



**PATHFOLIO
SYSTEMS**

UFS RECRUITMENT SYSTEM

SOFTWARE ENGINEERING

**AMOGELANG SEIPONE
NALEDI DUBE**

**SEPINARE MOKHOTSOA
MPUCUKO LUTU
VUYO NZIMANDE**

Table of Contents

Table of Figures.....	5
Chapter 1: Project Proposal	7
Project Description	7
System Specifications	7
Student Features	7
Employer Features.....	7
Administrative Features	7
Project Scope	8
In Scope	8
Out of Scope	8
Target Audience.....	8
Initial Presentation of Additional Features.....	8
Conclusion.....	10
Chapter 2: Document Specifications.....	11
Introduction	11
Core System Use Cases	11
Login	12
Login UI.....	12
Login UML.....	13
Logout.....	16
Student Features.....	18
Professional Profile Management.	18
Profile Management UI	18
Profile Management UML	19
Application Portal	23
Application Portal UI.....	23
Application Portal UML.....	24
Academic Integration	27
Academic Integration UI.....	27
Academic Integration UML.....	28
CV Builder Tool	31
CV Builder UI.....	31
CV Builder UML.....	31
Job Matching and Notifications	35
Job Matching and Notifications UI	35
Job Matching and Notifications UML	36

Skills Tags and Badging	39
Skills and Badging UI.....	39
Skills and Badging UML.....	40
Employer Features.....	43
Mark Access	43
Mark Access UI	43
Mark Access UML	44
Employer Registration	47
Employer Registration UI.....	47
Employer Registration UML.....	48
Employer Deregistration.....	52
Employer Deregistration UI	52
Employer Deregistration UML	53
Advanced Searching and Filtering	55
Advanced Searching and Filtering UI	55
Advanced Searching and Filtering UML.....	56
Job Posting.....	59
Job Posting UI	59
Job Posting UML	60
Interview Scheduling System.....	64
Interview Scheduling System UI	64
Interview Scheduling System UML	65
Administrators Features	68
Analytics Dashboard	68
Analytics Dashboard UI.....	68
Analytics Dashboard UML.....	69
User Access Control	72
User Access Control UI.....	72
User Access Control UML	73
Support Chatbot	75
Support Chatbot UI.....	75
Support Chatbot UML.....	76
Chapter 3: Software Project Management Plan	78
Introduction	78
Software Project Management Plan.....	78
Chapter 4: Quotation	89
Chapter 5: Design Documentation.....	90
Noun Extraction	90

Description Summary	90
Table	91
CRC Cards.....	92
Pseudo Code	95
Class Diagram.....	101
State Charts.....	102
State Charts: Student.....	102
State Charts: Employer	103
State Chart: Administrator.....	104
Sequence Diagrams.....	105
Login	106
Forget Password	107
Registration.....	108
Employer Deregistration.....	109
Logout.....	110
Profile Management.....	111
Application Portal	112
Academic Integration	113
CV Builder	114
Job Matching	115
Skills Tags and Badging	116
Mark Access	117
Advanced Searching Filtering	118
Job Posting.....	119
Interview Scheduling	120
Dashboard.....	121
User Access Control	122
Support Chatbot	123
UFS Recruitment System - Time Allocation (12-week Project).....	124

Table of Figures

Figure 1: Presentation slide 1	9
Figure 2: Presentation slide 2	9
Figure 3: Summary Use Case.....	11
Figure 4: Login UI	12
Figure 5: Login UML	13
Figure 6: Profile Management UI.....	18
Figure 7: Profile Management UI.....	19
Figure 8: Application Portal UI	23
Figure 9: Application Portal UML.....	24
Figure 10: Academic Integration UI	27
Figure 11: Academic Integration UML	28
Figure 12: CV Builder UI	31
Figure 13: CV Builder UML	31
Figure 14: Job Matching and Notification UI	35
Figure 15: Job Matching and Notification UML	36
Figure 16: Skills and Badging UI	39
Figure 17: Skills Tag and Bagging UML.....	40
Figure 18: Mark Access UI.....	43
Figure 19: Mark Access UML.....	44
Figure 20: Employer Registration UI	47
Figure 21: Employer Registration UML.....	48
Figure 22: Employer Deregistration UI.....	52
Figure 23: Employer Deregistration UML	53
Figure 24: Advanced Searching and Filtering UI	55
Figure 25: Advanced Searching and Filtering UML	56
Figure 26: Job Posting UI.....	59
Figure 27: Job Posting UML.....	60
Figure 28: Interview Scheduling System UI.....	64
Figure 29: Interview Scheduling UML	65
Figure 30: Analytics Dashboard UI	68
Figure 31: Dashboard UML	69
Figure 32: User Access Control UI.....	72
Figure 33: User Access Control UML.....	73
Figure 34: Chatbot UI	75
Figure 35: Chatbot UML	76
Figure 36: User CRC Card	92
Figure 37: Student CRC Card	92
Figure 38: Employer CRC Card	92
Figure 39: Administrator CRC Card	92
Figure 40: Application CRC Card	93
Figure 41: CV CRC Card	93
Figure 42: Skills CRC Card	93
Figure 43: Portfolio CRC Card.....	93
Figure 44: Tools CRC Card	94
Figure 45: Chatbot CRC Card.....	94
Figure 46: Interview CRC Card	94
Figure 47: Class Diagram.....	101
Figure 48: Student State Chart.....	102
Figure 49: Employers State Chart	103

Figure 50: Administrators State Chart 104

Chapter 1: Project Proposal

Project Description

The University of the Free State (UFS) has commissioned the development of an innovative recruitment and student profile platform to enhance graduate employability and streamline employer engagement. The system will function as a bridge between academic achievement and career placement, integrating with existing UFS databases to provide real-time updates on student information.

The platform will be available as both a mobile application and a web-based application, ensuring accessibility across devices. Students will be able to create professional profiles, showcase skills, and apply for opportunities directly, while employers can search for suitable candidates and manage recruitment activities efficiently.

This project supports UFS's strategic goal of preparing students for the workplace and providing companies with a reliable source of qualified entry-level talent. Future system extensions may include alumni tracking, career path recommendations, and advanced AI-driven features, ensuring the platform remains scalable and adaptable.

System Specifications

Student Features

- **Professional Profile Management** - Create and update bios, skills, qualifications, and achievements
- **Academic Integration** - Secure API-based connection with UFS databases for automated mark updates, with student control over visibility
- **CV Builder Tool** - AI-assisted generation of CVs based on academic and profile data
- **Job Matching & Notifications** - Automated recommendations for internships, part-time work, and graduate roles
- **Application Portal** - Apply for jobs directly through the platform and track application progress
- **Skill Tags & Badging** - Visual representation of student competencies and certifications
- **Portfolio Showcase ("SkillStack")** - Upload projects, coursework, and achievements for employers to view

Employer Features

- **Advanced Candidate Search & Filtering** - Filter by GPA, skills, course background, graduation year, etc.
- **Employer Profiles & Job Posting** - Verified company profiles with job/internship postings
- **Interview Scheduling System** - Calendar-based scheduling with reminders and notifications
- **Optional Mark Access** - Employers may request academic information, shared only with student consent

Administrative Features

- **Analytics Dashboard** - Track job placements, employer activity, and skills demand trends

- **User Access Control** - Role-based access for students, employers, and administrators
- **Support Chatbot** - Assists with job search, CV generation, and profile setup
- **Skill Gap Analysis Tool** - Highlights gaps between student skills and job requirements, recommending training/resources

Project Scope

In Scope

- Development of a web application (primary focus for this project)
- Basic mobile application design and integration (included as part of system vision but secondary for this phase)
- All core functionalities outlined under student, employer, and administrative features
- Integration with UFS academic databases for real-time updates
- Initial testing, deployment, and training for university administrators
- Design of the Software Project Management Plan (SPMP), accounting for blended team collaboration (remote and in-person)

Out of Scope

- 1 Full implementation of the mobile app (beyond design stage)
- 2 Advanced AI-driven features (career path predictions, full automation of skill gap analysis) - these are future enhancements
- 3 Alumni portal and peer endorsements in the initial phase

Target Audience

- Students (undergraduate and postgraduate)
- Employers (local, national, and international recruiters)
- UFS Administrators (career services, academic staff)
- Alumni (future expansion)

Initial Presentation of Additional Features

As part of the project process, each group was required to prepare and present additional features for possible inclusion in the 2025 UFS Recruitment and Student Profile System. Our group developed a PowerPoint presentation outlining several ideas to enhance the system.

Although our proposed features were not selected for the final design, the presentation exercise was valuable in helping us think creatively about user needs and system expansion. During class, features from other groups were chosen instead, and these are reflected in the System Specifications of this document (Interview Scheduling, Skill Gap Analysis, and Portfolio Showcase).

This process demonstrated the importance of collaboration and critical evaluation in software engineering: not every idea is adopted, but each contributes to refining the overall system requirement

A copy of our group's initial presentation slides is included below for reference.

INTRODUCTION

- To enhance the UFS recruitment platform, our team at PathFolio Systems has designed a smart video interview feature
- This functionality aims to improve the screening process by allowing:
 - Students to record and review video responses
 - Admins to use AI tools to pre-screen candidates
 - Employers in turn receive high-quality pre-assessed applicants
- This feature improves efficiency, reduces bias in shortlisting, and helps employers focus only on the strongest candidates.



Figure 1: Presentation slide 1

AI-assisted Video interview system

Student Side

- Receives structured interview questions
- Records short video responses per question
- After all the responses, the videos are displayed in a list and students can preview, re-record or submit

Admin Side

- Uses AI tools to track eye-movement(to detect reading off script)
- Monitor body language, tone and speech
- AI then filters the interviews based on quality
- Admin reviews the best submissions and send it to the employers

Employer Side

- Receives shortlisted student profiles(after admin review)
- Sends correspondence to the selected students via email or phone
- Schedules the real(physical) interview outside the system.

Figure 2: Presentation slide 2

Conclusion

The 2025 UFS Recruitment and Student Profile System is a strategic initiative that addresses the growing need for stronger connections between students and employers. By providing an integrated platform that supports professional profile management, academic data integration, job matching, and employer engagement, the system will streamline the recruitment process and improve student employability outcomes.

The inclusion of features such as interview scheduling, portfolio showcases, and skill gap analysis ensures that the platform not only facilitates job applications but also helps students prepare for their careers more effectively.

By focusing on usability, security, and adaptability, this project lays the foundation for a long-term solution that benefits students, employers, and the University of the Free State alike.

Chapter 2: Document Specifications

Introduction

The system specification document outlines the functional requirements, user roles, and key features of the UFS Recruitment and Student Profile system, including profile management, academic integration, CV generation, job applications, and candidate search. It details how students, employers and administrators interact with the system for recruitment, career development and academic integration within the scope of a web-based platform, and targets students, employers and UFS administrators and primary users.

Core System Use Cases

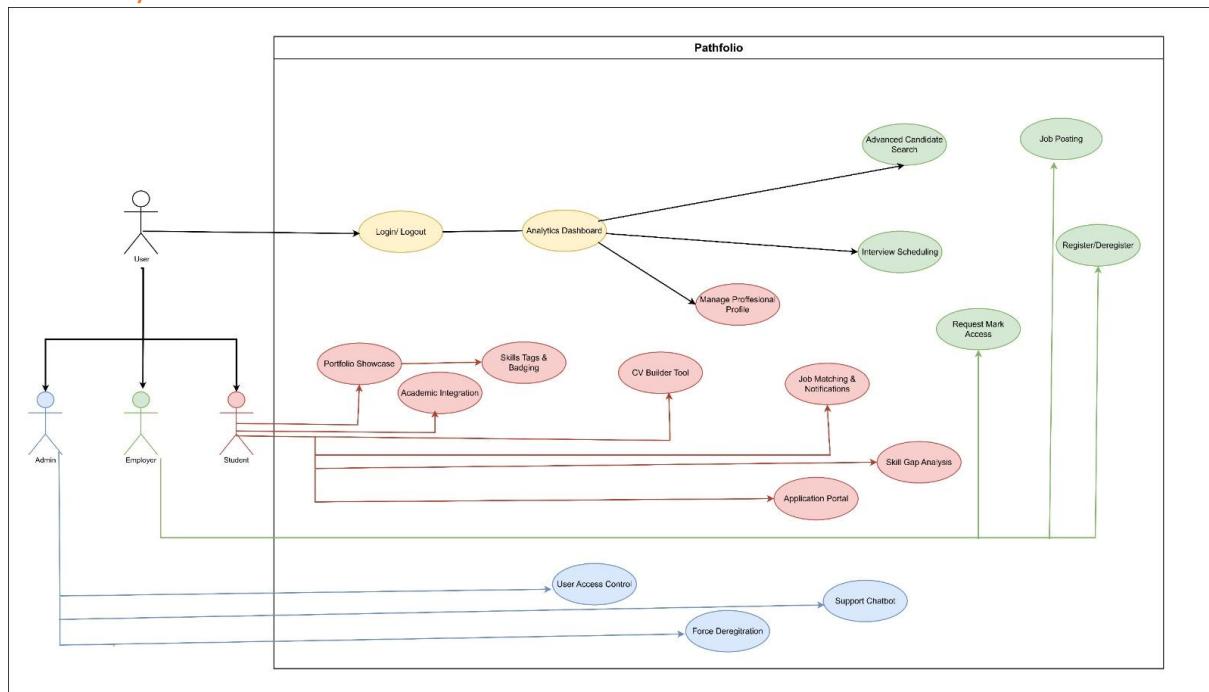


Figure 3: Summary Use Case

Login

This is the point of entry to the system. With textboxes for the user's email address and password. For as long as there is no input for the email address and password fields, the sign in buttons are greyed out.

