CHAPTER 1

**Chapter 1: Introduction**

**1.1** **Overview**

We need to know how to file a cybercrime complaint in India online & offline and where to report cybercrime in India. Cyber cell is a place where we can register our complaints against any crime or fraud done through internet or technology. The cyber cell departments are accessible in almost every state, ready to deal with cyber complaints in India. Since the numbers of reports for cybercrime complaints online or offline have been increasing heavily in India, therefore, cyber cell departments and the cyber police have identified various types of complaints.

**1.2** **Existing System**

The lots of data on paper/register documents are generated in the organisation about client complaint and its resolutions by gathering information from various sources needed to categorised and sorted out so that many operations can be performed on it and these data can be processed which takes lots of time due to not having required kind of application system.

**1.3** **Problem Statement**

This problem is based on Cyber Cell, which includes registering the new complaints on paper are needed to be handled, arranged, extracted, precise in systematic manner. The progress and action taken on complaints Cyber Cell and accessing information in more precise and accurate information is always required for time utilisation purpose in processing data. In order to actions taken on particular complaints investigations by cyber cell organisation, punishment for cybercrime, the first & foremost step is to lodge complaints against the crime. It is needed to file a written complaint with the cyber cell. In the written complaint, along with needed to provide your name, contact details, and address for mailing and other purposes. The problem of complex functions and large memory space is also identified to resolve it with simple, convenient, less memory required as well as user-friendly application. Along with application system able to persist its data to file and via database solution. Data and its behaviour needs to categorised and properly managed in system so that anything retrieval needed at any point of time can be done easily and quickly in less time.

**1.4 Proposed System**

The project focused on three aspects: ATM relevant bank complaints, Job oriented online complaints, Social media complaints. In this extracting named entities from narrative complaints, then action based on it by cyber cell for each type of complaints. Also awareness creation can be taken from solved complaints that will reduce the number of upcoming complaints. Java is very popular language which is used for design and develops back-end

1

source code in Cyber Cell Modernization Application System. In this application system, extracting named entities from narrative complaints, then action based on it by cyber cell for each type of complaints are handled in designed and implemented in the application using Java language. Convenience and practical approach is mainly considered in the design of the screens interface layouts. All the basic functions for the complaints in application, display list of complaints and all other functions according to research results of the project demand. Performance on the basis of data categorised in application for complaints of cyber cell will be enhanced. The agile methodology used in this project application development. The further section of this will provide more glance of the problem identification, proposed system, solution brief description, need and scope with background details identified.

**1.5** **Need and Scope**

With the development of technology and the advancement of widespread use of the internet, cybercrime has become a real threat to society. With increase in number of cybercrime, no of complaints registered are also increasing. Cyber Cell is organisation where these complaints are registered and an action on it occurs. At Cyber Cell, the requirement of such system was always there. With requirements of Cyber Cell, the application system is proposed.

The Cyber Cell Modernisation System considers these 4 principles-Detect, Deter, Protect, Adapt.

**1.6** **Report Organisation**

* Chapter 1 states the overview of the project with discussing about the discussing about the existing systems in today’s scenario. Describing about the problem statement we are facing about the system. We have given our proposed solution considering all the shortcoming of the previously used system.
* Chapter 2 states the literature survey i.e. the background details of our system including the software engineering paradigm and explaining about the technologies ( Software and Hardware requirement ) which we have used building the system.
* Chapter 3 states about the Analysis of the whole system i.e. identification of system requirement about the feasibility study-Technical Feasibility, Financial Feasibility, operational Feasibility.
* Chapter 4 states about the Design of the whole system including all the UML diagrams all the tools used with ER Diagram and Data Flow Diagram and Data Dictionary.
* Chapter 5 states about the whole code of the system, java code , typescript ,html ,css scripts

2

* and its integration and adaptability.
* Chapter 6 states the Testing phase of our system all the different testing methods and strategies and we have run test cases.
* Chapter states the conclusion of the whole system explaining the advancement of our project in future.
* References: The books, websites, journals, blogs which we have referred.

3

**Chapter 2: Literature Survey**

**2.1 Study**

1. As we all know that in many areas customized application software’s are required to solve organisational day to day activities.
2. Such customization is framed with help of completed domain analysis of the functionalities performed at organisation at regular basis which can be solve with help of application software by reducing manual efforts.
3. Among the difficulties faced by people of organisation by lots of paper work and searching for data, arranging them and solving complaints, some centralised solutions needed in such busy environment.
4. This kind of application software’s is ready to welcome to automate the process to reduce manpower and time consumed for that manual process.
5. The main aim is to satisfy the user of application and also reduce time spent on the manual process which is to complete the cycle of registering complaints, processing it and reaching to solutions.
6. Our ultimate motto is to mitigate the time consumption in processing of complaints and eliminate the paper work of searching/ sorting for information from distributed places, thereby accomplishing both organisation user and complainant citizen demands.

**2.2 Problem Methodology**

The agile methodology for this project application development provides alternative to traditional methods of project management such as the waterfall methodology. The agile way promotes adaptive planning, evolutionary development and continuous improvements. This is iterative and flexible approach can be used in solving complexity in researching for project. According to the research results of project demand,, the basic requirements of project system and its function structure as user login for registering data of complaints of Cyber Cell according to entities identified, then processing it according to its type and details are accumulated. The complete managing application for complaints of Cyber Cell as describe in the research of the system problem through different angles.

**2.3 Software Engineering Paradigm**

The principle of software design mainly includes the following:

4

1. Reliability:
2. The reliability of the software design must be determined. The reliability of the software refers to the ability to avoid fault occurred in the process of system running, as well as the ability to remedy troubles once the fault occurs.
3. Reusability:
4. Commonness of similar codes, and come out new method abstractly and reasonably. Prefer for more generic design.
5. Understand ability:
6. The understand ability of software not only require clear and readable document, but the simplified structure of the software itself, which requires the designer possess keen insight and creativity, and know well about the design objects.
7. Simple program:
8. To keep the program simple and clear, good programmers can use simple program to solve complex problems.
9. Testability:
10. Testability means that the created system has a proper data collection to conduct a comprehensive test of the entire system.

**2.4 Software Development Life Cycle:**

A software development methodology is an outline or frame work to plan and control the process of developing a software application. The methodology for every project is designed to suite the specific needs of each project. The Modified Waterfall Model allows a return to a previous phase for verification or validation, ideally confined to connecting steps. So this project purpose, we made use of modified waterfall model for software development methodology.

The systems development life cycle (SDLC) is a conceptual model used in project management that describes the stages involved in an information system development project, from an initial feasibility study through maintenance of the completed application.

I implemented Waterfall model in this project. The whole process is divided into phases.

Typically, the output of one phase act as an input for the next phase sequentially.

5

**2.4 Technology Methodology**

**2.4.1 Hardware Requirements**

The used hardware for development of application system:

1. RAM: 16 GB
2. Processor: i5 8th generation

**2.4.2 Software Requirements**

The required software for development environment:

* Windows XP onwards(Used Windows 10 Pro)
* Postgresql for database support(Used version 14)
* Java Development Kit(JDK)(Used version 11.0.13.0)
* Eclipse IDE Environment(Used version 2021-09 (4.21.0))
* Spring Tool Suite (STS) plug-in for the Eclipse IDE that is designed to give powerful, integrated environment which build Java Spring Boot Applications.
* Node js (Used version 16.13.0) for Angular (Used version 12.2.11) UI development.
* Visual Studio Code (Used version 1.63.2) for Angular UI development.

**Java Programming Language**

Java is a high level, robust, object-oriented and secure programming language. It is used to develop mobile apps, web apps, desktop apps, games and much more.

**Java Advantages:**

* Java works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.)
* It is secure, fast and powerful
* It is open-source and free

**Spring Boot Framework**

Spring Boot is an open source Java-based framework used to create a micro Service. Spring Boot provides a good platform for Java developers to develop a stand-alone and production-grade spring application that you can just run.

**Features:**

* Create stand-alone Spring applications

6

* Embed Tomcat, Jetty or Undertow directly (no need to deploy WAR files)
* Provide opinionated 'starter' dependencies to simplify your build configuration
* Automatically configure Spring and 3rd party libraries whenever possible
* Provide production-ready features such as metrics, health checks, and externalized configuration
* Absolutely no code generation and no requirement for XML configuration

**Advantages:**

There are few advantages to its developers -

* Easy to understand and develop spring applications
* Increases productivity
* Reduces the development time
* It provides a flexible way to configure Java Beans, XML configurations, and Database Transactions.
* It provides a powerful batch processing and manages REST endpoints.
* In Spring Boot, everything is auto configured; no manual configurations are needed.
* It offers annotation-based spring application
* Eases dependency management
* It includes Embedded Servlet Container

7

**Chapter 3: Analysis**

**3.1 Identification of System Requirements**

System requirements are the requirements at the system level that describe the functions which the system as a whole should fulfil to satisfy the user’s needs and requirements and is expressed in an appropriate combination of textual statements, views and non-functional requirements also ensuring the safety .security, reliability, etc. that will be necessary.

System requirements play major roles such as:

* From the basis of system architecture and design activities.
* From the basis of system integration and verification activities.
* Act as reference for validation and stakeholder acceptance
* Provide a means of communication between the various technical staff that interact throughout the project.

System requirements are considered in detail during System Definition. Neither can be considered complete until consistency between the two has been achieved as demonstrated by traceability, for which a number of iteration may be needed.

**3.2 Functional Requirements**

To implement the above goals, the following requirement needs to be followed:

* Specifying the application and various components of the architecture.
* Specifying the bindings between the tasks and the resources either manually or by the design tools.
* Specifying the port interconnections between the resources.
* Analysis: Extracting the data required for analysis and the doing the analysis.

**3.3 Non-Functional Requirement**

In system and requirement engineering, non-functional requirements a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours. They are contrasted with functional requirement that define specific behaviour or functions. The plan for implementing functional requirement is detailed in the system architecture, because they are usually Architecturally Significant Requirements.

