MyZindagi – An Online Healthcare Platform

# Software Design Document

Version 1.0

03/05/2018

|  |
| --- |
| Authors:  Hamza Farrukh BSSE 02153026  M. Ansab Majeed BSSE 02153188  Tanseer Javed BSSE 02153187  M. Sanan BSSE 02153051  Usama Ayub BSSE 02153028 |

Table of Contents

Contents

[Software Design Document i](#_Toc513658351)

[Table of Contents ii](#_Toc513658352)

[1. Introduction 3](#_Toc513658353)

[1.1 Purpose 3](#_Toc513658354)

[1.2 Scope 3](#_Toc513658355)

[1.3 Intended Audience 3](#_Toc513658356)

[2. Overview 4](#_Toc513658357)

[2.1 General Overview 4](#_Toc513658358)

[2.2 Assumptions/Constraints/Risks 4](#_Toc513658359)

[3. Design Considerations 5](#_Toc513658360)

[3.1 Goals and Guidelines 5](#_Toc513658361)

[4. System Architecture and Architecture Design 6](#_Toc513658362)

[4.1 General 6](#_Toc513658363)

[4.2 Hardware Architecture 6](#_Toc513658364)

[4.3 Software Architecture 6](#_Toc513658365)

[5. System Design 7](#_Toc513658366)

[5.1 System Architecture Diagram 7](#_Toc513658367)

[5.2 Sequence: 8](#_Toc513658368)

## Introduction

### Purpose

The SDD of MyZindagi documents the necessary information needed to effectively define architecture and system design for the application whose requirements and functionalities were outlined in the SRS. The design documents were made during the software development life cycle incrementally and iteratively, based on the system development methodology used for developing MyZindagi. This document has been written in accordance with the IEEE standards.

### Scope

This Software Design Document has been created for the Web service MyZindagi. In this document we will examine how MyZindagi aims to reduce common problems faced by patients looking for a doctor that fits their requirements, in their respective locations. The MyZindagi system is composed of three main components: a public web application where users can browse public doctor profiles and health organization profiles, a private web application which can be used by system administrators or managers to create or update profiles, appointments or medical records, and a database containing all the information of the patients, doctors and health organizations. The system features can be decomposed into two groups as well: core features, which are essential for the service to function, and extra features, which are meant to add additional functionality.

### Intended Audience

The audience for whom this document is intended include, but are not limited to, the project manager, project team, and development team. Some portions of this document, like the user interface, may be shared with the client/user, and stakeholders with whom the interface must be discussed for improving the design.

## Overview

### General Overview

MyZindagi is an online healthcare platform that seeks to act as the intermediary between a user and a doctor, create a comprehensive database of all doctors in Pakistan, take over certain responsibilities of a doctor’s personal assistant and facilitate potential patients by maintaining a medical history record and by helping them schedule appointments with doctor by the development of an online healthcare system that can be easily accessed by anyone with an internet connection. MyZindagi is a free, open source Web service and it may be used and studied without restriction.

### Assumptions/Constraints/Risks

#### Assumptions

The user of the service knows how to operate a computer, connect to the internet and navigate web pages.

The user understands general terms used in the system.

#### Constraints

Major constraint for the development of the system is making it easy-to-use for people of all ages, with a special focus on people between the ages of 40-60. Other than that, the system is being built using majorly HTML, CSS, PHP and Javascript due to the development teams limited experience with other technologies. Time is also a constraint, which may result in fewer features in the initial release (but the core functionality must remain unaffected.)

## Design Considerations

### Goals and Guidelines

Following are some basic goals and priorities for the system design that require more focus than others:

#### Usability:

The service should be usable by people of all ages, but there must be special attention to people of ages 40-60, people with limited technical knowledge and people whose ability to navigate interfaces may be impaired by visual or motor disabilities since the userbase must include a wide range of demographics including senior citizens, parents (especially mothers) and disabled persons.

#### Updatability:

The system must allow continuous updation on a large-scale since information may have to be renewed completely or partially, at various odd times because patients contact information, medical information, doctor’s contact information, timetables and appointment processes, and health organization’s contact information, operation timings and practicing faculty may change frequently. The server-side management application and the database must be built to facilitate such changes.

