IZMIR UNIVERSITY OF ECONOMICS
FACULTY OF ENGINEERING
COMPUTER ENGINEERING

Design Document

SE 302 PROJECT

Author(s): Aybala Sena Susever
Begüm Köse
Berkay Gülen
Ceyda Dağcı
Selin Arslan
Zeynep Gürgün

Supervisor: İlker Korkmaz

1. Introduction

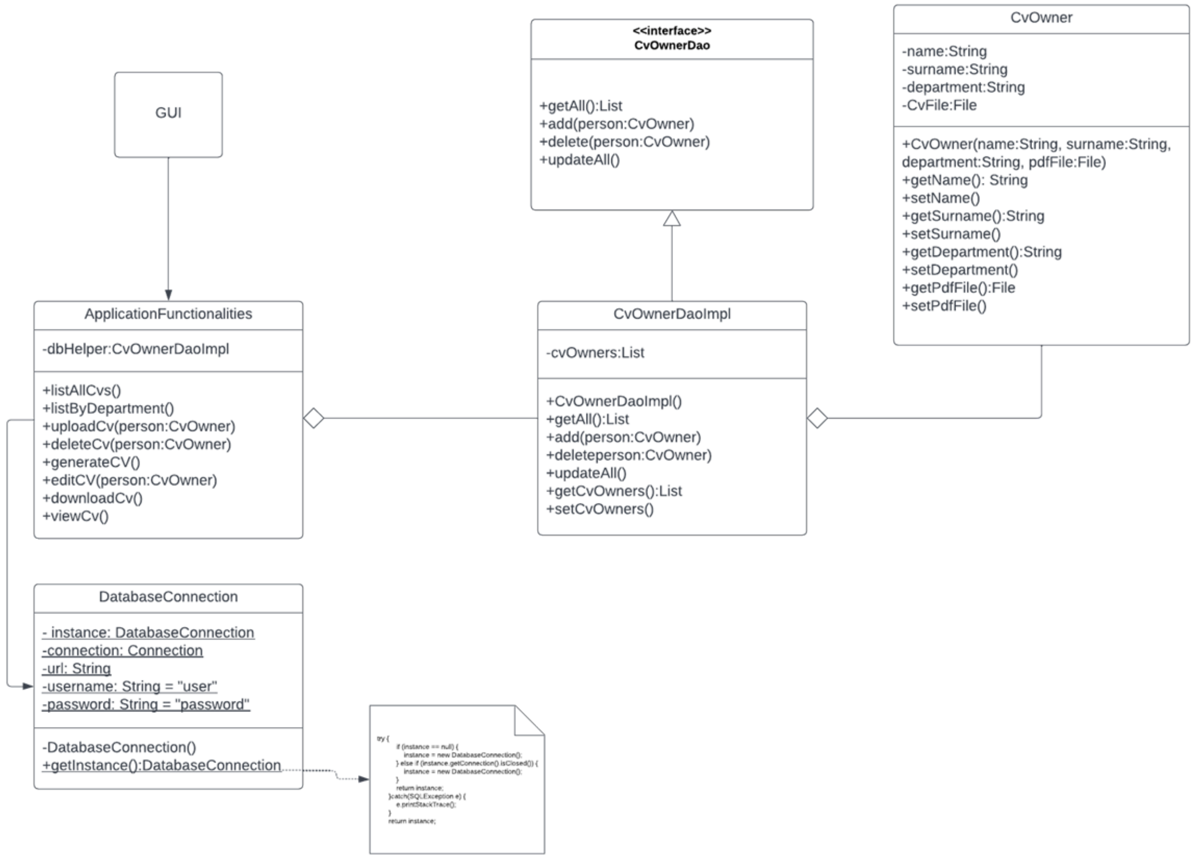
The Resume Management System is a desktop application running on Windows that can easily add, delete, edit, upload or view CVs. This application is mainly for department heads at our school but anyone who wants to collect CVs from various people such as students or academicians can also use this application. Moreover, Java programming language will be used for implementation.

2. Software Architecture

CV Database is a standalone desktop application that does not use any other application or services. CV Database has sets of internal classes for handling the necessary operations. The operations and system is supported with GUI for better looking application and make the software easy to use.

2.1 Structural Design

CV Database based on the operations that performed in **ApplicationFunctionalities** class. Our system fetches the data from the database and sets the data into a model class called **CvOwner**. The system flow and functionalities are now based on this fetched data.



```
try {
        if (instance == null) {
            instance = new DatabaseConnection();
        } else if (instance.getConnection().isClosed()) {
            instance = new DatabaseConnection();
        }
        return instance;
    }catch(SQLException e) {
        e.printStackTrace();
    }
    return instance;
```

Figure 1: UML Class Diagram

As you can see in Figure 1, **CvOwner** class is a model class which have the name, surname, department and CvFile information.

