Software Requirements Specification

For

Online Doctor

(Robust.com)

Table of Contents

1.Introduction

1.1 Purpose

This document describes the software requirements and specification for an online doctor cum health care website namely Robust.com so as to serve as a guide on one hand and a software validation document for the prospective client on the other.

1.2 Document Conventions: Font : Times Now Roman 12

1.3 Intended Audience and Reading Suggestions

The document is intended for all the stakeholders customer and the developer (**designers, testers, maintainers**). The user is assumed to have basic knowledge of English ,computers and computer operations.

1.4 Definitions and Abbreviations

1.4.1 Definitions

- **Account**

A single account on the website which will be unique per user and one user is not allowed to have more than one account.User's account would be used to access all the features and services offered.

- **Admin**

Administrator.He is the sole authority to Modify the user accounts from the server end and is responsible for Modifying and Verifying User Accounts is asked.

- **User**

The holder of one account on the website. A user account can be maintained/accessed by anyone they shares their password with , the correspondence is not relevant to this problem.

- **Doctor**

The users with knowledge in medical field and has practice experience of at least 5 years and marks above certain qualifications marked up by the developers.

- **User Id**

A unique ID assigned to both; The Users and Doctors once they open an account on the database. Each User's ID starts with a 'u' and Doctor's ID starts with 'd'. Different formats helps the system and the developers to show up things and features differently. Each account is owned by a single user, but multiple users can be their if password is being shared,so the possibility of simultaneous use of the same account from different locations must be considered.

- **Stock**

Total collection of a given medicine in the warehouse which can be directly checked by the user it self.

- **Consultation**

A video/Audio/Textual interface set up between user and doctor where they can talk.All the data would be encrypted and both user,doctor and the system can get the conversation done between both only on request.

1.4.2 Abbreviations

Throughout this document the following abbreviations are used:

- ➢ k : is the maximum requests a user can make in a day from one account.
- ➢ m: is the maximum quantity a user can request medicine for.
- ➢ n : is the minimum cash to be paid for consultation
- ➢ t : is the total stock available in the warehouse of medicines.

1.5 Project Scope

The software supports a computerized health care network. The network enables users to diagnose simple problems by answering some simple questions and diagnose for other diseases by consulting the Experts being available online through a web interface. The Website identifies a user by his credentials , i.e.,User Id and password. It collects information about the user's health,age . gender and saves up them in the data base. The Admin is having different approach towards the services.The doctors are having a different interface with different features.The database is keeping all the records . The software must handle concurrent accesses to the same account correctly.

2. Overall Description

2.1 Product Perspective

The website network does not work independently. It works in parallel of both, user side and client side server and both the servers are connected to the database which is being updated at every instance whenever some changes are being observed.

Communication interface: The website communicate with the database via a communication network.

Software interface: The commands and updates sent via the communication network are specific to the target database. At present,single database would be there participating in this.

Hardware interface: The software will run on any basic computer which haves an internet connection and can even run on mobile phones.

User Interfaces

Customer: The user interface should be intuitive, such that 99.9% of all new users are able to complete their tasks without any assistance.

Admin: Admin is responsible for removing/adding users on requests. The doctors can only be approved once Admin views the supported documents uploaded by them and verifies it.

Maintainer: The maintainer is responsible for adding new features to the website and maintaining existing website. A maintainer should be possible to add a new feature to the website within minimum time and without taking down the website for longer duration .

2.2 Product Features

The website should work 24 hrs. The website identifies a user by a user id and password. It collects simple data about user and keep it in the database.User can diagnose himself by answering some simple questions.For other times, he can simply consult to a available doctor and can get real-time assistance online. The database required enough security measures to keep the server safe and secure.User credentials must be stored in encryption so that they can't be altered through some other agencies in any case.The software must handle concurrent accesses to the same account correctly.