8

Broadly, functional requirements define what a system is supposed to do and non-functional requirements define how a system is supposed to be. Functional requirements are usually in the form of “System shall do< requirement>”, an individual action or part of the system, perhaps explicitly in the sense of a function, a box description input, output, process and control functional model.

**3.4 Feasibility Study**

Feasibility study aims to uncover the strengths and weakness of a proposed venture, opportunities and threats present in the environment and the resources required carrying through. Feasibility study evaluates the project’s success.

**3.4.1 Technical Feasibility**

To design application system which meets the basic requirement, deep understanding of Java language, the Eclipse ide, Postgresql databases, the micro services architecture, application of framework and other technical knowledge are needed (framework is the core of the application and pattern that all programmers uses in developing).Based on the related digital technology information, resources and spirit of willing to learn, the digital technology is feasible.

**3.4.2 Financial Feasibility**

The design of application based on Angular framework requires elaborate designs of screens in application, by adopting typescript and Java language as technical support of this application, with plug-in tools, and combination of Java JDK11.0.13 version lead to comprehensive and smoothly design and development of application. Therefore, the whole process of development doesn’t need to spend any money that is economic feasibility.

**3.4.3 Operational Feasibility**

Strengthened the criteria for primary key specification, primary key be unique across whole database (a so-called enterprise key i.e. a primary key whose value is unique across all relations.), not just unique within the relational table to which it applies. This criterion makes a primary key more like in object oriented databases called an object identifier. In this project, we have used complaint number as so called primary key. The progress and action taken on complaints Cyber Cell can be processed in this system as part of ATM bank, Job Online, Social Media and other structure created in this project database design which will provide feasibility to access information in more precise and accurate framework.

9

**Chapter 4: Project Planning**

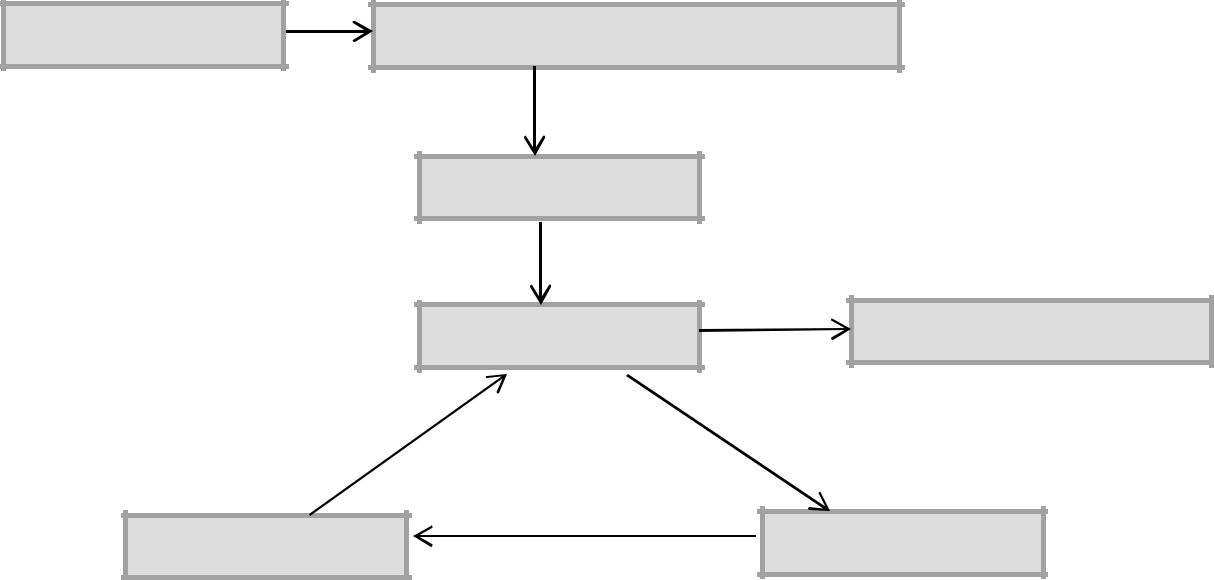
Planning is an ongoing process of refinement, each iteration becoming more detailed and more accurate than the last. Over successive iterations, the emphasis and purpose of planning will shift.

Planning is one of the most important project management and time management techniques. Planning is preparing a sequence of action steps to achieve some specific goal. If you do it effectively, you can reduce much the necessary time and effort of achieving the goal. Some of the important activities that mark this phase are - development of schedule, GANTT charts, estimating and reserving resources, planning dates and modes of communication with stakeholders based on milestones, deadlines and important deliveries.

During the feasibility and project start-up, the main purpose of planning will be estimate timescales and the risks of not achieving target completion dates. As the project proceeds beyond the completion dates. As the project beyond the feasibility ensuring availability and flow control.

Throughout the project, until the final has reached monitoring and preplanning must continue to correct any drift that might prevent, meeting time or targets.

Planning of Cyber Cell Modernization Application System programming planning is elaborated with diagram shown below:



Problem Analysis

Proposed Solution & Design Program

Code Program

Test Program

Document & Release

Correct errors

Diagnose errors

Figure: Project Planning

10

**Chapter 5: Design**

**5.1 Introduction to UML**

Unified modelling language (UML) is a general purpose modelling language. The main aim of UML is define a standard way to visualize the way a system has been designed. It is quite similar to blueprints used in other fields of engineering.

UML is not a programming language; it is rather a visual language. We use UML diagrams to portray the behaviour and structure of system. UML helps software engineers, businessmen and system architects with modelling, design and analysis.

**5.2 UML Diagrams**

Complex applications need collaboration and planning from multiple teams and hence require a clear and concise way to communicate amongst them.

Businessmen do not understand code, so UML becomes essential to communicate with non-programmers essential requirements, functionalities and processes of the system.

A lot of time is saved down the line when teams are able to visualize processes user interactions and static structure of the system.

UML is linked with object oriented design and analysis. UML makes the use of elements and from associations between them to form diagram. Diagram in UML can be broadly classified as:

1. Structural Diagrams: Capture static aspects or structure of a system. Structural

Diagrams include: Component Diagrams, Objects Diagram, Class Diagrams and Deployment Diagrams.

1. Behaviour Diagram: Capture dynamic aspects or behaviour of the system. Behaviours

diagram include: Use Case Diagram , State Diagram, Activity Diagram and Interaction Diagram.

**5.2.1 Use Case Diagram**

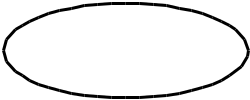
The purpose of a use case diagram in UML is to demonstrate the different ways that a user might interact with a system. Use case diagrams consist of 3 objects.

Actor: Actor in a use case diagram is any entity that performs a role in one given system. This could be a person, organization or an external system and usually drawn like skeleton.



11

Use Case: A use case represents a function or an action within the system. It’s drawn as an oval and named with the function.



System: The system is used to define the scope of the use case and drawn as a rectangle. This an optional element but useful when you’re visualizing large systems.

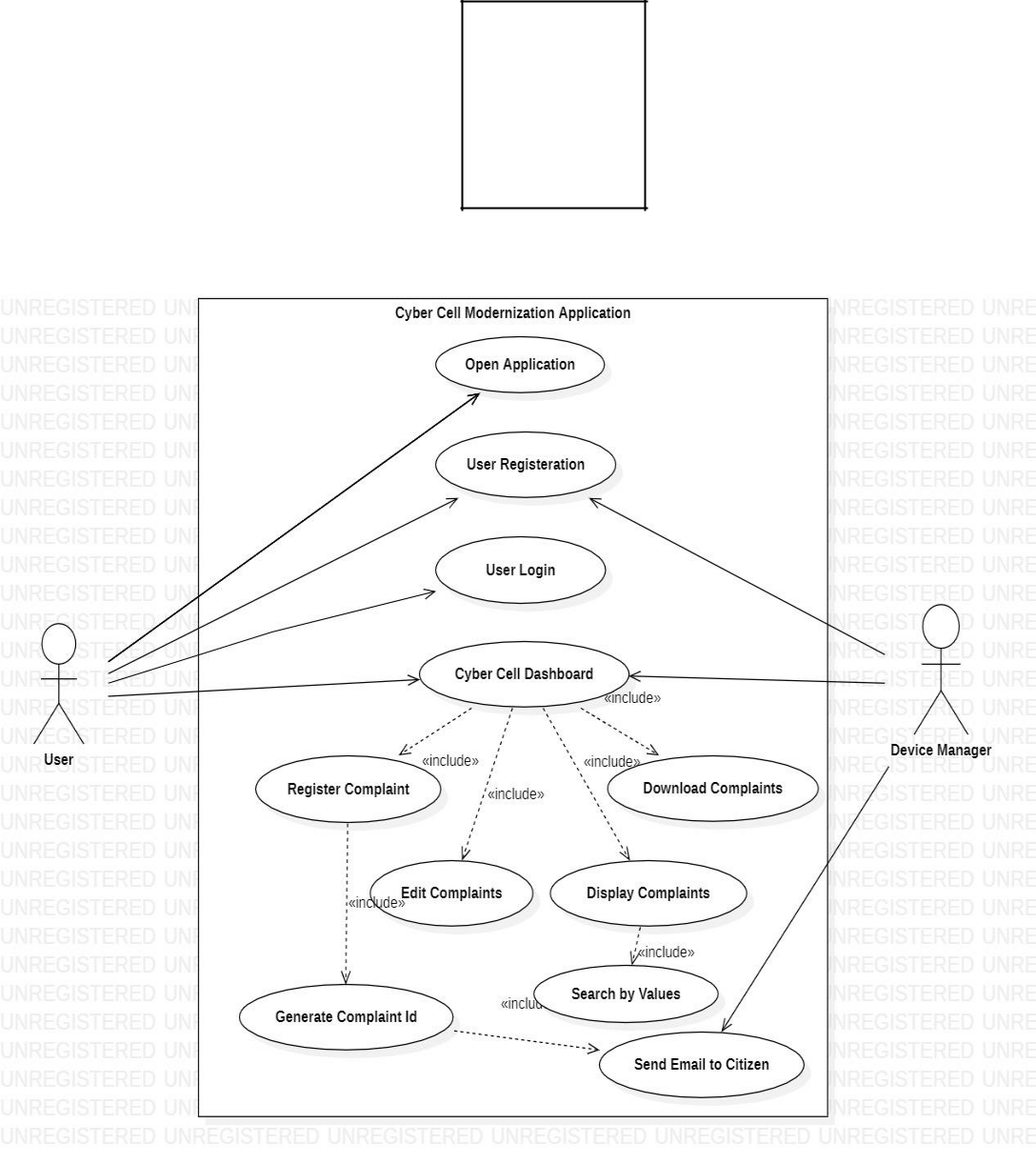


Figure: Use Case Diagram

12

**5.2.2 Class Diagram**

The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application. A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's:

* classes,
* their attributes,
* operations (or methods),
* And the relationships among objects.