CvOwnerDao is a Data Access Object Interface that defines the standard operations to be performed on a model object(s).

CvOwnerDaoImpl is a Data Access Object concrete class that implements **CvOwnerDao** interface. This class is based on operations between our java application and our database.

ApplicationFunctionalities class has operations for our CV Database work. This class is like a bridge between our GUI and our database operations.

For example, a user wants to add a new CV into the database. First, the user should click the upload icon on the navigation bar. After the user can see the corresponding page for uploading. Now, users should see fields for name, surname, department and CV File. After the user fills the fields, the user should click the upload button and this button triggers the **uploadCv(person:CvOwner)** function in the **ApplicationFunctionalities** class. In this function, **the dbHelper** field in the **ApplicationFunctionalities** class is called and makes the corresponding upload operation. **ApplicationFunctionalities also** works like a Facade for the entire system.

DatabaseConnection class is a simple **Singleton** class for making the connection between system and the database. This is a Singleton object because we wanted to make sure that there is only one object instance from this class.

Requirement :*After the user selects a file, the user should be able to delete the file with the delete icon if the CV file is selected.*

Requirement *: The user should be able to upload a file by clicking the upload icon on the navigation bar.*

Requirement *: The user should be able to generate new CVs with provided information by the user.*

Requirement *: The user should be able to edit his/her generated CV.*

2.2 Behavioural Design

While in structural design, the user shall be able to upload, delete, generate and edit files with database operations. The following methods are essential for the functionality of the system.

2.2.1 Filtering and searching the CV Files

The user should be able to filter the CVs by the CV owners name and department. Also, the user should be able to search CVs. Searching will be done according to the CV file names. All CVs stored in CvOwner objects and the CvOwner objects stored in a local List named cvOwners. We can access those CVs by using the cvOwners List. We have 2 filtering options. Those are filter by name, filter by department. We have 1 search option which is to search by CV file name.

- ❖ Filter by Name

ApplicationFunctionalities class provides a method for this filtering. It uses the list name cvOwners created previously. In order to use this function, the user has to enter something in the filter by name text field. After that, the method linearly searches the cvOwners list and compares the name variables. If there are CV owners with the given name input, they will be displayed on the main page with their name, surname, department and CV file.

- ❖ Filter by Department

ApplicationFunctionalities class provides a method for this filtering. It is the same as the previous filtering technique. This time, the user has to enter a department into a corresponding text box, after that, the method compares the department variables. If there are CV owners with a given department, they will be displayed on the main page.

- ❖ Search by CV File name

ApplicationFunctionalities class provides a method for this searching method. The user should be providing a String into the corresponding text field for searching. After that, the method compares the CV file’s name with the given name. if it is matched, it will be displayed in the main page.

- ❖ Downloading a CV file into the user’s local directory

Downloading a CV file into the local directory will be handled by the **ApplicationFunctionalities** class. In order to download a file, the user has to choose a pdf on the main menu. Users should be able to choose a file by clicking on it. After that, the user should be able to download it by clicking the download button. This button triggers the corresponding function in the **ApplicationFunctionalities** class**.** In this method, the target directory will be asked to the user with the help of java build in functions for downloading.

