

Software Requirement Specification (SRS)

for

Park My Wheels (PMW)

1. Introduction

This document lays out a project plan for the development of the “Park my Wheels” open source repository system by **Priyanka Gyanchandani**.

The intended readers of this document are current and future developers working on “Park my Wheels” and the sponsors of the project. The plan will include, but is not restricted to, a summary of the system functionality, the scope of the project from the perspective of the project team (me and my mentor), scheduling and delivery estimates, project risks, and how those risks will be mitigated, the process by which I will develop the project, and metrics and measurements that will be recorded throughout the project.

2. Overview

With the increasing percentage of vehicle ownership in recent times, parking has become a conflicting and confusing situation for lot of people. Most often people cannot find suitable or adequate space to park their vehicles and end up encroaching the roads. The project made will give the following facilities:

1. It will provide the parking lot to people .
2. It will save time of people as they knew their slot of parking rather than finding out the empty parking slot if available.
3. Saves from overcharging by the render in a hurry situation.
4. Incidences of violence over occupancy will get reduced.
5. Reduction in clogging of roads

3. Approach for specified Problem:

In my project, suitable process model is Agile software development model.

In the Agile model, the requirements are decomposed into many small parts that can be incrementally developed. The Agile model adopts Iterative development. Each incremental part is developed over an iteration. Each iteration is intended to be small and easily manageable and that can be completed within a couple of weeks only. Long-term plans are not made.

The other software development life cycle spiral model is complex and too much dependent on risk analysis with the difficulty in time management. Another incremental process model requires good planning and design and well defined module interfaces with more time as many successive iterations/versions are implemented then the desired system is released which would be not a good option for me as a student who is allotted 4 months span of time to complete the project. This is where the agile software development comes to the rescue. It is specially designed to curate the needs of the rapidly changing environment by embracing the idea of incremental development and develop the actual final product by design, implementing, coding, unit testing, allows post changes to project, and gives final product in short span of time.

4. Overall Description:

4.1 Customers

Everyone. Anyone who owns the vehicle and wants to book the prior parking slot can use this application.

4.2 Functionalities

Smart parking management is a web-based system with the following functionalities:

- * Register new users.
- * Log in registered users.
- * Logged in users can book for a parking spot from different parking zones, get parking receipt with parking details and checkout from a parking spot.
- * Admins can create parking zones indicating the number of parking spots available.
- * Admin can view parking details of all users.

4.3 Platform

It will be launched as a Web-based application.

4.4 Development Responsibility

I, Priyanka Gyanchandani, would be developing the software and I am responsible for the creation of the Database and all the other related kinds of stuff.

5. Goals and Scopes

The proposed project solves the problem of parking in highly populated area where there is less management and traffic police officers are required to clear the clogging of roads created by static vehicles who park their cars randomly on the road. Following things are taken care in making this project :

- * Users should be able to register through their already existing accounts.
- * They should book the parking slot prior and should book the another parking slot when checked out of the previous one.

