Halo Requirement Specification
Version 1.0
03/11/2022

Table of Contents

1. Executive Summary

1.1 Project overview

This project helps the user to organize various events and execute different operations on the event data, A creator can make new events and a superuser can delete the events, if there is a expired event or an inappropriate event

1.2 Purpose and Scope of This Specification

this project is intended for various event organizers that will enable them to search for various events and add new events, halo doesn't provide event history or previous event details, halo doesn't provide online support where people can access its functions on a website

2. Product/Service Description

2.1 User Characteristics

There are two types of users that will interact with this system referred as viewers, creators and superuser, and they can use the system for quite different purposes which are specified below.

- Creator functions:
  - creating new events
  - viewing events
- Viewer functions:
  - viewing the event
- Superuser functions:
  - viewing the event
  - deleting the events

2.2 Assumptions

- user is required to know how to start the project on Dev-c++
- user can provide input to the program

2.3 constrains

- The project shall be developed as a console application
- There will be no use of databases, only file systems

2.4 dependencies

- The project utilizes c++ header files to use the file handling functionality and various input and output functions without them the project can break

- Project needs GCC compiler for executing the code without which code will not be executed

3. Requirements

3.1 Functional requirements

3.1.1 Login system

Creator has to login into the console application in order to create new events, this will help prevent creation of invalid events by unknown individuals. User that tries to login has to provide his/her username and password in order to login to the creation system

3.1.2 Viewing Events

All the users must be able to view all the events that are listed by the creators, event page will include the details about the event, date of the event, the person or group who is organizing the event

3.1.3 Deleting Events

Super user must be able to delete the events made by creators, all the events can be deleted by the superuser

3.2 user interface requirements

- simple interface
- Easy navigation
- Purposeful layout

3.3 Non-functional requirements

3.3.1 Performance

The product shall take less than 500ms to start and load the program which will not depend on the internet connection as all the files are locally stored which ends its dependency on the internet connection

3.3.2 Security

The users can't make a new event without entering creator's login details and users can't delete user without entering the super user login details which helps to prevent unnecessary deletion of the events.

3.3.3 Reliability & availability

The files will not get deleted and will be accessible to the program, in case files are deleted new files will be created by the program, the program will only get effected if the user's system gets wiped out which will delete all the files required for the our program to work

3.3.4 Availability

Program is operational at any time of the day on particular computer where the program is made , the events are only accessible locally on the end device where program is set up.

3.3.4 Capacity

Number of events that can be stored in the program depends on the storage capacity of the system in which the code is set up, each event file is around 1KB then the events that can be stored in a 1TB system are 1,000,000,000 +

3.3.5 Maintainability

Code is divided into different classes and its members which makes the program to be easily understandable for the new developers and in case there is any error the bug can be easily located and fixed, in order to add new features, new members functions can be added to the existing classes

3.3.6 Usability

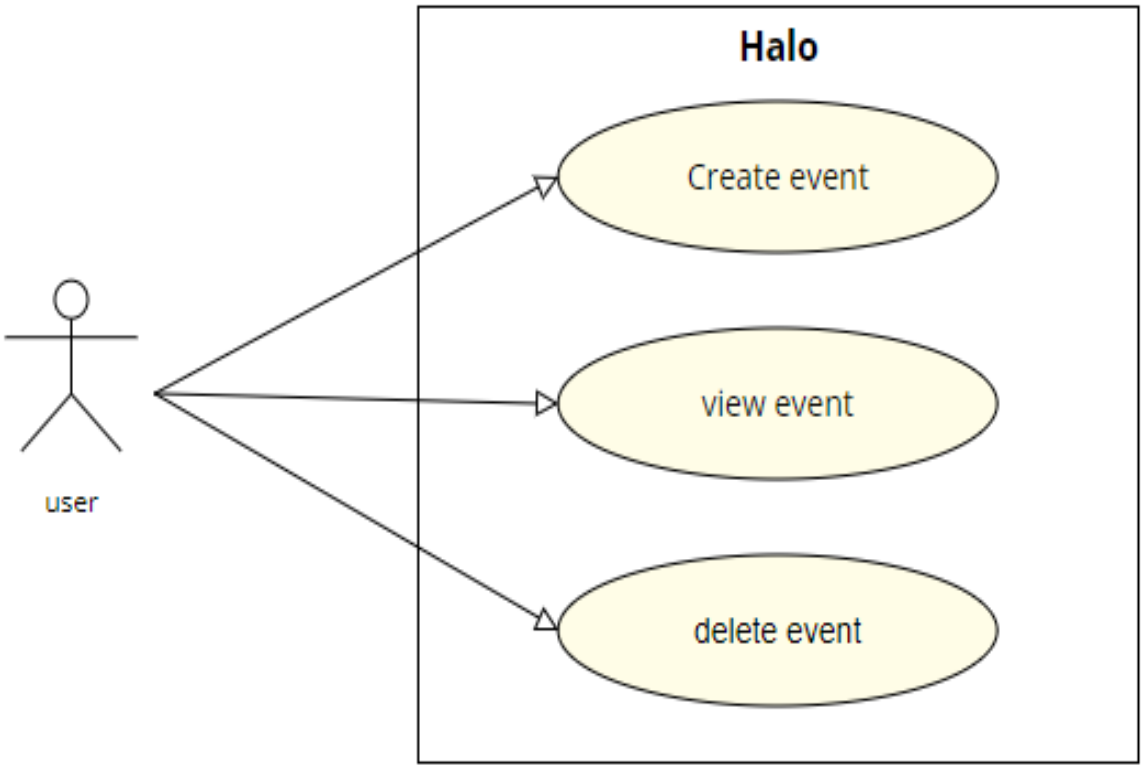
The system provides console user interface for anyone who wants to create, delete, view all the events. User doesn't need to go through the system files or configure the system in order to run the program