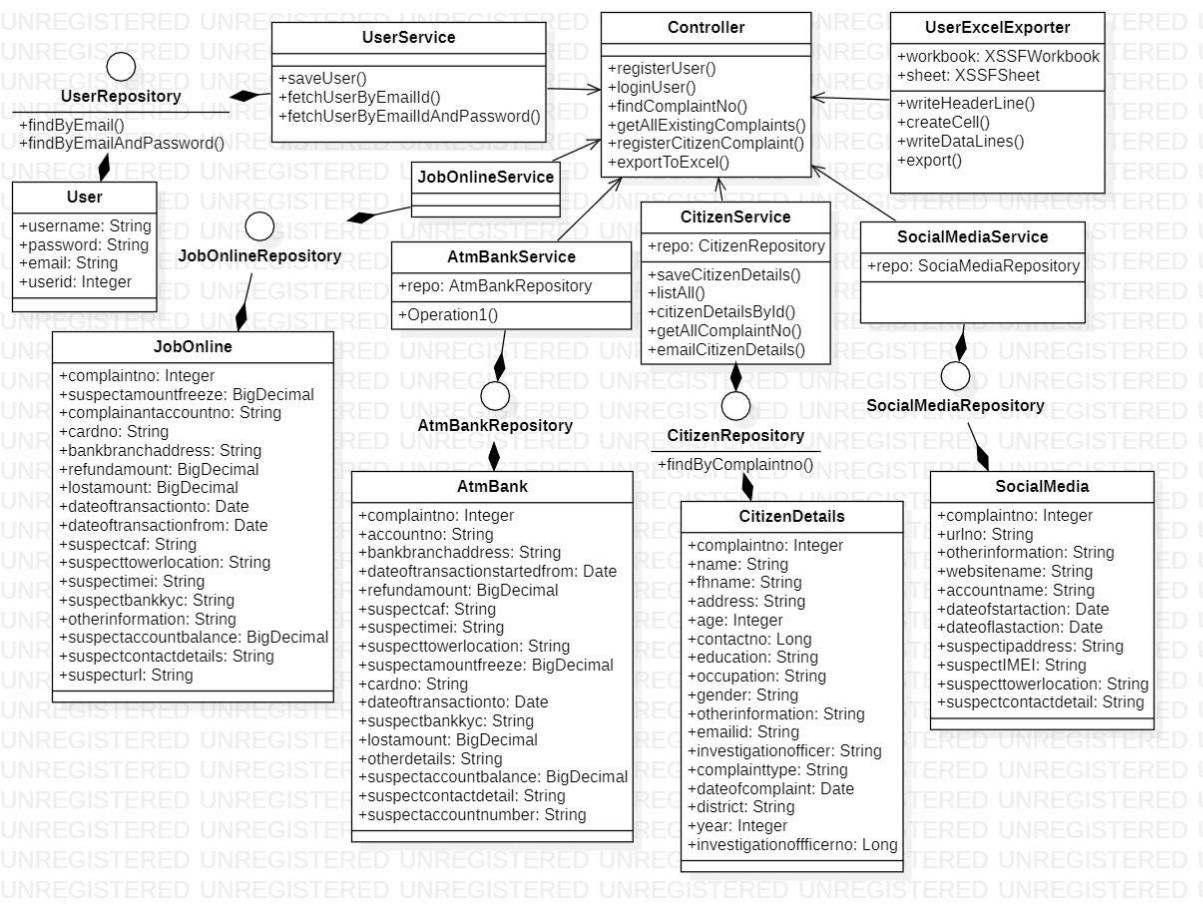


Figure: Class Diagram

13

**1.2.3** **Sequence Diagram**

A sequence diagram is a type of interaction diagram because it describe how-and in what order- a group of objects works together.

Sequence diagram are sometimes known as an event diagrams or event scenarios.Sequence diagram are time focus

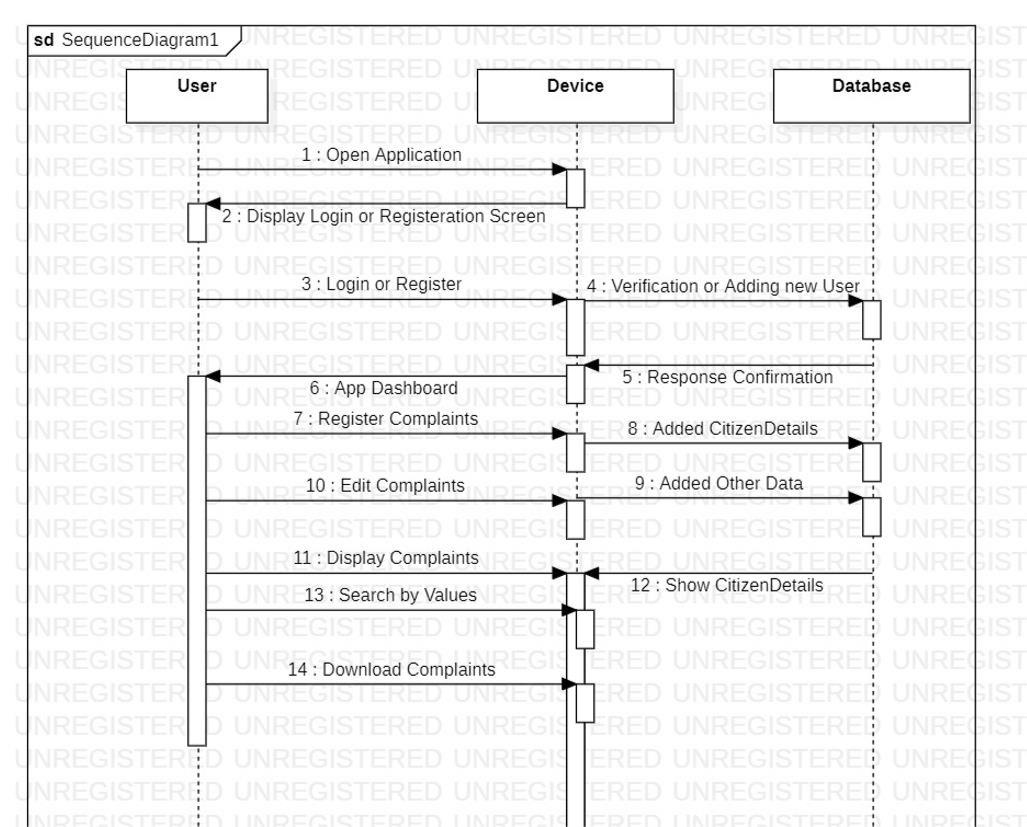


Figure: Sequence Diagram

14

**1.2.4 ER Diagram**

ER Diagram is a visual representation of data that describes how data is related to each other. In ER Model, we disintegrate data into entities, attributes and setup relationships between entities, all this can be represented visually using the ER diagram. ER Diagrams contain different symbols that use rectangles to represent entities, ovals to define attributes and diamond shapes to represent relationships.

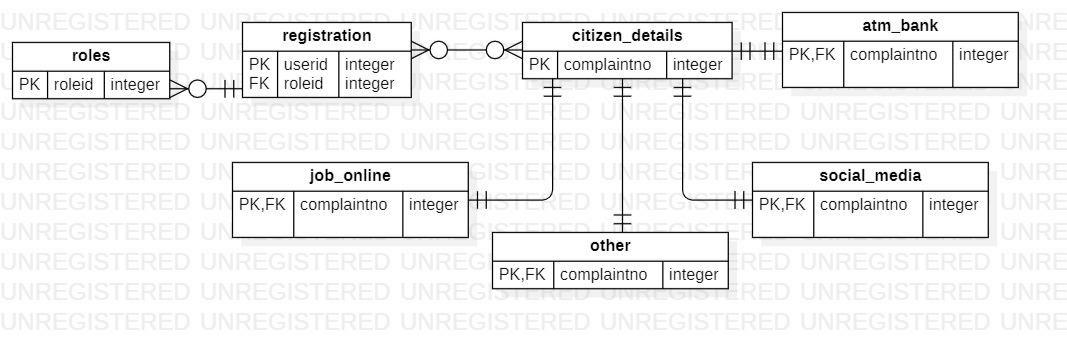


Figure: ER Diagram

15

**1.2.5 Activity Diagram**

An activity diagram portrays the control flow from a start point to f i n i s h point showing various decision paths that exist while the activity is being executed.

An activity diagram focuses on condition of flow and the sequence in which it happens. An activity diagram is a behavioral diagram i.e. it depicts the behavioral of a system.

