DOCUMENTATION - JOB HUNT

By TeamZero

Abrudean Sergiu (Team Leader & Scrum Master)

Developers:

Abrudan Andrei

Acatrinei Andreea Laura

Andrasi Ladisalu

Baican Vlad

Balea Dorin

Banciu Raul

Barcau Emanuel Teodor

Bendic Radu

Bilc Irina

Biro Anna

Boier Daniel

**Short presentation of the project**

Dorim sa venim in sprijinul persoanelor care au nevoie de ceva specific, de scurta durata, de cele mai multe ori realizabil de catre studenti, dar si de alte persoane.

Persoanele care ofera serviciile au sansa sa faca un ban in plus, fara sa implice gasirea unui full time job. Example: am nevoie sa-mi plimbe cineva cainele, de 3 ori pe saptamana, cate 1 ora, am nevoie de 1 persoana care sa imi ia copilul de la scoala, si sa faca 2 ore de engleza cu el, pana ajung eu acasa, etc.

Posibile functionalitati:

Generale: - newsletter - dashboard (statistici, informatii utile rapid de accesat, latest info, etc)

* Utilizatorul “”client”“:

1. isi creaza un cont cu detaliile lui - poate adauga unul sau mai multe job-uri si astepta aplicanti
2. poata cauta printre profilele disponibile si alege o persoana anume careia sa-i trimita cererea (bazata pe recomandari / review-uri)
3. poata adauga un review persoanei cu care a colaborat - poate recomanda o persoana unui alt “”client”" pe care il cunoaste

* Utilizatorul “”provider”“:

1. isi creaza cont cu detaliile lui
2. isi adauga lista de abilitati - vede cererile in aplicatie si poate aplica
3. poate oferi un review “”clientului”" cu care a colaborat .

* Nice to have features:

- social networks integration Extra details: - in functie de tehnologiile alese de echipa, putem avea atat aplicatie web cat si mobile.

**Project Requirements**

User types:

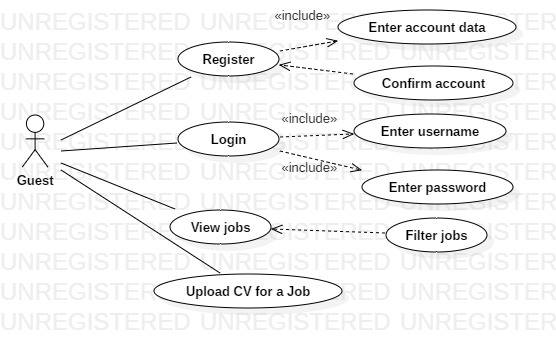
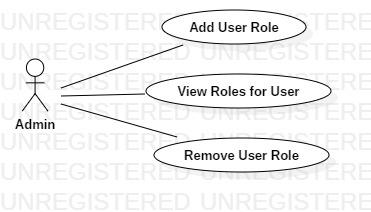
* Admin (can do everything a regular user can)
* Regular user