3.3.7 Testability

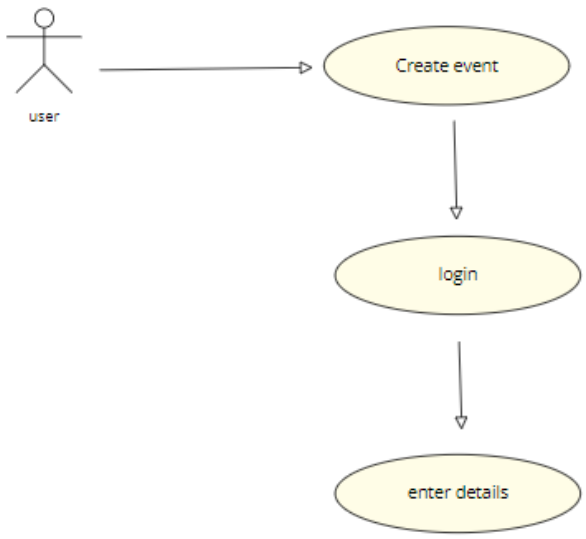
As the code is divided into classes and their member functions the testing process is easy because we simply have to check every class and its working opposed to checking different functions which are spread across the codebase

4. Use Cases

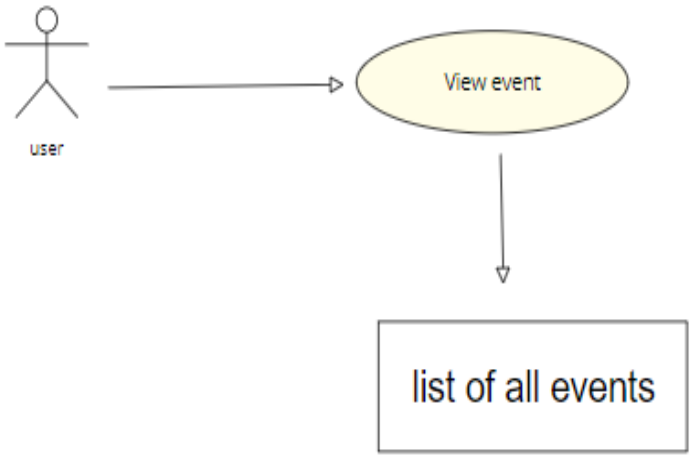
4.1 Opening Screen



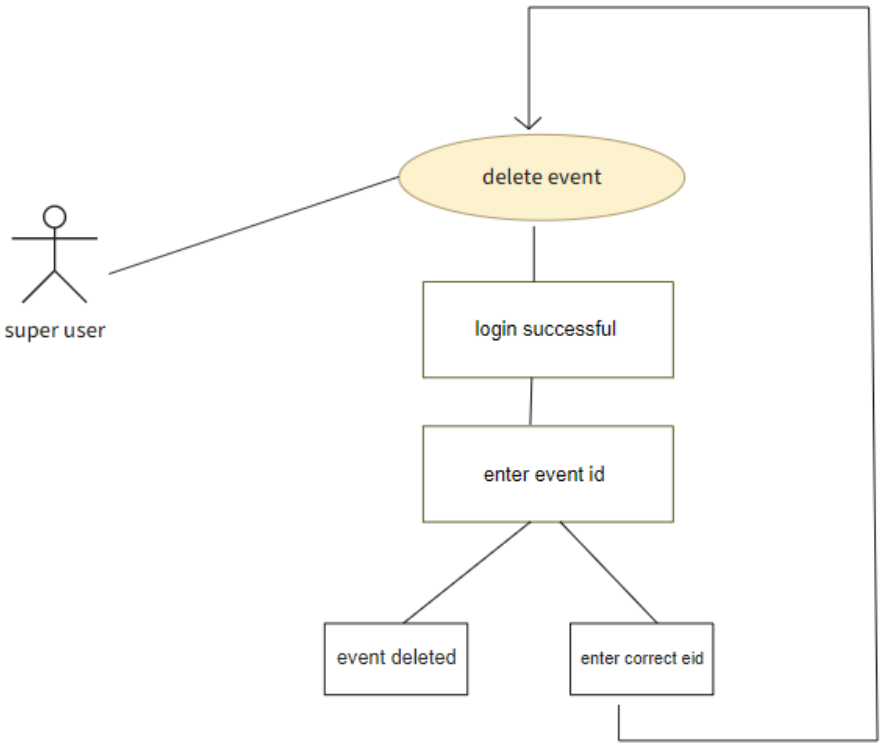
4.2 Event Creation Page



4.3 Event Viewing Page



4.4 Event deletion Page



Appendix A. References

- https://cplusplus.com
- C++ Tutorial (w3schools.com)
- File Handling through C++ Classes - GeeksforGeeks