

**Software Requirements**

**Specification**

**for**

**Schedule management System**

**Version 1.0 approved**

**Prepared by Haresh Chawla(10)**

**Vardhaman Roman(50)**

**30th of January, 2017**

Table of Contents

[1. Introduction 1](#__RefHeading__7_110842206)

[1.1 Purpose 1](#__RefHeading__9_110842206)

[1.2 Document Conventions 1](#__RefHeading__11_110842206)

[1.3 Intended Audience and Reading Suggestions 1](#__RefHeading__13_110842206)

[1.4 Product Scope 1](#__RefHeading__15_110842206)

[1.5 References 1](#__RefHeading__17_110842206)

[2. Overall Description 1](#__RefHeading__19_110842206)

[2.1 Product Perspective 1](#__RefHeading__21_110842206)

[2.2 Product Functions 2](#__RefHeading__23_110842206)

[2.3 User Classes and Characteristics 2](#__RefHeading__25_110842206)

[2.4 Operating Environment 2](#__RefHeading__27_110842206)

[2.5 Design and Implementation Constraints 2](#__RefHeading__29_110842206)

[2.6 User Documentation 2](#__RefHeading__31_110842206)

[2.7 Assumptions and Dependencies 2](#__RefHeading__33_110842206)

[3. External Interface Requirements 3](#__RefHeading__35_110842206)

[3.1 User Interfaces 3](#__RefHeading__37_110842206)

[3.2 Hardware Interfaces 3](#__RefHeading__39_110842206)

[3.3 Software Interfaces 3](#__RefHeading__41_110842206)

[3.4 Communications Interfaces 3](#__RefHeading__43_110842206)

[4. System Features 3](#__RefHeading__45_110842206)

[4.1 System Feature 1 3](#__RefHeading__47_110842206)

[4.2 System Feature 2 (and so on) 4](#__RefHeading__49_110842206)

[5. Other Nonfunctional Requirements 4](#__RefHeading__51_110842206)

[5.1 Performance Requirements 4](#__RefHeading__53_110842206)

[5.2 Safety Requirements 4](#__RefHeading__55_110842206)

[5.3 Security Requirements 4](#__RefHeading__57_110842206)

[5.4 Software Quality Attributes 4](#__RefHeading__59_110842206)

[5.5 Business Rules 5](#__RefHeading__61_110842206)

[6. Other Requirements 5](#__RefHeading__63_110842206)

Revision History

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

# Introduction

## Purpose

This project focuses on creating a calendar based notification system with auto update reminders. All your reminders for tests, submissions, tutorials, library dues and group updates will be presented in a clean, organized and easy to use environment. Hence adding ease for users to be in accord with the important events to be occurring.

## Document Conventions

The normal font which was used was times new roman and the font size was specific. We have used bold letters for the main points and also for the sub points.

## Intended Audience and Reading Suggestions

This document can be read by the age group of 18 to 30 and even corporate world maybe anything it would be mostly youngsters college going students and even the people who are tech savy and have the advanced knowledge of various kind of applications available for our needs in day to day life.

## Product Scope

The software which we are using would be web-app where we will be developing a convenient form of calender with auto update reminder system. This adds ease to scheduling and not missing out on any important events.

## References

Various books on web-app coding ,ieee papers for recommendations, various kind of blogs which substitute this problem, working technical papers on the same topic by IIT Institutes and books on web-app.

# Overall Description

## Product Perspective

Users often find it difficult to keep track of everything going on in their life. This results in forgetting important details, for instance of a student being the user can possibly lose track of dates in reference to occurrings  such as the dates when an assignment has to be submitted, or the time and day of a council meet. Whatsapp groups share this information, but as most students realize, it’s very hard to scroll through hundreds of messages to find what they need. And these groups are populated with a lot of irrelevant data and messages which makes finding important messages very hard. These are the kinds of problems that our project aims to solve.

## Product Functions

The user can join groups based on their real life groups. A user will most definitely be a part of the class group and can also join groups that he belongs to, like the college council or societies. Each of these groups have an admin . Say for instance that there is a submission on 27th September. The class group admin will select this date and write a reminder saying something like "submissions today". At the click of a button, everyone in the class group will have their calendars updated with 27th October marked with a message "submissions today". Every reminder will be marked on a calendar layout for easy access. All the reminders from all your groups will be presented in one place. You will also be able to use your calendar as a normal calendar to mark your personal dates or notes that no one else can see or edit.

## User Classes and Characteristics

The product can be used by any user who is accessing gmail account and needs a proper schedule. The product is beneficial for the users who needs to keep a track of their educational time-table specifying about tests, dates and timings.

## Operating Environment

The software would be made in web-app so it would be used on any electronic platform(laptops and personal computers) .The coding would be of web-app itself and it would be linked with the network for the location perspective.

## Design and Implementation Constraints

This project is going to have code for extracting just the important data from the user's mail and edit the personal calender of every user. Only the admin of a certain group can make changes in the calender regarding important notices. Key users can personalize their respective calenders.

