Software Requirements Specification

for

**WEB APPLICATION FOR AUTOMATIC SURVEY QUESTION CURATION**

**Version 1.0 approved**

**Prepared by**

**Azam Ahmad Khan**

**Chintan Rank**

**Monal Rupapara**

**University of Massachusetts, Boston**

**October 6th, 2019**

**Table of Contents**

**Table of Contents ii**

**Revision History ii**

**1. Introduction 3**

1.1 Purpose 3

1.2 Document Conventions 3

1.3 Intended Audience and Reading Suggestions 3

1.4 Product Scope 4

1.5 References 4

**2. Overall Description 5**

2.1 Product Perspective 5

2.2 Product Functions 5

2.3 User Classes and Characteristics 5

2.4 Operating Environment 5

2.5 Design and Implementation Constraints 5

2.6 User Documentation 5

2.7 Assumptions and Dependencies 6

**3. External Interface Requirements 6**

3.1 User Interfaces 6

3.2 Hardware Interfaces 6

3.3 Software Interfaces 6

3.4 Communications Interfaces 6

**4. System Features 7**

4.1 System Feature 1 7

4.2 System Feature 2 (and so on) 7

**5. Other Nonfunctional Requirements 8**

5.1 Performance Requirements 8

5.2 Security Requirements 8

5.3 Software Quality Attributes 8

5.4 Business Rules 8

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

The Surveyfy is a Web Application for Automatic Online Survey Question Curation. With the help of a machine learning model, the web app aims to create and deploy online surveys and analyze implicitly generated parameters so that data gathering is accurate and efficient.

This project has already been hosted on a container-based cloud Platform as a Service (PaaS) called **Heroku**. However, the project currently does not involve any analysis of data using machine learning models.

## Purpose

The purpose of this document is to present in detail, the description of the **changes and additions that will be**  made to the web application. This changes will be mostly made in the admin module.

The document is also intended for review and evaluation by the project guide, Dr. Kenneth Fletcher.

## Document Conventions

|  |  |
| --- | --- |
| Web App | The proposed Web Application |
| Stake Holder | Any person interacting with system (not the developers) |
| IEEE | Institute of Electrical and Electronics Engineers |
| Admin | Stakeholer with administrative privileages |
| Respondent | Stakeholer who interacts with it |
| UI | Interface with which the stakeholders interact (User Interface) |

## Intended Audience and Reading Suggestions

This document is created for Project’s Client and mentor for reviewing and monitoring the progress of

Project whereas for the Project Developers to analyse the requirements, develop the web app and

analyse the progress.

## Product Scope

The scope is to develop this current web app for stakeholders to be more user-friendly and more interactive. Admin will be able to create the surveys more efficiently and easily where as the part with the respondents will serve the same as before , by using the external machine learning system.

## References

Standard SRS format given by “Institute in Electrical and Electronic Engineers” (IEEE).

# Overall Description

## Product Perspective

*The perspective of this product is to develop the current web app with easily accessible admin module and making this web app’s UI more responsive with desktop and especially with mobile interface.*

## Product Functions

There will be two major stakeholders for the system, one would be admin and the other would be respondent.

There will be two major modules in system as:

1)Admin – Admin shall be having the privileges to create, edit, delete and managing the surveys.

2)Respondent – Respondents will have the access only for taking various survey tests created by admins.

## User Classes and Characteristics

This product could be used by various educational institutes, corporates, healthcare department, Human Resource Department and also could be used in many other sectors depending on the importance of survey results to them.

## Operating Environment

The Technologies that would be used are as follows:

1. Frontend - HTML 5, CSS, /Boot-strap, JavaScript, J-Query

2. Backend – Express.js Java Script

3. Runtime – Node.js

4. Framework – Express.js, body-parser, express-session,

passport, passport-local-mongoose, Git (version control

system)

5. DBMS – MongoDB

Some changes might be done with the technologies later on if required.

## Design and Implementation Constraints

The web app will be designed in a way that it would be able to handle around three thousand respondents.

