



Bilkent University

Department of Computer Engineering

CS 491 - Senior Design Project I

Project short-name: Recroute

High Level Design Report

Ege Şahin 21702300

Göktuğ Gürbüzürk 21702383

Aybars Altınışik 21601054

Ahmet Feyzi Halaç 21703026

Supervisor: Prof. Dr. Özgür Ulusoy

Jury Members: Dr. Shervin Arashloo and Dr. Hamdi Dibeklioglu

Innovation Expert: Ahmet Eren Başak

High Level Design Report

December 20, 2021

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

1. Introduction	3
1.1 Purpose of the system	3
1.2 Design goals	3
1.3 Definitions, acronyms, and abbreviations	4
1.4 Overview	4
2. Current software architecture	4
3. Proposed software architecture	4
3.1 Overview	4
3.2 Subsystem decomposition	5
3.3 Hardware/software mapping	6
3.3.1 Deployment Diagram	6
3.4 Persistent data management	6
3.5 Access control and security	7
3.6 Global software control	7
3.7 Boundary conditions	7
4. Subsystem Services	8
4.1 Presentation Layer	8
4.1.1 Web UI	8
4.1.1.1 FlowPages	9
4.1.1.2 ExamManagementPages	9
4.1.1.3 UserManagementPages	9
4.1.1.4 MainPages	9
4.1.1.5 ApplicantPages	9
4.1.1.6 LoginPages	9
4.2 Business Layer	10
4.2.1 Gateway Service	10
4.2.1.1 Authorizer	10
4.2.1.2 Controller	10
4.2.2 User Management Service	11
4.2.2.1 CompanyManagement	11
4.2.2.2 RoleManagement	11
4.2.2.3 UserManagement	11
4.2.2.4 Payments	11
4.2.3 Flow Management Service	12
4.2.3.1 FlowManagement	12
4.2.3.2 ApplicationManagement	12
4.2.3.3 InterviewManagement	12
4.2.4 Examination Service	13
4.2.4.1 ExamManagement	13
4.2.4.2 QuestionManagement	13
4.2.4.3 PoolManagement	13
4.2.5 Code Execution Service	14

4.2.6 Application Service	14
4.2.7 Authorization Service	14
5. Consideration of Various Factors in Engineering Design	15
6. Teamwork Details	16
6.1 Contributing and functioning effectively on the team	16
6.2 Helping creating a collaborative and inclusive environment	16
6.3 Taking lead role and sharing leadership on the team	16

1. Introduction

1.1 Purpose of the system

Normally, recruiters use different platforms at different stages in the online recruitment process. They find the candidates from a platform, test these candidates with the help of other platforms, and meet with those candidates from other online platforms. The fact that all these stages are carried out on different platforms and regulated manually creates a serious workload for recruiters. In order to reduce this problem and help companies, we aim to make all these stages easier and faster on a single platform and reduce the workload of recruiters.

Recroute is an online recruitment platform that facilitates the different stages in the recruitment process. The purpose of this platform is to ease the work of recruiters by managing all aspects of recruitment flow from a single platform. So, recruiters can use this system for three different stages of the recruitment flow and reduce their workload by not spending time between different applications.

1.2 Design goals

Efficiency:

For Recroute, efficiency is crucial because measuring technical proficiency is part of the system and it includes coding tests. These coding tests must be efficiently handled so that the system can warn the user about his/her mistakes with no or minor delays.

User-friendliness:

The main purpose of the system is to reduce the workload of recruiters. In order to do this, the system must also be user-friendly and easy to use.

Security:

The system must ensure that applicants' data will not be shared with any 3rd party. Companies must accept the privacy policy.

Reliability:

The system must be reliable so that if any problems occur during any stage of the recruitment process the information must be stored in the system correctly so no files go missing such as application forms of applicants.

Modifiability:

The recruitment flow may include other steps in the future which are not used today. The system should be easily modifiable when changes are needed in the future.

Scalability:

Since the system may include many companies at the same time, it must be scalable to handle a large number of users from various companies.

1.3 Definitions, acronyms, and abbreviations

- **Recruiter:** Company or a person who is responsible to hire employees.
- **Recruitment Flow:** The process of analyzing, testing, and interviewing the job applicants and then finding the appropriate candidates for the job.
- **Stage:** A step on the recruitment flow. Can be one of the following: Application, Test, or Interview.
- **Applicant:** Person who applies for a job advert.
- **Job advert:** Overall application process created by a company. It is composed of a recruitment flow and general information about the job.

1.4 Overview

This report will provide the high-level design of the Recroute. First, the proposed software architecture will be explained through subsystem decomposition, hardware, and software mapping, persistent data management, access control, and security, global software control, and boundary conditions. Afterwards, the report will explain the subsystem services and consideration of various factors in design. At last, the teamwork details section will explain the collaboration and communication processes of our team.

