**Heavy Weight Group 2 # - Alumni Network**

*Team Members: Dion Chapel*

*Shravya Donthisaram*

*Naman Manocha*

*Mahalavanya Sriram*

## ***PROJECT PROPOSAL***

The goal is to create an alumni website that maintains Alumni and student data, clubs, alumni student interaction forum , reunions, information about hiring, list of events that takes place on the campus and also allows the institute to communicate about the latest updates to the alumni, students and leverage the network.

## ***PROJECT ENVIRONMENT***

Project database will be MySql Workbench 8.0.18.CE, Service will be in Java8

## ***HIGH LEVEL REQUIREMENTS***

### **Initial user roles**

|  |  |
| --- | --- |
| **User Role** | **Description** |
| Alumni | Alumni can login to the website and view the updates,events and communicate with the students |
| Students | Students can login to the website, view the latest updates, participate in the events and communicate with the alumni attending them |
| Administrator | Administrative users for the application |

### 

### **Initial user story descriptions**

|  |  |
| --- | --- |
| **Story ID** | **Story description** |
| US1 | As a User, I want to create an account and login to the website |
| US2 | As a student, I can follow the alumni of the institute who is a part of the website and can communicate with them |
| US3 | As an administrator, I can add various events, news regarding the institute where the alumni and students will be the participants |
| US4 | As an alumni, I can login and follow other alumni and students of the institute. |
| US5 | As an alumni, I can post information about the hiring opportunities and events |
| US6 | As a student, I can view about upcoming hiring opportunities and events |
| US7 | As an alumni, I can join exclusive clubs and interact with other alumni who are a part of that particular clubs |
| US8 | As an administrator, I can add various clubs which Alumni can be a part of |
| US9 | As a user, I can search alumni/student by degree, stream, company, location, industry, experience |

## 

## 

## 

## ***HIGH LEVEL CONCEPTUAL DESIGN***

**Entities:**

Student

Alumni

Administrator

Events

Clubs

Message

Hiring post

Account

User

**Relationships:**

Administrator creates events

Administrator creates news

Administrator creates clubs

Alumni creates a job opportunity

Alumni joins club

User views/searches job opportunities

User views/searches events

User views/searches news updates

User creates account

Users search for other users by qualifiers (degree, company…)

User follows user

User messages User

User can refer to alumni or student user

# **Sprint 1**

## ***REQUIREMENTS***

|  |  |
| --- | --- |
| **Story ID** | **Story description** |
| US1 | As a User, I want to create an account and login to the website |
| US2 | As a student, I can follow the alumni of the institute who is a part of the website and can communicate with them |
| US3 | As an administrator, I can add various events, news regarding the institute where the alumni and students will be the participants |
| US4 | As an alumni, I can login and follow other alumni and students of the institute. |
| US5 | As an alumni, I can post information about the hiring opportunities and events |
| US6 | As a student, I can view about upcoming hiring opportunities and events |
| US7 | As an alumni, I can join exclusive clubs and interact with other alumni who are a part of that particular clubs |

## ***CONCEPTUAL DESIGN***

Entity: **Account**

Attributes:

User\_id[single-valued]

Password[single-valued]

Account\_Type[single-valued]

Account\_Type here is student,administrator or alumni account

Entity: **Alumni**

Attributes:

Alumni\_id[single-valued]

Alumni\_name[single-valued]

Current\_work\_place[single-valued]

Prior\_work\_place[multi-valued]

Address[multi-valued]

Contact\_Number[multi-valued]

email\_id[multi-valued]

Entity: **Student**

Student\_id[single-valued]

Student\_name[single-valued]

Department[single-valued]

Address[multi-valued]

Year[single-valued]

Credits\_till\_date[single-valued]

GPA[single-valued]

Contact\_Number[multi-valued]

email\_id[multi-valued]

Entity: **Club**

Club\_id[single-valued]

Club\_name[single-valued]

Club\_leader[single-valued]

Club\_members[multi-valued]

Club\_description[single-valued]

Entity: **Event**

Event\_name[single-valued]

Event\_description[single-valued]