## System Architecture and Architecture Design

### General

The user web application will include a search engine for doctors and organizations, profiles for the doctors and organizations, and an appointment request system. The information included in all these sections will be input through the management web application.

The management web application will contain sections for adding, editing and removing information about the patients, doctors and organizations, and keeping track of website usage and its users. The information will be stored in the database.

The database will contain information on every patient, doctor and organization in raw form which must be queried and presented to the user through the management web application and the user web application.

### Hardware Architecture

The entire system will be stored on Amazon Web Services and the team shall interact with it through their management system.

### Software Architecture

The system will be designed using an MVC architecture (Model, View, Controller) where:

**Views:** Login/Register Page, Search Engine/Results Page, Patient Profile Page, Doctor Profile Page, Organization Profile Page, Appointment Request Page, Management Landing Page, Profile Edit Page, Web Traffic Info Page, User Info Page

**Model:** MongoDB database

**Controller:** Will define request-handling logic.

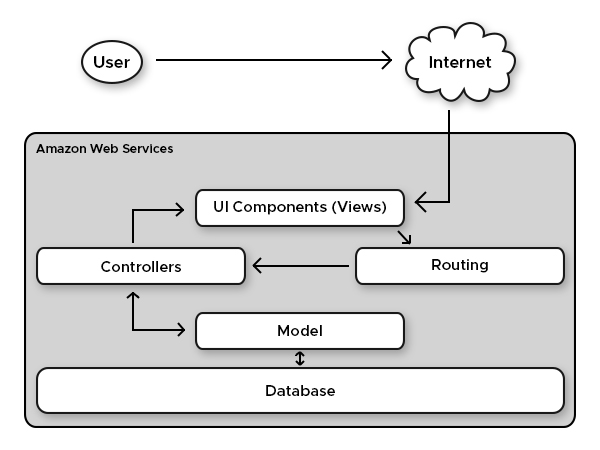
#### Security Software Architecture

Amazon Web Services attributes high priority to security; their security platform offers DDoS mitigation, data encryption, monitoring and logging, identity and access control, and penetration testing.

Other than that, the developers will need to follow all security guidelines in both front-end and back-end applications to ensure that there are no mistakes on our end.

