**Student Activity Management System**

**Team TEN-acious (Group 10):**

Rajia Shareen

Guna Chandrika Paturi

Malcolm Lipscomb

Matthew Goodman

Vinutna Gannu

Devireddy Venkata Sai Krishna Reddy

**Report Date:**

**04-30-2017**

**Index**

|  |  |
| --- | --- |
| **Page No.** | **Contents** |
| 3 | Project Requirements |
| 4 | Scope of Work |
| 5 | Business Rules |
| 6 | Constraints |
| 7 | EER Diagram |
| 8 | EERD Narrative and Features |
| 9 | ERD Relational Mapping |
| 10 | Tables |
| 13 | Implementation |
| 17 | Screenshots |
| 24 | Conclusion |
| 24 | Future Scope |
| 25 | Team Members Information and Roles |
| 26 | Project Plan |

**Project Requirements**

The project will allow any generic university to keep track of the attendance record of students to academically oriented learning activities. This application will keep track of what events are happening at the school, as well as who attends them. Generally, event attendance has been estimated with the help of sign-in sheet , or just not handled at all in some cases. But with the database keeping track of attendance, popular events can be repeated, poorly attended ones can be updated or removed, and required participation can easily be tracked by professors.

If maintained, this will open up an avenue to streamline the process of scheduling events, getting that information out to students, and determining whether or not the event was a success.

**Scope of Work**

1. Tracking the number of students that attended any academic event that occurred on campus.
2. Keep track of students by a particular major and by the event.
3. Store and list each event a student has attended.
4. Advisor’s can view the list of students and number of students attended for an event.
5. Store and display a student’s information for an advisor if necessary.

**Business Rules**

1.Persons (generalization – specialization). This will end up being a parent table for Persons and child tables for Students and Faculty. Students have Major Code, Minor Code, Year and OnCampus Y or N (whether they live on campus or not). Faculty have Highest Degree, Department, and Position Title.

2. Students have one and only one Advisor (Faculty). Advisors (Faculty) have one to many students.

3.Events have an ID, a Name, description, a start date and time, an end date and time, Location, Department, and a Category (Speaker, Club Meeting, Athletic Event, Play, Career event, etc.).

4.Students attend from zero to many events. Events have from zero to many students attend.

5.Departments have courses. A course can be a Major or Minor and is associated with one and only one department. A Department can have one to many courses.

6.Faculty are assigned to one and only one Department. A department has from one to many faculty members.

7.An event can be listed in one and only one category. A category can have one to many events.

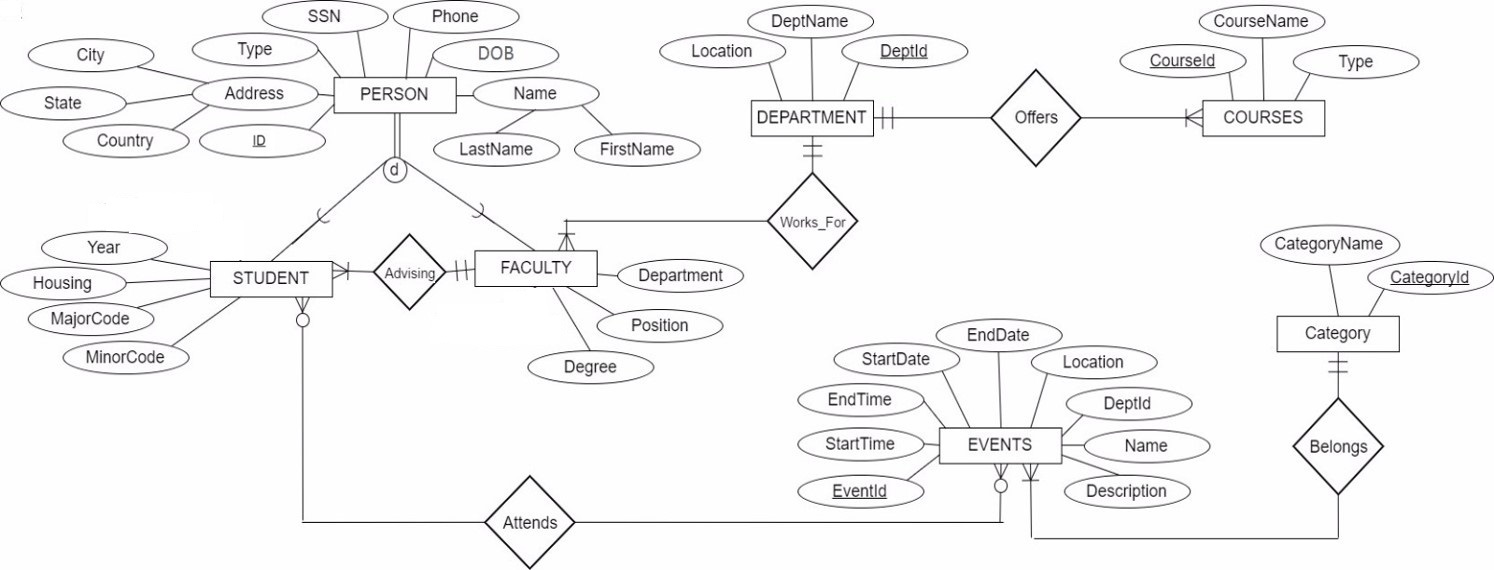
**Constraints**

1.Students can only view the events that they have attended and upcoming.

