Software Design Specification

for

EventBook

Prepared by

Group Members:

Name	ID
Md. Fahad Muntasir	162 0683 642
Rubayet Zaman	162 0684 042
M.J. Darad	162 0520 042
Sams Uddin Ahamed	162 0129 042

Submitted to:

Dr Nabeel Mohammed

Submission date: 24.02.2019

Table of Contents

1. Introduction

1.1 Purpose :

This Software Design Specification document aims to represent the detailed design of our project named “EventBook” where organizers can create an event and the participants can enroll at an specific event. The client may see the events happening nearby with the maps and can also see notifications of the events if s/he already enrolled at that event,
This document will contain:

- Architectural Pattern used in this project.
- Database Schema
- Class Diagrams
- Detailed description of each class
    - Responsibility of each class
    - Methods with parameters and values
- Weekly breakdown of the tasks.

1.2 Scope of the Project :

This project will bring all the participants and event organizers under the same roof. Since there’s no dedicated platform for the event planners to host an event this piece of software will play a vital role. Also participants looking for their desired event or competition will be able to join those. This project will open a door for the people with various skill to advertise their own events and earn money from those.

1.3 Definitions, Acronyms and Abbreviations :

- User - Anyone using the software.
- Administrator - Person having the full access into the system.
- SDS - System Design Specification.
- UML Diagram - Unified Modeling Language Diagram.
- Home page- The main representing page the user will see at first.
- Registration Page– Where user can register.
- Login Page– Where user can log in.

- Profile- The personal page with all information for each logged in users.

1.4 References :

- https://laravel.com/
- https://www.android.com/
- https://www.mysql.com/
- https://developer.mozilla.org/en-US/
- https://www.w3schools.com/

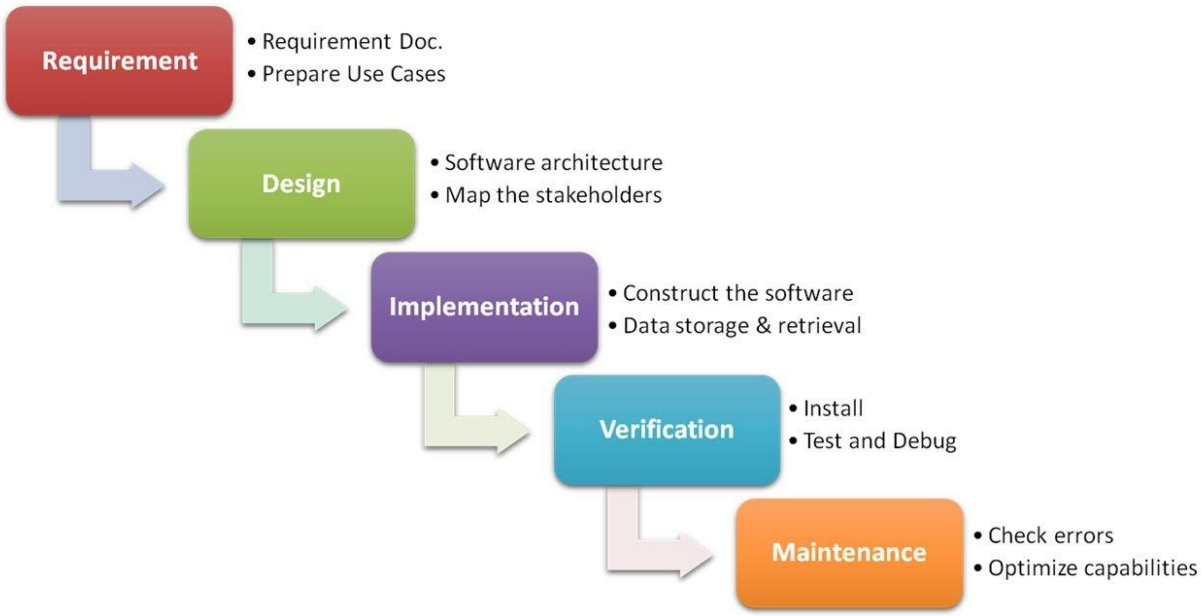
1.5 System Overview :

2. Architectural Pattern Used :

2.1 Approach :

The approach we have picked for this piece of software is the Waterfall Model where there are some steps like requirement analysis, design and implementation, unit testing and maintenance.