***Requirement** : The user should be able to see all of the uploaded CVs on the main page.*

***Requirement** :The user should be able to filter CVs by name, department*

***Requirement** :The user should be able to search CVs according to the uploaded file names.*

***Requirement** :The user should be able to download the generated or selected CVs to his/her local directory.*

***Requirement** :*The user should be able to see all of the uploaded CVs on the main page.

3. Diagrams

3.1 Use Case Diagram

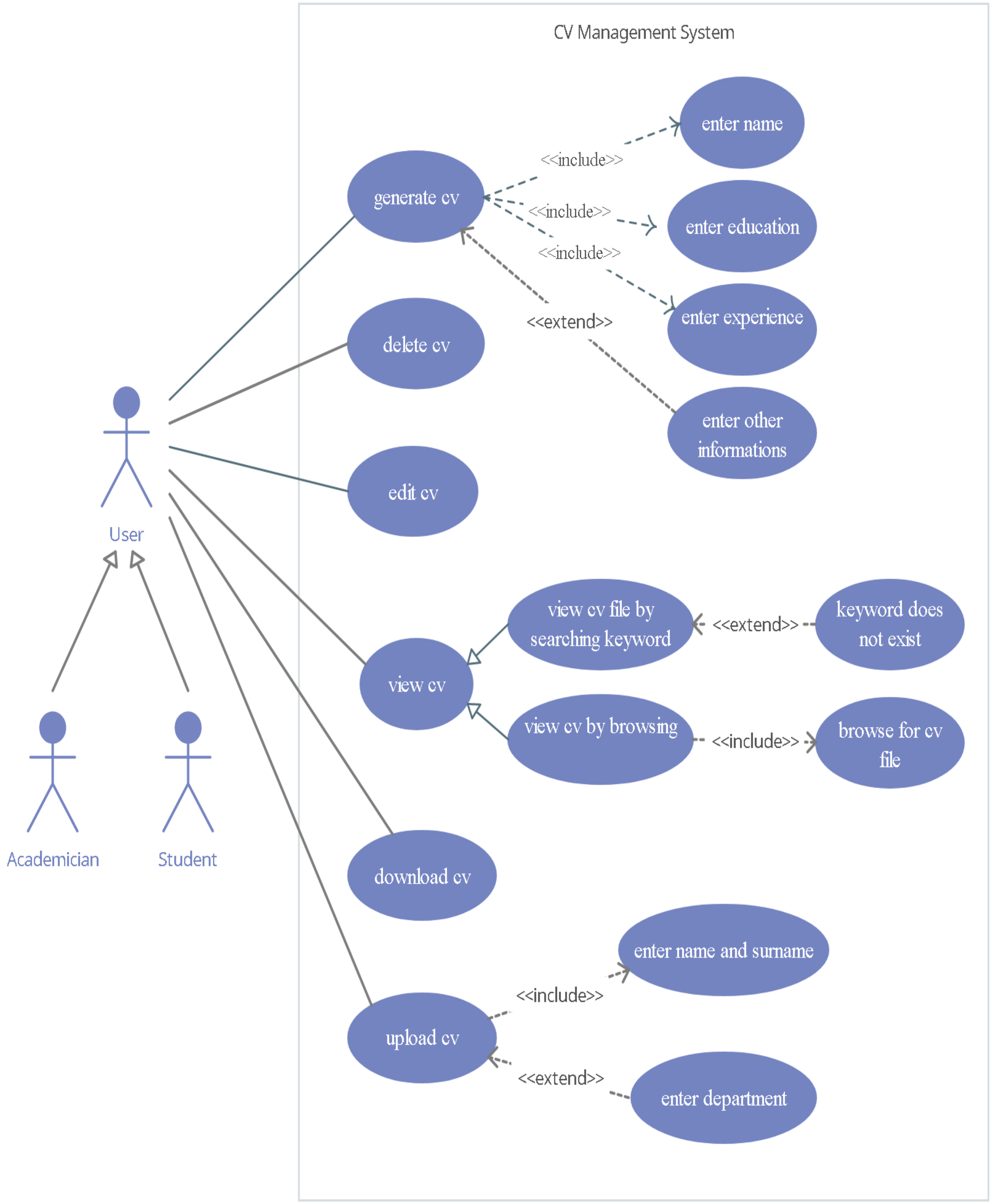


Figure 2: Use Case Diagram

The CV Management System does not require a login for the user to use the system directly, so when the application is opened, the person using the application on their computer is automatically counted as a user. This application is planned to be able to use the department heads in our school as a user, but anyone who wants to collect resumes from various people such as students and academics can also use this application as shown in the use case diagram figure. The system does not communicate with any external system or user.

The system provides some operations to the user such as generating, deleting, editing, viewing, downloading and uploading a cv. When the user generates a new cv, they shall enter the name, education and experience information of the cv owner. If desired, they can also enter other information such as certificates, skills or interests. When viewing a cv, the user shall view a cv by searching name or they shall browse a cv. If the user wants to search and view a cv by typing the name and the name does not match the existing CVs in the system, the system shall issue a warning saying that the file does not exist.