2. Students cannot make any changes to an event.

3. Advisors can only view the list of students that have been assigned for them and the events in which students have participated.

**EER Diagram**

****

**EERD Narrative And Features**

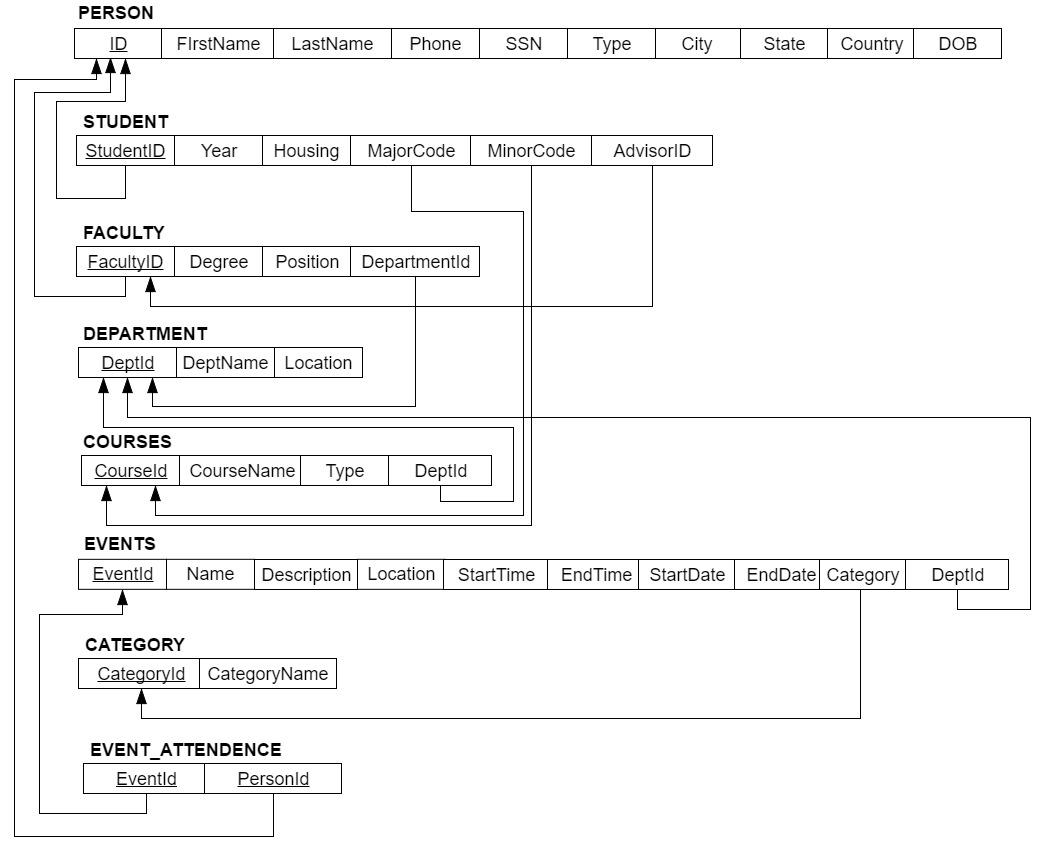
**Narration of EERD:**

The above diagram depicts the entity relationship of our Student Activity Management System. Each person is stored in the Person table. The person entity is specialised into two sub-entities namely “Student” and “Faculty”. A Faculty member can advise from one-to-many students, but a student must only have one advisor.Each Faculty works for one department while a Department can have from one-to-many faculty. Departments offer from one-to-many courses, but note that courses aren’t affiliated with multiple departments. Only one department must be associated with a course. Students can attend from zero-to-many Events where as an Event can be attended by zero-to-many students. In simple words, Students can attend multiple events or none at all. Just as events can host multiple students or it can be a complete no-show. This many-to-many relationship is expressed in a separate EventAttendance table that correlates the studentID of the students that attend with the eventID. This allows the University to keep track of attendance for administrative purposes. Events are hosted by departments, but each event can belong to only one category.

**Features:**

* **Generalization/Specialization Implementation** - As noted in the narrative above, there was a general entity in Person that was specialized into 2 entities. A person either has to be a Student or Faculty in a disjoint and complete specialization.
* **Tables are in 3rd Normal Form**
* **Many-to-Many relationship** - Also in the narrative, there was mention of a many-to-many relationship between Student and Event. To resolve that, we created a separate table with the primary keys from both to help accurately track attendance.

