Do connect

Capstone Project
C2-G10

Mentor:Ashutosh Trivedi

Capstone Project

Group – CI G10

Group members

1. Ankit Dwivedi
2. Rohan Rajendra Bagul
3. Ashruba Lahu Bhore
4. Shubham Kumar Singh
5. Rajuladevi Pavan srivatsa
6. Hemanth Akurathi

Mentor:Ashutosh Trivedi

Problem Statement

- Build an end to end popular Q/A Web Application where technical questions can be asked and answered.
- Application can be accessed by two roles, User and Admin
- Admin has all the privilages like creating, updating, retrieving and deleting questions and answers as for User, but has additional privilages like approving Q/A and deleting Users.

System specifications

Frameworks:

- Angular
- Java Spring Boot

Languages:

- Type Script
- Java
- MySql

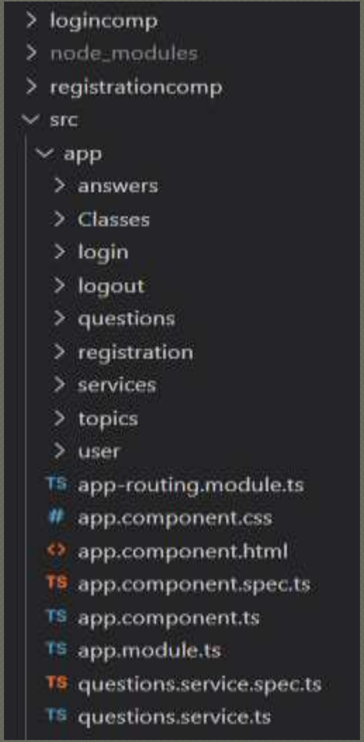
IDE:

- Eclipse
- VS code

What is Angular ?

- AngularJS is an MVC framework for browser based apps. It is a open source and originally developed at Google.
- The goal is building CRUD style business applications.
- Use declarative programming for UI and imperative programming for the logic.
- Supports modern desktop and mobile applications.

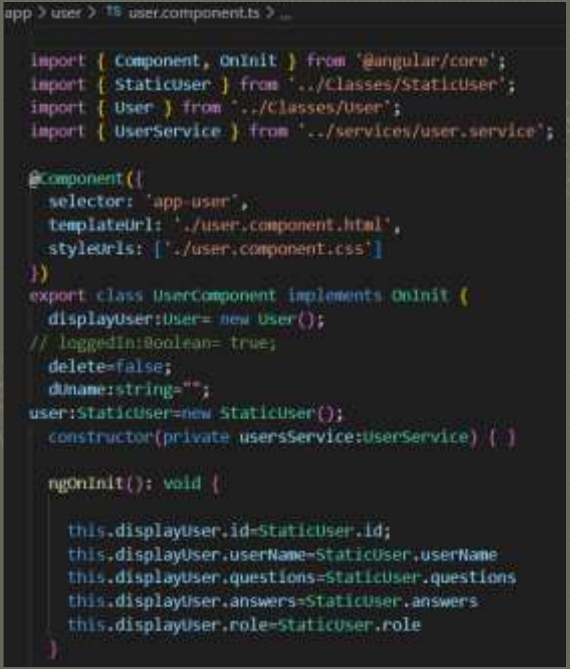
Components(frontend)



```
> logincomp
> node_modules
> registrationcomp
v src
  v app
    > answers
    > Classes
    > login
    > logout
    > questions
    > registration
    > services
    > topics
    > user
    TS app-routing.module.ts
    # app.component.css
    <> app.component.html
    TS app.component.spec.ts
    TS app.component.ts
    TS app.module.ts
    TS questions.service.spec.ts
    TS questions.service.ts
```

- **There are 4 components in the frontend :**
- User
- Question
- Answer
- Admin
- Image_model