## 2.6 Assumptions and Dependencies

The assumption is that it will be used mostly on mobile or computers compatible with the optimal performance of the web app. The usage of charts and graphs for viewing the survey reports on the admin dashboard.

# External Interface Requirements

## User Interfaces

**Major changes**

This is where the majority of changes will be made. The current **admin module** to create surveys is not so user friendly. The admin dashboard layout/interface - such certain buttons, menus are placed and how a function responds will be changed. We shall be using Google Forms / Qualtrics / Survey Monkey’s UI as a reference to make the changes in our front end.

The UI for respondents if needed, shall be improved.

After these changes are completed, only then machine learning model shall be implemented with the data collected using this application.

## Hardware Interfaces

The web app will not have a hardware interfaces as the operating system will take care

of the functionalities***.***

## Software Interfaces

This is an existing project in which the versions of technologies being used could be updated in future.

## Communications Interfaces

A session is created between the web app and the respondent. The data is uploaded to

the database after the respondent clicks submit button. The admin creates a survey and

generates a link, which is sent to a respondent.

# System Features

System features is the part of the documentation where the proper functionality of the system is derived with the flow of events through which the application is going to work. Here, use cases gives the visual representation of the new system involving the entities and their corresponding functionalities.

**Use Cases:**

**Prerequisites:**

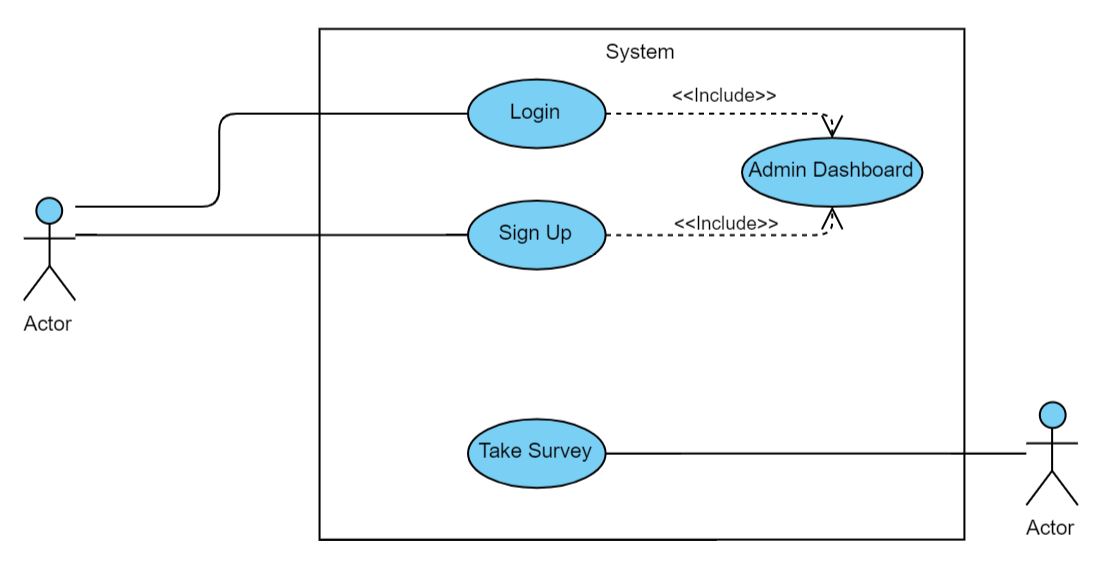
1. This use case begins when a user visits Surveyfy Home page.

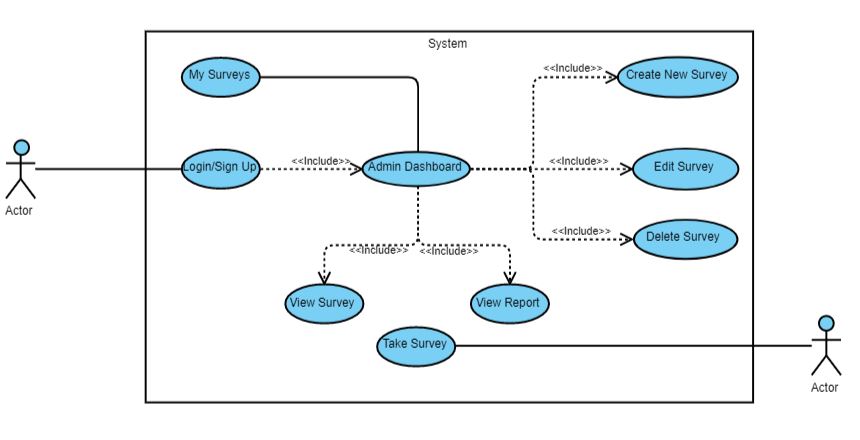
2. A User can be an Admin (who creates surveys) or a Respondent.

