

Event Registration System

| | |
|-----------------|----------------|
| Subject | CL-VIII |
| Batch | S9 |
| Group ID | 12 |

| Name | Roll No. | E mail | Phone no. |
|------------------|-----------------|--|------------------|
| Sakshi Tantak | 43159 | sakshi99tantak@gmail.com | 9403878063 |
| Rucha Shinde | 43164 | hrucha.s144@gmail.com | 9657041410 |
| Shreeya Thigale | 43170 | shriyathigale20@gmail.com | 9970788249 |
| Riddhi Toshniwal | 43171 | riddhi.toshniwal@gmail.com | 8983487710 |

Problem Statement

We propose an **“Event Registration System”** that can manage all the events that are conducted in college/University. This system automatically generates certificates and issues it via mails. It can efficiently store, maintain, and retrieve data from its database. The system provides latest notifications to the user and intends on saving the users’ time by easing out the process of registering for events and staying in loop with the further process regarding these events. The application maintains data in a ***centralized way*** which is available to all the ***event managers***. It is very easy to manage ***historical data*** in databases. Participants can register for any happening event from anywhere. Event managers can keep records of participants and the system can easily inform them by messages and emails. Thus the system shall simplify this task and make it more efficient. It intends on automating the process of registration.

Operation of the proposed system depends on its various users. These various user-types are mentioned below.

1. Admin
2. Manager
3. Normal User (Participants)/Volunteer

As no papers required so it reduces cost. It would be beneficial because only one time development efforts are required and no special skills or training required to run the system.

The proposed system is better than ordinary registration process and management because it is faster than older systems, the database keeps records and it is an onclick time saving system.

The system comprises of the following modules:

Registration : Participants can register online for any event.

Event : Event manager can generate events using this module. They can edit, delete the information of any generated event.

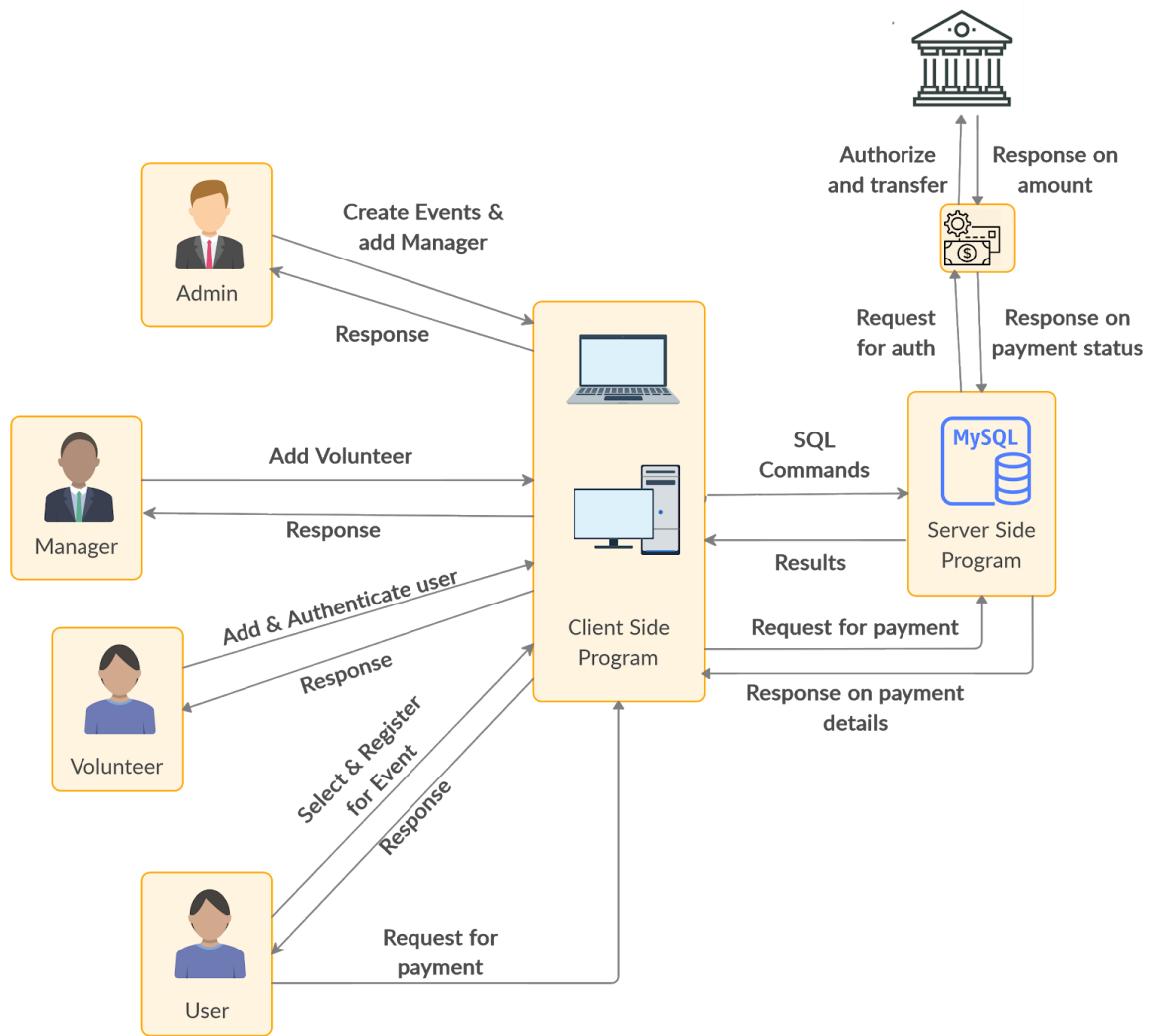
Certificate : System can generate certificates automatically and mail it to the particular participants.

