

Deliverable 4 – Project Phase 2 CSCE 5430 (Fall 2021)

Project Title - Sociogram (Web Application)

Team Name – Code6

Group Members –

1. Sathwik Gaddi
2. Tanvi Thirunathan
3. Satish Thammaneni
4. Sai Karthik koncherlakota
5. Raja Srinivas Nelluri
6. Sri Snigdha kotharu
7. Venkata Rajashekar Reddy Chintala
8. Putta Ramya Priya

Version 3.0

Document History

Version	Date	Author	Comments
3.0	29 th Oct, 2021	Code6	Deliverable 4

1. Requirements -

In this second phase of our project, Sociogram, we are adding additional features to the previous version of the website like Discussion Forums, additional UI changes. For this we're gathering requirements that will be utilized to create the project's second phase.

1.1 **REQ1** – User must register or signup to their account to start the discussion forums.

1.2 **REQ2** – User should have access to Discussion forums by selecting the Discussion Forums on the Top Nav Bar.

1.3 **REQ3** - CREATE DISCUSSION will allow user to create a new Discussion. User will have access to a form which contains the field for Question Title and Question Description.

1.4 **REQ4** – After submission of the form, the following discussion should be available to answer by all the users upon successful authentication and validation.

1.5 **REQ5** – ALL DISCUSSIONS will allow user to view all the available discussions and answer them.

1.6 **REQ6** – Users can be able to either Upvote or Downvote to an answer for any discussion available.

1.7 **REQ7** – MY DISCUSSIONS will allow user to view their discussion posted previously and they can find the answers they received.

As per the previous development phases plan, we must include Communication feature i.e., a user must be able to text with other users. But due the lack of knowledge by all the members of the team and due to the time constraints, we are not going to include this feature in the whole project. Below is the updated development phases plan.

1.8 Development Phases:

We are going to develop our application in three different phases by adding of some extra features in each phase.

Phase 1: In this phase we are going to create a website for sociogram, and we are going to add three crucial features for sociogram.

1. **User Profile:** Create a login and sign-up page to get personalized user profile. The passwords are hashed using bcrypt algorithm to ensure the privacy of the user.
2. **Upload data:** Providing access to the clients to upload posts, media, and articles. Include a feature to add privacy for each of their post.
3. **Find friends:** Find friends with similar habits and hobbies with the help of suggestions. User will provide their interests when they register to find their identical.

Requirements

REQ1: User login page should have a form with username and password fields, a login button, and a signup up button which on successful validation redirects to home page of the application and error message on Invalid credential confirmation. A Signup button should be added for new users to register for the application.

REQ2: Upon successful validation for login, a json web token should be sent from Node Js server to frontend to authorize user throughout the session.

REQ3: Signup page should have a form with First name, Last name, Username, password, confirm password, and a dropdown for interests and hobbies. On successful form validation by JavaScript the page should be redirected to Login page.

REQ4: Home Page of the application should have buttons to upload a new post, discussion forum, find friends and logout.

REQ5: Upload post screen should contain a form with fields for tagline, upload post and privacy.

REQ6: Find friends page should contain a list view to display people with similar interests.

Phase 2: In this phase we are adding two features they are:

1. Discussion forums feature: Open community page where users can raise queries, answer other queries, discuss with a whole lot of people.
2. Update Password: User can be able to access the account and update their password.

Requirements

REQ1: Upon successful json web token authorization, user should be able to access Discussion Forums.

REQ2: Discussion forum page should consist of a list view of all the discussions available in the application. **REQ3:** Upvote and Downvote buttons should be available for each discussion.

REQ4: Update Password page should contain a form for Previous and new passwords.

REQ5: After submitting, successful validation of data and authentication of user, Password of the current user should be changed.

Phase 3: Assigning booking ticket feature to the application. In this feature user can book tickets such as movies and events etc. Along with that we will fix bugs raised in our application and concentrate to improve performance will consider in this phase.