Login UI

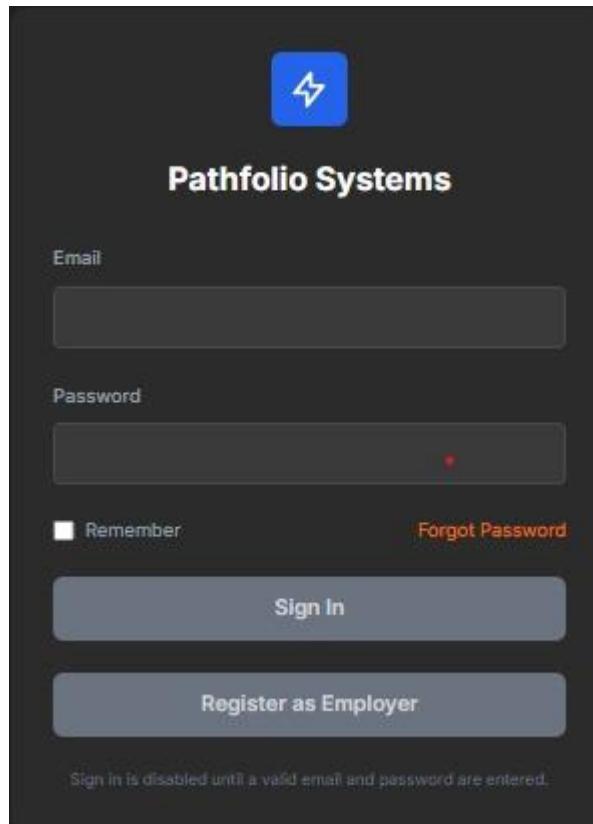


Figure 4: Login UI

Login UML

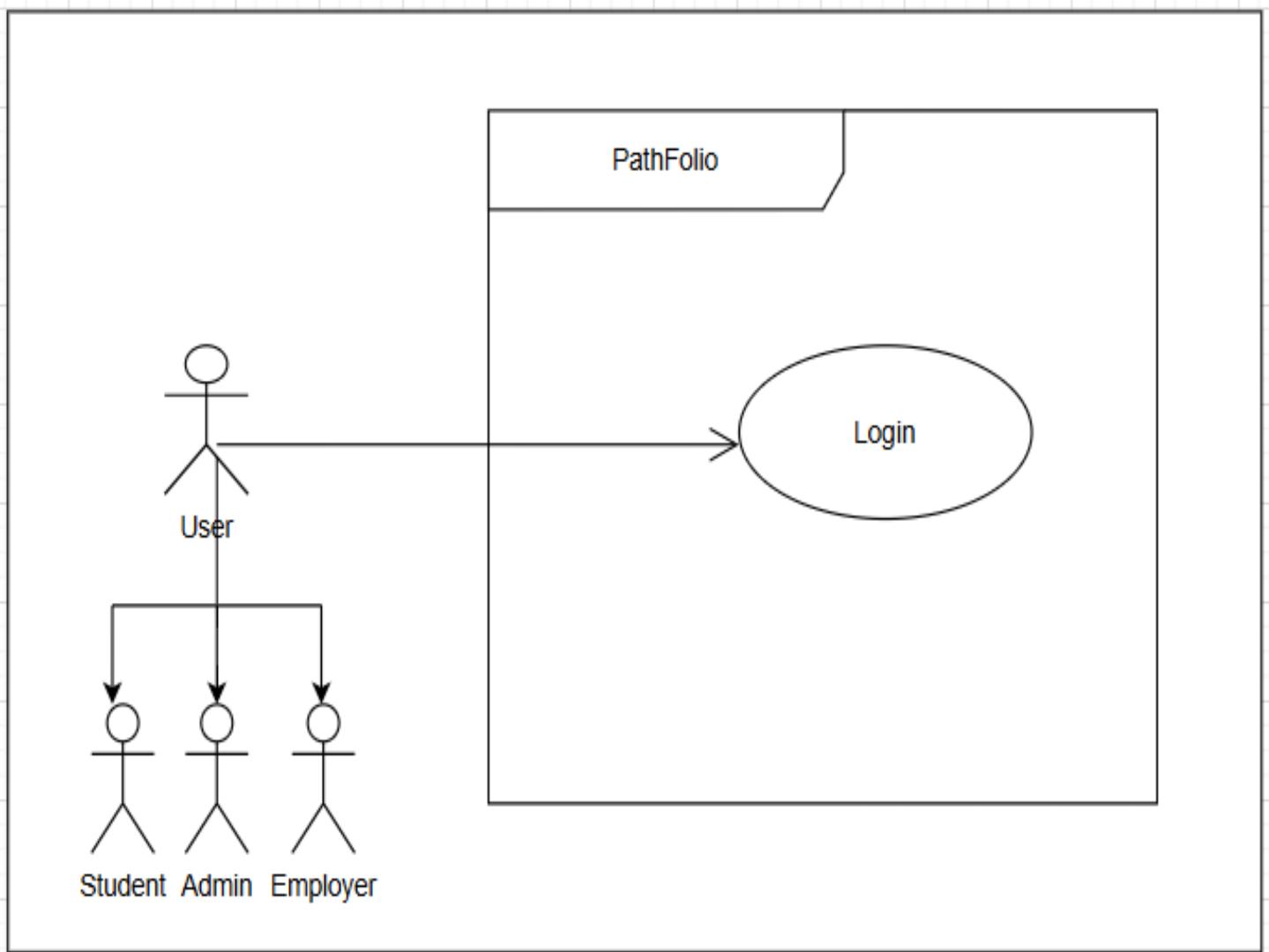


Figure 5: Login UML

Step-by-step

Step 1: Accessing the Login page

1.1 The user opens the Pathfolio system, then they are presented with the login page

Step 2: Entering the credentials

2.1 The system prompts the user to enter their email address and password on the textboxes provided.

The credentials provided by the user should match the ones in the database embedded in the Pathfolio system.

Step 3: Validation of the credentials.

3.1 The user enters their email address and password on the provided textboxes.

3.2 The user clicks the “login” button to try gaining access to the system.

3.3 The system checks if the fields' standards defined (non-empty and error formatting) are met and prompts the user to the fields if improperly entered.

3.4 Once the user has corrected the fields, the system compares the input of the user to the one in the database embedded in the Pathfolio system.

3.4.1 If the provided credentials are correct, the user is granted access to the system.

3.4.2 If the provided credentials are incorrect, the system will display an error message notifying the user that login attempt was unsuccessful.

Remarks (about the system)

Handling Incorrect credentials

1.1 When the user enters incorrect credentials, the system increments the login attempt counter for that user.

1.2 The system displays an error message to inform the user of the incorrect login details "Incorrect details entered"

1.3 Clear the login fields for another attempt.

First Lookout after three more unsuccessful attempts

If the user fails to make a login attempt, the system repeats the process as follows:

1.1 The system locks the user's account temporarily.

1.2 The system sets the timestamp for when the lockout occurred.

1.3 The system displays a message to prompt the user to change the password.

1.4 The system disables the login form for 30 seconds.

Second Lookout after three more unsuccessful attempts.

If the user fails five more login attempts (after the initial temporary lockout is filled), the system repeats the process as follows:

1.1 The system locks the user's account again.

1.2 The system displays a message to prompt the user to change their password.

- "Contact support for assistance"

1.3 The system disables the login form indefinitely until the password is changed.

Temporary Lockout

1.1 During the 30 second lockout, the system prevents the user from entering the credentials.

1.2 The system displays a countdown timer to inform the user of the remaining time.

1.3 After 30 seconds, the system unlocks the user's account and reset the login attempt counter to zero.

Forgot Password Feature

- If a user has forgotten their password, they can click on the "forgot password" link found on the login page.

1.1 The user navigates and clicks the "forgot password" button.

1.2 The system redirects the user to a forgotten password page.

1.3 On the forgot password page, the system presents a textbox. Along with the label text that says:
“Please provide the email address you used for registration.”

Info Tip []:

- If the user has changed their email address contact support.

1.4 The user enters their email address in the textbox.

1.5 The system validates the email address:

1.5.1 If the email address is correct:

1.5.1.1 The system sends a unique secret code to the user’s email address

1.5.2 If the email address is incorrect

1.5.2.1 The system displays an error message “The email address entered is incorrect and does not match the one in the database”

Info Tip []:

- Go back and check if the email address entered is correct.
- If you are using a different email address, contact support.

1.6 The system will present the user with a textbox. Along with the label text that says: “Enter your unique code”.

1.7 The user enters the unique code they received on their email address.

1.8 The system verifies the code.

1.8.1 If the code is correct, the system presents two password fields for input.

- Info Tip []: Password Standards.
 - Length: At least 12 characters long
 - Cases: At least 1 uppercase and lower letter.
 - Symbol: Must contain at least 1 special character (e.g., !@#\$%).
 - Number: Must contain at least 1 numeric number (e.g. 0-9)
 - **Note 1:** Passwords should be unique and used for only one account.
 - **Note 2:** Avoid dictionary words.
 - **Note 3:** Avoid previously used passwords.

1.8.2 if the code is incorrect:

1.8.2.1 The system will prompt the user to enter the code again.

- If there are 3 failed attempts, the system will lock the textbox and highlight the button to “Send Unique Code”.
- If the user fails the second time, the user will be promoted to the contact support. The textbox and “Send Unique Code” button will be disabled.

1.9 The system automatically places the cursor in the first password textbox failed.

1.10 The user enters their new password in the first textbox.

1.11 The system validates the password against the set standards.

1.11.1 If the password meets the standards:

- The system proceeds to the following step 1.12.

1.11.2 If the password does not meet the standards:

- The system clears the first password field and prompts the user to try again, step 1.10.

1.12 The system enables the 2nd textbox then places the cursor in the textbox without user interaction. Along with the text that says: “Confirm password”.

1.13 The user re-enters their new password in the second textbox.

1.14 The user navigates and clicks the “Reset Password” button.

1.15 The system validates the password in textboxes 1 and 2 respectively

1.15.1 If the passwords match:

- 1.15.1.1 The system proceeds to step 1.16
- 1.15.2 If the passwords do not match:
 - 1.15.2.1 The system will prompt the user back to the textboxes; both the textboxes will be surrounded with a red highlight. Along with a message that says: "The passwords do not match".
 - 1.15.2.2 The user fixes the error.
- 1.16 The system presents a message to the user "Your password has been successfully changed".
- 1.17 The system redirects to the login page.

Step 4: Handling system idle state

- 4.1 If the system detects inactivity for 10 minutes.
- 4.2 The system triggers a pop-up message with a 30 second timer countdown, with a "Close" button. Along with message that says: "**Session is about to expire**": Your session appears to be inactive. Click the "Close" button before the timer gets to zero to renew your session to your session will end.
- 4.3 The user clicks the close button:
 - 4.3.1 The system triggers the pop-up message to close. The user is returned to the page they were previously in.
 - 4.3.2 No input from the user.
 - 4.3.2.1 The counter gets to zero then the system ends the session.
 - 4.3.2.2 The system then redirects to the login page.

Logout

Disclaimer: In real life the logout functionality would be an independent functionality, but for the scope of this project it is covered under login.

Step 1: Scanning the sidebar

- 1.1 The user is already logged into the Pathfolio system.

Step 2: Clicking the logout button

- 1.2 The user navigates the main sidebar and clicks the "logout" button.

Step 3: Confirming Logout

- 3.1 The system displays a confirmation pop-up message box with the following text "Are you sure you want to log out".
 - 3.1.1.The user clicks "Yes" button.
 - 3.1.1.1. The system logs the user out of the system.
 - 3.1.1.2. The system terminates the session and redirects the user to the login page.
 - 3.1.2.The user clicks "No" button:
 - 3.1.2.1. The system cancels the logout process.
 - 3.1.2.2. The user remains in the system(logged-in), the confirmation pop-up message box closes and is returned to the last page.

Best case Scenario

1. John lands on the login page of the system.
2. John then enters their email address and password, which they used when they registered correctly.
3. John navigates to the login button and clicks it, and the system validates the credentials.
4. John access to the system is granted.

Worst case scenario

1. Doe lands on the login page of the Pathfolio system.
2. Doe enters his credentials (email address and password) and clicks the Login button.
3. The system compares the credentials, finds them incorrect, and displays an error message. The login field(textboxes) is cleared for a new input.
4. Doe re-enters the credentials, but the system still rejects them. Another error message appears; the textboxes are cleared again.
5. On her third attempt, the system still does not grant access. This time the system temporarily locks her account for 30 seconds. A message box alerts her of the lockout, and a countdown timer appears on the screen.
6. After the 30 second lockout, the system unlocks her account and reset the login attempt counter, allowing her to try again.

Alternative 1

1. Jay is presented with the login page of the system.
2. Jay then enters his email address and password.
3. The system reports incorrect details inserted, Jay then realizes he forgot his password.
4. Jay skims through the login page and sees a “Forgot Password” link and clicks it. The system redirects him to the password recovery page.
5. On the Forgot Password page, Jay promptly follows the on-screen instructions to reset password. The process involved an email verification.
6. After successful completion of the required steps, Jay can reset his password.
7. With his newly constructed password, Jay returns to the login page and inputs his email address and the password. This time, the system grants access to the homepage.

Alternative 2

- 4.1 Andrehette navigates the login page and clicks the “Logout” button on the main sidebar.
- 4.2 A confirmation pop-up message box appears and asking a whether she is sure she wants to logout of the system, with the “Yes” and “No” buttons clearly highlighted.
- 4.3 Andrehette clicks “Yes” button on the pop-up message box.
- 4.4 The system processes the input and redirects it to the login page.

Disclaimer!!!: The users must be logged into the system to be able to use this functionality.

Disclaimer: This feature is the default landing page after a successful login. Each user role is assigned a predefined sidebar containing role-specific features.

Student Features

Professional Profile Management.

Description

This feature allows students to interact with the Pathfolio system for creating, updating, and maintaining their professional profile for employer viewing.

Disclaimer: The Employer Profile Management feature uses the exact same interface and process as Employer Registration. No separate breakdown is provided to avoid repeating identical functionality.

Profile Management UI

The screenshot shows the 'Profile Management' section of the Pathfolio application. On the left, a sidebar menu lists 'Pathfolio' sections: Dashboard, Profile Management (which is selected and highlighted in blue), Academic Integration, CV Builder, Job Matching, Skills Tags and Badging, and Sign Out. The main content area has a dark background with light-colored input fields and buttons. At the top right are 'Edit' and 'Save' buttons. The 'Profile Management' header includes the sub-instruction 'Manage your personal details, achievements, skills, and education.' Below the header, there are fields for First name (Jane), Last name (Doe), Email (jane.doe@student.edu), and Phone (+27 82 505 4500). A placeholder 'No photo' is shown with a 'Change Photo' button and a note about supported file types (JPG, PNG, Max size 5 MB). A 'Bio' field contains the text 'Computer science student focusing on software engineering.' In the 'Achievements & Interests' section, there's a text input field with 'Add achievement/interest (press Enter)' placeholder and an 'Add' button. Below it is a list item 'Dean's List 2023'. In the 'Skills' section, there's a text input field with 'Add skill (press Enter)' placeholder, an 'Add' button, and a 'Upload Certifications' button. A note at the bottom states 'Supported cert types: PDF, DOCX. Max size per file 5 MB.' The overall design is clean and modern, using a dark color palette with blue highlights for active links.