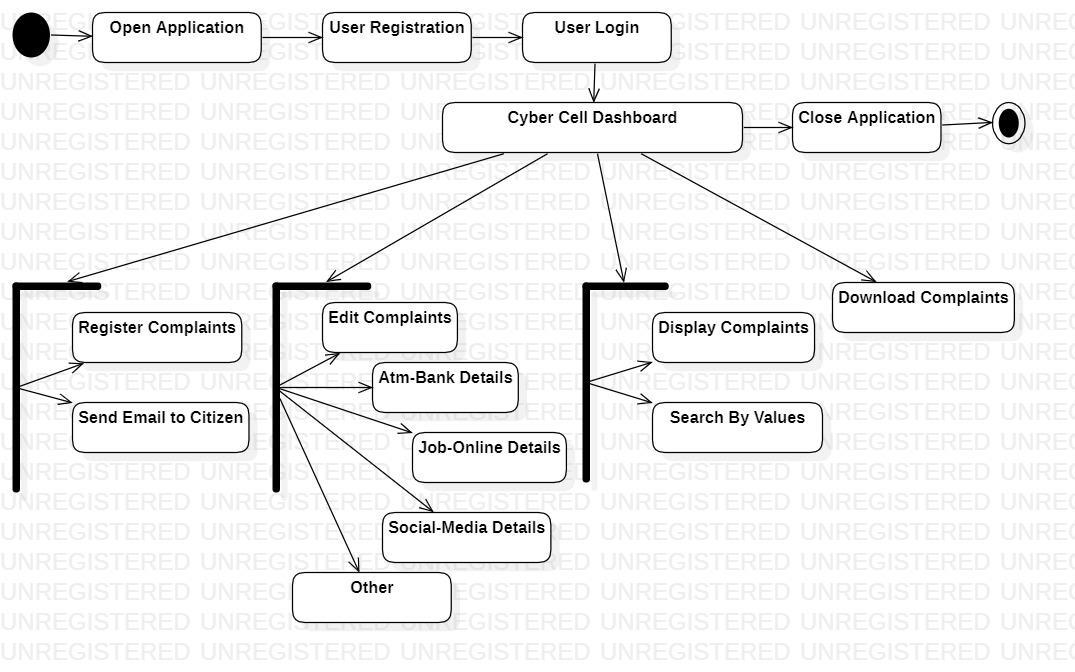
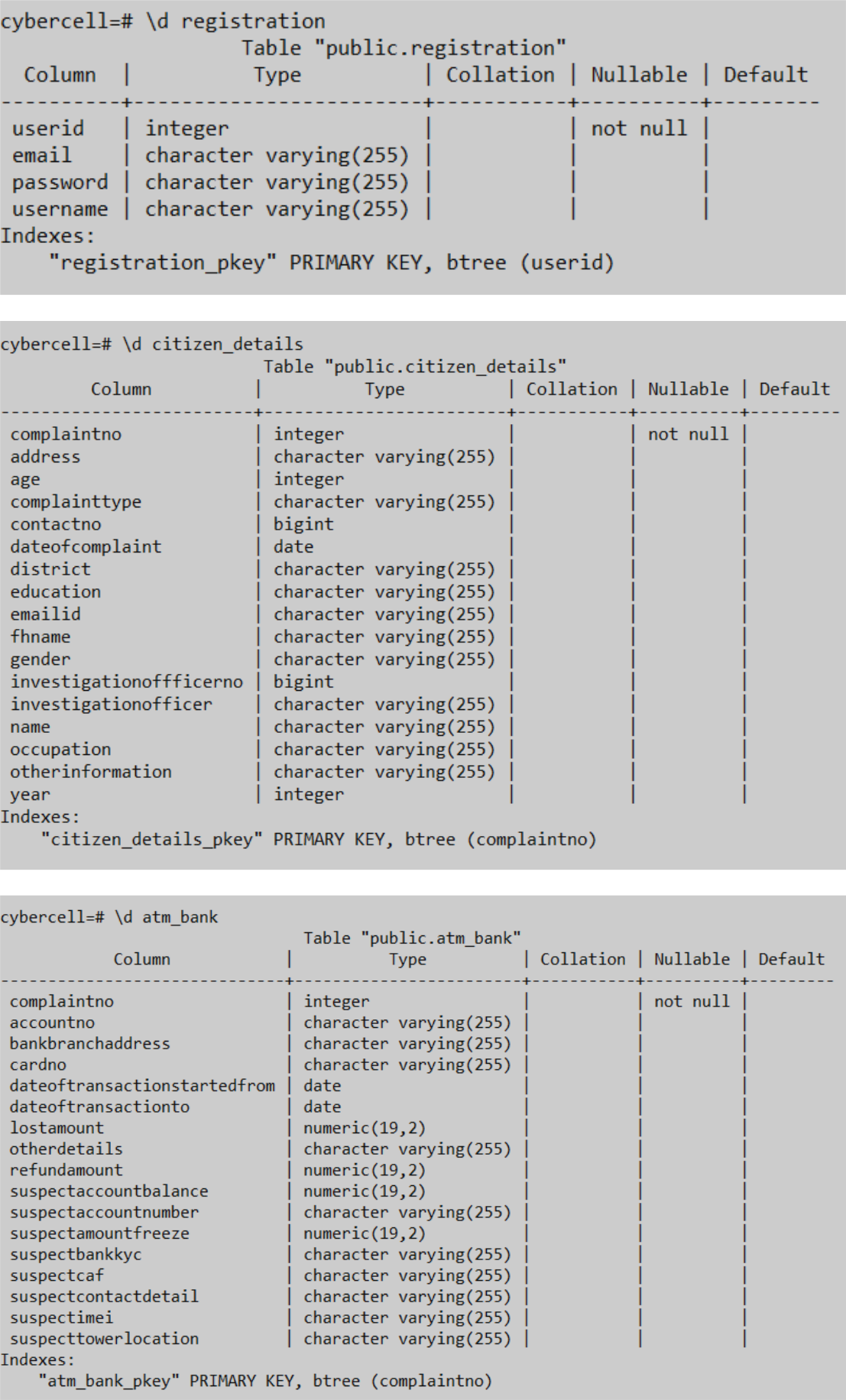


Figure: Activity Diagram

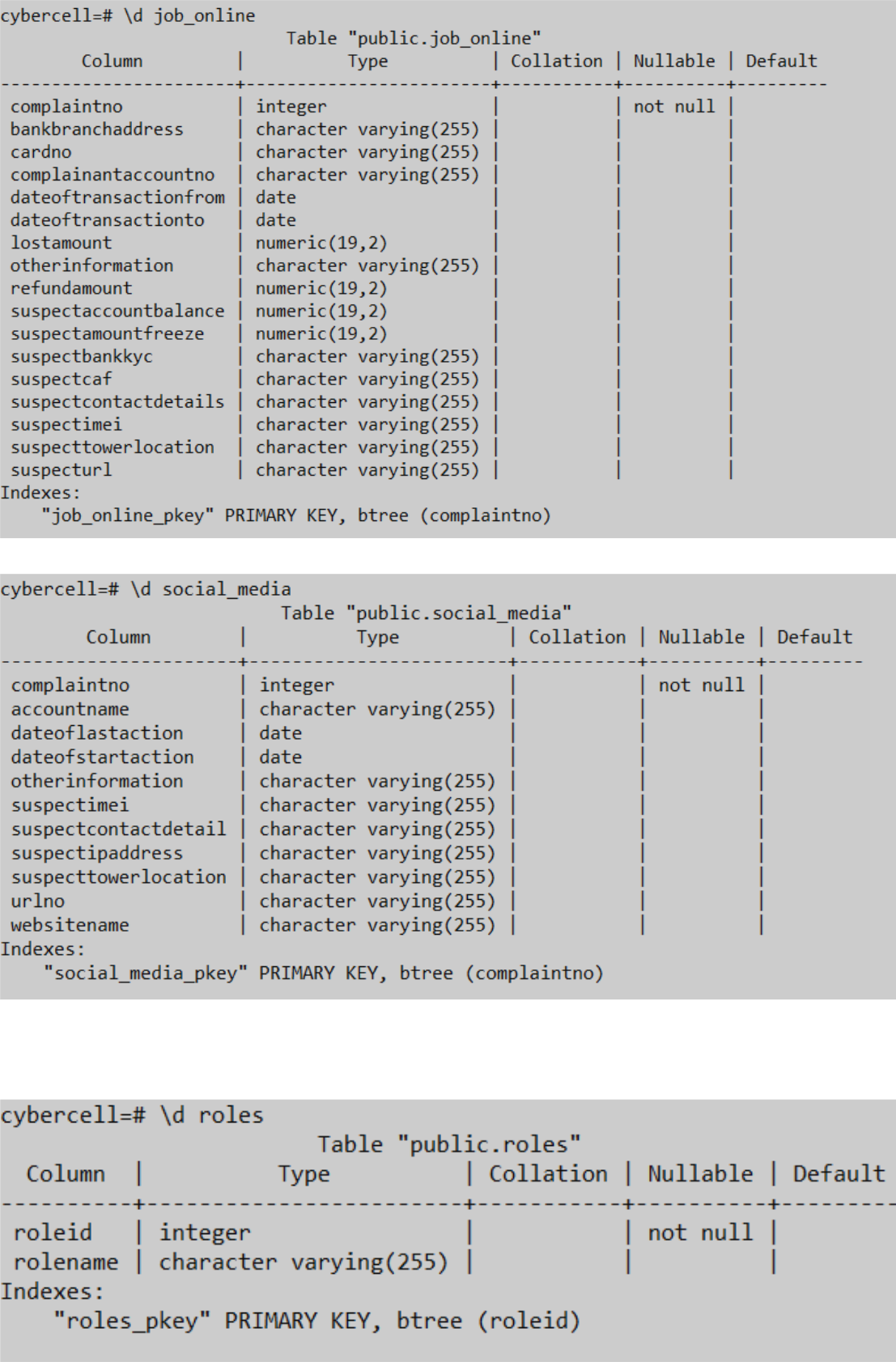
**1.2.6 Table Structure**

The database schema present in Postgresql is displayed below which are of registration table, citizen details table, atm bank table, job online table, social media table, other details table.