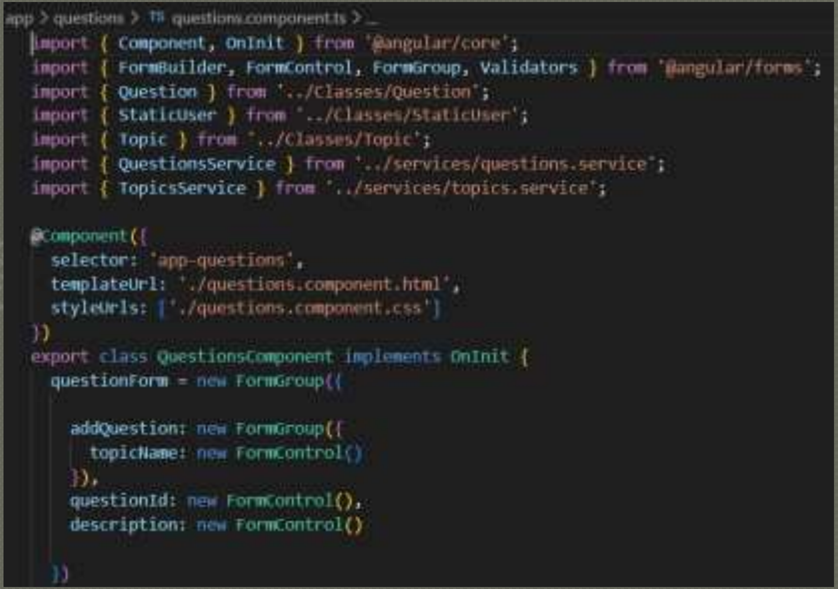
User Component



app > user > TS user.component.ts > ...

```
import { Component, OnInit } from '@angular/core';
import { StaticUser } from '../Classes/StaticUser';
import { User } from '../Classes/User';
import { UserService } from '../services/user.service';

@Component({
  selector: 'app-user',
  templateUrl: './user.component.html',
  styleUrls: ['./user.component.css']
})
export class UserComponent implements OnInit {
  displayUser:User= new User();
// loggedIn:Boolean= true;
  delete=false;
  dUname:string="";
user:StaticUser=new StaticUser();
  constructor(private usersService:UserService) { }

  ngOnInit(): void {

    this.displayUser.id=StaticUser.id;
    this.displayUser.userName=StaticUser.userName
    this.displayUser.questions=StaticUser.questions
    this.displayUser.answers=StaticUser.answers
    this.displayUser.role=StaticUser.role
  }
```

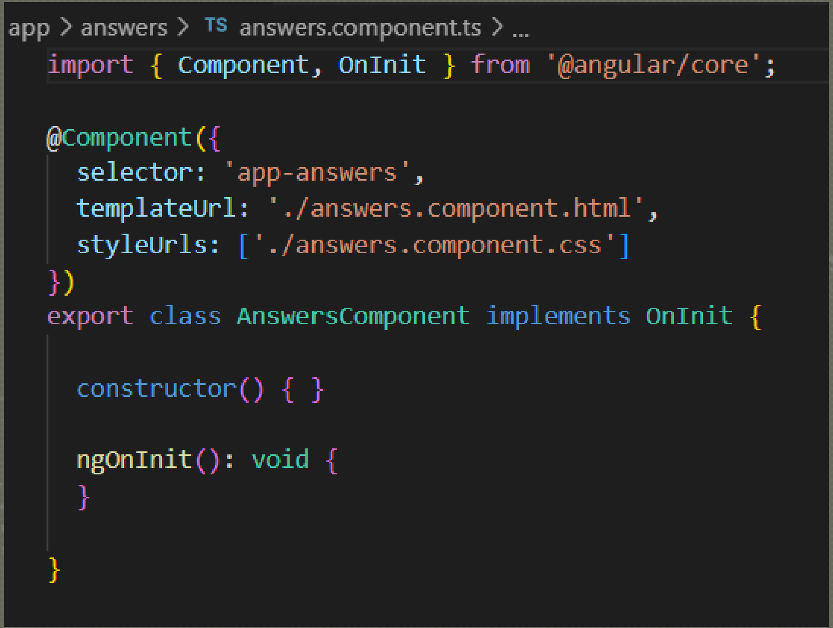
Question Component



app > questions > TS questions.component.ts > ...