Figure 6: Profile Management UI

Profile Management UML

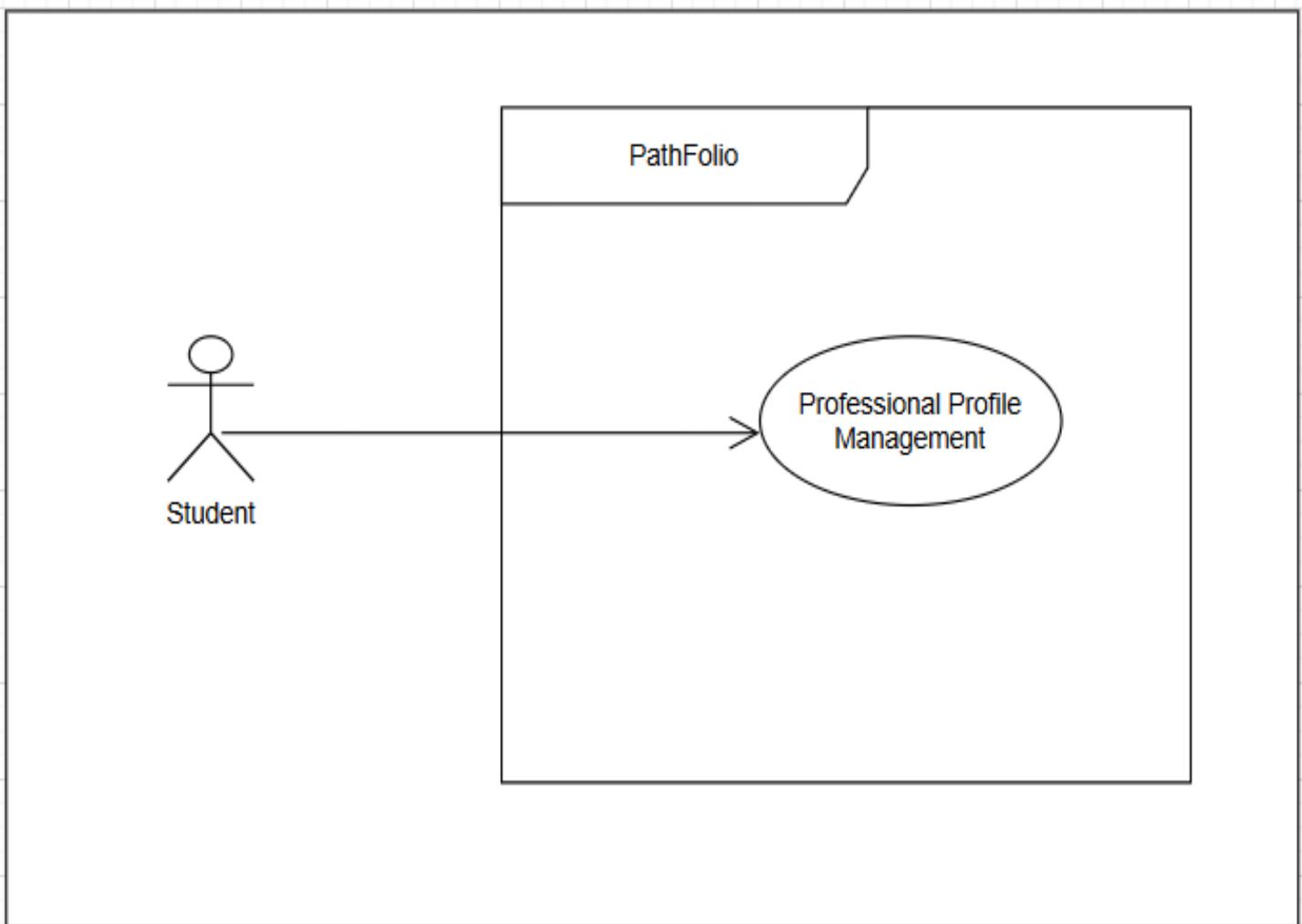


Figure 7: Profile Management UI

Step-by-step

Note!!! The user in this scope is student, User and Student will be used interchangeably.

Step 1: Accessing the Profile management

- 1.1 The student is granted access to the system, and the dashboard is presented to them.
- 1.2 There is a sidebar on the top right corner of the page, where there is a “Profile” link on.
 - 1.2.1 The user clicks on the Profile link on the side bar.
- 1.3 The user is redirected to the Profile Management interface.
- 1.4 The textboxes in this interface are filled with information from the database that the user used during registration.

Step 2: Editing the basic information

- 2.1 There is an “Edit” button at the top left corner of the interface.
 - The only way the user will be able to edit any of the fields on the interface, is if they click on this button.
- 2.2 The user then can move the cursor and point click on the textbox they wish to update.

Step 3: Adding Profile Details

3.1 Achievements and interest

3.1.1 The student clicks to the “Add Achievement” button on the interface.

3.1.2 The system presents an area for entering achievements

3.1.3 Once entered, the system displays the achievements as tags.

3.2 Skills management

3.2.1 The student clicks on the “Add Skills” button provided on the interface.

3.2.2 The system shows an area box for entering the skills (e.g. Microsoft Office)

3.2.3 The system presents an optional “Upload Certifications” button.

3.2.3.1 The student clicks on the “Upload Certifications” button.

3.2.3.2 The system opens the device’s file manager.

3.2.3.3 The student selects the desired file, and the file undergoes file validation.

3.2.4 Once the student has entered certificates, the system displays them as tags.

3.3 Education and Experience

3.3.1 The student clicks on the “Education” button and the system redirects them to the education and experience page.

3.3.1.1 In the page, the student can edit the institution, qualifications, and duration.

3.3.2 All the fields relating to the experience of the student are placed below the education fields, and the student navigates to them to make any desired changes.

3.3.3 The entries are stored and displayed in the Education and Experience section of the profile management page/interface.

Step 4: Upload a Profile Photo

4.1 The student clicks the “Profile Photo” button.

4.2 The system presents two options:

a. Take a photo using the device’s camera.

i. The system will ask for permission from the user, and once the user has permitted to use the camera. The student will then be able to take a snap using the device’s camera through the Pathfolio system.

b. Choose an image from the device’s storage.

i. The system accesses the devices gallery through the file manager.

4.3 The student selects the image they want to make their profile photo, and it’s previewed before saving.

Step 5: Saving Profile

5.1 The student clicks the “Save” button.

5.2 The system validates all completed fields and uploaded files.

a. If errors are found, the system highlights them and prompts correction

- 5.3 If all the inputs are valid, the system saves the profile and displays a confirmation popup message, that shows that all edited fields are saved.

Remarks

Handling Missing or Invalid Fields

Disclaimer!!!: If a required field (e.g. Email Address) is missing, the system prevents saving and highlights the field in red.

If the entered email address format is invalid (e.g. "user@"), the system displays a message "Please enter a valid email address".

Handling Duplicate Skill Entry

2.1 If the student tries to add the same skill twice, the system prevents duplication and displays "This skill has already been added".

Handling Profile Photo Errors

3.1 If the uploaded photo is in an unsupported format (e.g. txt file) the system displays: "Unsupported file format. Please upload a valid image (JPG, PNG)".

3.2 If the uploaded photo exceeds maximum file size limit, the system displays:
"File too large. Maximum allowed size is 5 mb."

Handling File Validation

4.1. If the uploaded file is an unsupported format (e.g. txt file) the system displays:
"Unsupported file format. Please upload a valid document (PDF, DOCX)"

4.2. If the uploaded file exceeds the maximum file size limit, the system displays:
"File too large. Maximum allowed size is 5MB.".

Confirmation and Updates

5.1. Each time the profile is updated, the system overwrites previous details with the new data.

5.2 The confirmation message always appears after a successful update.

Best case scenario

1. Amo logs in and is directed to the dashboard.
2. Amo clicks on the "Profile" link and is redirected to the Profile Management page.
3. The textboxes show the information she entered when registering.
4. Then she clicks on the "Edit" button for her to be able to interact with the interface.
5. She then clicks on the "Add Achievement" button, and an area box appears which is what she uses to enter the new achievements.
6. Amo clicks on the "Add skills" button, and the system shows area box which accepts user input. She then continues to enter the new skills in the provided area.

7. She is then prompted to attach a file to the new skill added. She then is prompted to file manager, to select a file. She continues to insert the file and is accepted.
8. She then clicks on the Profile Photo; to select an image, she is prompted to allow the system to access the camera of the device she is using. Then camera system is open for her to be able to take a picture. The picture is then accepted as profile photo once Amo specifies that they are content with the snap.
9. She then clicks on save, which shows a “loading” signage. Every entry is then saved and shown on the interface on Profile Management page.

Worst case Scenario

1. David logs into the system and clicks the “Profile” link on the sidebar.
2. He is redirected to the Profile Management interface with all fields pre-filled from registration.
3. David clicks the “Edit” button to make changes.
4. While updating skills, David mistakenly adds the same skill twice.
5. The system prevents the duplication and displays the message: *“This skill has already been added.”*
6. David tries to save the profile without correcting this, leaving him unsure why the save did not work.
7. David attempts to upload a profile photo in an unsupported format (e.g., a .txt file).
8. The system displays: *“Unsupported file format. Please upload a valid image (JPG, PNG).”*
9. David clicks Save without addressing the errors and is repeatedly prompted without knowing which fields need correction.
10. The profile is not saved, leaving David confused and unable to complete his profile update.

Alternative 1

1. Roche logs into the system and navigates to the Profile Management link.
2. Roche then clicks on the Profile Management link and is directed to the page for managing profile.
3. She then clicks on the “Edit” button.
4. She continues to click on the “Add Achievement” button, and she enters an achievement that is already in the system, the system shows a message box that reads “The Achievement has already been added”, and the skill is not entered.
5. She clicks on the “Add Skill” button, and she enters the relevant skills and then on the last entry, she enters a skill already entered. The system shows a message that reads “The Skill has Already been added”. And it is not entered.
6. She clicks on the “Profile Photo” button and she tries to enter an image from the device’s storage; she selects a file(snap) that is not in the correct format. The system displays a message that reads “Unsupported File Format, “Please upload a valid image (JPEG, JPG, PNG)”. She is redirected to the storage of the device again.
7. She clicks on the “Save” button once all image standards set in the system are satisfied. Then every entry saved is shown on the Profile Management page.

Application Portal

Description

The Application portal is designed to give students a direct and streamlined pathway to internships, part-time opportunities, and graduate programs. Through this feature, students can browse job postings, submit applications, and monitor their progress with a secure and user-friendly interface.

Application Portal UI

The screenshot shows the Pathfolio Application Portal interface. At the top, there's a navigation bar with a home icon, the text "Pathfolio - Application Portal", "Browse and apply for live opportunities", and links for "Profile" and "Logout". On the left, a sidebar menu includes "Dashboard", "Application" (selected), "Academic Integration", "CV Builder", "Job Matching and Notifications", and "Skills Tags and Badging". The main content area has two sections: "Available Opportunities" and "Profile Summary".

Available Opportunities:

- Software Intern** at Stark Industries (Deadline: 2026/11/10)
Skills: Python, Git
Status: Gicut 2024 | View Details | Apply
- Data Analyst Graduate Program** at Wayne Enterprises (Deadline: 2025/12/11)
Skills: Writing, SEO
Status: Gicut 2023 | View Details | Apply
- Part-time Graphic Designer** at Daily Bugle (Deadline: 2025/10/31)
Skills: Photoshop
Status: Gicut 2025 | View Details | Apply

Profile Summary:

- No photo
- Description: 1000.83
- Applications closed

At the bottom, there are buttons for "Upload Certifications" and "Download Profile (2MB)".

Figure 8: Application Portal UI

Application Portal UML

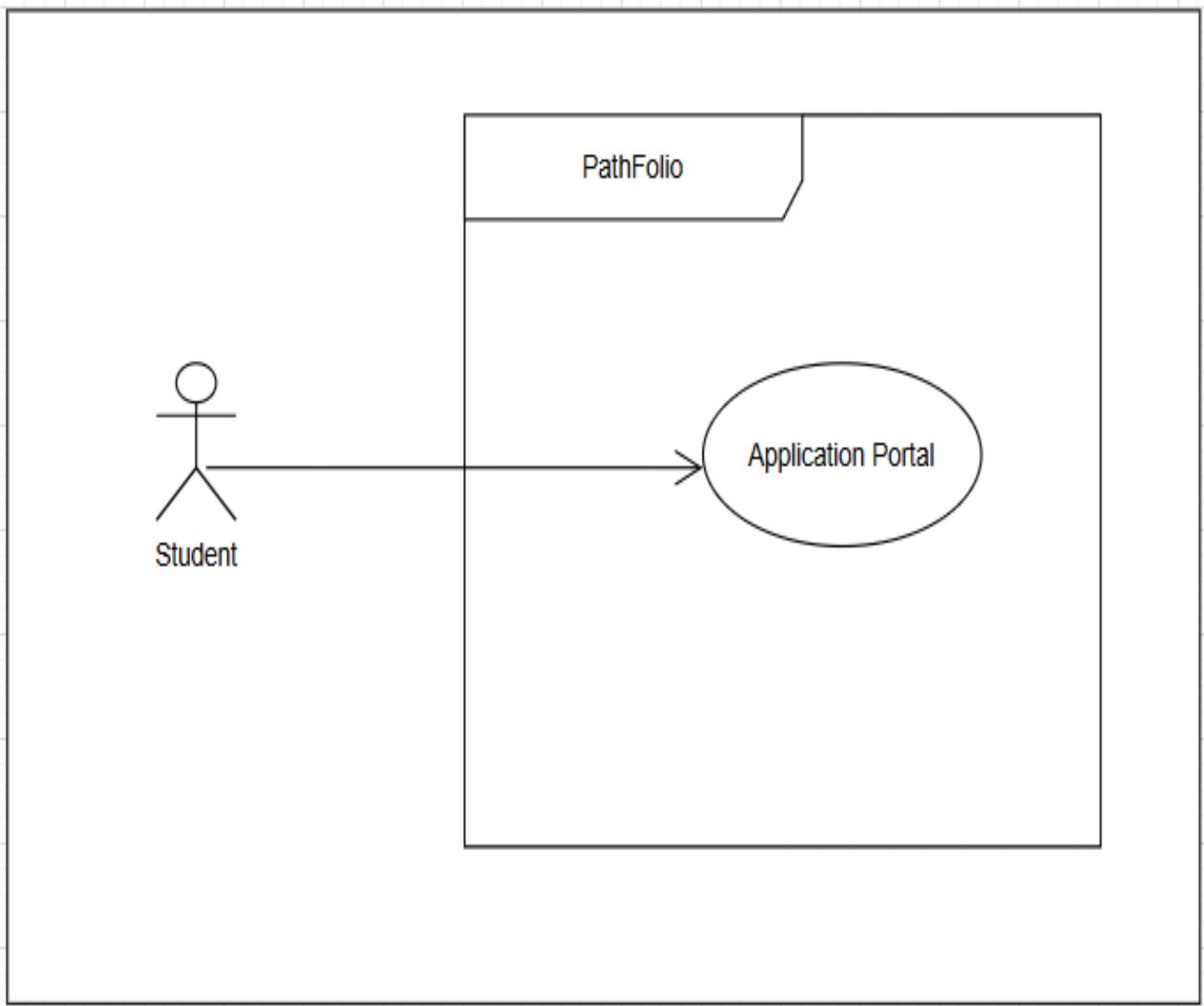


Figure 9: Application Portal UML

Step-by-step

Step 1: Accessing the Application Portal

1.1 The student logs into the system using credentials.

- The system loads the dashboard.
- The “Application” tab is visible on the main navigation sidebar.

Step 2: Browsing Available Jobs

2.1 The student views a list of available opportunities displayed in a grid format.

- Each job post includes title, employer name, type (internship/part-time/full-time), deadline, and a “View Details” button.

2.2 The student can filter opportunities by:

- Job type (dropdown: Internship, Graduate Program, part-time)
- Skills required.
- Graduation year or course background.

Step 3: Viewing Job Details

3.1 The student clicks the “View Details” button in a job post.

3.2 The system opens a dedicated job descriptions page containing.

- Job title
- Employer profile link.
- Detailed description.
- Requirements
- Application Deadline
- Application button.

Step 4: Initializing an application

4.1 The student clicks on the “Apply Now” button.

4.2 The system opens an application form pre-filled with data from the student’s profile and CV builder.

- **Editable fields:**
 - Cover letter
 - Additional Documents.
 - Contact details.
- **Non editable fields:**
 - Name
 - Student Number since it is auto filled from the UFS database.

Remarks

Handling Job Expiry

1. If the job application deadline has passed, the system disables the “Apply Now” button.
2. A label appears on the posting that reads “Applications closed”.

Tracking Updates

1. The system updates the application statuses automatically based on employer feedback.
2. Students receive a notification via the email address they used, when there are any new developments.
3. If the employer sends an offer, the system highlights the application in green with the label that reads “Offer read”.

Best Case Scenario

1. Naledi clicks the “Application Portal” button on the side bar.
2. The system loads the Application form with prefilled fields.
3. Naledi navigates to a bell icon in the top bar with a red badge showing new job posting.
4. She clicks on it and sees the latest job posting notifications.
5. She scrolls through skimming to see several relevant jobs.
6. She uses filter to refine search, thus clicking on the filter by button.
 - Category: IT
 - Location: Johannesburg
 - Job Type: Internship.