3. User clicks Login button.

4. The System displays a new page where user enters User Email, Password and clicks Login button.

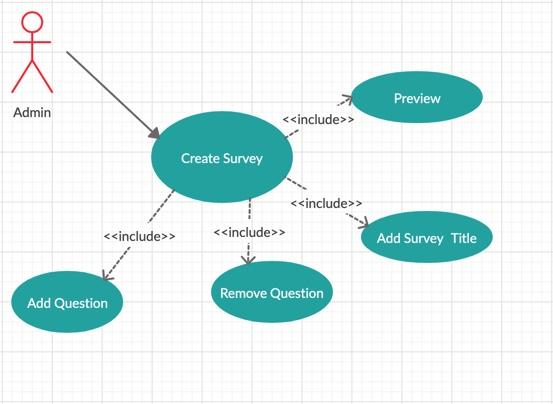
5. User lands to the Admin Dashboard page.



**Event Flow for Create Survey:** 

1. The Admin is at the Dashboard page.

2. Admin clicks Create Survey to create a new survey.



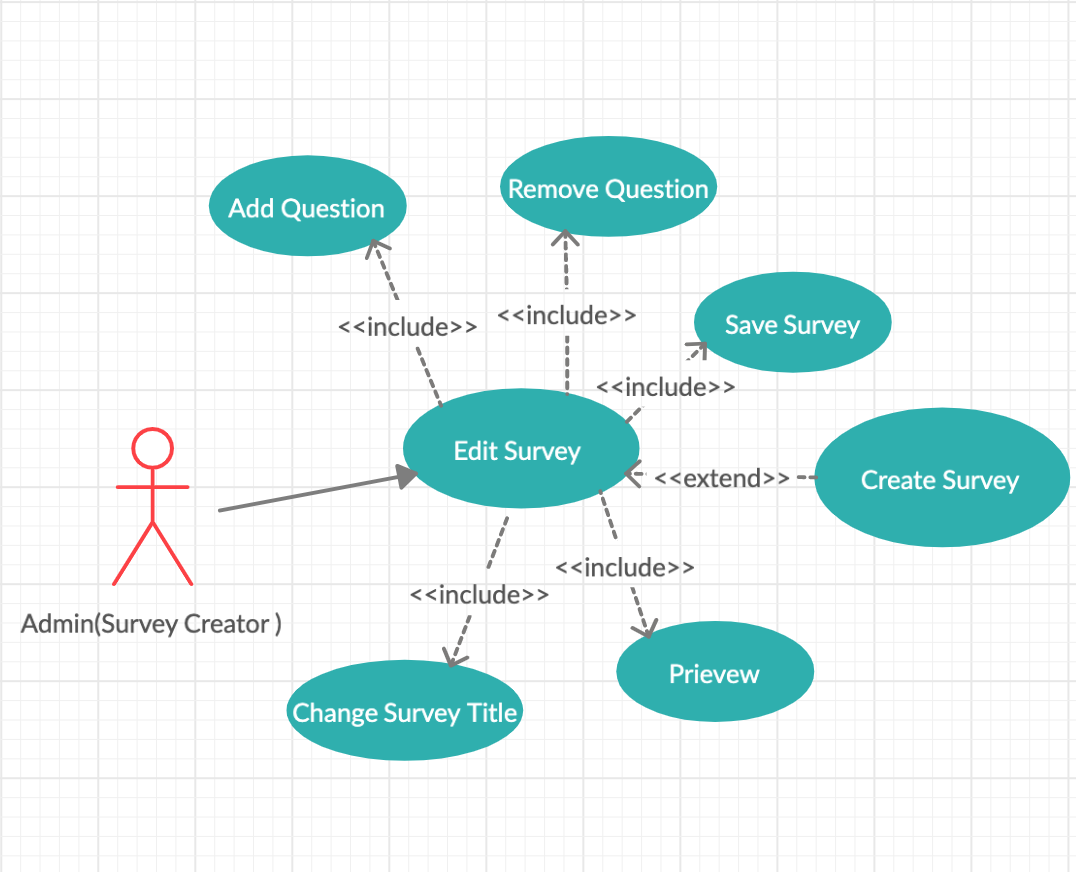
1. Admin sees a Popup where he will add a Survey Title and clicks Create Survey button.

