Disaster Management Application

Mobile Computing WS 20/21 Project Documentation
Guide: Prof. Dr .Trick, Mr. Frick
Submitted By : Yash Vyas 1266490
vyas@stud.fra-uas.de
Group No : B – 29

ACKNOWLEDGEMENT

I would like to thank Prof. Dr. Trick and Mr. Frick, for the continuous support and guidance provided for this project. Furthermore, I would also like to put my sincere gratitude for providing all the necessary information and useful links via moodle which proved to be very helpful for the successful completion of this project

TABLE OF CONTENTS

***Abstract*—In catastrophic situations WMN based disaster network infrastructure can be implemented using battery powered routers. On this network foundation Disaster Management application is deployed which can be used for human resources management and quick communication between Govt. different departments and victim in case of emergency.**

Keywords—WrelessMeshNetwork WMN, Network Function Virtualization NFV , node , communication infrastructure , Web application , spring boot application , MySQL db. Server, Docker Application, RescueTeam, Request Generation, Authentication, Resilience network

BACKGROUND

Following project is developed according to the scope of the research project VirtO4WMN of Prof. Dr. Trick's research group of telecommunications networks. The research project investigates the possibility to deploy an emergency communication infrastructure in case of a natural disaster destroyed or heavily damaged the existing communication infrastructure. The emergency communication infrastructure is constructed from battery-supplied multi-radio wireless outdoor-routers which are deployed by first responders in the affected region. The outdoor-routers are connecting to each other and are building a wireless mesh network (WMN)-based disaster network, which provides the basis for an IP-based communication. Various network functionalities and services for the end-users are provided within the WMN-based disaster network by utilizing the concept of NFV.

I. INTRODUCTION

Catastrophic events can disrupt the telecommunication infrastructure on which today's society is dependent for its proper functioning and maintaining order in the region. During such events region is affected for a prolonged period which brings the necessity of a network that can recover quickly and can maintain its functioning flexibly as there can be circumstances when hardware may become faulty.
In [1] authors have provided scientific derivation and definition for the prior setting of WMN based disaster network, and definition of challenges which possibly can result from environmental based events. Also, emergency corrections capable of resolving the influences of an occurring challenge are defined.
Following disaster network will be deployed in the region with the help of first responders using battery-supplied multi-radio wireless outdoor routers. These routers will establish a cluster-based mesh network which will build a foundation for IP-based network infrastructure. Routers are particularly used to integrate network function virtualization (NFV) into the mesh network which will help disaster network to maintain adaptable network service and better process provisioning. Router resources will be utilized to establish the required NFV-Infrastructure. [1]
The disaster network needs to function and render its functionality to the user at any event to remain ready and thus usable. In [2] authors present various necessities for the WMN-based disaster network. Among other necessities, a continuous working infrastructure, as well as the support of flexible and dynamic infrastructure, are crucial for the disaster network. This happens from the perspective, that the net-work

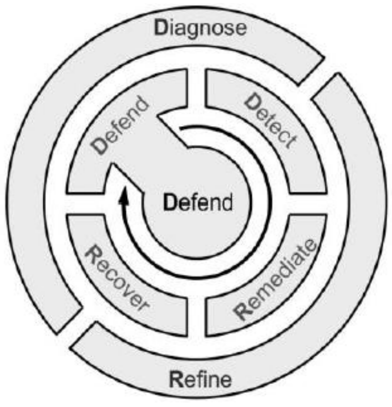


Figure 1 D2R2+DR model **[4]**

foundation of the disaster network is permanently endangered to predictable and unplanned occur-ring events, such as node failures. These events result from the disaster environment and the concept of the WMN-based disaster network itself. Among other circumstances, an existing node might fail either due to a drained battery or spontaneously due to its destruction resulting from an aftershock or outbreak-ing fire resulting from the catastrophe. Also, a new node might join the network due to the deployment of a corresponding new outdoor router by the first responders for a geographical expansion of the network. Network availability is distributed among the nodes which are the primary function of NFV orchestration to guarantee a functioning communication infrastructure.
In [3] authors have provided standardized NFV orchestrator which are logically "centralized component" for a failure at a single point As NFV orchestrator standardized in [3] is a logically "centralized component" that provides an opportunity for a "singular brings the necessity of particular NFV orchestration which can sustain the network infrastructure during on-going events.
In [1] authors introduced a normal state of WMN operational network based on the D2R2+DR model which was introduced in [4] which includes its services requirements, possible challenges and emergency measures and corrections to maintain its pre-defined states.

On this network foundation, Disaster Management application developed can be used for resources management and communication application by all people i.e. citizen, victim, and Govt. Department employees i.e. rescue teams, management department employees for stability in the region.

In [5] authors have complied possible challenges that are usually faced by communication network for disaster operations such as ***Popularity, Usability, Practicability, Capacity, Sustainability, Adaptability, Operability, Connectivity, Security*** detailed explanation can be found in [6] [7] [8]

In [5] authors have described the event when terminal needs the communication infrastructure to start the communication as drawbacks of device-to-device mode.

Next section of report is structured as follow section [II] project details ***defines aim, use case diagram, description, tech-stack, network architecture, application architecture, implementation strategy of application, application testing, guide .***

II. Project Details

A. Aim: Self-contained dockerized web application for management of human resources of different Govt-departments and communication links between people in the affected region with the employees.

B. Use Case:

1) Decsription:

a) Medical Assistance:

Victim of calamities can suffer medical injuries which can require quick transportation, guidance regarding the condition of the patient, and

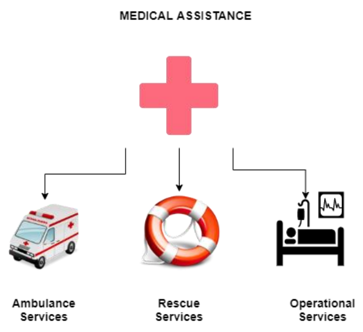


Figure 2 *Medical Assistance*

other problems can be solved by communicating with the medical team assigned to the victim via Disaster Management applications.

b) Fire Assistance:

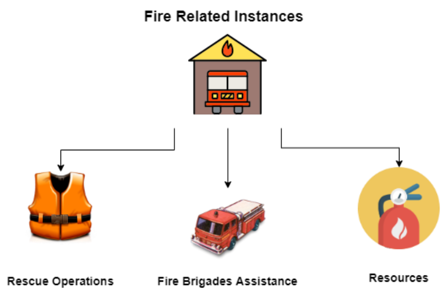


Figure 3 *Fire Assistance*

During calamities, fire outbreaks can happen which can cause serious damage to the region's people and infrastructure. During such scenario, victim can contact fire department which will assign the victim's request to fire department's rescue team which will be in constant communication with the victim while reaching the victim's location.

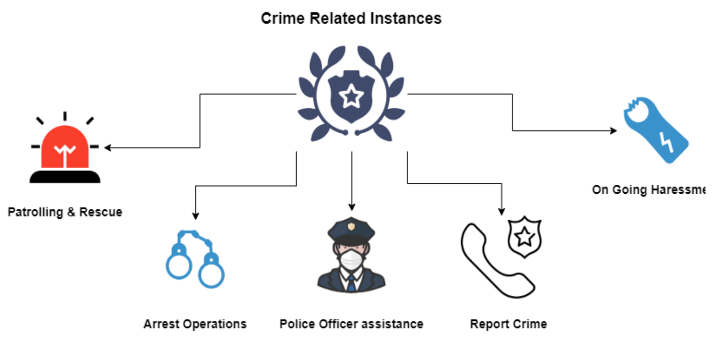


Figure 4 *Crime Assistance*

c) Crime Assistance:

Victim can report crime and ask for assistance, via Disaster Management application to contact the police authority. Different examples of assistances: Ambulance Service. Doctor's advice. Rescue Operations. Fire Brigade Crime Report.

C. UML Diagrams:

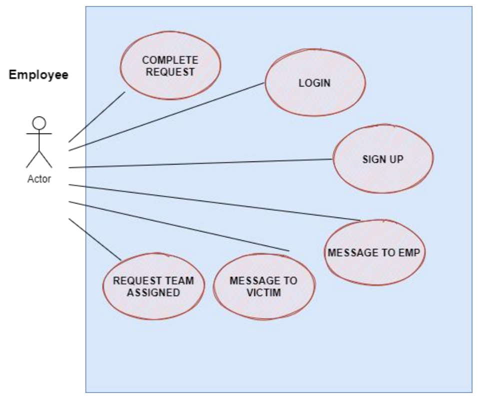


Figure 5 *Employee/User UML*

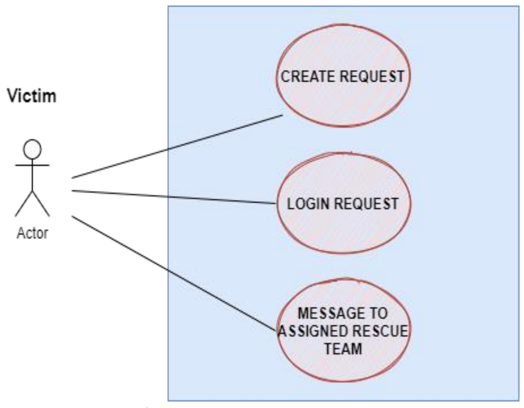


Figure 7 *Victim UML*

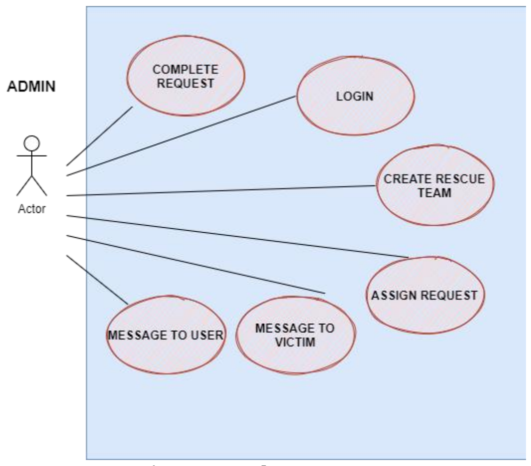


Figure 6 *Admin UML*

D. Architecture:

1) Network Architecture:

WMN architecture can be classified into three types [9]:

a) BackBone:

The mesh routers/gateways form an infrastructure for clients and can be connected to various networks, e.g. Internet. Furthermore, the mesh routers have minimal mobility. [5]

b) Client:

Client meshing provides peer-to-peer networks among client devices. In this type of architecture, client nodes constitute the actual network to perform end-user applications to customers. [5]

c) Hybrid:

This architecture is the combination of backbone and client meshing, as shown in Fig. 8. Mesh clients can access the network through mesh routers and directly connect to other mesh clients. The backbone provides connectivity to other networks [5]

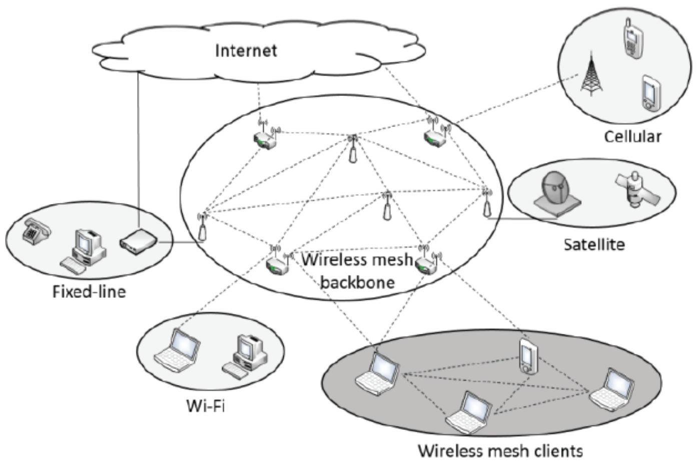


Figure 8 *Hybrid WMN network* **[5]**

Normal State WMN network:
Several NSC [Normal state Cluster] together constitute WMN network

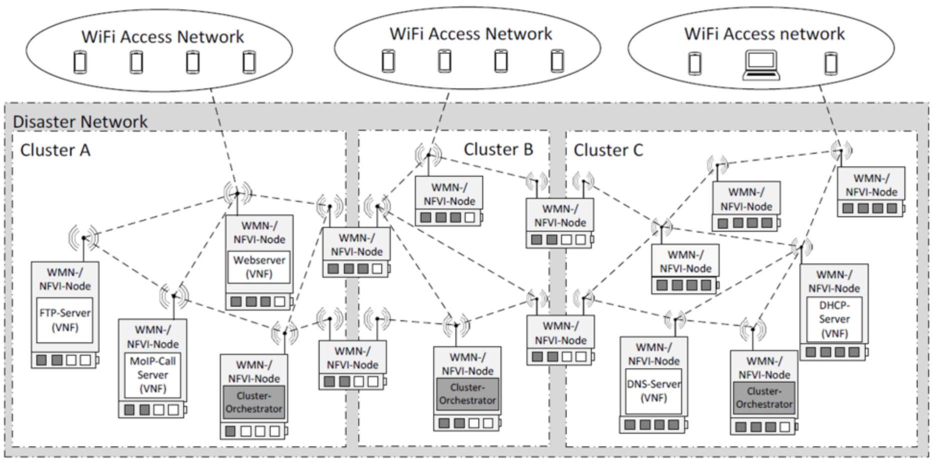


Figure 9 *Normal State of WMN network* **[2]**

2) *Web Application Architecture:*

- The **client server** is the presentation layer and provides the user interface. Following is provided using angular application.
- The **application server** corresponds to the application layer, housing the business logic used to process user inputs and queries the entities related to the project to be stored in the database. Application server is provided using spring boot application.
- The **database server** is the data layer of a web application. It runs on database management software, for the following application we are using MySQL

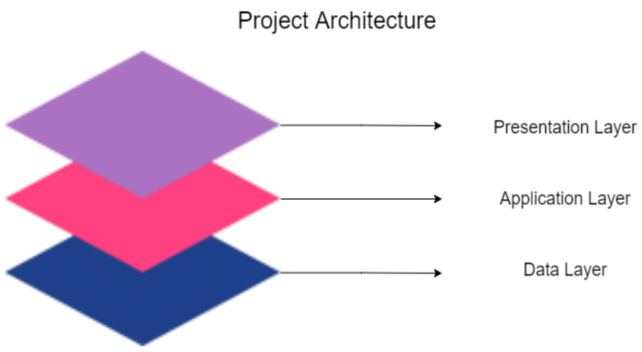


Figure 10 *Project Architecture*

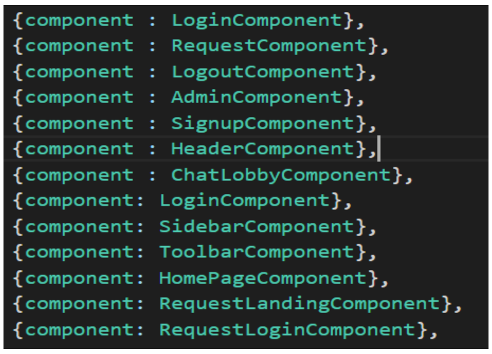
E. *Implementation Strategy*

1) *Angular application:*

a) Introduction

Angular is Typescript based open source web application framework

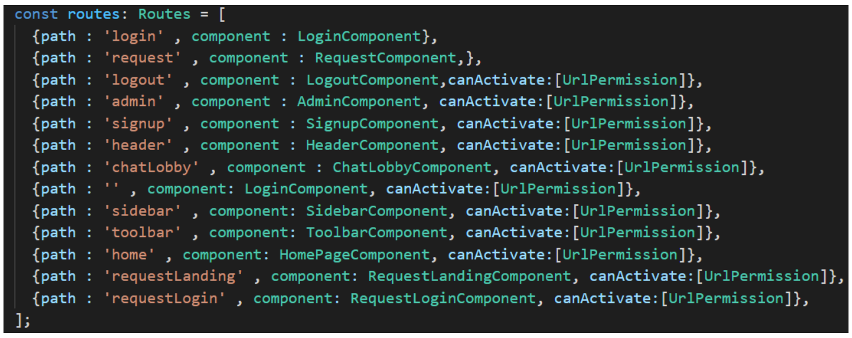
b) Components:



```
{component : LoginComponent},
{component : RequestComponent},
{component : LogoutComponent},
{component : AdminComponent},
{component : SignupComponent},
{component : HeaderComponent},
{component : ChatLobbyComponent},
{component: LoginComponent},
{component: SidebarComponent},
{component: ToolbarComponent},
{component: HomePageComponent},
{component: RequestLandingComponent},
{component: RequestLoginComponent},
```

Figure 11 *All Angular Components*

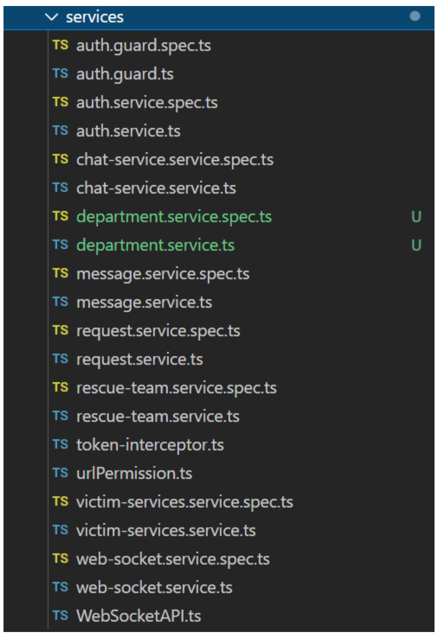
c) Routes:



```
const routes: Routes = [
  {path : 'login' , component : LoginComponent},
  {path : 'request' , component : RequestComponent,},
  {path : 'logout' , component : LogoutComponent,canActivate:[UrlPermission]},
  {path : 'admin' , component : AdminComponent, canActivate:[UrlPermission]},
  {path : 'signup' , component : SignupComponent, canActivate:[UrlPermission]},
  {path : 'header' , component : HeaderComponent, canActivate:[UrlPermission]},
  {path : 'chatLobby' , component : ChatLobbyComponent, canActivate:[UrlPermission]},
  {path : '' , component: LoginComponent, canActivate:[UrlPermission]},
  {path : 'sidebar' , component: SidebarComponent, canActivate:[UrlPermission]},
  {path : 'toolbar' , component: ToolbarComponent, canActivate:[UrlPermission]},
  {path : 'home' , component: HomePageComponent, canActivate:[UrlPermission]},
  {path : 'requestLanding' , component: RequestLandingComponent, canActivate:[UrlPermission]},
  {path : 'requestLogin' , component: RequestLoginComponent, canActivate:[UrlPermission]},
];
```

Figure 12 *Angular routes*

d) Services:

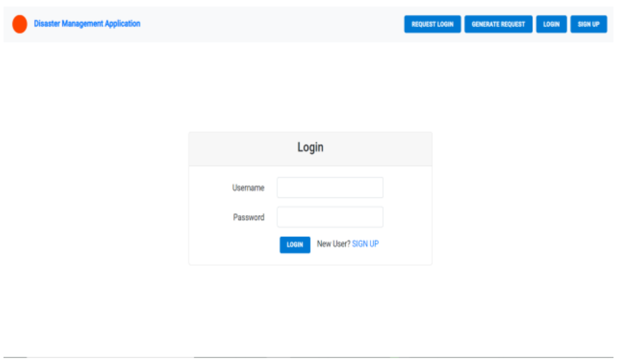


```
services
TS auth.guard.spec.ts
TS auth.guard.ts
TS auth.service.spec.ts
TS auth.service.ts
TS chat-service.service.spec.ts
TS chat-service.service.ts
TS department.service.spec.ts    U
TS department.service.ts         U
TS message.service.spec.ts
TS message.service.ts
TS request.service.spec.ts
TS request.service.ts
TS rescue-team.service.spec.ts
TS rescue-team.service.ts
TS token-interceptor.ts
TS urlPermission.ts
TS victim-services.service.spec.ts
TS victim-services.service.ts
TS web-socket.service.spec.ts
TS web-socket.service.ts
TS WebSocketAPI.ts
```

Figure 13 *Angular Services*

e) Dialogs & ScreemShots:

- Login Page: Input Username | Password.
- Signup Page: Input Email | Username | Password | Departments.
- Victim Login Page: Input Victim name | location.
- Create Request: Input Victim name | no. of people | location | nature as assistance needed.