**ERD Relational Mapping**



**Tables**

|  |  |
| --- | --- |
| **Person** | |
| **ID (PK)** | INT |
| **SSN** | VARCHAR |
| **DOB** | DATE |
| **pType** | VARCHAR |
| **Phone** | VARCHAR |
| **City** | VARCHAR |
| **State** | VARCHAR |
| **Country** | VARCHAR |
| **FirstName** | VARCHAR |
| **LastName** | VARCHAR |

|  |  |
| --- | --- |
| **Student** | |
| **StudentID (PK)** | INT |
| **Year** | INT |
| **Housing** | VARCHAR |
| **MinorCode** | INT |
| **MajorCode** | INT |
| **sYear** | VARCHAR |

|  |  |
| --- | --- |
| **Users** | |
| **ID (PK)** | INT |
| **Username** | VARCHAR |
| **Password** | VARCHAR |

|  |  |
| --- | --- |
| **Faculty** | |
| **FacultyID (PK)** | INT |
| **DeptID (FK)** | INT |
| **Degree** | VARCHAR |
| **Position** | VARCHAR |

|  |  |
| --- | --- |
| **Department** | |
| **DeptID (PK)** | INT |
| **DeptName** | VARCHAR |
| **Location** | VARCHAR |

|  |  |
| --- | --- |
| **Courses** | |
| **CourseID (PK)** | INT |
| **CourseName** | VARCHAR |
| **Type** | VARCHAR |
| **DeptID (FK)** | INT |

|  |  |
| --- | --- |
| **Events** | |
| **EventID (PK)** | INT |
| **EventName** | VARCHAR |
| **Location** | VARCHAR |
| **DeptID (FK)** | INT |
| **StartTime** | TIME |
| **EndTime** | TIME |
| **StartDate** | DATE |
| **EndDate** | DATE |
| **Description** | VARCHAR |
| **CatID (FK)** | INT |

|  |  |
| --- | --- |
| **Category** | |
| **CategoryID (PK)** | INT |
| **CategoryName** | VARCHAR |

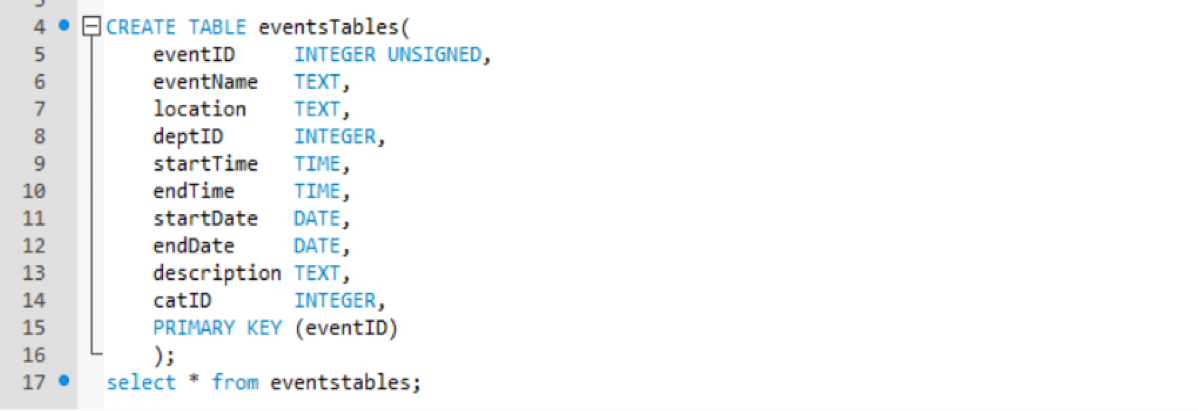
|  |  |
| --- | --- |
| **EventsAttendance** | |
| **EventID (PK, FK)** | INT |
| **StudentID (PK, FK)** | INT |

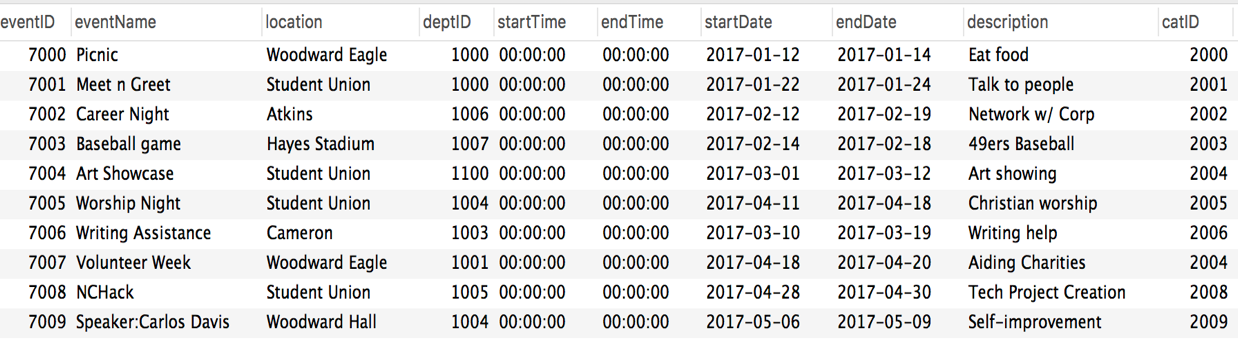
**Database: Physical Implementation**

* We have created a database and named it as “Student activity management system” and then 9 tables were created in the database.
* All the tables that are present in our database are in the normalized form.
* The storage engine used is “Innodb”, which is the default storage engine that comes with the MySql 5.5
* The User Interface was implemented by using HTML, CSS, JavaScript and frameworks like BootStrap for reponsive web design.
* We have used php, a server scripting language for web development.
* Two completely different modules have been created for Advisors and Students with different privileges.

