**CHANGE HISTORY**

**PREFACE**

TABLE OF CONTENTS

[1. INTRODUCTION 6](#_Toc476651135)

[1.1 Purpose and Intended Audience 6](#_Toc476651136)

[1.2 Scope 6](#_Toc476651137)

[1.3 Definitions, Acronyms and Abbreviations 6](#_Toc476651138)

[1.4 References 6](#_Toc476651139)

[1.5 Overview of the Document 6](#_Toc476651140)

[2. OVERALL DESCRIPTION 6](#_Toc476651141)

[2.1 Product Perspectives and Context 6](#_Toc476651142)

[2.1.1 System Interfaces 6](#_Toc476651143)

[2.1.2 User Interfaces 6](#_Toc476651144)

[2.1.3 Hardware and Software Interfaces 6](#_Toc476651145)

[2.1.4 Communications Interfaces 6](#_Toc476651146)

[2.1.5 Memory Constraints 6](#_Toc476651147)

[2.1.6 Operational Context 6](#_Toc476651148)

[2.2 Product Functions 6](#_Toc476651149)

[2.3 User Characteristics 6](#_Toc476651150)

[2.4 General Development Constraints 6](#_Toc476651151)

[2.5 Project Assumptions and Dependencies 6](#_Toc476651152)

[3. SPECIFIC REQUIREMENTS 7](#_Toc476651153)

[3.1 External Interfaces 7](#_Toc476651154)

[3.1.1 Hardware and Software Interface Specifications 7](#_Toc476651155)

[3.1.2 Detailed Description of Inputs and Outputs 7](#_Toc476651156)

[3.2 Functional Requirements 7](#_Toc476651157)

[3.2.1 Use Case Diagram 7](#_Toc476651158)

[3.2.2 Use Case Report 7](#_Toc476651159)

[3.2.3 Activity Diagram 7](#_Toc476651160)

[3.3 Performance Requirements 7](#_Toc476651161)

[3.4 Database Requirements 7](#_Toc476651162)

[3.4.1 ERD 7](#_Toc476651163)

[3.4.2 Data Dictionary 7](#_Toc476651164)

[3.5 Object/Analysis Model 7](#_Toc476651165)

[3.5.1 Per Use Case Activity Diagram 7](#_Toc476651166)

[3.5.2 Per Use Case Class Diagram 7](#_Toc476651167)

[3.5.3 Per Use Case Communication/Sequence Diagram 7](#_Toc476651168)

[3.6 Design Models 7](#_Toc476651169)

[3.6.1 Per Use Case Activity Diagram 7](#_Toc476651170)

[3.6.2 Per Use Case Class Diagram 7](#_Toc476651171)

[3.6.3 Per Use Case Communication/Sequence Diagram 7](#_Toc476651172)

[3.7 System Models 7](#_Toc476651173)

[3.7.1 Component Diagram 7](#_Toc476651174)

[3.7.2 Deployment Diagram 7](#_Toc476651175)

[3.8 Software System Attributes 7](#_Toc476651176)

[4. SUPPORTING INFORMATION 8](#_Toc476651177)

[4.1 Appendices 8](#_Toc476651178)

[4.2 Index 8](#_Toc476651179)

# INTRODUCTION

## Purpose and Intended Audience

## Scope

## Definitions, Acronyms and Abbreviations

## References

## Overview of the Document

# OVERALL DESCRIPTION

## Product Perspectives and Context

### System Interfaces

### User Interfaces

### Hardware and Software Interfaces

### Communications Interfaces

### Memory Constraints

### Operational Context

## Product Functions

## User Characteristics

## General Development Constraints

## Project Assumptions and Dependencies

# SPECIFIC REQUIREMENTS

## External Interfaces

### Hardware and Software Interface Specifications

Not yet identified. To be addressed after system testing.

### Detailed Description of Inputs and Outputs

1. LOG-IN

The users input their identification number and password and then click the log-in button. The system processes and identifies the validity of the details entered and if the user is a valid user, the system will allow the user to use it based on the user’s privileges.

1. SIGN-UP

The users input their basic information such as their identification number which can be the student or employee id depending on their designation, first name, last name, contact number, email, and account password. The system processes and identifies the validity of the details entered and record it to the database. After the registration, the system will redirect the user to the home page.

1. VIEW UPCOMING EVENTS

The users will be able to view the upcoming events in list view in the home page of the system both if you are signed in or event not.

1. CREATE EVENT

The user will act as an organizer and input the event’s primary details such as title, short description, start and end time or date and the desired location. These details we’re being submitted and checked by the system to verify if the desired location is available by the said time and date.

