

## **Assignment 1**

**[Class]**

**[Student Name]**

**[Student ID]**

## Contents

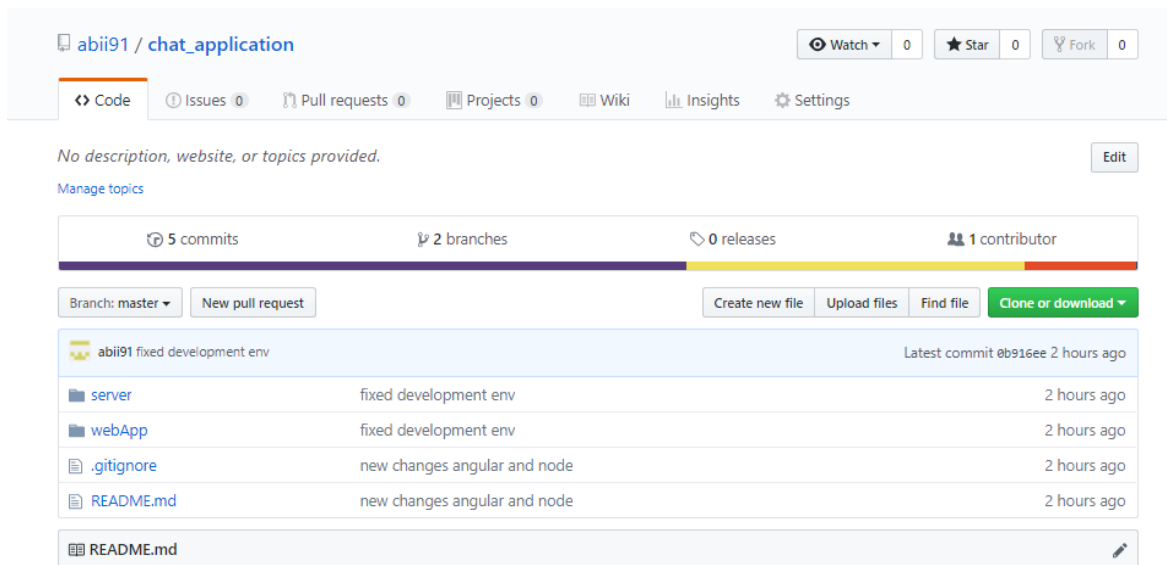
Organization of Git repository.....	3
Data structures used in the client and server.....	5
Division of responsibilities between client and server.....	5
List of routes.....	5
Angular architecture:.....	8

# Chat System

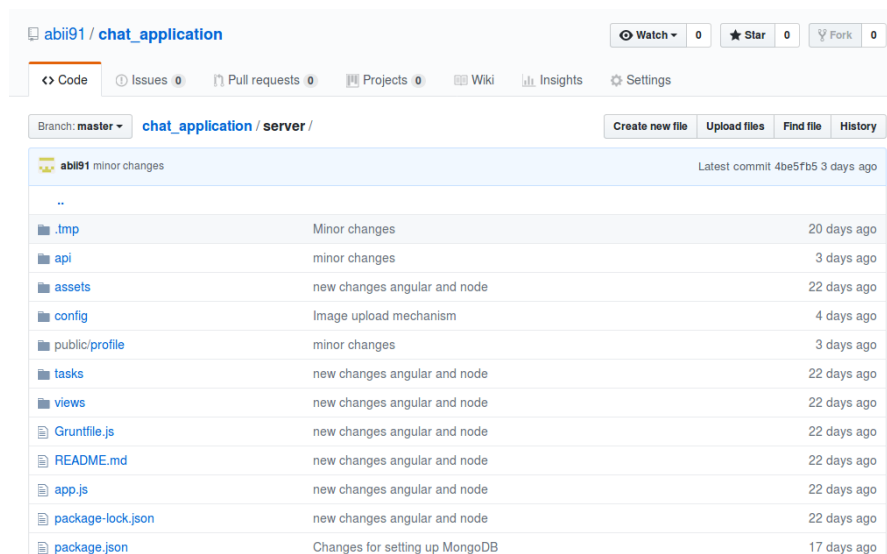
## Organization of Git repository

- I created a new repository in Git-hub as “chat\_application”. I have used to commit my changes. Created separated branch for each assignment and created PR and merge this PR. Added node modules and bower components in gitignore