2. Admin goes to a new page where he adds questions that page.

3. Admin saves the question in the survey by clicking Save button.

4. Admin goes back to Dashboard Page where he can share the survey using Sharable Link.

**Event Flow for Edit Survey:**



1. Admin is at Dashboard Page.

2. Admin clicks on Edit Survey for the corresponding Survey to be edited.

3. Goes to a new page where Admin can edit Survey title, edit questions, add/remove questions.

4. When done clicks on Save Survey button.

5. Goes back to Dashboard Page where he can share the survey using Sharable Link

**Event Flow for Delete Survey:**

1. Admin is on Dashboard page.

2. Admin clicks on Delete Survey button corresponding to the survey he wants to delete.

3. Pop up is displayed to confirm the deletion.

4. Clicking on Confirm button will delete the survey.

5. Admin lands back to the Dashboard page.

**Event Flow for View Survey:**

1. Admin is on Dashboard page.

2. Admin clicks on View Survey button corresponding to the survey he wants to delete.

3. The admin goes to a new page where ‘detailed view’ of the survey is displayed.

4. The admin can see results and can edit, delete, analyze the survey clicking its Edit/Delete/View Report buttons.

**Event Flow for My Surveys:**

1. The Admin is on Dashboard.

2. Admin clicks on My Surveys button.

3. The admin goes to a new page where detailed view of all surveys are displayed.

**Event Flow for View Report Survey:**

1. The Admin is on Dashboard or in View Survey page.

2. Admin clicks on View Report button for the corresponding Survey to be analyzed.

3. Goes to a new page where analytics of that survey are displayed.

## 4.1 Admin Survey Manager

The Edit Survey UI, which is a part of admin’s module, would be changed accordingly using the reference as of Google Forms/Survey Monkey/Qualtrics. The layout/appearance of Create Survey’s UI shall also be changed. The major focus shall be on making the admin to create the survey questionnaire very easily with the help of new survey editor layout.

## 4.2 Admin Dashboard

The admin dashboard would be able to navigate easily through all the surveys on the same page, add-on to this it will state information like the total number of questions in each survey that have been created. The Dashboard would be able to manage surveys, edit surveys, delete surveys with a new interface. All the changes stated above would be made keeping in mind that the application will remain responsive.

## 4.3 Pre-Built Survey Templates - (To Be Decided)

One of the major changes would be to include ‘Inbuilt Templates’ that could be used by Admin to create the survey on basis of preloaded questionnaire. These templates would be loaded according to the choice that shall be made by admin by pressing the links or buttons for given pre-built template.

# Other Nonfunctional Requirements

## Performance Requirements

The performance constraints specify the timing characteristics of the software. In

our project since we are partitioning the public cloud and also by assigning job to the

server with highest memory reduces the access/process time, hence improves the

performance.

## Security Requirements

The web app will be implemented with passport, express-session as well as passport-local mongoose like framework. This requirements might change if required.

## 5.3 Software Quality Attributes

Other software attributes are adaptability, availability, correctness, flexibility,

interoperability, maintainability, portability, reliability, reusability, robustness, testability,

and usability. Write these to be specific, quantitative, and verifiable when possible. At the

least, clarify the relative preferences for various attributes, such as ease of use over ease

of learning.

## 5.4 Business Rules

*<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>*