

What's Cooking

Meal Spot Software Requirements Specification For Online Restaurant System

Version 1.0

grade: 88

comments: quite some dev ideas are given, the diagrams including use-case and E/R are well put (88 contains bonus for these parts). However, the major flaw of this report is: you didn't provide adequate details for each and every oval in your use-case diagram, which is the gist of this report (your grade is far lower than 88 if just focused on this flaw). Hope your team can be more focused next time.

| | |
|-------------------------------------|---------------|
| Meal Spot | Version: 1.0 |
| Software Requirements Specification | Date: 11/4/19 |
| What's-Cooking-Project-Report | |

Revision History

| Date | Version | Description | Author |
|-------------|----------------|--|--|
| 11/04/19 | 1.0 | Starting design for our application. Use-Case and ER diagrams | Oren Ben-Meir, Maui Arcuri, Florence Fong, Warin Wohab |

| | |
|-------------------------------------|---------------|
| Meal Spot | Version: 1.0 |
| Software Requirements Specification | Date: 11/4/19 |
| What's-Cooking-Project-Report | |

Table of Contents

| | |
|--|----|
| 1. Introduction | 4 |
| 1.1 Purpose | 4 |
| 1.2 Scope | 4 |
| 1.3 Definitions, Acronyms, Abbreviations | 4 |
| 1.4 References | 5 |
| 1.5 Overview | 5 |
| 2. Overall Description | 6 |
| 2.1 Use-Case Model Survey | 6 |
| 2.2 Assumptions and Dependencies | 7 |
| 3. Specific Requirements | 7 |
| 3.1 Use-Case Reports | 7 |
| 3.2 Supplementary Requirements | 12 |
| 4. Supporting Information | 13 |
| 4.1 Index | 13 |
| 4.2 Appendix | 13 |

| | |
|-------------------------------------|---------------|
| Meal Spot | Version: 1.0 |
| Software Requirements Specification | Date: 11/4/19 |
| What's-Cooking-Project-Report | |

Software Requirements Specification

1 Introduction

The purpose of this document is to address and organize all components that will define the Food Delivery system. Meal Spot is a food ordering from restaurants system that will handle users based on their type. The system will be designed using the Django framework. Tools and languages that will be used for the development of this system are defined in this document. The assumptions and dependencies of this application, use-case and ER data models, and the specific requirements for our application are discussed in further detail.

1.1 Purpose

Meal Spot provides an easy integrated interface for ordering from restaurants and handling the restaurants and deliveries, depending on the type of user. Customers can order and review food and delivery from any existing restaurant on the site of their choice. Existing restaurant managers can interact with their employees (employment, salary, select delivery) and their customers (complaints). Employees (cook, salesperson) can manage their own resources (food item, supply). Delivery people can bid on deliveries and review customers.

1.2 Scope

Meal Spot is a web based application developed using the Django framework that allows different types of users to sign in with their own personalized interface by rendering different pages on a browser. Each user has their own unique actions they can perform.

1.3 Definitions, Acronyms, Abbreviations

Front-end: Refers to the part of the program that the user interacts with directly, i.e the user interface.

Back-end: Refers to the part of the program that the user doesn't directly interacts with.

Model: The data(base) structure of an application and its associated logic and constraints.

View: The user interface of an application

Controller: The section of the application that receives input from the user and interacts with the data(base) of the program (i.e. the model) to then have the output be sent back to the user interface (i.e. the view). The controller controls how the view interacts with the model.

Model-View-Controller (i.e. MVC): A software design pattern that separates the software into three main components, model, view, and controller.

| | |
|-------------------------------------|---------------|
| Meal Spot | Version: 1.0 |
| Software Requirements Specification | Date: 11/4/19 |
| What's-Cooking-Project-Report | |

Model-View-Template (i.e. MVT): A software design pattern that is similar to MVC used for web-design. It redefines the view as the template and the controller as view. The view (or controller in MVC terms) has the role of both interacting of the database and then combining the results of that output with a template (specifically an HTML template) to render the user as a web-page. The main difference between MVT and MVC is that the view(MVT)/controller(MVC) must render an HTML page in MVT where in MVC it can output any result.