Requirement *: The user should be able to generate new CVs with provided information by the user.*

Requirement *: After the user selects a file, the user should be able to delete the file with the delete icon if the CV file is selected.*

Requirement *: The user should be able to edit his/her generated CV.*

Requirement *: The user should be able to search CVs according to the uploaded file names.*

Requirement *: The user should be able to see all of the uploaded CVs on the main page.*

Requirement *: The user should be able to filter CVs by name, department and interests.*

Requirement *: The user should be able to download the generated or selected CVs to his/her local directory.*

Requirement *: The user should be able to upload a file by clicking the upload icon on the navigation bar*

3.2 Activity Diagrams

3.2.1 Filtering CV’s

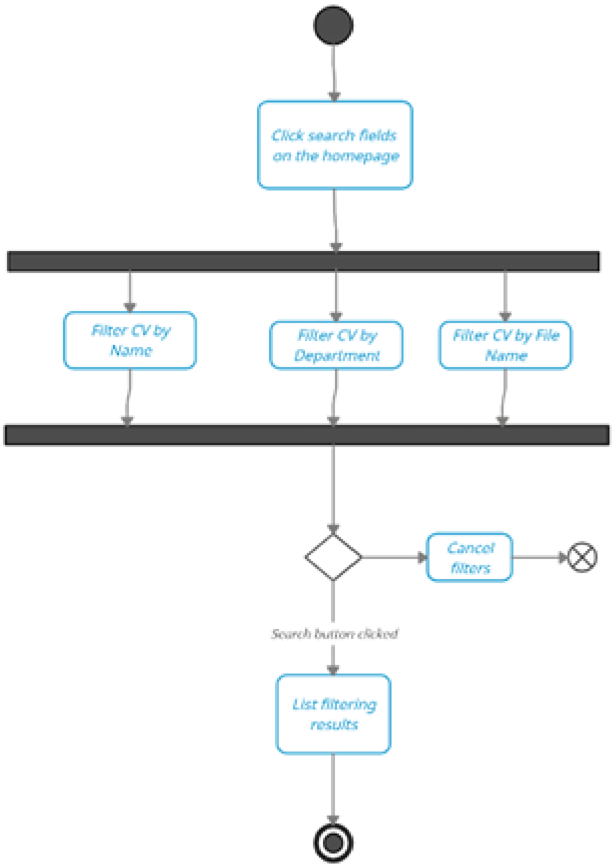


Figure 3: Filtering Activity Diagram

The filter CV’s activity diagram shows filtering CV’s on homepage. The user can filter the CV’s by filling name, department, interest search or file name fields, but does not want to change it then the user can click cancel and discard filters. If the user wants to apply filters, then clicks the Search icon. This diagram will meet the following Requirements.

***Requirement**: The user should be able to search CVs according to the uploaded file names.*

***Requirement**: The user should be able to filter CVs by name and department.*

***Requirement** : After the filtering the user should be able to see the filtered CVs automatically on the main page.*

3.2.2 Generating New CV

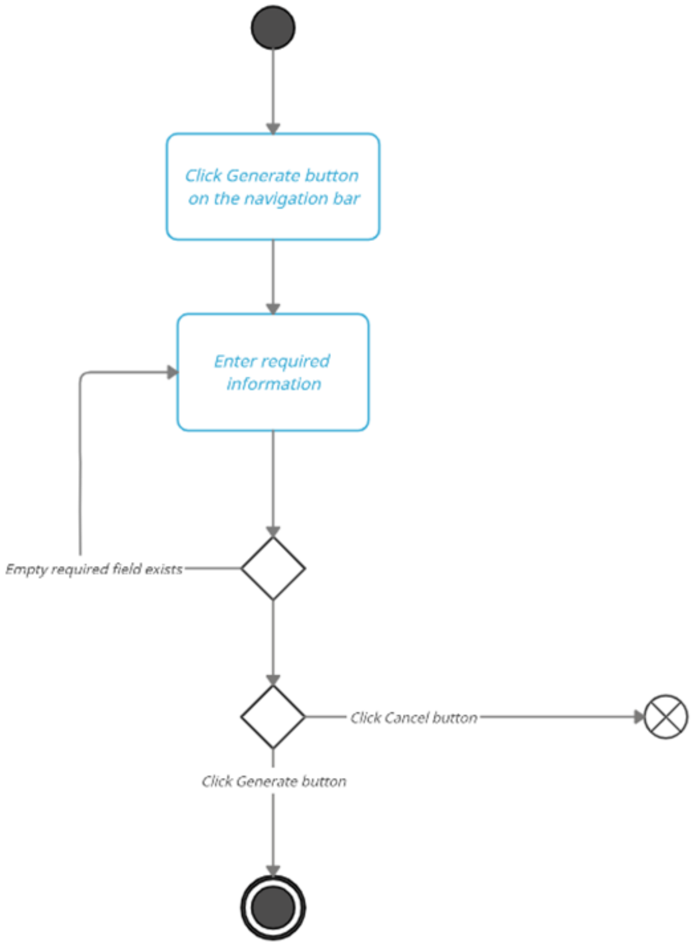

Click Generate button on the navigation bar
Enter required information
Empty required field exists
Click Cancel button
Click Generate button

Figure 4: Generate Activity Diagram

This activity diagram shows generating a new CV. The user can select the Generate button on the navigation bar, then the user can fill in the required information. If any empty required field exists, the user enters these parts too. If the user wants to save this document, then clicks the Generate CV button. If the user wants to delete it, then click the Cancel button. This diagram will meet the following Requirements.