2.3 Operating Environment

The hardware, software and technology used should have following specifications:

- Ability to validate the user
- Active Internet Connection
- Continuous power supply
- Ability to connect to user with the system
- Ability to take input from user
- Ability to validate user

2.4 Design and Implementation Constraints

- Login

Validate User:

- Validate for User's login details
- Validate that the User is entering correct details
- If details are wrong, prompt error message "WRONG CREDENTIALS"

Validate for Forgotten Id's:

- Validate if No. Of login attempts are more than
- If Password is forgotten, prompt error message, "Use Mobile number to reset"
- If User is forgotten, prompt error message, "Use Find my account section to Access your account back"

Validate for Disabled Card:

- Validate the disabled account
- If account is disabled, prompt error message, "Your Account has been temporarily disabled <No. Of Unsuccessful attempts exceeds the allowed limit>"

Validate Password:

- Validate that the password is not blank
- If Password is blank, prompt error message "Password field is Empty"
- Validate that the password entered matches the password on file
- If password does not match, prompt error message "Password is Incorrect"

Lock Account:

- If number of consecutive unsuccessful login exceeds five attempts, lock account temporarily
- Maintain Consecutive Unsuccessful Login Counter
- Increment Login Counter
- For every consecutive Login attempt, increment logic counter by 1.
- Reset login counter to 0 after login is successful.

2.6 Assumptions and Dependencies

- System has an up time of 100% and never goes down.
- Active and stable internet connection is there.
- User Shares his correct credentials and problems.
- Single Login from one Id at one time.

3.
3.1 Functional Requirements

The functional requirements are broadly written into a single section , i.e. requirements of the user and are validated accordingly.

3.1.1   Requirements of the Users

The requirements for the User are organized in the following way General requirements, requirements for authorization, requirements for a transaction/Consultation.

General

Functional requirement 1:

- **Description:** Initialize a user by signing him up.
- **Input:** User is assigned user id and password of his choice
- **Processing:** Storing the credentials in the database.
- **Output:** Credentials are set.User is created

Functional requirement 2:

- **Description:** If no doctor is available at the moment for consultation , the system should display Queue Number.

Functional requirement 3:

- **Description:** If the doctors aren't available at a particular moment An error message is displayed.
- **Input:** A doctor is Searched for consultation from 'COUNSULT NOW' option.
- **Processing:** The vacant doctors are searched at the particular instance and a list is being generated accordingly.
- **Output:** Display an error message.
- **Authorization**: The authorization starts after a user has entered his user id and password.

Functional requirement 4:

- **Description:** The System has to check that customer enters Correct Credentials
- **Input:** Customer enters the user id and password.
- **Processing:** Check if it is a valid credentials by verifying from the database. It will be valid if

  ➢ The information entered can be read.

➢ It is not wrong and matches with one that is saved in database.

- **Output:** Display error message and return login screen if it is invalid.

Functional requirement 5:

- **Description**: If the credentials is valid, the system should read the user id's initial to check the user type.
- **Input:** Valid User Details
- **Processing:** Read the initials of the user id.
- **Output:** Initiate authorization dialog accordingly.

Functional requirement 6:

- **Description Authorization dialog:** The user is requested to enter his password. The system verifies the user id and password with the one in database/
- **Input:** Password from user, User id.
- **Processing:** Verify for correctness and return true if it matches the database.
- **Output:** Accept or reject authorization from the server.

Functional requirement 7:

- **Description:** Different negative answers from the server for authorization dialog.
- **Input:** Response from server or authorization dialog:

    -"Bad password" if the password was wrong.

    -"Bad User Id" if the Id doesn't matches with the one present in database.

     -"Bad Account" if there are problems with the account.

- **Processing:** If the User End gets any of these messages from the server, the login would be failed and user would be shown relevant display message.
- **Output:** User is Logged out and prompted to login screen and error message is displayed.

Functional requirement 8:

- **Description:** If password and user id are Correct, the authorization process is finished.
- **Input:** The user gets accepted from the server from authorization process.
- **Processing:** Finishing authorization.
- **Output:** Start menu for website dialog.

