



**EAST WEST UNIVERSITY**

## **CSE347: System Design and Analysis**

### **Sec-01**

# **Online Platform Providing Qualified Technicians On Demand**

## **Software Requirement Specification**

**Submitted to-**

Md. Mohsin Uddin

Senior Lecturer

Department of Computer Science & Engineering,

East West University

**Submitted by:**

<b>Student Name</b>	<b>Student ID</b>
Sakib Haider	2020-1-60-104
Gazi Fayaz Ahmed	2020-1-60-099
Mahidy Hasan Plabun	2018-2-60-082

**Submission Date: 9th September, 2022**

# MISTRI

## **Introduction:**

The software requirements specifications, SRS will provide a description of our web application MISTRI and It will provide a complete understanding of what is to be expected of this website. SRS will give users the specifications on what the system should do. The SRS of MISTRI will provide the managers and owners the gathered data on the wanted system. The clear understanding of MISTRI and its functionality will allow for the correct website to be developed for the end user. Also the SRS manual will help developers design the system and afterwards enhance the work flow of developing and testing. The owners and the manager will read the SRS and evaluate if the manual is ready to be used for the next phase of SDLC.

## **Definition Problem:**

The goal of MISTRI is to help consumers find the best technicians for their household maintenance. Whenever there is a need for maintenance in the house, it is a stressful job to find someone for the work. Even if we can find someone, most of the time they are not qualified enough. What if there is a single system that can find you the best person for the work without running around and calling strangers? That is where MISTRI comes in. After completing registration, the customers can find qualified technicians according to their needs. All the customer needs to do is to click “**Hire**”. Admin will select a perfect Mistri for the customer. Admin will take care of everything for the customer. The customer can see their own history of works that they have requested to the system.

## **Motivation:**

The intended audience of this document would be the owner and service providing technicians, customers and the project team with the objective to refer to and analyze information. The SRS document can be used in any case regarding the requirements of the project and solutions that have been taken. The document would finally provide a clear idea about the system that is built. The project is aimed to build a fully functional system in order to achieve the efficiency of the system providing an easy process for general people to hire technicians. The overall mission of system development also includes making the system user friendly and secure.

## **Challenges:**

We may find many loopholes in the system. In payment functionalities, a lot of loopholes may appear which the users may exploit. We have to find those loopholes and try to close them for fair use of the system.

**MISTRI** is a relatively unique concept so we lack a proper reference point for quality software. Thus it is a challenge to design **MISTRI** as a well optimized web application.

Another major challenge is to finish the code in time. the system has very sophisticated echo system to build and we are suspecting that we would not have enough time to finish the whole code according to our plan

## **Non-Functional Requirements:**

Non-functional requirements define the needs in terms of performance, logical database requirements, design constraints, standards compliance, reliability, availability, security, maintainability, portability, safety requirements, Efficiency, Testability, Robustness, Integrity. There are 3 main sections that we want to talk about. Those are-

### **Product Requirements:**

#### **Usability:**

The system should have a user-friendly UI. Users should have smooth experience with the Web application.

#### **Efficiency:**

The software's algorithmic performance should be adequate, and memory management should be intelligent. Because our system cannot afford to spend too much time processing data. It also requires a large amount of storage space to store data.

#### **Dependability:**

The system should be designed in such a way that clients can be confident in its security and authenticity.

#### **Security:**

The authentication and authorization system should be fully optimized by maximum security. We will use a cloud based system for database integration which will be secured and flexible to use and in that way it will be almost impossible for the hackers to integrate the system in an unethical way.

As an example, no user can use others personal information. they are allowed to see only the authorized part. One cannot change others personal information. The users will have full impedance to choose the work he wants to do.

### **Organizational Requirements:**

#### **Environmental:**

The system should be designed in such a way that it maximizes the use of the CPU and servers to perform its operations, thereby having a lower physical impact on the environment and saving energy.

