Database Management System

Course Project

Project No: 2C

MockFriends Database

GithubLink: https://github.com/sudwebd/ClassroomFriends

S John S (15114060)

Shubham Garg (15114068)

Shubham Raj (15114069)

Sudhanshu Sambharya (15114071)

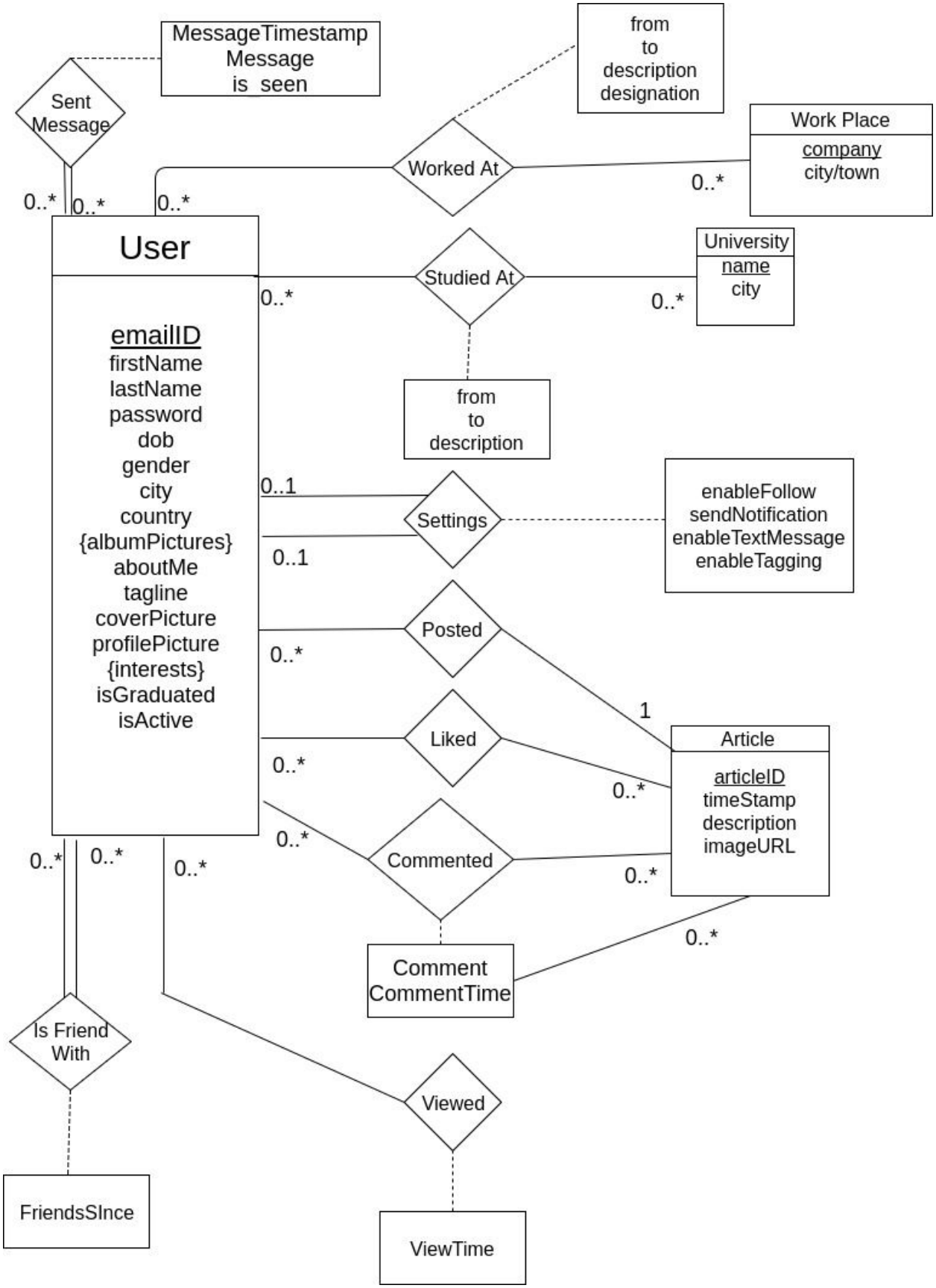
Ujjawal Prasad (15114074)

Before we begin with the project, here are the list of ASSUMPTIONS made while implementing the project.

ASSUMPTIONS:

1. **A person CANNOT send multiple messages at same point of time.**
2. **A person CANNOT like/comment/upload multiple contents at a given point of time.**
3. **A person CANNOT upload multiple pics in a single post.**
4. **A person might or might not have profile and cover pic.**
5. **A person might or might not like/comment on content.**
6. **A person can like/comment his/her content.**
7. **Relationships are limited to** Friends**.**
8. **A person might or might not comment same thing more than once.**
9. **A person might work in same company more than once on different designations but only at 1 designation at a given time.**
10. **“Share article” option is NOT available**
11. **Company names and university names are unique.**

Entity Relationship Diagram (ER Diagram)



Functional Dependencies & Candidate Keys

Since, albumPictures and interests are multi-valued attributes, these are already been broken to different tables for simplicity.