Event\_id[single-valued]

Organizing\_Club[single-valued]

Entity: **Job**

Job\_name[single-valued]

Job\_id[single-valued]

Job\_description[single-valued]

Company[single-valued]

Phone\_number[single-valued]

Website[single-valued]

Alumni\_publisher[single-valued]

Entity: **Administrator**

Administrator\_id[single-valued]

Administrator\_name[single-valued]

Contact\_Number[multi-valued]

email\_id[multi-valued]

Relationship: **Administrator** creates **events**

Cardinality:Many to Many

Participation:

Administrator has total participation

Events has total participation

Relationship: **Administrator** creates **Clubs**

Cardinality:One to Many

Participation:

Administrator has total participation

Clubs has total participation

Relationship: **Alumni** creates **Job posting**

Cardinality: One to Many

Participation:  
 Alumni has partial participation

Job posting has total participation

Relationship: **Alumni** joins **Club**

Cardinality: Many to Many

Participation:

Alumni has partial participation

Club has partial participation

Relationship: **User** searches **Job postings**

Cardinality: Many to Many

Participation:

User has partial participation

Job postings has total participation

Relationship: **User** searches **Events**

Cardinality: Many to Many

Participation:

User has partial participation

Events has total participation

Relationship: **User** creates **Account**

Cardinality: One to One

Participation:

User has total participation

Account has total participation

Relationship: **User** searches **Users**

Cardinality: Many to Many

Participation:

User has total participation

Relationship: **User** follows **user**

Cardinality: Many to Many

Participation:

User has partial participation

***LOGICAL DESIGN***

Table: **Account**

Columns:

User\_id

Password

Account\_Type

Justification: User\_id will be email id of the users (Alumni or Students)

Table: **Alumni**

Columns:

Alumni\_id

Alumni\_name

Current\_work\_place

Prior\_work\_place

Address

Contact\_Number

Email\_id [Foreign Key; references **User\_id** of **Account**]

Justification: **Alumni\_id** is the primary key for this table (Could be numeric like our 800 id)

**Email\_id**  is the foreign key for this table referred from Account table

Table: **Student**

Columns:

Student\_id

Student\_name

Department

Address

Year

Credits\_till\_date

GPA

Contact\_Number

Email\_id [Foreign Key; references **User\_id** of **Account**]

Justification: **Student\_id** is the primary key for this table (Could be numeric like our 800 id)

**Email\_id**  is the foreign key for this table referred from Account table

Table: **Clubs**

Columns:

Club\_id

Club\_name

Club\_Leader [Foreign Key; references **Alumni\_id** of **Alumni**]

Club\_Description

Justification: **Club\_id** is the primary key for this table

**Leader**  of the club will be an alumni. Hence it will have the foreign key that is alumni\_id

Table: **Club** **Members**

Columns:

Club\_id [Foreign Key; references **Club\_id** of **Club**]

Alumni\_id [Foreign Key; references **Alumni\_id** of **Alumni**]

Justification: **Club\_id** and **Alumni\_Id** are the foreign key for this table

Table: **Job postings**

Columns:

Job\_id

Job\_name

Description

Company

Contact\_number

Website

Publisher[Foreign Key; references **Alumni\_id** of **Alumni**]

Justification: **Job\_id** is the primary key for this table

**Publisher** of the job postingwill be an alumni. Hence it will have the foreign key that is alumni id

Table: **Event**

Columns:

Event\_id

Event\_name

Event\_Description

Organizing\_Club [Foreign Key; references **Club\_id** of **Club**]

Justification: **Event\_id** is the primary key for this table

**Organizing\_Club** is the forieng key from the club table. A event can be organised by only one club but participants can be anybody

**Sprint 2**

### 

|  |  |
| --- | --- |
| **Story ID** | **Story description** |
| US1 | As a User, I want to create an account and login to the website |
| US2 | As a student, I can follow the alumni of the institute who is a part of the website and can communicate with them |
| US3 | Clubs creates various events regarding the institute where the alumni and students will be the participants |
| US4 | As an alumni, I can login and follow other alumni and students of the institute. |
| US5 | As an alumni, I can post information about the hiring opportunities and events |
| US6 | As a student, I can view about upcoming hiring opportunities and events |
| US7 | As an alumni, I can join exclusive clubs and interact with other alumni who are a part of that particular clubs |
| US8 | As an administrator, I can create various clubs which Alumni can be a part of |
| US9 | As a user, I can search alumni by company, industry, experience |
| US10 | As a user, I can search student by degree, stream. |

We have updated new user stories as a part of Sprint 2 and these user stories are highlighted in orange. **Login and logout, Searches and views are not stored in the database**.