#### **Operational:**

The system should be designed so that there is a hierarchy of users and different levels of database views. Because there are various types of users. The database and the system should not be accessible to all operators in the same way.

### **External Requirements**

#### **Compliance**

In order to maintain transparency, the system must abide by Bangladeshi legislation and the ICT acts.

#### **Ethical**

The system must be designed so that there is no loophole for accessing and obtaining raw data in order to use them for one's own gain. The system must practice all the ethical laws of the software industry.

If any user tries to violate any rules of behavior. He must be restricted from using MISTRI.

## **Legislative**

The system must be robust enough to safeguard preserved data while guaranteeing account security and safety. Additionally, when creating the system, user privacy policies should be adhered to.

## **Functional Requirements:**

Function requirements define the fundamental actions that a system must perform. The functional requirements of the system are mainly divided into five parts - Landing page, Registration Stage, Log in Stage, System for technicians and System for customers.

### **Landing Page**

- The home page for the system.
- Options to register and log in will be here.
- Most of the functionalities of the system can be viewed from this page.

### **Registration Stage**

- To register to the system, it will validate the information of the person trying to register.
- Only one account can be made with an email or a phone number.
- Only one account can be made with a phone number.

### **Log in Stage**

- To log in, the user would need to provide the correct email address.

### **System for Technicians:**

- Application system for new technicians.
- Must include their address, phone number, picture, email and certification (if any).
- Option to add payment method.
- Log in option for already registered technicians.
- Option to reset the password if forgotten.
- Option to change the Password.
- Option to update all the information.
- List of all completed work history.
- Shows the overall rating of the technician.
- Balance of account - where current balance will be stored.
- Option to withdraw balance through the selected payment method.
- Pending jobs.
- After completion of job payment will be automatically added to the account balance.
- Option to leave a review about the customers.