7. The system refreshes and shows matching results.
8. Naledi clicks on the “View Details” on a selected posting.
9. The system loads the job description, requirements and deadline.
10. Naledi clicks on the “Apply Now”.
11. The system verifies Naledi’s profile is complete.
12. Her profile and CV are automatically attached.
13. She clicks on the submit button.
14. The system displays a confirmation message that reads “Your application has been submitted successfully”.
15. Naledi clicks on the “My Applications” tab
16. A list of submitted applications is displayed with:
 - Job title,
 - Company
 - And date applied.
17. Application status is set to “Submitted”.
18. A progress tracker is showing the stages of the application.
19. The offer is highlighted in green with label “Offer made”.

Worst case scenario

1. Amo clicks Application Portal from the sidebar.
2. The system loads the portal interface.
3. Amo browses job postings and clicks View Details on a position.
4. Amo clicks Apply Now without having completed the profile.
5. The system blocks submission and displays a popup:
“You must complete your profile before applying. Redirecting to Profile Management.”
6. Amo is redirected to fill in missing details.

Alternative scenario 1

1. **David logs in** and navigates to the **Application** tab from the sidebar.
2. He sees a **long list of job postings** but does **not use any filters** to narrow down relevant roles.
3. He scrolls manually and **clicks “View Details”** on the first appealing job.
4. After reading the description, he selects **“Apply Now.”**
5. The **application form opens with pre-filled data**, but David **leaves the Cover Letter blank**.
6. He submits the application **without personalizing it**.
7. The system accepts the application, but it may be considered **generic and low effort by employers**.

Alternative scenario 2

1. David logs in and opens the Application Portal.
2. He sees a job posting labelled “Software Intern – ABC Corp” and immediately clicks “View Details.”
3. Excited, he presses “Apply Now,” but the system disables the button due to a passed deadline.
4. A label appears saying: “Applications Closed.”
5. David returns to the jobs list and continues browsing for active postings.

Academic Integration

Description

Secure integration with UFS databases for academic information sharing with granular privacy control.

Academic Integration UI

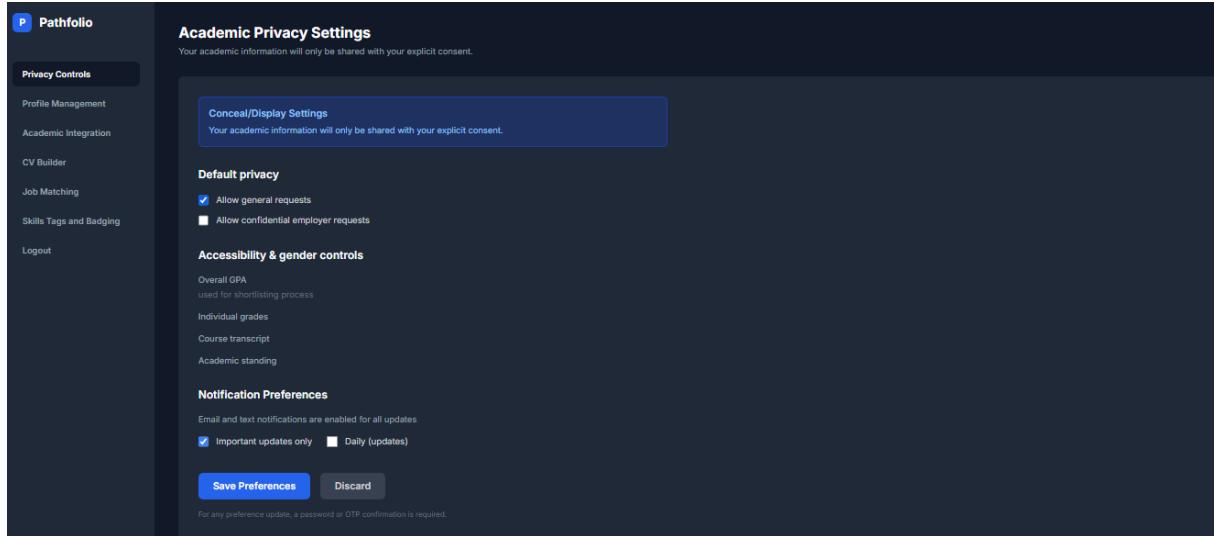


Figure 10: Academic Integration UI

Academic Integration UML

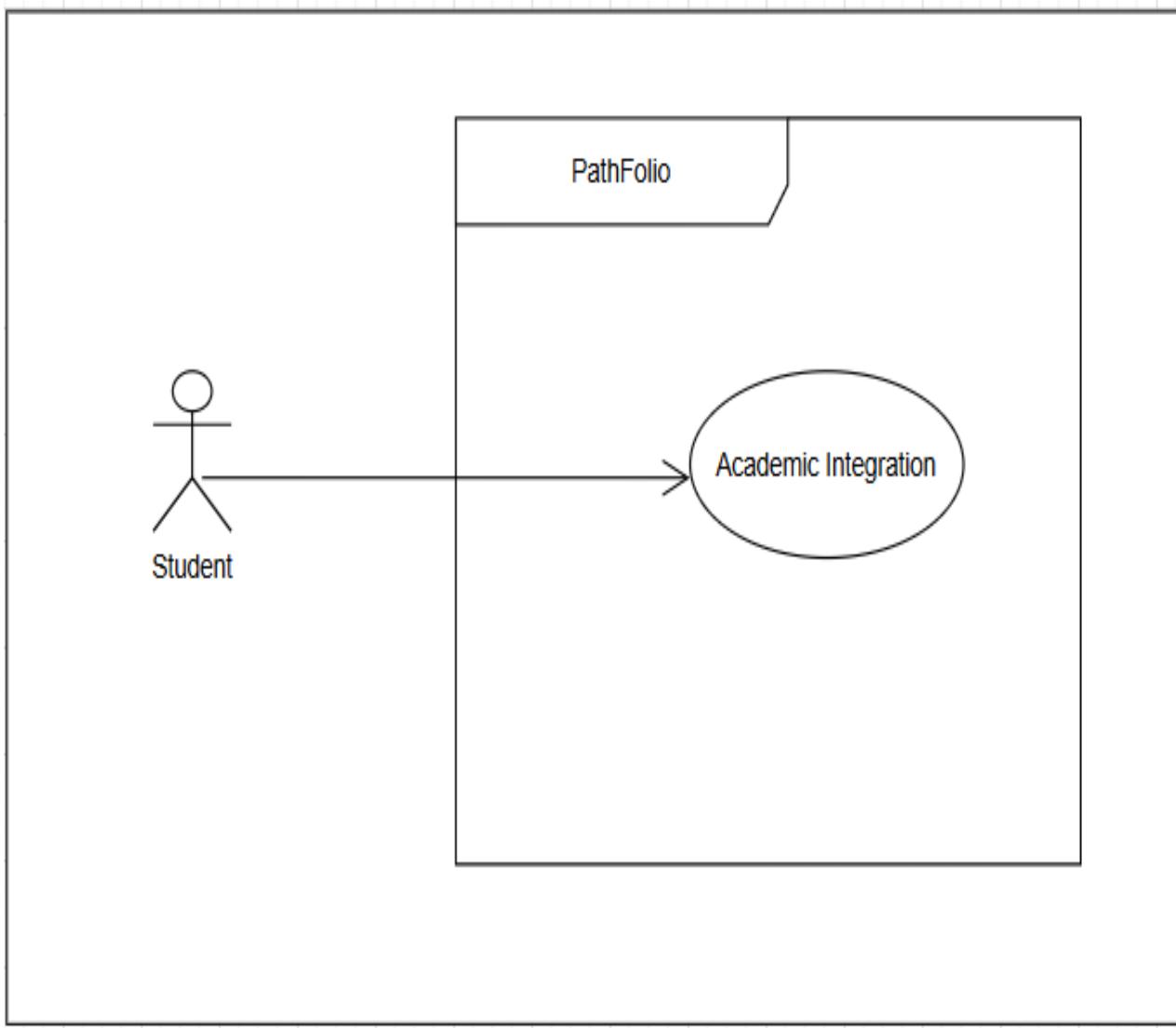


Figure 11: Academic Integration UML

Step-by-step

Step 1: Accessing Integration Settings

1.1 The student navigates to the “Profile” dashboard.

1.2 The student navigates to the “Academic Information” button and clicks on it.

1.2.1 Under the “Academic Information” button, they are shown current integration state by the system. This information is what is displayed below here.

- Not connected.
- Connected
- Error
- Pending Consent

1.2 The student clicks on “Start Integration” button.

Step 2: Authorization Setup.

2.1 The system displays a consent form that outlines precisely what will be requested and the reasons behind the requests. Permission groups are broken down into:

- Current semester marks
- Cumulative GPA
- Course enrolment history.
- Academic standing.
- Provide granular toggles so the student can opt-out of certain categories before authorization commences.
- Display legal and privacy summaries (e.g. retention period, purpose)

2.2 The student then clicks “Authorize Integration”, for which the button is disabled until they click the “I consent” checkbox.

Step 3: UFS Database Authentication

- 3.1 The system redirects to UFS secure authentication portal.
- 3.2 The student is prompted to enter UFS student credentials at the official UFS portal.
- 3.3 Then UFS validates against the student database.
- 3.4 On success, UFS system redirects back to the system (Pathfolio) with an authorization code and the original state.
- 3.5 The platform backend exchanges the code for an access token and exchange happens from server to server.

Best case scenario

1. Lebo logs in and navigates to her Profile dashboard.
2. She selects “Academic Information.” The system shows the status as “Not Connected.”
3. She clicks “Start Integration.”
4. A consent form appears with all permission groups (GPA, marks, enrolment history, academic standing).
5. Lebo carefully reviews and enables all toggles, then checks the “I Consent” box.
6. She clicks “Authorize Integration.”
7. She is redirected to the UFS secure authentication portal and enters her UFS credentials.
8. UFS validates the credentials and redirects back with authorization.
9. The system confirms: “Integration Successful — Status: Connected.”

Worst Case Scenario

1. Lebo opens Academic Information and sees “Start Integration.”
2. She opens the consent form but does not read anything and immediately clicks “Authorize Integration.”
3. The button is disabled because she forgot to check the “I Consent” box.
4. Frustrated, she keeps clicking repeatedly, thinking the system is broken.
5. Lebo abandons the setup halfway, assuming the feature is too complicated or unresponsive.

Alternative scenario 1

1. Lebo starts integration and opens the consent form.
2. She is comfortable sharing GPA and Academic Standing but disables “Current Semester Marks.”
3. She checks “I Consent” and clicks “Authorize Integration.”
4. She logs in via UFS authentication portal.

5. The system connects but displays “Status: Partially Connected — Limited Academic Data Accessible.”

Alternative scenario 2

1. Lebo proceeds to “Start Integration.”
2. She accepts the consent form and is redirected to UFS login.
3. She mistypes her student ID or password.
4. UFS portal displays “Invalid Credentials.”
5. Unable to continue, she closes the window, returning to Pathfolio showing “Pending Consent / Not Connected.”

CV Builder Tool

Description

AI-assisted CV creation using profile and academic data with professional templates.

CV Builder UI

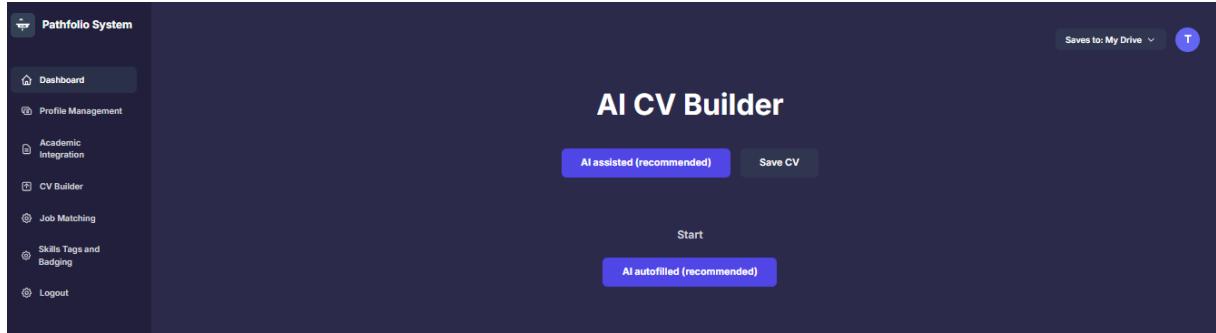


Figure 12: CV Builder UI

CV Builder UML

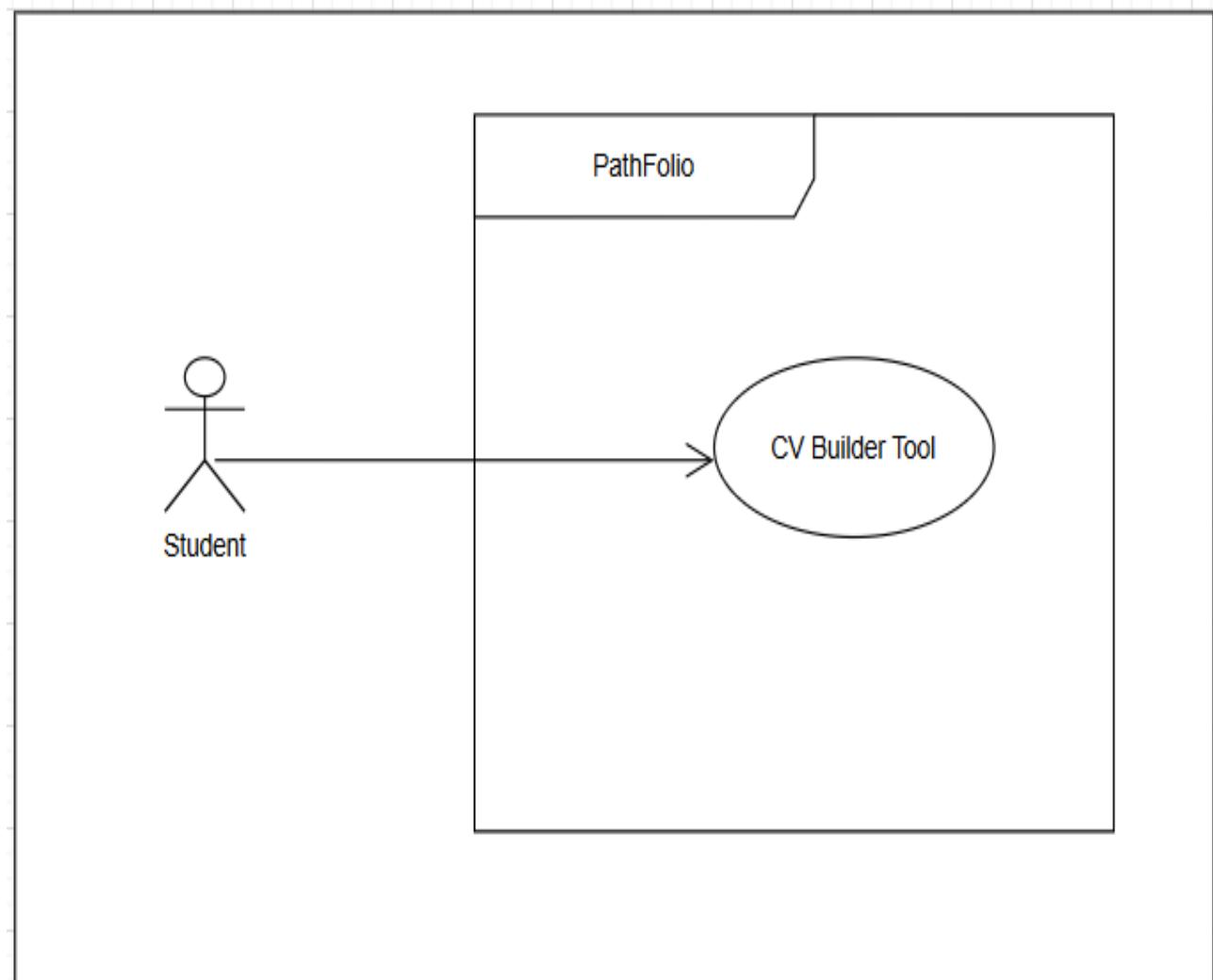


Figure 13: CV Builder UML

Step-by-step

Step 1: Accessing the CV Builder

1.1 The student logs into the system and navigates to the “*Career Tools*” section from the sidebar.

1.2 From the options, the student selects “AI CV Builder” for an assisted process or selects the “Create New CV” link for a blank start.