API (i.e. Application Programming Interface): A set of routines, protocols, and tools for building software applications.

HTML: A markup language used to structure a web-page.

CSS: A language used to style a web page but can be used to style the user-interface of any other type of application.

Python: A general purpose interpreted high level programming language

Javascript: A high level interpreted scripting language primarily used to program web pages and to make them interactive. With the introduction of Node.js, its capabilities and functionalities now extend way beyond front-end web programming.

Django: A python web framework used to create web applications. It is mainly meant to follow an MVT pattern though it can be reconfigured to follow an MVC pattern.

Relational Database: A type of database that organize data into tables with rows and columns. It allows data to reference other data in the database and uses special operations to read data by combining tables and adding certain constraints.

Relational Database Management System (i.e. a RDBMS): a software system that enables users to create, access, and control a relational database.

SQL: A computer language used to interact with a RDBMS

Data Model: A description of how data will be organized in a database.

Agile: A method used to develop software where users performed tasks to be completed in a given period of time and once done, new tasks are added. It is characterized by many meetings by members of a software development team to discuss their own progress and the development of the software as a whole.

Git: a software version control system that tracks changes in source code during development. It uses a data structure to log information about these changes called a repository.

GitHub: A website that stores a git repository online and is used or project management.

1.4 References

Django: <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/introduction>

MVC, and MVT: <https://over.io.com/django-1-11/mvc-pattern-and-django/>

1.5 Overview

be careful if no one in your team knew well about Django, many team fell to it eventually.

| | |
|-------------------------------------|---------------|
| Meal Spot | Version: 1.0 |
| Software Requirements Specification | Date: 11/4/19 |
| What's-Cooking-Project-Report | |

In the rest of the document, we will have a description of the functionality of the application, its dependencies, the technologies used, and the use-case and data model for our application.

2 Overall Description

Meal Spot is an online restaurant system that caters to several different users including customers, restaurant managers, salespeople, chefs, and delivery people. Customers may order food. Once approved by a restaurant manager, their order is prepared. For delivery to the customer, delivery people bid on the order for delivery price. There are also many other features allowing chefs, managers, and salespeople to manage the restaurant.

2.1 Use-Case Model Survey

This is the integrated use-case overview of the app. It combines all user types of customers and restaurant staff, and their corresponding functions. Section 3 splits apart the users into subsets to provide easier understanding.



over-kill, but fine.

| | |
|-------------------------------------|---------------|
| Meal Spot | Version: 1.0 |
| Software Requirements Specification | Date: 11/4/19 |
| What's-Cooking-Project-Report | |

2.2 Assumptions and Dependencies

We will assume that we are using the latest versions of both Django and Python and both are fully functional. We will also assume that for any web-api used, we will also have an API key for said API if it's required. We will also assume our app will run on a python virtual environment.

