Technical Report - Project specifications

Work and travel app

Course:	IES - Introdução à Engenharia de Software
Date:	Aveiro, 26/1/2022
Students:	105648: Ana Atanasova 105625: Tanja Milososka 105635: Aleksandra Zdravkova 106590: Aurimas Arlauskas

Table of Contents

1 Introduction

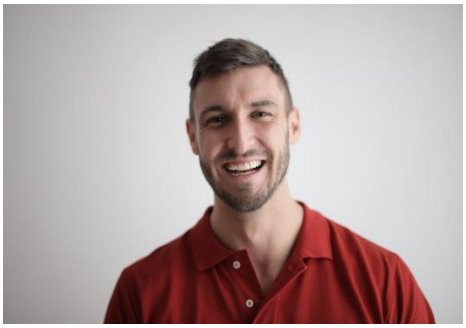
As students, we often want in our free time during summer, to go to other countries and to travel or work. But there is also an option called Work-and-Travel when you can do both of the things at the same time. In order to help ourselves and other students, we had an idea of making web page that will help us speed up the applying process and getting the job sooner and in a most efficient way. That will be done by creating a profile, getting information about the jobs and messaging all the employers at once, instead of writing the mail to each one separately.

2 Product concept

Vision statement

The main concept of the application is, for a given data base with e-mails of people who are providing job and companies, to write one email that will have the basic information about us and to send the mail at once to all the people that are listed. Our application will allow you to log in and let you try out the application for free with restricted options. With trying out the application for free you could only send up to 30 emails at a time and see fewer company details and information. You can also sign up as a pro user, where all the options will be available to you. Then you can send up to 500 emails to a company and the user will see more information about the companies. So basically, if you decide to go to Work-And-Travel and use our application the process will be way easier for the appliers. This will be unique application that is different from the things that already exist. Our main goal is to make the life of the students way easier providing them with this type of service.

Personas



Regarding the personas that will use the application, first we have one student who's name is Stefan. He is 22 years old and he is studying Computer Engineering. He is from Macedonia. He was thinking about applying for job in USA during the summer, in order for him to get some money and go to masters in Portugal, but he was not sure how can he do that and he didn't want to spend many of his time just writing emails. One of his friends recommended our application to him.

Here is also Ivana. She is 48 years old. She lives in USA in Boston and she have her own restaurant. She likes working with young people, especially with people that come from other countries because in that way she can get in contact with different cultures and learn new things. She knows about Work-And-Travel and she always put information about her restaurant in order for students to get in contact with her. Usually during the summer she is offering job to 3 students in different working positions. Until now everyone is satisfied. She is glad that the application is helping young people to contact her even more.

Main scenarios

After the talk with his friend, Stefan decided to give it a try to the application. He made his own profile and decided to write an email. He first wrote description about himself and about what kind of job he is requiring. He clicked send and he checked his email to confirm if the mails are send. After he confirmed, he went to the application and decided to get the full version. Now, he can search in which state he want to go and what kind of job will be perfect for him. He saw the restaurant of Ivana and he liked all the rules and the things that are offered. The process of finding job will be limited to minimum. He thinks that this application is the best thing that happened to him.

After getting the information that this kind of application is existing, Ivana

decided to check if someone actually contacted her through it. She saw the email from Stefan and decided to replay to him. They were in contact for a while, and after they agreed that it will be perfect if they work together. In this case the application turned out good for both of sides. Everyone was satisfied.

User stories:

As a user, I would like to be able to sign in with my personal email and password
As a user, I would like to first check if the application is worth for using by sending emails to yourself and sign up as basic user
As a user, I would like to get a list of companies with the type of the job that they are providing and the state where they are located.
As a user, I would like to be able to do online payment for the full version.
As a user, I would like to be able to send all the emails at once.
As a company owner, I would like to get in contact with people who are searching for job, and offer them a job.
As a company owner, I would like to be able to register my company.
As an admin, I would like to be able to see all the changes

3 Architecture notebook

Key requirements and constrains

The core technologies used for the project:

- ReactJS
- Java Spring Boot
- MySQL database
- Python/JS script for data generation

All of the mentioned technologies can run both in Linux or Windows environment, so

the deployment process should be relatively simple. The system is intended to be dockerized, which would mean a more robust and reliable solution.

The system implementation will also include Scheduler. The Scheduler will be used for constant checkups on the Tasks that still need to be done. As we are depending on Gmail SMTP server, it has a limit of 1500 daily emails. If the paid user would intend to send more emails than that we need to make sure that the task continues successfully after the 24-hour period.

