130618F-Vinujan.S

Survey Builder System
Software Requirements Specification

Version <1.0>

Revision History

Date	Version	Description	Author
3//4/2016	1.0	Initial creation	Vinujan.S

Table of Contents

Software Requirements Specification

1. Introduction

1.1 Purpose

The purpose of this document is to fully describe the survey builder system. It is designed for patient management system to conduct survey though it can be used to any surveying purpose. This document also includes non-functional requirements, design constraints, and other factors related with the application to provide a complete and comprehensive description of the requirements for the software. Furthermore, this documents shows how the system interfaces are designed in detail.

1.2 Scope

- Admin - create questions with question type and sub-questions and answer type specifications including options, text and multimedia features like images. Review minable data
- Normal User - Answer to questionnaire
- Analytic User - Can extract minable data

1.3 Definitions, Acronyms, and Abbreviations

The following table explains the terms and abbreviations used in this document.

MTBF	Mean time between failure
MTTR	Mean time to repair
KLOC	Thousands of lines of code
SRS	Software Requirement System
RUP	Rational Unified Process
Analytic user	A user role who can access analyzed data
Admin	A user role who can create questions and answer
Normal user	A user role who face questionnaire
BSD	Berkeley software distribution.

1.4 References

[1] Se.inf.ethz.ch, 'Requirement Specification' 2011. [Online].Available:http://se.inf.ethz.ch/courses/2011a_spring/soft_arch/exercises/02/Requirements_Specification.pdf. [Accessed 04 04 2016].
[2] En.wikipedia.org, 'Rational Unified process' 2016. [Online].Available: http://en.wikipedia.org/wiki/Rational_Unified_Process. [Accessed 04 04 2016].

1.5 Overview

This document is created according to the Rational Unified Process standard format. Chapter two gives the overall description. This section describes the general factors that affect the product and its requirements. It provides the background of the requirements. Product perspective, product functions, user characteristics, constraints, assumptions and dependencies and requirements subsets. Chapter three provides the details about the specific functional requirements and non-functional requirements. It also includes design constraints, interfaces, and application standards, purchased components, licensing and copyrights. Final chapter includes the justification for the SRS template selection.

2. Overall Description

2.1 Product Perspective

This is a web application where admin user will create questions and answer. These questionnaire will be stored in database. For normal user suitable questionnaire will be displayed and answer will be stored in database. Answer data will be anonymised and stored in database to retrieve data for analytic user purpose.

2.2 Product Functions

With this application question will be created and showed to user. The data from user will be stored and also will be used for analytic purpose.

2.3 User Characteristics

Admin will drag and drop questions types and answer types. Questions can be nested. Questions and answer may have image, audio and video. These multimedia files will also be stored in database.

Normal user will answer to questionnaire where the answer basically text but it can be image, audio or video.

Analytic user can see his needed data by graphical and tabular view.

2.4 Constraints

Web server capacity is a constraint for this application. Because server will be accessed simultaneously by many users.

In addition to that internet connection is another constraint for this application. Without internet users cannot access anything.

2.5 Assumptions and dependencies

First assumption is application will work in most browser types. Then hardware requirement is assumed to be enough to run browser and application.

2.6 Requirements Subsets

Core Requirements

- Admin should create questions.
- Admin should specify answer type and answer.
- User should be get survey.
- Analytic user should get statistical anonymised data in graphical and tabular form.

3. Specific Requirements

3.1 Functionality

The main functional requirements of this android mobile application are follows.

- Creation of questions.
- Creation of answer.
- Normal user facing question.
- Analytic user retrieve data.

3.1.1 Creation of questions

Admin can create questions by drag and dropping question type according to need from question type tray. Question type tray will contain possible types of question. After that admin can input question. Question can contain sub questions. Admin can add sub question as adding questions. Questions can contain images, audio and video. These multimedia file will be stored in order to survey user and also admin take image, audio and video using his computer at the moment.

3.1.2 Creation of answer

Creation of answer is like creation of question. After creating a question admin should create answer. Admin can input multimedia files as answer. Admin also can add description of answer and comment box for user to input detail other than given answer given.

3.1.3 Normal user facing question

User will be prompt with suitable questionnaire. Questionnaire will change according to time, and place and other factors for user to user. User can partially answer and later can finish questionnaire.

3.1.4 Analytic user retrieving data

Analytic user can get tabular and graphical form of data by giving constraints of need. Privacy of answer will not be shown by encrypting sensible data. This application is initially developed for patient surveying so anonymity is crucial. Application can be customized according to purpose questionnaire used.

3.2 Usability

3.2.1 Time required learn about the application and initial configuration time

First time user should be easily learn about the application in few minutes. Guidelines to the new user should be given.

