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **RID** | **description** | **Category** | **Attribute** | **Details & Boundary Constraints** |
| FR-1 | Register | Functional | System | Users can create account in the system. |
| FR-2 | Login | Functional | System | User can Sign into their account after registration |
| FR-3 | Logout | Functional | System | User can logout of their Signed in Profile |
| FR-4 | Search | Functional | System | Customer can search for specific Restaurant |
| FR-5 | Add to Cart | Functional | System | Customer can add selected item(s) to cart |
| FR-6 | Manage Cart | Functional | System | Customer can Remove or Update Cart |
| FR-7 | Place Order | Functional | System | Customer can place order from cart. |
| FR-8 | Review | Functional | System | Customer can provide Textual and Numeric (Rate) review after Order Completion |
| FR-9 | Customer Order History | Functional | System | Customer can View Order History. |
| FR-10 | Rate Previous Order | Functional | System | Customer can rate any previous Orders that were not Reviewed. |
| FR-11 | Community Posting | Functional | System | Customer Can create Post in Community. |
| FR-12 | Community Interaction | Functional | System | Customer can Like a post and post comments on them |
| FR-13 | Manage Profile | Functional | System | Customers can modify their profile such as Name, Email, Contact and Profile Photo |
| FR-14 | Forget Password | Functional | System | Users can reset their password to a new one. |
| FR-15 | Manage Menu Categories | Functional | System | Restaurant can add and Remove different food categories into menu |
| FR-16 | Manage Food Items | Functional | System | Restaurant can add and remove food items in their respective categories. |
| FR-17 | Add Discount | Functional | System | Restaurant will be able to apply flat discount. |
| FR-18 | Manage Restaurant Information | Functional | System | Restaurant can Modify their profile such as Name, Location, Profile Photo and Contact. |
| FR-19 | Order Requests | Functional | System | Restaurant can either Accept order request generated by Customer or Reject it |
| FR-20 | Order Status | Functional | System | Restaurant can update Customer order by changing it between Preparing, Order Dispatched and Order Delivered |
| FR-21 | Restaurant Order History | Functional | System | Restaurant can view their previous order history. |
| FR-22 | View Order Information | Functional | UI | Admin can view information of both customer and restaurant regarding orders |
| FR-23 | Restaurants Registration | Functional | System | Admin can register a restaurant in the system. |
| FR-24 | Modify restaurant status | Functional | System | Admin can change a restaurant status to active or inactive. |
| FR-25 | Manage restaurant profile | Functional | System | Admin can manage restaurant’s profile |
| NFR-1 | Login Response Time | Non-Functional | Response Time | The user will login in system within 0.3 seconds. |
| NFR-2 | System Response Time | Non-Functional | Response Time | The response time of the system will be 0.30 seconds. |
| NFR-3 | Cart Updating delay | Non-Functional | Response Time | The system will take 0.5 seconds time to Update the Cart. |
| NFR-4 | Community Posts Load time | Non-Functional | Response Time | The system will take 3 seconds time to display the posts in Community Tab |
| NFR-5 | Inconsistent Data Check | Non-Functional | User Information | The system will check for incomplete and accurate entry. |
| NFR-6 | Generation of Reports | Non-Functional | User Information | The system will generate reports when required. |
| NFR-7 | Modification/View Data | Non-Functional | User Information | Profile creation and modification can only can only be done by customers and Restaurants. Rest of operations will be handled by Admin. |
| NFR-8 | Response Time | Non-Functional | User Information | System response will be so fast. User don’t have to wait more than 1 second. |
| NFR-9 | Security | Non-Functional | User Information | Due to Django, the system will be completely secured |
| NFR-10 | Compatibility | Non-Functional | User Information | Website can be accessed through browsers which support HTML 5 and above e.g. Firefox, Chrome, Explorer |

## List of Actors

Table 5: Actors

|  |  |
| --- | --- |
| **ACTORS** | **DESCRIPTION** |
| Admin | Admin will give access to account users to their profile and perform many appropriate actions. |
| Customer | Customers will receive recommendations on restaurants and will order food items and provide reviews on them |
| Restaurants | Restaurants will Receive Customer’s Order Requests and process them accordingly |

## List of use cases

Table 6: List of Use Cases

|  |  |  |
| --- | --- | --- |
| **USE CASE ID** | **PRIMARY ACTORS** | **USE CASES** |
| UC-1 | Customer | Customer Should be able to Register with his credentials on the website |
| UC-2 | Customer, Restaurant, Admin | User Should be Able to Login with his Correct Credentials such as Username and Password. |
| UC-3 | Customer | Customer Should be Able to search for a Specific Restaurant |
| UC-4 | Customer | Customer Should be able to add food items to the cart |
| UC-5 | Customer | Customer should be able to Update or Modify his Existing Cart |
| UC-6 | Customer | Customer should be able to place order from any restaurant |
| UC-7 | Customer | Customer Should be able to post in community. |
| UC-8 | Customer | Customer Should be able to provide review after completion of his order |
| UC-9 | Customer | Customer Should be Able to View his complete order History |
| UC-10 | Restaurant | Restaurants should be able to create and modify categories for their food items |
| UC-11 | Restaurant | Restaurants should be able to add food items to created categories |
| UC-12 | Restaurant | Restaurants should be able to Accept and Reject Customers Order Requests when received |
| UC-13 | Restaurant | Restaurants should be able to set and update order status of Accepted orders |
| UC-14 | Restaurant | Restaurants should be able to View complete Order history |
| UC-15 | Admin | Admin should be able to view customers profile information |
| UC-16 | Admin | Admin should be able to view customer and restaurant Order History |
| UC-17 | Admin | Admin should be able to register a restaurant. |
| UC-18 | Admin | Admin Should be able to manage restaurant food items and categories. |

## Other References

* <https://docs.anaconda.com/>
* <https://docs.djangoproject.com/en/3.1/>
* <https://docs.python.org/3/>
* <https://www.w3schools.com/>
* <https://code.visualstudio.com/docs>

# Appendix

## Glossary of terms

|  |  |
| --- | --- |
| Course Catalog | Repository of all training objects held within the system including online courses, quick courses, events, libraries, and curricula. |
| USER (Customer, Restaurant and Admin) | A User signifies 3 main actors in out system who are fundamental users of functionalities. |
| External Training | Training taken outside of the LMS catalog(s) |
| Facility | Physical space used for instructor‐led training session. |
| Session | Specified scheduled instances of an event |
| Accessibility | The degree to which your website is easy to use. |
| Algorithm | A specific set of steps for carrying out certain tasks. Algorithms are used extensively in computer programming to arrive at a solution for a specific problem. |
| Browse | The process by which a user accesses various web resource. |
| Case Sensitive | A situation where it is important if a letter or a set of letters are in upper or lower case. |
| Community | Communication between internet users that occurs through various mediums that allow for text-based communication such as instant message programs, chat rooms, and forums. |
| Database | A collection of data stored in a computer in such a way that a program or a webpage can easily find, select, and/or manipulate the desired data. Typically, databases are organized by fields, records and files. A field is one piece of data, a record is a collection of fields, and a file is a collection of records. |

## Pre-requisites

Our System’s pre-requisites are mentioned below:

* Django Framework
* MYSQL 5\*
* Local server Xammp
* Visual Studio Code
* Windows 10
* Python 3.7
* Jupyter Notebook