GitHub repository is organized as follows



- The server folder contains following files/folders



- The webApp folder contains following folders/files

Branch: master chat\_application / webApp /

Create new fileUpload filesFind fileHistory

abil91 profile picLatest commit 860ae19 3 days ago

..

config

new changes angular and node

22 days ago

docs

new changes angular and node

22 days ago

gulp

new changes angular and node

22 days ago

src

profile pic

3 days ago

.travis.yml

new changes angular and node

22 days ago

CHANGELOG.md

new changes angular and node

22 days ago

LICENSE.txt

new changes angular and node

22 days ago

README.md

Minor changes

20 days ago

bower.json

Image upload mechanism

4 days ago

gulpfile.js

new changes angular and node

22 days ago

package-lock.json

Image upload mechanism

4 days ago

package.json

Image upload mechanism

4 days ago

preview.html

new changes angular and node

22 days ago

- Further the chat\_application > webApp > src contains html files.

abii91 / chat\_application

Watch

0

Star

0

Fork

0

Code

Issues0

Pull requests0

Projects0

Wiki

Insights

Settings

Branch: master

chat\_application / webApp / src /

Create new file

Upload files

Find file

History

abii91 new changes angular and node

Latest commit b48a400 3 hours ago

..

app

new changes angular and node

3 hours ago

assets

new changes angular and node

3 hours ago

sass

new changes angular and node

3 hours ago

404.html

new changes angular and node

3 hours ago

auth.html

new changes angular and node

3 hours ago

index.html

new changes angular and node

3 hours ago

reg.html

new changes angular and node

3 hours ago

## **Data structures used in the client and server**

At server side the following entities being used

- 1) Users
- 2) Groups
- 3) Channels
- 4) Roles

At web app/Client side following use cases added

- 1) Chat
- 2) My Groups -> Channels
- 3) Users
- 4) Settings -> To add groups, channels and assign users

## **Division of responsibilities between client and server**

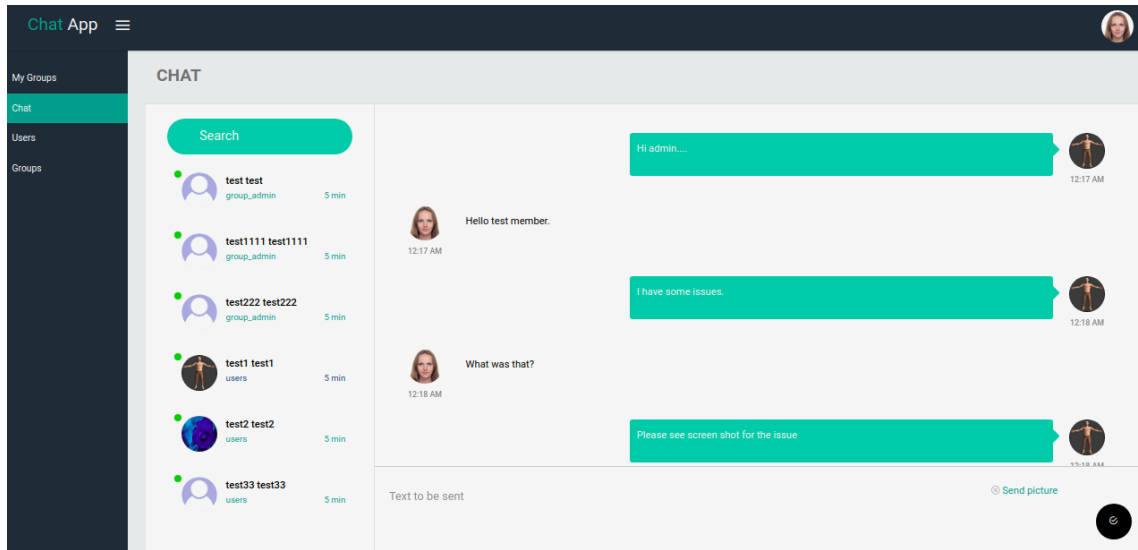
The server is exposed by REST API performing the basic CRUD operations. The web app enables the interaction with end users. Get data from user and sends it to the server through REST API. There are two roles Super Admin and Group Admin. The Group Admin can create users, groups and channels. He can assign users to channels and channels to groups. The Super admin can create user with group admin role and all other roles by group admin. User can change his profile picture and send chat messages to any user and channels.