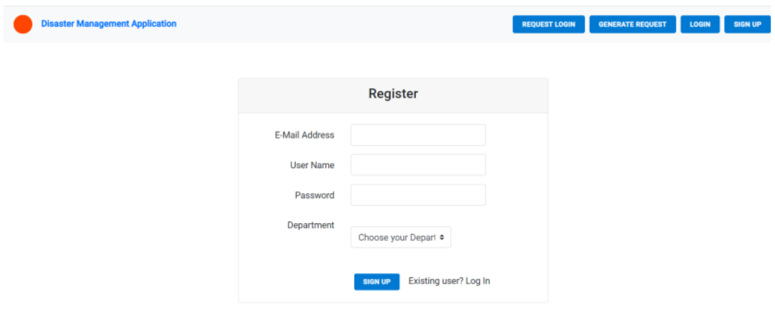
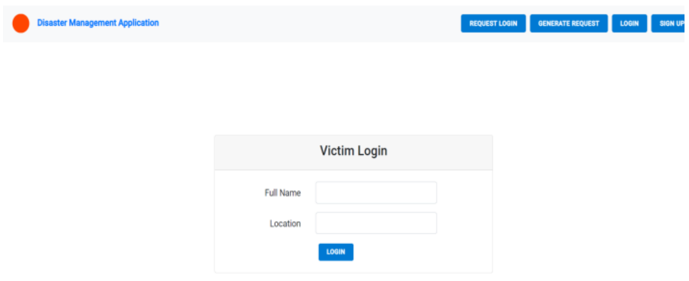


Figure 14 *Login Page*



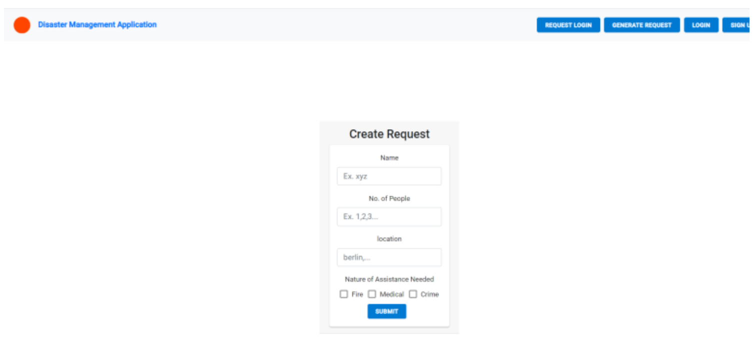
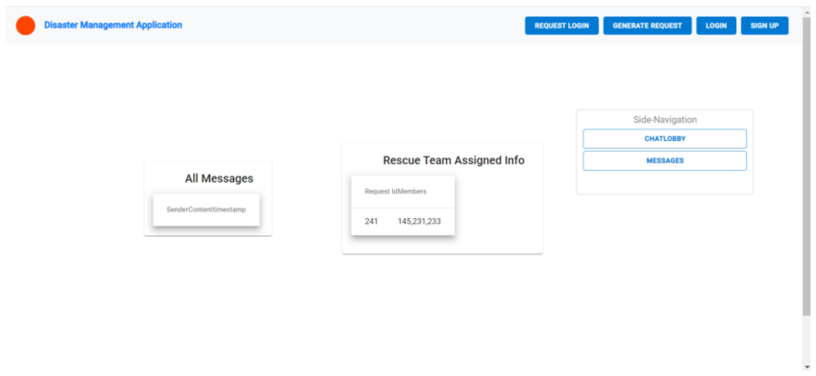


Figure 17 *Create Request*

Figure 15 *Victim Login Page*

- Victim Dashboard: Victim can see all messages sent to him | also can see rescue Team assigned details.



All Messages

SenderContenttimestamp

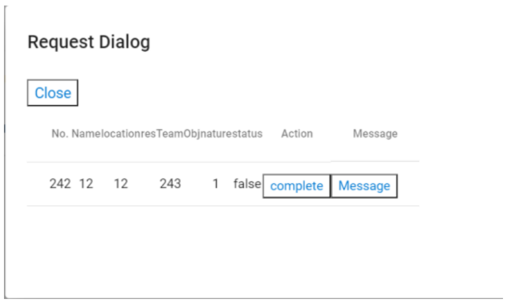
Rescue Team Assigned Info

Request Id	Members
241	145,231,233

Figure 18 *Victim Dashboard*

- Request Dialog: User/Admin can see request dialog which shows request assigned to user participated rescue Team. Action button to complete request or message victim is also provided.

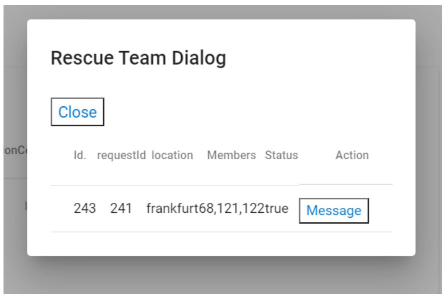
- Rescue Team Dialog: User/Admin can see rescue team details and action button to message rescue team.



Request Dialog

No.	Name	location	resTeamObj	nature	status	Action	Message
242	12	12	243	1	false	complete	Message

Figure 20 *Request Dialog*

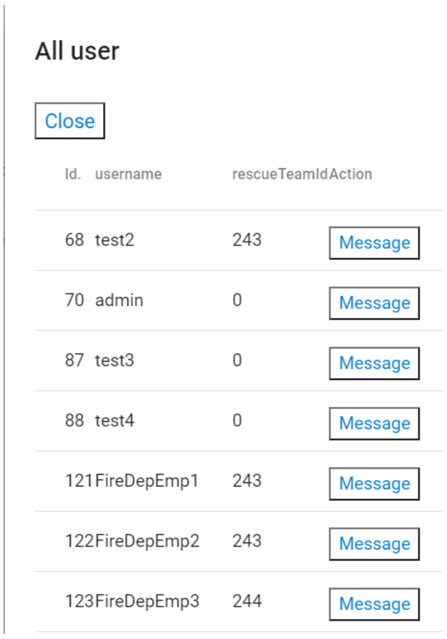


Rescue Team Dialog

Id.	requestId	location	Members	Status	Action
243	241	frankfurt	68,121,122	true	Message

Figure 19 *Rescue Team Dialog*

- All Users Dialog: Admin/User can see all other employees from different departments | Message can also be send to other user.

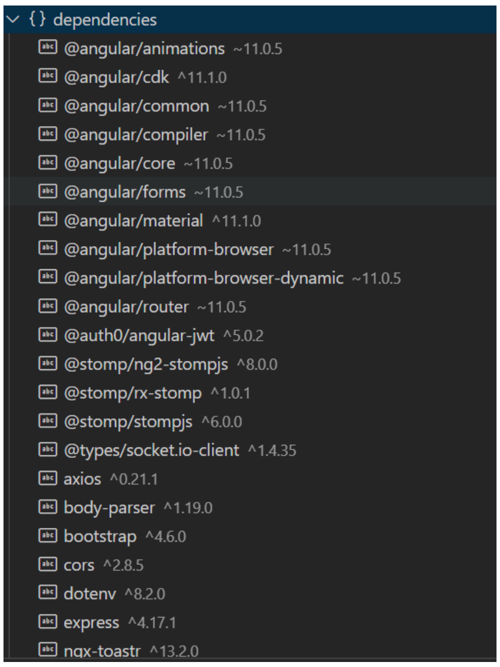


All user

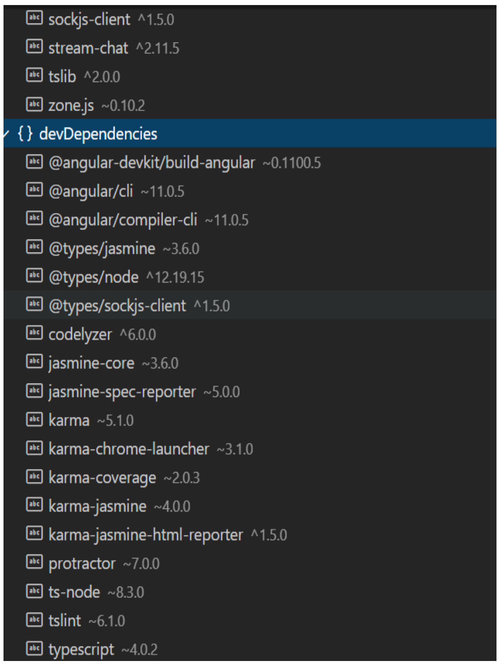
Id.	username	rescueTeamId	Action
68	test2	243	Message
70	admin	0	Message
87	test3	0	Message
88	test4	0	Message
121	FireDepEmp1	243	Message
122	FireDepEmp2	243	Message
123	FireDepEmp3	244	Message

Figure 21 *Employees Dialog*

f) Dependencies:



```
{ } dependencies
    @angular/animations ~11.0.5
    @angular/cdk ^11.1.0
    @angular/common ~11.0.5
    @angular/compiler ~11.0.5
    @angular/core ~11.0.5
    @angular/forms ~11.0.5
    @angular/material ^11.1.0
    @angular/platform-browser ~11.0.5
    @angular/platform-browser-dynamic ~11.0.5
    @angular/router ~11.0.5
    @auth0/angular-jwt ^5.0.2
    @stomp/ng2-stompjs ^8.0.0
    @stomp/rx-stomp ^1.0.1
    @stomp/stompjs ^6.0.0
    @types/socket.io-client ^1.4.35
    axios ^0.21.1
    body-parser ^1.19.0
    bootstrap ^4.6.0
    cors ^2.8.5
    dotenv ^8.2.0
    express ^4.17.1
    ngx-toastr ^13.2.0
```



```
    sockjs-client ^1.5.0
    stream-chat ^2.11.5
    tslib ^2.0.0
    zone.js ~0.10.2
{ } devDependencies
    @angular-devkit/build-angular ~0.1100.5
    @angular/cli ~11.0.5
    @angular/compiler-cli ~11.0.5
    @types/jasmine ~3.6.0
    @types/node ^12.19.15
    @types/sockjs-client ^1.5.0
    codelyzer ^6.0.0
    jasmine-core ~3.6.0
    jasmine-spec-reporter ~5.0.0
    karma ~5.1.0
    karma-chrome-launcher ~3.1.0
    karma-coverage ~2.0.3
    karma-jasmine ~4.0.0
    karma-jasmine-html-reporter ^1.5.0
    protractor ~7.0.0
    ts-node ~8.3.0
    tslint ~6.1.0
    typescript ~4.0.2
```

Figure 22 *Angular app dependencies*

2) Spring boot application:

a) Introduction:

The Spring Framework is an application framework and inversion of control container for the Java platform. The framework's core features can be used by any Java application, but there are extensions for building web applications on top of the Java EE (Enterprise Edition) platform. Although the framework does not impose any specific programming model, it has become popular in the Java community as an addition to the Enterprise JavaBeans (EJB) model. The Spring Framework is open source. [10]

b) Configurations:

❖ *Web Security Adapter:*

Following adapter is part of Spring Security.