Functional requirement 9:

- **Description:** If the password is entered wrong for several number of times , user is asked to verify himself by resetting the password by the registered phone number. A message will be displayed that the user has to reset the password to login again.
- **Input:** Entering a wrong password for the five time in succession

- **Processing:** Initiate authorization process Response from server is to temporally lock the account.
- **Output:** Display error message that the customer should use the reset option now.

Functions: These are the requirements for the different functions the System should provide after authorization.

Functional requirement 10:

- **Description:** Kind of Transaction user wants to do(e.g Consultation,Get Insured ,Order Medication etc)
- **Input:** Authorization successfully completed. Select input from various listed options.
- **Processing**: Option is selected and processed side by side.
- **Output:** Separate screen with those details would be opened.

Functional requirement 11:

- **Description:** User is connected to the doctor for consultation
- **Input:** Customer has chosen the doctor and paid the consultation fees from his account
- **Processing:** Error if the doctor isn't available at that instance or fees isn't being paid yet.
- **Output:** Start connecting again or re-initiate transaction dialog if the amount is not being paid yet..

Functional requirement 12:

- **Description:** Perform transaction.
- **Input:** Initial step was successful.
- **Processing:** Send request to the database to set up a connection.
- **Output:** Wait for response from the bank computer.On success,A connection is settled up between user and the Doctor.

Functional requirement 13:

- **Description:** If the Consultation was successful,User is provided option to order medicine.
- **Input:** Server gets message "transaction succeeded" from user's end computer.
- **Processing:** System generates a report electronically signed by consulted doctor and sent to user via email and shown on screen.

- Dialog:You may keep the report for further purpose.
- **Output:** After the Customer has taken the report,he's prompted to buying medication page.

Functional requirement 14:

- **Description:** If interested, user can buy medication online.
- **Input:** The consulted medicines are automatically shown up on the screen.
- **Processing:** System checks the given report for prescription and give the list of medication being suggested to the user
- **Output:**User is asked to complete transaction or can abort the screen.

Functional requirement 15:

- **Description:** If the user has aborted the screen he's being redirected to the main page..
- **Input:** System gets message "transaction aborted" from the server.
- **Processing:** User End displays error message. Dialog: You'll be redirected within a moment
- **Output:** Redirected to the main page

4. External Interface Requirements

4.1 User Interfaces

The customer user interface should be intuitive, such that 99.9% of all new users are able to surf and navigate on the website without any assistance and problems.

4.2 Hardware Interfaces

The hardware should have following specifications:

- Ability to read the user inputs
- Touch screen for mobile connections
- Stable and an Active internet connection.
- Ability to connect to data base
- Ability to give outputs to users through a display.
- Ability to validate a user

4.3 Software Interfaces

The software interfaces are specific to the target website and user consultation systems**.**

5.Other Nonfunctional Requirements

5.1Performance Requirements

- It must be able to perform in unstable connections also..
- Uninterrupted interrupted connections
- High data transfer rate
- Able to handle many users at a time.
- Quick Response every time.

5.2 Safety Requirements

- Must be encrypted in the back-end
- Must be DDoS protected.
- Must have a detection technique for distinguishing between bots and Humans.
- Must be able to detect illegal/impermissible commands.

5.3 Security Requirements

- Users accessibility is censured in all the ways
- Users are advised to change password every two months
- Users are advised not to share their account with anyone
- The maximum number of attempts to enter password will be five

5.4 Software Quality Attributes

Security

Performance.

5.4.1 Availability: The website has to be available 24 hours a day with at least 100 doctors online at minimal.

5.4.2 Security: The network should provide maximal data security .In order to make that much more transparent there are the following requirements:

1. It must be impossible to plug into the network.
2. It must be impossible to interrupt / peek through one's connection

5.4.3 Maintainability: Only maintainers are allowed to maintain the system and the server .

6. Other Requirements

6.1 Data Base

The system must be able to use several data formats according to the data formats that are provided by the data bases of different users.A saved user data must have all the required attributes, i.e Atomicity, Consistency, Isolation, Durability.

6.2 Constraints

- Internet connection is a major factor.
- Only one active session is permissible at one time.
- Password must always be hidden behind special characters.