1.3 The system then loads the CV builder interface, displaying the available features like:

- Template selection
- Profile integration.
- AI suggestions

Step 2: Template Selection

2.1 The student is presented with categories of templates to choose depending on the need. The template selection includes:

- 2.1.1 Industry (e.g. Engineering, Business, Healthcare, etc.)
- 2.1.2 Experience Level (e.g. Graduate, Internship, Entry-level)
- 2.1.3 Design style may include modern, traditional and creative.

2.2 The student skims through the templates, clicks on a preview for a close look, and decides which layout best matches their career goals.

2.3 Once the student is satisfied, they continue to click the “Use this template” button and the chose design become the working canvas.

Step 3: Data Integration and Pre-population.

3.1 The system extracts existing data from the student’s profile and academic records.

- Auto-filled section will therefore include the following:
 - Name and contact details.
 - Academic background
 - Skills and competencies.
 - Work experience or internship details.
 - Portfolio or projects highlighted.

3.2 The student reviews the auto-filled draft and chooses which sections to keep, edit or exclude.

Step 4: AI-Assisted Content Enhancement

4.1 The AI-engine scans the auto-filled CV and makes suggestions tailored to the student’s field.

- 4.1.1 Generates a professional summary statement aligned with the student’s discipline or field.
- 4.1.2 Convert the academic projects of the student into achievement-driven bullet points.
- 4.1.3 Suggests skills keywords often used by employers in that industry.
- 4.1.4 Recommend industry-appropriate phrasing for professionalism and conciseness.

4.2 For each suggestion made by the AI-engine, the student has an option to accept, edit, or request alternative wording. This ensures that the CV feels more personalized and industry ready.

Step 5: Manual Content Addition

5.1 Students can freely add or update sections that the AI did not cover. Editable fields include:

- Personal statement / career objective
- Detailed work experience entries
- Extra skills not listed in the profile

- Certifications and special achievements
- References (optional)

5.2 The interface provides real-time formatting assistance, grammar checks, and spelling so that the CV remains polished.

Step 6: CV Optimization and Finalization.

6.1 Once the CV draft has been completed, the system runs a quality analysis:

- 6.1.1 Checks of the CV is the correct length (1 – 2 pages for graduates).
- 6.1.2 Optimizes keywords for Applicant Tracking System.
- 6.1.3 Ensures formatting and date consistency across sections,
- 6.1.4 Grammar and spelling are flagged.

6.2 The system will assign a quality score and displays optimization suggestions.

6.3 The student can then:

- 6.3.1 Preview the CV in PDF format and observe how it will look to employers.
- 6.3.2 Save it to their account for future service.
- 6.3.3 Download it in PDF or Word format.

Best Case Scenario

1. Lutu logs into the system and navigates to the “*Career Tools*” on the sidebar.
2. He clicks on the “*Create CV*”.
3. The system automatically populates the CV with profile data, academic record, skills tags and verified badges.
4. Lutu reviews the CV draft and makes minor changes to it. The click on the “edit” button on the interface to make necessary changes.
5. He continues to edit the necessary fields and previews the changes.
6. The system makes suggestions to Lutu using the integrated AI-engine it. Then he previews the suggestions and keeps them.
7. He downloads the polished CV in PDF format without errors.
8. His CV is now in a standard necessary for their specified fields.

Worst Case Scenario.

1. Vuyo logs into the system and navigates to the “*Career Tools*”.
2. He clicks on the “*Create CV*” button.
3. The system automatically populates the CV, but the data is incorrect.
4. Therefore Vuyo must manually edit the sections and verify that the entries are correct.
5. The system does not make the suggestions tailored to Vuyo’s discipline.
6. The CV is created with partial data, and certain achievements are missing, certainly weakening their job applications.

Alternative 1

1. Pete logs into the system and chooses the CV template suitable for a job he’s applying for.
2. The system makes suggestions based on the selected job type.
3. Pete accepts the suggestions and adds personal information on the template, and downloads the CV in multiple formats (PDF, WORD)
4. The resulting CV is tailored, complete and ready for immediate submission for job applications.

Alternative 2

1. Piet logs into the system and chooses a template suitable for the job application they are thinking of.

2. The system makes recommendations that are not relevant to Piet.
3. Piet struggles to manually adjust the CV and remove irrelevant content.
4. Certain achievements are omitted because he has not yet updated the skills tags or portfolio item.
5. The final CV is incomplete or is inconsistent, requiring additional manual labour before he can submit.

Job Matching and Notifications

Description

Automated job recommendations based on the student's profile with customizable notification preferences.

Job Matching and Notifications UI

The screenshot shows the 'Pathfolio System — Job Matching' interface. On the left, a sidebar menu includes 'Dashboard', 'Profile Management', 'Academic Integration', 'CV Builder', 'Job Matching' (which is selected), 'Skills Tags and Badging', and 'Logout'. The main area has tabs for 'Job Requirements' and 'Scan'. Under 'Job Requirements', there are sections for 'Company Size' (Startup, Mid-level, Large Corp, currently Startup), 'Relevant experience' (with a note about using at least one job post), 'Desired performance' (Top 1% candidate, hireable), and 'Company representation' (Mr. E Silica) and 'Desired representation' (Jane Doe/John). On the right, a 'Team' section shows 'Team Sigma' and a 'Job Seeker' section showing 'Step 1 of 4'. A progress bar indicates 'Matching generation based on jobs... (alpha version)'. At the bottom, a large blue button says 'Good matching'.

Figure 14: Job Matching and Notification UI

Job Matching and Notifications UML

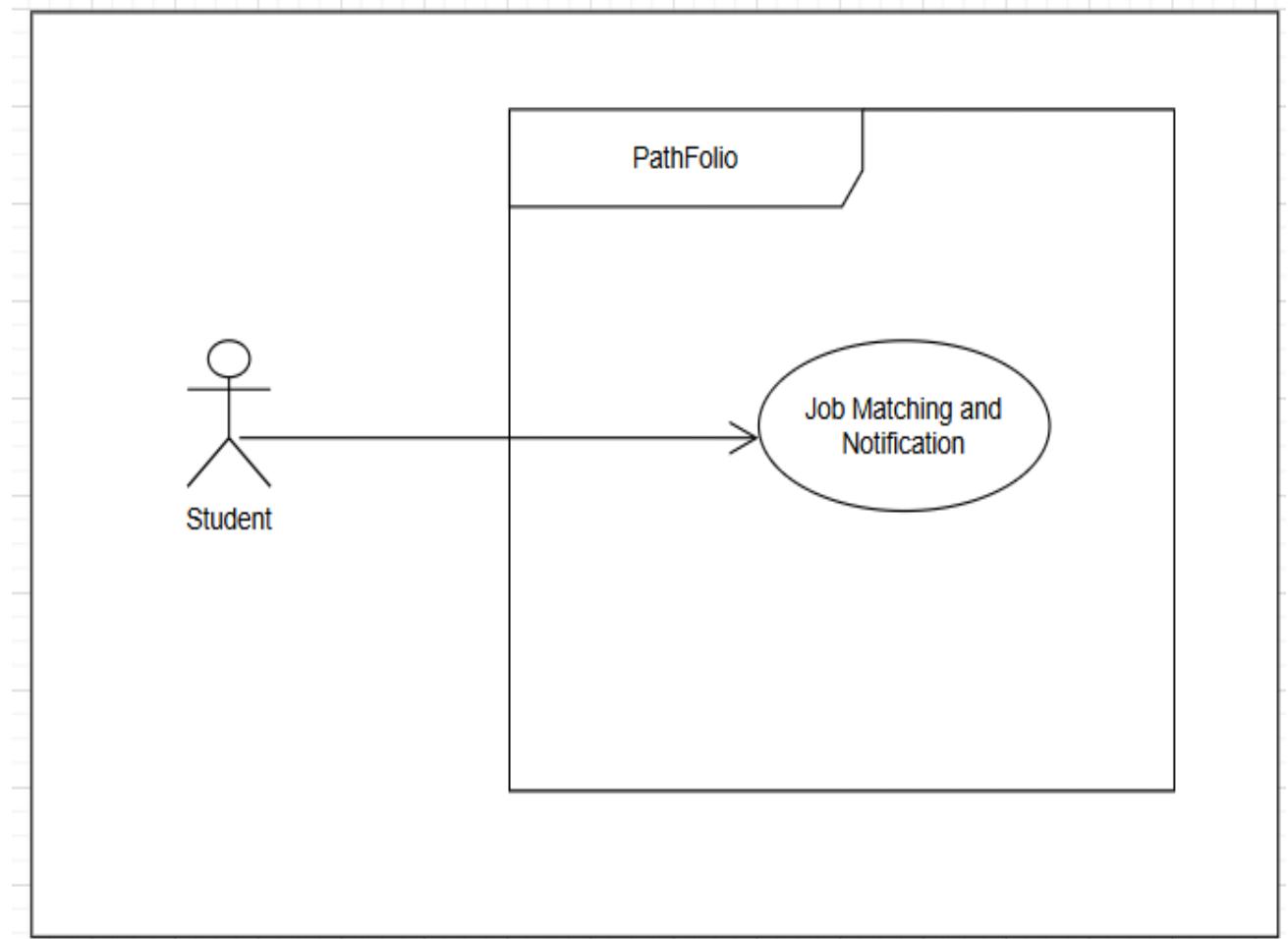


Figure 15: Job Matching and Notification UML

Step-by-step

Step 1: Profile Completion for Matching

- 1.1 The student accessed the “Job Preferences” from the dashboard sidebar.
- 1.2 The system displays the job completion form with required fields:
 - Preferred job types (Internship, Part-time, Graduate)
 - Industry preferences.
 - Location preferences.
 - Salary expectations.
- 1.3 Availability dates are displayed as well.
- 1.4 The student completes any missing or incomplete information. Then the system validates the entered data and highlights any missing information with a prompt that reads “*Please select at least one job type*”.

Step 2: Matching algorithm configuration.

- 2.1 The student sets preference options for the matching process:
 - Academic requirements strictness (e.g. Exact match / Flexible)
 - Skills Matching priority (e.g. Essential or Preferred)
 - Location search radius.
 - Company size preferences
 - Notification frequency (e.g. Immediate, Daily feed, Weekly summary)

2.2 The system saves these preferences and applies them in subsequent job matching.

Step 3: Initial job matching process

3.1 The system analyses the student profile for the following:

- Academic performance and program of study
- Skills and competencies
- Previous work or internship experience
- Portfolio projects
- Job preferences and criteria.

3.2 The system applies matching criteria to available job postings.

- Alignment with academic requirements
- Skills compatibility
- Location preference match
- Experience level requirements
- Industry focus alignment.

Step 4: Match results generation.

4.1 The system generates a ranked list with job postings that matches with the following details:

- 4.1.1 Compatibility score and match percentage for each job
- 4.1.2 Highlighted key matching criteria
- 4.1.3 Recommendation reasons (why this job suits the student)
- 4.1.4 Application deadlines

4.2 Quick application button are present for direct submission.

4.3 The student can filter or sort the job matches by:

- Match score.
- Job type.
- Posting date.

Step 5: Notification system.

5.1 The system sends notifications according to student preferences.

- Immediate: High job match (90%+).
- Weekly summary: Moderate matches.

5.2 The system displays deadline reminders for saved or shortlisted jobs.

5.3 Notifications are sent via:

- 5.3.1 In-built app notifications
- 5.3.2 Email alerts.
- 5.3.3 Mobile push notification.

5.4 The student can adjust notification setting at any time via the “*Notification Preference*” button.

Step 6: Continuous learning

6.1 The system tracks the student’s behaviour and feedback:

- Applications submitted.
- Job marked “*Not interested*”.

6.2 The matching algorithm embedded in the system adapts over time based on observed behaviour.

6.3 The student receives matching performance analytics and insights.

6.4 Periodically, the system prompts the student to update job preferences to improve future recommendations.

Best Case Scenario

1. Lesedi logs into the system and navigates to the “*Job Matching*” section.
2. The system accurately identifies suitable job type based on the data from his profile, skills, academic performance and preferences.
3. All the relevant job opportunities are displayed clearly with deadlines and application instructions.
4. Lesedi receives timely notifications and applies to several positions, with an increased chance of selection.

Worst Case Scenario.

1. Thabang logs into the system and navigates to the “*Job Matching*” section. Some suitable job postings are not shown.
2. The system sends notifications to him that are outside his interests.
3. Thabang must thereby manually scroll through the job posting in search of suitable postings.
4. There is a risk of missing deadlines for roles that would have been appropriate.

Alternative best-case scenario

1. Lerato logs into the system and navigates to the “*Job Matching*” section.
2. The system sends the notifications for high priority or recently posted positions that align with her interests.
3. She then filters the postings by application deadline.
4. Each recommendation has a tailored note explaining why the job is a match for her.
5. She applies promptly and receives interview invitations from multiple employers.

Alternative worst-case scenario

1. Thato’s profile lacks certain certifications, so the system misses matching him to ideal opportunities.
2. Notifications arrive in bulk, some of which are irrelevant which causes delay and confusions.
3. Thato must spend extra time verifying the jobs appropriateness which would cause him to miss deadlines.

Remarks:

1. Missing profile data prevents the algorithm from producing optimal matches.
2. Notification frequency can be changed anytime, and change take effect immediately.
3. Students can manually remove or hide job recommendations they do not want to see.
4. All analytics and preferences are student specific.

Skills Tags and Badging

Description

This feature allows students to showcase specific skills and competencies earned through coursework, projects, or extracurricular activities. Visual badges and skill tags are displayed on the student's profile, providing employers with a quick, verifiable way to assess practical abilities beyond academic grades. The system ensures that only validated skills are displayed, enhancing the credibility of the student's profile.

Skills and Badging UI

The screenshot shows a dark-themed user interface for managing skill badges. On the left, a sidebar menu includes 'Dashboard', 'Profile' (which is selected and highlighted in blue), 'Skills & Tags', and 'Example storage demo (sessionStorage)'. The main area is titled 'Your skills & badges' and contains three items:

- Microsoft Office** coursework - 2023/10/12
Proficient in Word and Excel
Source: Coursera
Status: Verified
Visibility: Visible to all employers
Actions: Edit, Delete
- Trellis Predictor (project)** - 2023/10/12
ML project predicting stock
Status: Unverified
- Debating Society** extracurricular - 2023/10/12
Active member and speaker
Source: Club
Status: Unvetted
Visibility: Visible to all employers
Actions: Edit, Delete

At the top right, there are 'Profile' and 'Logout' buttons. A note at the top right says 'Items marked **Vetted** are confirmed by verifiers'.