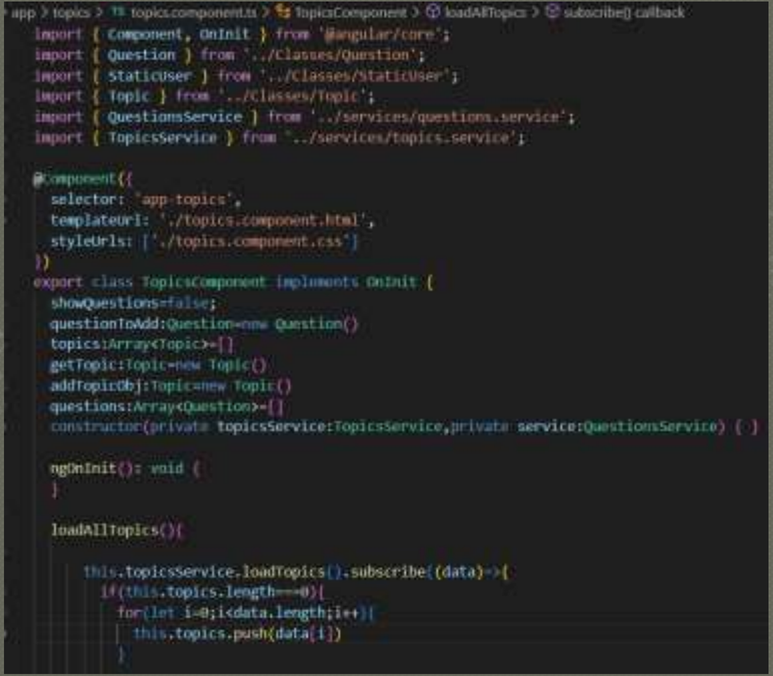
```
import { Component, OnInit } from '@angular/core';
import { FormBuilder, FormControl, FormGroup, Validators } from '@angular/forms';
import { Question } from '../Classes/Question';
import { StaticUser } from '../Classes/StaticUser';
import { Topic } from '../Classes/Topic';
import { QuestionsService } from '../services/questions.service';
import { TopicsService } from '../services/topics.service';

@Component({
  selector: 'app-questions',
  templateUrl: './questions.component.html',
  styleUrls: ['./questions.component.css']
})
export class QuestionsComponent implements OnInit {
  questionForm = new FormGroup({

    addQuestion: new FormGroup({
      topicName: new FormControl()
    }),
    questionId: new FormControl(),
    description: new FormControl()

  })
```

Answer Component


app > answers > TS answers.component.ts > ...

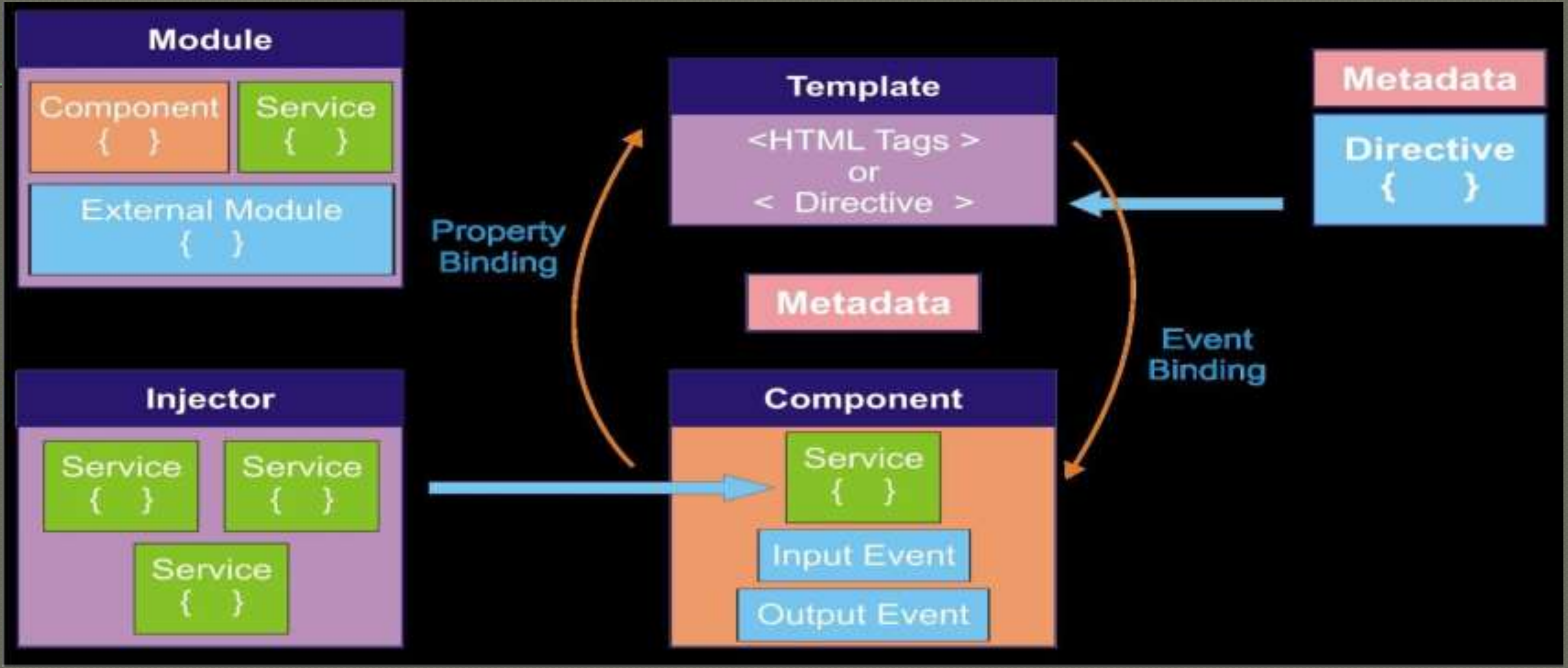
```
import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'app-answers',
  templateUrl: './answers.component.html',
  styleUrls: ['./answers.component.css']
})
export class AnswersComponent implements OnInit {

  constructor() { }

  ngOnInit(): void {
  }

}
```