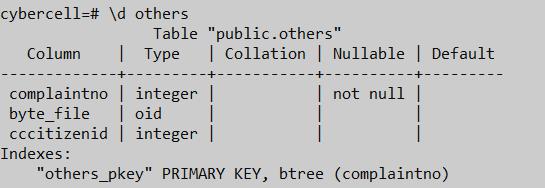
16



17



18

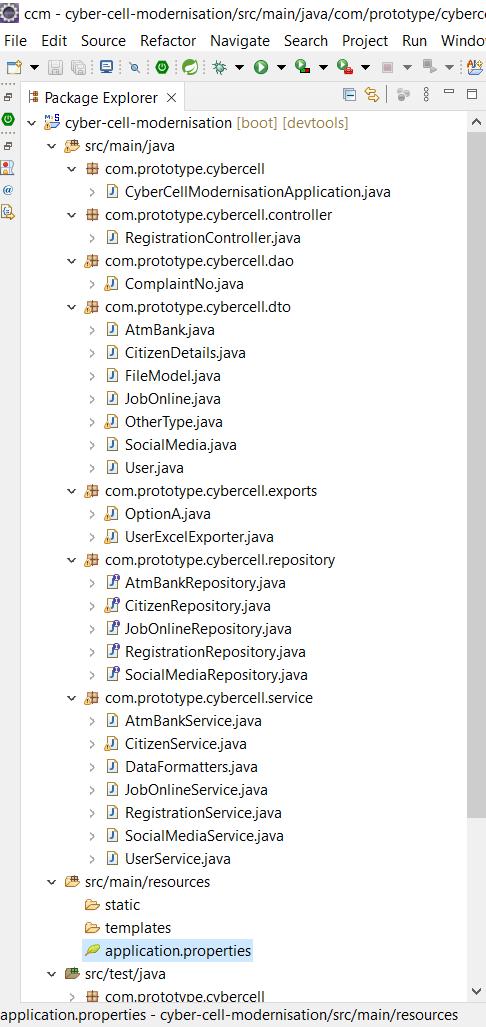


**Chapter 6: Implementation**

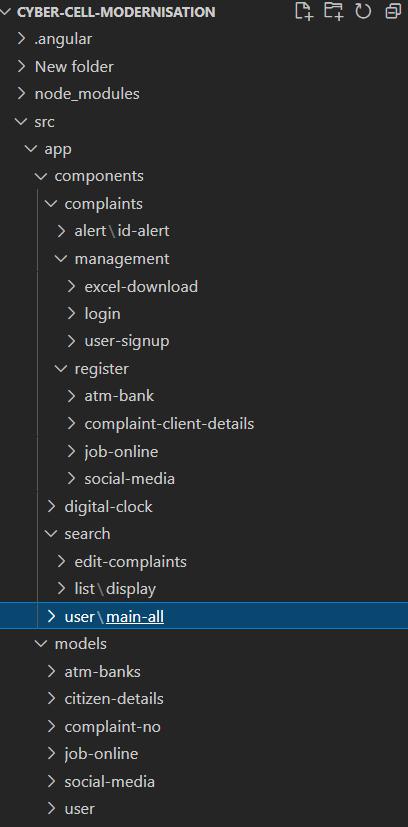
**6.1 Coding(Main Module)**

The main module consist of basically two aspects backend and frontend. In the backend the controller is the end points of the rest application programming interface of various services in the java spring boot framework. The model, view and controller framework is used in development of Angular as defining the view.

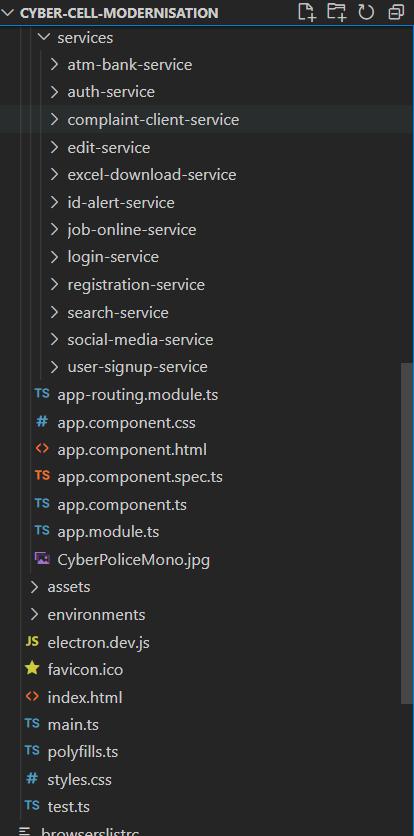
19



20



21

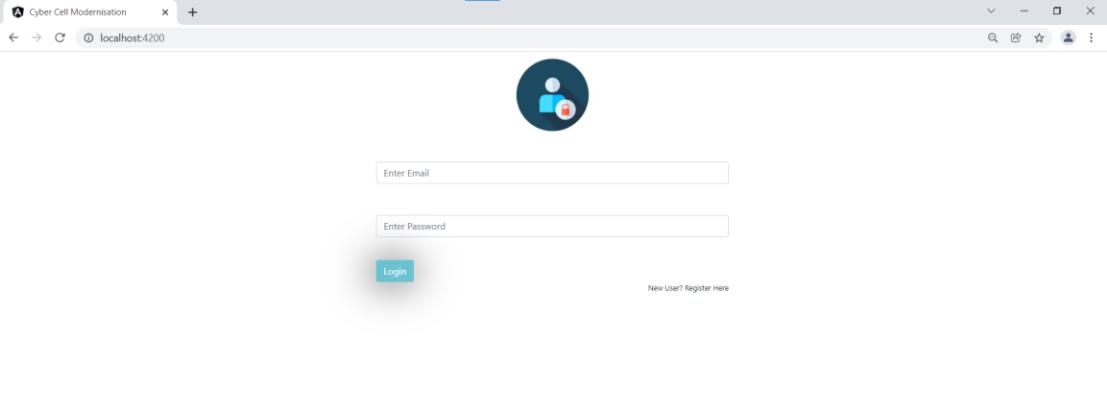


22

**6.2 Results: Screen Shots**

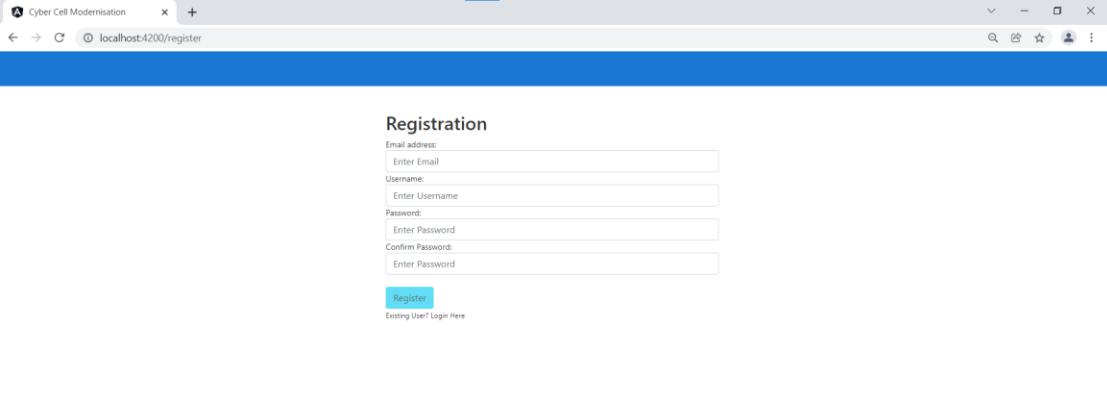
6.3.1 Login Screen

The first screen when user interacting with the application. The input parameters using is email id and password.



1.3.2 Registration Screen

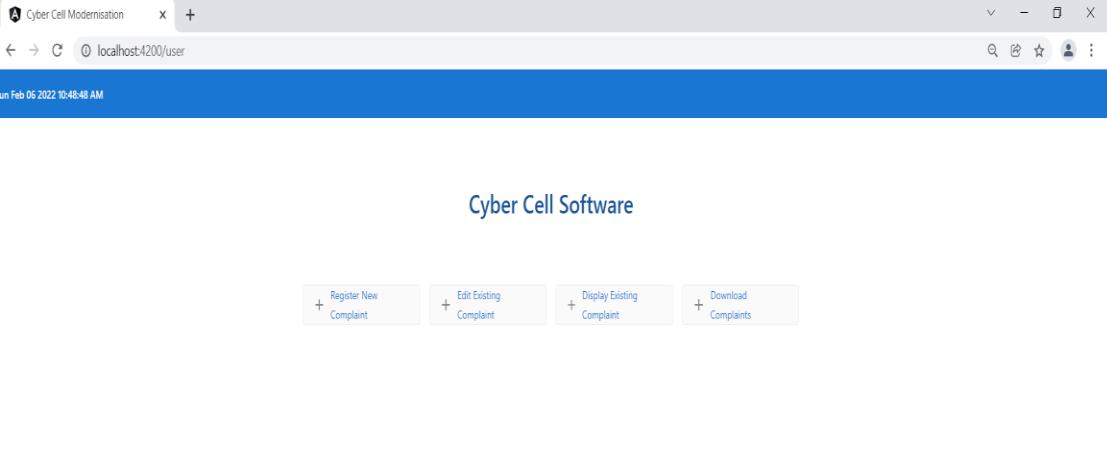
If the login screen is not verified then it is redirected to registration screen which registers the user of application with email address, username, password as parameters.