2.2 Diagram of the Approach :



2.3 Design :

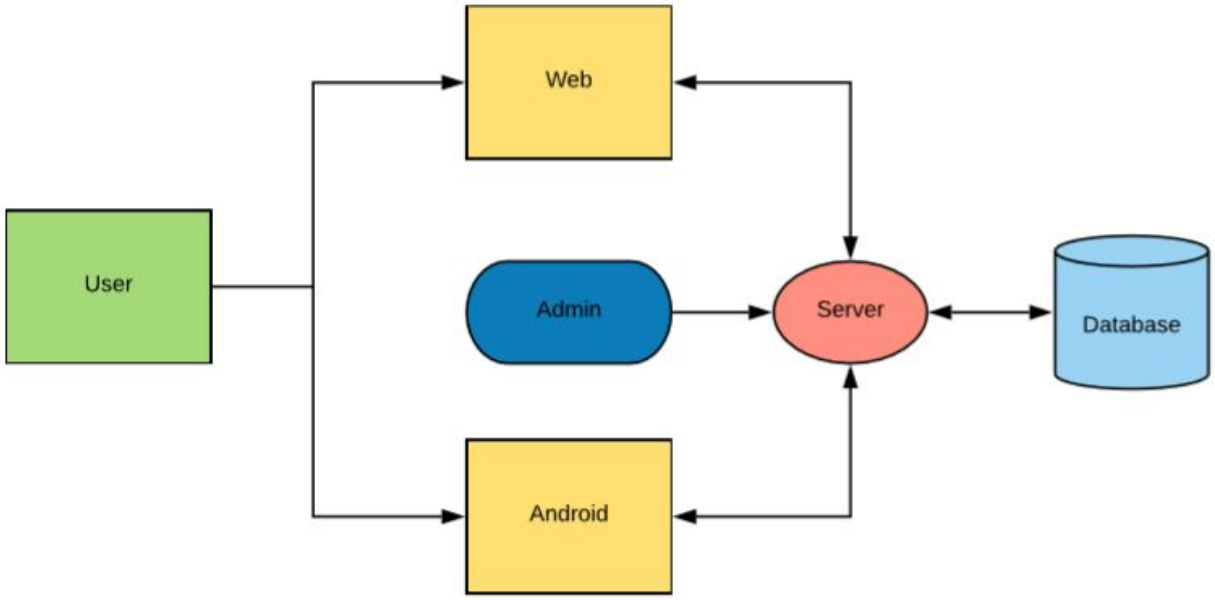


Figure: Design Diagram

2.4 Design Pattern :

For this project we have selected MVC framework which is Model-View-Controller framework. The reason behind picking this one is, MVC architecture allows application logic to be separated from user interface.

1. Model controls the data flow and the behaviour of the system. It handles request for the information and respond to it.
2. View works with the user interface.
3. After receiving the user input from the controller it makes call to the model object.