By configuring it, we can secure our application. During its configuration we assign the endpoints which can be accessed by authenticated users and further aspects regarding application security,

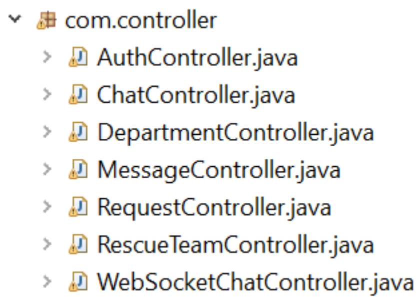
❖ *Web Socket Message Broker:*

A WebSocket is a persistent connection between a client and server. WebSockets provide a bidirectional, full-duplex communications channel that operates over HTTP through a single TCP/IP socket connection. At its core, the WebSocket protocol facilitates message passing between a client and server.

❖ *Web MVC:*

It's a module of the spring framework dealing with the Model-View-Controller, or MVC pattern. It combines all the advantages of the MVC pattern with the convenience of spring *c) Controllers:*

Rest Controller annotation is used to create RESTful web services using Spring MVC. Spring Rest Controller takes care of mapping request data to the defined request handler method.

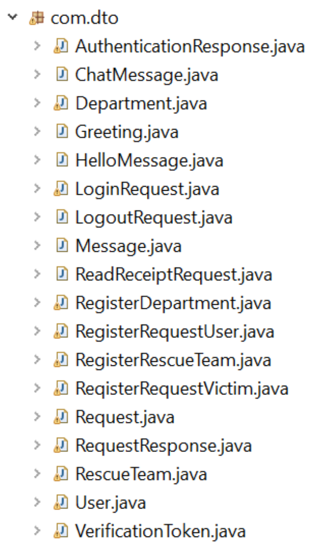


```
com.controller
  AuthController.java
  ChatController.java
  DepartmentController.java
  MessageController.java
  RequestController.java
  RescueTeamController.java
  WebSocketChatController.java
```

Figure 23 *Spring Controllers*

d) JPA Entites:

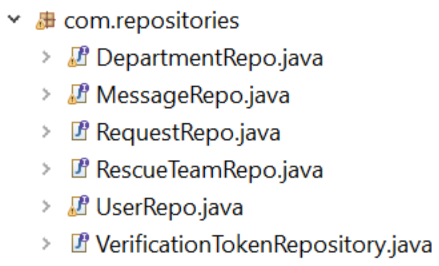
Entities represent persistent data stored in a relational database automatically using container-managed persistence. .An entity can aggregate objects together and effectively persist data and related objects using the transactional, security, and concurrency services of a JPA persistence provider.



```
com.dto
  AuthenticationResponse.java
  ChatMessage.java
  Department.java
  Greeting.java
  HelloMessage.java
  LoginRequest.java
  LogoutRequest.java
  Message.java
  ReadReceiptRequest.java
  RegisterDepartment.java
  RegisterRequestUser.java
  RegisterRescueTeam.java
  ReqisterRequestVictim.java
  Request.java
  RequestResponse.java
  RescueTeam.java
  User.java
  VerificationToken.java
```

Figure 24 *JPA entities*

e) Repositories:



```
com.repositories
  DepartmentRepo.java
  MessageRepo.java
  RequestRepo.java
  RescueTeamRepo.java
  UserRepo.java
  VerificationTokenRepository.java
```

Figure 25 *JPA repositories*

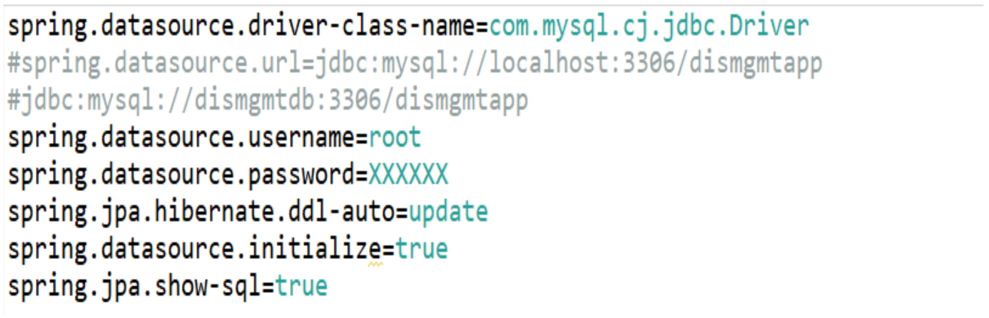
The Java Persistence API (JPA) is the standard way of persisting Java objects into relational databases. The JPA consists of two parts: a mapping subsystem to map classes onto relational tables as well as an Entity Manager API to access the objects, define and execute queries, and more.

3) MySql Database Server:

a) Introduction

It is an open source Relational Database Management System (RDBMS) that uses Structured Query Language (SQL). SQL is the most popular language for adding, accessing and managing content in a database. It is most noted for its quick processing, proven reliability, ease and flexibility of use.

b) Configuration with spring boot



```
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
#spring.datasource.url=jdbc:mysql://localhost:3306/dismgmtapp
#jdbc:mysql://dismgmtdb:3306/dismgmtapp
spring.datasource.username=root
spring.datasource.password=XXXXXX
spring.jpa.hibernate.ddl-auto=update
spring.datasource.initialize=true
spring.jpa.show-sql=true
```

Figure 26 *Spring Boot database configuration*

4) FlowChart:

a) Request handling:

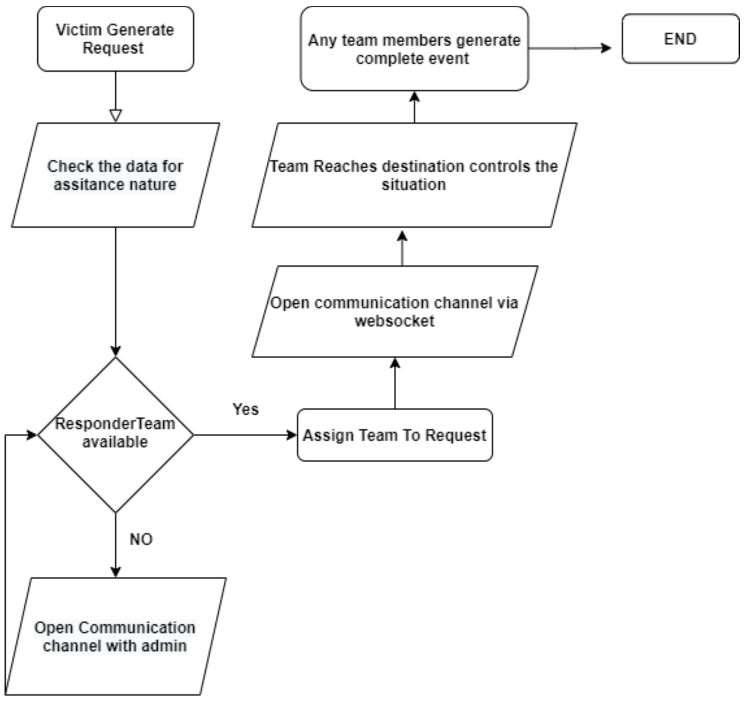

Victim Generate Request
Check the data for assitance nature
ResponderTeam available
Yes
Assign Team To Request
NO
Open Communication channel with admin
Open communication channel via websocket
Team Reaches destination controls the situation
Any team members generate complete event
END

Figure 27 *FlowChart requestHandling*

5) Network Call:

a) Request Handling:

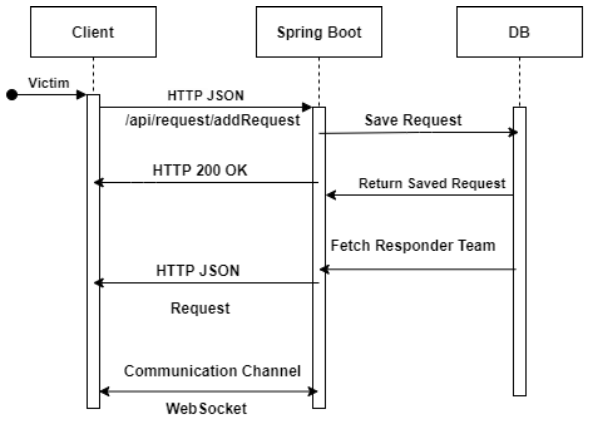


Figure 28 *NetworkCall-requestHandling*

6) Entity-Relationship Diagram:

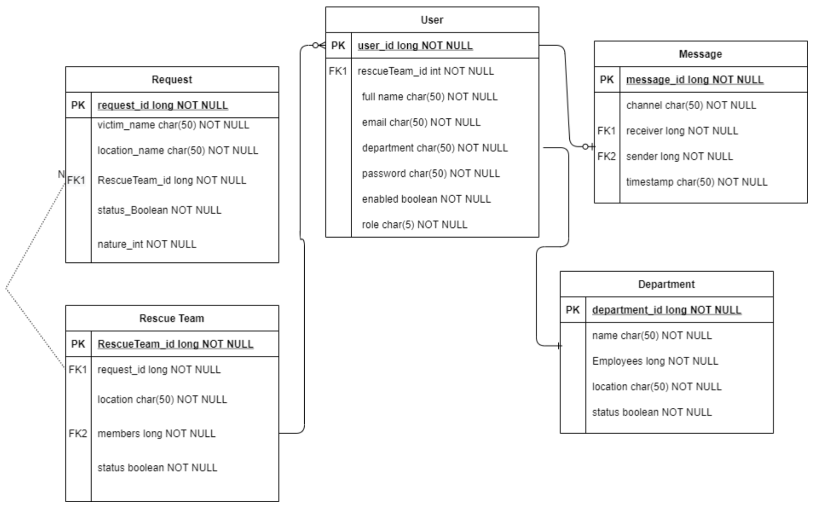

User
PK user_id long NOT NULL
FK1 rescueTeam_id int NOT NULL
full name char(50) NOT NULL
email char(50) NOT NULL
department char(50) NOT NULL
password char(50) NOT NULL
enabled boolean NOT NULL
role char(5) NOT NULL
Message
PK message_id long NOT NULL
channel char(50) NOT NULL
FK1 receiver long NOT NULL
FK2 sender long NOT NULL
timestamp char(50) NOT NULL
Request
PK request_id long NOT NULL
victim_name char(50) NOT NULL
location_name char(50) NOT NULL
N
FK1 RescueTeam_id long NOT NULL
status_Boolean NOT NULL
nature_int NOT NULL
Rescue Team
PK RescueTeam_id long NOT NULL
FK1 request_id long NOT NULL
location char(50) NOT NULL
FK2 members long NOT NULL
status boolean NOT NULL
Department
PK department_id long NOT NULL
name char(50) NOT NULL
Employees long NOT NULL
location char(50) NOT NULL
status boolean NOT NULL