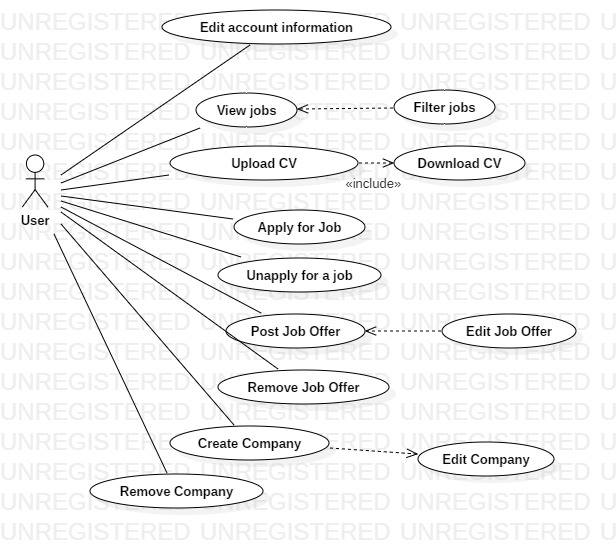
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Type of user** | **Action** | **Call** | **URL** | **Parameters** |
| Guest | Create a new account, providing the necessary data. Validates the email address and the password. Sends a confirmation email. | POST | /register | Email  Password  FirstName  LastName  City  Address  Zipcode  Phone |
| Guest | Login into an  account. Checks if username and password are valid. | GET | /login | Email  Password |
| Guest | Get all users. Optionally, it can return a user by id. | GET | /user/{userid} | Optional parameter: userid, else return all accounts |
| Guest/User | Get job offers | GET | /job-offers | limit  start |
| Guest/User | Get all Job types | GET | /job-types | No parameters |
| Guest/User | Filter Jobs by different criteria | GET | /job-offers/filter | Possible parameters:  End Date  Job Type  Start Date  Starting Salary |
| Guest/User | Get Company by userid | GET | /company-of-user | userid |
| Guest/User | Get company by id | GET | /company/{companyid} | companyid |
| User | Add CV | PUT | /add-cv/{userid} | CV |
| User | Download CV | GET | /download-cv/{userid} | userid |
| User | Register Company | POST | /register-company/{userid} | Company:  - Name  - Description  - Email  - PhoneNumber  - Job Offers  - Posted Job Offers |
| User | Update information, same parameters as register can be edited and updated. | PUT | /update | Email  Password  FirstName  LastName  City  Address  Zipcode  Phone |
| Admin/User | Add a job offer. Admins can add job offers, just as users can, by providing the necessary data. | POST | /job-offer-service/create | Title  Description  PhoneNumber  Email  Address  Type  City |
| Admin/User | Remove job offer by job id and userid | DELETE | /job-offer-service/delete | jobId  userid |
| Admin/User | Edit job offer. Besides the id, which cannot be edited, all parameters can be updated | PUT | /job-offer-service/update | Title  Description  PhoneNumber  Email  Address  Type  City |
| Admin | Delete user by id. | DELETE | /delete/{userid}/{deleteUserId} | userId  deleteUserId |
| Admin | Get the role of a user by it's username | GET | /admin-service/role/{username} | username |
| Admin | Add User Role | POST | /admin-service/role/add/{userid}/{role} | role  userid |
| Admin | Remove Role | DELETE | /admin-service/role/remove/{userid}/{role} | role  userid |

