**Team members:**

Alvin Joseph Ambooken: 2020357

Jatin Tyagi: 2020381

Pratyush Jain: 2020396

Shoumik Lodh: 2020407

**Scope of the Project**

We have created a messaging platform similar to WhatsApp or Telegram. Users can send messages to anyone once they search their unique email-id using the given search functionality. Groups or Direct messages are the two forms of chatting with each other. Further, users can star important messages that they can retrieve later as needed. Moreover users can also search messages in their groups or direct messages.

**Problem Statement and Stakeholders**

With existing messaging platforms becoming more complicated and feature rich day by day, their remains a need for a simpler messaging platform that is both intuitive and completes its functionality efficiently. Hence we created our messaging platform for users ranging anywhere from a child to an elder who wishes to message someone and is comfortable in maneuvering a website. The users can expect an easy to understand and intuitive interface for the platform. Users should be consciously aware of the personal information that they post in our service.

Another stakeholder of the messaging platform is the owner of the platform ie. the company themselves who are responsible for monitoring its users personal data.

Underlined: Primary Key

**Bold:** Foreign Key

**Entities:**

1. users(user\_id, status, name, e-mail, password)
2. chats(chat\_id, chat\_type)
3. messages(msg\_id, **chat\_id**, **sender\_id**, create\_time, content)
4. groups(gname, gdesc, **chat\_id**)

**Relations:**

1. Users\_has\_Chats(Many to many) (**user\_id,chat\_id**) from Users(user\_id) to Chats(chat\_id)

Attributes: Admin

1. User\_stars\_messages(many to many)(**user\_id,msg\_id,chat\_id**) from users(user\_id) to messages(msg\_id)
2. Users\_receive\_messages(many to many)(**user\_id,msg\_id,chat\_id**) from users(user\_id) to messages(msg\_id)
3. One to one relationship (Chats have groups) from chats(chat\_id) to groups(**chat\_id**)(weak entity)
4. One to many relationship (Chats contain messages) from chats(chat\_id) to messages(msg\_id **chat\_id**)
5. One to many relationship (User sends messages) from users(user\_id) to messages(msg\_id, **sender\_id**)

**Relational Schema:**

**Entities:**

1. users:

| user\_id | int | Primary key |
| --- | --- | --- |
| status | varchar | Default: “Hey there! I am using app.” |
| name | varchar | Not null |
| e-mail | varchar | Unique, Not null |
| password | varchar | Not null |

1. chats

| chat\_id | int | Primary key |
| --- | --- | --- |
| chat\_type | varchar | Not null |

1. groups ( Derived from chats entity) (Weak Entity)

| chat\_id | int | FK References chats, Primary key |
| --- | --- | --- |
| gdesc | varchar | - |
| gname | varchar | Not null |

1. messages

| msg\_id | int | Primary key |
| --- | --- | --- |
| chat\_id | int | FK References chats |
| sender\_id | int | FK References users |
| content | varchar | Not null |
| create\_time | datetime | Not null |

**Relations:**

1. users\_stars\_messages

| user\_id | int | Primary Key, FK References users |
| --- | --- | --- |
| chat\_id | int | Primary Key, FK References chats |
| msg\_id | int | Primary Key, FK References messages |

1. users\_receives\_messages

| user\_id | int | Primary Key, FK References users |
| --- | --- | --- |
| chat\_id | int | Primary Key, FK References chats |
| msg\_id | int | Primary Key, FK References messages |

1. users\_has\_chats

| user\_id | int | Primary Key, FK References users |
| --- | --- | --- |
| chat\_id | int | Primary Key, FK References chats |
| Admin | Tinyint | Not null |