## ***CONCEPTUAL DESIGN***

Entity: **Account**

Attributes:

Email\_id[single-valued]

User\_id[single-valued]

Password[single-valued]

Account\_Type[single-valued]

Account\_Type here is student, administrator or alumni account

Entity: **User**

Attributes:

User\_id[single-valued]

Name[composite]

First\_name[single-valued]

Last\_name[single-valued]

Address[composite]

Address\_line\_1[single-valued]

Address\_line\_2[single-valued]

City[single-valued]

State[single-valued]

Zip\_code[single-valued]

Contact\_Number[single-valued]

Entity: **Alumni**

Attributes:

Alumni\_id[single-valued]

Current\_work\_place[single-valued]

Prior\_work\_place[single-valued]

Industry[single-valued]

Experience[single-valued]

Entity: **Student**

Student\_id[single-valued]

Department[single-valued]

Year[single-valued]

Credits\_till\_date[single-valued]

GPA[single-valued]

Stream[single-valued]

Degree[single-valued]

Entity: **Club**

Club\_id[single-valued]

Club\_name[single-valued]

Club\_leader[single-valued]

Club\_members[multi-valued]

Club\_description[single-valued]

Entity: **Event**

Event\_name[single-valued]

Event\_description[single-valued]

Event\_id[single-valued]

Organizing\_Club[single-valued]

Entity: **Job**

Job\_name[single-valued]

Job\_id[single-valued]

Job\_description[single-valued]

Company[single-valued]

Contact\_Number[single-valued]

Website[single-valued]

Alumni\_publisher[single-valued]

Entity: **Administrator**

Admin\_id[single-valued]

Relationship: **Club** creates **events**

Cardinality:One to Many

Participation:

Club has partial participation

Events has total participation

Relationship: **Administrator** creates **Clubs**

Cardinality:One to Many

Participation:

Administrator has total participation

Clubs has total participation

Relationship: **Club\_leader (alumni)** leads **Clubs**

Cardinality:One to Many

Participation:

Alumni has partial participation

Clubs has total participation

Relationship: **Alumni** creates **Job posting**

Cardinality: One to Many

Participation:  
 Alumni has partial participation

Job posting has total participation

Relationship: **Alumni** joins **Club**

Cardinality: Many to Many

Participation:

Alumni has partial participation

Club has partial participation

Relationship: **User** creates **Account**

Cardinality: One to One

Participation:

User has total participation

Account has total participation

Relationship: **User** follows **user**

Cardinality: Many to Many

Participation:

User has partial participation

***LOGICAL DESIGN***

Table: **Account**

Columns:

Email\_id

User\_id[Foreign Key; references **User\_id** of **User**]

Password

Account\_Type\_id[Foreign Key; references **Account\_Type\_id** of **Account\_Type**]

Justification: Email\_id will be email id of the users (Alumni or Students)

Highest normalization level: <1NF/2NF/3NF/**4NF**>

Table: **Account\_Type**

Columns:

Account\_Type\_id

Account\_Type

Highest normalization level: <1NF/2NF/3NF/**4NF**>

Table: **User**

Columns:

User\_id

First\_name

Last\_name

Address\_line\_1

Address\_line\_2

City

State

Zip\_Code

Contact\_Number

Highest normalization level: <1NF/2NF/**3NF**/4NF>

Justification:There is a transitive dependency among columns Address\_line\_1, Address\_line\_2, City, State, Zip\_code.