## User Documentation

The application would be user friendly leading to no difficulty in understanding the working of the web-app process. Here, web-app is used to create an optimized form of schedule creating program which is referred here in IEEE SRS format.

## Assumptions and Dependencies

The assumption here made is that the admin is the accessing master who can initialize and edit the calender for a certain group of users whereas user is the accessing slave who can edit his/her personalized calender only.

# External Interface Requirements

## User Interfaces

A calendar is VESIT Network is a collection of all the events, submissions, due dates, Group updates etc. in a date wise form. Every student will have one calendar. If a student belongs to a particular group, all the events of that group would be reflected in the calendar.

## Hardware Interfaces

Network would be used as the hardware because for the connection of the web-app with the data shared in gmail accounts.

## Software Interfaces

The class group admin will select the date on which there is a specific announcement to be made and write a reminder saying something like "submissions today". At the click of a button, everyone in the class group will have their calendars updated with the specific date marked with a message "submissions today". Every reminder will be marked on a calendar layout for easy access. All the reminders from all the groups will be presented in one place.

## Communications Interfaces

The user will also be able to use his/her calendar as a normal calendar to mark the personal dates or notes that no one else can see or edit. Apart from the calendar, a BROADCAST option will be available which can be used to send important messages to everyone in the group. To keep unnecessary clutter to a minimum, only the group administrators will be able to broadcast messages.

# System Features

## System Feature 1

**4.1.1** Stimulus/Response Sequences

The web app would be having a quick response as soon the user searches for the important updates.

**4.1.3** Functional Requirements

The user should have a personal computer or a laptop and should also have internet connection.

## System Feature 2 (and so on)

The user can sync his/her daily time-table with the calendar. Cloud based storage could be used to upload all the assignment sheets and other material in well organized folders.

# Other Nonfunctional Requirements

## Performance Requirements

The performance requirements include that the web-app which is going to be developed should not have any kind of issues occuring during its function. In this the user logs in, searches for the group, selects the calendar and views the events and important updates in his/her calendar.

## Safety Requirements

The security requirements include that the application should be having the security for the users id and the information that is provided by the user.

## Security Requirements

The user will be able to access his/her time-table according to the updated schedule. Privacy of every account is restricted to the user and hence ensuring privacy i.e not at all invaded.

## Software Quality Attributes

Adaptability of the web-app should be good. Its availability, correctness, flexibility, interoperability, maintainability, portability, reliability, robustness, testability, and usability should also be verified. The web-app should be tested well.

## Business Rules

The business rules include that the tester should check the web-app every two-three months for detecting some errors and for the proper functionality of the system.

# Other Requirements

The database of the application should be having the data of the users which have been registered previously and the data should be secured.

# 6.1 Maintainability Requirements

The web-app should be easy to handle and update. The code should be written in a way that it favors implementation of new functions.

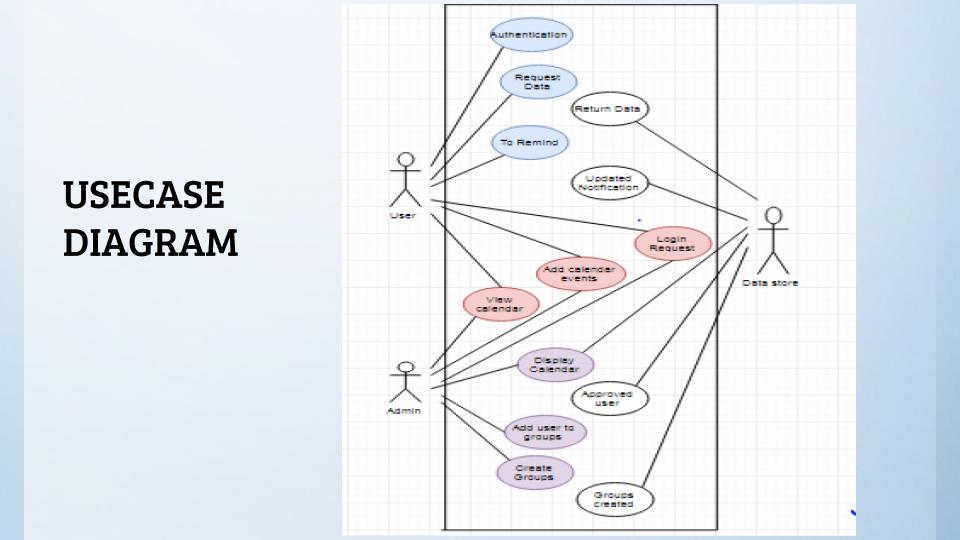
# 6.2 Portability Requirements

The application should be portable with IOS and Android.

Appendix A: Glossary

We have not used any acronyms in this Software Requirement Specification.

Appendix B: Analysis Models

**

Appendix C: To Be Determined List

1).<http://technav.ieee.org/tag/8609/scheduling-algorithm#societies>

2).<http://technav.ieee.org/tag/8609/scheduling-algorithm#conferences>

3).<http://ieeexplore.ieee.org/document/6784130/>