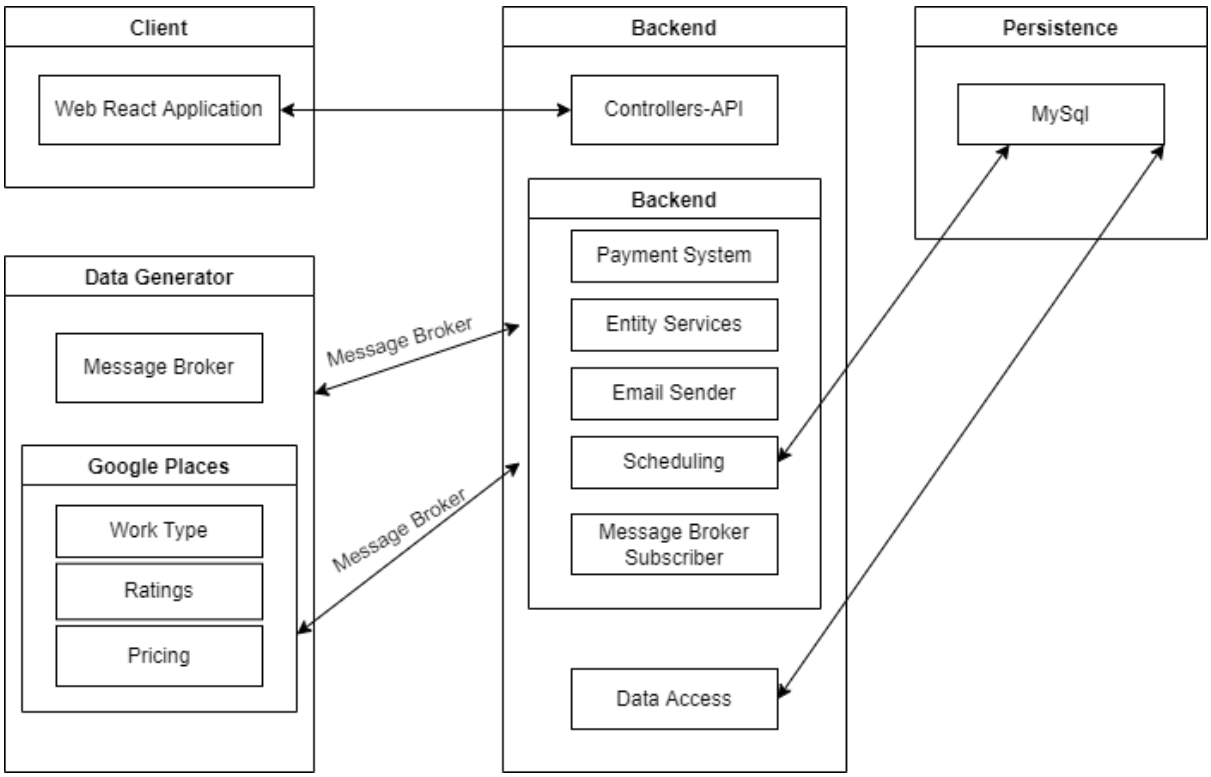
The main dependency of this system, that will rely on an outside source will be the company list that is used in our system. As of right now, we have achieved to gather over 200 static HTML files, that contain all of the relative information about the company. This list includes the Company's name, email address, website URL, state and current address. We intend to use the provided information to perform constant lookups on the information on the company to figure out their business work type (e.g. Barman, Waiter, Amusement). As this information is ever changing and we intend to provide the most relevant information as a service.

If we would consider this project as an actual business plan, we would contact the companies that have gathered these lists and ask for access to their databases, so that the information would be more reliable.

Architectural view

- ReactJS. It will be responsive (mobile and desktop friendly) frontend application which will access the endpoints of the Java Backend API. Most the website will be dynamic, fetching the data from the service. The static parts of the website will be the home page and policies page.
- Java Spring Boot Application will be our API, which will provide authorized controllers to be accessed by the ReactJS application. As we intend to have different roles in the system (public, user, admin) we must provide a correct authorization system. This will include JWT token authentication. The backend will also connect to the MySQL database which will hold tables for Users, Tasks and Companies. The business logic will also include the Emailer, which will be a static class with a sole intention of sending emails to the recipients.

- MySQL database will be hosted on the same environment as the backend is. Current intention is to use a docker image database with the intention of not redoing it in the end.
- Python/JS script will create a message queue subscription service for backend. This will allow the system to fetch data and separate two different entities of the system. As a message broker we are using RabbitMQ.
- Libraries and other dependencies that are we using are Stripe, Jira and GooglePlaces. We used stripes about our online payment, Jira about dividing the work between us and with googleplaces we connected the company location with google and from there we extracted the information that we needed.
- As a scheduler mentioned above we used the JobRunr library.

