toGetJob

Technical Documentation

Gaia Meola 0306689

toGetJob

Academic year 2024/2025

Index

1.1 INTRODUCTION

1.1.1 AIM OF THE DOCUMENT

This document's primary goal is to present the fundamental and specific requirements of the **toGetJob** system. I will review all essential internal and external aspects related to the system. Various diagrams will support this analysis to provide clarity and a comprehensive understanding of the system's functionality, structure, and design.

1.1.2 OVERVIEW OF THE DEFINED SYSTEM

The system was created with the goal of providing a simple and intuitive platform that allows graduating students and companies to communicate easily.

Users of the platform will have the option to register either as **Students** or **Recruiters**.

Students will be able to view job announcements posted by recruiters, apply for job positions they are interested in and also review companies they have already worked for.

Recruiters will be able to publish job announcements, receive student job applications and invite other recruiters to manage specific job announcements.

1.1.3 HARDWARE AND SOFTWARE REQUIREMENTS

Software: Draw.io for creating diagrams related to system design. Scene Builder for generating FXML files used in the graphical user interface development. IntelliJ IDEA as the development platform.

Hardware: Operating Systems:Windows 7 or later. macOS v10.7 or later. Processor: Minimum 1 GHz, recommended 2 GHz. Memory (RAM):Minimum 2 GB, recommended 4 GB. Hard Disk: Minimum 3 GB, recommended 5 GB.

1.1.4 RELATED SYSTEMS: PROS AND CONS

Linkedin:

Pros: It allows browsing articles about the job market and watching educational videos.

Cons: It is not specifically aimed at graduating students.

Indeed:

Pros: It allows viewing the average salary for a job position.

Cons: It is not specifically aimed at graduating students.

1.2 USER STORIES

1) **As a** student, **I want to** find job announcements, **so that I** get to know companies where I would like to work.

2) **As a** company recruiter, **I want to** post a job announcement, **so that I** receive job applications from students.

3) **As a** student, **I want to** filter job announcements, **so that I** find the one that best suits my needs.

1.3 FUNCTIONAL REQUIREMENTS

1) **The system shall** display job announcements selectable by the student.

2) **The system shall** allow the recruiter to temporarily disable the reception of job applications for a specific job announcement.

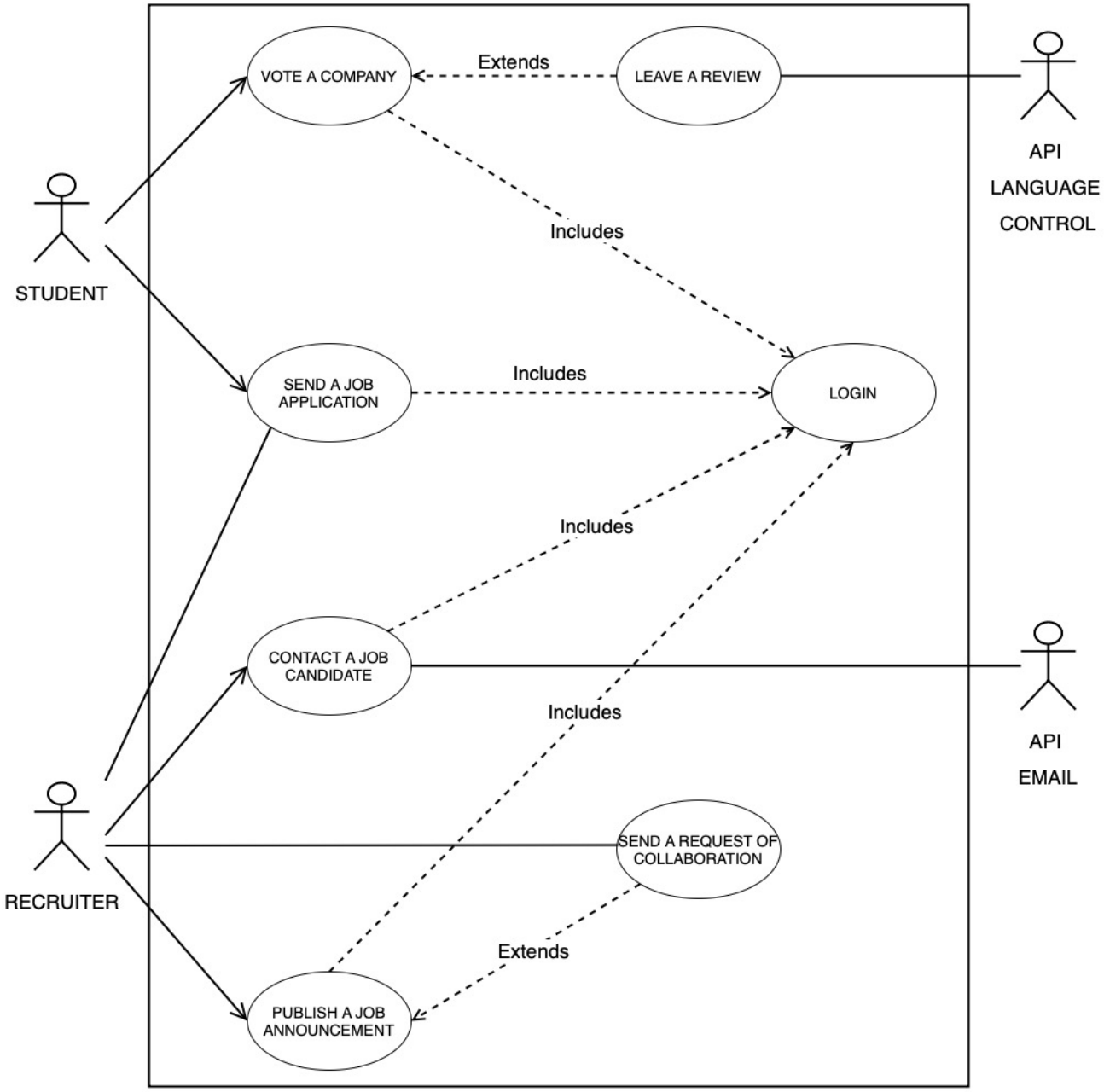
3) **The system shall** allow students to track the *status** of their job applications in real time.