2. Current software architecture

In the current system there is no platform which combines the recruitment process. Companies use different platforms at different stages in this process. LinkedIn, kariyer.net and similar platforms are used for creating and publishing job adverts. Candidates apply to the job adverts from these platforms. After that, companies choose some of these candidates and continue the recruitment process by testing those applicants. In order to test these applicants companies are using either their own tests or they ask applicants to solve a test from another platform such as Hackerrank and Codility to test their technical proficiency. By examining the results of the test, they decide which candidates should go to the next stage. For the candidates who are in the last stage(interview stage), personal interview times are determined and the meetings are arranged via the Zoom or other online meeting platforms. The details of the meeting were sent to the candidates via email manually.

3. Proposed software architecture

3.1 Overview

Below is the explanation of the proposed software architecture of Recroute. First the subsystem decomposition is explained with a diagram. Then hardware software mapping is explained with a deployment diagram. After that persistent data management, access control and security, global software control and boundary conditions are explained.

3.2 Subsystem decomposition

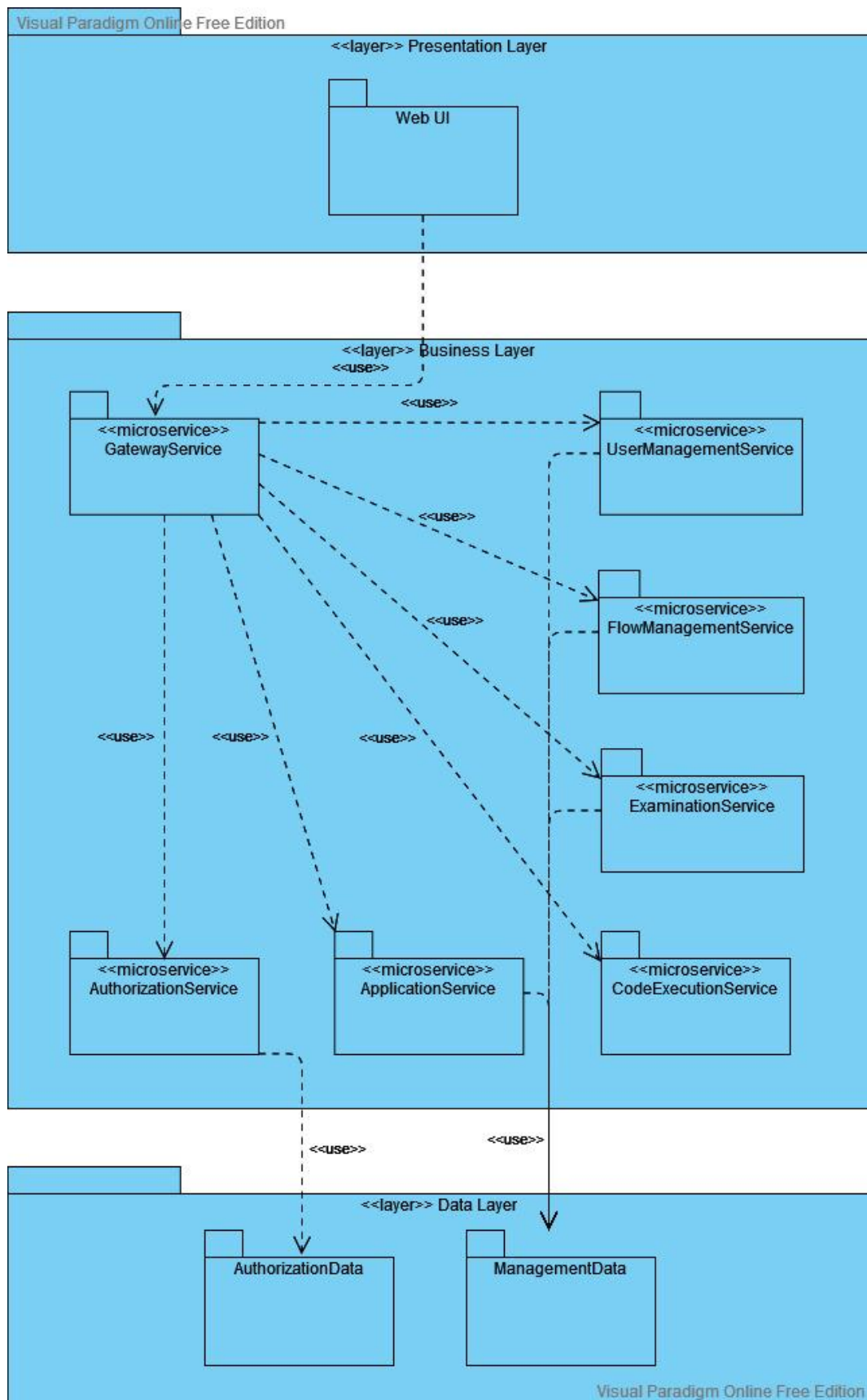


Figure 1. Recroute Subsystem Decomposition

3.3 Hardware/software mapping

3.3.1 Deployment Diagram

Visual Paradigm Online Free Edition

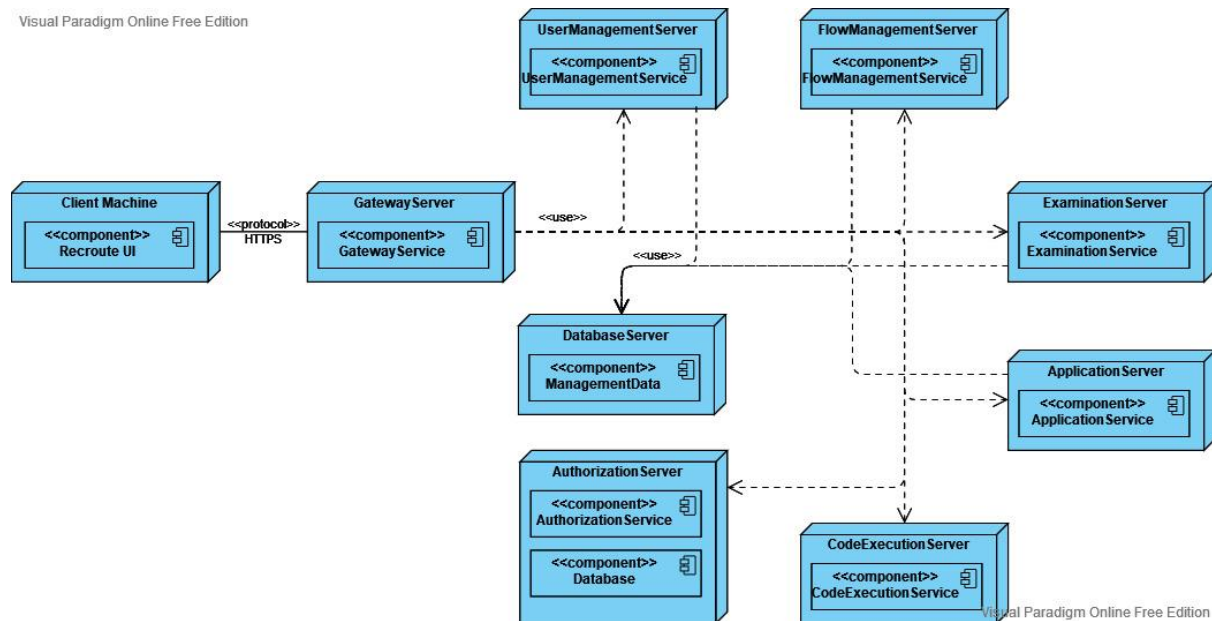


Figure 2. Deployment Diagram

Above is the deployment diagram of the system. Each microservice will be deployed on a different machine to meet the scalability quality attribute of the system. Scalability improves as the number of server instances increases. As the system is designed to use the microservices architecture pattern, each module is independent. However, UserManagement, FlowManagement, Examination, and Application services use a common database, ManagementData, as the data models that these services use will be relational.

3.4 Persistent data management

The application will have persistent data storage to store information about applicants, job adverts, and user credentials. More specifically, we will store resumes and other personal information of the applicant required during each job application. For each company registered to the system, we will store information about company details and job adverts of it. Since the recruitment flow consists of different stages, for each stage, we will also store the results of the applicants and the analyzes or comments made by the company at that stage. In addition to that details about job adverts will be stored to enable companies to reuse them in another job advert. Lastly, we will store the user credentials such as username, password, email address etc. To store all this information, we will use AWS S3 since it facilitates secured and low-latency data storage from the cloud. However, during the development of the project, we are planning to use Heroku instead of AWS S3 in order to decrease the cost because Heroku offers a free plan for hosting PostgreSQL databases.