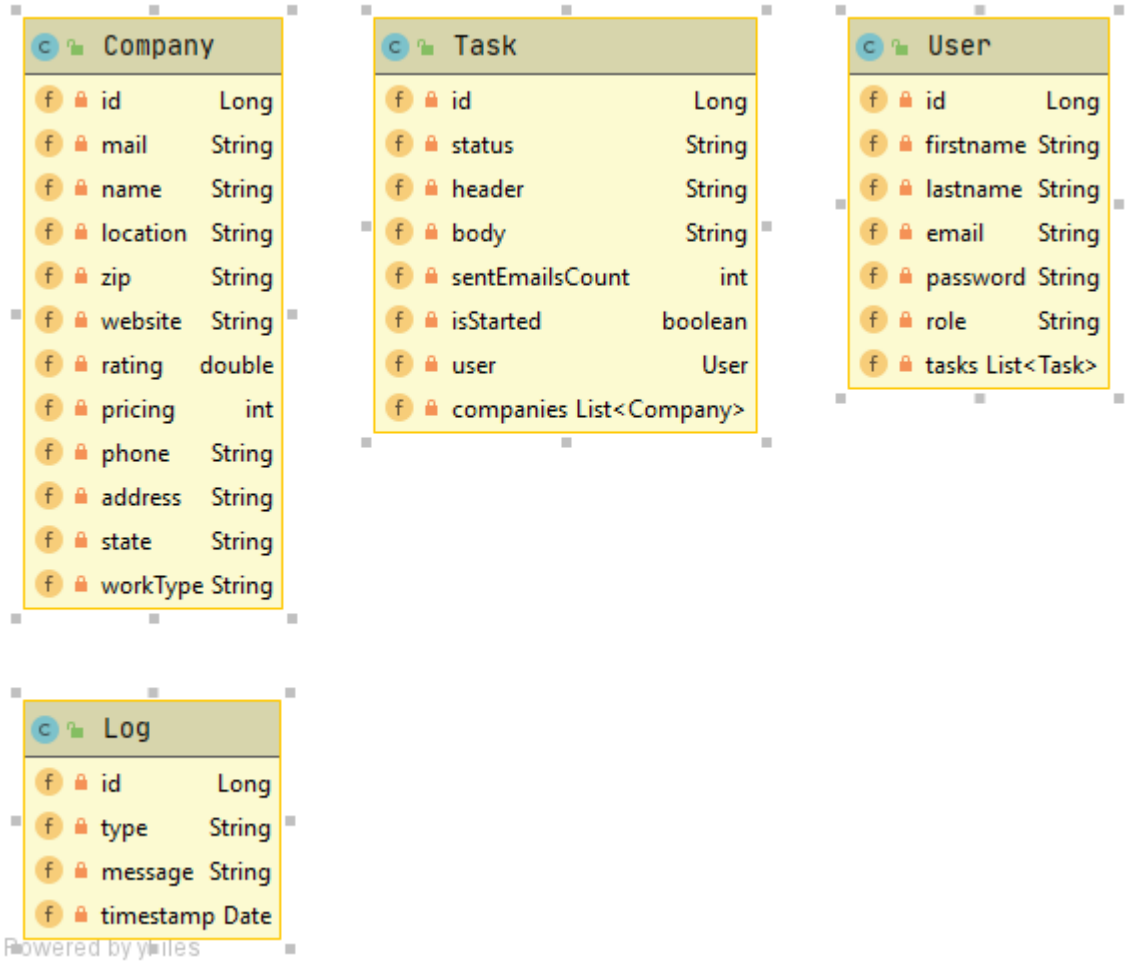


Module interactions

- ReactJS → Spring Boot API. Calling the endpoints to GET, POST, PUT, DELETE data;
- Data Generator → Spring Boot API. Subscriber relationship with the message broker. Serve the queued newly parsed data to the API so that it could be renewed in the database.
- Spring Boot API → MySQL database. Make SQL statements using ORM and JPA to get and put data.

Since the project will be deployed in our servers and the frontend will be accessible through a website all of the changes and updates can be rolled out through the server by rebuilding the project. This means that deployment is very dynamic and can be done momentarily. It's also possible to setup a release pipeline, so that the changes in the Main branch would be automatically tested and rebuilt and deployed.

4 Information perspective



5 References and resources

https://reactjs.org/

https://dev.mysql.com/doc/

https://www.ibm.com/cloud/learn/message-brokers

https://docs.aws.amazon.com/cloudcontrolapi/

https://spring.io/guides/tutorials/rest/