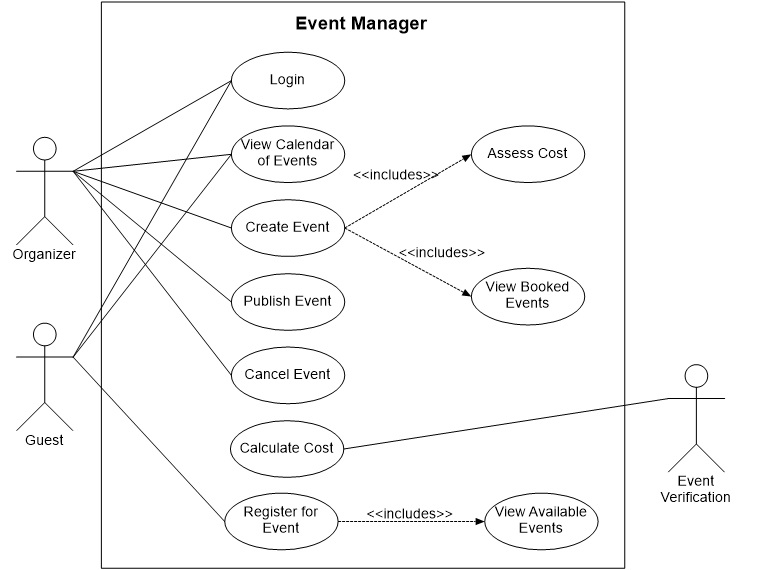
1. JOIN EVENT

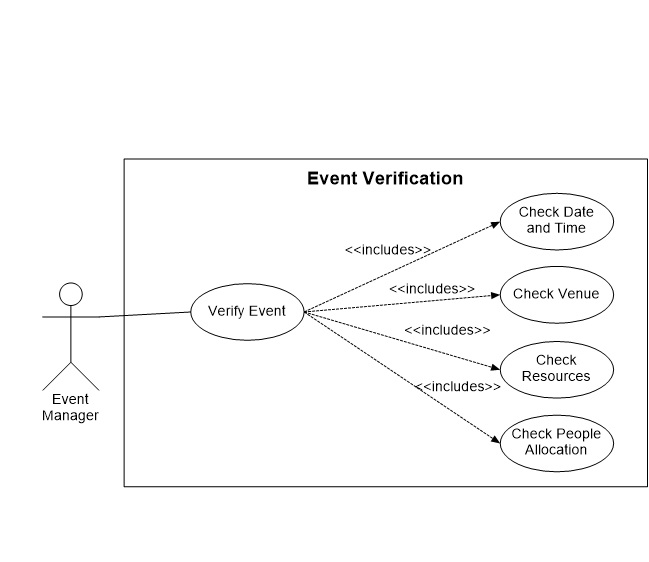
The user will be able view the complete details of an event and has the choice if he or she will join or not this event. If the user chooses to join the event, then the system will list the user as a guest.

## Functional Requirements

### Use Case Diagram

.





### Use Case Report

Table I: Use-case Description of Login.

|  |  |
| --- | --- |
| Use Case | **Login** |
| Short Description | This allows to validate user login in order to use the system. |
| Actors | Organizer  User |
| Pre-Conditions | User must have a registered account to the system. |
| Post Conditions | User must be logged in. |
| Main Flow | 1. Insert the identification id in the designated text field. 2. Insert the password in the designated password field. 3. Click **Log In** button to access the system. |
| Alternative Flows | * 1. Login not successful   Incorrect email or password  Account doesn’t exist |

Table II: Use-case Description of View Calendar of Events

|  |  |
| --- | --- |
| Use Case | **View Calendar of Events** |
| Short Description | This allows the user to view the old and upcoming events in calendar view. |
| Actors | Organizer  Guest |
| Pre-Conditions | User is either logged in or not. |
| Post Conditions | Old and upcoming events are viewed. |
| Main Flow | 1. User will navigate the calendar view. 2. The system will query from the server all the events within the calendar view. 3. The system will display the queried data on the calendar. |
| Alternative Flows | 2.1 Query not successful.  Slow internet connection. |

Table III: Use-case Description of Create Event.