Dictionary:

- ***Status***: *The job application status can be one of the following: Pending, Approved, Rejected. "Pending" means that the job application is still under review and has not yet been accepted by the recruiter. "Approved" indicates that the job application has been accepted, while "Rejected" means that the recruiter has declined it.*

1.4 USE CASE

1.4.1 OVERVIEW DIAGRAM



Note:

The diagram does not entirely reflect what has been implemented:

- *The use cases Vote a Company, Leave a Review and Send a Request of Collaboration have not been implemented in the code.*

1.4 USE CASE

1.4.2 INTERNAL STEPS

Name: **Send a job application** (Student):

1) The student selects the entry "*Send a job application*".
2) The system allows the student to specify optional *filtering criteria** for job announcements.
3) The student confirms his *filtering criteria** by selecting the entry: "*Proceed*".
4) The system displays the filtered list of *successful** job announcements, based on the specified criteria (if any).
5) The student selects a job announcement.
6) The system displays the selected job announcement.
7) The student selects the entry: "*Send your job application*".
8) The system prepares a *job application form** to be filled out.
9) The student fills out the job application form.
10) The student indicates they have completed by selecting the entry: "*Submit*".
11) The system saves the job application in the student's account.
12) The system *notifies** the *recruiter** of the new job application.
13) The recruiter accepts/rejects the job application.
14) The system saves the job application in the recruiter's account.

Extensions:

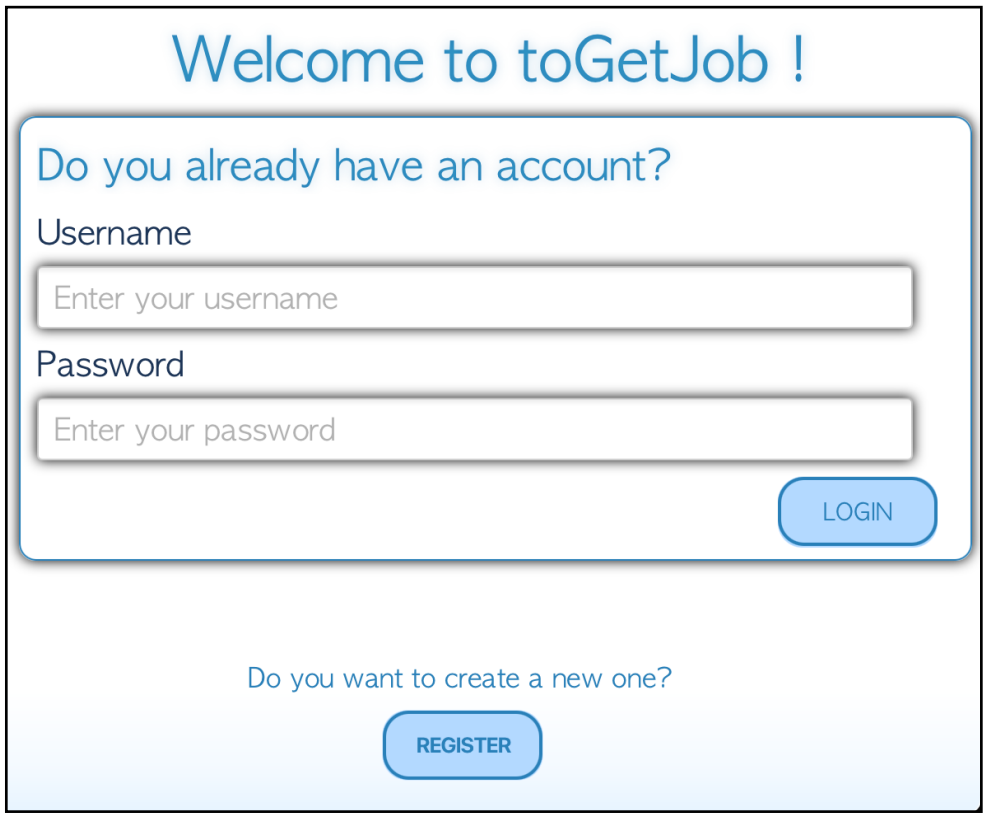
10a) ***The company has temporarily disabled the option to receive job applications***: the system informs the student that the company will temporarily not accept job applications. Proceed to **step 4**.
10b) ***The student has already submitted a job application to the company***: the system will show the message: "This application has already been processed and cannot be modified or deleted. Please visit the 'View sent job applications' section to view the current status of your application." Proceed to **step 4**.
13a) ***The recruiter accepts the job application***: the system updates the job application's status as "Accepted ". Proceed to **step 14**.
13b) ***The recruiter rejects the job application***: the system updates the job application's status as "Rejected". The use case then **terminates**.