3.2.2 Graphical User Interface (GUI)

User interface should be attractive. Front end will be used using materialised graphical user interface. So that users will be able to use this application with a specific interest. Clickable buttons must be show clearly. Rather than explaining all things, having read more options. Use contrasting color to attract the user.

3.2.3 Less data usage

To get the information from the back end database this system use internet connection so the amount of data used to drive information should be less

3.3 Reliability

3.3.1 Availability

System should be available for 99% time, since this application will become useless if it is not available on users' applicable time.

3.3.2 MTBF and MTTR

MTBF is 1 year and the MTTR is 1 hour as this application is expected to operate on emergency. Emergencies cannot be predicted.

3.3.3 Accuracy

This application should produce accurate results since during a blood emergency even the small faults cannot be tolerable as they may lead to a life to danger.

3.3.4 Maximum bug rate

There will be a maximum of 1 bug/KLOC

3.3.5 Security consideration

As the answers collected from normal user are sensible data. Unauthenticated access to database will be restricted for this application. Authentication process will updated time to time according to technological changes in order to prevent hackers intrude the system.

3.4 Performance

3.4.1 Response time for a transaction

Average response time for a transaction is 1 minute and the maximum acceptable response time for a transaction is 2 minutes. As specified earlier, during an emergency, response time plays a critical role in changing user`s mind set.

3.4.2 Capacity

Currently, the application is expected to handle less than 1000 user at a time. But in the future, the capacity is expected to increase.

3.4.3 Resource Utilization

As this is mobile application, utilization of resources should be very much minimized.

3.4.4 Resource Utilization

The maximum transaction per second will be 1

3.5 Supportability

3.5.1 Scalability

Scalability can be increased by considering size of simultaneous users in two ways. That are scaling up and scaling out. Scaling up means increasing the number CPUs or adding faster CPUs on a single box. Scaling out means increasing the number of CPU boxes. Scaling up and scaling out want increase throughput if the application is IO bounded. Architecture of the application should be designed avoiding centralized component and high latency component because these two category will create upper limit for throughput. Database should be maintained in order to speedup operation in large amount of data. Database can be horizontally (sharing) and vertically (partitioning) spitted to speed up database operations when amount of data increases.

3.5.2 Expected changes

Need for change will come when admin wants new types of question (or answer) added into question (or answer) type collection tray (admin drag and drop question or answer type in order to create questions or answer) and analytic user want new analytic data. These can be simply handled by further development

3.5.3 Maintainability requirements

Changing a process in this application easy because of using mongo db. Storing of different new types of questions and their answer is feasible. As written in scalability, efficiency can be maximized by reducing architecture bottlenecks and scaling database.

3.5.4 Configurability

Application can be distributed by adding or removing component according to customer needs.

3.5.5 Localizability

This is the ability of the application to be geared toward local conditions and requirements. Language support can be achieved by using UTF-8 encoding.

3.5.6 Install ability

This is the requirements related to system installation and the ease with which it can be done. There is no need for this as long as this is a web application.

3.5.7 Compatibility requirements

As long as it is a web application almost all browser supports well. Compatibility not as issue.

3.6 Design Constraints

3.6.1 Software Languages

This application should be developed using JavaScript for client side and java for server side programming.

3.6.2 Development Tools

For development intellij idea ide (java) and WebStrom ide (JavaScript) should be used.

3.6.3 Development Tools

To save the question and answer created by admin and to save the answer given by normal user mongo dB will be used. To track sub questions neo4j will be used. For analyzing data postgresql will be used.

3.7 On-line User Documentation and Help System Requirements

3.7.1 Help

Online help should be provided to first time users and the returning users also should be provided some help.

3.8 Purchased Components

The system doesn’t use any purchased components.

3.9 Interfaces

3.9.1 User Interfaces

Admin user will have user interface with questions and answer tray and he can drag and drop question and answer template in the blank space provided to edit questions and answers.

Normal user will have attractive materialised user interface with questions and answer.

Analytic user will have interface to select constraints to retrieve needed data form.

3.9.2 Hardware Interfaces

Web application can be access via any smart phones and computers.

3.9.3 Software Interfaces

Popular browser are recommended to access this web application.

3.9.4 Communications Interfaces

Communication should happen with client device and the server in order to fulfill the requests from the clients.

3.10 Licensing Requirements

As the front end will be developed by rectjs wich is a JavaScript framework from Facebook it has BSD license. This application will be released under BSD license. So this application should satisfy the requirements enforced by BSD license.

3.11 Legal, Copyright, and Other Notices

Since this is open source software, the necessary source codes will be provided under BSD license. Redistribution and modification of the software should happen according to that license. Lifetime support and maintenance will be provided to the users of this system.

3.12 Applicable Standards

Software quality standard in accordance with ISO/IEC 9126:2001 and ISO/IEC 25010:2011.

4. Supporting Information

Table of contents is in the starting page & please refer the feasibility document for more supporting information.