3.5 Access control and security

Access control is significant in terms of security because we will store the personal information about each applicant and user credentials like passwords for each company. There will be three different actors in the system. The first one is the applicant who needs to give some personal information during the job application. This information will be seen only by the company that advertises the job. The second type of actor is company. Each company will be able to see the statistics about the test results and the sensitive information of applicants who applied to their job adverts. The companies will not have access to applicant information in other job adverts belonging to different companies and all the companies will be able to see their own credentials. Lastly, administrators will have no direct access to sensitive information like passwords. We will keep all these and more in accordance with the General Data Protection Regulation. In this respect, we will encrypt all the sensitive information before storing it to keep the data confidential even if there is access to the database.

3.6 Global software control

Global software control in Recroute follows the event-driven control mechanism. It is crucial that multiple recruiters can use the system simultaneously. This can be done through event handlers. Each recruiter can control different parts of different recruitment processes simultaneously. Our system will use asynchronous event handlers for this purpose. Whenever a new step is created in the recruitment flow, the corresponding event handler will receive the request and take the necessary actions.

As the applicants complete different steps of recruitment, they will continue to the next step with permissions of recruiters. If no applicant completes a step in the recruitment and no recruiter creates a new step, the system is stationary for the given time for each active step of a recruitment. If the time given is over before the applicant completes that level, the applicant is rejected automatically.

3.7 Boundary conditions

Recroute will have three boundary conditions. These are initialization, termination, and failure. Initialization is starting the application. Termination is closing the application. Failure is a failure in the application.

- **Initialization:**

The user must have an internet connection and access to a compatible web browser in order to use the program since the program is a web application. In addition to that, the user must have a recruiter account registered for a company. Each recruiter of a company will use different recruiters accounts registered for the company and enter the system and see the recruitment stages of different applicants.