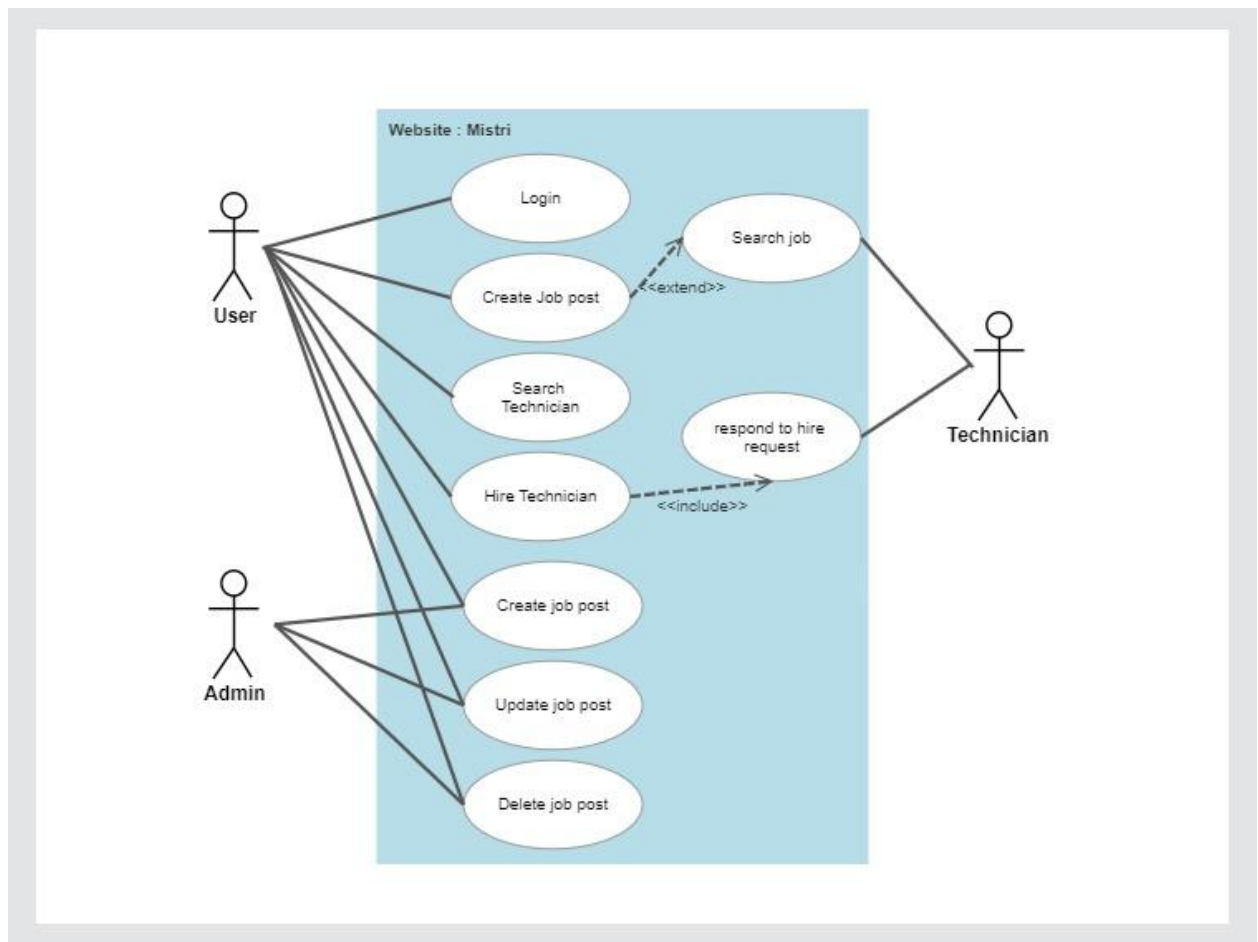
### **System for Customers:**

- Registration system for new customers
- Log in option for already registered customers.
- Option to reset the password if forgotten.
- Must include their address, phone number, email information.
- Option to add payment method.