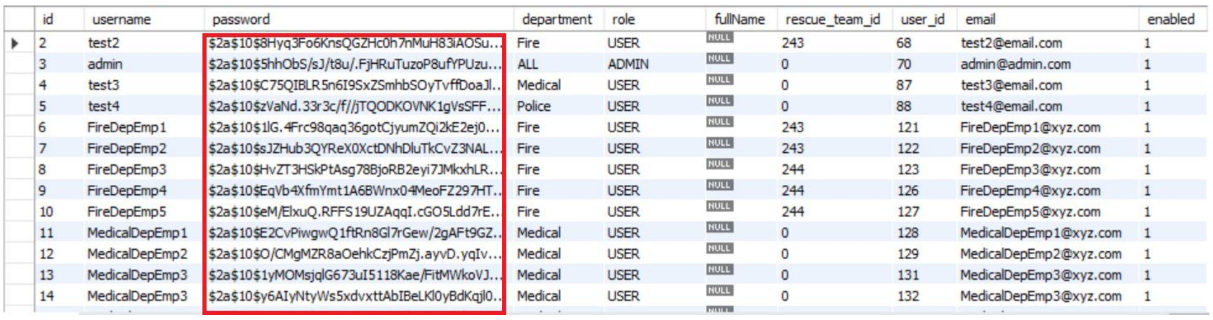
Figure 29 *Entity Relationship*

7) API's

/***api/auth*** : /signup / login /logout /userByUsername /getAllEmp
/***api/department*** : /addDepartment /getAll /logout /userByUsername /getAllEmp
/***api/request*** : /addRequest /getById /getAll /Completerequest /getAllEmp /victimLogin /getUserActiveRequest /getAllActiveRequest
/***api/rescueTeam*** : /addRescueTeam /getById /getRequestByRescueTeamId /getAll /assignRequestToResTeam
/***api/chat*** : /{roomId}/sendMessage /{requestId}/send / {rescueTeamId}/sendToRescueTeam /{userId}/sendToUser /{roomId}/addUser

8) *Data Security:*

JWT token for authentication and Password encoders are used during the saving process of data in DB.



id	username	password	department	role	fullName	rescue_team_id	user_id	email	enabled
2	test2	$2a$10$8Hyq3Fo6KnsQGZHc0h7nMuH83iAOSu...	Fire	USER	NULL	243	68	test2@email.com	1
3	admin	$2a$10$5hhObS/sJ/t8u/.FjHRuTuzoP8ufYPUzu...	ALL	ADMIN	NULL	0	70	admin@admin.com	1
4	test3	$2a$10$C75QIBLR5n6I9SxZSmhbSOyTvffDoaJl...	Medical	USER	NULL	0	87	test3@email.com	1
5	test4	$2a$10$zVaNd.33r3c/f//jTQODKOVNK1gVsSFF...	Police	USER	NULL	0	88	test4@email.com	1
6	FireDepEmp1	$2a$10$1lG.4Frc98qaq36gotCjyumZQi2kE2ej0...	Fire	USER	NULL	243	121	FireDepEmp1@xyz.com	1
7	FireDepEmp2	$2a$10$sJZHub3QYReX0XctDNhDluTkCvZ3NAL...	Fire	USER	NULL	243	122	FireDepEmp2@xyz.com	1
8	FireDepEmp3	$2a$10$HvZT3HSkPtAsg78BjoRB2eyi7JMkxhLR...	Fire	USER	NULL	244	123	FireDepEmp3@xyz.com	1
9	FireDepEmp4	$2a$10$EqVb4XfmYmt1A6BWnx04MeoFZ297HT...	Fire	USER	NULL	244	126	FireDepEmp4@xyz.com	1
10	FireDepEmp5	$2a$10$eM/ElxuQ.RFFS19UZAqqI.cGO5Ldd7rE...	Fire	USER	NULL	244	127	FireDepEmp5@xyz.com	1
11	MedicalDepEmp1	$2a$10$E2CvPiwgwQ1ftRn8Gl7rGew/2gAFt9GZ...	Medical	USER	NULL	0	128	MedicalDepEmp1@xyz.com	1
12	MedicalDepEmp2	$2a$10$O/CMgMZR8aOehkCzjPmZj.ayvD.yqIv...	Medical	USER	NULL	0	129	MedicalDepEmp2@xyz.com	1
13	MedicalDepEmp3	$2a$10$1yMOMsjqlG673uI5118Kae/FitMWkoVJ...	Medical	USER	NULL	0	131	MedicalDepEmp3@xyz.com	1
14	MedicalDepEmp3	$2a$10$y6AIyNtyWs5xdvxttAbIBeLKl0yBdKqjl0...	Medical	USER	NULL	0	132	MedicalDepEmp3@xyz.com	1

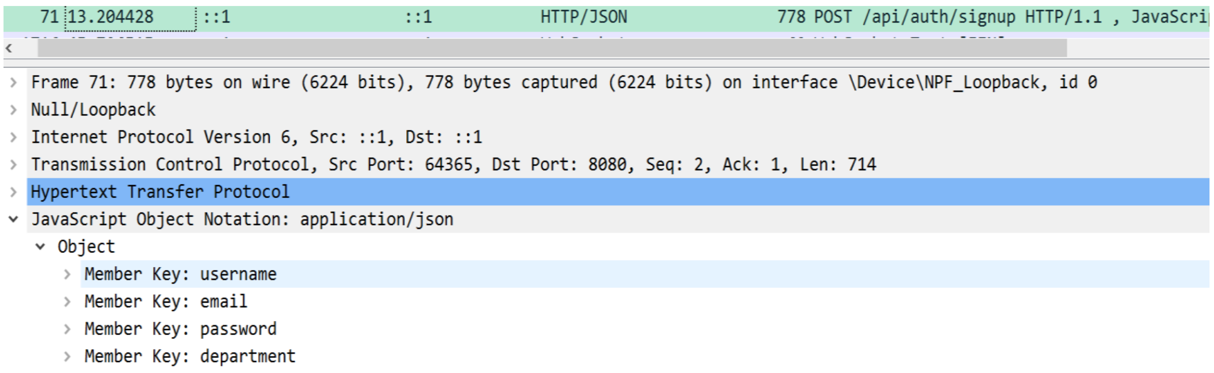
Figure 30 *Password Encryption*

F. *Application Tesing:*

1) *WireShrak Captures:*

a) *User/Employee Sign-up : Request , Response.*

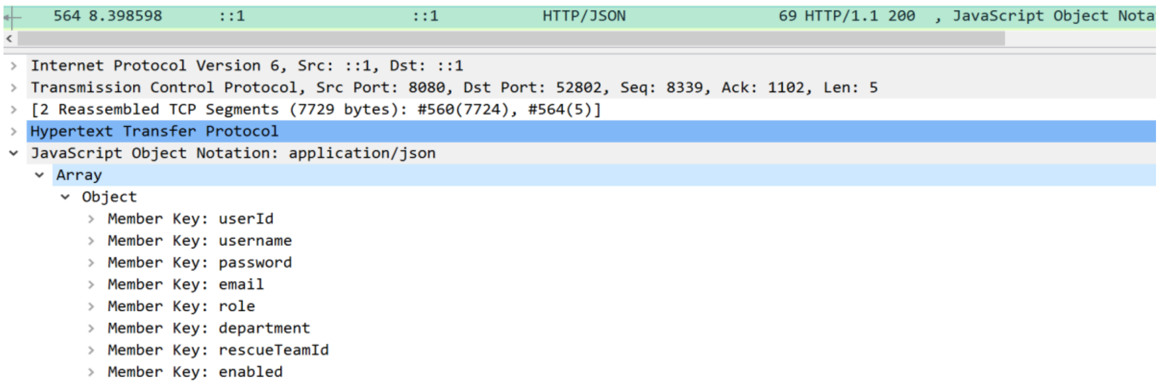
- /api/auth/signup POST request.



```
71 13.204428    ::1          ::1          HTTP/JSON          778 POST /api/auth/signup HTTP/1.1 , JavaScri
> Frame 71: 778 bytes on wire (6224 bits), 778 bytes captured (6224 bits) on interface \Device\NPF_Loopback, id 0
> Null/Loopback
> Internet Protocol Version 6, Src: ::1, Dst: ::1
> Transmission Control Protocol, Src Port: 64365, Dst Port: 8080, Seq: 2, Ack: 1, Len: 714
> Hypertext Transfer Protocol
v JavaScript Object Notation: application/json
  v Object
    > Member Key: username
    > Member Key: email
    > Member Key: password
    > Member Key: department
```