- **Termination:**

The user can log out from the system by clicking the logout button. When the recruiter closes the web browser without specifically logging out, they will be automatically logged in when they open the webpage again until the tokens are

expired. However during the time that the webpage is closed, that recruiter account can not perform any operations.

- **Failure:**

If somehow the recruiter lost connection to the system, they will not have access to the server thus they cannot view anything on the platform. The data on the database however will not be affected from this disconnection.

If any of the API's that the program uses become out-dated, some of the features of the system may become unavailable.

Database errors may result in incorrect data transactions.

4. Subsystem Services

Architecture style of the application is Three Tier Client & Server with Microservices architecture pattern. First layer is the presentation layer where the web based user interface is presented. Second layer is the business layer where backend services are decomposed to obtain multiple microservices. Last layer is the data layer which contains databases related to the services.

4.1 Presentation Layer

4.1.1 Web UI

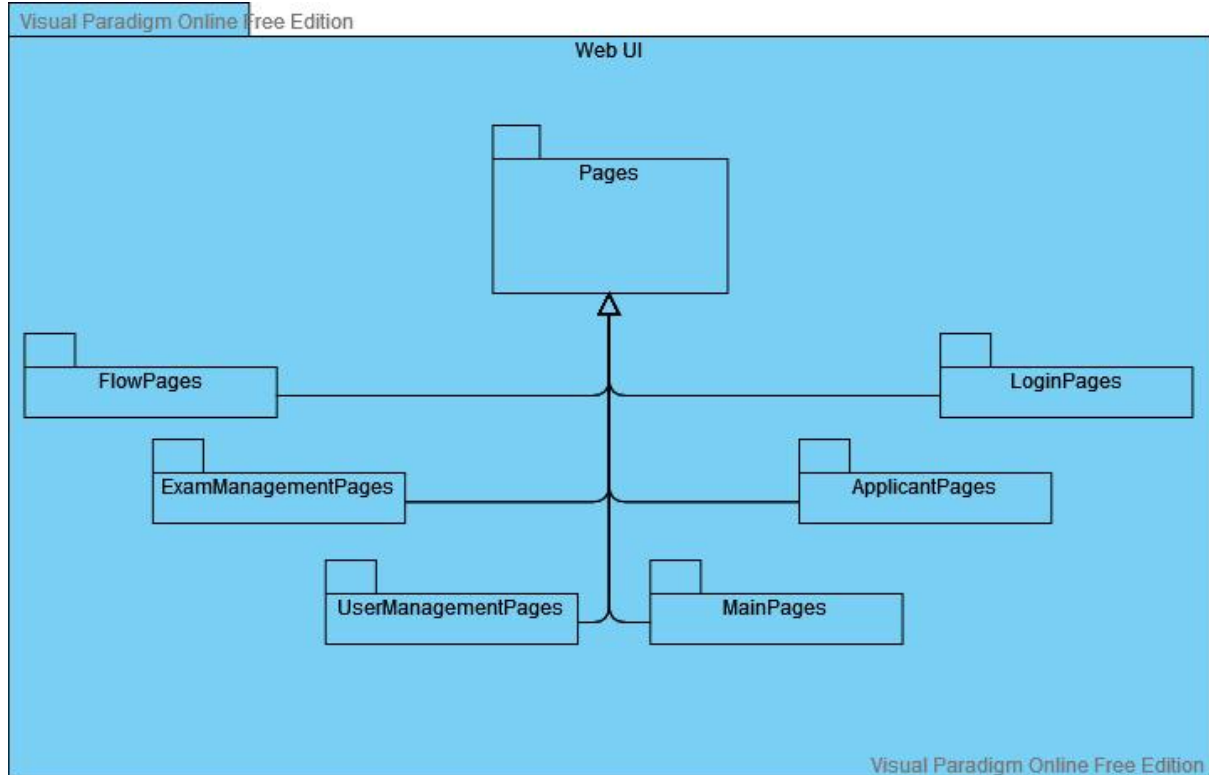


Figure 3. Web UI

In the Web UI package, modules are decomposed into pages modules. Pages modules contain pages related to that domain.

4.1.1.1 FlowPages

This module is composed of pages and corresponding UI components for interacting with any flow of the user. These interactions can be seeing the flow as a complete diagram, adding a new process to flow, removing a process from flow, or observing the current state of the flow (applicant information for every process). Also, this module contains a page for listing all flows of the user and a page for listing the template flows that can be used directly by a user.

4.1.1.2 ExamManagementPages

This module is composed of all the pages and their corresponding UI components about managing all the exams of a user. Also, a template store page for pre-defined exams is also in this module. Exam management is done through an exam editor where users can add, remove or edit any question in the exam.

4.1.1.3 UserManagementPages

This module is composed of all the pages and their corresponding UI components about managing all the users in a company account. A company account can see all the users belong to it, add or remove any users, or change the access roles of those users. For example, a company account can restrict any user to read-only access to applications.

4.1.1.4 MainPages

This module is composed of all the landing pages whose purpose is to introduce the application to the customers.