Functional Dependencies:

1. **User**
   (emailId, firstName, lastName, password, dob, gender, city, country, aboutMe, tagline, coverPicture, profilePicture, isGraduated, isActive)
   - emailId -> firstName, lastName, password, dob, gender, city, country, aboutMe, tagline, coverPicture, profilePicture, isGraduated, isActive

2. **Album** (emailId, pictureURL)
   - No FD
3. **Interests** (emailId, Interest)
   - No FD
4. **Friends** (emailId1, emailId2, friendsSince)
   - {emailId1, emailId2} -> friendsSince

5. **Article** (articleId, emailId, description, timestamp, imageURL)
   - articleId -> emailId, description, timestamp, imageURL

6. **University** (universityName, city)
   - universityName -> city

7. **Education** (emailId, universityName, from, to, description)
   - {emailId, universityName} -> from, to, description

8. **WorkPlace** (company, city)
   - company -> city

9. **Work** (emailId, company, designation, from, to, description)
   - {emailId, company, designation} -> from, to, description
   - {emailId, company, from} -> designation, to, description
   - {emailId, company, to} -> designation, from, description

10. **Settings**
    (emailId1, emailId2, enableFollow, sendNotification, enableTextMessage, enableTagging)
    - {emailId1, emailId2} -> enableFollow, sendNotification, enableTextMessage, enableTagging

11. **ArticleLiked** (emailId, articleId)
    - No FDs

12. **CommentedOn** (emailId, articleId, Comment, CommentTime)
    - {emailId, articleId, CommentTime} -> Comment

13. **Viewed** (emailId, articleId, ViewTime)
    - {emailId, articleId} -> ViewTime

14. **SentMessage** (emailId1, emailId2, message, messageTimeStamp, isSeen)
    - {emailId1, emailId2, messageTimeStamp} -> message, isSeen

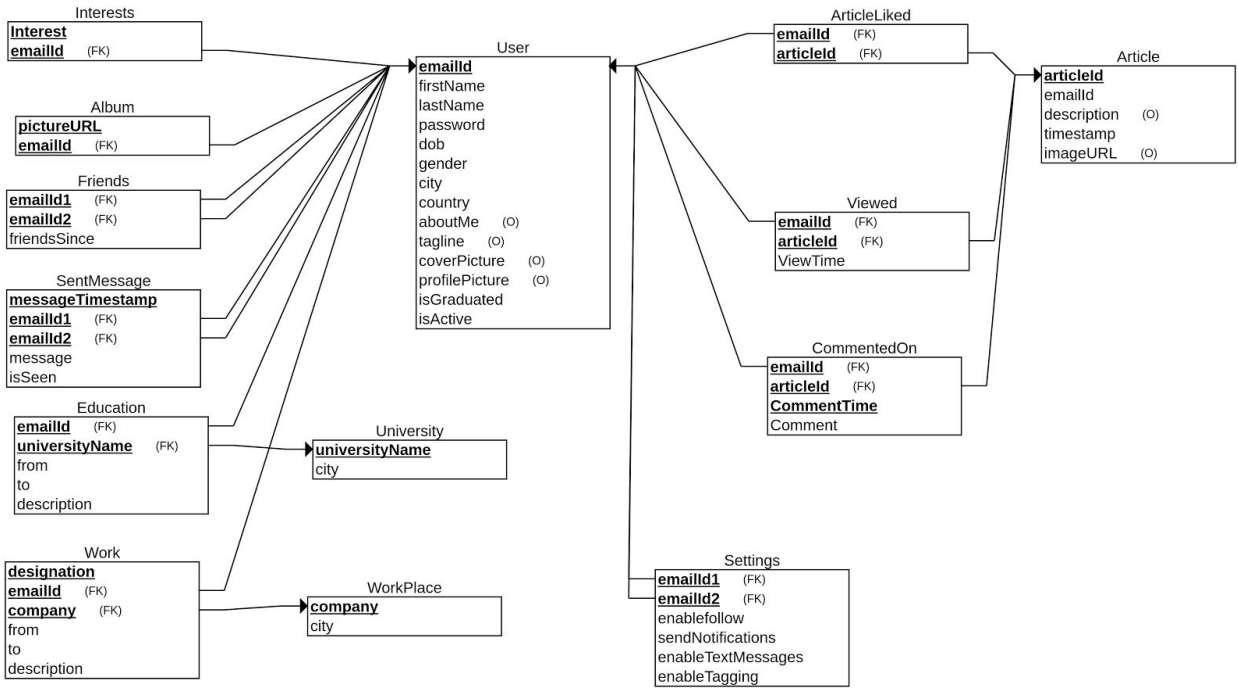
Candidate Keys

1. **User** : {emailId}
2. **Album** : {emailId, pictureURL}
3. **Interests** : {emailId, Interest}
4. **Friends** : {emailId1, emailId2}
5. **Article** : {articleId}
6. **University** : {universityName}
7. **Education** : {emailId, universityName}
8. **Workplace** : {company}
9. **Work** :
   {emailId, company, designation}, {emailId, company, from}, {emailId, company, to}
10. **Settings** : {emailId1, emailId2}
11. **ArticleLiked** : {emailId, articleId}
12. **CommentedOn** : {emailId, articleId, CommentTime}
13. **Viewed** : {emailId, articleId}
14. **SentMessage** : {emailId1, emailId2, messageTimeStamp}

Minimal Cover

Since there is no FD in any relation, in which LHS is not a candidate key, the above written dependencies are also the minimal cover for the schema.

Schema Diagram



1NF, 2NF, 3NF, BCNF 4NF, 5NF?

Observation: In all our tables, we have FDs of type X -> A, X ∈ Candidate Key.

1. **1NF:**
    a. Since all our columns are atomic, the tables are in 1NF.

2. **2NF:**

    a. Since X ∈ Candidate Key, no non-prime attribute can be dependent on a subset of candidate key.

3. **3NF:**

    a. Since X ∈ Candidate Key, there cannot be a functional dependency from non-prime attribute to non-prime attribute

4. **BCNF:**

    a. Since X ∈ Candidate Key ∈ Super Key, this means all our tables are in **BCNF.**

Since there are no multi-valued dependencies or Join Dependencies, the schema is in **5NF**