Figure 31-*SignUp request*

- Sign up response with userId department enabled and rescueTeamId assigned

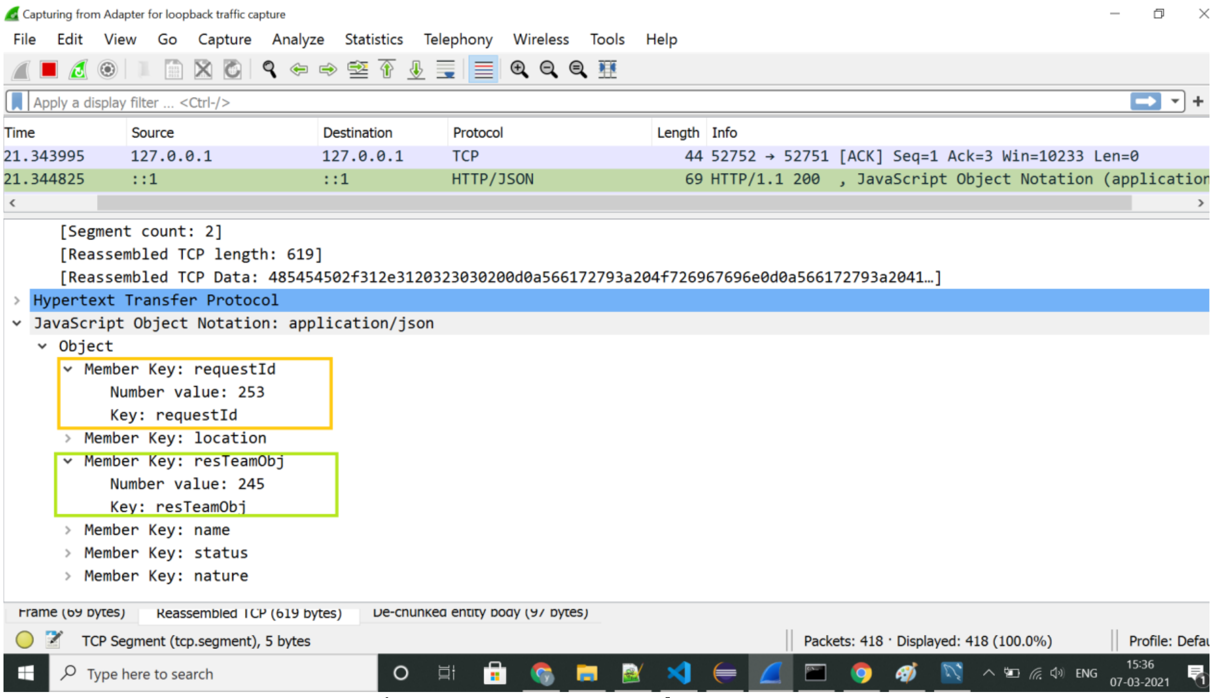


```
564 8.398598    ::1          ::1          HTTP/JSON          69 HTTP/1.1 200  , JavaScript Object Nota
> Internet Protocol Version 6, Src: ::1, Dst: ::1
> Transmission Control Protocol, Src Port: 8080, Dst Port: 52802, Seq: 8339, Ack: 1102, Len: 5
> [2 Reassembled TCP Segments (7729 bytes): #560(7724), #564(5)]
> Hypertext Transfer Protocol
v JavaScript Object Notation: application/json
  v Array
    v Object
      > Member Key: userId
      > Member Key: username
      > Member Key: password
      > Member Key: email
      > Member Key: role
      > Member Key: department
      > Member Key: rescueTeamId
      > Member Key: enabled
```

Figure 32 *SignUp response*

b) Request Assigned to user in RescueTeam/Responser Team:

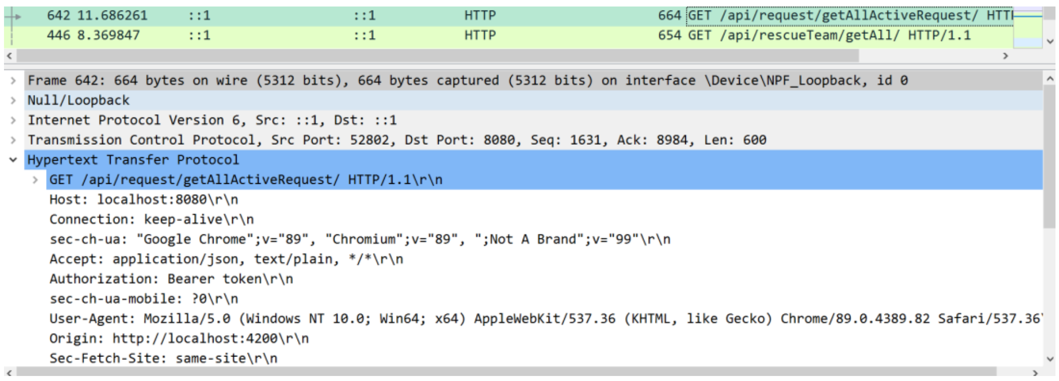
- Request Assigned to rescue team.



```
Capturing from Adapter for loopback traffic capture
File  Edit  View  Go  Capture  Analyze  Statistics  Telephony  Wireless  Tools  Help
Apply a display filter ... <Ctrl-/>

Time        Source      Destination  Protocol   Length  Info
21.343995   127.0.0.1   127.0.0.1    TCP            44  52752 → 52751 [ACK] Seq=1 Ack=3 Win=10233 Len=0
21.344825   ::1         ::1          HTTP/JSON      69  HTTP/1.1 200  , JavaScript Object Notation (applicatior

      [Segment count: 2]
      [Reassembled TCP length: 619]
      [Reassembled TCP Data: 485454502f312e3120323030200d0a566172793a204f726967696e0d0a566172793a2041…]
> Hypertext Transfer Protocol
v JavaScript Object Notation: application/json
  v Object
    v Member Key: requestId
        Number value: 253
        Key: requestId
    > Member Key: location
    v Member Key: resTeamObj
        Number value: 245
        Key: resTeamObj
    > Member Key: name
    > Member Key: status
    > Member Key: nature

Frame (69 bytes)   Reassembled TCP (619 bytes)   De-chunked entity body (97 bytes)
TCP Segment (tcp.segment), 5 bytes        Packets: 418 · Displayed: 418 (100.0%)    Profile: Defau
Type here to search                       ENG  15:36  07-03-2021
```

Figure 33 *Request Assigned to Rescue Team*

c) Active Request Assigned to user:

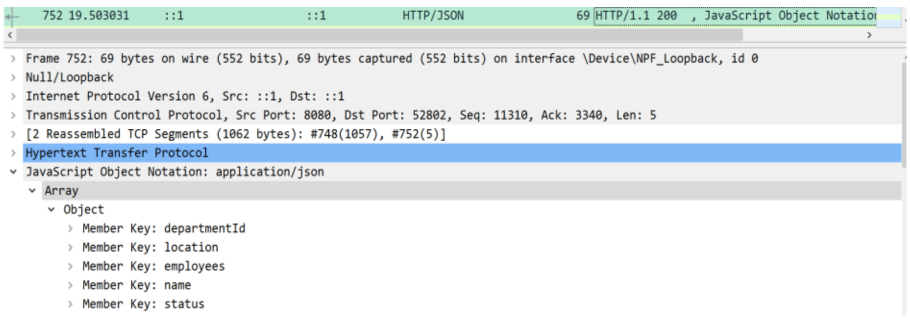


```
  642 11.686261   ::1          ::1          HTTP          664 GET /api/request/getAllActiveRequest/ HTTI
  446 8.369847    ::1          ::1          HTTP          654 GET /api/rescueTeam/getAll/ HTTP/1.1

> Frame 642: 664 bytes on wire (5312 bits), 664 bytes captured (5312 bits) on interface \Device\NPF_Loopback, id 0
> Null/Loopback
> Internet Protocol Version 6, Src: ::1, Dst: ::1
> Transmission Control Protocol, Src Port: 52802, Dst Port: 8080, Seq: 1631, Ack: 8984, Len: 600
v Hypertext Transfer Protocol
  > GET /api/request/getAllActiveRequest/ HTTP/1.1\r\n
    Host: localhost:8080\r\n
    Connection: keep-alive\r\n
    sec-ch-ua: "Google Chrome";v="89", "Chromium";v="89", ";Not A Brand";v="99"\r\n
    Accept: application/json, text/plain, */*\r\n
    Authorization: Bearer token\r\n
    sec-ch-ua-mobile: ?0\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.82 Safari/537.36
    Origin: http://localhost:4200\r\n
    Sec-Fetch-Site: same-site\r\n
```

Figure 34 *User Active Request*

d) Department:

- Departments’ details call from User/Admin component.



```
  752 19.503031    ::1          ::1          HTTP/JSON        69 HTTP/1.1 200  , JavaScript Object Notatio

> Frame 752: 69 bytes on wire (552 bits), 69 bytes captured (552 bits) on interface \Device\NPF_Loopback, id 0
> Null/Loopback
> Internet Protocol Version 6, Src: ::1, Dst: ::1
> Transmission Control Protocol, Src Port: 8080, Dst Port: 52802, Seq: 11310, Ack: 3340, Len: 5
> [2 Reassembled TCP Segments (1062 bytes): #748(1057), #752(5)]
> Hypertext Transfer Protocol
v JavaScript Object Notation: application/json
  v Array
    v Object
      > Member Key: departmentId
      > Member Key: location
      > Member Key: employees
      > Member Key: name
      > Member Key: status
```

Figure 35 *Department*

G. *Guide:*

1) *Installation:*

- Unzip Category_B-Group_29 | Go to Implementation/springboot-backend

  OR

  Download repositories from

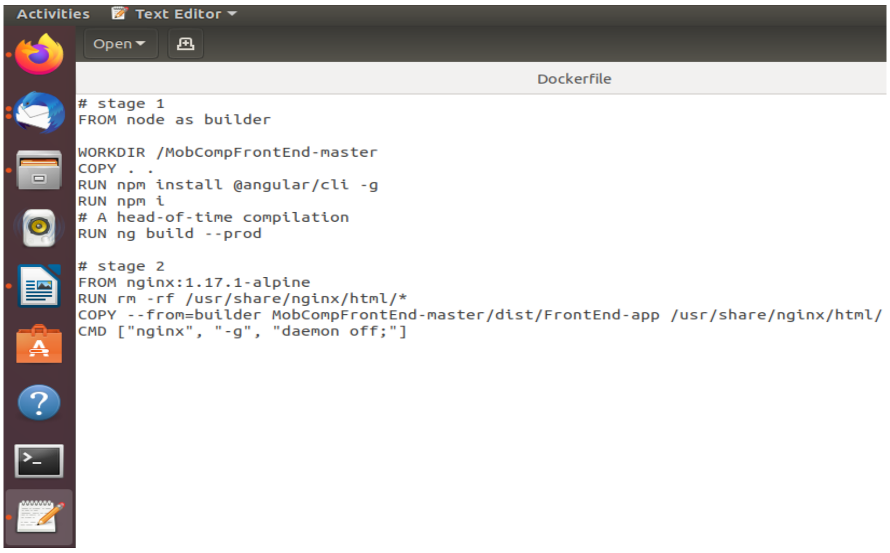
  [BackEnd: https://github.com/yashvyas95/MobCompBackEnd.git](https://github.com/yashvyas95/MobCompBackEnd.git)

  [FrontEnd: https://github.com/yashvyas95/MobCompFrontEnd.git](https://github.com/yashvyas95/MobCompFrontEnd.git)

  Install Required Dependencies mentioned in [section 2 E-f].