1.3.3 Dashboard Screen

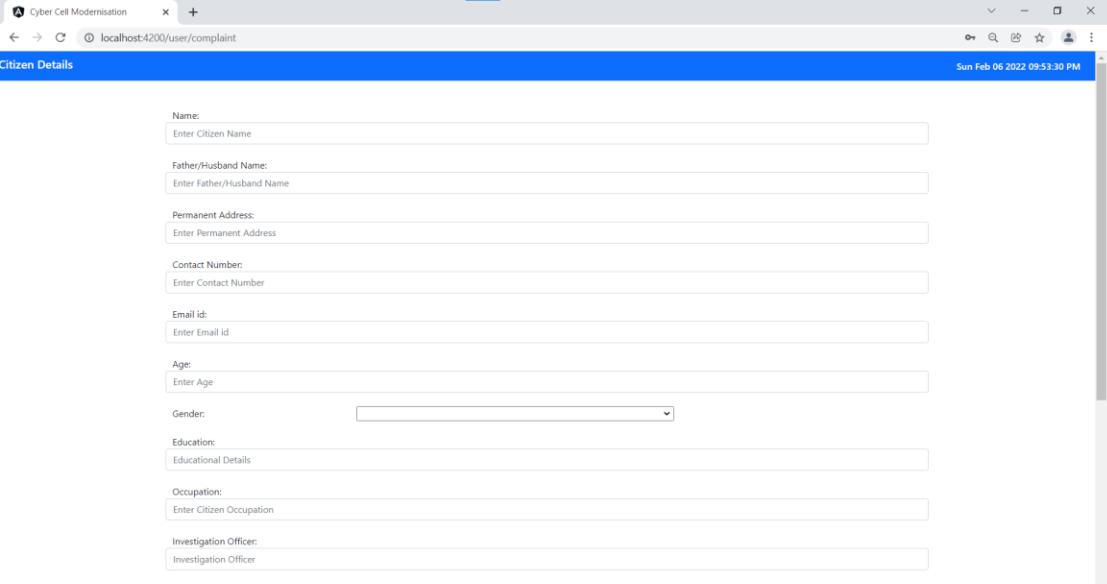
While verification successful for user will displays the dashboard of the cyber cell application, with the functionalities provided by the application.

23

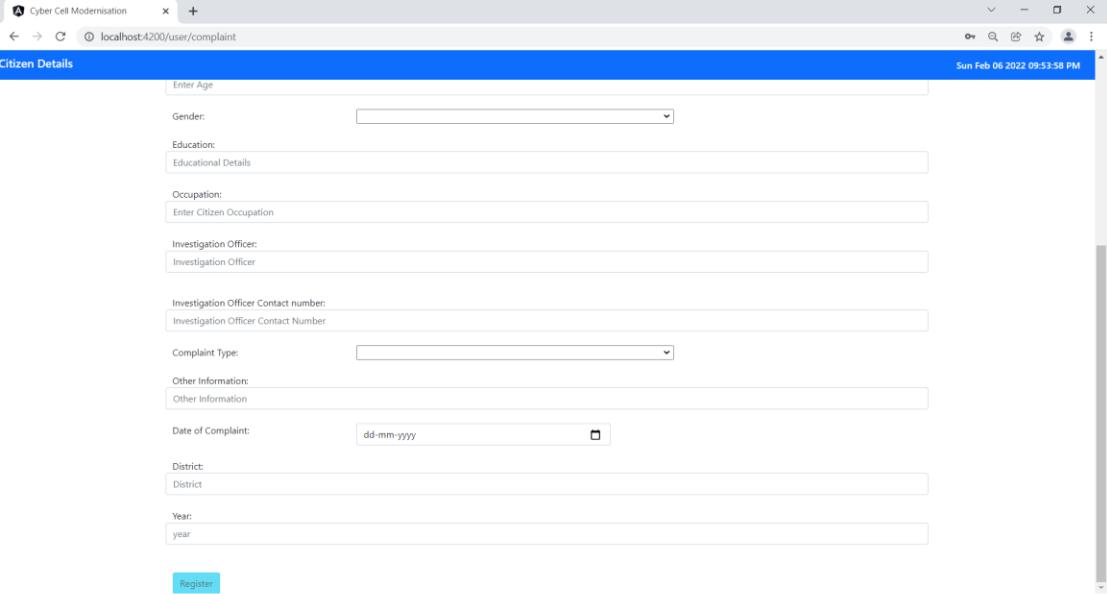


1.3.4 Register Citizen Details Screen

This is the first screen of the dashboard functionality for registering the citizen’s complaint details for the first time by the user.

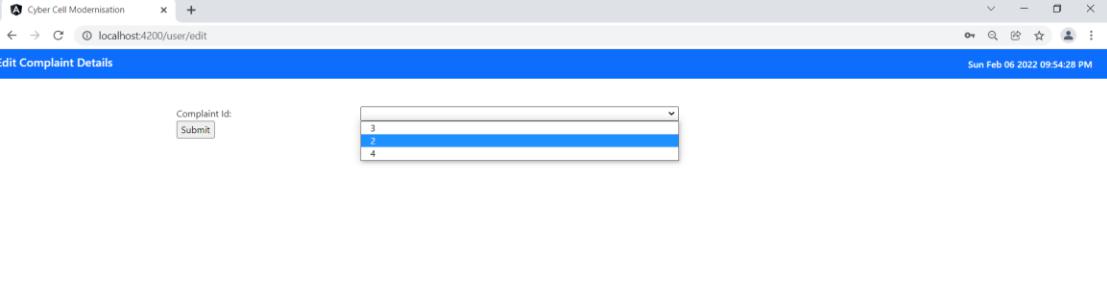


24



1.3.5 Edit Details Screen

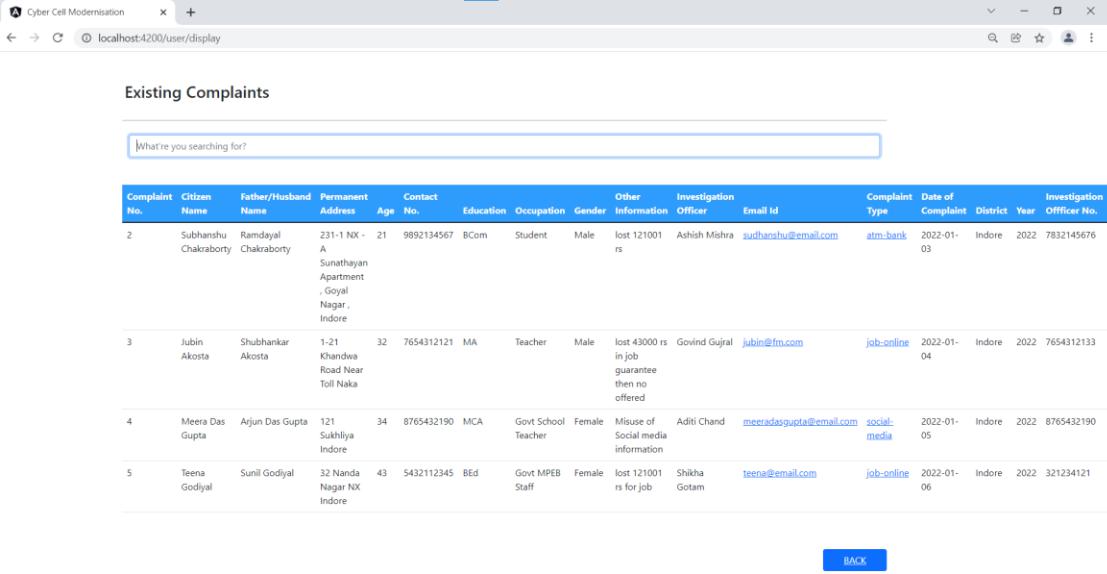
This includes the multiple complaints registered in the application with all complaint id for adding additional details in the application.



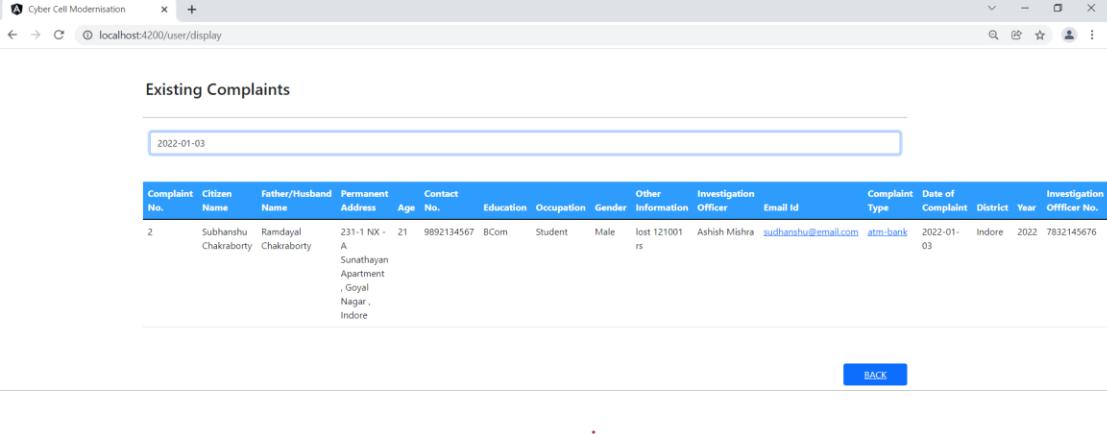
1.3.6 Display Screen

All the registered complaints can be seen from the display functionality of this application.