4.1.1.5 ApplicantPages

This module is composed of all the pages and their corresponding UI components that an applicant will see on the website. These pages can be any application form or proficiency tests.

4.1.1.6 LoginPages

This module is composed of all the pages about the login process. Login and signup pages can be an example for these pages.

4.2 Business Layer

4.2.1 Gateway Service

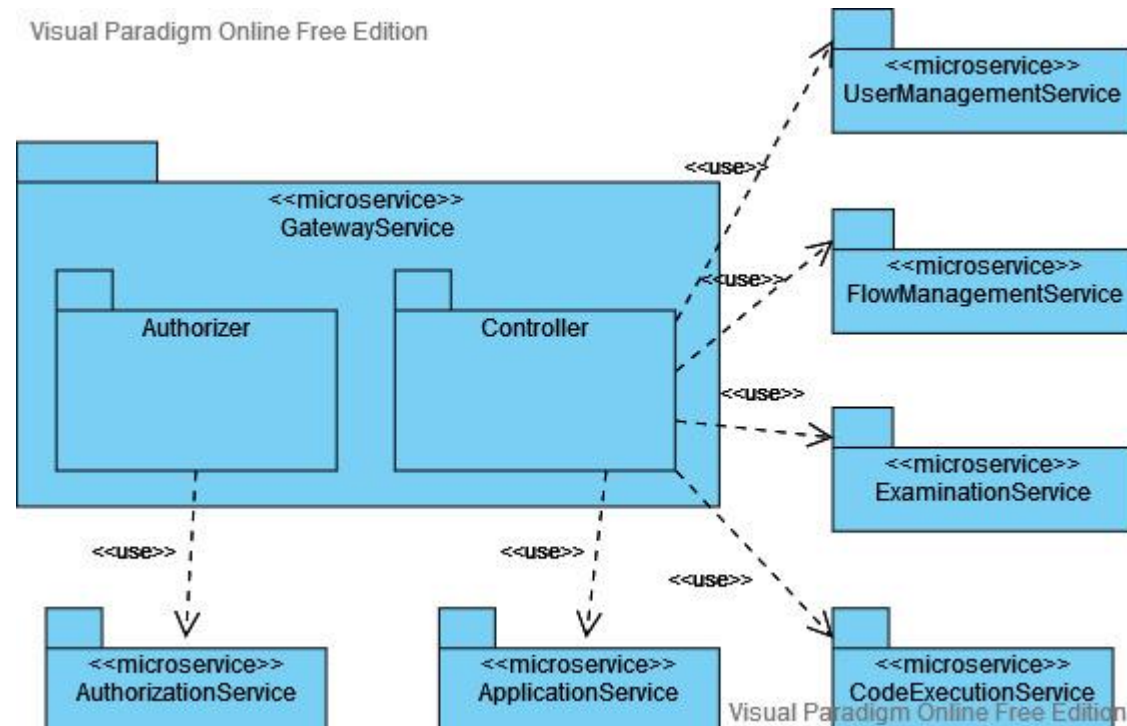


Figure 4. Gateway Service

Gateway service is responsible for routing the requests to the rest of the services. Using API gateway is very useful in microservices architecture because it provides benefits to both user and developer such as:

- Defines a better API.
- Services may change over time and should be hidden from the clients.
- Services may use different communication protocols internally.
- Limits the access to services from outside for better security.
- Load balancer may be added to the system easily.
- Number of service instances may change over time which should be hidden from the clients.

4.2.1.1 Authorizer

Authorizer module is responsible for checking whether the user is authorized to use the corresponding resource. This can be either an API endpoint that eventually uses one of the microservices or a specific website. Access roles of the users are also checked in this module to be able restrict the pages and endpoints to a specific type of user.

4.2.1.2 Controller

Controller module is responsible for redirecting the requests to any API endpoints to the corresponding microservice. One advantage of this module is one can define all the endpoints in the same place and use different microservices at the same time.