**Creation of Tables:**

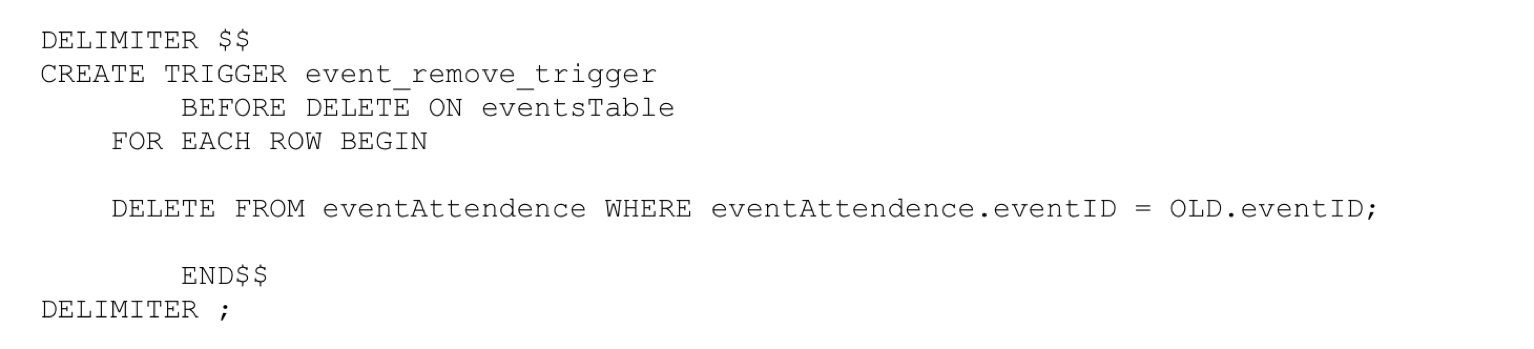
All the tables were created by using the CREATE statement. For instance, eventTables was created as follows:

****

****

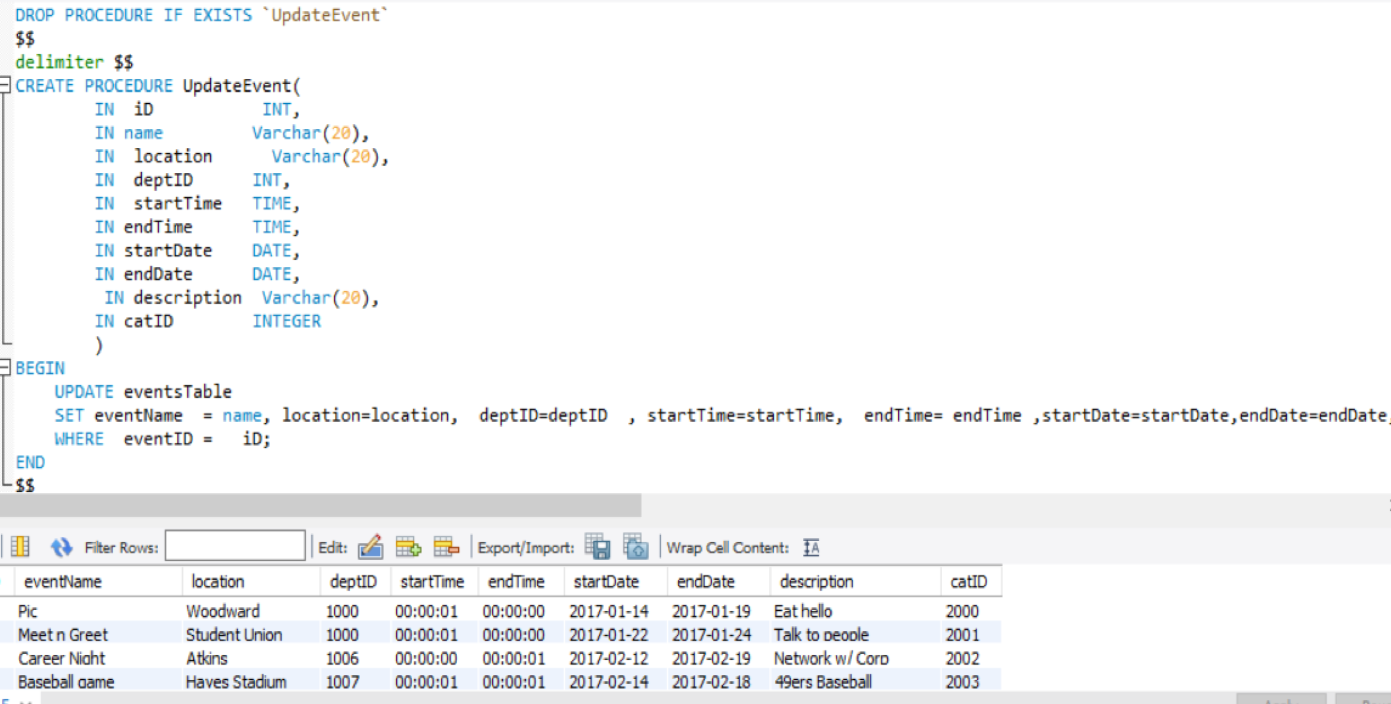
**Triggers** :

We implemented a trigger to handle the deletion of events created. If an event is created, it can be systematically removed from the table, and wiped from the event attendance table to prevent the appearance of students attending ‘phantom’ events. Basically, there will be no null values left hanging.