Notification : Participants can get notifications about event timings, venue or any other updates.

User Module : Users can register for an event and can easily check event details, timings and can contact the event manager easily.

Volunteer Module : Volunteer can view the details of his/her event participants. Volunteers have to update the winner of his/her event in the system.

Admin Module : Database administrator has all the rights. He can add, edit, delete any information in the system. He can add new events, event managers, Volunteers as well as update them.



Use Case Scenarios

In the flow of the system, a typical use case scenario could be, Admin will make managers for different events and allocate Login ID. Thereafter Normal User shall enter his details i.e., Name, Enrollment Number, Course, Branch, College and select events. Then the manager will check user entries and allot registration id for authentication purpose. Managers can also change the date, time, venue of the events. Here, the manager is responsible for checking the team size and details and to restrict unauthorized people or wrong entries. After these all processes, Admin will check all details of the user, work of the manager and Admin can also make any changes at any stage.

Considering the use case of a user, the flow shall be as follows. The user shall input his username and password, which shall be checked as to whether he is registered in the database as a normal user. If the user is found in the database, he/she shall be directed to login. Once he/she enters the login details, i.e., username and password, the details shall be authenticated and checked. If found valid, he shall be directed to the profile to select events. Once the user selects events from the list of available events, he shall be redirected to the payment process. The payment shall display the event details and amount required to be paid, ask for confirmation to proceed and redirect the user to the payment gateway, if affirmative. In lieu of a successful payment, the user shall be brought to the logout page. In an alternate case, if the user is unable to login due to password memory, he can choose to reset his password using the Forgot Password link. If he/she clicks on the link, the user shall be requested to enter the email id to reset password. A reset password link shall be sent to the entered email id.

Otherwise, if the user is not found to be a registered user, he/she shall be directed to the new user registration. The registration process shall require the user to enter his/her details like username, password. After successful registration and authentication by the admin/volunteer/manager and updation of the database, the user shall be directed to the login page, after which, the same process as mentioned earlier will be resumed.

We also consider the use case of adding a manager. A manager can be added by the admin. The admin shall send a request to add a manager to the system. He/she shall be required to enter the event required and the details of the manager, i.e., his

id, username and password, and these details will be checked across the field constraints of the database. Upon successful verification, the manager with specified credentials shall be added to the database and the admin shall be sent a response of confirmation. The manager can then login with his allotted login id, username and password.

Following are the non-functional requirements of the system :

Performance Requirements:

The system needs to be reliable, generate appropriate error messages for transactions and load the web pages within seconds and also prevent system crash in the event of high traffic on the app/site.

Safety Requirements:

The Users should be authenticated based on their Google account or phone numbers. The database should be backed up well and properly at multiple sites to avoid loss of data in case of failures and prevent the loss of data due to clashing or concurrent transactions that can cause system failures.

Features:

The system needs to be integrated with the payment gateway to support in-app payment for events.