Requirements:

REQ1: Booking page should contain a list view of all the events/movies.

REQ2: Locust should be used to perform load testing.

2. UML Design –

2.1 Class Diagram

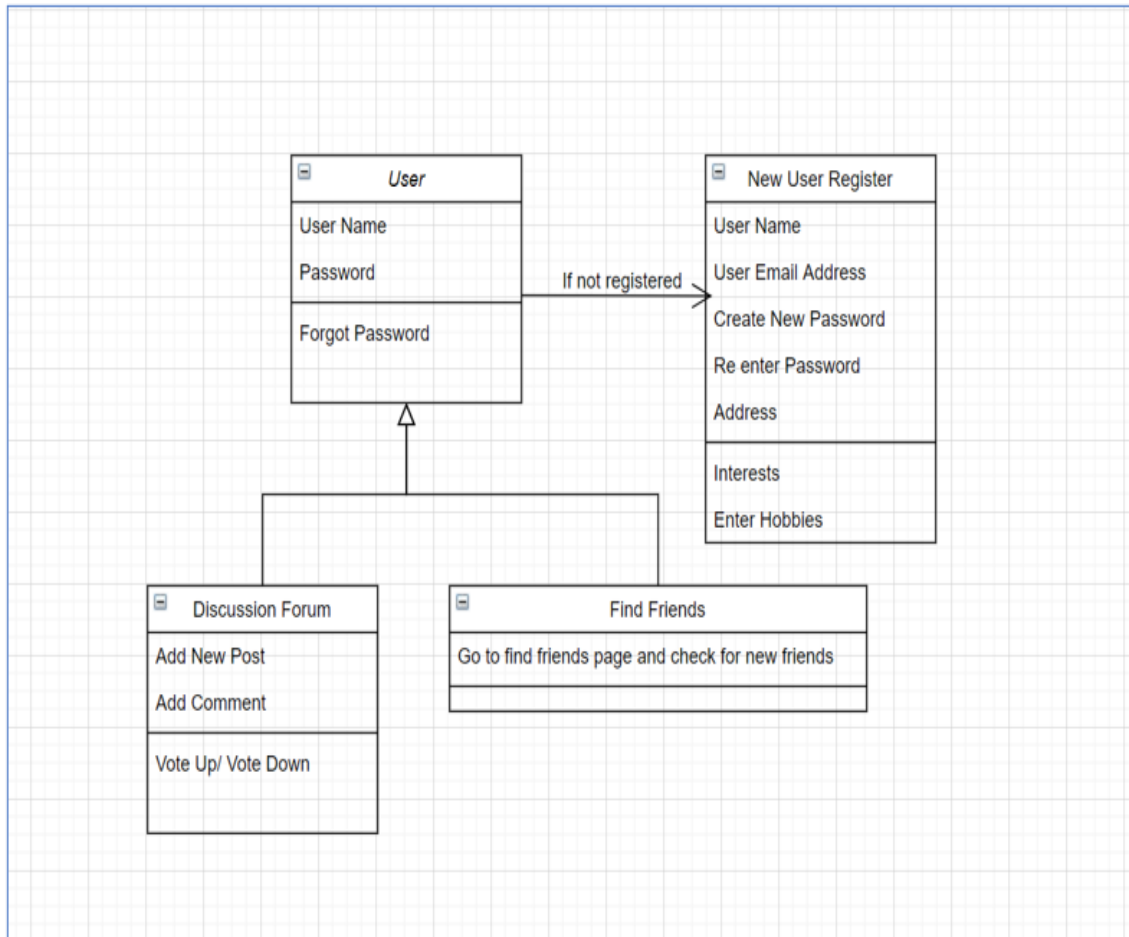


Fig 2.1 Class Diagram for Discussion Forum