**Stored Procedures**:

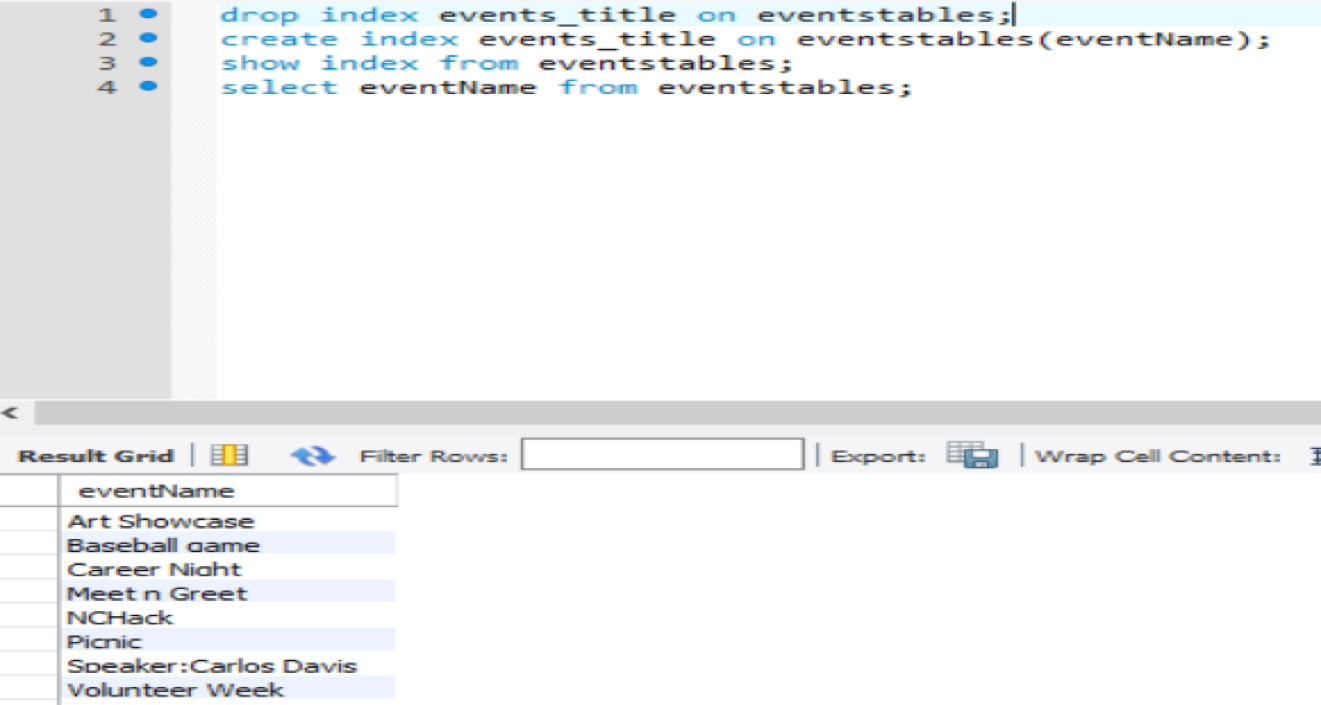
We have used stored procedure for updating an event in the eventsTable as shown below:



**Query Optimization with Indexing**

Indexing is a way of sorting the number of records on multiple fields. One of the dis-advantage of indexing is that indexes require additional space on the disk.

we have achieved query optimization with indexing to display the name of the events as shown below:

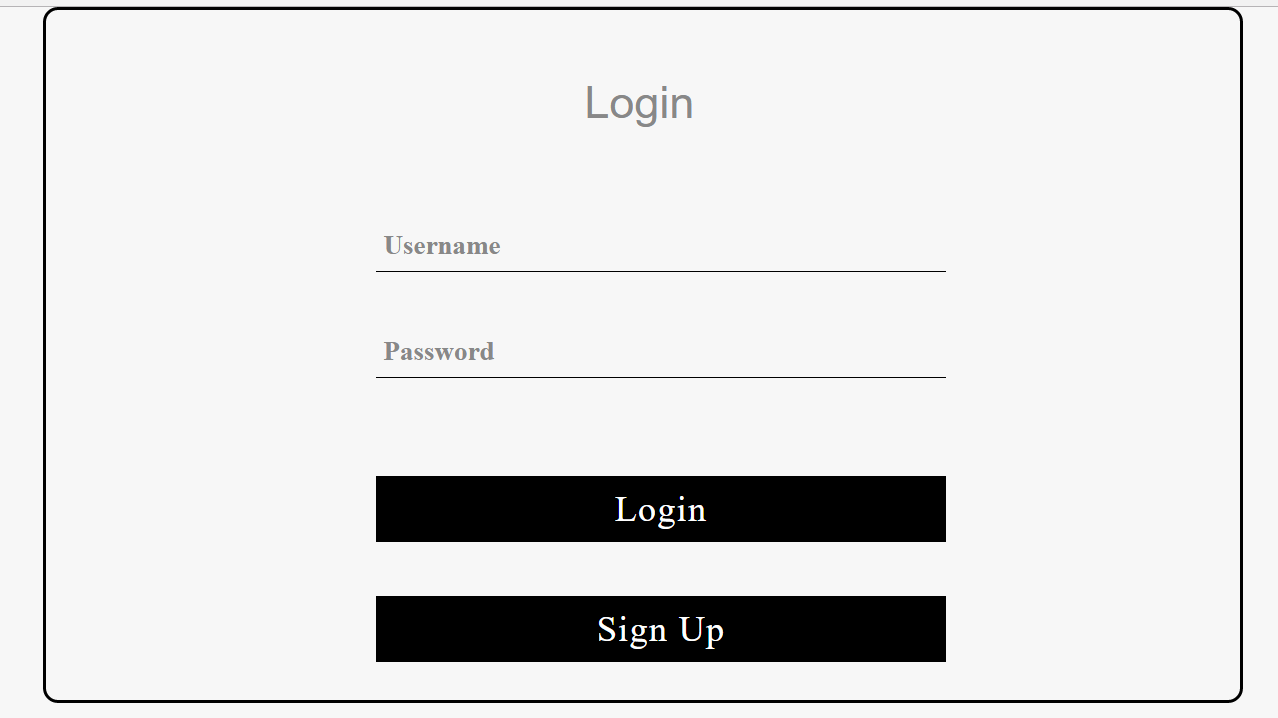
****

**Special Design and Features**

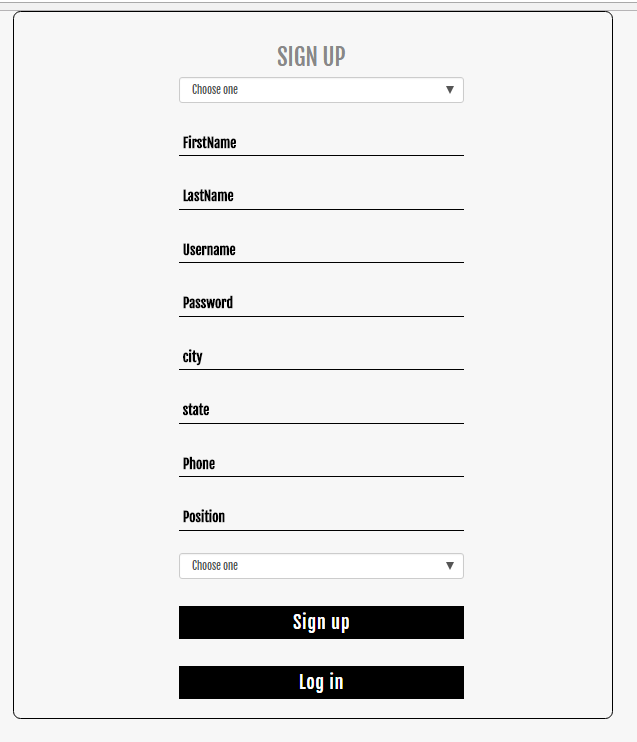
* Different modules have been created for advisors and students with different privileges.
* Differentiating the advisors and students during the signup is done by taking input explicitly from the user and storing it in the ‘type’ column of the ‘person’ table.

