Software Requirements Specification

for

PPA Membership Maintenance System

Prepared by G.D.Janith Ronaka
T.T.T.Rathnaweera
H.M.D.L.Gunasiri
B. Hingalagoda

10/08/2018

Table of Contents

Revision History

Name	Date	Reason for Changes	Version
Submission 1	10/08/2018	Initial Submission	1.0

1. Introduction

1.1 Purpose

This SRS is to provide a detailed documentation of the functional and non-functional requirements of the Past Pupil Association Membership Maintenance System which is going to be developed as a **MEAN** stack web application. This system handles the manual process of Past Pupil Association in a convenient way by enabling most of its facilities to its members in an on-demand web based interface. The main intention of developing this system is to provide a fast and quality service for the past pupil association members while making the administration tasks easier than the manual process in a semi-automated environment.

1.2 Document Conventions

Font style - Times new roman
Main Headings - Font size 23(Bold)
Sub Headings - Font size 17(Bold)
Rest of the Text - Font size 12
Line spacing - 1.15
Important terms will be highlighted in bold letters.

1.3 Intended Audience and Reading Suggestions

The content of this document is intended to be referred by the following parties.

Development Team	Main system development team
Testers	The testers who will test the system
Client	This document is a legal binding between the client and the development team.
Academic Supervisors/ Lecturers	Supervisors and lecturers who in-charge of the CCP2 module

1.4 Project Scope

The Past pupil Association of Sirimavo Bandaranaike Vidyalaya is the main body that handles all the services and projects conducted by the past pupils of the school. Therefore the following operations can be identified as the main services and duties of the past pupil association.

- Offer committee memberships to the past students
- Organize fundraising events
- Offer scholarships to the current students
- Collect donations from the past students and well wishers
- Send annual meeting invitations and other relevant information mails
- Issue service letters to the members
- Maintain the accounts

PPA Member Maintenance System seeks to support the processes of past pupil association by automating most of the above mentioned tasks.

1.5 References

[1] Angular.io. (2018). Angular Docs. [online] Available at: https://angular.io/.

[2] Node.js. (2018). Docs | Node.js. [online] Available at: https://nodejs.org/en/docs/.

[3] Expressjs.com. (2018). Express - Node.js web application framework. [online] Available at: https://expressjs.com.

[4] MongoDB. (2018). MongoDB for GIANT Ideas. [online] Available at: https://www.mongodb.com/ [Accessed 9 Aug. 2018].

[5] Npmjs.com. (2018). npm. [online] Available at: https://www.npmjs.com/ [Accessed 9 Aug. 2018].

[6] MongoDB. (2018). Fully Managed MongoDB, hosted on AWS, Azure, and GCP. [online] Available at: https://www.mongodb.com/cloud/atlas.

[7] Heroku.com. (2018). Cloud Application Platform | Heroku. [online] Available at: https://www.heroku.com/ [Accessed 9 Aug. 2018].

2. Overall Description

2.1 Product Perspective

The **PPA** Membership Maintenance System replaces most of the manual processes of the past pupil association. The current existing method of offering memberships is a manual workflow where the past students fill out a lengthy application form and submit it to the association and the respective committee members review the application and offer the membership. Likewise the accountings also keep in manual ledgers. With the proposed system all of these workflows are taken care by the web application and provide the both administration and the committee members a user-friendly interface to fulfill their requirements fast and easily. Another main benefit of the system is it provides a dedicated webpage even for the well-wishers and guests to see the projects and events organize by the PPA and let them donate for the any project or event as they wish. The accounting part is automatically handled by the system and the administration of the committee can request various reports of accounts for any further use.

2.2 Product Features

The development team has identified 4 key areas of the existing workflow. The product is focus its features covering those 4 areas which listed below.

1. Membership Services
2. Project & Event Management
3. Accounting
4. Report Generation

The proposed PPA Membership Maintenance System offers the following features which covers the above 4 key areas. (These features are further discussed under the System Features section)

- Apply for Membership
- Review Membership Applications
- Request/Generate and Issue service letters
- Generate Member Profile Reports
- Query the Member Profiles for Social Connections
- Collect Donations
- Automated Accounts Handling
- Send Meeting Invitation Emails
- Add/Edit Projects
- Post Updates about Projects
- Add Important Events on System Calendar
- View Finalized Budget Reports of the Projects
- Add Suggestions and Complaints

2.3 User Classes and Characteristics

This developed web software solution will have three major characters who will interact with the finished solution. According to the user permission levels they will have ability to view and manipulate data.