Requirement: *The user should be able to generate new CVs with provided information by the user.*

3.2.3 Editing CV

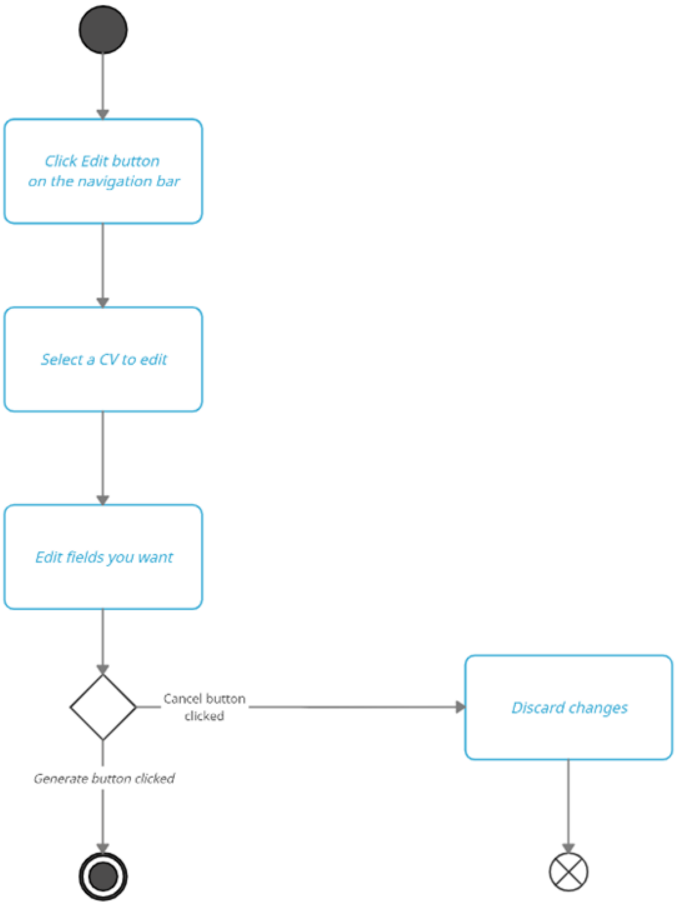


Figure 5: Editing Activity Diagram

This activity diagram shows editing a CV. The user can click the Edit button on the navigation bar, then the user can select a CV and edit fields he/she wants. If the user wants to discard changes, then click the Cancel button. If the user wants to save these changes, click the Generate button. This diagram will meet the following Requirement.

Requirement : *The user should be able to edit his/her generated CV.*

3.2.4 Downloading CV as PDF

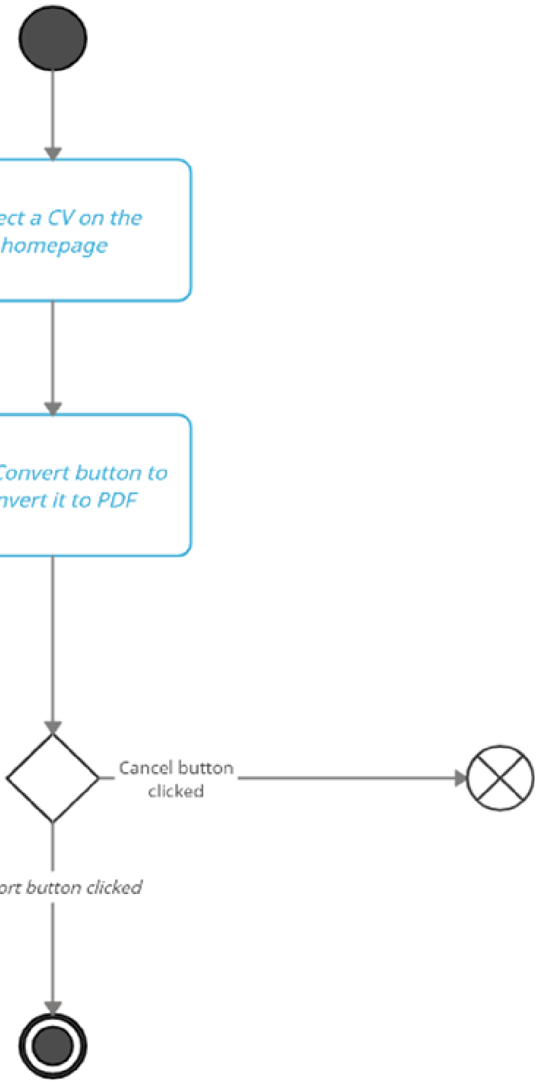

Select a CV on the homepage
Click Convert button to convert it to PDF
Cancel button clicked
Export button clicked

