Software Requirements

Specification

**for**

**Automation of Reminder to Student**

**Activities**

#### Prepared by IT004-Gondaliya Harsh K. IT012-Bhalodiya Rushi K.

**01/10/2023**

# Table of Contents

[Table of Contents ii](#_TOC_250006)

Revision History ...............................................................................................................................

1. [Introduction 1](#_TOC_250005)
   1. Purpose Error! Bookmark not defined.
   2. Product Scope 1
   3. Environmental Charachteristics Error! Bookmark not defined.
2. [Overall Description 1](#_TOC_250004)
   1. [Product Perspective 1](#_TOC_250003)
   2. [Product Features 1](#_TOC_250002)
   3. [User Classes and Characteristics 1](#_TOC_250001)
   4. [Operating Environment 2](#_TOC_250000)
   5. Design and Implementation Constraints 2
   6. User Documentation **Error! Bookmark not defined.**
3. **Functional Requirments** Error! Bookmark not defined.
   1. System Feature 1 **Error! Bookmark not defined.**
   2. System Feature 2 (and so on) **Error! Bookmark not defined.**

###### 4. Scenario-Based Model ...............................................................................................................

4.1 Introduction........................................................................................................................................

* 1. Use Case Scenario **Error! Bookmark not defined.**
  2. Use Case Description **Error! Bookmark not defined.**

###### 5. External Interface ......................................................................................................................

5.1 User Interfaces ...................................................................................................................................

* 1. Hardware Interfaces **Error! Bookmark not defined.**
  2. Software Interfaces **Error! Bookmark not defined.**
  3. Communications Interfaces **Error! Bookmark not defined.**

1. **Other Nonfunctional Requirements** Error! Bookmark not defined.
   1. Performance Requirements **Error! Bookmark not defined.**
   2. Safety Requirements **Error! Bookmark not defined.**
   3. Security Requirements **Error! Bookmark not defined.**
   4. Software Quality Attributes **Error! Bookmark not defined.**
2. **Other Requirements** Error! Bookmark not defined.

**Appendix A: Glossary** Error! Bookmark not defined.

**Appendix B: Analysis Models** Error! Bookmark not defined.

**Appendix C: To Be Determined List** Error! Bookmark not defined.

# Introduction

#### Product Scope

*This Software Requirements Specification provides a complete description of all the functions and constraints for Web-based Event Management System.*

#### Environment Characteristics

Web-based Event Management System is used to create a dedicated website to manage and organize events. The accessibility of the system should facilitate communication between users, client and hosts. It should allow client to create an event that includes the host, type of event, time, place, and eligibility. End users should be able to sign up to attend the event.

# Overall Description

#### Product Perspective

Our Collage have such a busy schedule that students and faculty members are always conscious about events and tasks. But this consciousness is not always enough to guarantee the participation in events so needed a mass customized website to manage their official events. Thus the service of our system comes into being.

#### Product Features

*<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>*

#### User Classes and Characteristics

1. Student: The largest user group of the system. The whole project become useless without them. They will participate in events.
2. Teacher: Faculty members are an essential part of this project. They will create events.They have direct influence over Event Management System.
3. Host: Host will manage the site and it has some rules and regulation to maintain our system. Everyone have to follow them strictly.

#### Operating Environment

Operating environment for the event management system is as listed below.

* + - distributed database
    - client/server system
    - Operating system: Windows, Linux.
    - database: mongoDB database
    - front-end: React
    - back-end: Express and Node

# Functional Requirments

#### Authentication

###### Register User :

Description :- You can register as a faculty member or student using this feature. You can enter a username and password and register as a result of that functionality.

Input :- Required Detail of User Output :- Successfully register

###### Login :

Description :- After registering, you can access the faculty or student functionality by logging in with your registered username and password.

Input :- Required Detail of User Output :- Successfully Login

###### Forgot Password

Description :- If a user uses the forgot password feature, they can change their password after providing a verification email.