2.2 Sequence Diagram

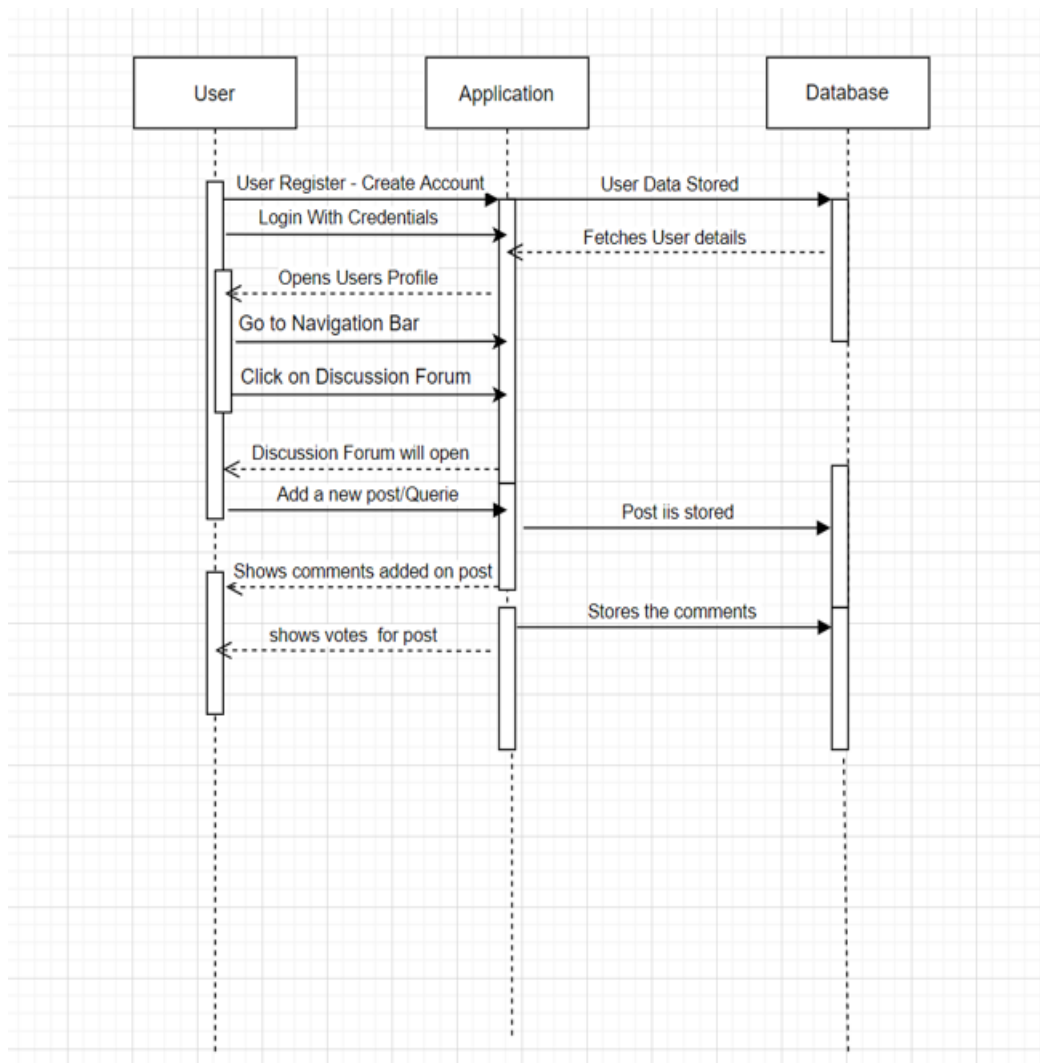


Fig 2.2 Sequence Diagram for Discussion Forum

2.3 Use Cases Diagram (Non-Error and Error Cases)

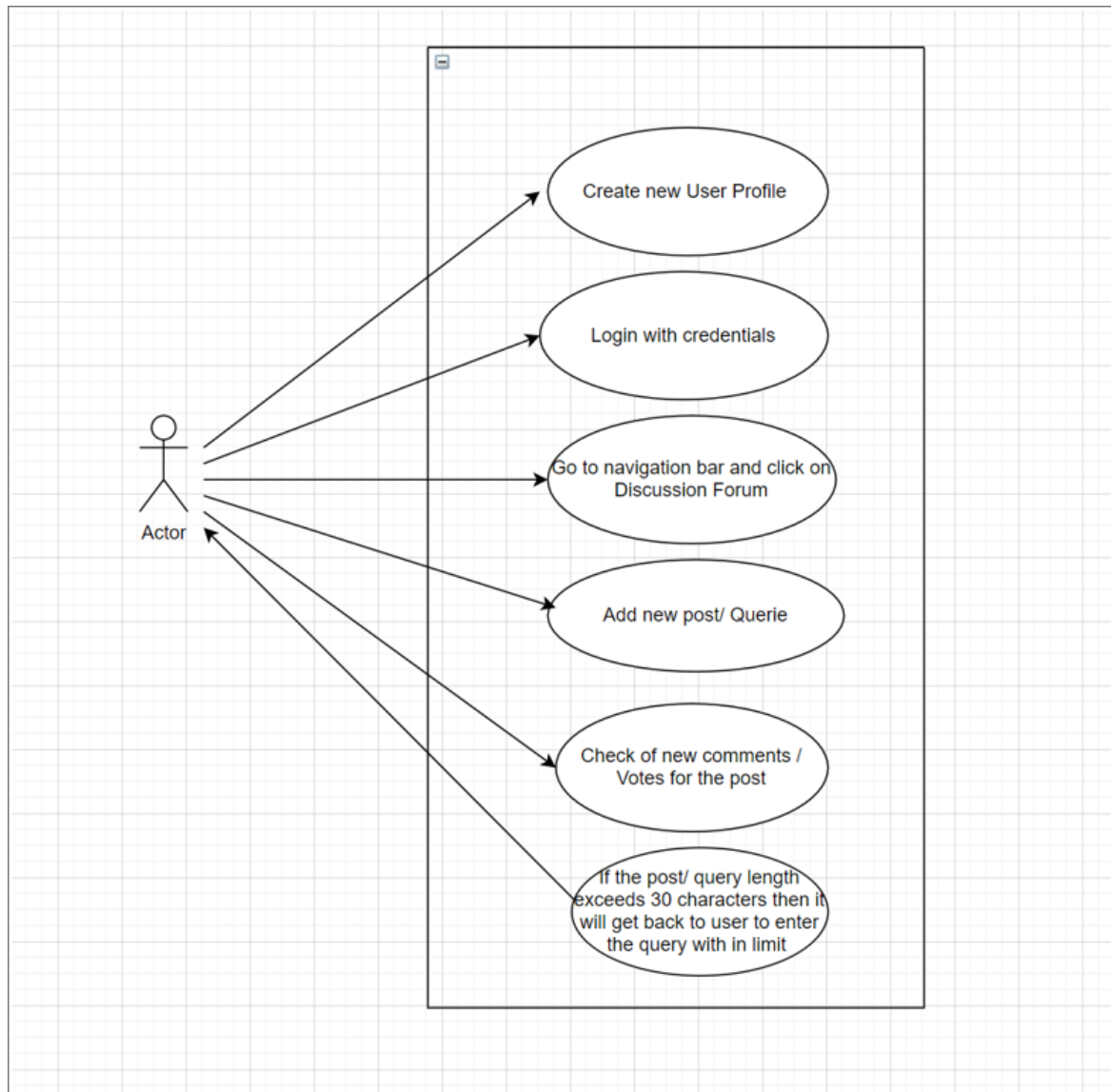


Fig 2.3 Use Case Diagram for Discussion Forum

3. Test Cases –

3.1 Discussion forum – All Discussions

Description: A user should be able to view all discussions.

Precondition: A discussion is to be posted for user to view them.

Assumption: A discussion is already created, and supported browser is being used.