Figure 16: Skills and Badging UI

Skills and Badging UML

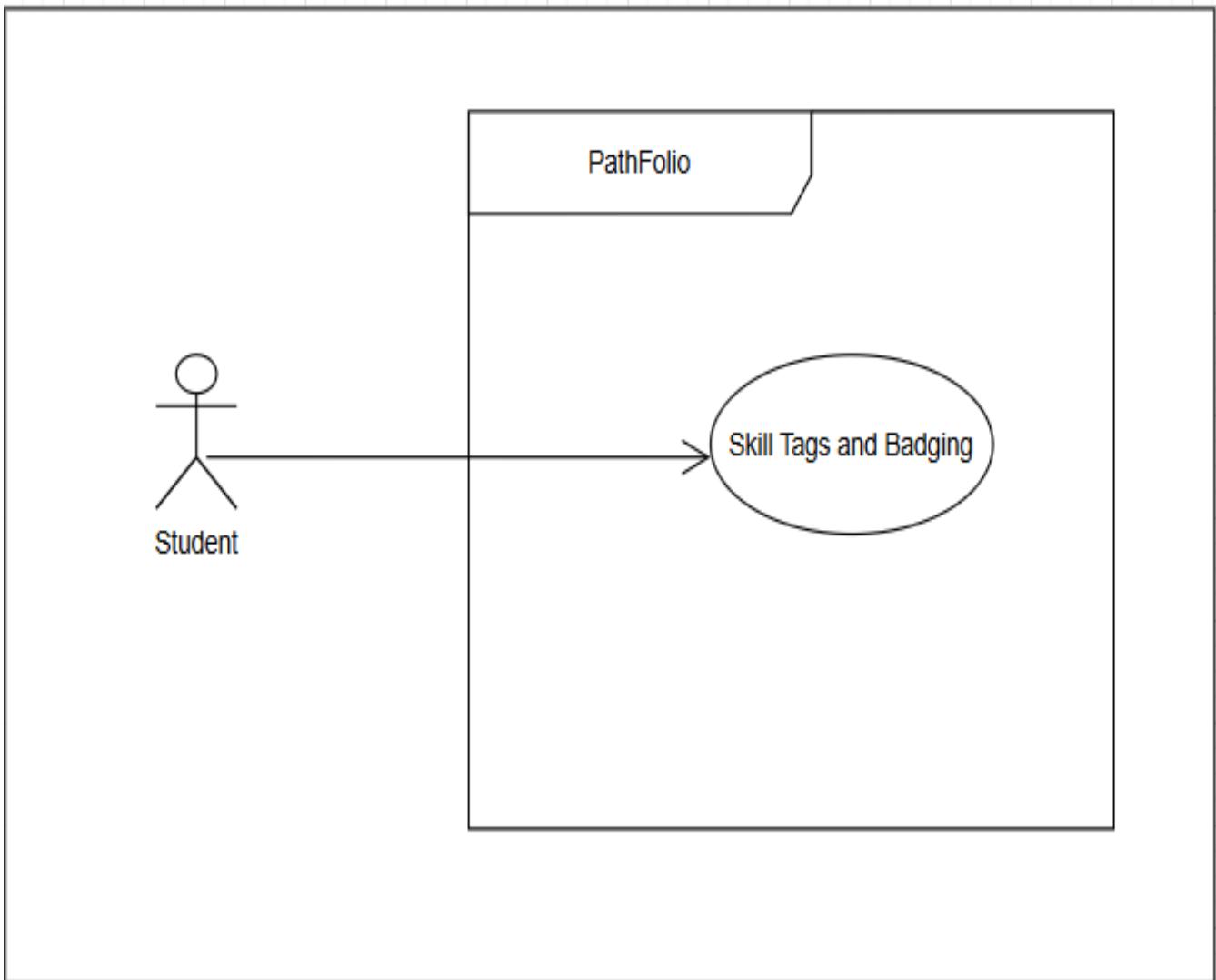


Figure 17: Skills Tag and Bagging UML

Step-by-step

Step 1: Accessing Skills tags and bagging.

- 1.1 The student logs into the system and dashboard loads where the “Skills and tags” is visible to them under the profile page.
- 1.2 If the student is on another page, they can click “Profile” tab from the sidebar and navigate to the “Skills and Tag” subsection.

Step 2: Viewing existing skills and badges.

- 2.1 The system displays all the skills and badges previously added or verified.

Badges may include:

- Academic
- Project-based.
- Extracurricular competencies.

- 2.2 Each badge has metadata showing the source, date earned and verification status.

- 2.3 The student can add a new skill or badge by clicking the “Add Skills/Badge” button.

2.4 Afterwards, the system will display a form that requires the following:

- Skill name
- Type (coursework, project, extracurricular)
- Description
- Evidence upload

2.5 The student fills the form and clicks “submit” button.

Step 4: Verification Process

- 4.1 The system relays the submitted form to the appropriate verifier (e.g. lecturer, administrator)
- 4.2 Once the submission has been verified, it is only this time it will appear on the profile of the student with a “Verified” label.

Step 5: Managing visibility

5.1 The student can toggle the visibility settings for each badge and skill.

5.2 Options for visibility includes:

- Visible to all employers
- Visible only to selected employers.

5.3 The changes are saved automatically or by clicking the “Save Changes” button.

Step 6: Removing or Updating Skills and Badges.

- 6.1 The student clicks on the “Edit” or “Delete” button next to the badge.
- 6.2 For editing, they can update descriptions or replace evidence files.
- 6.3 For deletion, the system displays a confirmation prompt that reads “*Are you sure you want to delete this skill/badge?*”.
- 6.4 Upon confirmation, the badge is removed from the profile.

Best Case scenario

1. Nthabeleng logs into the system and sees all previously earned badges correctly displayed.
2. She adds a new badge for a recent project, uploads the evidence files and submits it for verification.
3. The verifier approves it within a day, and the badge is immediately marked “Verified”.
4. She adjusts the visibility so that the potential employer can see the relevant skills.
5. Then she applies for a job, employers quickly identify her validated competencies.

Worst case scenario

1. Boitumelo logs into the system and notices some older badges showing are still pending verification because the verifier has not yet verified them.
2. He attempts to add a new skill, but their upload does not meet the verification requirements.
3. The badge remains in “*Pending Verification*”, so he can’t display it to the employers yet.
4. Only a portion of the skills are visible, limiting how well their profile reflects their actual capabilities.
5. He must wait for verification to complete before the employers can see the new competencies.

Alternative best-case scenario

1. Tasha logs into the system and finds that several new skills from a recent workshop and extracurricular activities have already been suggested by the system.
2. She quickly reviews, confirms ad submit these badges.

3. All submitted badges are verified within hours and she instantly sees them in the profile.
4. She enables visibility to specific job applications, creating highly targeted and impressive profile.

Alternative worst-case scenario.

1. Micheal logs into the system and sees that previously earned badges are still pending.
2. He attempts to add a new badge for a recent workshop and realized he can't since they no longer have a digital copy of the certification.
3. The badge cannot be submitted or verified.
4. Only the previously earned badges are visible to the employers, meaning some of the skills are not represented on the profile.

Employer Features

Mark Access

Description

Granular control system for sharing academic information with employers while maintaining privacy and compliance.

Mark Access UI

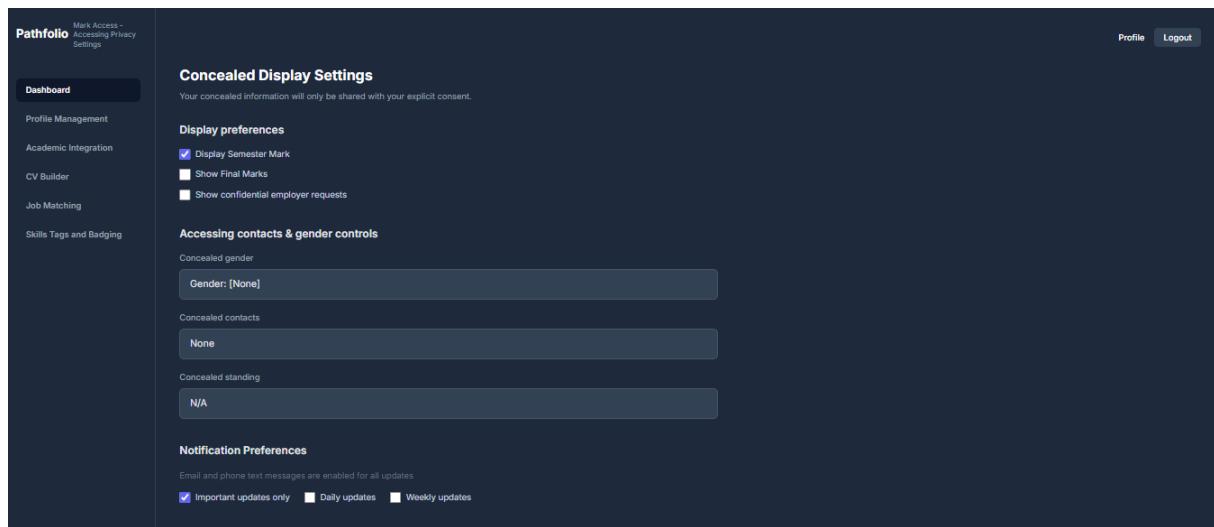


Figure 18: Mark Access UI

Mark Access UML

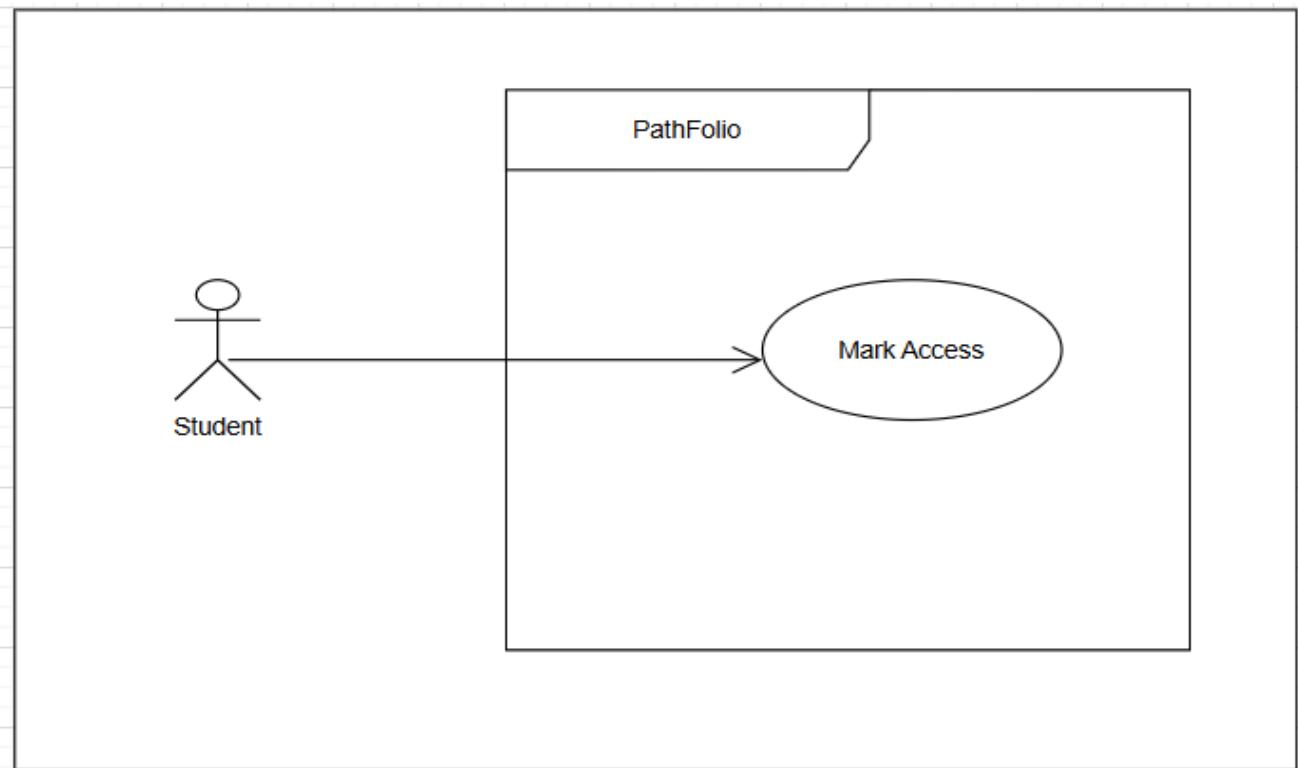


Figure 19: Mark Access UML

Step-by-step

Step 1: Academic Privacy settings access

1.1 The student logs into the system and navigates to the “*Privacy Controls*” button on the sidebar present on the dashboard.

1.2 Once the student has clicked on the “*Privacy Control*” button, they are redirected to a page with the following layout:

- Current sharing settings.
- Pending employer settings.
- Request history.

1.3 There is a highlighted sentence that is visible to the student on the page that reads “*Your academic information will only be shared with your explicit consent*”.

Step 2: Configure default sharing preferences.

2.1 The system displays the following default policy options which are checkboxes that can be marked by the student:

- “Never share automatically”
- “Allow general requests”
- “Pre-approve verified employers”.

2.2 The following academic information categories are presented:

- Overall GPA
- Individual grades
- Course transcripts
- Academic standing.

2.3 Each category has granular controls with explanatory tooltips.

Step 3: Employer request management.

3.1 The system displays a multi-channel notification which a student can choose from any of the following:

- In-platform notification with red badge
- Email notification with request details
- Optional SMS notification

3.2 The system will display the request details which will include:

- Employer name and company.
- Position applied for
- Specific information requested.
- Requested reason.

Step 4: Request processing and confirmation

4.1 The student navigates to the final authorization link on the page. Which will have the following checkboxes.

- Review sharing summary
- Legal consent checkbox
- Digital signature/PIN confirmation
- Click "Authorize Sharing"

4.2 The student confirms of successful sharing of data.

Step 5: Request History and analytics

5.1 The student can request historical request which will show a statistical summary of total requests, employer feedback integration.

Step 6: Privacy Audit trail

6.1 The system will display complete access logs.

6.2 The system again will show employer compliance tracking.

6.3 The data should be automatically deleted after expiry.

Best case scenario.

1. Emily (Student) clicks the Privacy Controls button on the sidebar.
2. She reviews the highlighted consent message.
3. Emily configures default sharing preferences to Pre-approve verified employers.
4. An employer request appears for a job position Emily applied for.
5. She reviews the request details, checks the legal consent and digital signature boxes, and clicks Authorize Sharing.
6. The system confirms that her academic information has been successfully shared.
7. Emily checks the request history and sees that the request is logged with analytics data.

Worst case scenario

1. Emily (Student) navigates to **Privacy Controls** on the sidebar.
2. She quickly selects **Allow general requests** without reading the tooltips.
3. Multiple employers send requests for detailed academic information.
4. Emily clicks **Authorize Sharing** without carefully reviewing the specific details requested.
5. The system shares more data than she intended.
6. Emily later realizes she has shared information with employers she did not want to, causing confusion and concern.