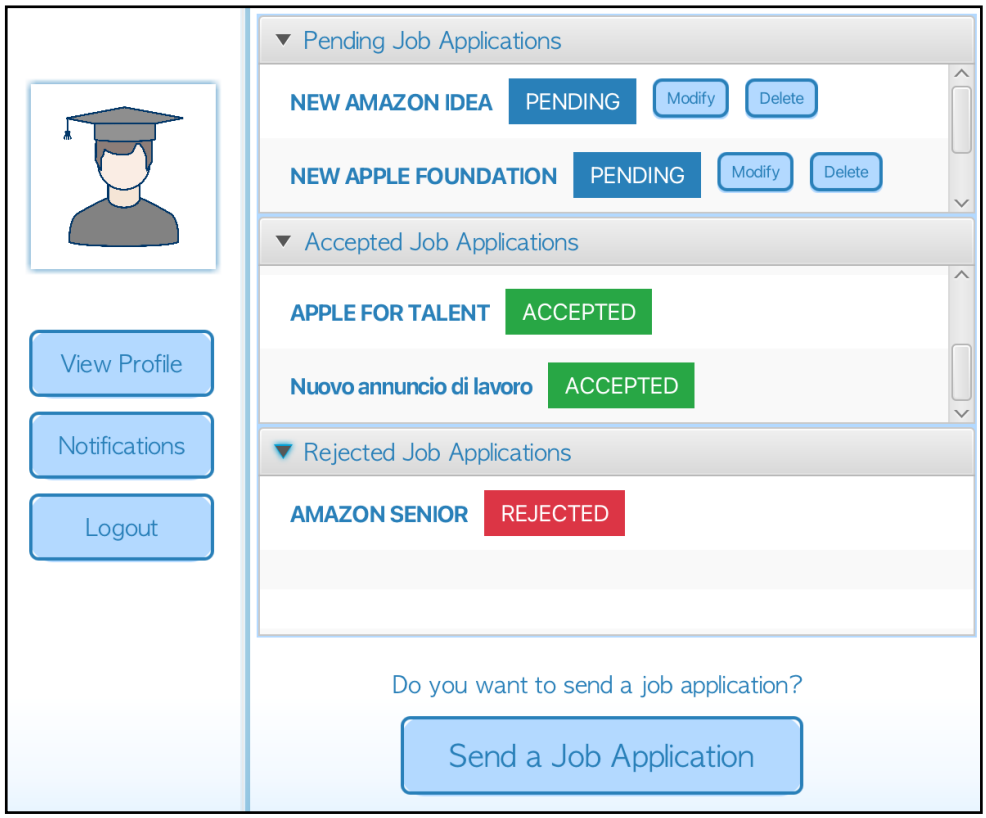
Dictionary:

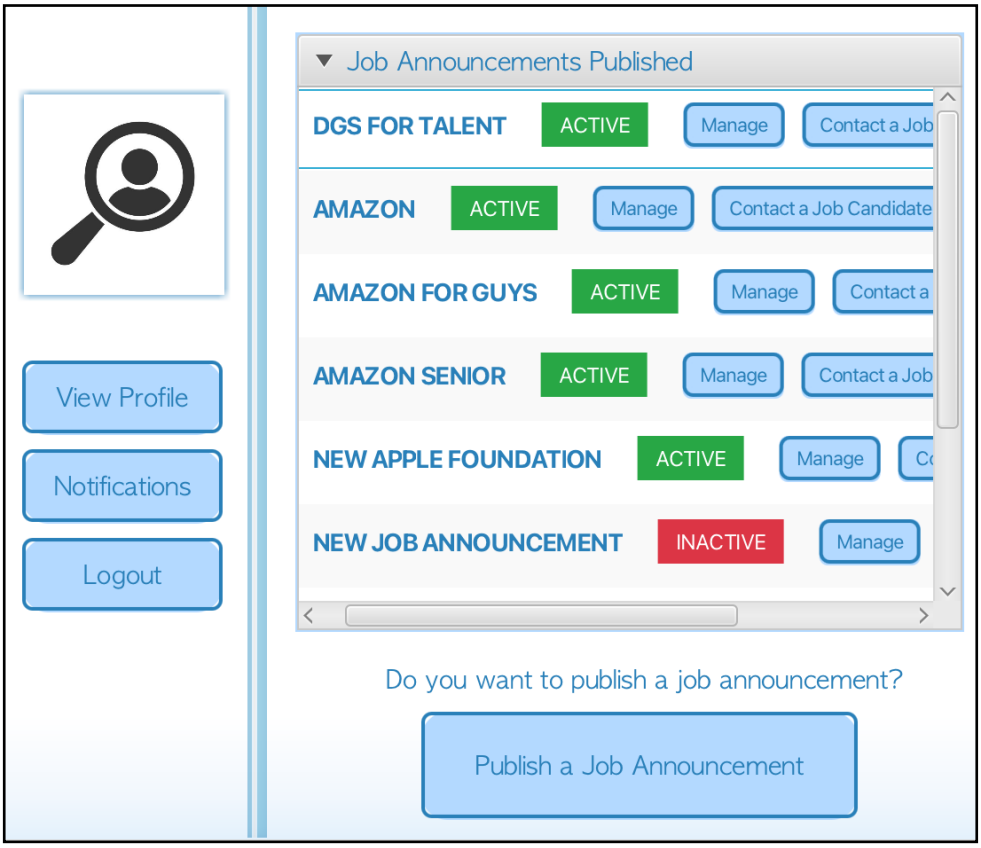
- ***Filtering Criteria****: Job Title, Job Type, Role, Location, Working Hours, Company Name, Salary.*
- ***Successful****: The job announcements that match the applied filters.*
- ***Job Application Form****: Cover Letter.*
- ***Recruiter****: The user who published the job announcement or someone working with the publisher of the job announcement (collaborators).*
- ***Notifies****: A notification is an automatic alert generated by the system that informs the recruiter about an application sent by the student. These notifications are generated every time a candidate applies in response to a published job announcement.*

Note:
Since the use case involves interaction between two actors, only the steps performed by the second actor, which are essential for a general understanding of the use case, have been included, avoiding the details of his specific actions.