4.2.2 User Management Service

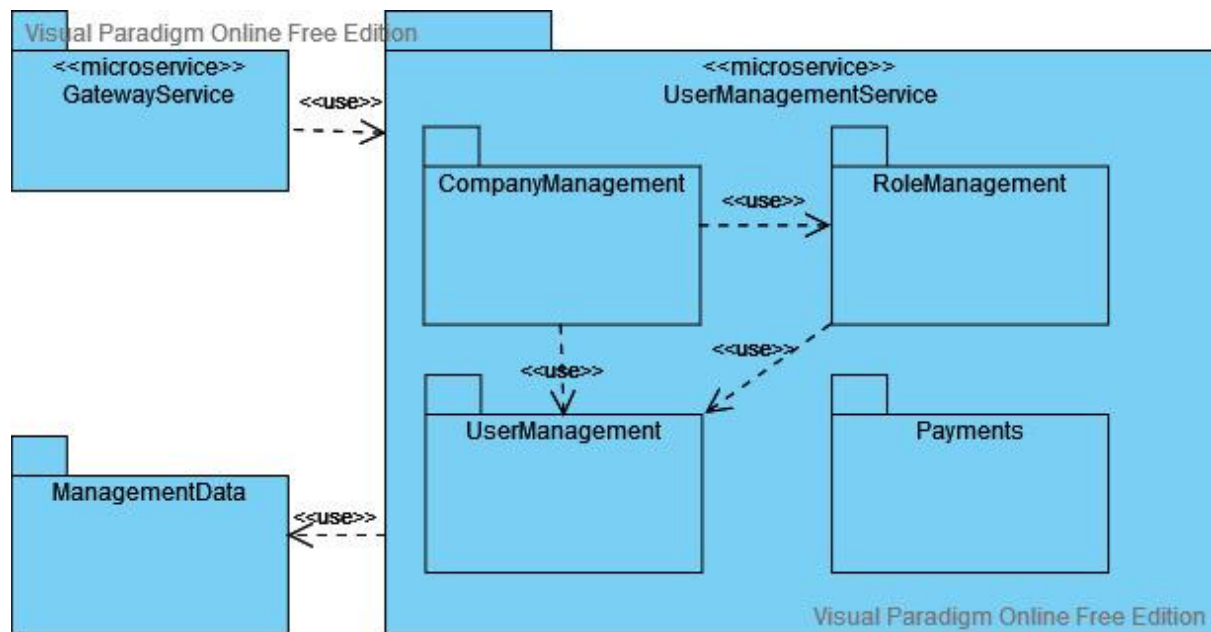


Figure 5. User Management Service

User management service is responsible for the user and account operations. It also includes payment operations as it is related to the account tariff.

4.2.2.1 CompanyManagement

This module is responsible for managing the information of the company account. This information can be anything that can describe the company such as name, mission, vision etc.

4.2.2.2 RoleManagement

This module is responsible for managing the roles of the users. These roles can be edited so that a specific role can access specific parts of the application. On the other hand, this restriction can be about the data rather than features. For example, company account can restrict one specific user's access to some of the flows.

4.2.2.3 UserManagement

This module is responsible for managing the users in a company. Company can send invitation emails to a specific person who then signs up to the website and access the materials that company allows. Furthermore, company can add, remove or edit users in its reach.

4.2.2.4 Payments

This module is responsible for managing operations about the payment that is done by users of the application. There may be different plans which offer different features for a different price. Managing these plans along with managing the payments done by a user is handled in this module.

4.2.3 Flow Management Service

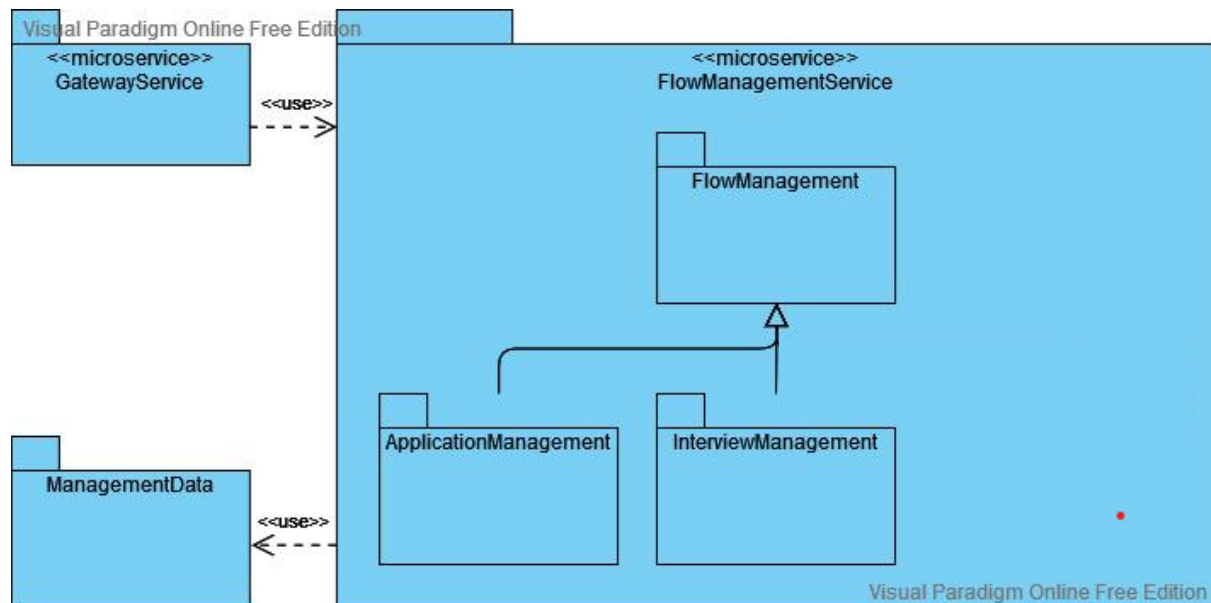


Figure 6. Flow Management Service

Flow management service is responsible for operations related to the recruitment flow. The creation of the flow and stages as well as the details of them are operated under this microservice.