|  |  |
| --- | --- |
| Use Case | **Create Event** |
| Short Description | This allow the organizer to create an event by filling up the necessary information such as the name of the event, desired date, time, and venue, as well as the organizer’s information. |
| Actors | Organizer |
| Pre-Conditions | User must have and account registered to the system. |
| Post Conditions | Event creation section must be viewed. |
| Main Flow | 1. System loads the section. 2. System displays event information fill up form. 3. Organizer enters all required and supporting information regarding the event. 4. Organizer clicks the **Submit** button. 5. System prompts user to finalize action. 6. Organizer decides that the action to submit is final. 7. Includes (Validate entry) 8. Event that has been created will be shortlisted. 9. System prompts user about the availability of date and time, venue, resources and people allocation. 10. Organizer clicks **Assess Cost** button. 11. System displays summary of event expenses. |
| Alternative Flows | 4.1 User press **Cancel** without submitting the event information.  4.2 System displays warning message to know if the user really. intends to cancel the action.  4.3 User doesn’t intend to cancel and return to the form.  4.4 System returns to the form.  5.1 User chooses to correct entered information.  5.2 System returns to the form.  6.1 Entry is invalid.  6.2 System disregards the entry. |

Table IV: Use-case Description of Publish Event.

|  |  |
| --- | --- |
| Use Case | **Publish Event** |
| Short Description | This allows the organizer to create notification or advisory, thus publishing the necessary details about a particular event which is already registered. |
| Actors | Organizer |
| Pre-Conditions | There is event information submitted by the organizer.  User must have an account registered to the system. |
| Post Conditions | Notification regarding an upcoming event must be sent to other users/guests. |
| Main Flow | 1. Organizer clicks the **Event Picker** button. 2. System prompts the information about the booked event. 3. Organizer clicks the **Publish** button. |
| Alternative Flows | 3.1 System sends notification to the users/guests about the event published. |

Table V: Use-case Description of Cancel Event.

|  |  |
| --- | --- |
| Use Case | **Cancel Event** |
| Short Description | This allows the organizer to create a notification or advisory, thus publishing the necessary details about a particular event which is already registered. |
| Actors | Organizer |
| Pre-Conditions | There is an event information submitted by the organizer.  User must have an account registered to the system. |
| Post Conditions | Notification regarding an event cancellation must be sent to other users/guests. |
| Main Flow | 1. Organizer clicks the **Event Picker** button. 2. System prompts the information about the booked event. 3. Organizer clicks the **Cancel** button. 4. System removes the event from the database. |
| Alternative Flows | 3.1 System sends notification to other users/guests about the event cancelled. |

Table VI: Use-case Description of Calculate Cost.

|  |  |
| --- | --- |
| Use Case | **Calculate Cost** |
| Short Description | This allows the even manages to compute for the total cost of a particular event depending on venue rental, personnel pay, energy fee and other misc. fee. |
| Actors | Event Manager |
| Pre-Conditions | There is event information submitted by the organizer.  User must have an account registered to the system. |
| Post Conditions | Event cost must be computed. |
| Main Flow | 1. System checks the individual cost. 2. System multiplies the cost per hour. 3. System computes for total cost. |
| Alternative Flows | * 1. System sum up all individual cost.   2. System checks the time duration. |

Table VII: Use-case Description of Register for Event.

|  |  |
| --- | --- |
| Use Case | **Register for Event** |
| Short Description | This allows the guest to register for an event. |
| Actors | Guest |
| Pre-Conditions | There is published event information submitted by an organizer.  User must have an account registered to the system. |
| Post Conditions | Guest must be able to register for an event and ticket must be issued to the guest. |
| Main Flow | 1. Guest clicks the **Event Picker** button. 2. System prompts the information about the booked event. 3. Guest fills up information about him. 4. Guest clicks the **Register** button. 5. System will list the user as expected guest to the said event. |
| Alternative Flows |  |

Table VIII: Use-case Description of Verify Event.

|  |  |
| --- | --- |
| Use Case | **Verify Event** |
| Short Description | This allows the event manager to check the availability of date and time, venue, resources and people allocation depending on the event information inputted by the organizer. |
| Actors | Event Manager |
| Pre-Conditions | There is event information submitted by the organizer.  User must have an account registered to the system. |
| Post Conditions | A message must be prompt to the organizer regarding the event verification and must be saved in the database afterwards. |
| Main Flow | 1. System checks the availability. 2. System prompts a message to the organizer that the event has been successfully registered. 3. System prompts a message to the organizer that create another event having the schedule not yet taken. |
| Alternative Flows | 2.1 System displays a success message.  3.1 System displays warning message. |

### Activity Diagram

## Performance Requirements

The system should be able to handle multiple users and can perform multiple tasks simultaneously and should have a fast response mechanism in evey instance of use.

## Database Requirements

### ERD

### Data Dictionary

## Object/Analysis Model

### Per Use Case Activity Diagram

### Per Use Case Class Diagram

### Per Use Case Communication/Sequence Diagram

## Design Models

### Per Use Case Activity Diagram

### Per Use Case Class Diagram

### Per Use Case Communication/Sequence Diagram

## System Models

### Component Diagram

### Deployment Diagram

## Software System Attributes

# SUPPORTING INFORMATION

## Appendices

## Index