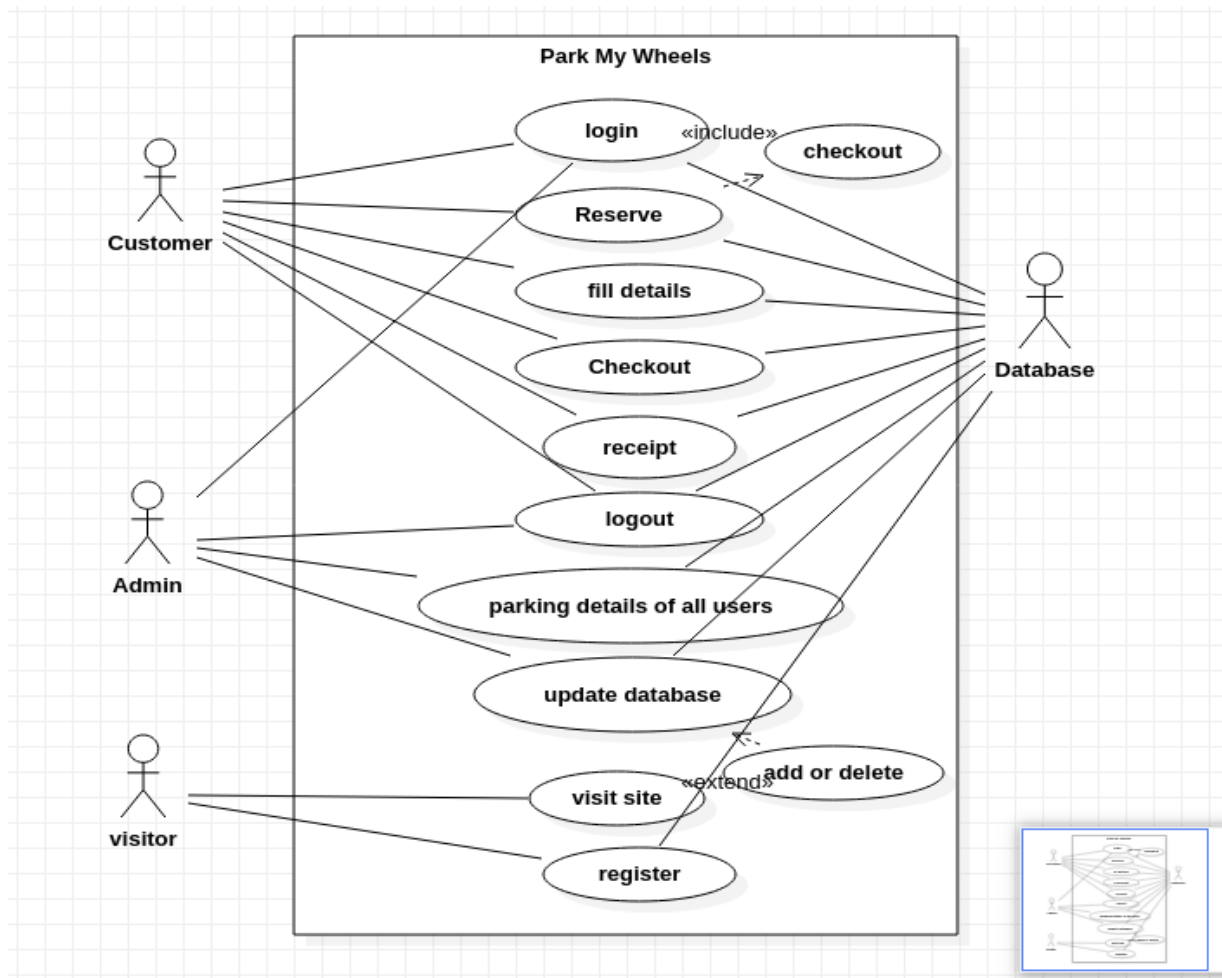
6. Deliverables

I'll deliver the following during the course of development:

- Feature specification
- Product design
- Test plan
- Development document
- Source code

7. Usecase diagram:

System should support this usecase:



8. Software tools:

Following would be the languages I would use to develop my application within the stipulated time period:

- * Python 3.9

- Programming Language

- * Django 3.1.6

- Web Framework

- * My SQL

- Backend

- pipenv (not mandatory but highly recommended)