**Technologies**

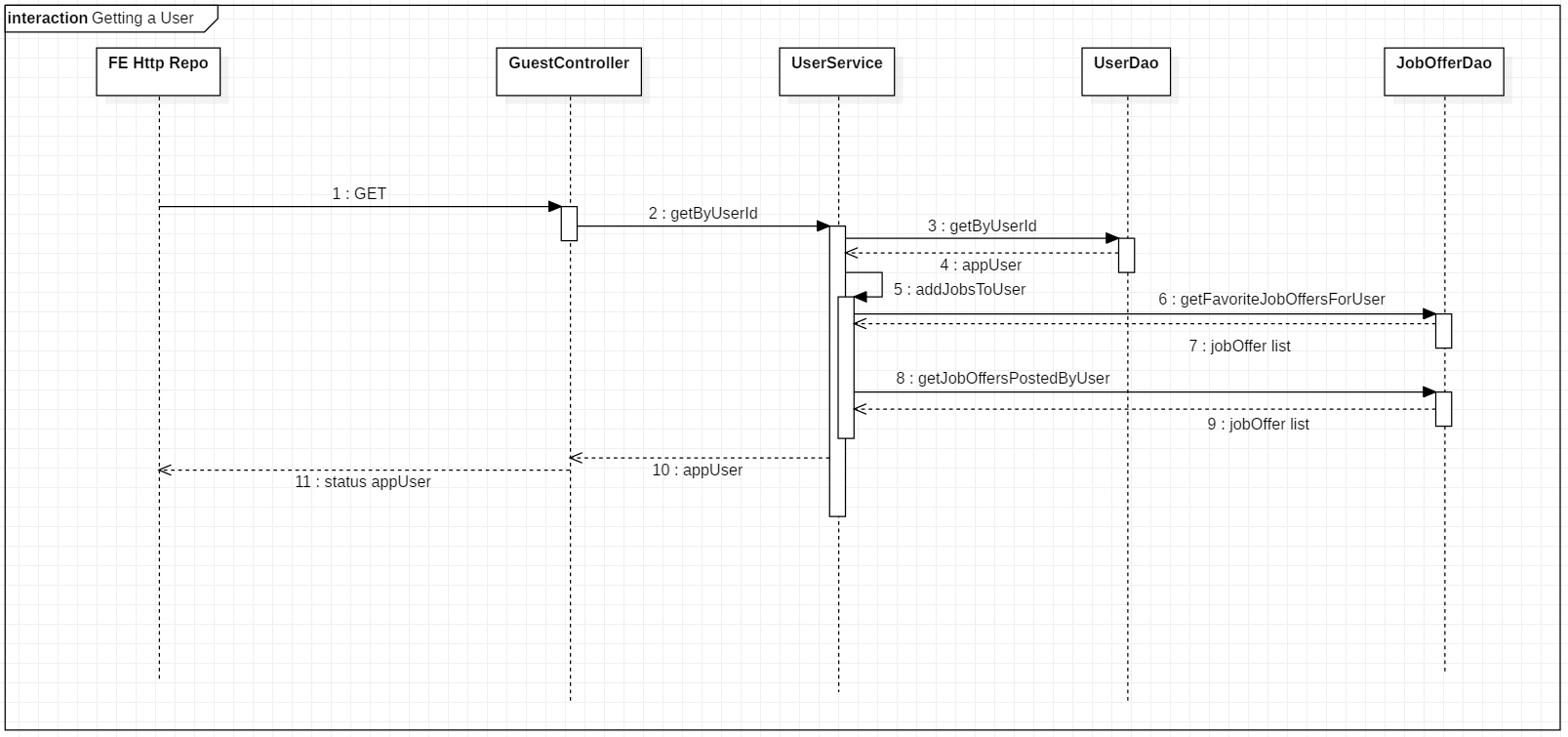
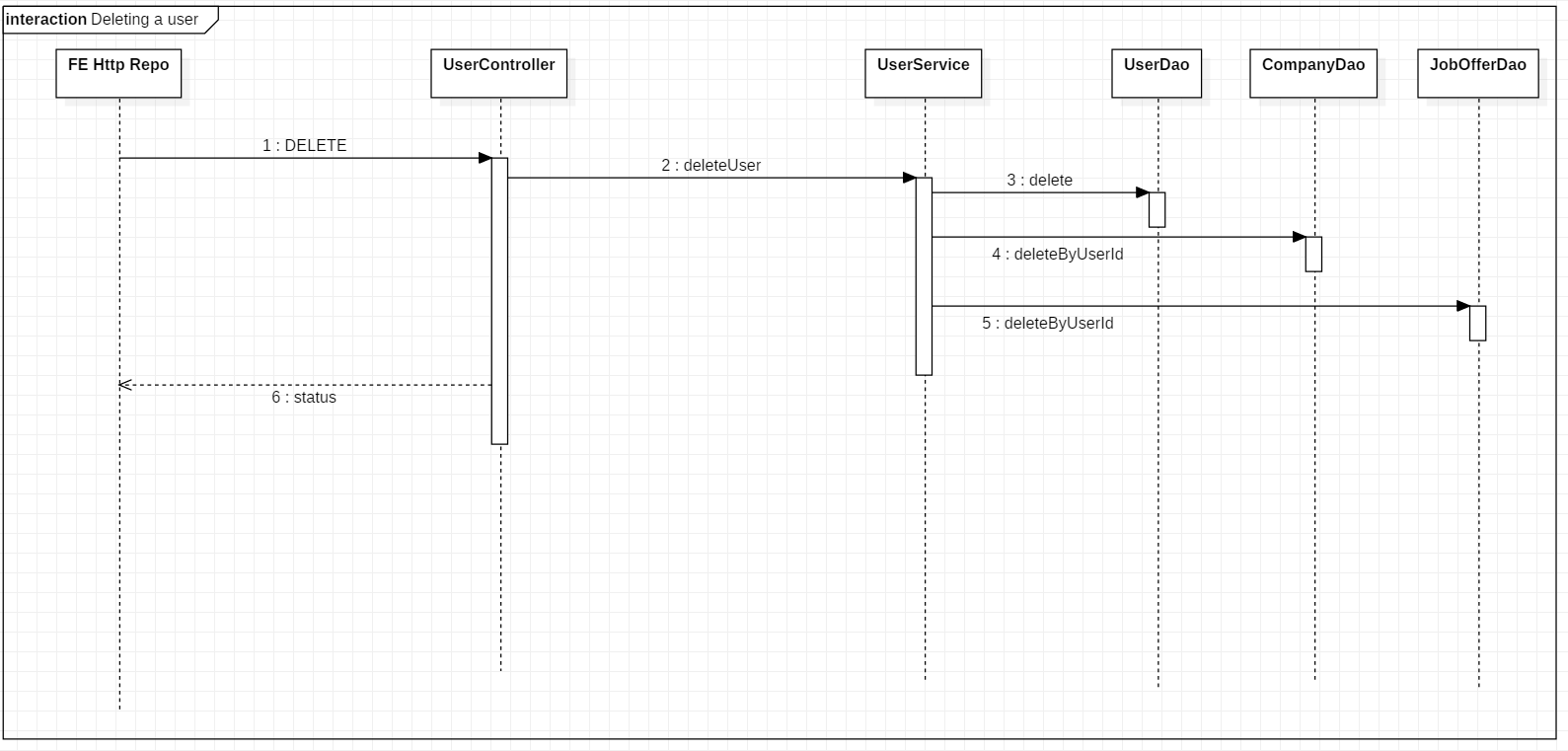
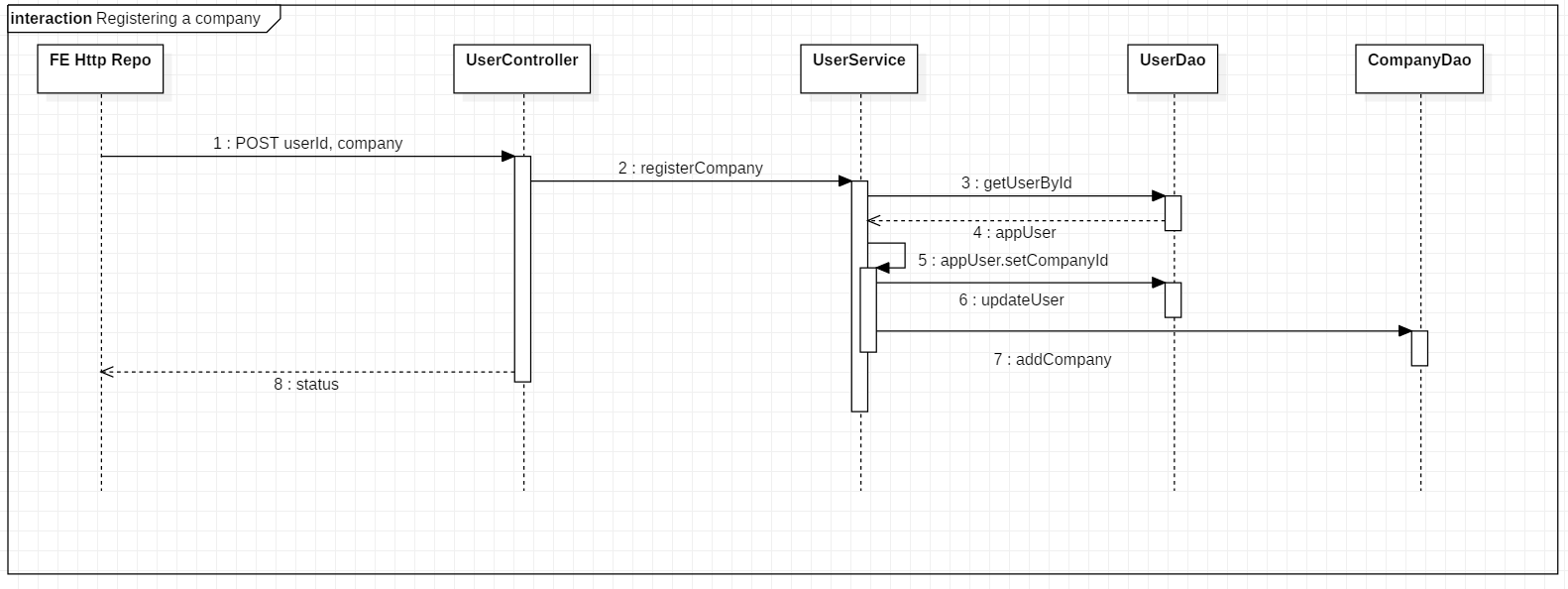
* The project was written in Java.
* For the creation of the database, we used PostgreSQL in order to easily be able to connect the database to our project.
* spring, postgres, springboot, react-front, spring secutiry
* On the back-end part of the project, the part of the project that manages the database operations and the link to the database we used Spring and Springboot.
* The security is provided by Spring Security.
* On the front-end part of the project we used React.