**Screenshots**

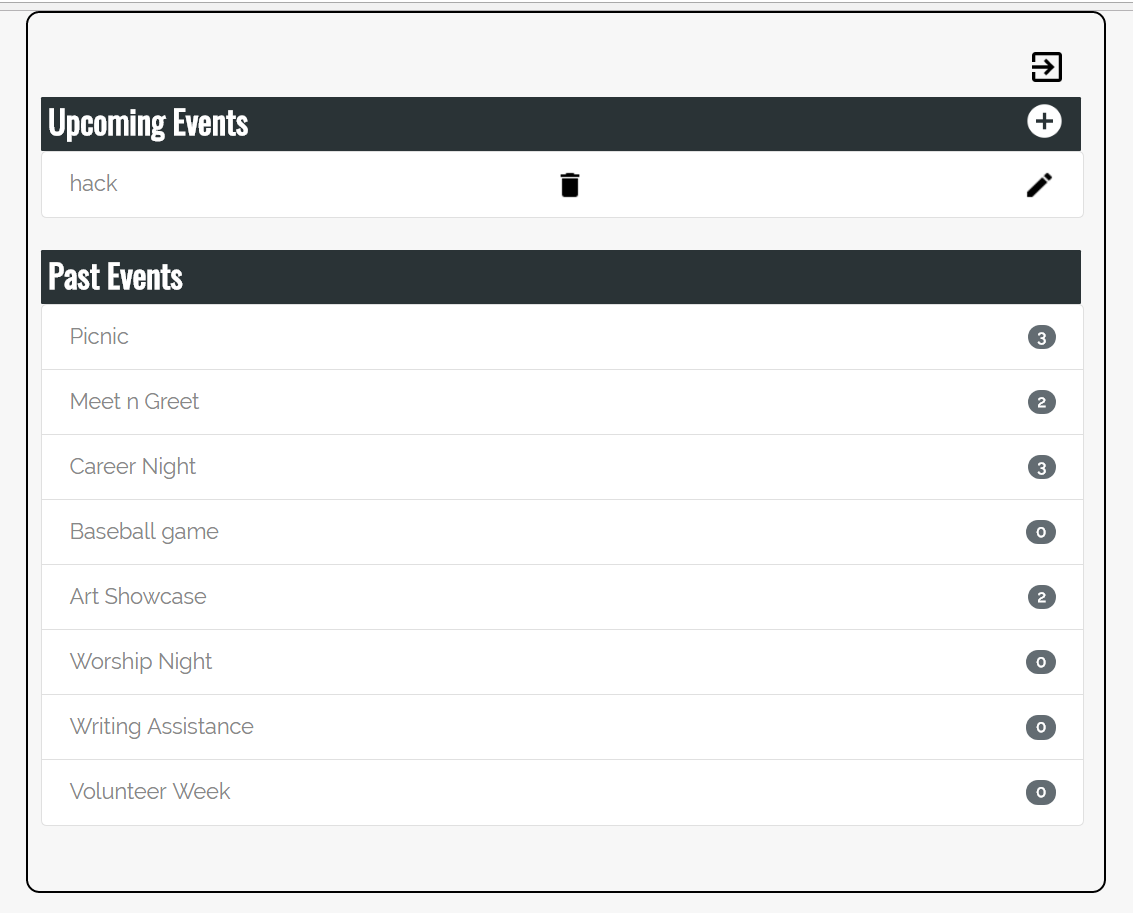
**Signin**

****

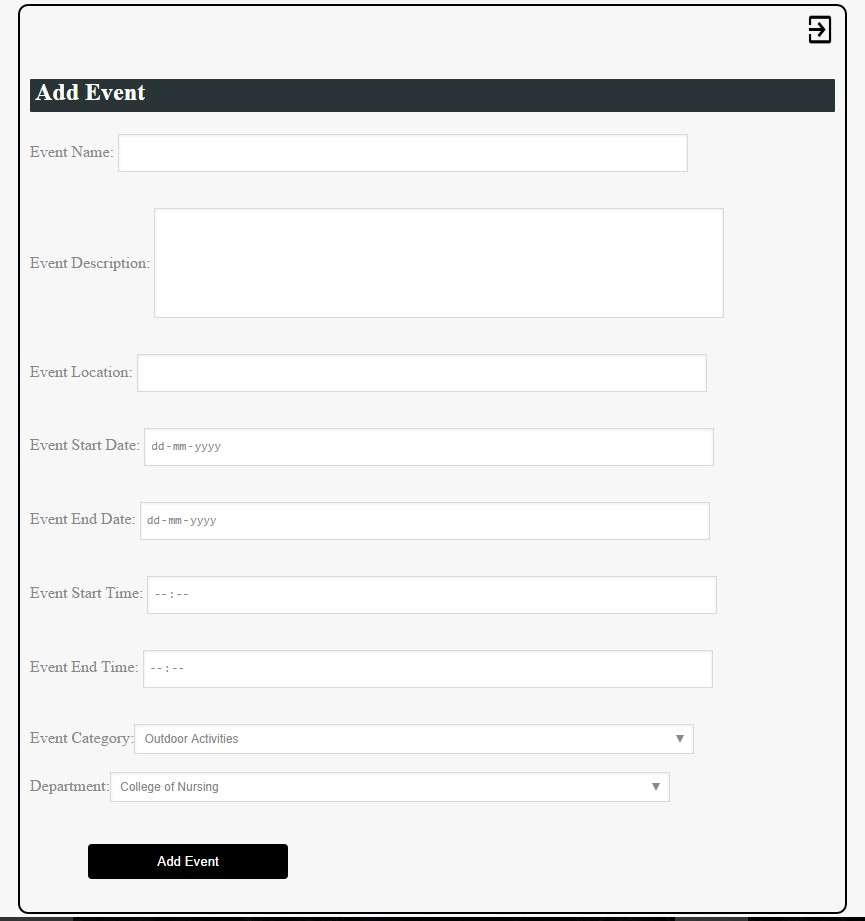
**Signup**

****

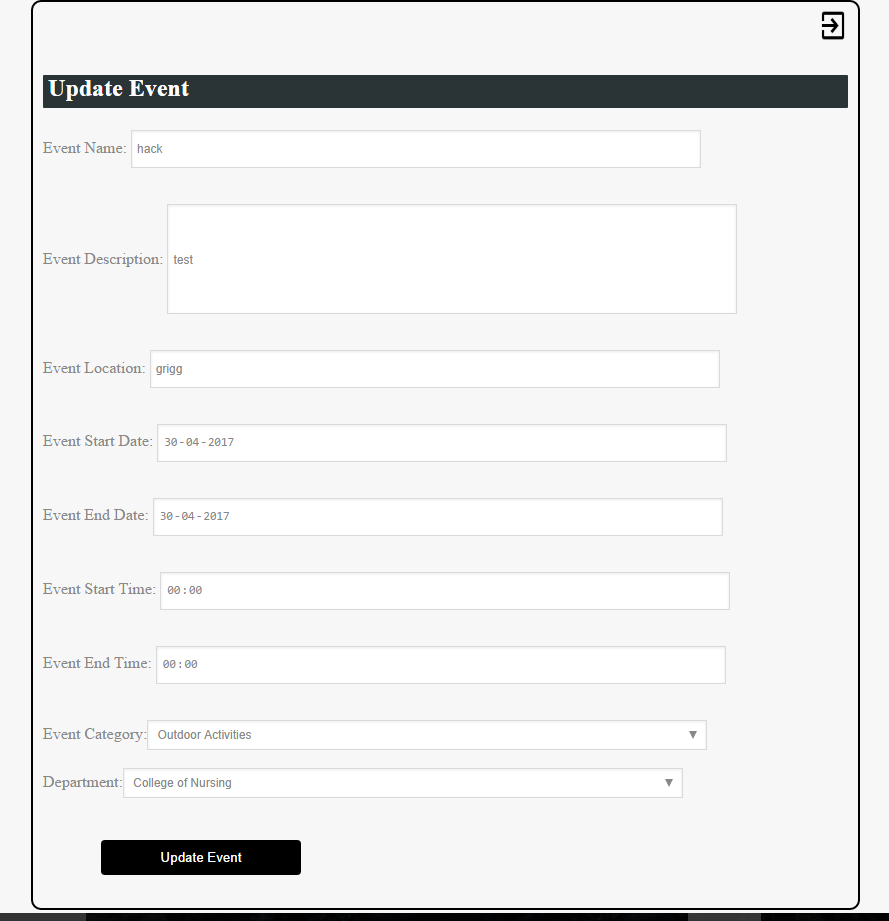
**Advisor’s Home**

****

**Add an Event**

****

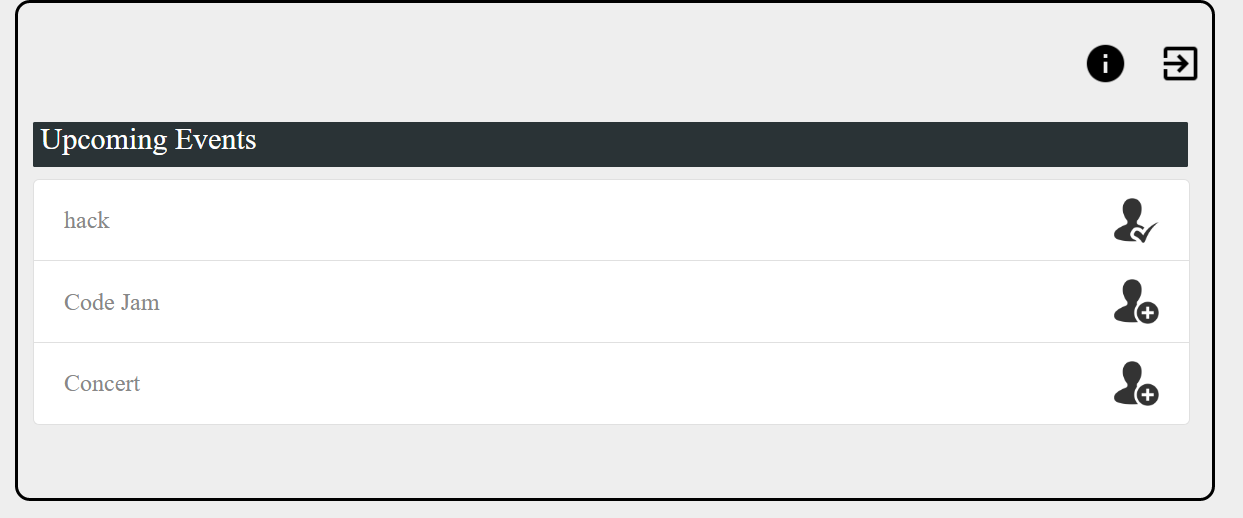
**Update Event**

****

**Event Details**

****

**Student’s Home**

****

**Student’s Information**

****

**Conclusion**

* System to track of upcoming events.
* Effectively implemented CRUD capabilities.
* Only Advisor can

-CREATE Event

-READ Event

-UPDATE Event

-DELETE Event

* Students can

- VIEW and register for Events

**Future Scope**

* Data Analysis based on events attendance, eventspriority

- By analysing the data from the past events , we can expect which type of events are likely to succeed or fail.

* **Mobile App**:

-The future is in Mobile apps. Creating a mobile app to track events, sign up to attend, and track attendance would take this to the next level.

**Team Members and Roles**

|  |  |
| --- | --- |
| **Team Member** | **Role** |
| Rajia Shareen | Project Manager/PHP developer |