Table: **Alumni**

Columns:

Alumni\_id [Foreign Key; references User\_id of Account]

Current\_work\_place

Prior\_work\_place

Industry

Experience

Justification: **Alumni\_id** is the primary key for this table (Could be numeric like our 800 id)

**Alumni\_id**  is the foreign key for this table referred from Account table

Highest normalization level: <1NF/2NF/3NF/**4NF**>

Table: **Student**

Columns:

Student\_id [Foreign Key; references User\_id of Account]

Department

Year

Credits\_till\_date

GPA

Stream

Degree

Justification: **Student\_id** is the primary key for this table (Could be numeric like our 800 id)

**Student\_id**  is the foreign key for this table referred from Account table

Highest normalization level: <1NF/2NF/3NF/**4NF**>

Table: **Administrator**

Columns:

Admin\_id [Foreign Key; references User\_id of Account]

Justification:**Admin\_id** is the primary key for this table (Could be numeric like our 800 id)

**Admin\_id**  is the foreign key for this table referred from Account table

Highest normalization level: <1NF/2NF/3NF/**4NF**>

Table: **Clubs**

Columns:

Club\_id

Club\_name

Club\_Leader [Foreign Key; references **Alumni\_id** of **Alumni**]

Club\_Description

Created\_by [Foreign Key; references **Admin\_id** of **Administrator**]

Justification: **Club\_id** is the primary key for this table

**Leader**  of the club will be an alumni. Hence it will have the foreign key that is alumni\_id (one to many mapping of Alumni leads clubs) and the Created\_by which is alumni\_id will also be a foreign\_key (one to many mapping of Administrator creates clubs)

Highest normalization level: <1NF/2NF/3NF/**4NF**>

Table: **Job postings**

Columns:

Job\_id

Job\_name

Description

Company

Contact\_number

Website

Publisher[Foreign Key; references **Alumni\_id** of **Alumni**]

Justification: **Job\_id** is the primary key for this table

**Publisher** of the job postingwill be an alumni. Hence it will have the foreign key that is user id

( one to many mapping of Administrator creates clubs)

Highest normalization level: <1NF/2NF/3NF/**4NF**>

Table: **Event**

Columns:

Event\_id

Event\_name

Event\_Description

Organizing\_Club [Foreign Key; references **Club\_id** of **Club**]

Justification: **Event\_id** is the primary key for this table.**Organizing\_Club** is the foreign key from the club table ( one to many mapping of Clubs create Events). A event can be organised by only one club but participants can be anybody

Highest normalization level: <1NF/2NF/3NF/**4NF**>

Table: **Club** **Members**

Columns:

Club\_id [Foreign Key; references **Club\_id** of **Club**]

Alumni\_id [Foreign Key; references **Alumni\_id** of **Alumni**]

Primary Key: Combination of Club\_id and Alumni\_id

Justification: **Club\_id** and **Alumni\_id** are the foreign key for this table

This table is a cross-reference table representing membership to a club by an alumni for the many-to-many relationship.

Highest normalization level: <1NF/2NF/3NF/**4NF**>

Table: **Follow**

Columns:

Follower\_id[Foreign Key; references **User\_id** of **User]**

Following\_id [Foreign Key; references **User\_id** of **User]**

Primary Key: Combination of Follower\_id and Following\_id

This table is a cross-reference table representing ownership of events by clubs for the many-to-many relationship

Highest normalization level: <1NF/2NF/3NF/**4NF**>

**Sprint 3**

|  |  |
| --- | --- |
| **Story ID** | **Story description** |
| US1 | As a User, I want to create an account and login to the website |
| US2 | As a student, I can follow the alumni of the institute who is a part of the website and can communicate with them |
| US3 | Clubs creates various events regarding the institute where the alumni and students will be the participants |
| US4 | As an alumni, I can login and follow other alumni and students of the institute. |
| US5 | As an alumni, I can post information about the hiring opportunities and events |
| US6 | As a student, I can view about upcoming hiring opportunities and events |
| US7 | As an alumni, I can join exclusive clubs and interact with other alumni who are a part of that particular clubs |
| US8 | As an administrator, I can create various clubs which Alumni can be a part of |
| US9 | As a user, I can search alumni by company, industry, experience |
| US10 | As a user, I can search student by degree, stream. |
| US11 | As a user, I can join an event |
| US12 | As a user, I can search an alumni by year of graduation |