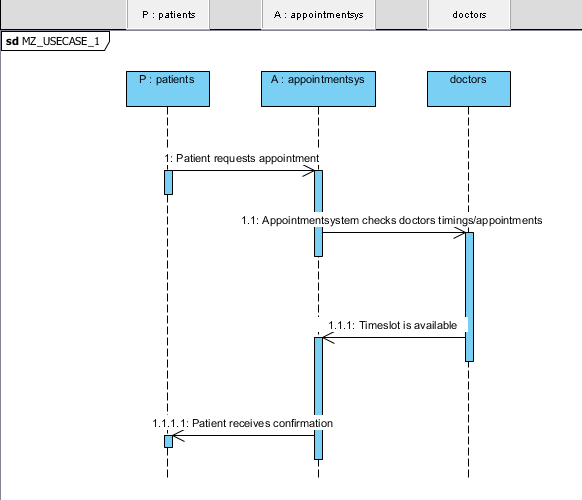
## System Design

### System Architecture Diagram

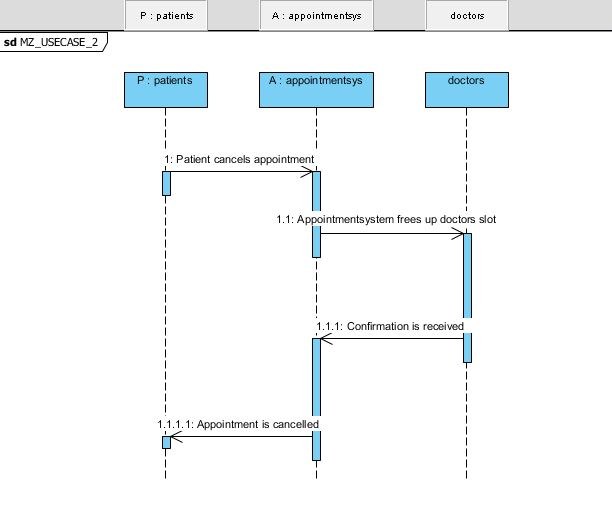


### Sequence:

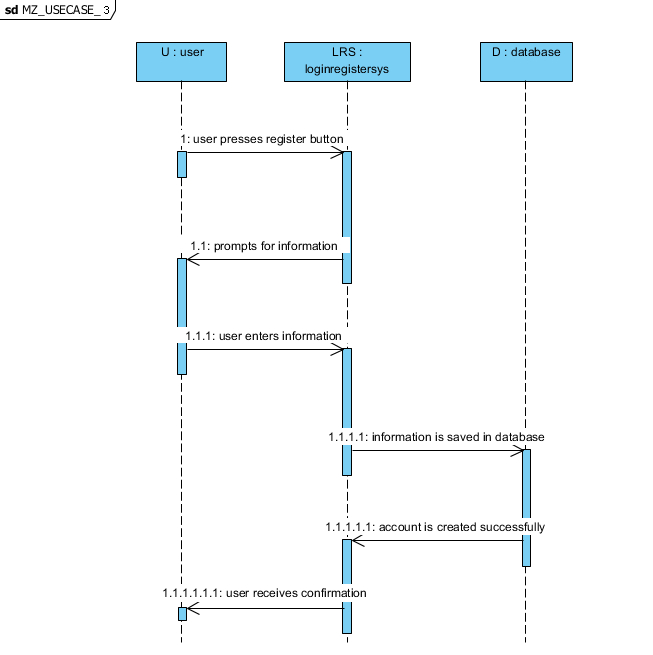
|  |  |
| --- | --- |
| **USE CASE #1** | **Request Appointment** |



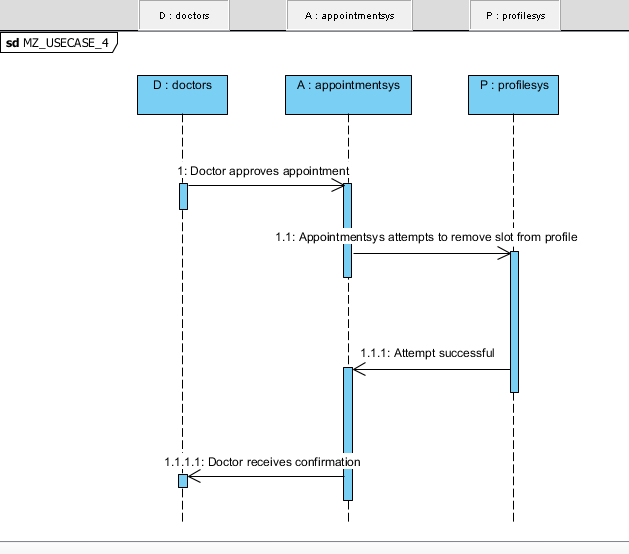
|  |  |
| --- | --- |
| **USE CASE #2** | **Cancel Appointment** |



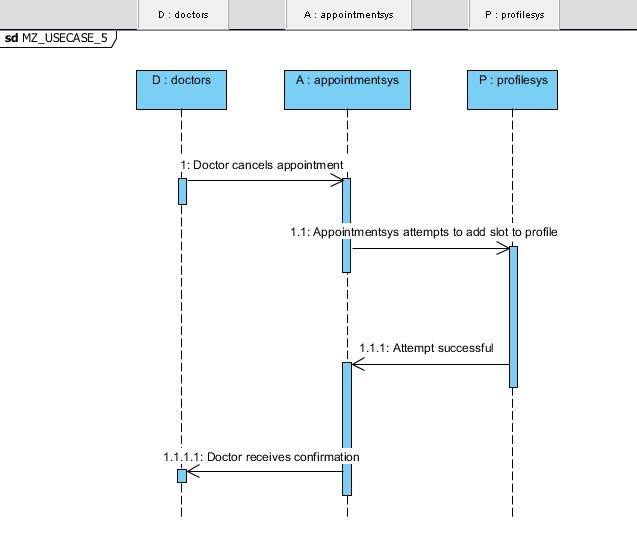
|  |  |
| --- | --- |
| **USE CASE #3** | **Login/Register** |