4.2.3.1 FlowManagement

This module is responsible for managing all flows of a company account. Core features that are handled by this module are:

- Creating new flows
- Adding processes such as application form, interview, or proficiency exam to a particular flow
- Changing the order of these processes
- Editing or deleting a particular flow

4.2.3.2 ApplicationManagement

This module is responsible for managing all the application forms of a company account. Core features that are handled by this module are:

- Creating new application forms
- Adding, removing, or editing the questions in the form
- Editing or deleting a particular application form

4.2.3.3 InterviewManagement

This module is responsible for managing all the interview meetings of a company account. A company account can see all the interviews arranged for a particular flow. Moreover, it can

arrange new meetings, assign them to different users, edit the meeting information or send reminder emails both to the applicant and interviewer.

4.2.4 Examination Service

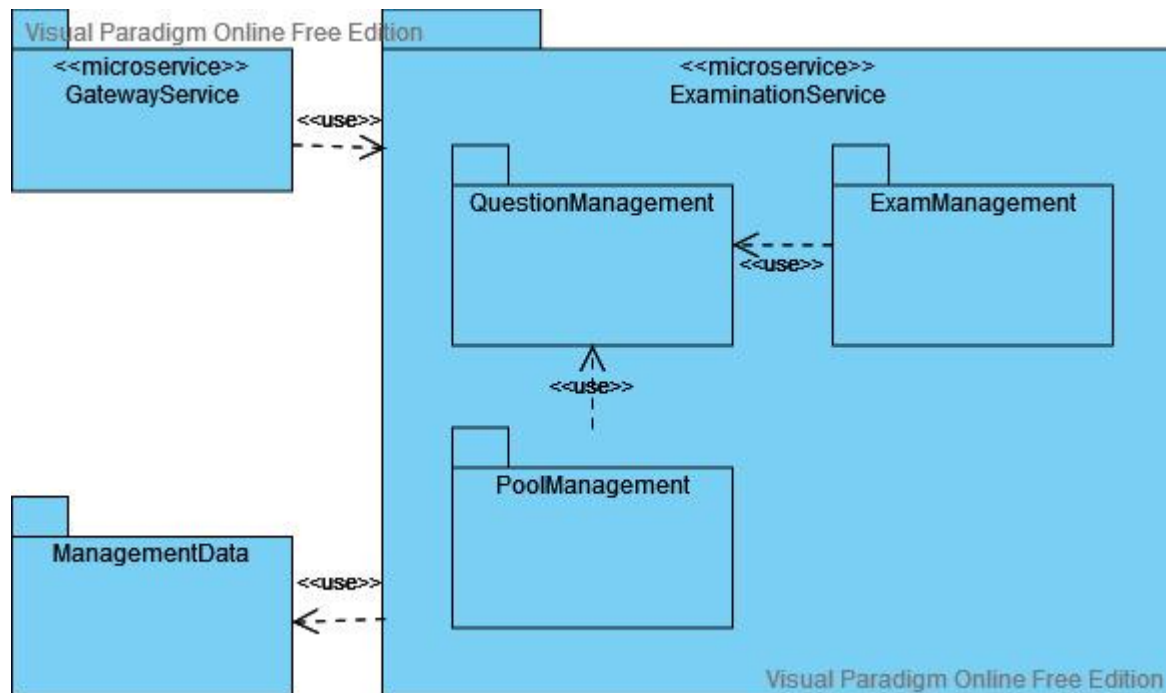


Figure 7. Examination Service

Examination service is responsible for the exams that recruiters will arrange. Exam management and details will be done on this service.

4.2.4.1 ExamManagement

This module is responsible for managing all the proficiency exams of a company account. Users can change settings of an exam like time limit for each question, deadline for the exam, etc. with the help of this module.

4.2.4.2 QuestionManagement

This module is responsible for creating new questions, editing or deleting them. Questions can be open-ended, multiple-choice, or coding questions. These questions can be used in any proficiency exam of the company account.

4.2.4.3 PoolManagement

This module is responsible for handling the question pool in the application. Every user can create a new question and add it to the public pool so that everyone can use it. Also, there can be private questions where access to these questions can be managed by the creator.

4.2.5 Code Execution Service

As a code execution service, we will use an external service called Judge0. Judge0 is able to run code snippets, measure the complexity and return the results. This is why it is designed as a separate microservice.