| Guna Chandrika Paturi | Project Manager/UI developer |
| Malcolm Lipscomb | Lead Report Writer/PHP developer |
| Matthew Goodman | Lead SQL Programmer/UI developer |
| Vinutna Gannu | Lead WEB/PHP Developer |
| Devireddy Venkata Sai Krishna Reddy | Project Supervisor/SQL Programmer |

**Contact Information**

|  |  |  |
| --- | --- | --- |
| **Team Member** | **Email** | **Phone** |
| Shaik Rajia Shareen | rshaik3@uncc.edu | 704 858 6724 |
| Guna Chandrika Paturi | gpaturi@uncc.edu | 301 795 5740 |
| Malcolm Lipscomb | mlipsco6@uncc.edu | 864 838 5011 |
| Matthew Goodman | mgoodm22@uncc.edu | 704 692 1387 |
| Vinutna Gannu | vgannu@uncc.edu | 980 938 9825 |
| Devireddy Venkata Sai Krishna Reddy | vdevire1@uncc.edu | 980 938 9874 |

**Method Of Communication and File Sharing for Team**

1. Method of Communication: WhatsApp group
2. Method of File Sharing: GoogleDrive, GoogleDocs

**Project Plan**

|  |  |
| --- | --- |
| **Activity** | **Date** |
| **Team Building** | March 17, 2017 |
| **Project Deliverable I** | April 2, 2017 |
| **Project Deliverable II** | April 17, 2017 |
| **Project Deliverable III** | April 30, 2017 |