2 STORYBOARDS

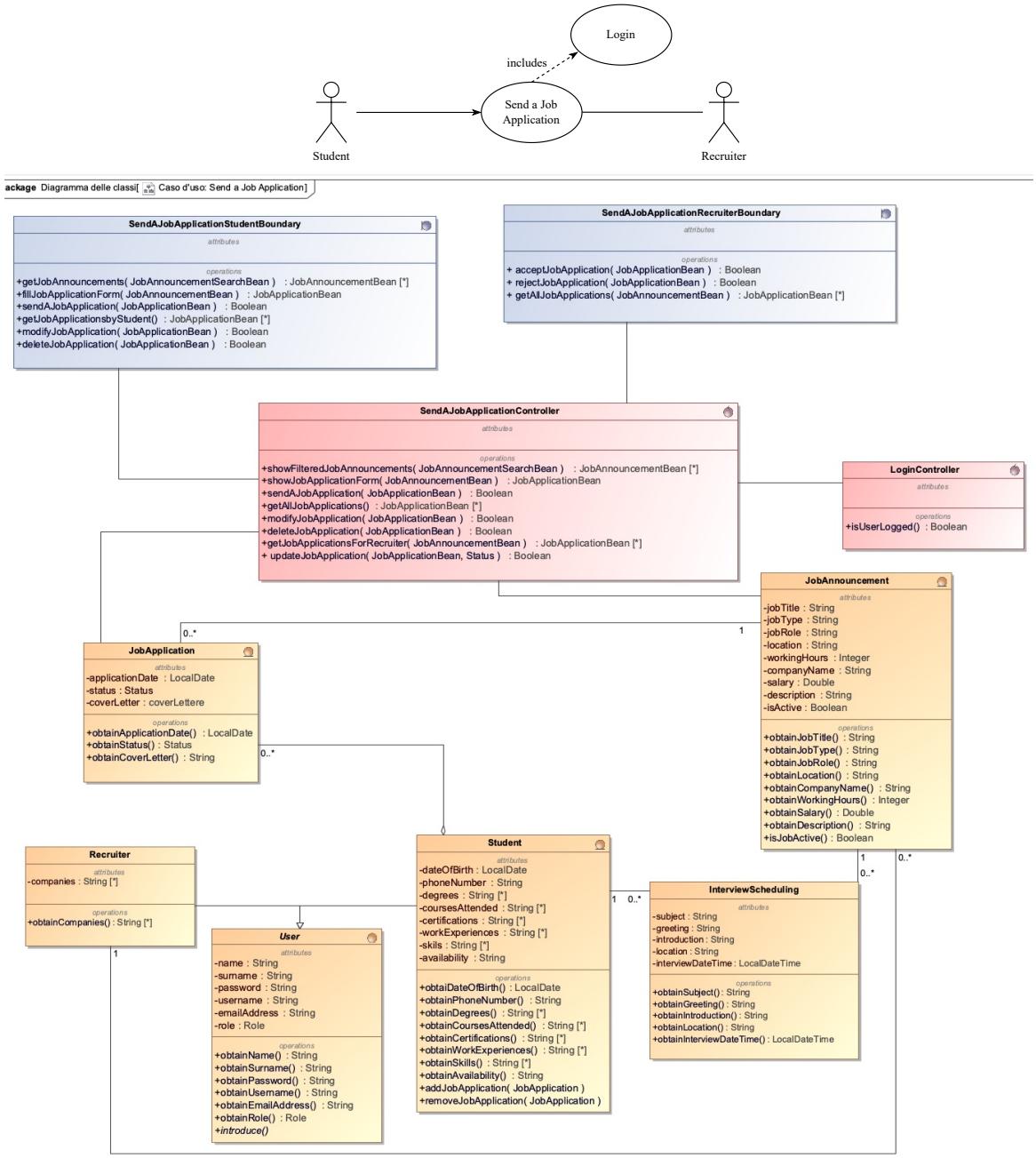





View Profile
Notifications
Logout
Job Announcements Published
DGS FOR TALENT
ACTIVE
Manage
Contact a Job
AMAZON
ACTIVE
Manage
Contact a Job Candidate
AMAZON FOR GUYS
ACTIVE
Manage
Contact a
AMAZON SENIOR
ACTIVE
Manage
Contact a Job
NEW APPLE FOUNDATION
ACTIVE
Manage
NEW JOB ANNOUNCEMENT
INACTIVE
Manage
Do you want to publish a job announcement?
Publish a Job Announcement