Input :- Email address of registered User

Output :- Password will be change (Database also)

#### Event Management :-

###### Create Event :-

Description :- Only faculty users can create events, and in this sort of form, they must enter the event's title, description, your email, the recipient's email, and any attachments.

Input :- All the detail of required field

Output :- Event generated with unique event id.

###### Event Schedule :-

Description :- Only faculty members can schedule events, and in this type of form, they must provide the date and time of deliver mail , which is that students receive mail.

Input :- All the detail of required field Output :- Event Schedule with specified time

###### Update Event :-

Description :- Only faculty members are able to update events that have previously been created and they can also reschedule the event.

Input :- Update the event’s attributes

Output :- event regarding information updated.

###### View Event Template :-

Description :- Only student users are able to access that functionality, and they can view the application's event template and send emails with faculty approval.

Input :- -

Output :- Student user see the templates.

1. Scenario Based Model
   1. Use Case Scenario

Table 4.1: Use Case Scenario

|  |  |  |  |
| --- | --- | --- | --- |
| Level - 0 | Level - 1 | Level - 2 | Actors |
| Event Management System | Authentication | Sign Up | Student |
| Verification | Admin |
| Sign In | Admin, Faculty,Student |
| Sign Out | Admin, Faculty,Student |
| Change Password | Admin, Faculty,Student |
| Create User | Admin |
| Remove User | Admin |
| Event | Create event | Admin, Faculty,Student |
| Update event | Admin, Faculty,Student |
| Sub event | Admin, Faculty,Student |
| User activity |  | Admin, Faculty,Student |

## Use Case Description

We will elaborate use case scenario to use case diagram, description, activity diagram and swim - lane diagram. Here is the use case diagram of level-0 for Event Management System.

Student

Admin

Faculty

Use Case Diagram of EMS (Level-0)

This is the elaborated form of level-0 for Event Management System.

1.1

Authentication

1.2

Event

Student

1.3

Template

Admin

1.4

User Activity

Teacher

Use Case Diagram of EMS (Level-1)

### Authentication

We can further section Authentication system into sub-systems.

Student

1.1.1

Sign Up

1.1.2

Sign In

1.1.3

Sign Out

1.1.4

Admin

Change Password

1.1.5

Create User

1.1.6

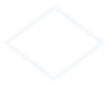
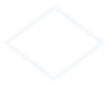
Remove User

Faculty

Use Case Diagram of Authentication (Level-1.1)

##### Sign up

no



Valid

yes

Verified

yes

Registration Complete

Send Verification Request

Enter Information

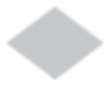
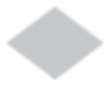
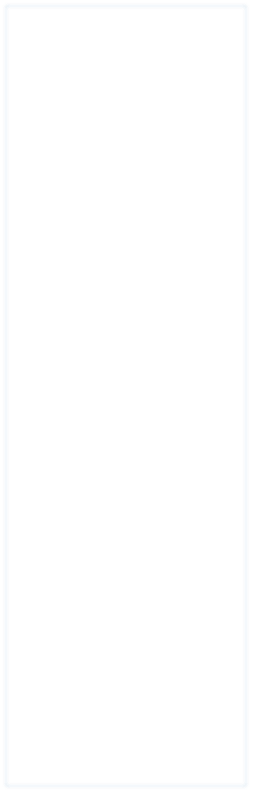
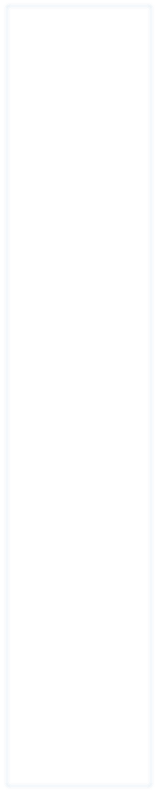
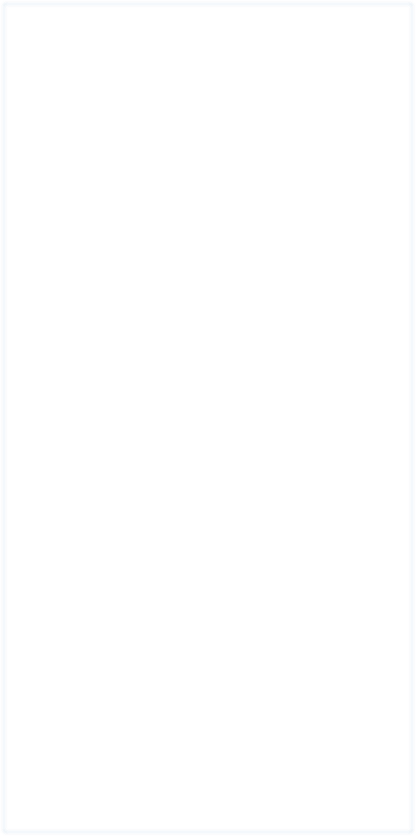
Click On Sign Up

