Group Project Documentation   
Basic Cross-Platform Application Programming With .NET

*Event Management System*

**Prepared by Group 9**

***Nguyễn Tuấn Khải***

***Nguyễn Dũng***

***Nguyễn Đức Long***

**Ho Chi Minh City, 2022**

**Table of Contents**

**Table of Contents ii**

**Revision History ii**

**1.** **Project Introduction 1**

1.1 Product Perspective 1

1.2 User Classes and Characteristics 1

**2.** **Database Design 1**

**3.** **System Architecture 1**

**4.** **Implementation 2**

4.1. Deployment Considerations [2](#_heading=h.2s8eyo1)

4.2. Screenshots and explanations [2](#_heading=h.17dp8vu)

**5.** **References 2**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| *Nguyen Tuan Khai* | *01 February 2022* | *Initial draft* | *1.0 draft 1* |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Project Introduction

## Product Perspective

*This event management system will allow users and organizers to access and manage all aspects of an event, including registration, (marketing, engagement, integrations,) physical planning and preparation, reporting and analytics, and more.*

*Product Features*

*The product allows user:*

* + - * *Update database with new events and clients;*
      * *Create query and get all information about events and clients;*
      * *Manage user system;*
      * *Keep track of a user`s history events;*
      * *Deduct cancelled event;*

## User Classes and Characteristics

There are 3 user classes in developing system: Admin, Host and Guest.

Admin is main class in our system, because it can access to needed details about events, clients and histories. For example, admin can get to know what client take what event, and what client host what event, what client be in what group, what client post what event. Moreover, admin can edit, delete and create events in the system with appropriate message sent to clients. Admin can add, update or remove role host (member type 2) for users in the system. Admin can add, update or remove group for users in the system.

Member type 1 (Guest) can get access to partial information about events and histories. For example, guest can get to know the profile, invitation, ticket/event histories, group and role. Guest can edit its own profile, response to invitation, remove or request to be in specific group, remove or request to be upgraded to host role (member type 2), post comment in the Event Post section.

Member type 2 (Host) can get access to partial information and higher privilege about events, histories, analytics and managements. For example, host can get to know the profile, invitation list and invitation response, ticket/event histories, attendance, attendance group and role. Host can edit its own profile, create events and manage allowed event group, create invitations and track responses, manage attendance through tickets, modify events information and post comment in the Event Post section.

Member type 1 account (default role) can be created through registration or added by admin or host. Member type 2 account can be achieved through upgrade review from member type 1 or being added by admin. Admin cannot be created or added through any endpoints of the system, admin is integrated in the database or can only be modified by the system admin of the database.

# Database Design

Diagram

Description automatically generated

# System Architecture

The system will operate in a server with IIS Application pool ASP.Net Core version, MS SQL 2022 Express version with 100mb of space storage and 30mb database storage. The users can access the application anywhere on the Internet, best performance in US and EU where datacenter of the company located.

The application will be designed with the priority from admin to member type 2 and member type 1 respectively. There will be a models layer to help communicate with database, services layer to help validations and contain logical business functions and a views/razor pages layer where authenticated and authorized users can access to perform actions of the system.

The following technologies/services can be implemented to help the app more useable: firebase authentication, JWT authentication & authorization.

# Implementation

## Deployment Considerations

The deployment can be done through building a portable version of the app and host it in IIS Application pool server ASP.NET Core or using Docker image container to host it with the versions specified in JSON files.   
User can access anywhere around the world, but the time is consistent to only 1 time zone, further parallel time zone for events can be considered to be developed in the future. With Docker image, the app is fully scalable in the future. Considering Soome.com hosting, data storage can be upgraded to fulfill the need of uses for the system.

## Screenshots and explanations

*<Screen flow | Dialog Map>*

*<The screenshots and explanations>*

# References