Alternative scenario 1

1. Emily (Student) clicks Privacy Controls on the sidebar.
2. She chooses Never share automatically as her default preference.
3. An employer request comes in for her overall GPA.
4. Emily ignores the request notification for several days.
5. The request remains pending in the system, and she forgets to review it, delaying potential interview opportunities.

Alternative scenario 2

1. Emily (Student) accesses Privacy Controls via the sidebar.
2. She decides to pre-approve verified employers but accidentally leaves one academic category unchecked (e.g., course transcripts).
3. An employer requests access to the unchecked category.
4. Emily clicks Authorize Sharing without realizing the missing category.
5. The employer only receives partial data, leading to incomplete evaluation.

Employer Registration

Description

The registration functionality process allows Employers to create accounts on the platform. Employers register by providing their organizational and contact information in order to post opportunities and access student profiles. Admins oversee the registration process, verify employer details where required, and perform additional administrative tasks related to managing registration data.

Disclaimer: The Student Registration feature is not implemented as a standalone process. Since the system integrates with the UFS student database, all student accounts are automatically sourced from existing university records — therefore, students are not required to manually register.

Employer Registration UI

The screenshot shows the 'Employer Registration' form within a Pathfolio application. The left sidebar contains navigation links: Privacy Controls, Profile, Academic Info, Skills & Badges, Applications, and Logout. The main form area has a title 'Employer Registration' and a note 'Form may be generated from CV/Resume or data may be auto-filled'. It includes fields for Organisation / Applicant*, Expected Remuneration (per year), Registered ID*, Base salary (w/o bonuses)*, Website, Incentives/commissions/bonuses as a % of b.s., Working arrangement*, Severance payout, Contact number*, Contact Person, Contact Email (work)*, and CV/Resume Drag 'n' Drop. There are also two accept terms checkboxes: 'I accept the T&C for this program*' and 'I accept company policy'. At the bottom are 'Submit' and 'Clear' buttons.

Pathfolio
Academic Privacy Settings

Privacy Controls

Profile

Academic Info

Skills & Badges

Applications

Logout

Employer Registration
Form may be generated from CV/Resume or data may be auto-filled

Organisation / Applicant * Expected Remuneration (per year)

Registered ID * Base salary (w/o bonuses) *

Website Incentives/commissions/bonuses as a % of b.s.

Working arrangement * Severance payout

Contact number * Contact Person

Contact Email (work) *

CV/Resume Drag 'n' Drop

Drop file here or click to upload

Organisation address

I accept the T&C for this program *

I accept company policy

Submit Clear

Figure 20: Employer Registration UI

Employer Registration UML

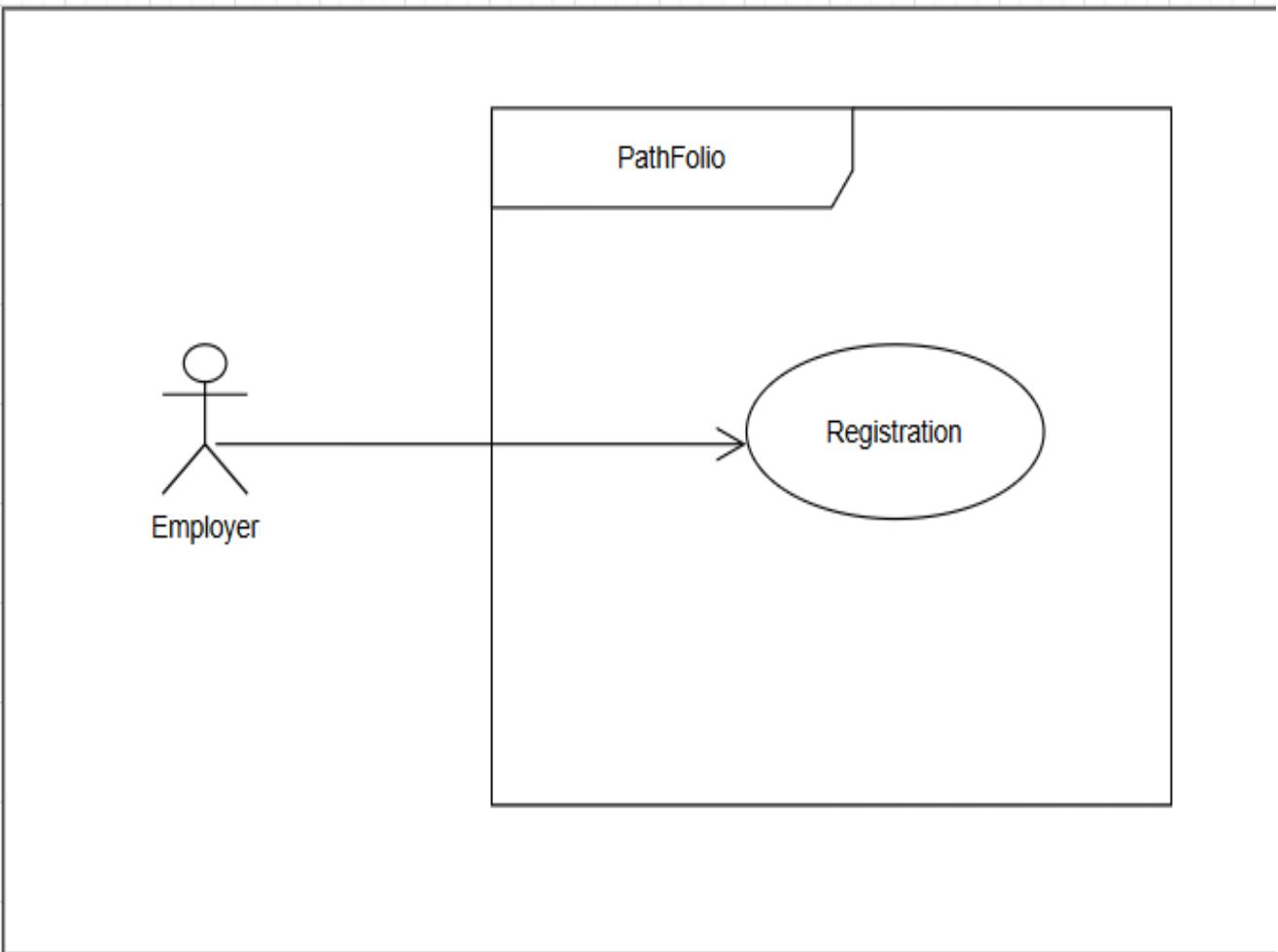


Figure 21: Employer Registration UML

Step by step

Step 1: Initiating Registration

- 1.1 The employer navigates to the system's public page and clicks the "Register" button.
- 1.3 The system redirects the employer to the "Employer Registration" page.
- 1.4 The "Employer Registration" page contains required fields for the organization, contact details and account login credentials.

Step 2: Collecting Information for the Employer

Disclaimer!!! All the fields marked with an asterisk(*) are mandatory and must be completed to proceed with the registration process.

- 2.1 The system will have a field for the organization's name, where the employer must provide legal name of the organization in the provided textbox. (*)
- 2.2 The page has a label that prompts the employer to enter the organization's registration number or VAT number. (*)
- 2.3 The employer selects the organization's industry from a dropdown menu. The dropdown menu will entail the following exemplary demonstration:
 - IT/Software Development.
 - Engineering/Manufacturing

- Education/Training.
- Health/Medical Services.
- Finance/Banking
- Retail and Other (opens a textbox for manual entry).

2.4 Primary Contact Name: (*)

- 2.4.1 The representative of the organization is prompted to enter their first and last name.

2.5 Contact email address: (*)

- 2.5.1 The representative of the organization must provide their email address.
- 2.5.2 If invalid, an error message appears.

2.6 Contact phone number: (*)

- 2.6.1 The representative of the organization provide phone numbers that will be used for any inquiries.
- 2.6.2 The phone number provided must be 10 digits, if fewer than 10 digit numbers are provided, an error message will be invoked.

2.7 Organisational Address: (*)

- 2.7.1 The employer enters their physical and postal address in the provided textbox under the corresponding label for address.

2.8 Organisational website

- 2.8.1 The employer provides a URL for their website (optional)
- 2.8.2 If the link is incorrect, the system invokes an error message.

2.9 Upload Verification Documents:

- 2.9.1 The employer uploads files such as the company's registration documents.
- 2.9.2 The files should be in a format that the system supports i.e. PDF, Word

2.10 The employer should create a password that conforms to the following rules:

- Info Tip []: Password Standards.
 - Length: At least 12 characters long
 - Cases: At least 1 uppercase and lower letter.
 - Symbol: Must contain at least 1 special character (e.g., !@#\$%).
 - Number: Must contain at least 1 numeric number (e.g. 0-9)
 - **Note 1:** Passwords should be unique and used for only one account.
 - **Note 2:** Avoid dictionary words.
 - **Note 3:** Avoid previously used passwords

2.11 Then they have to confirm the password, by re-entering the password they entered previously.

2.12 Terms and Conditions: (*)

- 2.12.1 The employer must click on the checkbox next to the terms and conditions label to agree to them.
- 2.12.2 If unchecked, the system disables the submit button.

2.13 *CAPCHA Verification

- 2.13.1 The employer completes a CAPCHA Verification to confirm that they are not a bot.

Step 3: Submission and Validation

- 3.1 Employer clicks on the “**Submit**” button.
- 3.2 The system validates all the required fields, email format, phone number length and password standards.
- 3.3 The system checks for duplication of entry.

- 3.4 If errors exist at this stage, inline messages are displayed and the employer must correct the.
- 3.5 If all validation passes, the system creates a pending employer account and shows confirmation that reads “Check your email to verify account”.
- 3.6 The verification email is sent with an activation link.
- 3.7 The employer clicks on the link, and the system marks the email as verified and sets the status to “Pending Admin Approval”.

Best case scenario

1. Vuyo clicks the “Register” button on the platform’s public page.
2. The system immediately loads the **Employer Registration** page with all required fields visible.
3. Vuyo selects **Employer** as the account type.
4. Vuyo completes all mandatory fields (*marked with an asterisk*) correctly:
5. Vuyo creates a password that meets all system requirements and confirms it correctly.
6. Vuyo checks the **Terms and Conditions** box and completes the CAPTCHA verification.
7. Vuyo clicks **Submit**, and the system validates all inputs.
8. The system confirms there are no duplicate accounts, creates the pending employer record, and displays a success message.
9. A verification email is sent to Vuyo’s contact email with an activation link.
10. Vuyo clicks the verification link; the system marks the email as verified and sets the account status to **Pending Admin Approval**.
11. An Admin reviews Vuyo’s registration, verifies all details, and **approves** the account.
12. The system sends an approval notification to Vuyo, and the account is now **Active**.
13. Vuyo logs in successfully and can access all employer features (post job opportunities, view student profiles, manage applications).

Worst case scenario

1. Vuyo clicks the “Register” button on the platform’s home page and the Employer Registration page loads successfully.
2. Vuyo begins entering all the required details but leaves the “Contact Email” field blank, which is a mandatory field.
3. Confident everything is correct, Vuyo clicks the Submit / Register button. Nothing happens.
4. Frustrated, Vuyo clicks the Register button multiple times, but the system does not provide a clear error message indicating which field is missing.
5. Vuyo attempts to continue filling other fields and clicks the Register button again, but the system still does not process the registration because the mandatory field remains empty.
6. After several attempts, Vuyo becomes confused and assumes the system is broken or unresponsive.
7. Eventually, Vuyo notices a small validation message appearing near the empty mandatory field, highlighting that it must be completed before submission.
8. Vuyo fills in the missing information, clicks Register, and the system finally accepts the registration and sends it for Admin approval.

Alternative best-case scenario

1. Vuyo clicks the “Register” button on the platform’s home page and opens the Employer Registration form.
2. Vuyo fills in all required details correctly, including organization name, contact details, and documents.
3. For the contact email, Vuyo enters a generic email address (e.g., vuyo123@gmail.com) instead of an official company domain.
4. The system attempts to validate the email and identifies that the domain does not match the organization name or is a free email provider (Gmail, Outlook, Yahoo).
5. The registration submission is blocked, and Vuyo sees a warning: “Please provide an official

- company email address matching your organization.”
6. Confused, Vuyo tries to submit again with the same email, thinking it’s a minor warning. The system still rejects it, preventing registration.
 7. Vuyo tries multiple workarounds, including changing the email slightly (vuyo123@company.com), but the system rejects it because it doesn’t exactly match the registered domain of the organization.
 8. Frustrated, Vuyo contacts support to clarify the issue. Support explains that only official company emails are allowed and asks him to obtain a valid email address.
 9. Vuyo finally updates the email to a valid official domain and successfully submits the registration. The system creates the account and sends a verification email to the corrected address.

Alternative Scenario 2

1. Naledi clicks the “Register” button on the platform’s public page, and the Employer Registration page loads instantly.
2. Naledi selects Employer as the account type and fills in all other mandatory details correctly.
3. When prompted to create a password, Naledi enters one that meets all the platform’s security requirements:
 - a. Minimum 8 characters
 - b. At least one uppercase letter
 - c. At least one lowercase letter
 - d. At least one number
 - e. At least one special character (e.g., @, #, \$)
4. The system provides real-time feedback, showing a green checkmark next to each requirement as it is met.
5. Naledi re-enters the password in the confirmation field, and the system immediately verifies that both entries match. A success message appears: “Passwords match.”
6. Naledi clicks Submit, and the system validates all fields, including the password, without errors.
7. A confirmation message appears: “Registration successful! A verification email has been sent to your official company email.”
8. Naledi clicks the verification link, the account is verified, and the admin approves the registration.
9. Naledi logs in successfully using her secure password and can access all employer features immediately.

Employer Deregistration

Description

The employer chooses to deactivate or permanently delete their account. This prevents access to the platform, removes job postings, and restricts candidate searches. If the employer has active job postings or pending candidate applications, the system may require administrative approval before completion.

Employer Deregistration UI

The screenshot shows the 'Employer Deregistration' page. On the left is a dark sidebar with a blue profile icon at the top, followed by navigation links: Dashboard, Employer Registration (which is bolded), Employer Deregistration, Advanced Searching and Filtering, Job Posting, Interview Scheduling, and Logout at the bottom. The main content area has a light background. At the top, it says 'Employer Deregistration' and 'Permanent deactivation can take up to 3 working days'. Below this are four input fields: 'Email Address (Work)' containing 'john.doe@abccompany.com', 'New Password (work)' with placeholder 'Enter new password', 'Country *' with a dropdown menu showing 'Select a country', and 'Reason *' with a text area placeholder 'Provide a reason for the action'. At the bottom are four buttons: 'Delete Account Permanently' (red), 'Delete User' (red), 'Deactivate User' (gray), and 'Save Now' (blue).