**Server:** Server is being used as a REST api. The server receives request from client, authenticate the request, perform specific actions, do db operations and send JSON response back to client.

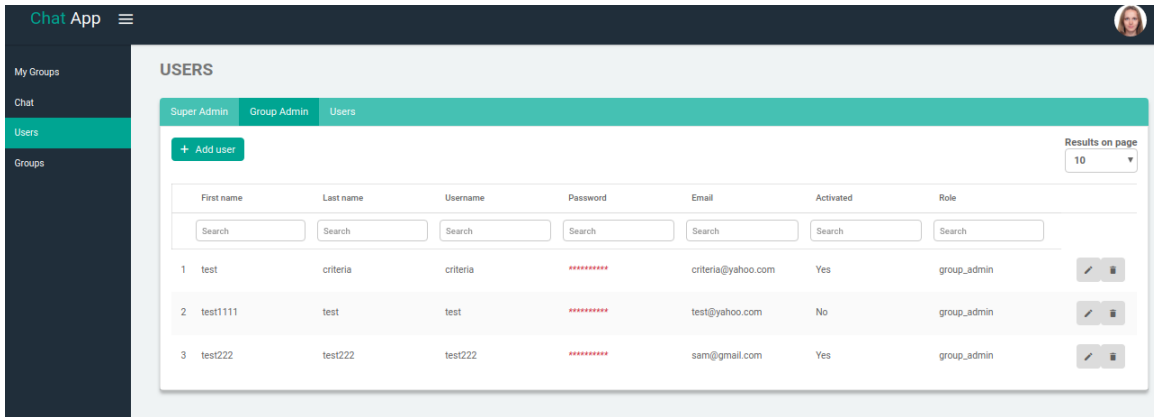
**Client:** Client providing the interface to the users. The users perform action on site and get result on UI.

Following is the structure of web app.

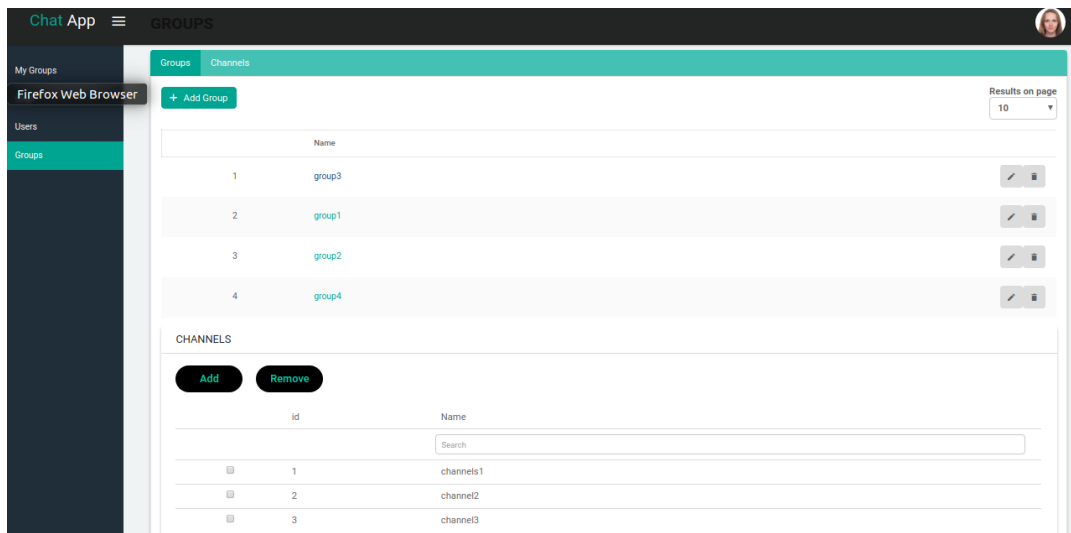
- 1.** The chat screen looks like below



## 2. The users screen



## 3. The Groups screen



## List of routes

Following are the routes/urls used.