We have updated new user stories as a part of Sprint 3 and these user stories are highlighted in green. **Login and logout, Searches and views are not stored in the database**

## ***CONCEPTUAL DESIGN***

Entity: **Account**

Attributes:

Email\_id[single-valued]

User\_id[single-valued]

Password[single-valued]

Account\_Type[single-valued]

Account\_Type here is student, administrator or alumni account

Entity: **User**

Attributes:

User\_id[single-valued]

Name[composite]

First\_name[single-valued]

Last\_name[single-valued]

Address[composite]

Address\_line\_1[single-valued]

Address\_line\_2[single-valued]

City[single-valued]

State[single-valued]

Zip\_code[single-valued]

Contact\_Number[single-valued]

Entity: **Alumni**

Attributes:

Alumni\_id[single-valued]

Current\_work\_place[single-valued]

Prior\_work\_place[single-valued]

Industry[single-valued]

Experience[single-valued]

Graduation\_year[single-valued]

Entity: **Student**

Student\_id[single-valued]

Department[single-valued]

Year[single-valued]

Credits\_till\_date[single-valued]

GPA[single-valued]

Stream[single-valued]

Degree[single-valued]

Entity: **Club**

Club\_id[single-valued]

Club\_name[single-valued]

Club\_leader[single-valued]

Club\_members[multi-valued]

Club\_description[single-valued]

Entity: **Event**

Event\_name[single-valued]

Event\_description[single-valued]

Event\_id[single-valued]

Organizing\_Club[single-valued]

Event\_date[single-valued]

Entity: **Job**

Job\_name[single-valued]

Job\_id[single-valued]

Job\_description[single-valued]

Company[single-valued]

Contact\_Number[single-valued]

Website[single-valued]

Publisher[single-valued]

Entity: **Administrator**

Admin\_id[single-valued]

Relationship: **Club** creates **events**

Cardinality:One to Many

Participation:

Club has partial participation

Events has total participation

Relationship: **Administrator** creates **Clubs**

Cardinality:One to Many

Participation:

Administrator has total participation

Clubs has total participation

Relationship: **Club\_leader (alumni)** leads **Clubs**

Cardinality:One to Many

Participation:

Alumni has partial participation

Clubs has total participation

Relationship: **Alumni** creates **Job posting**

Cardinality: One to Many

Participation:  
 Alumni has partial participation

Job posting has total participation

Relationship: **Alumni** joins **Club**

Cardinality: Many to Many

Participation:

Alumni has partial participation

Club has partial participation

Relationship: **User** creates **Account**

Cardinality: One to One

Participation:

User has total participation

Account has total participation

Relationship: **User** follows **user**

Cardinality: Many to Many

Participation:

User has partial participation

Relationship: **User** joins **Events**

Cardinality: Many to Many

Participation:

User has partial participation

Event has total participation

***LOGICAL DESIGN***

Table: **Account**

Columns:

Email\_id

User\_id[Foreign Key; references **User\_id** of **User**]

Password

Account\_Type\_id[Foreign Key; references **Account\_Type\_id** of **Account\_Type**]

Justification: Email\_id will be email id of the users (Alumni or Students)

Highest normalization level: <1NF/2NF/3NF/**4NF**>

Indexes:

Index #: clustered

Columns: Email\_id

Justification: As Primary key is email\_id, it forms a clustered index on this table

Table: **Account\_Type**

Columns:

Account\_Type\_id

Account\_Type

Highest normalization level: <1NF/2NF/3NF/**4NF**>

Indexes:

Index #: clustered

Columns: Account\_Type\_id

Justification: As Primary key is Account\_Type\_id, it forms clustered index on this table

Table: **User**

Columns:

User\_id

First\_name

Last\_name

Address\_line\_1

Address\_line\_2

City

State

Zip\_Code

Contact\_Number

Highest normalization level: <1NF/2NF/**3NF**/4NF>

Justification:There is a transitive dependency among columns Address\_line\_1, Address\_line\_2, City, State, Zip\_code.