25



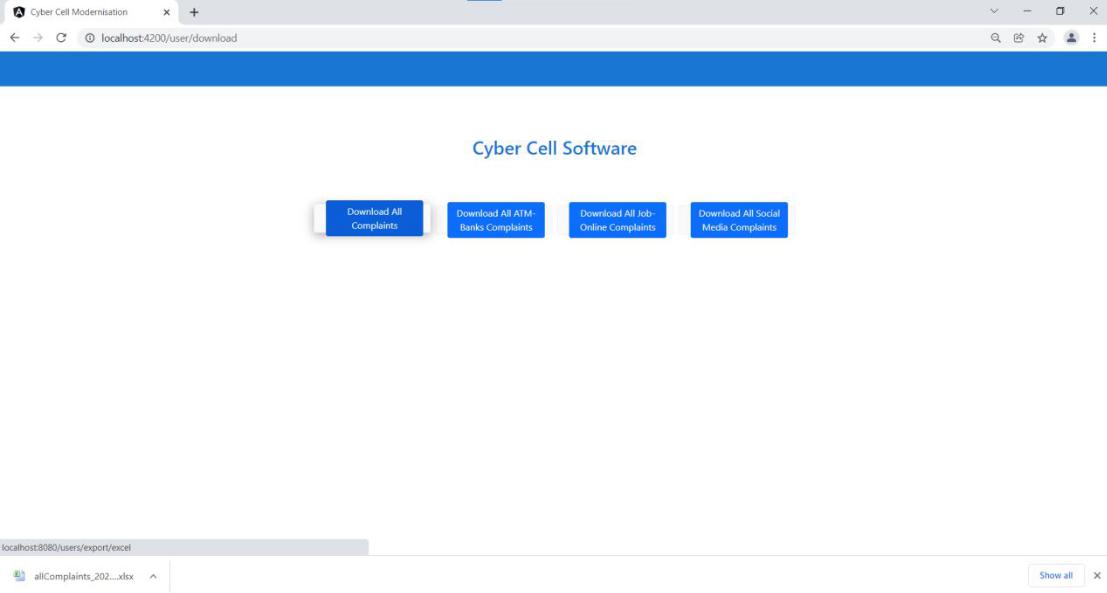
1.3.7 Search Working Screen



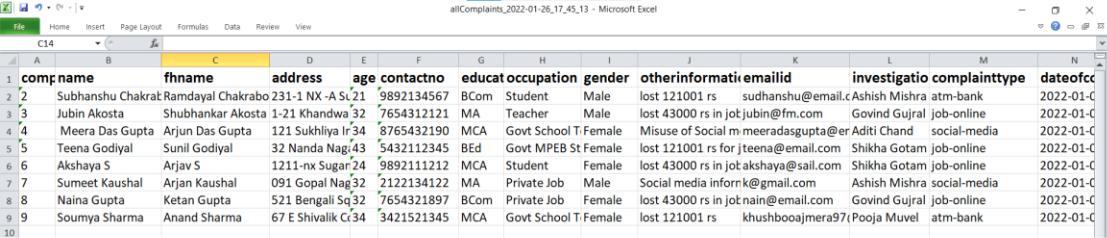
1.3.8 Download Complaints Screen

To download the citizen details in the excel format this screen provides the functionality for the same.

26



1.3.9 Downloaded File Screen



27

**Chapter 7: Testing**

**7.1 Testing Objectives**

This checklist is specifically designed to test the characteristics of a application. Obviously, it test only generic app characteristics.

The checklist is split into five different fields:

* Environment: These are the things to check the application is compatible with different devices.
* Device specific characteristics: These are characteristics that are related to the device on which the app is used.
* Application checks: These are things to check that have to do with functionality that is frequently used in an application.
* Application User Interface Checks
* Store Specific Check

System testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. Software testing fundamentals define the overriding objectives for software testing. Testing is one of the steps in the software engineering process.

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding a yet undiscovered error. A successful test is one that uncovers an as yet undiscovered error.

**Test case for Login Page**

**Functional Test Cases**

1. Verify if a user will be able to login with valid email id and valid password.
2. Verify if a user will be able to login with valid email id and invalid password.
3. Verify the login page for both, when the field is blank and login button is clicked.

**Non Functional Test Cases**

Verify the login page by pressing “Back Button” of browser. System should not allowed to login until you log out.

**Test case for Signup Page**

1. Verify that the required/mandatory fields are marked with \* against the field.

28

1. Verify that clicking create an account button after entering all the required fields, submits the data to the server.
2. Check that not filling the mandatory fields and clicking the submit button will lead to a validation error.

29

**Chapter 8: Conclusion**

Based on current technical feasibility and researched information this project includes all functionality in Cyber Cell Modernization Application System. The aim is to automate its existing manual system by the help of computerized equipment’s and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same.

**8.1 Conclusion**

This project is based on research on working pattern of Cyber Cell, which includes registering user of system, then registered user can register the new complaints which are previously registered on paper are now can be handled from this software system. The progress and action taken on complaints Cyber Cell can be processed in this system as part of ATM bank, Job Online, Social Media and other structure created in this database designed.

**8.2 Future Work**

AI features can be added for generating suggestions for new registered complaints from processed older data of complaints.

In this system, with feedback from Cyber Cell organisation can definitely improve and enhance the user experience in future versions.

30

**References**

**Web Sites**

1. https://docs.oracle.com/en/java/javase/11/books.html
2. https://spring.io/learn
3. https://angular.io/start
4. https://www.w3schools.com/java/default.asp

31

**Appendix**

**Controller.java**

package com.prototype.cybercell.controller;

import java.io.IOException;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.HashMap;

import java.util.List;

import javax.servlet.http.HttpServletResponse;

import org.springframework.beans.factory.annotation.Autowired; import org.springframework.web.bind.annotation.CrossOrigin; import org.springframework.web.bind.annotation.GetMapping; import org.springframework.web.bind.annotation.PathVariable; import org.springframework.web.bind.annotation.PostMapping; import org.springframework.web.bind.annotation.RequestBody; import org.springframework.web.bind.annotation.RestController;

import com.prototype.cybercell.dto.AtmBank; import com.prototype.cybercell.dto.CitizenDetails; import com.prototype.cybercell.dto.JobOnline; import com.prototype.cybercell.dto.SocialMedia; import com.prototype.cybercell.dto.User;

32

import com.prototype.cybercell.exports.UserExcelExporter; import com.prototype.cybercell.repository.CitizenRepository; import com.prototype.cybercell.service.AtmBankService; import com.prototype.cybercell.service.ChartService; import com.prototype.cybercell.service.CitizenService; import com.prototype.cybercell.service.JobOnlineService; import com.prototype.cybercell.service.SocialMediaService; import com.prototype.cybercell.service.UserService;

@RestController

public class Controller {

@Autowired

private UserService uservice = new UserService();

@Autowired

private CitizenService cservice = new CitizenService();

@Autowired

private AtmBankService abservice = new AtmBankService();

@Autowired

private JobOnlineService joservice = new JobOnlineService();

@Autowired

private SocialMediaService smservice = new SocialMediaService();

33

@Autowired

private ChartService chservice = new ChartService(); @Autowired

private CitizenRepository crepo;

@PostMapping("register-user")

@CrossOrigin(origins = "http://localhost:4200")

public User registerUser(@RequestBody User user) throws Exception { String tempEmail = user.getEmail();

if (tempEmail != null && !"".equals(tempEmail)) {

User u = uservice.fetchUserByEmailId(tempEmail);

if (u != null) {

throw new Exception("User with " + tempEmail + "

already exist");

}

}

return uservice.saveUser(user);

}

@PostMapping("register-client")

@CrossOrigin(origins = "http://localhost:4200")

public CitizenDetails registerCitizenComplaint(@RequestBody CitizenDetails cd) throws Exception {

return cservice.saveCitizenDetails(cd);

}

34

@GetMapping("edit")

@CrossOrigin(origins = "http://localhost:4200")

public List<Integer> findComplaintNo() throws Exception {

System.out.println("Inside list method in controller"); return cservice.getAllComplaintNo();

}

@GetMapping("/{complaintId}")

public List<CitizenDetails> citizenDetailbyId(@PathVariable("complaintId") Integer complaintId) {

return cservice.citizenDetailsbyId(complaintId);

}

@GetMapping("display")

@CrossOrigin(origins = "http://localhost:4200")

public List<CitizenDetails> getAllExistingComplaints() { return (List<CitizenDetails>) crepo.findAll();

}

@PostMapping("register-atmbank")

@CrossOrigin(origins = "http://localhost:4200")

public AtmBank registerAtmBank(@RequestBody AtmBank cd) throws Exception {

return abservice.saveCitizenDetails(cd);

}

35

@PostMapping("register-jobonline")

@CrossOrigin(origins = "http://localhost:4200")

public JobOnline registerJobOnline(@RequestBody JobOnline cd) throws Exception {

return joservice.saveCitizenDetails(cd);

}

@PostMapping("register-socialmedia")

@CrossOrigin(origins = "http://localhost:4200")

public SocialMedia registerSocialMedia(@RequestBody SocialMedia cd) throws Exception {

return smservice.saveSocialMedia(cd);

}

@PostMapping("login")

@CrossOrigin(origins = "http://localhost:4200")

public User loginUser(@RequestBody User user) throws Exception { String tempemail = user.getEmail();

String temppwd = user.getPassword();

User obj = null;

if (tempemail != null && temppwd != null) {

obj = uservice.fetchUserByEmailIdAndPassword(tempemail,

temppwd);

}

if (obj == null) {

throw new Exception("Bad Credentials");

}

return obj;

36

}

@GetMapping("/users/export/excel")

public void exportToExcel(HttpServletResponse response) throws IOException {

response.setContentType("application/octet-stream");

DateFormat dateFormatter = new SimpleDateFormat("yyyy-MM-dd\_HH:mm:ss");

String currentDateTime = dateFormatter.format(new Date());

String headerKey = "Content-Disposition";

String headerValue = "attachment; filename=allComplaints\_" + currentDateTime + ".xlsx";

response.setHeader(headerKey, headerValue);

List<CitizenDetails> listUsers = cservice.listAll();

UserExcelExporter excelExporter = new UserExcelExporter(listUsers);

excelExporter.export(response);

}

@GetMapping("/all-index")

public Integer getAllComplaintTypeCount() throws Exception { System.out.println("inside count controller");

return chservice.getAllComplaintTypeCount();

}

37

@GetMapping("/type-index")

public HashMap<String, Integer> getAllComplaintWiseCount() throws Exception {

System.out.println("inside count controller");

return chservice.getAllComplaintWiseCount();

}

}

* **CitizenService.java**

package com.prototype.cybercell.service;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.List;

import java.util.Properties;

import javax.mail.Message;

import javax.mail.MessagingException;

import javax.mail.Session;

import javax.mail.Transport;

import javax.mail.internet.InternetAddress;

import javax.mail.internet.MimeMessage;

38

import org.springframework.beans.factory.annotation.Autowired; //import org.springframework.data.domain.Sort; import org.springframework.stereotype.Service;

import com.prototype.cybercell.dto.CitizenDetails;

import com.prototype.cybercell.repository.CitizenRepository;

@Service

public class CitizenService {

@Autowired

private CitizenRepository repo;

CitizenDetails es = null;

String regexMail = "^[a-zA-Z0-9\_!#$%&'\*+/=?`{|}~^.-]+@[a-zA-Z0-9.-]+$";