- Admin:

  Admin is expected to have little bit of IT knowledge to identify how the system behave. Admin will have the permission to do things like add users to the system, add projects, create an event calendar, manage finance, and generate reports and other administrative tasks.

- System Users:

  System user are expected to have basic IT knowledge to interact with the system. Users will have permissions to view and make donations to the projects, add suggestions and complaints about projects, request services letters and view details about donated projects.

- Guests:

  Guest users are expected have basic IT knowledge to work with the web solution. Guest will have permission to view projects and make donations to them and view event calendar.

2.4 Operating Environment

This Web software solution will be developed with the capability to run in any Web Browser that currently available. System users and the gust users will be able to access through this web solution to the database server that will be developed using Mongo DB, across the internet.

2.5 Design and Implementation Constraints

The user interfaces along with client-side logic of the web application is to build using Angular [1] framework and a modern web browser is needed to properly view the application.

2.6 User Documentation

A user manual on how to use the application will be provided with the application.

2.7 Assumptions

- Users have basic knowledge on web browser usage.

2.8 Dependencies

The application will depend on the following.

Angular	Javascript framework which is mainly used to build user interfaces and client-side programming.
Node [2] and Express JS [3]	Javascript frameworks which is used to middle tier logic along with server.
MongoDB [4]	No-SQL database which is used to data as collections of objects.
Node Package Manager [5]	Javascript package manager which is used to install various packages and resolve their dependencies.
MongoDB Atlas [6]	Cloud database hosting service which is used to host MongoDB database.
Heroku [7]	Cloud platform which is used to build and host the web environment.

3. System Feature

Following are the features which is supported by the proposed system that will facilitate the end user to cover the key areas of the existing workflow.

3.1 Add Projects

Use Case Id	REQ-1
Use Case Name	Add Project
Priority	High
Description	Admin should be able to add projects to the system
Actor	Administrator
Prerequisites	User should be logged in as an administrator
Main Flow	1. Navigate to the Projects page 2. Click on the Add Project button 3. Add a project name and fill the relevant info fields. 4. Click the Next button 5. Add images to the project gallery 6. Add the project committee 7. Click on Finish button

3.2 Query for Social Connections

Use Case Id	REQ-2
Use Case Name	Query for Social Connections
Priority	Medium
Description	Committee members (Administrators) should be able to view the members who have relevant social connections which can be used when a project is being organized. Ex: When organizing a blood donation camp committee members should be able to view the members who are or who have connections with doctors.
Actor	Administrator
Prerequisites	User should logged in as an Administrator
Main Flow	1. Navigate to the Social Connections page.. 2. Enter the required occupation / skill set to query 3. Click on the Search button. 4. View the members list who have connections. 5. Send an email to selected users or all users.

3.3 Request service letters

Use Case ID	REQ-3
Use Case Name	Request service letters
Priority	Low
Description	Member should be able to request service letter
Actor	Member
Prerequisites	User should be logged in.
Main Flow	1. Navigate to request service letter page 2. Fill in information 3. Click on “Request” button

3.4 Accept Membership Applications

Use Case ID	REQ-4
Use Case Name	Accept Membership Applications
Priority	High
Description	Admin should be able to view applications and accept
Actor	Admin
Prerequisites	User should logged in as an administrator
Main Flow	1. Navigate to review applications page 2. Click on an application 3. View application details 4. Click on “Accept” button

3.5 Reject Membership Applications

Use Case ID	REQ-5
Use Case Name	Reject Membership Applications
Priority	Medium
Description	Admin should be able to view applications and reject
Actor	Admin
Prerequisites	User should logged in as an administrator
Main Flow	1. Navigate to review applications page 2. Click on an application 3. View application details 4. Fill rejection reason 5. Click on “Reject” button