3.1 CLASS DIAGRAM

3.1.1 *VOPC related to the use case* ***Send a Job Application***

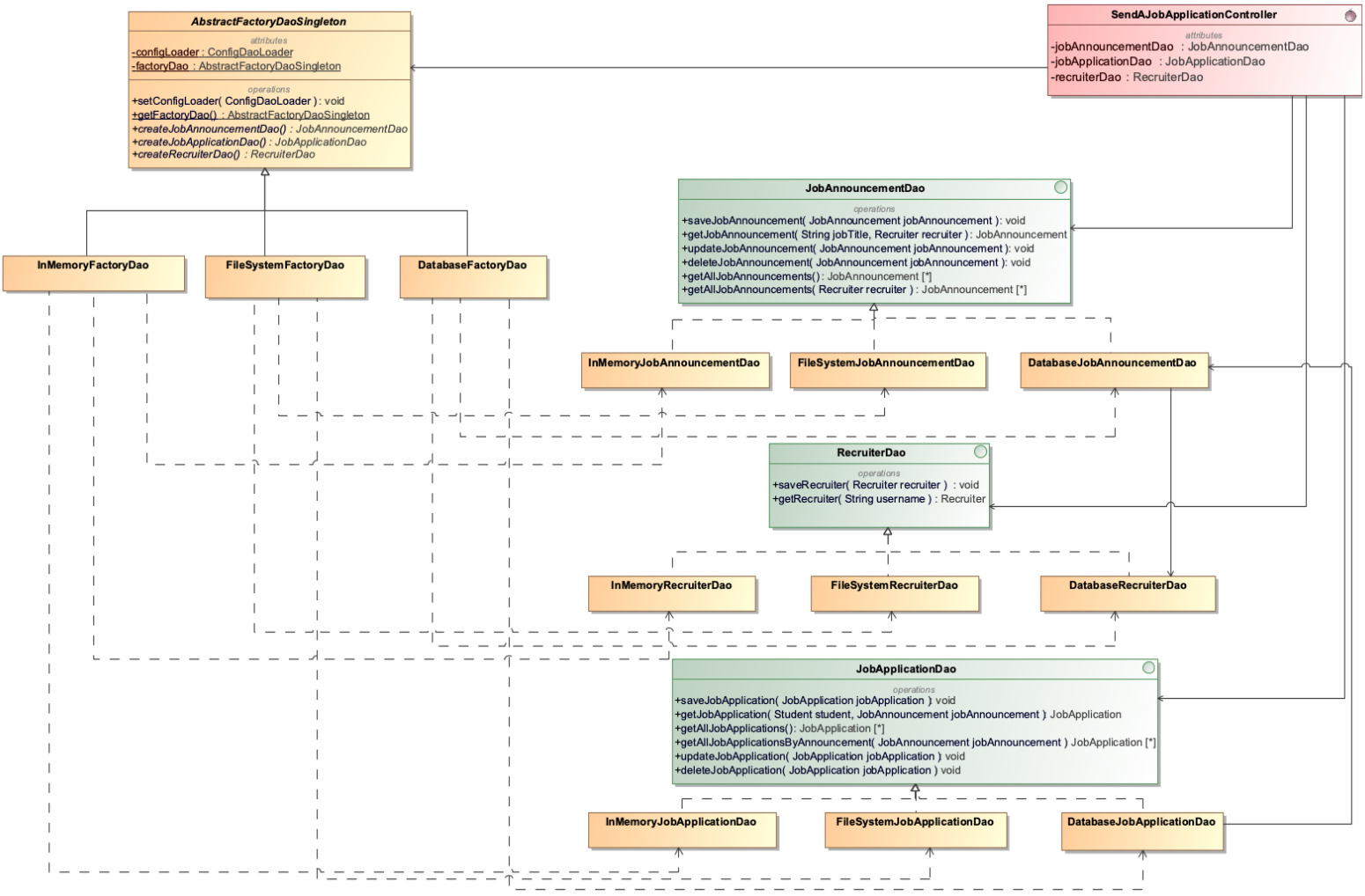


Note:

The diagram represents a simplified version compared to the implemented code, focusing more on visualizing the overall structure.