**Use Case Diagrams**

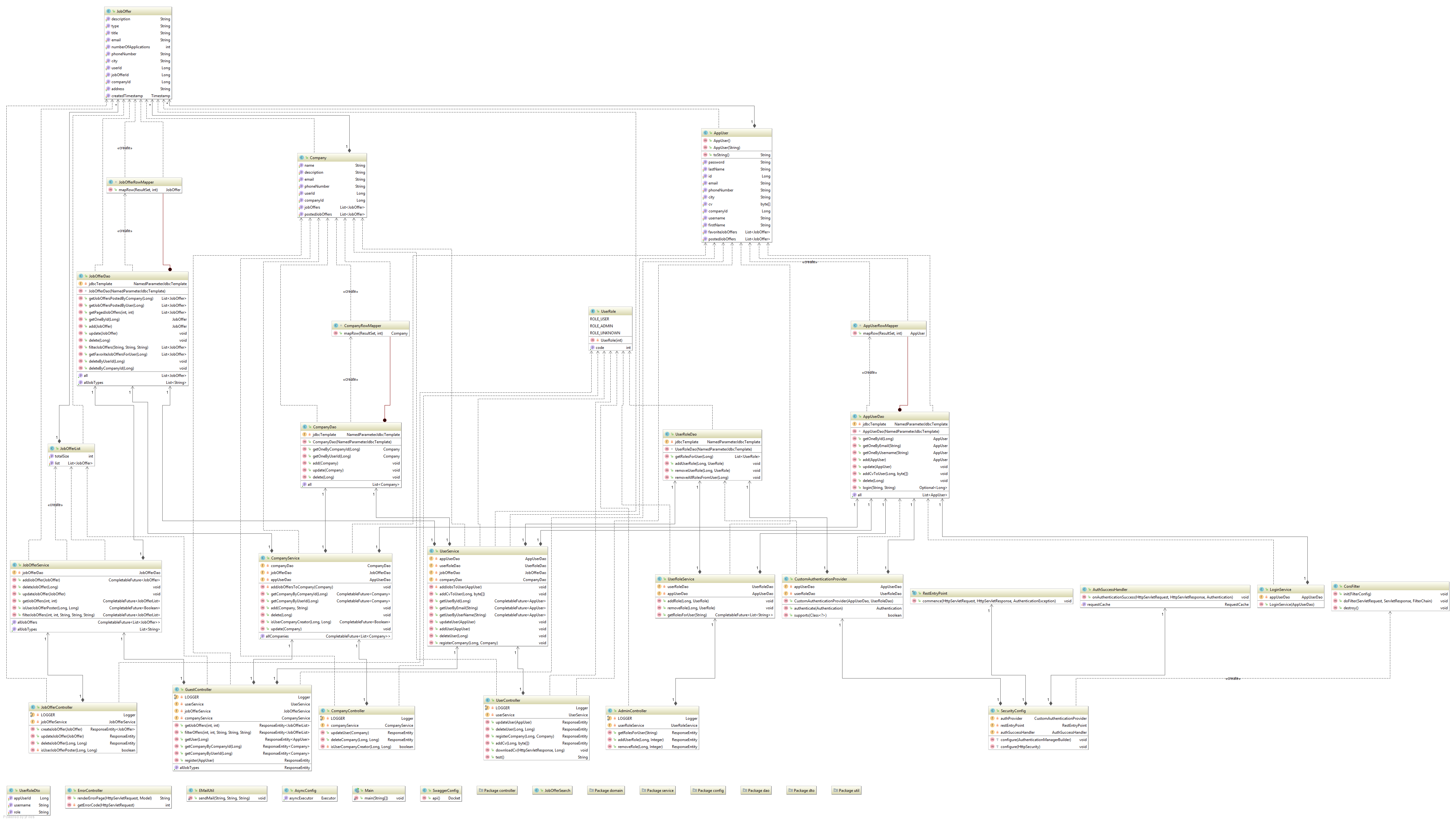
****

****

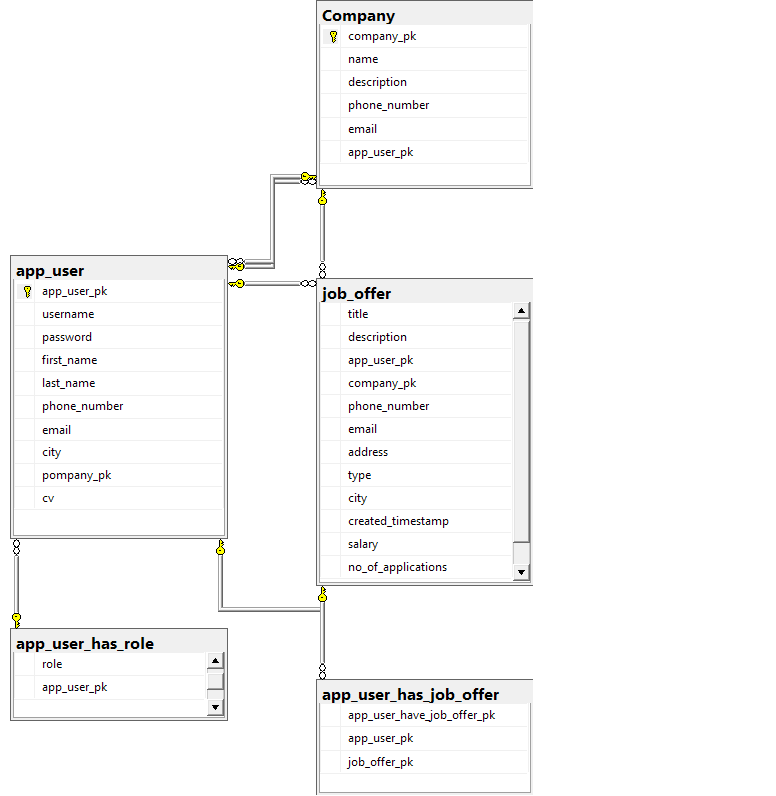
**Sequence Diagram**



**Class Diagram** (more in-detail picture appears on the presentation)



**Database Diagram**



**Future releases:**

* Better UX
* Mobile support
* In-app chat
* Include rating system
* Job recommendation system
* Third party log-in