
Software Requirements Specification

for

Work Order Management System

Version 1.0

Prepared by

Name: **Deepak Sharma**

deepakpvt26@gmail.com

Contact : +91-93199 79805

Instructor: Mr. Pranjul Sir

Course: Django Batch – 7:00 PM

Date: Jan, 18 2023

Contents

1 INTRODUCTION	3
1.1 PURPOSE	3
1.2 PRODUCT SCOPE	3
1.3 INTENDED AUDIENCE AND DOCUMENT OVERVIEW	3
1.4 DEFINITIONS, ACRONYMS AND ABBREVIATIONS	4
1.5 DOCUMENT CONVENTIONS	4
1.6 REFERENCES AND ACKNOWLEDGMENTS	4
2 Product Functionality.....	5
2.1 User Side	5
2.2 PRODUCT SCOPE	6
3 User Case Diagram.....	8
4 ER Diagram.....	9

1 Introduction

Work Order Management System is a platform that will allow Client(users) to directly put their orders regarding the products supply, service requirements or any upcoming project to company database using the web browser by themselves or they can simply upload their written work orders on behalf of their company.

They can have a proper follow up with company about the work orders that would be completed with proper circumstances or anything more required.

For the company's purpose, this will provide the big role to just don't wasting time on paper works or maintaining their data by entering them into the excel or any other software they use, it is already sorted on the web browser that which work order has been completed or not and any other details linked to the order also.

1.1 Purpose

The software requirements specification for Work Order Management System is intended to provide a detailed explanation of the features and design specifications for this application. Additionally, it will serve as a tool to communicate the design plans with users so that both parties(Client and Staff Members) may fully understand the specifications, requirements, and design of the Work Order Management System project.

1.2 Product Scope

Work Order Management System is a web application that is designed to allow users to easily put their orders to the company database from which they are ordering and also for company staff that reduces the paper work and any other work on maintaining the large data of work orders and other details linked directly to the work order like completion date, to check that products is available or not, payment to that completed work order is completed or not.

It just covers all the human efforts in an easy way and in scalable form. Most of important that the changes in work order can be done on website that reduces the wrong delivery of required work and it goes on real time at both side that if any changes is made then it is important to be get updated automatically in the database because on working the big numbers of work orders there are common challenges arises due to it.

1.3 Intended Audience and Document Overview

This document is intended for faculty and teaching assistants of our team, and anyone else interested in learning more about the Work Order Management System application.

The remainder of this document provides the general product description and a technical outline for the requirements of this system. Section two will give a high-level description of the project functionality and implementation details. There will be given a more detailed description of the specific requirements for different components of the application, which include various interfaces and functional requirements.

1.4 Definitions, Acronyms, and Abbreviations

- Application Programming Interface (API): a protocol used as an interface so that software components can communicate with each other
- Django: a high-level Python Web framework
- SQLite: inbuilt database of Django.
- Python: high-level programming language

1.5 Document Conventions

This document uses Times New Roman font throughout the entire document. Major header lines use bold, size 24 font. Subsections use bold, size 18 font. Smaller subsections use bold, size 14 font. Any part of this document that is not a header of some kind uses size 11 font. The one-inch margins are maintained throughout this document as well.

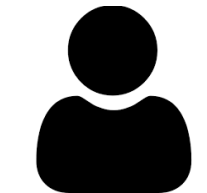
1.6 References and Acknowledgements

- Django:
<https://www.djangoproject.com/>
- SQLite:
Inbuilt in Django
- Python:
<https://www.python.org/>

2 Overall Description

2.1 Product Functionality

- The user first will have signup with details he/she required.



Client(User)

- Name
- Company Name
- Email
- Address
- Contact
- Password

- The admin would also have to sig up with their details.



shutterstock.com - 393536320

Admin(Company Staff)

- Name
- Company Email
- Contact
- Password
- Emp_Code

- On Login User and Admin will be directed to their interface respectively.

- Email
- Password

At User (Client Side)

- They got to see the status of their work order like at **home** page.

- Completed.
- On Going.
- Not Started yet. (So, they can do simply follow ups)
- User can take action on extending the dates of work as per their requirements.
- Make Payment** (also can in advance)
- Edit.

- **Profile Page**, there will be a page which contains some info about the client and additionally it has a **To Pay** Variable assigned to each client at 0 if there is any if he paid in advance then the amount is in positive sign other wise negative sign will imply the payment remaining to pay.
- **Put Your Orders** page, On while placing the Order or uploading the written work order the details required.

- Work Order Number
- Work Order Date
- Billing Address
- Shipping Address
- Concern Person
- Contact
- Till date completion.
- Payment Terms (advance, 50-50, etc)

- **About Us** page, the users can easily see the details of company and other work history records and completion award.
- **Contact** page, will be containing records of contact details of company side staffs and users for any further queries.

At Admin (Company staff Side)

- At **Home** Page, the staff person got to see all the work orders from all the clients.

- List of all Work Orders displaying their company name and Work Order Number with status remarks and **Update Button**.
- They are all sorted in that way like completed marked status would be listed in the last of the list.