4.2.6 Application Service

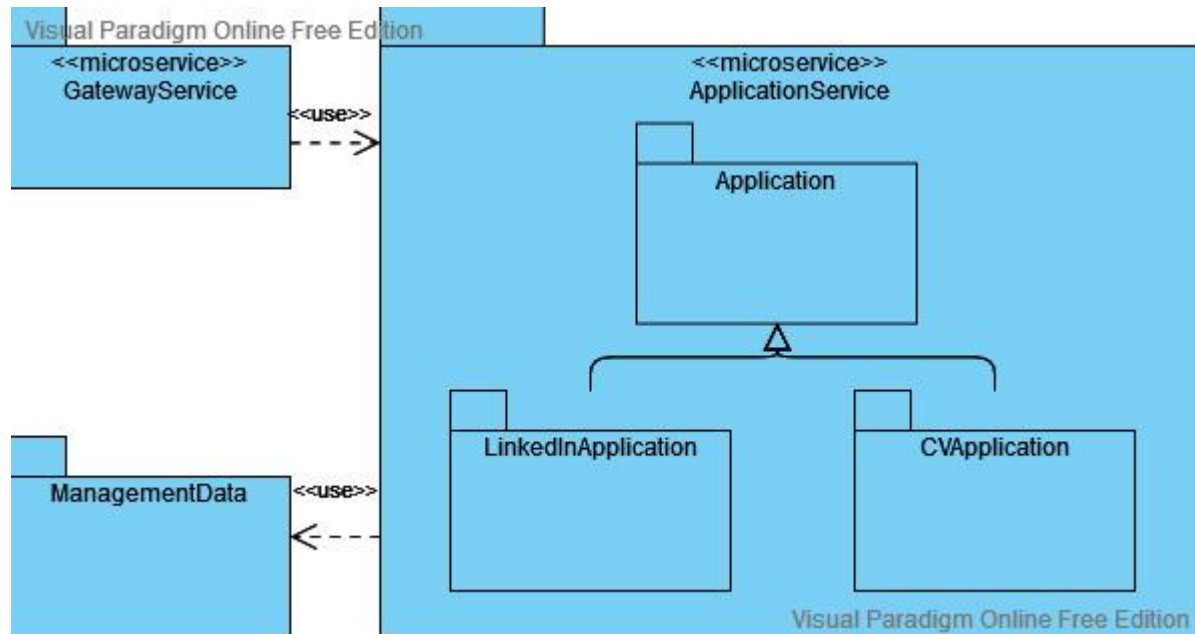


Figure 8. Application Service

Application service is responsible for filling application forms. An applicant may fill an application form using either a LinkedIn account or a CV. They will be implemented in this microservice.

Application using CV feature will be implemented using natural language processing techniques and most common libraries for this field are supported in the Python programming language. Using Python for this microservice will improve the quality of this feature. Besides, implementing application service as a separate microservice will provide reusability.

4.2.7 Authorization Service

As an authorization execution service, we will use an external service called Auth0. Auth0 is able to manage authorization operations conserving OAuth 2.0 standards. Auth0 also supports SSO and it is safe for scalable environments.

5. Consideration of Various Factors in Engineering Design

Recroute is an online platform which combines the different recruitment steps into a single platform. Therefore, it does neither affect nor is affected by public health, public safety and public welfare in any way at all. Also environmental and economic factors are irrelevant for the design of Recroute.

Global Factors:

Recroute is imagined as an online platform to be used by companies and applicants around the world. However, people speak different languages around the globe. This is why the general interface of the Recroute will be English so that people from different countries can interact with the platform.

Cultural Factors:

In different parts of the world, information shared by the applicants through the internet might be different from other parts of the world. This is why the privacy boundaries of different cultures need to be respected while designing Recroute.

Social Factors:

In Recroute platform, while people are applying to a job advert they share personal information through the system. In designing Recroute, the security of this information must be maintained.

Table 1: Factors

	Effect Level	Effect
Public Health	0	-
Public Safety	0	-
Public Welfare	0	-
Global Factors	4	Interface should be English
Cultural Factors	9	Privacy
Social Factors	9	Data protection
Environmental Factors	0	-
Economic Factors	0	-

6. Teamwork Details

6.1 Contributing and functioning effectively on the team

As developers of the Recroute project we believe that good teamwork can be achieved with people who fulfill their responsibilities and help each other when faced with various obstacles. In order to achieve this, we held meetings at regular intervals. In these meetings, we were discussing how the problems can be eliminated and how to share the works that need to be completed at the next stage. While sharing the work, we considered the past experiences of the team members in order to enable them to function effectively. This kind of sharing minimizes the problems that may arise and prevents delaying of the project by ensuring that everyone contributes at the maximum level.

6.2 Helping creating a collaborative and inclusive environment

All team members helped to create a collaborative environment by joining meetings. Thanks to these meetings, we get everyone's opinion on the subject and create an inclusive environment. We used many synchronous and asynchronous communication tools for that purpose. Also, Jira is used to distribute tasks and track progress more easily. In this way, team members could see what each other was doing and could help each other when needed. During the implementation phase, we also used Github to collaborate with other team members.

6.3 Taking lead role and sharing leadership on the team

We prepared different work packages which are explained detailly in the Analysis Report. For each work package we selected a leader who prepares the tasks to be completed at that work and controls the progress of the work at regular intervals until the deadline. Leadership roles are distributed fairly among the team members in order to improve their leadership skills without giving all responsibility to one person.

7. Glossary

- **Application form:** Form that should be filled by the applicants when applying for a job.
- **Job advert:** Overall application process created by company. It is composed of a recruitment flow and general information about the job.
- **Applicant:** Person who applies for a job advert.
- **Recruiter:** Company or a person who is responsible to hire employees.
- **Recruitment Flow:** The process of analyzing, testing and interviewing the job applicants and then finding the appropriate candidates for the job.
- **Stage:** A step on the recruitment flow. Can be one of the following: Application, Test or Interview.
- **Sandbox development service:** Service that is responsible for executing code written by an examiner in a coding test.