Figure 6: Downloading Activity Diagram

This activity diagram shows downloading a CV. The user can select a CV and click the Convert button to convert it to PDF. If the user wants to download it , then clicks the Export button, otherwise the user can click the Cancel button. This diagram will meet the following Requirements.

Requirement: *After generating a CV, the user should be able to convert it to PDF format.*

Requirement: *The user should be able to download the generated or selected CVs to his/her local directory.*

3.2.5 Deleting CV

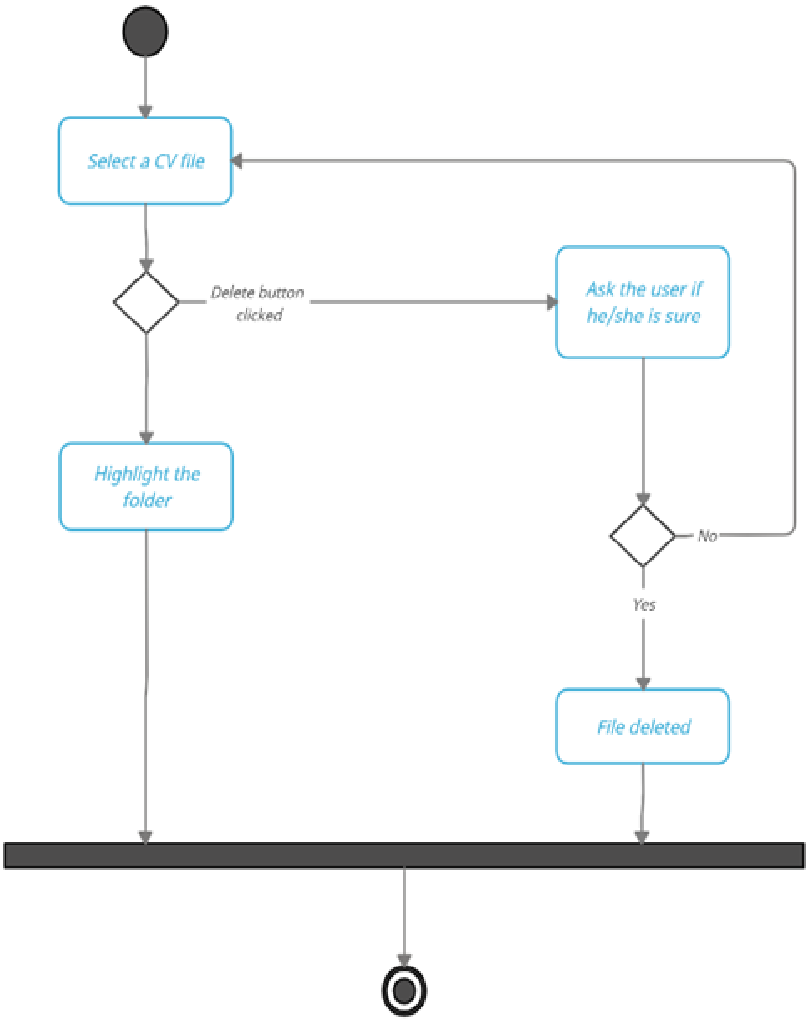


Figure 7: Deleting Activity Diagram

This activity diagram shows deleting a CV. The user can select a CV and click the delete button. The confirmation message is shown and if the user clicks Yes, the file is deleted, otherwise the process finishes without deletion. This diagram will meet the following Requirement.

Requirement *: After the user selects a file, the user should be able to delete the file with the delete icon if the CV file is selected.*