Topic Component


app > topics > TS topics.component.ts > TopicsComponent > loadAllTopics > subscribe() callback

```
import { Component, OnInit } from '@angular/core';
import { Question } from '../Classes/Question';
import { StaticUser } from '../Classes/StaticUser';
import { Topic } from '../Classes/Topic';
import { QuestionsService } from '../services/questions.service';
import { TopicsService } from '../services/topics.service';

@Component({
  selector: 'app-topics',
  templateUrl: './topics.component.html',
  styleUrls: ['./topics.component.css']
})
export class TopicsComponent implements OnInit {
  showQuestions=false;
  questionToAdd:Question=new Question()
  topics:Array<Topic>=[]
  getTopic:Topic=new Topic()
  addTopicObj:Topic=new Topic()
  questions:Array<Question>=[]
  constructor(private topicsService:TopicsService,private service:QuestionsService) { }

  ngOnInit(): void {
  }

  loadAllTopics(){

      this.topicsService.loadTopics().subscribe((data)=>{
        if(this.topics.length===0){
          for(let i=0;i<data.length;i++){
            this.topics.push(data[i])
          }
```

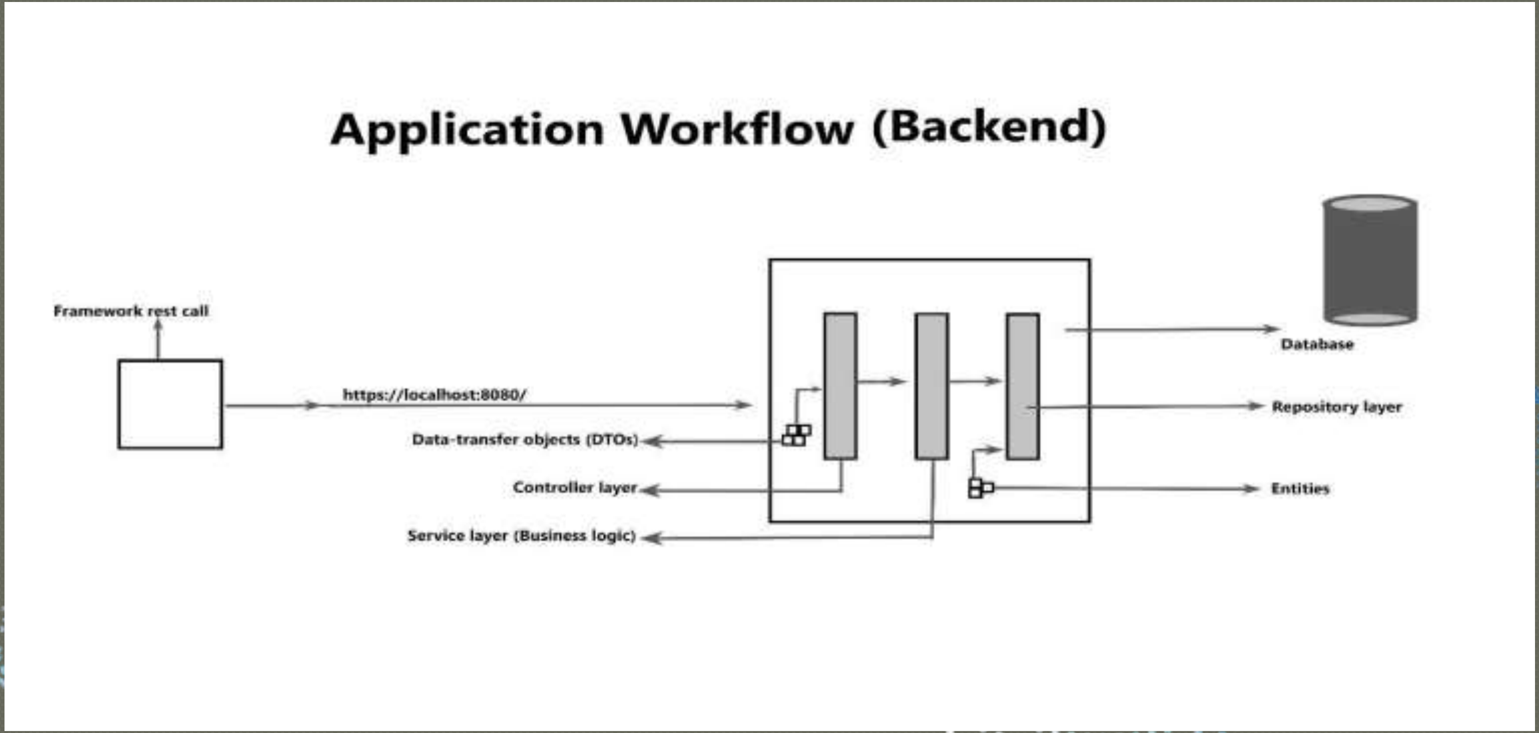
Architecture



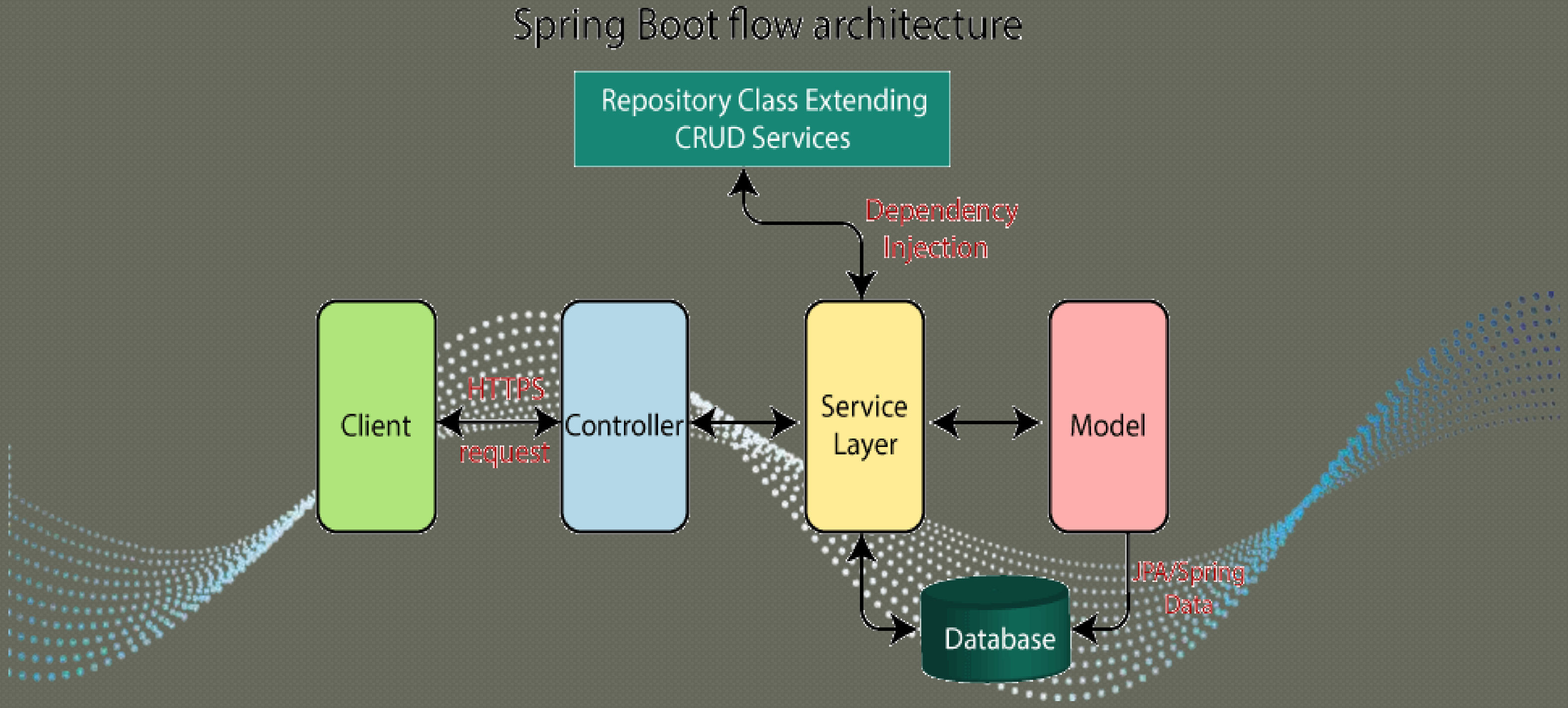
Concepts and Terminology

Concepts	Terminology
Template	HTML with additional markup used to describe what should be displayed
Directive	Allows a developer to extend HTML with his own elements and attributes
scope	Context where the model data is stored so that templates and controllers can access
Compiler	Processes the template to generate HTML for the browser
Data Binding	Syncing of data between the Scope and the HTML(two-way)
Dependency Injection	Fetching and setting up all the functionality needed by a component
Module	A container for parts of an application
Service	Reusable functionality available for any view