3.6 Apply for Membership

Use Case ID	REQ-6
Use Case Name	Apply for Membership
Priority	High
Description	A guest should be able to apply for membership
Actor	Guest
Prerequisites	-
Main Flow	1. Navigate to apply membership page 2. Fill in the details 3. Click on “Apply” button

3.7 Generate Report for the Donations Done

Use Case ID	REQ-7
Use Case Name	Generate Report for the Donations Done
Priority	Medium
Description	Admins hold be able to Generate Report for the Donations Done
Actor	Administrator
Prerequisites	User should be logged in as administrator
Main Flow	1. Navigate to report generation page 2. Select Donation reports 3. Enter query criteria 4. Click on Generate report 5. View / download report

3.8 Send Meeting Invitation Links

Use Case ID	REQ-8
Use Case Name	Send meeting invitation links
Priority	High
Description	A member should be able to send meeting invitation links
Actor	Member
Prerequisites	User should be logged in.
Main Flow	1. Navigate to home page 2. Click on “Create Meeting Request” button 3. Fill the relevant information for the meeting 4. Click “To” button to select members 5. Click “Send” Button

3.9 Generate Member Profile Reports

Use Case ID	REQ-9
Use Case Name	Generate Member Profile Reports
Priority	Medium
Description	An admin should be able to generate member profile reports.
Actor	Administrator
Prerequisites	User should be logged in.as administrator
Main Flow	1. Navigate to Report Generation page 2. Select Member Profile Reports 3. Select one or more members from the list 4. Click “Generate Report” button 5. View/Download the report

4. External Interface Requirements

4.1 Hardware Interfaces

The following are the minimum hardware requirements for the web application.

- Client Side
  - Minimum Dual Core Processor
  - 512MB RAM
- Server Side
  - Minimum Core 2 Duo Processor
  - 2GB RAM
  - 1GB Disk Space
- Database
  - 1GB RAM
  - 1GB Disk Space

4.2 Communications Interfaces

Web browser communicates with server using **HTTPS** protocol. To use this system a working internet connection is mandatory.

4.3 Software Interfaces

Client-side Software interface mainly depend on Angular 6 browser support. For laptops and desktops and mobile devices will support below mentioned web browsers

1.	Chrome	latest
2.	Firefox	latest
3.	Edge	2 most recent major versions
4.	IE	11, 10, 9
5.	IE Mobile	11
6.	Safari	2 most recent major versions
7.	iOS	2 most recent major versions
8.	Android	Nougat (7.0), Marshmallow (6.0), Lollipop (5.0, 5.1), KitKat (4.4)

Server-side software interfaces

1. Node.js web server 2. MongoDB server

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- Page loading time is minimized by using optimized web pages and lazy loading technique.
  Report generation should be optimized for quick results.
- System generated emails should be delivered to customers quickly.
- Server should be able to handle at least 20 concurrent users with minimum performance drop.

5.2 Safety Requirements

- Data saved in server should be backed up periodically to prevent data loss.

5.3 Security Requirements

- System should use up to date libraries with no critical security vulnerabilities.
- For communication via internet a secured protocol must be used.
- Critical user data must be well protected within the system using methods such as data encryption.
- Security level configuration on user accounts will restrict access for security critical data/ web pages. When users forgot their password, they should be able to reset their password by receiving an e-mail to registered e-mail account to that user account.