Test Steps:

- Navigate to Discussions on the Top Navbar.
- Click on “All Discussions”.

Expected Result: User must be successfully able to view all the discussions.

3.2 Discussion forum – My Discussions

Description: A user should be able to view his/her discussions that are created, and answers received for it.

Precondition: A discussion should be created by user to view them.

Assumption: A user has created a discussion and supported browser is being used.

Test Steps:

- Navigate to Discussions.
- Click on “My Discussions”.

Expected Result: User must be successfully able to view all the discussions created by him/her, number of answers and those answers that are posted by others.

3.3 Discussion forum – Create New Discussion

Description: A user must be able to successfully create a new discussion.

Precondition: The user must be logged in to access this feature.

Assumption: A supported browser is being used.

Test Steps:

- Navigate to Discussions.
- Click on “Create New Discussion”.
- Type “Question Title”.
- Type “Question Description”.
- Click on “Create Discussion”.

Expected Result: User must be successfully able to create a new discussion.

3.4 Discussion forum – Delete Discussion

Description: A user must be able to successfully delete a discussion form created by him/her.

Precondition: The user must be logged in to access this feature.

Assumption: A discussion which was already created by the user and supported browser is being used.

Test Steps:

- Navigate to Discussions.
- Click on “My Discussion”.
- Click on a Discussion that is created previously and user wants to delete.
- Click on “Delete”.

Expected Result: User must be successfully able to delete a discussion form created previously and view message “No Discussions Posted Yet!!” if the one which is deleted is the only discussion existing.

3.5 Change Password

Description: A user must be able to successfully change his/her password.

Precondition: The user must be logged in to access this feature.

Assumption: A user is willing to change his/her password but, still knows his old password.

Test Steps:

- Navigate to “Change password”.
- Type “Current Password”.
- Type “New Password”.
- Type same password “Confirm New Password”.
- Click on “Change Password”.

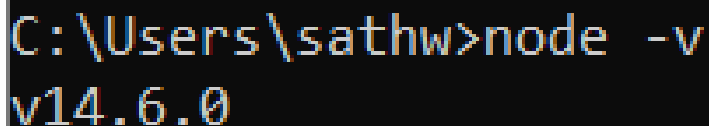
Expected Result: User must be successfully able to change his/her password.

4. User manual -

4.1 Installing NodeJS

To get started installing the NodeJS, follow these steps.

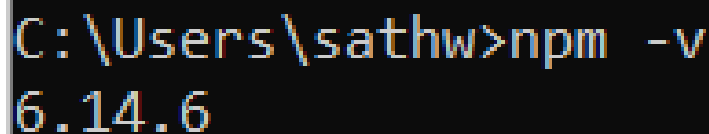
- a. Download NodeJS msi file from [here](#).
- b. Run the msi file.
- c. Check for installations of node and npm in command prompt using following commands:
- d. `node -v` (Upon successful installation you will see the following output)



```
C:\Users\sathw>node -v
v14.6.0
```

A terminal window with a black background and white text. The prompt is 'C:\Users\sathw>' and the command 'node -v' has been executed, resulting in the output 'v14.6.0'.

- e. `npm -v`. (Upon successful installation you will see the following output)



```
C:\Users\sathw>npm -v
6.14.6
```

A terminal window with a black background and white text. The prompt is 'C:\Users\sathw>' and the command 'npm -v' has been executed, resulting in the output '6.14.6'.

4.2 Installing Angular Cli

- f. Use npm to install angular cli.
- g. Use command `npm install -g @angular/cli`

h. Use the following to check for the successful installation.

