The Diagram of the project 'Applicant tracking system'

WebsocketMessageConfig (class)
Springframework
.messaging
.web.socket
.config.annotation
.StompEndpointRegistry
.WebSocketMessageBrokerConfigurer

GovernmentFinancialSupportApplication

The 'entry class' of the project. Contains the 'main' method.

SecurityConfig (class)
Springframework
.security
.HttpSecurity
.BCryptPasswordEncoder
JdbcUserDetailsManager
.SecurityFilterChain

WebsocketEnentListner (class)

ApplicantRepository (Spring data JPA)

Contains the methods:
existsByUsername,
findByUsername,
deleteByUsername,
findByAge,
findByOccupation,
findByDeletionIsTrue,
findEmailByUsername

SaveApplicant (class)
Contains the method:

saveApplicant
saves the collected from the applicant
and set by default data into the 'applicant'
table, saves the username and 'ROLE_APPLICANT'
into the 'authority' table

AuthorityRepository (Spring data JPA)
Contains the method:
deleteByUsername

MessageType (Enum class)
Spring Boot Lombok

Contains the parameters:
CHAT,
CONNECT,
DISCONNECT

ChatMessage (Entity class)
Spring Boot Lombok

Contains the parameters:
MessageType type,
message,
sender,
time

Applicants (Entity class)
Spring Boot Lombok

Contains the parameters:

name/ surname/ age
income/ email/ occupation
comments/ status/ deletion
password/ username (as id)

ApplicantService (class)

Contains the methods:

saveApplicant,
findAllApplicants,
findApplicantByUsername,
findApplicantByAge
findApplicantByOccupation,
findApplicantsByDeletion,
findEmailByUsername,
deleteApplicantByUsername,
getTotalNumberOfApplicants

EmailService (class)

Contains the method
sendEmail
which uses the methods of
JavaMailSender interface and
SimpleMailMessage class of
Spring framework

Authorities (Entity class)
Spring Boot Lombok

Contains the parameters:
authority/ username (as id)

ChatController (class)

Contains @GetMapping:
chat,

@MessageMapping:
chat.send,
chat.newUser

AccountController (class)

Contains @GetMapping:
account,
getAccountDetails (used by the Applicant).

getCurrentLoggedInUsername method is
to get the currently authenticated user

RegistrationController (class)

Contains the @GetMapping:
form

@PostMapping:
submitData

checks if exists by username,
hashes password using Bcrypt,
calls the 'saveApplicant' method of the SaveApplicant class,
if the applicant is over the 67 years old and not 'retired'

AdminController (class)

Contains the @GetMapping:
admin,
about_us,
manageApplicants

The business logic:
delete Applicant,
find Applicant by AGE,
find Applicant by Occupation,
find total number of Applicants,
export all Applicants into the CSV file
update the Status of the application

ApplicantController (class)

Contains the @GetMapping:
home,
login,
registration,
applicant,
news

The business logic:
Applicant's deletion request,
update of the Applicant's data

CSVExporter (class)

Contains the method:
exportApplicantsToCSV
which helps AdminController in saving
applicants.CSV file