- Create Dockerfile in respected application root folder.

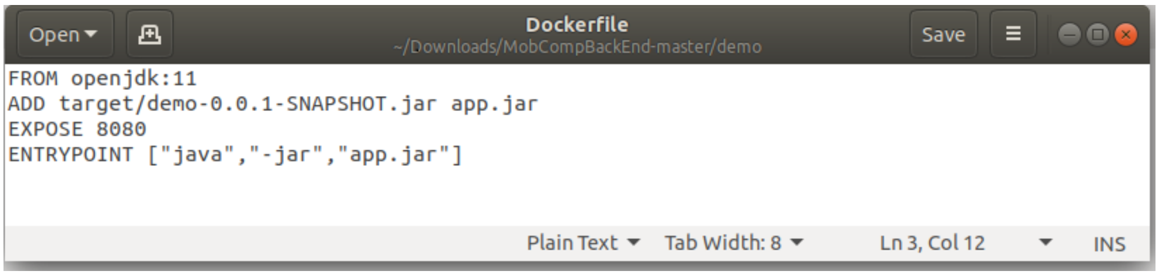
  >Angular Application DockerFile



```
# stage 1
FROM node as builder

WORKDIR /MobCompFrontEnd-master
COPY . .
RUN npm install @angular/cli -g
RUN npm i
# A head-of-time compilation
RUN ng build --prod

# stage 2
FROM nginx:1.17.1-alpine
RUN rm -rf /usr/share/nginx/html/*
COPY --from=builder MobCompFrontEnd-master/dist/FrontEnd-app /usr/share/nginx/html/
CMD ["nginx", "-g", "daemon off;"]
```

Figure 36 *DockerFile Angular*

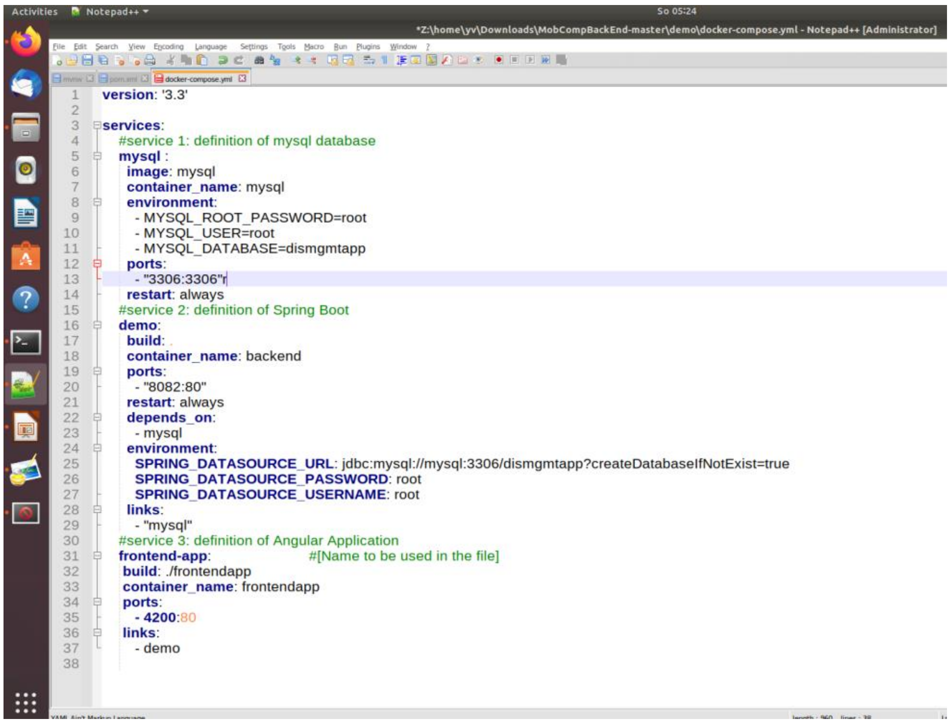
>SpringBoot Application DockerFile



```
FROM openjdk:11
ADD target/demo-0.0.1-SNAPSHOT.jar app.jar
EXPOSE 8080
ENTRYPOINT ["java","-jar","app.jar"]
```

Figure 37 *DockerFile SpringBoot*

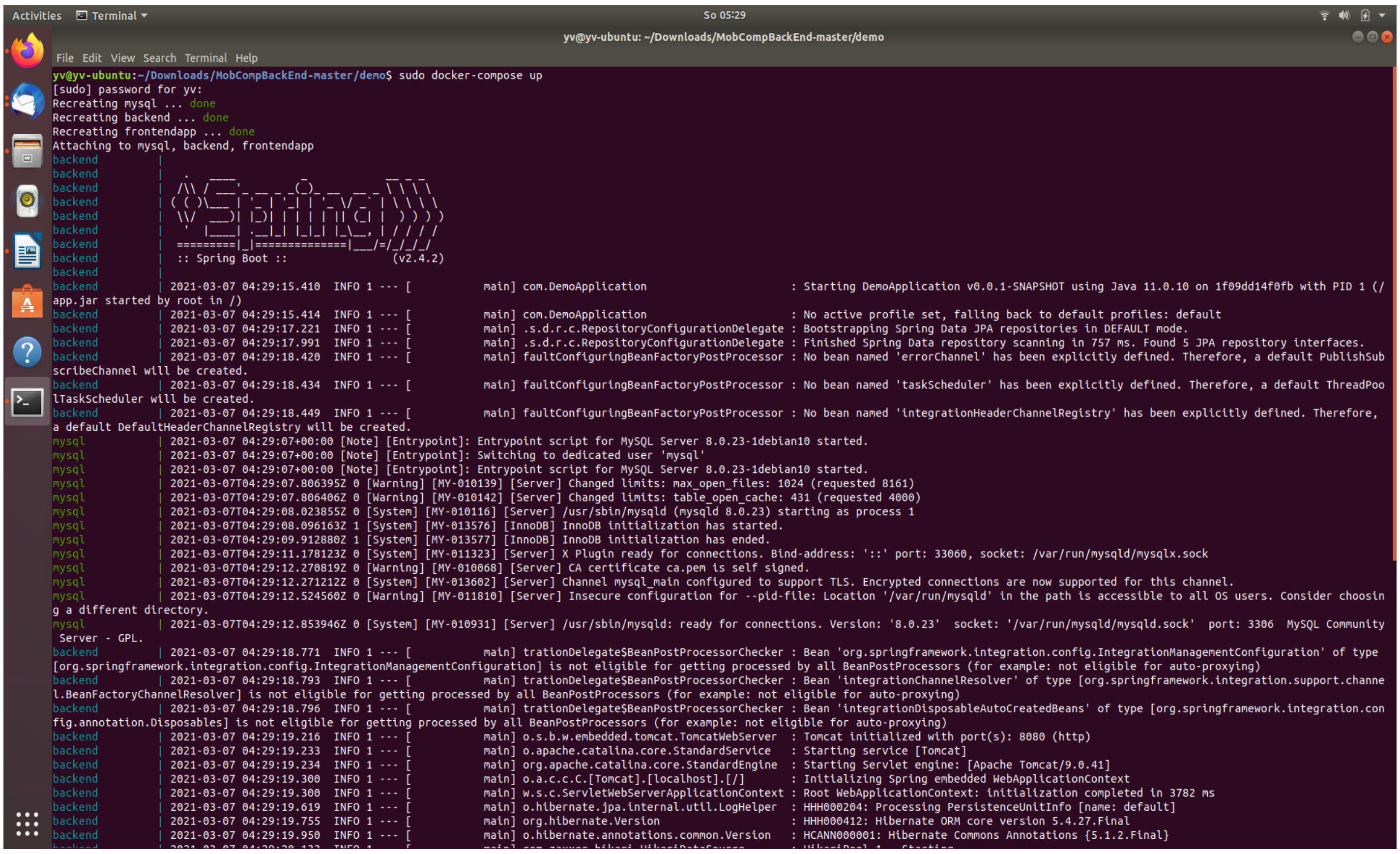
>Docker Compose File



```yaml
version: '3.3'

services:
  #service 1: definition of mysql database
  mysql :
   image: mysql
   container_name: mysql
   environment:
    - MYSQL_ROOT_PASSWORD=root
    - MYSQL_USER=root
    - MYSQL_DATABASE=dismgmtapp
   ports:
    - "3306:3306"
   restart: always
  #service 2: definition of Spring Boot
  demo:
   build: .
   container_name: backend
   ports:
    - "8082:80"
   restart: always
   depends_on:
    - mysql
   environment:
    SPRING_DATASOURCE_URL: jdbc:mysql://mysql:3306/dismgmtapp?createDatabaseIfNotExist=true
    SPRING_DATASOURCE_PASSWORD: root
    SPRING_DATASOURCE_USERNAME: root
   links:
    - "mysql"
  #service 3: definition of Angular Application
  frontend-app:            #[Name to be used in the file]
   build: ./frontendapp
   container_name: frontendapp
   ports:
    - 4200:80
   links:
    - demo
```

Figure 38 *Docker Compose File*

- Build Docker compose in root directory from terminal/command Line using command "docker compose build-up"