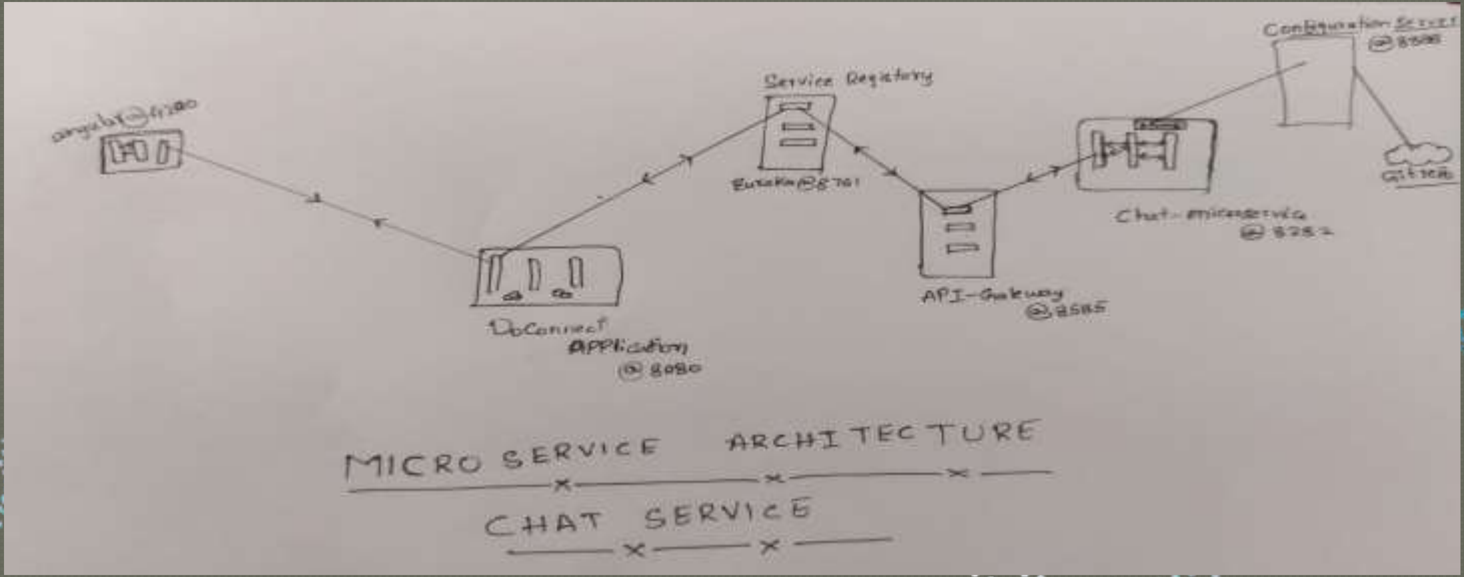
Backend Workflow



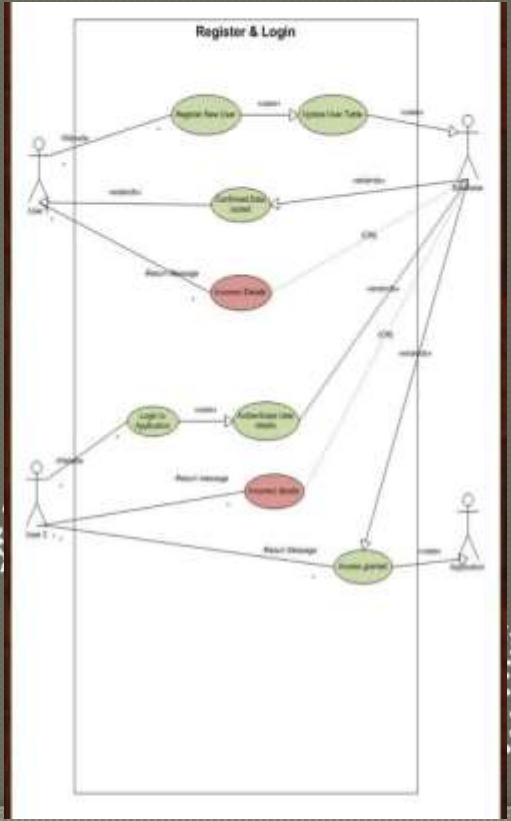
Spring boot architecture

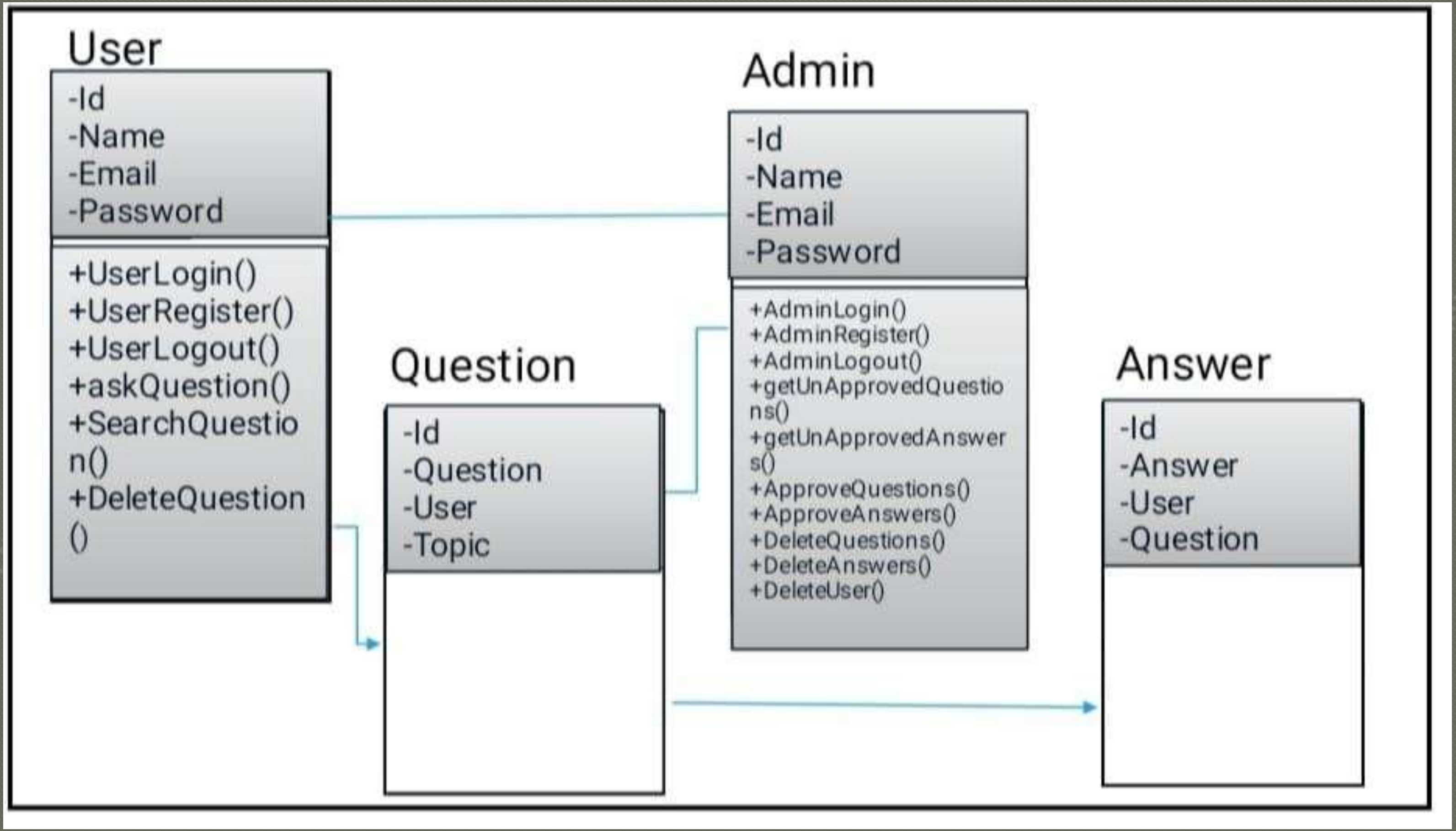


Chat-Microservice Architecture

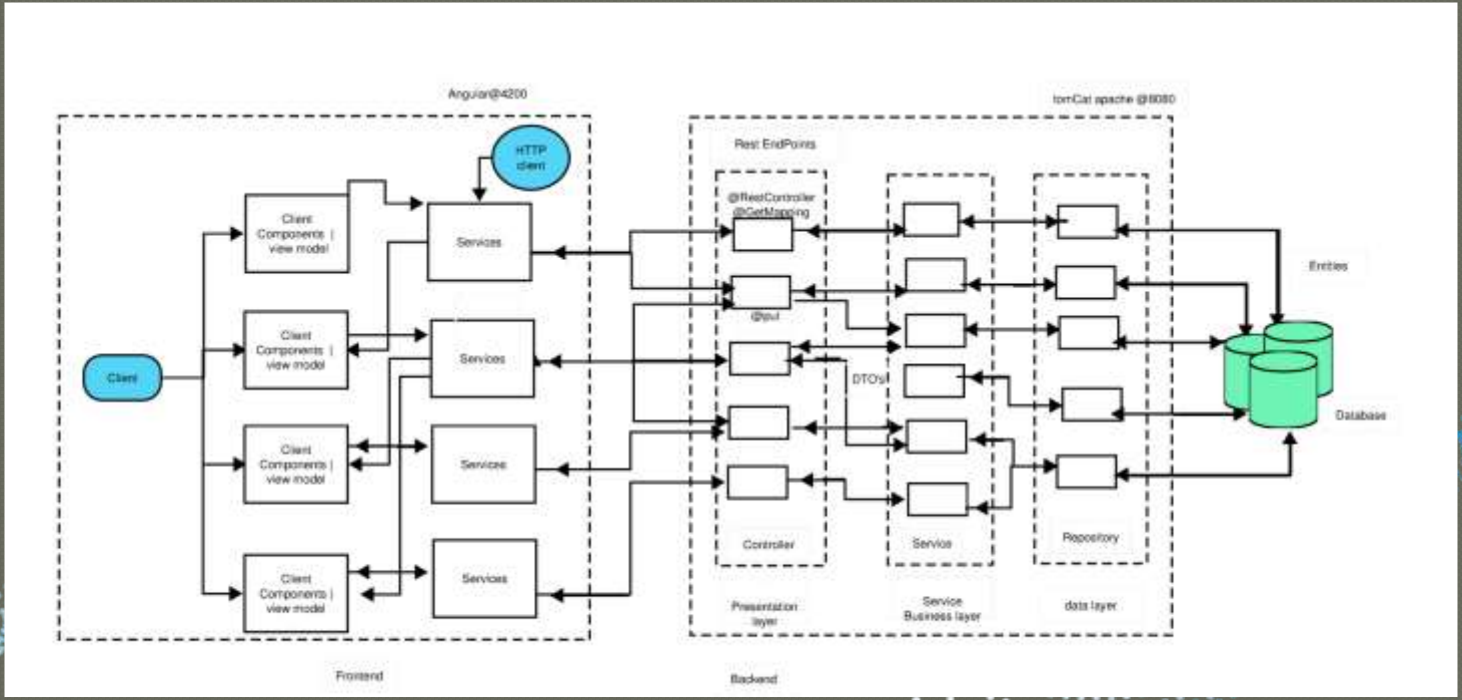


UML diagram


Register & Login
Register New User
Update User Table
Confirmed Data saved
Incorrect Details
Result Message
Login to Application
Authenticate User details
Result message
Incorrect details
Result Message
Access granted
Database
Application


User
-Id
-Name
-Email
-Password
+UserLogin()
+UserRegister()
+UserLogout()
+askQuestion()
+SearchQuestion()
+DeleteQuestion()
Admin
-Id
-Name
-Email
-Password
+AdminLogin()
+AdminRegister()
+AdminLogout()
+getUnApprovedQuestions()
+getUnApprovedAnswers()
+ApproveQuestions()
+ApproveAnswers()
+DeleteQuestions()
+DeleteAnswers()
+DeleteUser()
Question
-Id
-Question
-User
-Topic
Answer
-Id
-Answer
-User
-Question

Total Project Work Flow



Conclusion

- Question and Answer systems are used by a number of people for the basic need of answer retrieval, solving their doubts, assistance during their academics or basic discussions.
- The quality of the answers that are received needed to be improved and the wait time for receiving them was to be reduced, hence, this system was developed.
- The core of this system was to utilise the chattels of a general online network where a question gets forwarded to someone who can provide an answer meanwhile ensuring that quality of the same is good enough in a short time.
- The burden on the users that provide the answers is lessened by directly providing them with questions that they may be intrigued in. This is different when compared with general search engines like Google.