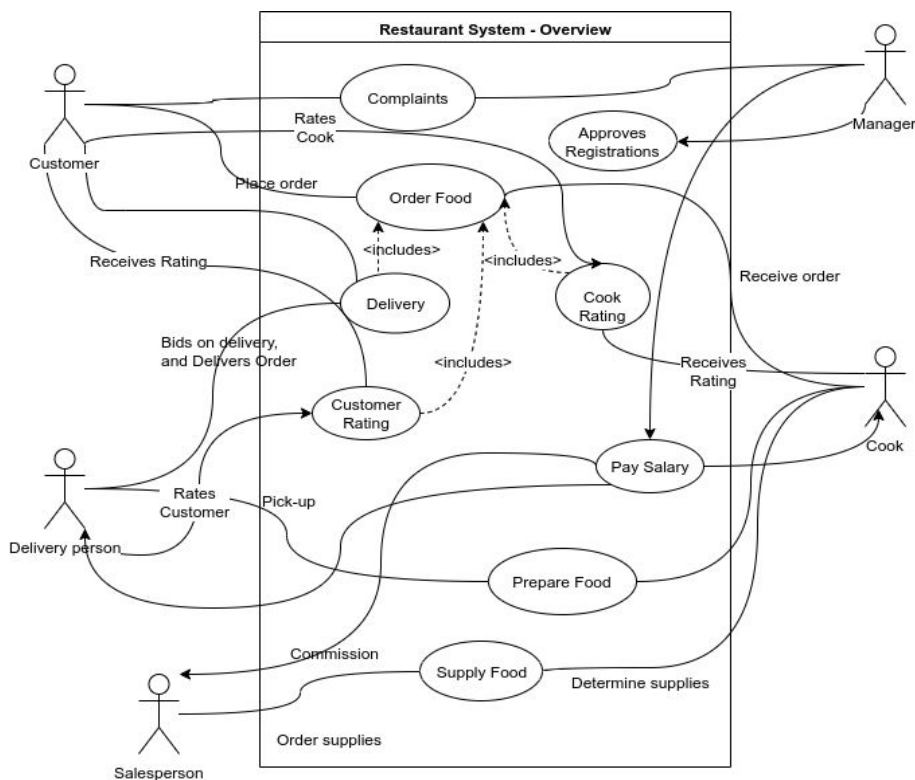
3 Specific Requirements

Our application will use the Django framework using python3.7, SQLite for development and PostgreSQL for deployment. We will use HTML templates, CSS, bootstrap, and Javascript for the front-end and python for the back-end. We also will use a google maps API. If possible we may also use scikit-learn for machine learning if we want to extend the functionality of our application.

3.1 Use-Case Reports

We divide up the integrated use-case diagram into 3 subsets: (a) Overview, (b) Employee-Manager, (c) Customer-Manager

a. Overview



| | |
|-------------------------------------|---------------|
| Meal Spot | Version: 1.0 |
| Software Requirements Specification | Date: 11/4/19 |
| What's-Cooking-Project-Report | |

Order Process:

The above use-case overview diagram generalizes basic functionalities of each user of the system. The ordering process of the system will look like this:

1. The Customer logs in and places an order with a restaurant
2. The Chef receives the order and prepares the food
3. Once the order is finished, the manager initializes a bidding procedure for the delivery people to bid on
4. The Delivery person who bids the lowest receives approval to deliver food and delivers the food to the customer
5. The delivery person may rate the customer
6. The customer may rate the chef
7. The chef may rate salespeople based on supplies ordered

In addition to interactions between the customers and the restaurant, within the establishment itself the manager and employees have their own interactions. The Employee-Manager diagram below shows just the use-case for the staff of the restaurant.

Manager:

- Approves registration / hires employees
- Starts bidding procedure for delivery people
- Modifies the status of staff:
 - If a delivery person who receives average rating less than 2 in the last three deliveries, they will receive a warning. The warning can be erased by the manager
 - Pays staff and decides commission for salespeople
- Views customer record and approves visitors who register
- Handles complaints

Salesperson:

- Decides supplies and manages supply ordering
- If a salesperson receives three 5 star rating in a row receives a 10% increase in commission. If the salesperson receives complaints from cooks three times, they receive a warning and a 10% decrease in commission.
- Each restaurant must have two salespeople

Cook:

| | |
|-------------------------------------|---------------|
| Meal Spot | Version: 1.0 |
| Software Requirements Specification | Date: 11/4/19 |
| What's-Cooking-Project-Report | |