String regexNumber = "^(?:(?:\\+|0{0,2})91(\\s\*[\\ - ]\\s\*)?|[0]?)?[789]\\d{9}|(\\d[ -]?){10}\\d$";

String regexId = "^\\d[0-9]{5,10}(?:\\d{2})?$"; String regexName = "[a-z]{1,10}";

public CitizenDetails saveCitizenDetails(CitizenDetails citizenDetails) {

return repo.save(citizenDetails);

}

39

public List<Integer> getAllComplaintNo() throws Exception { String JDBC\_DRIVER = "com.postgresql.Driver";

String DB\_URL = "jdbc:postgresql://localhost:5432/cybercell";

String USER = "postgres";

String PASS = "root";

System.out.print("USERPASS");

Connection conn = null;

Statement stmt = null;

List<Integer> cn = new ArrayList<Integer>(); System.out.println("Connecting to a selected database..."); try {

* Class.forName("org.postgresql.Driver"); Class.forName(JDBC\_DRIVER); System.out.println("Connecting to a selected database..."); conn = DriverManager.getConnection(DB\_URL, USER,

PASS);

System.out.println("Connected database successfully...");

System.out.println("Creating statement...");

stmt = conn.createStatement();

String sql = "SELECT complaintno FROM citizen\_details";

ResultSet rs = stmt.executeQuery(sql);

while (rs.next()) {

cn.add(rs.getInt("complaintno"));

40

System.out.println(cn.toString());

}

rs.close();

} catch (SQLException se) { se.printStackTrace();

}

return cn;

}

public void emailCitizenDetails(CitizenDetails citizenDetails) { String to = "kajmera.scsits@gmail.com";// change accordingly

String from = "ccmregisteredmail@gmail.com";// change accordingly/ String host = "localhost";// or IP address

// Get the session object

Properties properties = System.getProperties(); properties.setProperty("mail.smtp.host", host); properties.setProperty("mail.smtp.port", "8080"); Session session = Session.getDefaultInstance(properties); Integer complaintid = null;

* compose the message try {

MimeMessage message = new MimeMessage(session);

message.setFrom(new InternetAddress(from));

message.addRecipient(Message.RecipientType.TO, new

InternetAddress(to));

41

message.setSubject("Registered complaint");

message.setText("Dear Citizen, Your complaint id is " +

complaintid);

// Send message Transport.send(message); System.out.println("message sent successfully....");

} catch (MessagingException mex) { mex.printStackTrace();

}

// return "Citizen Details sent on registered email id";

}

public List<CitizenDetails> citizenDetailsbyId(Integer complaintId) {

return repo.findByComplaintno(complaintId);

}

public List<CitizenDetails> listAll() {

return repo.findAll();

}

}

42

**Angular**

**Main-all.component.html**

<div><h1>Cyber Cell Software</h1></div>

<p></p>

<style>

:host {

font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto, Helvetica, Arial, sans-serif, "Apple Color Emoji", "Segoe UI Emoji", "Segoe UI Symbol";

font-size: 14px;

color: #333;

box-sizing: border-box;

-webkit-font-smoothing: antialiased;

-moz-osx-font-smoothing: grayscale;

}

h1,

h2,

h3,

h4,

h5,

h6 {

margin: 8px 0;

}

p {

margin: 0;

}

43

.spacer {

flex: 1;

}

.toolbar {

position: absolute;

top: 0;

left: 0;

right: 0;

height: 60px;

display: flex;

align-items: center;

background-color: #1976d2;

color: white;

font-weight: 600;

}

.toolbar img {

margin: 0 16px;

}

.toolbar #twitter-logo {

height: 40px;

margin: 0 8px;

}

.toolbar #youtube-logo {

height: 40px;

44

margin: 0 16px;

}

.toolbar #twitter-logo:hover,

.toolbar #youtube-logo:hover {

opacity: 0.8;

}

.content {

display: flex;

margin: 82px auto 32px;

padding: 0 16px;

max-width: 960px;

flex-direction: column;

align-items: center;

}

svg.material-icons {

height: 24px;

width: auto;

}

svg.material-icons:not(:last-child) {

margin-right: 8px;

}

.card svg.material-icons path {

fill: #888;

45

}

.card-container {

display: flex;

flex-wrap: wrap;

justify-content: center;

margin-top: 16px;

}

.card {

all: unset;

border-radius: 4px;

border: 1px solid #eee;

background-color: #fafafa;

height: 40px;

width: 200px;

margin: 0 8px 16px;

padding: 16px;

display: flex;

flex-direction: row;

justify-content: center;

align-items: center;

transition: all 0.2s ease-in-out;

line-height: 24px;

}

.card-container .card:not(:last-child) {

margin-right: 0;

46

}

.card.card-small {

height: 16px;

width: 168px;

}

.card-container .card:not(.highlight-card) {

cursor: pointer;

}

.card-container .card:not(.highlight-card):hover {

transform: translateY(-3px);

box-shadow: 0 4px 17px rgba(0, 0, 0, 0.35);

}

.card-container .card:not(.highlight-card):hover .material-icons path { fill: rgb(105, 103, 103);

}

.card.highlight-card {

background-color: #1976d2;

color: white;

font-weight: 600;

border: none;

width: auto;

min-width: 30%;

position: relative;

47

}

.card.card.highlight-card span {

margin-left: 60px;

}

svg#rocket {

width: 80px;

position: absolute;

left: -10px;

top: -24px;

}

svg#rocket-smoke {

height: calc(100vh - 95px);

position: absolute;

top: 10px;

right: 180px;

z-index: -10;

}

a,

a:visited,

a:hover {

color: #1976d2;

text-decoration: none;

}

48

a:hover {

color: #125699;

}

.terminal {

position: relative;

width: 80%;

max-width: 600px;

border-radius: 6px;

padding-top: 45px;

margin-top: 8px;

overflow: hidden;

background-color: rgb(15, 15, 16);

}

.terminal::before {

content: "\2022 \2022 \2022";

position: absolute;

top: 0;

left: 0;

height: 4px;

background: rgb(58, 58, 58);

color: #c2c3c4;

width: 100%;

font-size: 2rem;

line-height: 0;

padding: 14px 0;

text-indent: 4px;

49

}

.terminal pre {

font-family: SFMono-Regular,Consolas,Liberation Mono,Menlo,monospace;

color: white;

padding: 0 1rem 1rem;

margin: 0;

}

.circle-link {

height: 40px;

width: 40px;

border-radius: 40px;

margin: 8px;

background-color: white;

border: 1px solid #eeeeee;

display: flex;

justify-content: center;

align-items: center;

cursor: pointer;

box-shadow: 0 1px 3px rgba(0, 0, 0, 0.12), 0 1px 2px rgba(0, 0, 0, 0.24);

transition: 1s ease-out;

}

.circle-link:hover {

transform: translateY(-0.25rem);

box-shadow: 0px 3px 15px rgba(0, 0, 0, 0.2);

}

50

footer {

margin-top: 8px;

display: flex;

align-items: center;

line-height: 20px;

}

footer a {

display: flex;

align-items: center;

}

.github-star-badge {

color: #24292e;

display: flex;

align-items: center;

font-size: 12px;

padding: 3px 10px;

border: 1px solid rgba(27,31,35,.2);

border-radius: 3px;

background-image: linear-gradient(-180deg,#fafbfc,#eff3f6 90%);

margin-left: 4px;

font-weight: 600;

}

.github-star-badge:hover {

background-image: linear-gradient(-180deg,#f0f3f6,#e6ebf1 90%);

51

border-color: rgba(27,31,35,.35);

background-position: -.5em;

}

.github-star-badge .material-icons {

height: 16px;

width: 16px;

margin-right: 4px;

}

svg#clouds {

position: fixed;

bottom: -160px;

left: -230px;

z-index: -10;

width: 1920px;

}

/\* Responsive Styles \*/

@media screen and (max-width: 767px) {

.card-container > \*:not(.circle-link) ,

.terminal {

width: 100%;

}

.card:not(.highlight-card) {

height: 16px;

margin: 8px 0;

52

}

.card.highlight-card span {

margin-left: 72px;

}

svg#rocket-smoke {

right: 120px;

transform: rotate(-5deg);

}

}

@media screen and (max-width: 575px) {

svg#rocket-smoke {

display: none;

visibility: hidden;

}

}

</style>

<!-- Toolbar -->

<div class="toolbar" role="banner">

<!-- <app-digital-clock></app-digital-clock>

<a style="text-align: right;">Login</a>

<a style="text-align: right;">Sign-Up</a> -->

</div>

<div>

<a routerLink="">

53

<div class="content">

<span><h2>Cyber Cell Software</h2><br><br>

</span>

</div>

</a>

<input type="hidden" #selection>

<div class="card-container">

<a routerLink="complaint">

<button class="card card-small" (click)="selection.value = 'component'" tabindex="0">

<svg class="material-icons" xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24"><path d="M19 13h-6v6h-2v-6H5v-2h6V5h2v6h6v2z"/></svg>

<span>Register New Complaint</span>

</button>

</a>

<a routerLink="edit">

<button class="card card-small" (click)="selection.value = 'material'" tabindex="0">

<svg class="material-icons" xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24"><path d="M19 13h-6v6h-2v-6H5v-2h6V5h2v6h6v2z"/></svg>

<span>Edit Existing Complaint</span>

</button>

</a>

<a routerLink="display">

<button class="card card-small" (click)="selection.value = 'component'" tabindex="0">

<svg class="material-icons" xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24"><path d="M19 13h-6v6h-2v-6H5v-2h6V5h2v6h6v2z"/></svg>

<span>Display Existing Complaint</span>

54

</button>

</a>

<a routerLink="download">

<button class="card card-small" (click)="selection.value = 'pwa'" tabindex="0">

<svg class="material-icons" xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24"><path d="M19 13h-6v6h-2v-6H5v-2h6V5h2v6h6v2z"/></svg>

<span>Download Complaints</span>

</button>

</a>

</div>

<div class="spacer">

<br><br>

<app-chart-enable></app-chart-enable>

</div>

</div>

<div class="content" role="main">

55