3.1 CLASS DIAGRAM

3.1.2 *DESIGN LEVEL related to the pattern:* ***Abstract Factory***

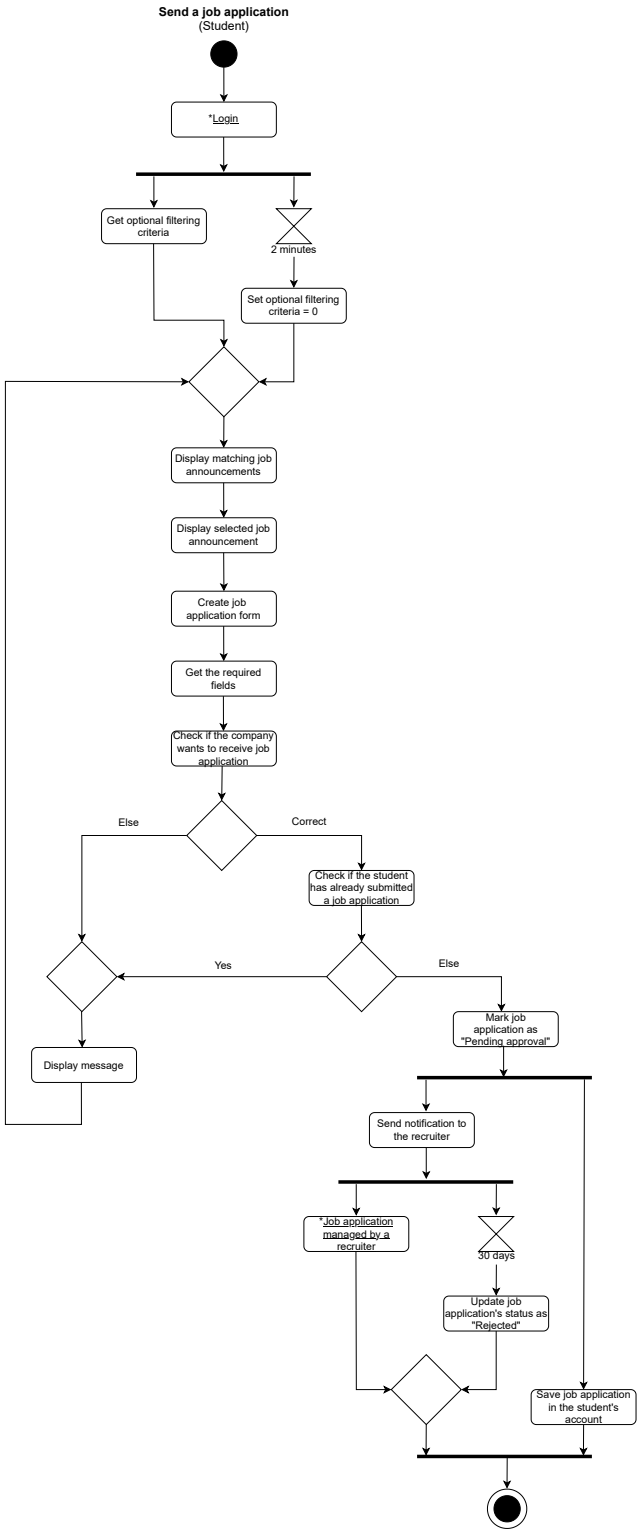


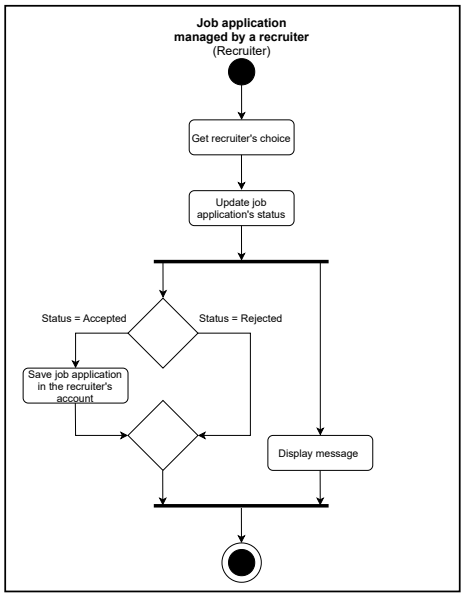
Note:

The diagram represents a simplified version compared to the implemented code, focusing more on visualizing the overall structure.

3.2 ACTIVITY DIAGRAM

Name: **Send a job application**





Note:

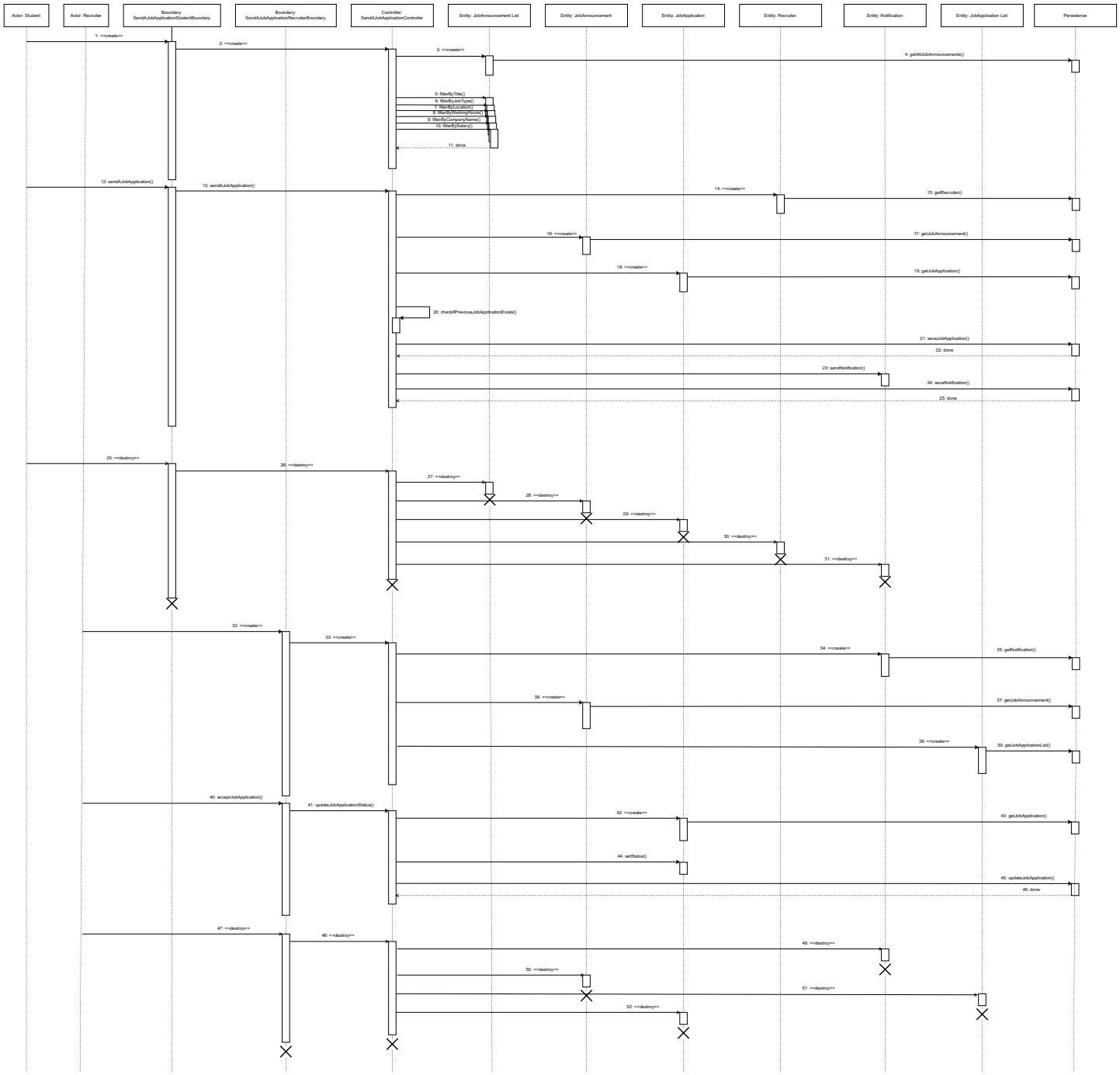
The diagram does not entirely reflect what has been implemented:

- *The timers have not been implemented.*

Since the use case involves interaction between two actors, only the steps performed by the second actor, essential for a general understanding of the use case, have been included, avoiding the details of the specific actions he carried out.

3.3 SEQUENCE DIAGRAM

Name: **Send a job application**



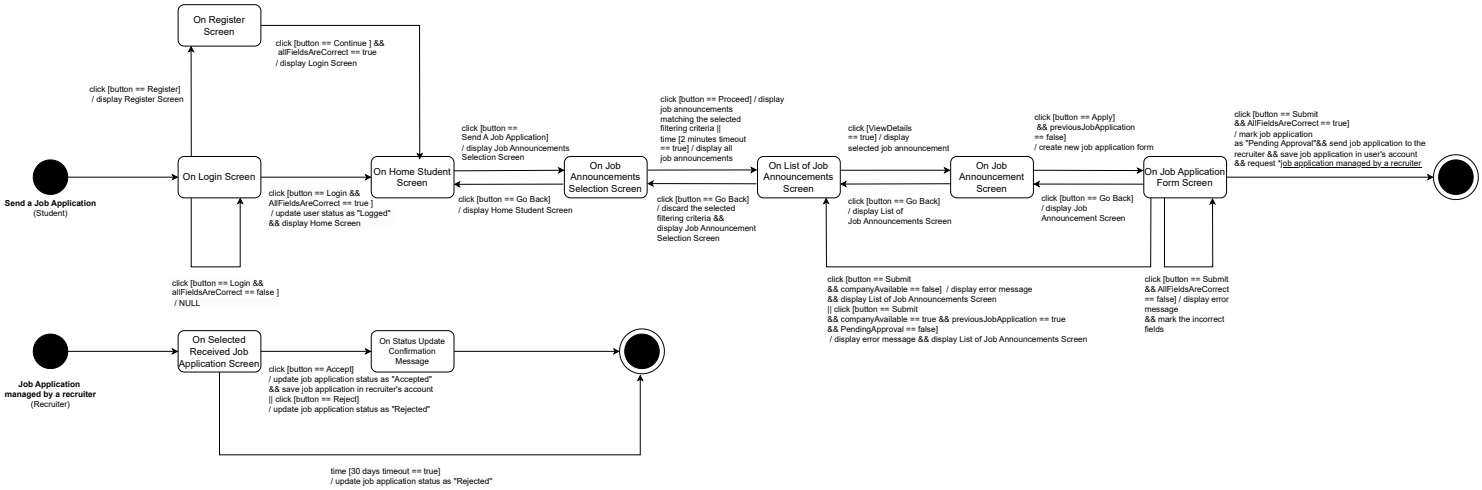
Note:

The diagrams do not entirely reflect what has been implemented:

- *The asynchronous filtering actions shown in the diagrams have not been implemented asynchronously in the code.*
- *The notification shown in the diagrams has not been implemented in the code.*

3.4 STATE DIAGRAM

Name: **Send a job application**



Note:

Since the use case involves interaction between two actors, only the steps performed by the second actor, essential for a general understanding of the use case, have been included, avoiding the details of the specific actions he carried out.

4 TESTING

For the testing phase, I have developed the following test classes:

1) ***Test Email***: checks if the email is already taken.

2) ***Test Login Username***: checks if the username is correct.

3) ***Test Null ConfigDaoLoader***: checks if an error occurs while configuring the DAO.

5 SONARCLOUD

The SonarCloud link associated with the project is:

https://sonarcloud.io/organizations/togetjob/projects