- After clicking that **update** button the admin goes to the page **Make Bills**, the staff person got to see all the work orders from all the clients.

- Every work order completed will be got an assigned Bill No. mentioning their item details also.
- **Some items could be marked not considered only after approval with the client with the client.**
- There may be multiple bills of single order on dependent to the work completion states.
- Some of created bills can be canceled and admin get to start making another bill.

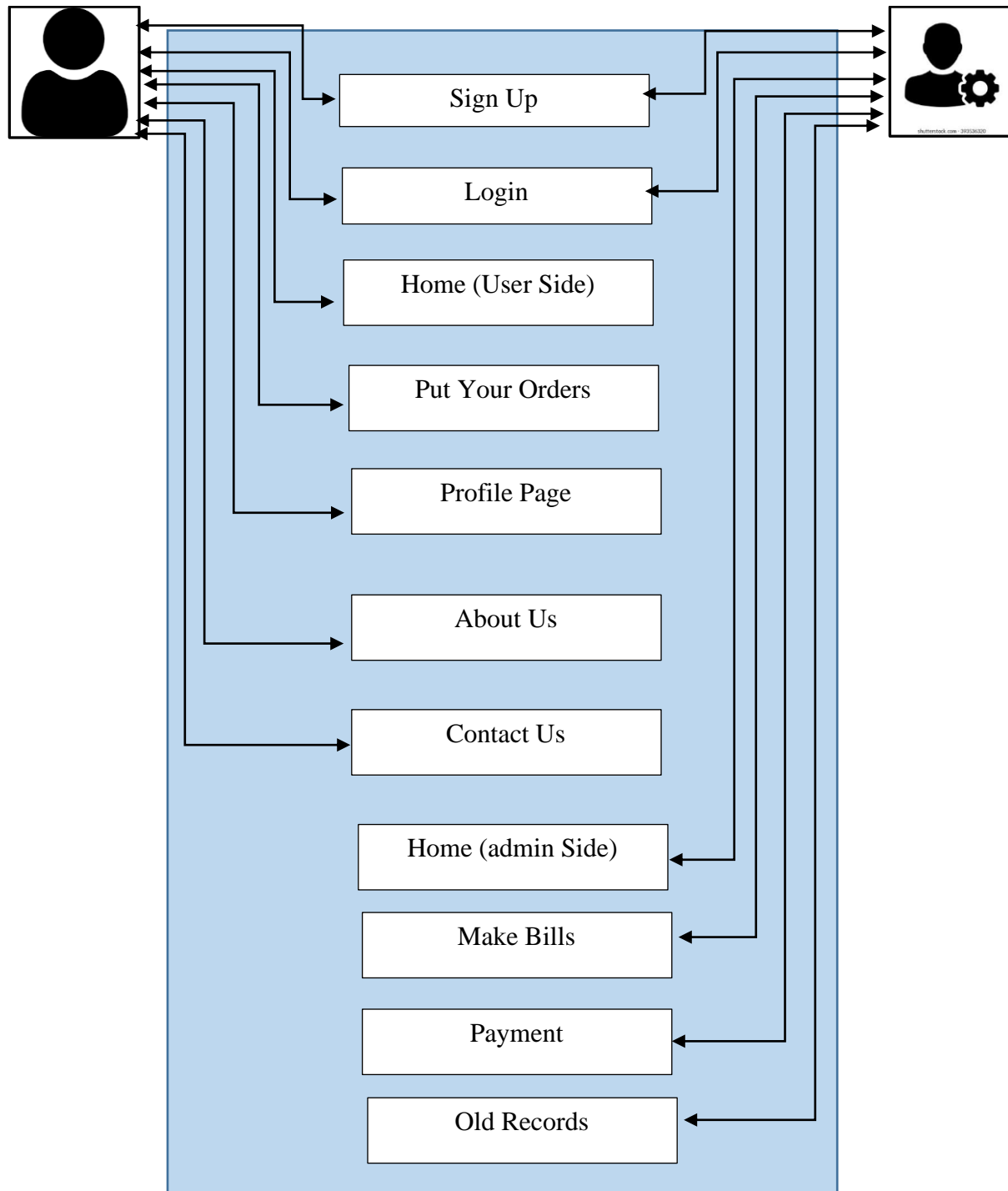
- **Payment** Page, this section will show the Payment received or pending against the bills you have created sorted by their company name + all bills of single Work Order get grouped together.

- Admin can update that status of Bills that it is paid, Not paid or partially paid.
- If all bills of completed work order paid then after 30 days **both work order and bills get to saved in another database and get removed from home page and payment section.**
- If bills are pending then admin can send reminder to the clients under the payment terms mentioned in Work Order.

- **Old Records** Page, in this page the admin can just go thorough old records if any data required by the company staff at any moment.

- Admin can search by Company name, Work Orders date, Work Order Number, Bill Number, Serial Number of Supplied of the products or any other field.
- Can download the records in form of excel.
- This database should be big enough to hold the data of 5-6 yrs.

3 User Case Diagram



4 ER Diagram