no

Activity Diagram – Level 1.1.1 (Sign Up)

Yes

Swim Lane Diagram – Level 1.1.1 (Sign Up)



Valid ?

Verified ?

No

Registration Complete

Send Verification Request

Yes

Enter Information

No

Click On Sign Up

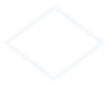
System

Student

Sign Up

##### Sign in

no



Valid

yes

Logged in

Enter Password

Enter Email

Click on Sign In

Activity Diagram –Level 1.1.2 (Sign in)

##### Sign out



Signed Out

Click on Sign Out

Activity Diagram – Level 1.1.3 (Sign out)

##### change password

Send an Email with link and set token

Request Password Reset

End

Does User Exist?

No

Yes

\

Yes

Is token and other identification info valid?

No

End

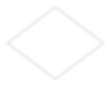
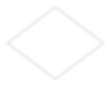
End

Reset Password

Activity Diagram – Level 1.1.4 (Change Password)

##### create user

no



Admin

yes

Select User Type

Enter Information

Valid

yes

User Created

Click on Create User

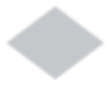
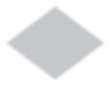
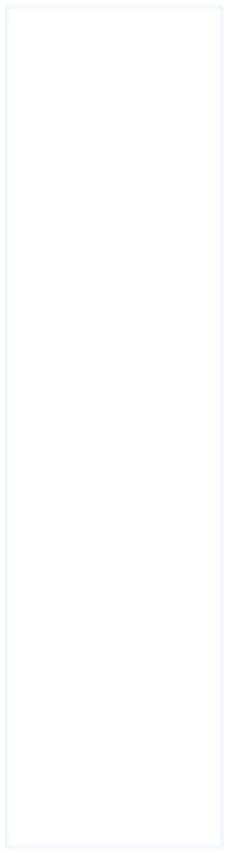
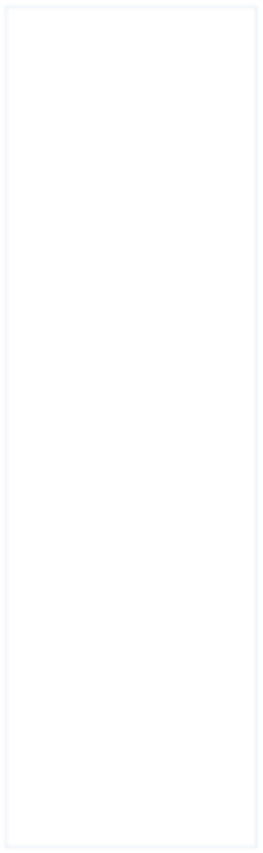
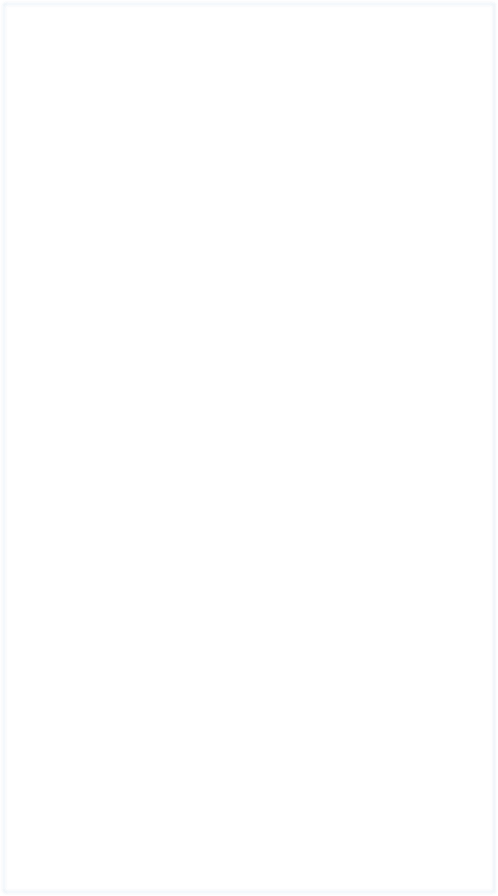
no