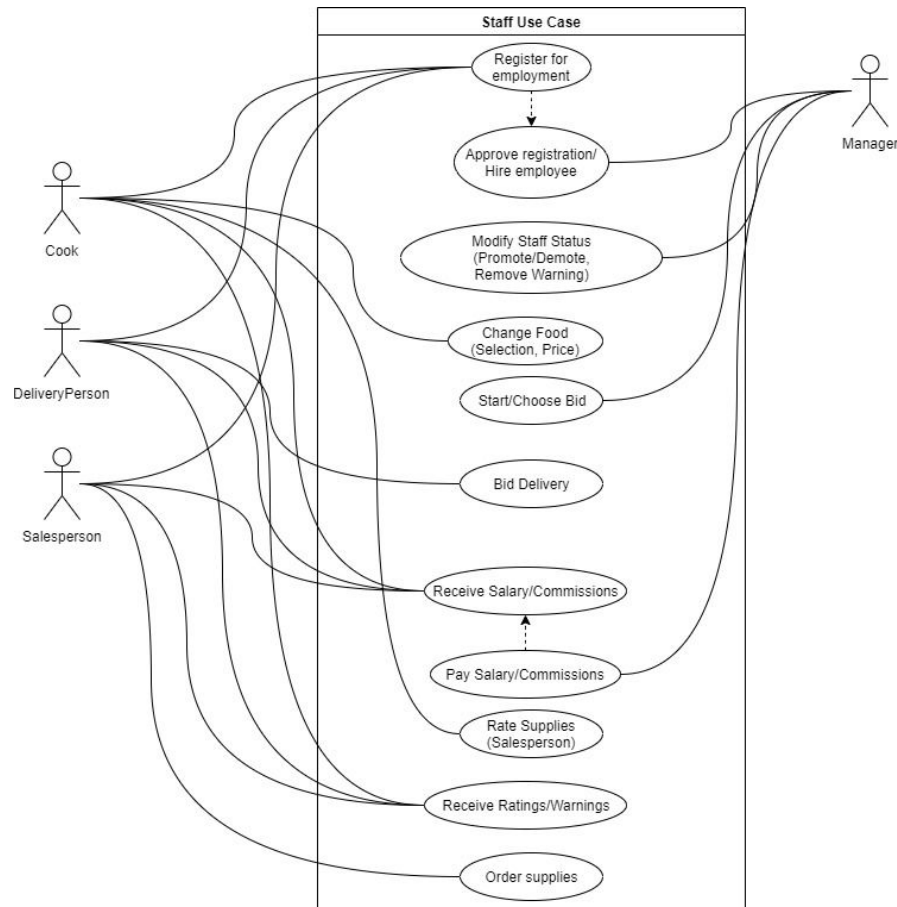
- Determines supply quantities
- Decides menu, food items, and prices of items
- Customer rates chef based on food from 1 - 5. Any rating less than three is a complaint with an accompanying sentence why.
- A food item with an average rating of less than 2 for the last three orders will be dropped
- Cook whose food item was dropped twice will be warned. A cook warned more than three times will be fired.
- Each restaurant must have at least two cooks

Delivery Person:

- Bids on deliveries
- Decides on the routes to deliver food to the customer
- Evaluates customer after delivery, however, they can no longer rate the customer after knowing the customer rating
- If the delivery person receives less than a 2 for the last three deliveries, they will receive a warning which can be erased by a manager. If they have more than three warnings, they will be fired

| | |
|-------------------------------------|---------------|
| Meal Spot | Version: 1.0 |
| Software Requirements Specification | Date: 11/4/19 |
| What's-Cooking-Project-Report | |

b. Employee-Manager Diagram



Customers have different privileges depending on their conditions and statuses. Below is the use-case diagram specifically for different types of customer users and how the manager handles the customers.

Guest User:

- A user who is not logged in.
- History of orders is not kept

Authenticated User:

- A user who is logged in.
- History of orders is kept

Unregistered Customer/Visitor:

- Authenticated User who is not registered in the restaurant. They are given the option to register to the restaurant.

| | |
|-------------------------------------|---------------|
| Meal Spot | Version: 1.0 |
| Software Requirements Specification | Date: 11/4/19 |
| What's-Cooking-Project-Report | |

- Visitor receives standard price
- Visitor cannot rate food but can read ratings

Registered Customer:

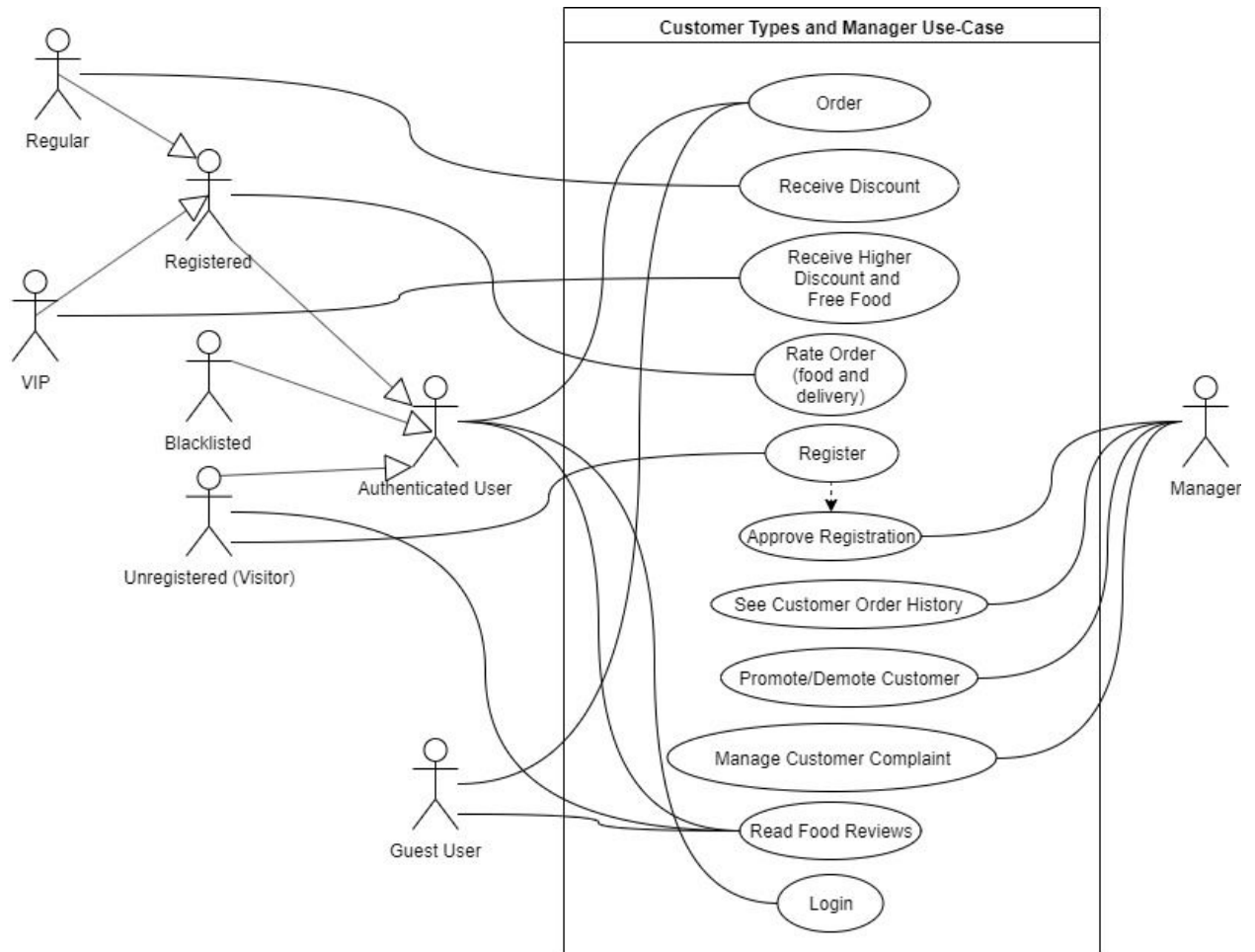
- A customer who has registered and has been approved by a manager
- There are 2 statuses of Registered Customers and each have their privileges.
 1. Regular
 - default customer status after approved registration
 - Regular receives discounted price
 - Regular can rate order (food, delivery)
 2. VIP:
 - VIP receives highest discounted price and free food items
 - VIP can rate order (food, delivery)
- Registered Customers have an average rating which affects their status. Average rating is determined by the average rating of the last four orders.
 - average rating of more than 4, the Regular is promoted to VIP
 - average rating less than 2, the Regular is demoted to Visitor. Essentially, registered customer loses registration status.
 - average rating of 1, the registered customer becomes blacklisted and can no longer register for the restaurant