Indexes:

Index #: clustered

Columns: User\_id

Justification: As Primary key is User\_id, it forms clustered index on this table

Index #: non-clustered

Columns: Address\_line\_1, Address\_line\_2,City,State,Zip\_code

Justification: this index is created with the name **Address** with above columns for the quick fetch of addresses of the user.

Table: **Alumni**

Columns:

Alumni\_id [Foreign Key; references User\_id of Account]

Current\_work\_place

Prior\_work\_place

Industry

Experience

Graduation\_year

Justification: **Alumni\_id** is the primary key for this table (Could be numeric like our 800 id)

**Alumni\_id**  is the foreign key for this table referred from Account table

Highest normalization level: <1NF/2NF/3NF/**4NF**>

Indexes:

Index#: Clustered

Columns: Alumni\_id

Justification:As Primary key is Alumni\_id, it forms clustered index on this table

Index#: Non-clustered

Columns: Graduation\_year

Justification: this index is created with the name **Graduation\_year\_indx** with the above column as this is the most frequently searched while looking for an Alumni in the database.

Index#: Non-clustered

Columns: Current\_workplace, Prior\_workplace  
Justification:this index is created with the name **Work\_Place** with above mentioned columns as this is the most frequently searched while looking for an Alumni in the database

Table: **Student**

Columns:

Student\_id [Foreign Key; references User\_id of Account]

Department

Year

Credits\_till\_date

GPA

Stream

Degree

Justification: **Student\_id** is the primary key for this table (Could be numeric like our 800 id)

**Student\_id**  is the foreign key for this table referred from Account table

Highest normalization level: <1NF/2NF/3NF/**4NF**>

Indexes:

Index#: Clustered

Columns: Student\_id

Justification:As Primary key is Student\_id, it forms clustered index on this table

Index#: Non Clustered

Columns: Stream,Degree

Justification: this index is created with the name **Academic\_info** with above mentioned columns as this is the most frequently searched while looking for a student in the database

Table: **Administrator**

Columns:

Admin\_id [Foreign Key; references User\_id of Account]

Justification:**Admin\_id** is the primary key for this table (Could be numeric like our 800 id)

**Admin\_id**  is the foreign key for this table referred from Account table

Highest normalization level: <1NF/2NF/3NF/**4NF**>

Indexes:

Index#: Clustered

Columns: Admin\_id

Justification:As the Primary key is Admin\_id, it forms clustered index on this table

Table: **Clubs**

Columns:

Club\_id

Club\_name

Club\_Leader [Foreign Key; references **Alumni\_id** of **Alumni**]

Club\_Description

Created\_by [Foreign Key; references **Admin\_id** of **Administrator**]

Justification: **Club\_id** is the primary key for this table

**Leader**  of the club will be an alumni. Hence it will have the foreign key that is alumni\_id (one to many mapping of Alumni leads clubs) and the Created\_by which is alumni\_id will also be a foreign\_key (one to many mapping of Administrator creates clubs)

Highest normalization level: <1NF/2NF/3NF/**4NF**>

Indexes:

Index#: Clustered

Columns: Club\_id

Justification:As the Primary key is Club\_id, it forms a clustered index on this table.

Table: **Job postings**

Columns:

Job\_id

Job\_name

Description

Company

Contact\_number

Website

JP\_Due\_date

JP\_last\_modified

Publisher[Foreign Key; references **Alumni\_id** of **Alumni**]

Justification: **Job\_id** is the primary key for this table

**Publisher** of the job postingwill be an alumni. Hence it will have the foreign key that is user id

( one to many mapping of Administrator creates clubs)

Indexes:

Index#: Clustered

Columns: Job\_id

Justification:As the Primary key is Job\_id, it forms a clustered index on this table.