- * VS Code

- Text Editor

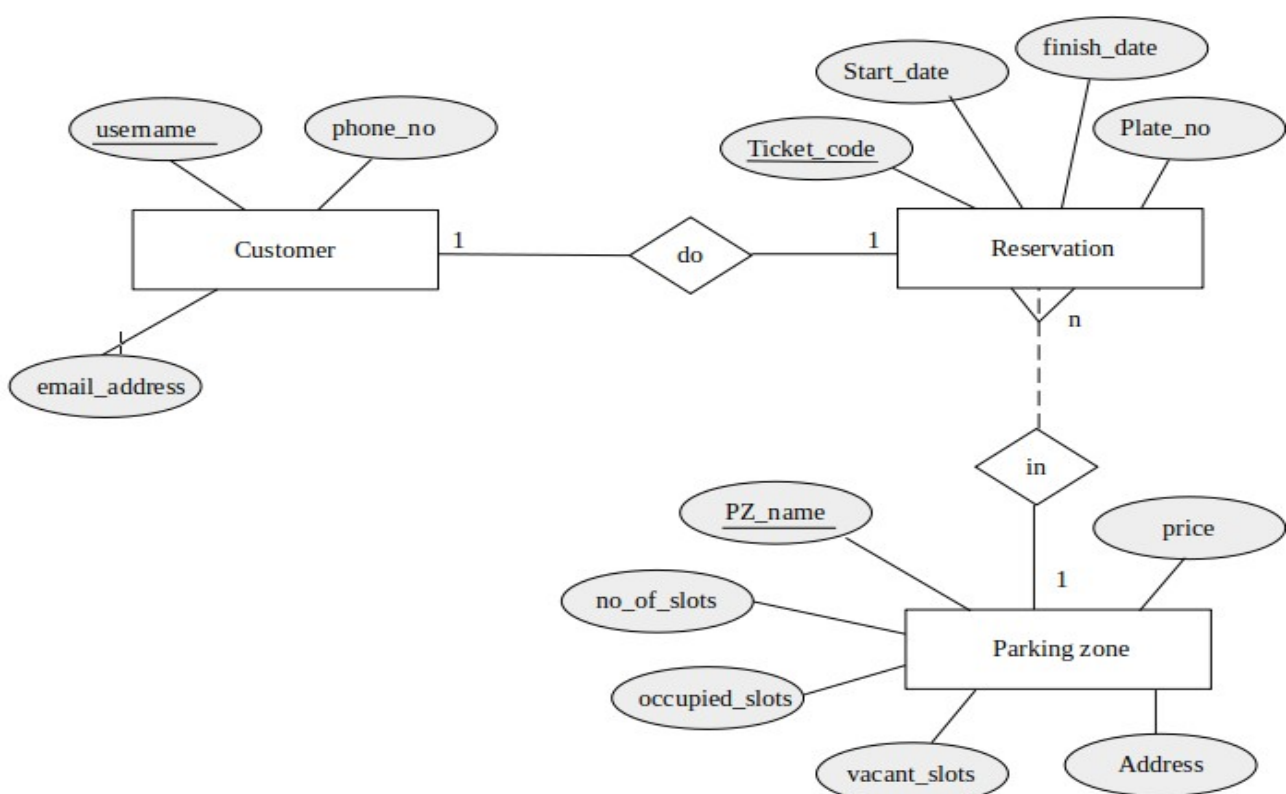
9. Non-Functional Requirements:

Following Non-Functional Requirements will be there in the insurance to the internet:

- (i) Secure access to consumer's confidential data.
- (ii) 24X7 availability.
- (iii) Better component design to get better performance at peak time.
- (iv) Flexible service based architecture will be highly desirable for future extension. Non-Functional Requirements define system properties and constraints. Various other Non-Functional Requirements are:

- ✓ Security
- ✓ Resource utilization
- ✓ Reliability
- ✓ Maintainability
- ✓ Portability
- ✓ Reusability

10. ER DIAGRAM:



11. Scheduling and Estimates:

Task	Start Date	Days to complete
Requirement analysis	07-08-2021	7
Research about the project	14-08-2021	7
study of django	21-08-2021	15
start with demo project	05-09-2021	25
revisited requirements	30-09-2021	4
Start with Django	04-10-2021	4
creation of model: signin	08-10-2021	2
adding bootstrap to project	10-10-2021	7
html part editing	17-10-2021	4
forms and validation	20-10-2021	3
testing the model – customer	24-10-2021	4
adding the model-reservation	28-10-2021	3
adding checkout functionality	31-10-2021	5
testing the whole project	05-11-2021	6
documentation	11-11-2021	7

Appendix A: Glossary

PMF refers to the name of the project that is., PARK MY WHEELS.