Blacklisted Customer:

- Authenticated User who was previously registered and is blacklisted as a permanent Visitor (no future registration) as a consequence of poor customer etiquette.

| | |
|-------------------------------------|---------------|
| Meal Spot | Version: 1.0 |
| Software Requirements Specification | Date: 11/4/19 |
| What's-Cooking-Project-Report | |

c. Customer-Manager



3.2 Supplementary Requirements

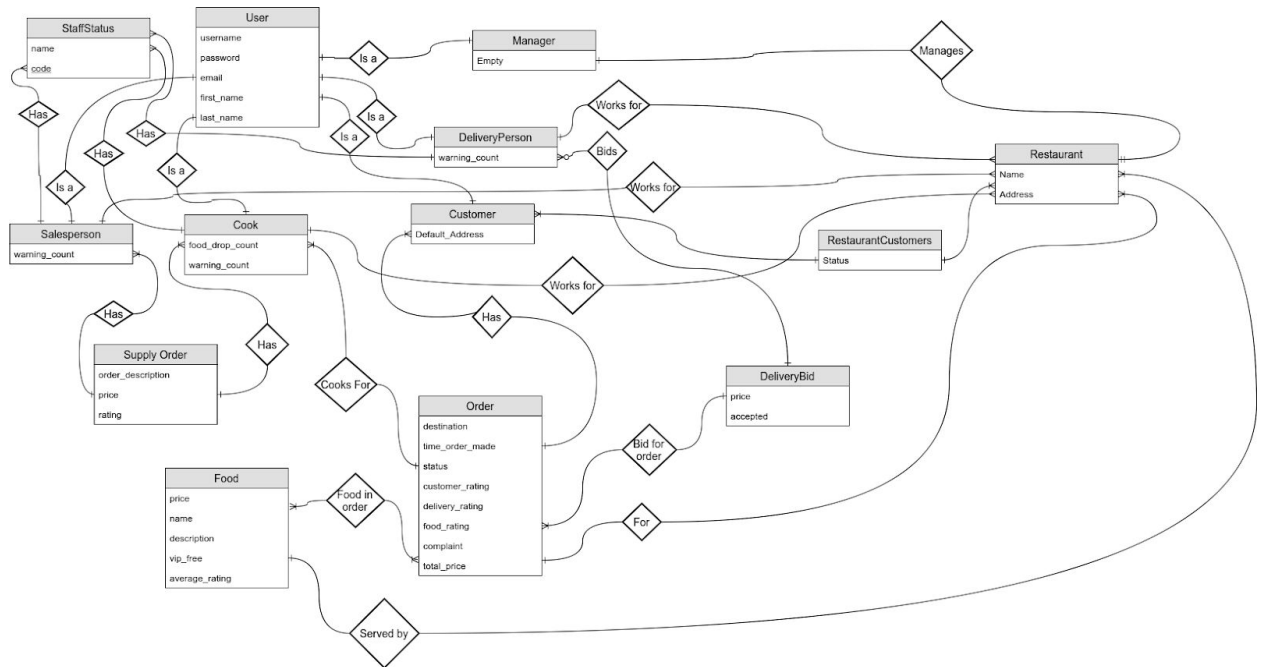
The restaurant management system also requires a few other features:

- The restaurant displays to customers the top three most relevant based on customer order history.
- The restaurant also displays the top three most popular food choices of visitors.

Below is an entity relation diagram which is a model of the data in the system. The database for Meal Spot will be set up based on the descriptions and relations in the diagram below.

| | |
|-------------------------------------|---------------|
| Meal Spot | Version: 1.0 |
| Software Requirements Specification | Date: 11/4/19 |
| What's-Cooking-Project-Report | |

ER Diagram



4 Supporting Information

4.1 Index

diagrams, 6, 7, 10, 12, 13

Use-case, 6, 7, 10, 11, 12

ER, 13

4.2 Appendix

Github: <https://github.com/OrenBen-Meir/Meal-Spot>