Index#: Non Clustered

Columns: Description, Company

Justification: this index is created with the name **Job\_info** with above mentioned columns as these are most frequently searched while looking for a job posting in the database

Highest normalization level: <1NF/2NF/3NF/**4NF**>

Table: **Event**

Columns:

Event\_id

Event\_name

Event\_Description

Organizing\_Club [Foreign Key; references **Club\_id** of **Club**]

Event\_date

Justification: **Event\_id** is the primary key for this table.**Organizing\_Club** is the foreign key from the club table ( one to many mapping of Clubs create Events). A event can be organised by only one club but participants can be anybody

Highest normalization level: <1NF/2NF/3NF/**4NF**>

Indexes:

Index#: Clustered

Columns: Event\_id

Justification:As the Primary key is Event\_id, it forms a clustered index on this table.

Index#: Non Clustered

Columns: Event\_name, Organizing\_Club

Justification: this index is created with the name **Event\_info** with above mentioned columns as these are most frequently searched while looking for an event posting in the database

Table: **Club** **Members**

Columns:

Club\_id [Foreign Key; references **Club\_id** of **Club**]

Alumni\_id [Foreign Key; references **Alumni\_id** of **Alumni**]

Primary Key: Combination of Club\_id and Alumni\_id

Justification: **Club\_id** and **Alumni\_id** are the foreign key for this table

This table is a cross-reference table representing membership to a club by an alumni for the many-to-many relationship.

Indexes:

Index#: Clustered

Columns: Combination of Club\_id and Alumni\_id

Justification:As the Primary key is the combination of Club\_id and Alumni\_id, it forms a clustered index on this table.

Highest normalization level: <1NF/2NF/3NF/**4NF**>

Table: **Follow**

Columns:

Follower\_id[Foreign Key; references **User\_id** of **User]**

Following\_id [Foreign Key; references **User\_id** of **User]**

Primary Key: Combination of Follower\_id and Following\_id

This table is a cross-reference table representing ownership of events by clubs for the many-to-many relationship

Highest normalization level: <1NF/2NF/3NF/**4NF**>

Indexes:

Index#: Clustered

Columns: Combination of Follower\_id and Following\_id

Justification:As the Primary key is the combination of Follower\_id and Following\_id, it forms a clustered index on this table.

Table: **Participants**

Columns:

Event\_id[Foreign Key; references **Event\_id** of **Event]**

User\_id[Foreign Key; references **User\_id** of **User]**

Primary Key: Combination of Event\_id and User\_id

This table is a cross-reference table representing many to many relationships of user joins events.

Highest normalization level: <1NF/2NF/3NF/**4NF**>

Indexes:

Index#: Clustered

Columns: Combination of Event\_id and User\_id

Justification: As the Primary key is the combination of Event\_id and User\_id, it forms a clustered index on this table.

View: **Follow\_Details**

Columns:

Follower\_name

Follower\_email

Following\_name

Following\_email

View: **Participants\_Details**

Columns:

Event\_name

Participants\_name

Participant\_email

View: Alumni**\_Details**

Columns:

Alumni\_id

Alumni\_Name

Alumni\_Email

Current\_work\_place

Address

Contact Number

Prior\_work\_place

Industry

Experience

Graduation\_year

View: Student**\_Details**

Columns:

Student\_name

Student\_Email

Stream

Degree

Stored Procedure:

To update Job Posting Due date in the Job postings Table

DELIMITER $$

CREATE PROCEDURE Job\_due\_date\_update(

IN Due\_date Datetime , IN J\_id int(255)

)

BEGIN

UPDATE Job\_Postings SET

JP\_Due\_date = Due\_date WHERE Job\_id = J\_id;

END $$

DELIMITER ;

CALL Job\_due\_date\_update('2019-08-03',2);

Trigger:

To update the modified date in Job Postings Table with every insert and update on Job Postings Table

CREATE TRIGGER Job\_posting\_update

BEFORE UPDATE ON Job\_Postings

FOR EACH ROW

SET new.JP\_last\_modified =current\_timestamp();

CREATE TRIGGER Job\_posting\_insert

BEFORE INSERT ON Job\_Postings

FOR EACH ROW

SET new.JP\_last\_modified =now();