P.L.B. Sampath (100476M)

# EventShare
# Software Requirements Specification
**Version <1.0>**

# Table of Contents

# Software Requirements Specification

## 1. Introduction

### 1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to provide a detailed description of the functionalities of the "EventShare" system. This document will cover each of the system's intended features, as well as GUI features.

### 1.2 Scope

The aim of this project is "improving events sharing" over the internet. This system will provide users to create event-sets like to-do lists, timetables, schedules, etc.. and to share them with others. Once these are created the user can publish the URL of the event-set. Then others can subscribe to that. The system also will send notifications to subscribers about day today events. This project includes developing two main components for users to interact with, a web component and an android component.

### 1.3 Overview

First this document gives an overall description of factors that affect the product and its requirements. It provides a background for the requirements. Then this discusses software requirements under sub topics like, usability, reliability, performance and etc.

## 2. Overall Description

This project includes an implementation of a web server to handle events, notifications and users. Also two client side user interfaces to interact with the server, a web component and an android component. Some of main features of this project offers are,

1. Registering for an account and signing in to the profile
2. Creating event-sets like to-do lists, timetables.
3. Publishing event-sets created.
4. Subscribing a published event set.
5. Receiving notifications of the event-sets.

# 3. Specific Requirements

## 3.1 Functionality

### *3.1.1 User registration*

A new user has to visit the web site for registration. User registration form asks for, a user name, name, password and email address. User can register by filling the form correctly and submitting.

### *3.1.2 User sign in*

A user will be asked for the user name and password when signing in. If the user cannot remember the password, the user can request for an email. After signing in user will redirect to a page where all the events are shown.

### *3.1.3 Creating a to-do list*

A user will have to,

1. Give a title and description to the to-do list.
2. Create events for the to-do list - An event of a to-do list includes name, description, date and time. When creating a to-do list.

### *3.1.4 Creating a time table*

A user will have to

1. Give a title and description to the time table.
2. Create events for the time table – An event of a time table includes name, description, day, start time, end time

When creating a time table.

### *3.1.5 Creating a schedule*

1. Give a title and description to the schedule.
2. Create events for the schedule – An event of a schedule includes name, description, start date and time, end date and time

When creating schedule.

### *3.1.6 Publishing an event-set (to-do list, time table or schedule)*

The creator of the event list can publish it by sharing the public URL given by the system. Also the user can let others to subscribe it by posting the WIDGET provided by the system on a web site.

### *3.1.7 Subscribing an Event set*

A user who owns an account can subscribe an event set with that URL or the WIDGET. A creator himself can subscribe his own event set.

### *3.1.8 Unsubscribing an Event set*

A user who subscribed an event set can unsubscribe it by login in to the account.

### *3.1.9 Changing an event-set*

A creator of an event-set can change it any time after login to the account. Changes will be automatically synchronized to the subscribers.

#### *3.1.10 Deleting an event-set*

A creator of an event-set can delete it any time after login to the account. Changes will be automatically synchronized to the subscribers.

#### *3.1.11 Sending notifications*

The system will send an email about all the events for a day at a time user pre specified.

#### *3.1.12 Google calendar synchronization*

All the events will be synchronized with the user's Google calendar. A user needs provide permission to the system to access user's Google calendar.

### 3.2 Usability

#### *3.2.1 Time taken for creating an event-set is less than a minute.*

To improve user friendliness this is something important. User shouldn't go through several pages to create an event.

#### *3.2.2 Good categorization.*

Different event-set types (To-do lists, timetables, schedules) should be able to be separately easily recognized by the users.

#### *3.2.3 Good notifications.*

Notification email should describe events with lesser words clearly. A user should be able to easily see the separate events.

#### *3.2.4 Provide error messages in a user friendly manner.*

When the system fails to perform an action requested by user the system has to provide proper error reports.

### 3.3 Reliability

#### *3.3.1 Sending notifications on time.*

Notifications should be sent on time as predefined.

#### *3.3.2 Security*

Probability of a user to hack into another account should be minimal. Security features should be tested again and again so that no bugs are there.

#### *3.3.3 Bugs*

Mechanism to report bugs for users is needed.

## 3.4 Performance

### *3.4.1 Response Time*

Response time for each request should be minimal.

### *3.4.2 Power saving*

Android component should save battery power.

## 3.5 Supportability

### *3.5.1 New feature requests from users*

Mechanism for users to send new feature requests is needed.

## 3.6 Design Constraints

### *3.6.1 Google calendar synchronization*

As the event-sets in the system are dynamic (can be changed by the user any time), Google synchronization should be dynamic. This can be achieved only if the user doesn't change the calendar trough Google user interface.

### *3.6.2 Programming languages*

PHP is for server side implementations.
JavaScript, HTML for client side web user interface.
Java for the android development.

## 3.7 On-line User Documentation and Help System Requirements

## 3.8 Interfaces

### *3.8.1 User Interfaces*

- Web user interface.
- Android user interface.

### *3.8.2 Software Interfaces*

- Interface between Google calendar API and server

### *3.8.3 Communications Interfaces*

- Interface between android app and the server.
- Interface between web GUI and the server.
- Interface between the web server and Google servers.