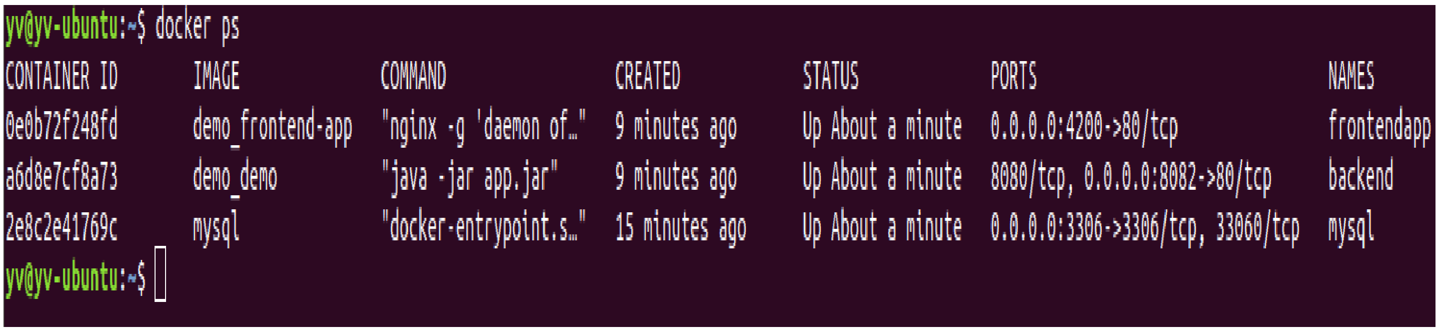


```
yv@yv-ubuntu:~/Downloads/MobCompBackEnd-master/demo$ sudo docker-compose up
[sudo] password for yv:
Recreating mysql ... done
Recreating backend ... done
Recreating frontendapp ... done
Attaching to mysql, backend, frontendapp
backend         |
backend         |   .   ____          _            __ _ _
backend         |  /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
backend         | ( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
backend         |  \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
backend         |   '  |____| .__|_| |_|_| |_\__, | / / / /
backend         |  =========|_|==============|___/=/_/_/_/
backend         |  :: Spring Boot ::                (v2.4.2)
backend         |
backend         | 2021-03-07 04:29:15.410  INFO 1 --- [           main] com.DemoApplication                      : Starting DemoApplication v0.0.1-SNAPSHOT using Java 11.0.10 on 1f09dd14f0fb with PID 1 (/
app.jar started by root in /)
backend         | 2021-03-07 04:29:15.414  INFO 1 --- [           main] com.DemoApplication                      : No active profile set, falling back to default profiles: default
backend         | 2021-03-07 04:29:17.221  INFO 1 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
backend         | 2021-03-07 04:29:17.991  INFO 1 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 757 ms. Found 5 JPA repository interfaces.
backend         | 2021-03-07 04:29:18.420  INFO 1 --- [           main] faultConfiguringBeanFactoryPostProcessor : No bean named 'errorChannel' has been explicitly defined. Therefore, a default PublishSub
scribeChannel will be created.
backend         | 2021-03-07 04:29:18.434  INFO 1 --- [           main] faultConfiguringBeanFactoryPostProcessor : No bean named 'taskScheduler' has been explicitly defined. Therefore, a default ThreadPoo
lTaskScheduler will be created.
backend         | 2021-03-07 04:29:18.449  INFO 1 --- [           main] faultConfiguringBeanFactoryPostProcessor : No bean named 'integrationHeaderChannelRegistry' has been explicitly defined. Therefore,
a default DefaultHeaderChannelRegistry will be created.
mysql           | 2021-03-07 04:29:07+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.23-1debian10 started.
mysql           | 2021-03-07 04:29:07+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
mysql           | 2021-03-07 04:29:07+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.23-1debian10 started.
mysql           | 2021-03-07T04:29:07.806395Z 0 [Warning] [MY-010139] [Server] Changed limits: max_open_files: 1024 (requested 8161)
mysql           | 2021-03-07T04:29:07.806406Z 0 [Warning] [MY-010142] [Server] Changed limits: table_open_cache: 431 (requested 4000)
mysql           | 2021-03-07T04:29:08.023855Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.0.23) starting as process 1
mysql           | 2021-03-07T04:29:08.096163Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
mysql           | 2021-03-07T04:29:09.912880Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
mysql           | 2021-03-07T04:29:11.178123Z 0 [System] [MY-011323] [Server] X Plugin ready for connections. Bind-address: '::' port: 33060, socket: /var/run/mysqld/mysqlx.sock
mysql           | 2021-03-07T04:29:12.270819Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self signed.
mysql           | 2021-03-07T04:29:12.271212Z 0 [System] [MY-013602] [Server] Channel mysql_main configured to support TLS. Encrypted connections are now supported for this channel.
mysql           | 2021-03-07T04:29:12.524560Z 0 [Warning] [MY-011810] [Server] Insecure configuration for --pid-file: Location '/var/run/mysqld' in the path is accessible to all OS users. Consider choosin
g a different directory.
mysql           | 2021-03-07T04:29:12.853946Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for connections. Version: '8.0.23'  socket: '/var/run/mysqld/mysqld.sock'  port: 3306  MySQL Community
 Server - GPL.
backend         | 2021-03-07 04:29:18.771  INFO 1 --- [           main] trationDelegate$BeanPostProcessorChecker : Bean 'org.springframework.integration.config.IntegrationManagementConfiguration' of type
[org.springframework.integration.config.IntegrationManagementConfiguration] is not eligible for getting processed by all BeanPostProcessors (for example: not eligible for auto-proxying)
backend         | 2021-03-07 04:29:18.793  INFO 1 --- [           main] trationDelegate$BeanPostProcessorChecker : Bean 'integrationChannelResolver' of type [org.springframework.integration.support.channe
l.BeanFactoryChannelResolver] is not eligible for getting processed by all BeanPostProcessors (for example: not eligible for auto-proxying)
backend         | 2021-03-07 04:29:18.796  INFO 1 --- [           main] trationDelegate$BeanPostProcessorChecker : Bean 'integrationDisposableAutoCreatedBeans' of type [org.springframework.integration.con
fig.annotation.Disposables] is not eligible for getting processed by all BeanPostProcessors (for example: not eligible for auto-proxying)
backend         | 2021-03-07 04:29:19.216  INFO 1 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port(s): 8080 (http)
backend         | 2021-03-07 04:29:19.233  INFO 1 --- [           main] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
backend         | 2021-03-07 04:29:19.234  INFO 1 --- [           main] org.apache.catalina.core.StandardEngine  : Starting Servlet engine: [Apache Tomcat/9.0.41]
backend         | 2021-03-07 04:29:19.300  INFO 1 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
backend         | 2021-03-07 04:29:19.300  INFO 1 --- [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 3782 ms
backend         | 2021-03-07 04:29:19.619  INFO 1 --- [           main] o.hibernate.jpa.internal.util.LogHelper  : HHH000204: Processing PersistenceUnitInfo [name: default]
backend         | 2021-03-07 04:29:19.755  INFO 1 --- [           main] org.hibernate.Version                    : HHH000412: Hibernate ORM core version 5.4.27.Final
backend         | 2021-03-07 04:29:19.950  INFO 1 --- [           main] o.hibernate.annotations.common.Version   : HCANN000001: Hibernate Commons Annotations {5.1.2.Final}
```

Figure 39 *Docker Compose Running Console*

- Access Client application [angular-FrontEnd] from browser “localhost:4200”

2) Checking Running Docker Containers:



```
yv@yv-ubuntu:~$ docker ps
CONTAINER ID   IMAGE              COMMAND                  CREATED          STATUS            PORTS                                NAMES
0e0b72f248fd   demo_frontend-app  "nginx -g 'daemon of…"   9 minutes ago    Up About a minute 0.0.0.0:4200->80/tcp                 frontendapp
a6d8e7cf8a73   demo_demo          "java -jar app.jar"      9 minutes ago    Up About a minute 8080/tcp, 0.0.0.0:8082->80/tcp       backend
2e8c2e41769c   mysql              "docker-entrypoint.s…"   15 minutes ago   Up About a minute 0.0.0.0:3306->3306/tcp, 33060/tcp    mysql
yv@yv-ubuntu:~$
```

Figure 40 *Checking running docker containers*

3) Possible Difficulties that can arise:

- Cross Origin Policy Issue:
  If CORS policy problem occurs with the
  Browser then try to disable –web-security
- Dependencies not installed:
  Install dependencies if not using docker containers.

4) Tutorial:

a) New User/Employees: Sign UP->email verification using token->Login

b) Victim: GenerateRequest{Givedetails}->Assigning_ResponderTeam->Communication->Complete_Request

c) RescueTeam: Employees are assigned to rescue team. Active Request dialog, message dialog, chat board dialog are on dash board

d) Admin: CRUD-> {Employee,Department,RescueTeam,Message}

H. Expansion Scope:

- Shortest path algorithms can be used for location management of IOT tags,
- Video-audio communication between victim and RescueTeam.

References

[1] A. P. T. U. T. A. L. B. G. Gregor Frick, "Possible Challenges and Appropriate Measures for a Resilient WMN-Based Disaster," 2020 World Conference on Computing and Communication Technologies.

[2] A. P. T. B. S. U. T. A. L. a. B. G. Frick, "“Requirements for a Distributed NFV Orchestration in a WMN-Based Disaster Network," 2019.

[3] “. F. V. (. ETSI and A. Framwork, "ETSI GS NFV 002 V1.2.1, 2014.".

[4] A. J. J. P. R. M. S. a. P. S. E. K. Çetinkaya, "“Resilience and survivability in communication networks: Strategies, Comput. Networks," vol. 54, no.pp. 1245–1265, Jun. 2010, doi: 10.1016/J.COMNET.2010.03.005..

[5] A. P. T. a. U. T. A.Lehmann, "Optimization of Wireless disaster network through network virtualiztion," Proceedings of the Eleventh International Network Conference, Frankfurt/Germany, 2016.

[6] B. D. S. a. J. A. S. C. B. Nelson, "The evolution of hastly formed networks for disaster response," IEEE Global Humanitarian Technology Conference , Seattle, Nov 2011.

[7] J.-S. Huang and Y.-N. Lien, "challenges of emergency communication for disaster operations," IEEE International Conference on Communication Systems (ICCS),, Singapore, 2012.

[8] H. X. N. Q.-T. V. a. P. S. K. Ali, "Disaster management communication networks," IEEE International Conference on Pervasive Computing and Communication Workshop, St. Louis, Mar 2015.

[9] X. W. I. F. Akyildiz, "“A survey on wireless mesh networks,” IEEE," pp. vol. 43, no. 9, pp. 23-30, Sept. 2005.

[10] "SpringBoot-Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Spring_Framework.