Activity Diagram – Level 1.1.5 (Create User)

No

No

Swim Lane Diagram – Level 1.1.5 (Create User)



Create user

Admin

System

Click on Create User

Admin ?

Yes

Select User Type

Enter Information

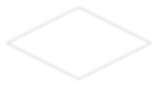
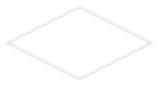
Valid ?

Yes

User Created

### Event

##### Create Event



yes

Logged In ?

no

Go to Log In Page

Valid ?

no

yes

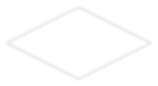
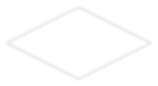
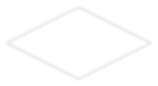
Upload Event Details

Click On New Event

Activity Diagram – Level 1.2.1 (Create Event)

### Sub event

##### add and delete Sub event



Logged In ?

yes

Owner ?

yes

no

Delete Sub Event

no

no

Valid ?

yes

Add Sub Event

View Event Details

Activity Diagram – Level 1.2.3 (Sub Event)

# External Interface

#### User Interfaces

* Front-end software: React, VS Code
* Back-end software: MongoDB Compass, NodeJS ,Postman

#### Hardware Interfaces

* Windows.
* A browser which supports React & JavaScript.

#### Software Interfaces

Following are the software used for the flight management online application.

#### Communications Interfaces



**Software used**

**Description**

Operating system

We have chosen Windows operating system for its best support and user-friendliness.

Database

To save the flight records, passengers records we have chosen MongoDB database.

To implement the project we have chosen MERN technology for its more interactive

VS code

support.

This project supports all types of web browsers. We are doing simple event registration for the user, etc.

# Other Nonfunctional Requirements

#### Performance Requirements

*<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>*

#### Safety Requirements

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed up log, up to the time of failure.

#### Security Requirements

Security systems need database storage just like many other applications. However, the special requirements of the security market mean that vendors must choose their database partner carefully.

#### Software Quality Attributes

* + - **AVAILABILITY:** The event should be available on the specified date and specified time as many users are doing advance reservations.
    - **CORRECTNESS:** The event should start from mention time and held on the mention destination.
    - **MAINTAINABILITY:** The host and faculty should maintain correct schedules of event.
    - **USABILITY:** The event schedules should satisfy a maximum number of users needs.

# Other Requirements

*<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>*

# Appendix A: Glossary

*<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>*

# Appendix B: Analysis Models

*<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams*.>

# Appendix C: To Be Determined List

*<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>*