3. Class Design:

3.1 Class Diagram:

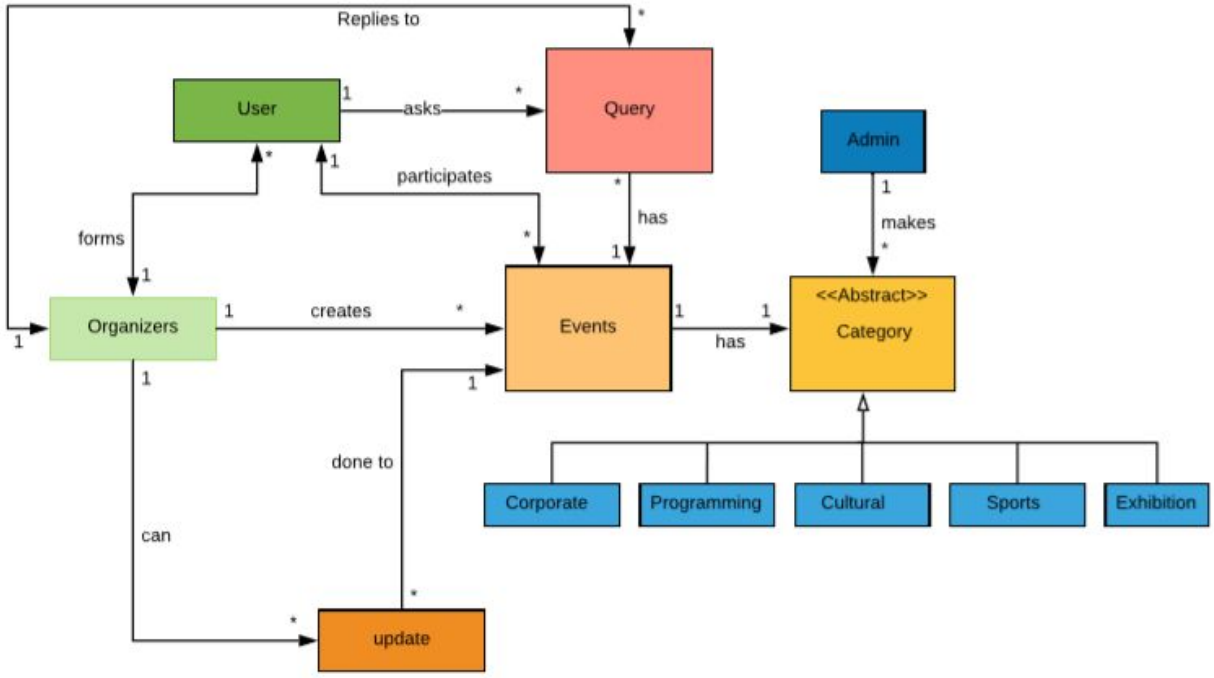


Fig: Class Diagram

3.2 Class Details:

User: This class will contain all the necessary information about an user after the registration and the stored data will be cross-checked during log in.
The attributes of this class is: userID, userName, email, password, gender, birthdate, address,

Organizers: User can create a group of users or can act alone to host an event.Organizing team can give update about the event as well as change the event details.

The attributes of this class is: userName(from User Class), organizerID, organizerName,

Events: Organizers host events and participants can participate in the events.The category of the event is also inherited in this class.
The attributes of this class is: organizerID(from Organizer class), userName(from User Class), eventName, eventCreatedTime, eventDate,eventPlace, eventCategory, eventDiscription, eventID.

Query: Event related questions are asked here by the participants and the organizers see to those queries.
The attributes of this class is: organizerID(from Organizer class), userName(from User Class),eventID(from Event class), queryID, queryQuestion, queryAnswers, queryTime.

Update: Organizers can update event informations if needed. If organizers needed to change event place or time or anything else, they can.
The attributes of this class is: organizerID(from Organizer class), eventName, eventCreatedTime, eventDate,eventPlace, eventCategory, eventDiscription, eventID.

Admin: Admin creates category table where the events get sorted according to their given category determined by the user.

4. Database Pattern

4.1 ER Diagram with Schema:

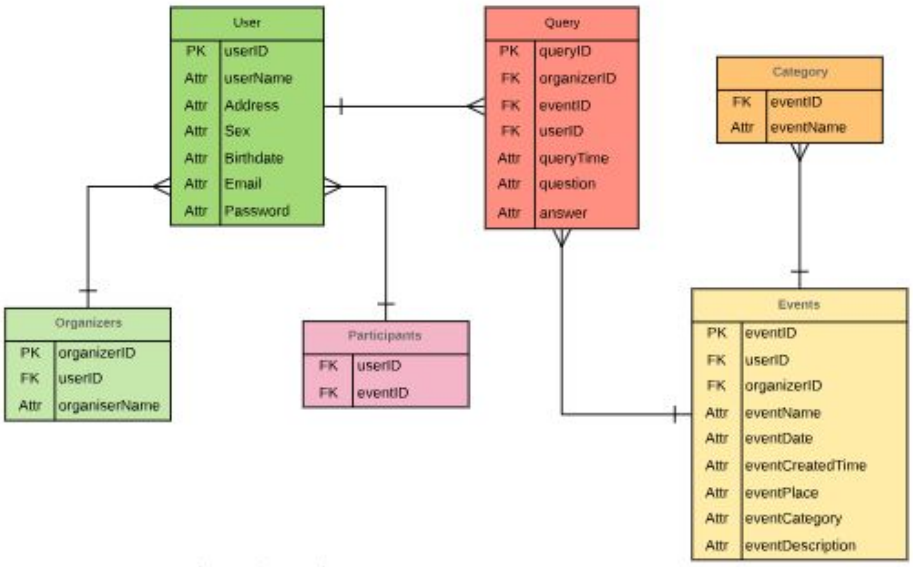


Fig: Database Schema

4.2 Description of the Diagram:

User:
The attributes of this class is: userID(PK), userName, email, password, Sex,birthdate,address,

Organizers:
The attributes of this class is: userName(from User Class-FK), organizerID(PK), organizerName,

Events:
The attributes of this class is:
eventID(PK), organizerID(from Organizer class-FK), userName(from User Class-FK), eventName, eventCreatedTime, eventDate,eventPlace, eventCategory, eventDiscription.

Query: The attributes of this class is:
organizerID(from Organizer class-FK), userName(from User Class-FK),eventID(from Event class-FK), queryID(PK), queryQuestion, queryAnswers, queryTime, question, answer.

Participants: The attribute of this class is :
userID(from user class-FK), eventID(from event class- FK).

5. List of APIs :

The following APIs we are going to use for this project :

1. FB Login API.
2. Google Login API
3. REST API.
4. Google MAPS API..