Ng -version

i. You will see the following output.

```
C:\Users\sathw>ng --version

Angular CLI

Angular CLI: 10.0.4
Node: 14.6.0
OS: win32 x64

Angular:
...
Ivy Workspace:

Package                                Version
-----
@angular-devkit/architect             0.1000.4
@angular-devkit/core                  10.0.4
@angular-devkit/schematics            10.0.4
@schematics/angular                  10.0.4
@schematics/update                    0.1000.4
rxjs                                  6.5.5
```

4.3 Installing MongoDB

- Download the in premises local version of MongoDB Community Server msi file from [here](#).
- To start the MongoDB server, run file
"C:\Program Files\MongoDB\Server\4.2\bin\mongod.exe"
- This starts the main MongoDB database process. The waiting for connections message in the console output indicates that the [mongod.exe](#) process is running successfully.

5. Compiling Instructions –

5.1 Getting the source code

- a. Get the source code from the git repository
<https://github.com/sathwikgaddi/Code6.git>
- b. Open command prompt and navigate to project source code folder.
- c. Run the command **npm install** to install all the dependencies required to run the program.
- d. Run command **ng serve** to start the angular server. You will see the following output on successful start of the server.

```
C:\myProjects\mean\mean\sociogram>ng serve
10% building 3/3 modules 0 active i @wds@: Project is running at http://localhost:4200/webpack-dev-server/
i @wds@: webpack output is served from /
i @wds@: 404s will fallback to //index.html

chunk {main} main.js, main.js.map (main) 148 kB [initial] [rendered]
chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 264 kB [initial] [rendered]
chunk {runtime} runtime.js, runtime.js.map (runtime) 6.15 kB [entry] [rendered]
chunk {styles} styles.js, styles.js.map (styles) 177 kB [initial] [rendered]
chunk {vendor} vendor.js, vendor.js.map (vendor) 6.98 MB [initial] [rendered]
Date: 2021-10-14T17:26:33.964Z - Hash: 6ad83e29f464c1521ce8 - Time: 16299ms
** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **
i @wdm@: Compiled successfully.
```

- e. Navigate to backend folder in command prompt and use command **nodemonserver.js** to start the server of the backend application.
- f. Open browser and search for **localhost:4200** to open our application.

6 Code Inspection Feedback –

Partner Group Name - SOUL-CYNICS

Project Name – Defining & Predicting the success of B2B sales

Members –

1. Rishwanth Reddy Baddipadaga
2. Meghana Thimmapuram
3. Preethi Mudireddy
4. Divya Kumbham
5. Neha Geereddy
6. Sri Harsha Swaraj Nadendla
7. Roja Kamble
8. Sai Nikhil Chitturi

Feedback by Partner Team (SOUL-CYNICS)

1. Code Formatting –

The partner team stated that the source code has blockers which means that the commented code still exists in the source code document.

Code6: We agree to the above point as practising to remove the commented part of the source code file helps remove the blockers while doing the code inspection and while debugging the errors if any.

We will further store the sub version of the code using the version control systems like SVN. By doing this, we will be able to retrieve the commented section if required any further in the project implementation.

7 Things Accomplished –

Earlier in phase 1 of the project development, we have implemented the below features

- a. Sign up
- b. Login
- c. Uploading media

In the current phase, we are developing the discussion forum feature along with enhancing the front-end part of the website.

We have excluded the feature “Messaging” because of the time constraints. This has been included in the requirements section under change in phase 2 development plan.

The feedbacks given by TA’s and partner team are keenly understood such that we keep in mind the points that are to be improvised and corrected.

Overall, we are keeping track of the project in a well-mannered way keeping all the aspects of it as planned.

8 Member Contribution Table –

Member Name	Contribution Description	Overall Contribution (%)	Note (If Applicable)
Sathwik Gaddi	User manual, Formatting	12.5 %	
Tanvi Thirunathan	Code Inspection Feedback, Member contribution Table, Formatting	12.5 %	
Satish Thammaneni	Things Accomplished	12.5 %	

Sai Karthik koncherlakota	Test Cases	12.5 %	
Raja Srinivas Nelluri	Instructions to compile and run	12.5 %	
Sri Snigdha kotharu	UML design	12.5 %	
Venkata Rajashekar Reddy Chintala	Requirements	12.5 %	
Putta Ramya Priya	UML design	12.5 %	