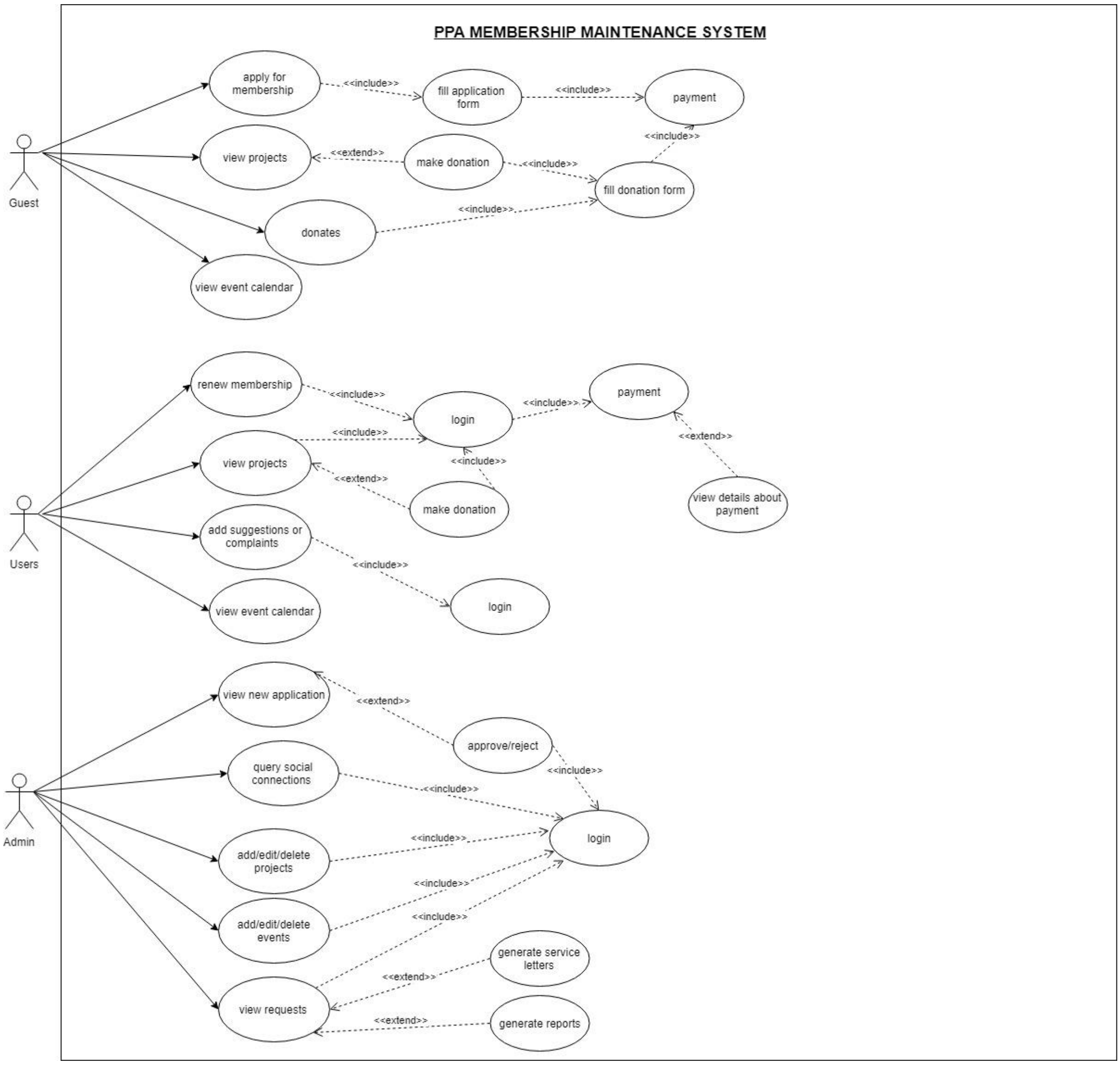
5.4 Software Quality Attributes

When developing the application should standards that will result in maintainable and secure system. Application should be tested after each sprint and ensure that the quality of the application is not compromised. Application should be tested on all supported browsers and make sure everything is working as expected.

Appendix A: Glossary

MEAN	Collection of JavaScript based technologies used to develop web applications
PPA	Past Pupil Association
HTTPS	Hypertext Transfer Protocol Secured

Appendix B: Use Case Diagram


PPA MEMBERSHIP MAINTENANCE SYSTEM
Guest
apply for membership
<<include>>
fill application form
<<include>>
payment
<<include>>
view projects
<<extend>>
make donation
<<include>>
fill donation form
donates
<<include>>
view event calendar
Users
renew membership
<<include>>
login
<<include>>
payment
view projects
<<include>>
<<include>>
<<extend>>
<<extend>>
make donation
view details about payment
add suggestions or complaints
<<include>>
login
view event calendar
Admin
view new application
<<extend>>
approve/reject
<<include>>
query social connections
<<include>>
login
add/edit/delete projects
<<include>>
add/edit/delete events
<<include>>
<<include>>
generate service letters
<<extend>>
view requests
<<extend>>
generate reports