3.2.6 Uploading CV

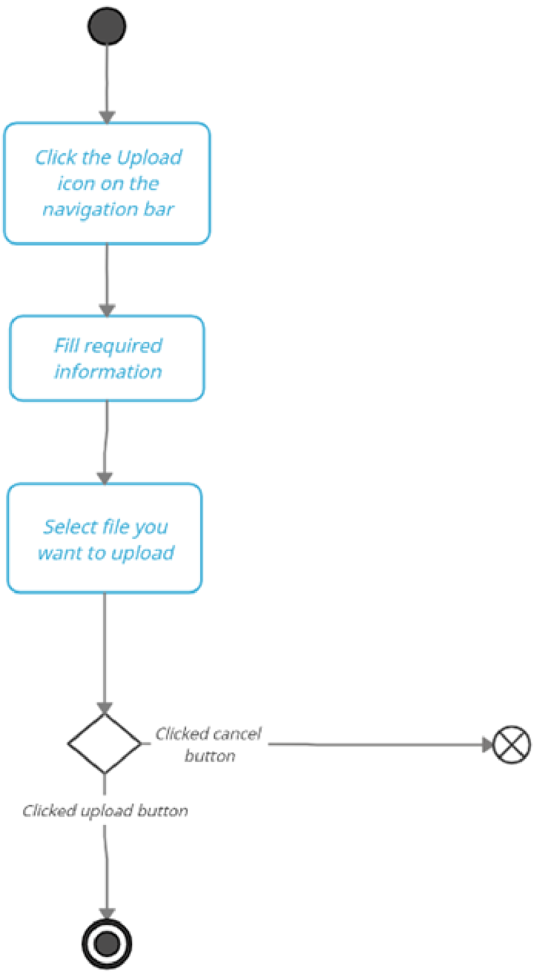

Click the Upload icon on the navigation bar
Fill required information
Select file you want to upload
Clicked cancel button
Clicked upload button

Figure 8: Uploading Activity Diagram

This activity diagram shows uploading a CV. The user can click the Upload icon on the navigation bar, then can fill required information and select a CV to upload. If the user clicks the upload button, the file is uploaded to the system. This diagram will meet the following Requirements.

Requirement : *After clicking the upload icon, the user should be navigated to the upload page.*

Requirement : *Inside the upload window, the user should be able to fill in the required information such as name, department, interests.*

Requirement : *The user should be able to see all of the uploaded CVs on the main page.*

3.3 Sequence Diagram

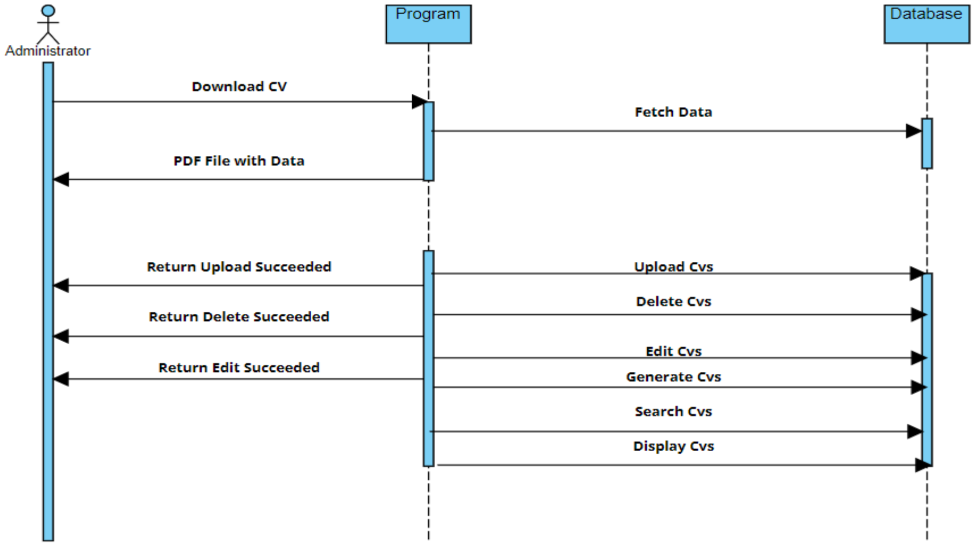


Figure 9: Sequence Diagram

The CV's sequence diagram shows the order in which messages are transferred between objects.The user can generate a CV with filling the required information if he/she does not want to upload. Also, if the user want to upload the CV he/she already has, the user can upload.If the upload process is done successfully in terms of accepting the appropriate format(PDF files), the user is informed about that success with a warning screen.The user can enter a name, department or file name to search the Cvs.The user can download a CV which is already generated or uploaded to the program and stored in database.The user can edit and delete a CV.If the delete and processes are done successfully,the user is informed about that success with a warning screen for an each of process.