- post /users/:id/uploadPhoto
- get /users
- get /getChatUsers
- get /getGroupChannels
- get /channels/getUnassignedChannels
- post /channels/assignChannels
- post /channels/removeGroupChannels
- get /channels/getChannelUsers
- post /channels/assignUsers
- post /channels/removeChannelUsers
- get /channels/getUserChannels
- get /groups/getUserGroups
- post /chat/sendChat
- post /chat/getUserChat
- post /chat/:id/uploadPhoto

- **Route to my Sign Up page**

Code is mentioned below

```
angular.module('BlurAdmin.pages.authSignUp', []) .config(routeConfig);
```

```
/** @ngInject */  
function routeConfig($stateProvider) {  
  $stateProvider  
    .state('authSignUp', {  
      url: '/authSignUp',  
      templateUrl: 'app/pages/authSignUp/authSignUp.html',  
      title: 'My Page',  
      controller: 'authSignUpCtrl',  
      sidebarMeta: {  
        order: 800,  
      },  
      authenticate: false  
    });  
}
```

- **Route to my Groups page**

```
angular.module('BlurAdmin.pages.myGroups',  
['BlurAdmin.pages.channelDetails'])
```

```

.config(routeConfig);
/** @ngInject */
function routeConfig($stateProvider) {
  $stateProvider
    .state('main.myGroups', {
      url: '/myGroups',
      templateUrl: 'app/pages/myGroups/myGroups.html',
      title: 'My Groups',
      controller: 'myGroupsCtrl',
      sidebarMeta: {
        icon: 'ion-person',
        order: 98,
      },
      data: {
        permissions: {
          only: ['users']
        }
      },
      authenticate: true
    });
}
})();

```

- **Route to chat page**

```

angular.module('BlurAdmin.pages.chat', [])
.config(routeConfig);

/** @ngInject */
function routeConfig($stateProvider) {
  $stateProvider
    .state('main.chat', {
      url: '/chat',
      templateUrl: 'app/pages/chat/chat.html',
      title: 'Chat',
      sidebarMeta: {
        icon: 'ion-person',
        order: 100,
      },
      data: {
        permissions: {
          only: ['super_admin', 'group_admin', 'users']
        }
      },
      authenticate: true
    });
}

})();

```



- **Route to chat page**

Code is mentioned below

```
angular.module('BlurAdmin.pages.users', [])
.config(routeConfig);

/** @ngInject */
function routeConfig($stateProvider) {
  $stateProvider
    .state('main.users', {
      url: '/users',
      templateUrl: 'app/pages/users/users.html',
      title: 'Users',
      controller: 'usersCtrl',
      sidebarMeta: {
        icon: 'ion-person',
        order: 102,
      },
      data: {
        permissions: {
          only: ['super_admin', 'group_admin']
        }
      },
      authenticate: true
    });
}
```

### **Angular architecture:**

Angular architecture like components, directives, models, routes, state provider, services, modules are implemented in chat system.

An AngularJS module defines an application. Some of the application modules are mentioned below:

- BlurAdmin.pages.config
- BlurAdmin.pages.main
- BlurAdmin.pages.myGroups
- BlurAdmin.pages.users
- BlurAdmin.pages.profile
- BlurAdmin.pages.settings
- BlurAdmin.pages.chat
- BlurAdmin.pages.authSignIn
- BlurAdmin.pages.authSignUp
- BlurAdmin.pages
- BlurAdmin.services
- BlurAdmin.directives

AngularJS directives are extended HTML attributes with the prefix ng-. Some of the directives used in application modules are mentioned below:

- ng-show
- ng-model
- ng-options
- ng-repeat
- ng-click
- ng-controller
- ng-submit
- ng-change

Some of the custom directives are as follows.

- File-upload

Some of the custom services are as follows.

- GeneralFactory
- generalHelper
- ChatRoles
- httpInterceptor