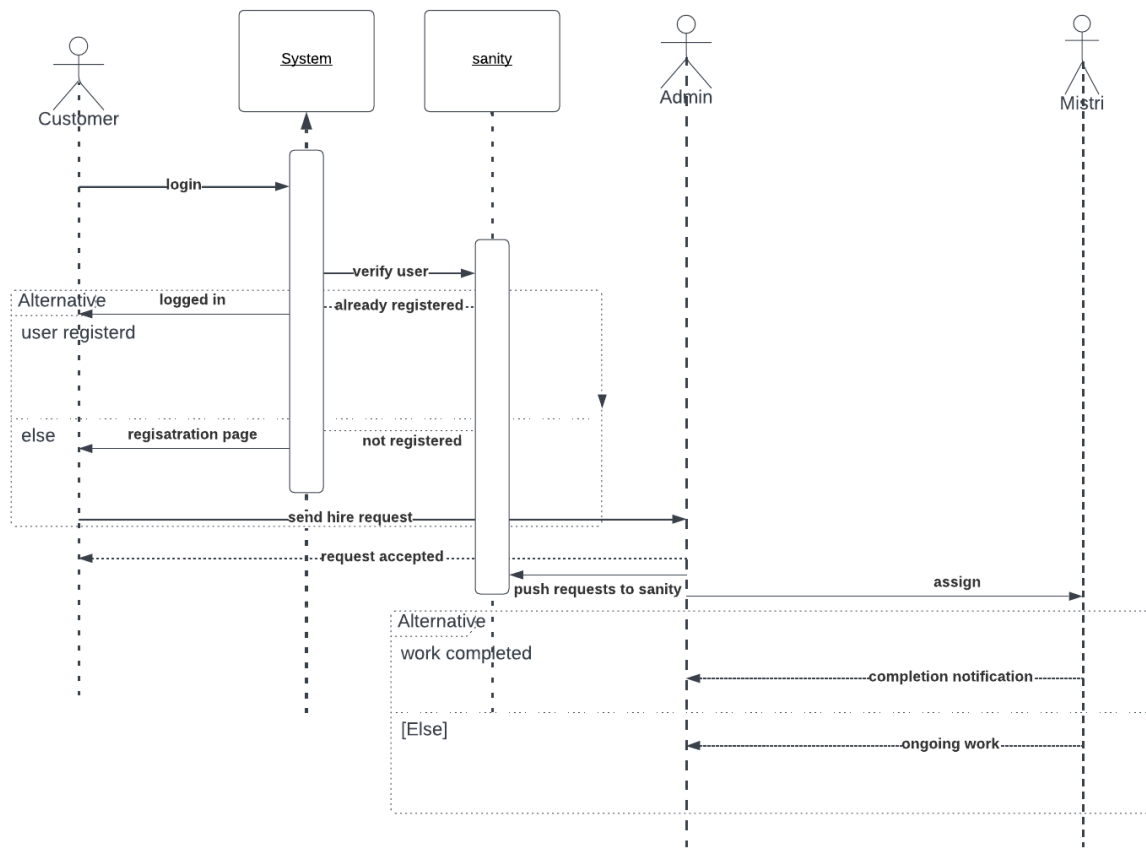


- Search option by desired technical service name.

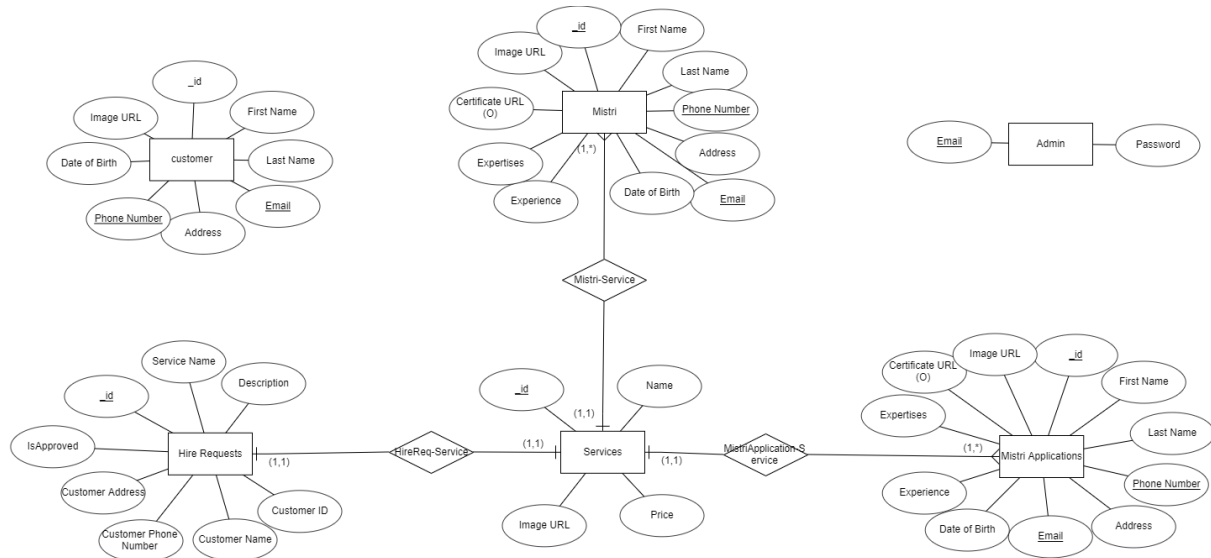
## Use Case Diagram



## Sequence Diagram:



## ER Diagram



## Technologies:

### Language

- For frontend we have used HTML, CSS, JavaScript and React.js
- For the backend we have used Next.js which is a library for React.js.

### Database

- For Database and authentication we are going to use Firebase and Sanity CMS

### Others

- Magic and Firebase Authentication
- For payment systems - Bkash, Rocket, Stripe API.
- Google map API.

- Socket.io for real-time chat application.