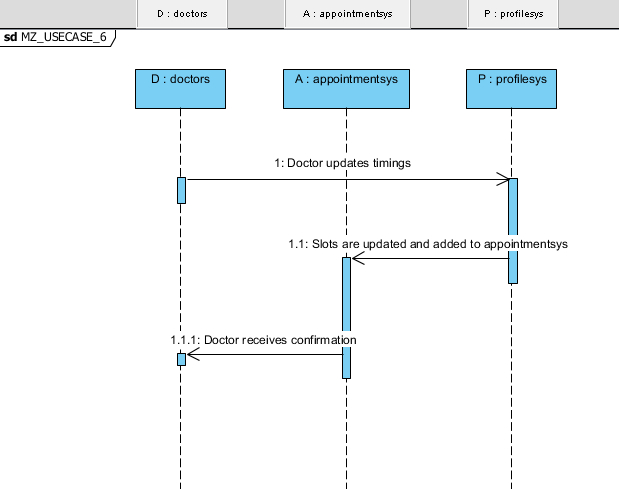
|  |  |
| --- | --- |
| **USE CASE #4** | **Appointment Approval** |



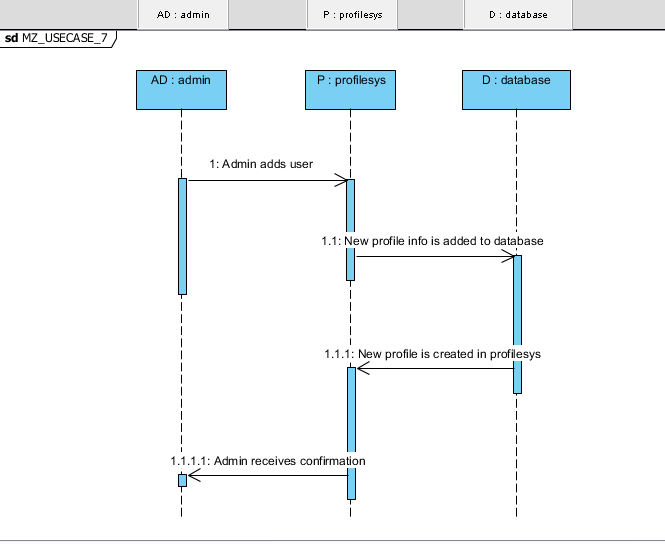
|  |  |
| --- | --- |
| **USE CASE #5** | **Appointment Cancellation (DOC)** |



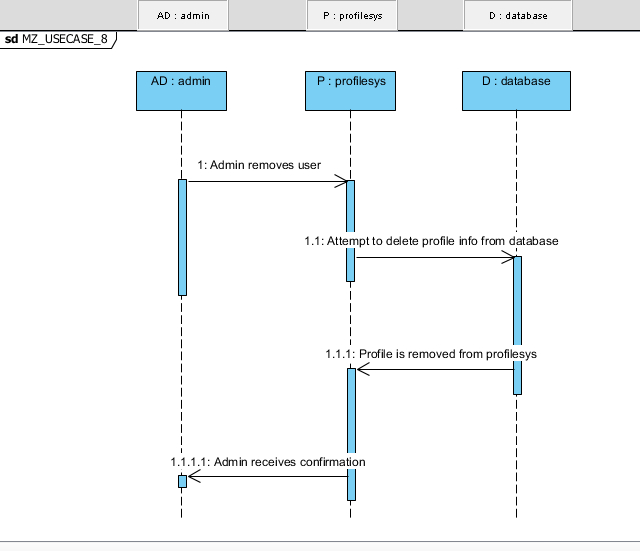
|  |  |
| --- | --- |
| **USE CASE #6** | **Set Timings (DOC)** |



|  |  |
| --- | --- |
| **USE CASE #7** | **Add User (ADMIN)** |



|  |  |
| --- | --- |
| **USE CASE #8** | **Remove User (ADMIN)** |



|  |  |
| --- | --- |
| **USE CASE #9** | **Set Timings (ADMIN)** |