***Requirement**: The user ought to be able to look for CVs using the names of submitted files.*

***Requirement**: The user should be able to search CVs.*

***Requirement** :With the user-provided data, the user should be able to generate new CVs.*

***Requirement** :The user should be able to edit CVs.*

***Requirement** :The created or chosen CVs have to be available for download to the user's local directory.*

***Requirement**: The user should be able to delete CVs.*

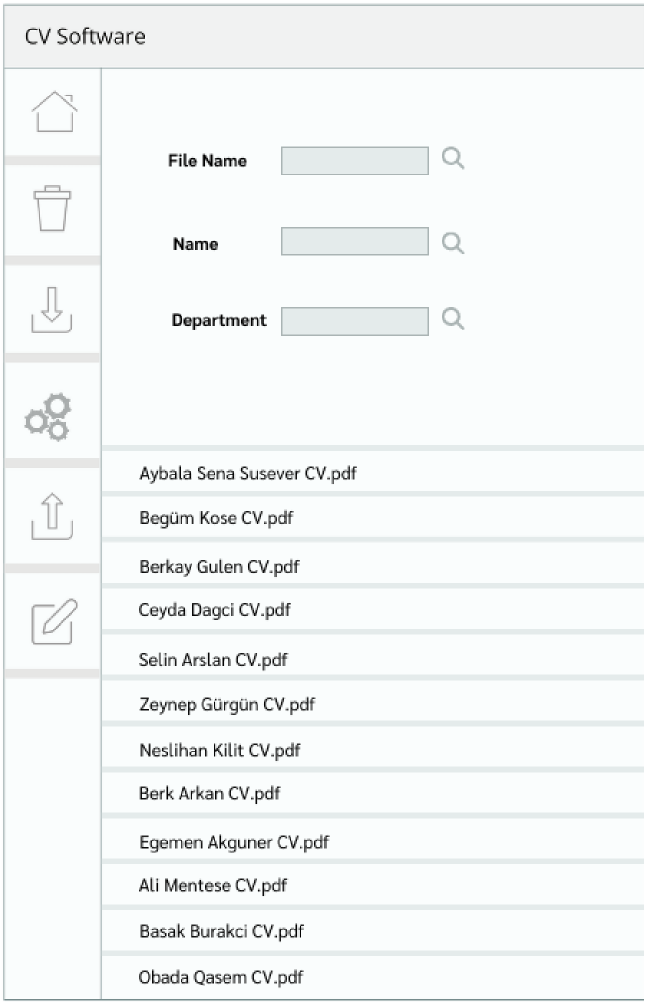
***Requirement**:The user should be able to view every CV that has been uploaded in one page.(home page)*

4. Graphical User Interface

Classes and their methods which are emphasized on below, grant a functional environment for the operations on the CV Management System. The GUI is going to grant the abilities to enter, edit or delete on the CV Management System, as well as displaying items on it.

The UI is going to reference guidelines of modern interfaces. It will have a menu bar on the left side on the screen with the following buttons tied to actions such as:

- Home Screen
- Search
  - Search by File Name
  - Search by Name
  - Search by Department
- Delete
  - Delete Selected CV
- Upload
  - Locate File and Upload to System
- Download
  - Download Selected CV to Device
- Generate
  - Generate Name
  - Generate Department
  - Generate Interest
  - Generate Text
- Edit
  - Edit Name
  - Edit Department
  - Edit Interest
  - Edit Text



Aybala Sena Susever CV.pdf
Begüm Kose CV.pdf
Berkay Gulen CV.pdf
Ceyda Dagci CV.pdf
Selin Arslan CV.pdf
Zeynep Gürgün CV.pdf
Neslihan Kilit CV.pdf
Berk Arkan CV.pdf
Egemen Akguner CV.pdf
Ali Mentese CV.pdf
Basak Burakci CV.pdf
Obada Qasem CV.pdf

Figure 10: GUI for CV Software application

“Home screen”, when clicked on, will take the user to the home screen of the CV Management System. When “Search” is selected, the user is going to have three different parameters to search items on the system. These parameters are file name, name and department information of the items. “Delete” action is straightforward as it will delete the selected CV from the CV management system. The action “Upload” will allow the user to select the file on their device and upload it to the CV Management System. “Download” action will let the user download the selected CV from the CV Management System to their device. “Generate” will let the user generate a new CV on the CV Management System by entering the required information such as “name”, “department”, “interest” and “text”. The action “Edit” will allow the user to edit a CV’s name, department, interest and/or text information on the CV Management System.

Interface is designed to be simple and straightforward as the menu bar on the left side of the screen will display the aforementioned options and actions with their respective icons, the rest of the screen will display the user what screen they’ve selected.

The project will have an extra READ.ME document within the software to clarify the usage of the software to the user.

Requirement : After the user opens the program the user should be able to see all the functions in application on the vertical navigation bar.

Requirement : The user should be able to see all of the uploaded CVs on the main page.