Figure 22: Employer Deregistration UI

Employer Deregistration UML

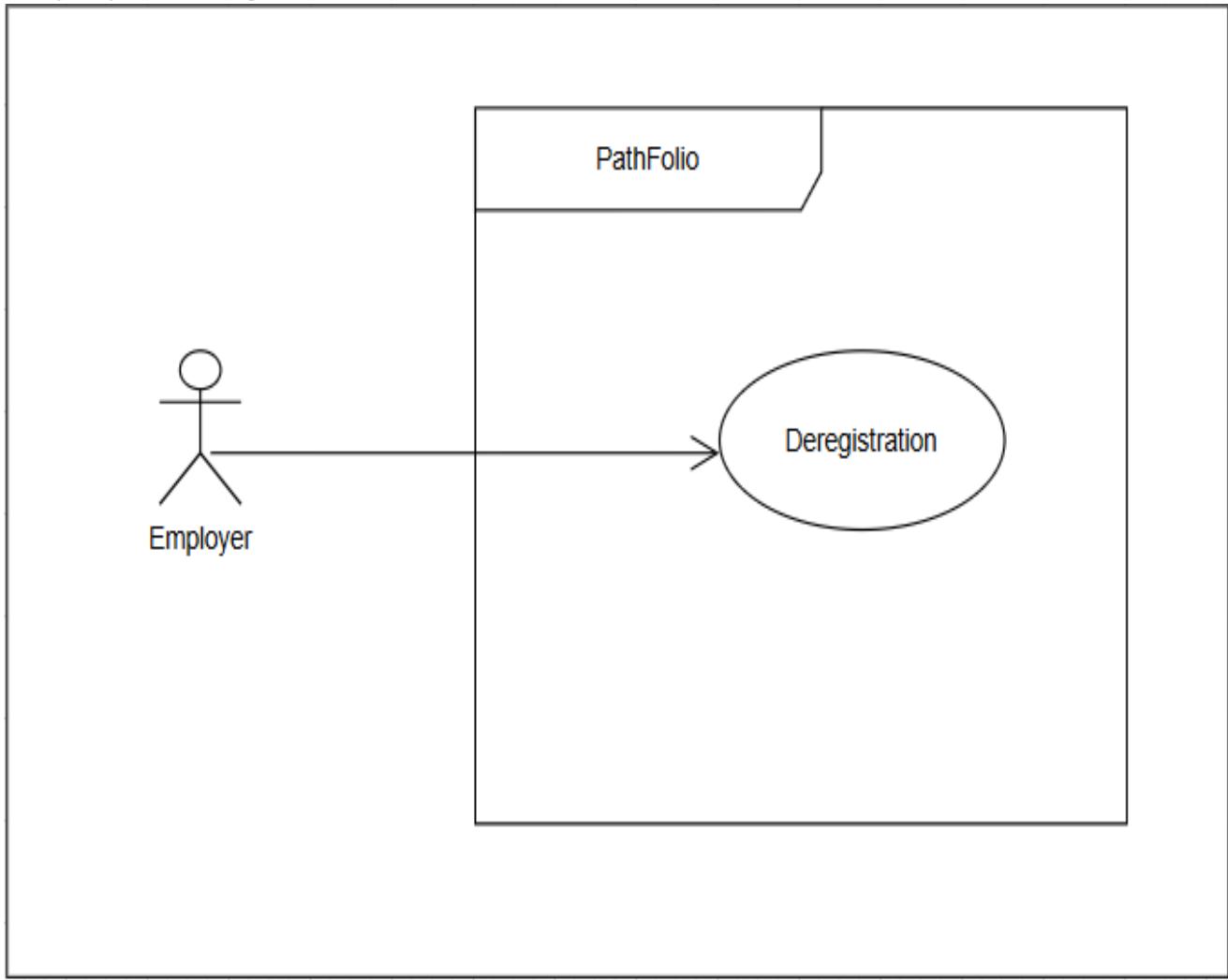


Figure 23: Employer Deregistration UML

Step-by-step

Step 1: Accessing the management section

- 1.1 The employer clicks on the sidebar on the top left corner of the UI.
- 1.2 The employer continues to click on the “Management”.
- 1.3 The system redirects to the “*Employer Management page*” which includes sections such as:
 - Company Profile
 - Job Posting
 - Analytics
 - Account Settings
- 1.4 The employer selects the “Account Settings” to manage their company registration.

Step 2: Initiate Deregistration

- 2.1 The system displays an option “*Deregister Company*” in the “*Account Settings*” page.
- 2.2 The employer clicks on the “*Deregister Company*” link.

Step 3: Confirmation of Deregistration

- 3.1 A confirmation pop-up message box appears after the employer has clicked on the “*Deregister Company*” button. The message reads “*Are you sure you want to deregister your company from the system? This action is irreversible and will remove all the job postings and company data*”.
- 3.2 Two buttons are presented: Yes (confirm) and No (cancel).
- 3.3 If the employer clicks “Yes”
 - The system marks the company account for deletion.
 - All active job postings are removed, and associated data is flagged for permanent deletion.

- A feedback message appears: “*Your company has been deregistered successfully.*”

3.4 If the Employer clicks **No:**

- The deregistration is canceled.
- A feedback message appears: “*Company deregistration has been cancelled.*”

Step 4: System Validation

- 4.1 The system validates that no critical processes will be disrupted.
- 4.2 If all the validations pass, the deregistration is finalized, and the employer is logged out.
- 4.3 The system redirects the employer to its public page confirming account removal.

Best case scenario.

1. Naledi (Employer) clicks the toggle bar, and the sidebar opens immediately.
2. She navigates to Management → Account Settings without delay.
3. The Deregister Company button is clearly visible and responsive.
4. Naledi clicks Deregister Company, and a confirmation pop-up appears instantly.
5. She clicks Yes, confirming the action.
6. The system validates that no critical processes are active, deletes all company data, removes job postings, and logs her out securely.
7. Naledi is redirected to the public page with a confirmation message: “Your company has been deregistered successfully.”

Worst case scenario

1. Naledi (Employer) clicks the toggle bar and opens the sidebar.
2. She navigates to Management → Account Settings and sees the Deregister Company button.
3. Naledi clicks Deregister Company, but she is unsure about the confirmation message and accidentally clicks No instead of Yes.
4. She thinks she has deregistered her company, but the system keeps the account active.
5. Naledi tries again, but this time she forgets to acknowledge that all job postings will be removed and cancelled midway.
6. Some postings remain active while she assumes the company is deregistered.
7. She needs to carefully read the messages and follow the steps properly to finally complete the deregistration success.

Advanced Searching and Filtering

Description

The Advanced Searching & Filtering feature enables employers to quickly identify and shortlist suitable student candidates by applying specific search criteria. Instead of browsing through all profiles, employers can refine results for candidates using filters such as GPA, skills, graduation year, or course background.

Advanced Searching and Filtering UI

The screenshot shows a dark-themed user interface for employer search. On the left is a sidebar with navigation links: Dashboard, Employer Registration, Employer Deregistration, Advanced Search and Filtering (which is highlighted), Job Postings, and Interview Scheduling. At the bottom of the sidebar is a Logout link. The main content area is titled 'Jobs' and has a sub-header 'Current search applicants'. It features a search bar with placeholder text 'Search by name, skills or keyword (e.g. Django, Zameels)'. Below the search bar are two input fields: 'e.g. John Smith' and 'e.g. Computer Science', followed by dropdown menus for 'Ambassador tier' and 'Availability'. To the right of these are 'Apply Filters' and 'Clear Filters' buttons. Another set of input fields below includes 'Org / Program based' and 'Enter string to be compared against', also with 'Apply Filters' and 'Clear Filters' buttons. The main results section displays three candidate profiles in cards:

- Sri V.** 20% match
To apply - please send an email to: sri.v@abccompany.com
Python, Rasa
View Shortlist
- Ahmed Tili** 23% match
test: sample text
gpa: 3.0, some certs - Sed, AWK
Python, Rasa, Go
View Shortlist
- Lil Wael** 20% match
short text about user bio
some gc - Django, text, ML
Python, JavaScript, VBA
View Shortlist

At the bottom of the results area, there is a link 'scroll down to view other search profile results'.

Figure 24: Advanced Searching and Filtering UI

Advanced Searching and Filtering UML

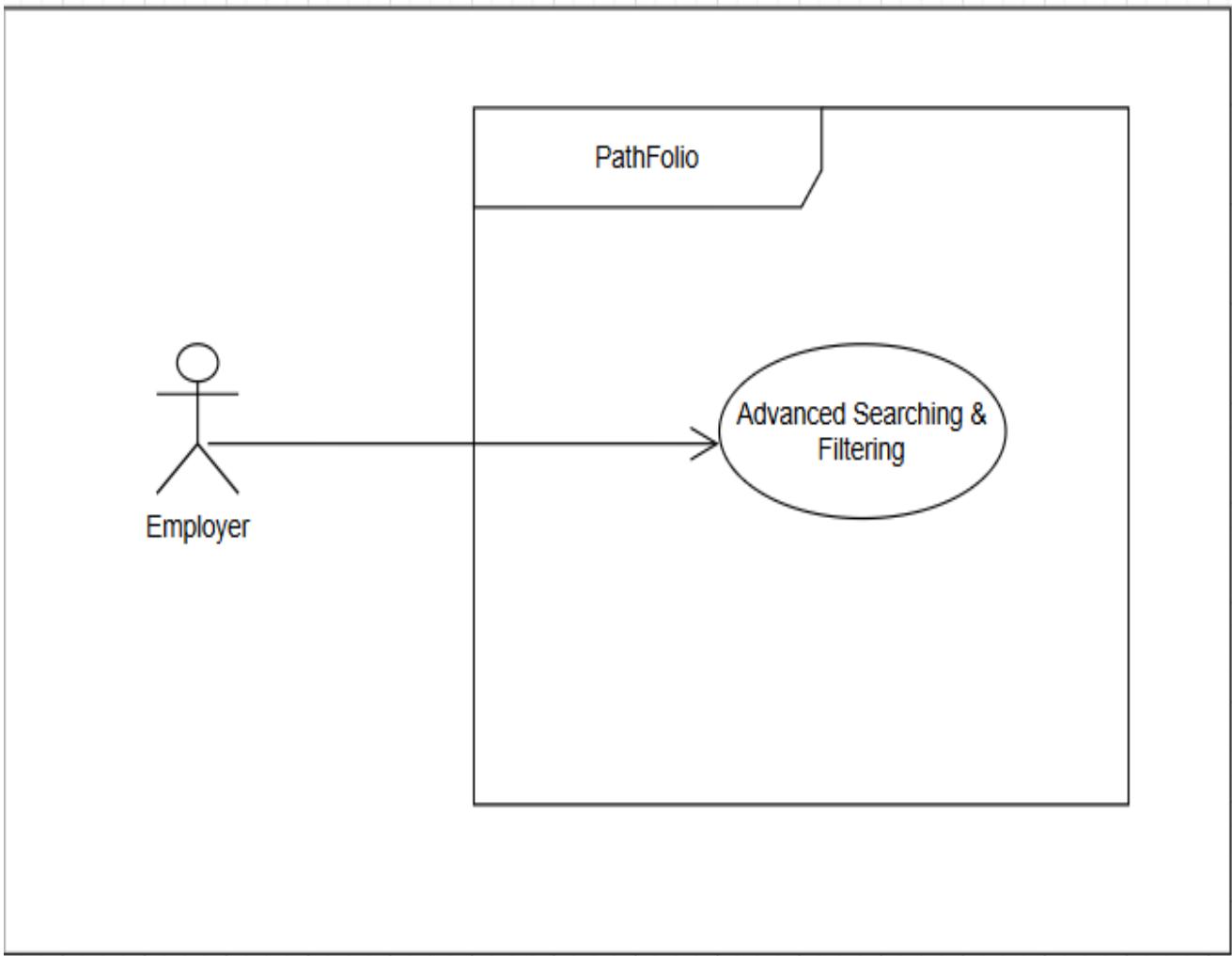


Figure 25: Advanced Searching and Filtering UML

Step-by-step

Step 1: Accessing the candidate's section

- 1.1 The employer logs into the system and is presented with the Dashboard.
- 1.2 From the dashboard, the employer clicks the sidebar in the top-left corner, and it expands.
- 1.3 The employer selects “*Candidates*” from the sidebar.
 - The system loads the “*Candidates*”, displaying a search bar at the top advanced filters directly below.

Step 2: Using the search bar

- 2.1 The employer enters a keyword in the search bar (e.g. “Computer Science” or “Java”).
 - The system searches across candidate profile (skills, course background, education).
 - Suggestion appears as the employer types.

Step 3: Applying Filters

- 3.1 The employer applies filters beneath the search bar to refine results:
 - Skills (e.g., Java, Python, Data Analysis)
 - Graduation Year (e.g. 2022, 2023)
 - Course background (e.g. Computer Science, Finance, marketing)
 - Additional Filters (Location, Work experience, availability if needed).

- 3.2 The employer clicks on the “Apply Filters” link.
- 3.3 The system updates the result list dynamically to show only candidates that match.
- 3.4 Active filters appear as removable chips below the search bar.

Step 4: Reviewing candidate results

- 4.1 The employer reviews the candidates results list, and each card shows

- Candidate name and short profile headline.
- Key skills
- GPA and course background.
- Education and graduation year.
- Location and availability.
- A match score (based on filters applied).

Step 5: Viewing a candidate profile

- 5.1 The employer clicks on the candidate card to view the Full Candidate Profile, including:

- CV and certificates.
- Achievements and interests.
- Education and work experience details
- Contact info.

Step 6: Taking actions on candidates.

- 6.1 From either the results list or profile, the employer may shortlist a candidate, save for later, send a message, export candidate data.

Step 7: Handling no results

- 7.1 If no candidate match is found the system displays a message that reads “No candidate found, try adjusting your APS, graduation year, or course filters.”

Step 8: Ending the session

- a. The employer may clear all filters or return to the Dashboard through the sidebar to end the session.

Best case scenario

1. Jacob logs into the system, is directed to the dashboard, opens the toggle bar, and selects “Candidates.” The system loads the Candidates interface with a search bar on top.
2. Jacob types “Computer Science” in the search bar, and the system displays matching candidates.
3. He applies filters: GPA 3.5+, Skills: Java & Data Analysis, Graduation Year: 2023, Course Background: Computer Science. Results update immediately.
4. Jacob reviews candidate cards showing name, GPA, skills, graduation year, location, and availability, with a high match score.
5. He clicks on one candidate to view the full profile, including education, experience, certificates, and achievements.
6. Jacob shortlists the candidate, saves the profile, and sends a message to arrange an interview.
7. He clears filters and returns to the dashboard.

Worst case scenario

1. Jacob logs in, goes to the Candidates section via the sidebar.
2. He types “**Data Science**” in the search bar but applies overly strict filters (GPA 4.0+, Skills: Python + Java + SQL, Graduation Year: 2022).
3. The system returns **no results** because no candidate meets all criteria.
4. A message appears: “*No candidates match your search. Try adjusting your filters.*”
5. Jacob realizes the mistake but cannot proceed until filters are relaxed.

Alternative 1

1. Jacob searches for “Marketing” candidates with GPA 3.0+.
2. Results show candidates, but one candidate has missing information (e.g., no graduation year).
3. The system displays available data but marks incomplete fields with “*Not Provided*.”
4. Jacob can still shortlist or request the candidate to update their profile.

Alternative 2

1. Jacob logs in and navigates to the Candidates section.
2. He enters “IT” in the search bar and adds filters: GPA 3.0+, Skill: Java.
3. He accidentally adds “Java” again as a skill filter.
4. The system prevents duplication and shows a message: “*This filter has already been applied.*”
5. Jacob removes the duplicate and continues searching successfully.

Job Posting

Description

The Job Posting feature allows employers to create, manage, and publish job opportunities directly on the platform. Employers can provide essential details such as job title, description, requirements, location, job type (internship, part-time, full-time), and application deadlines. This ensures that students have access to accurate and up-to-date opportunities, while employers can attract qualified candidates efficiently. Employers can also edit or remove postings as needed, ensuring job listings remain relevant and well-maintained.

Job Posting UI

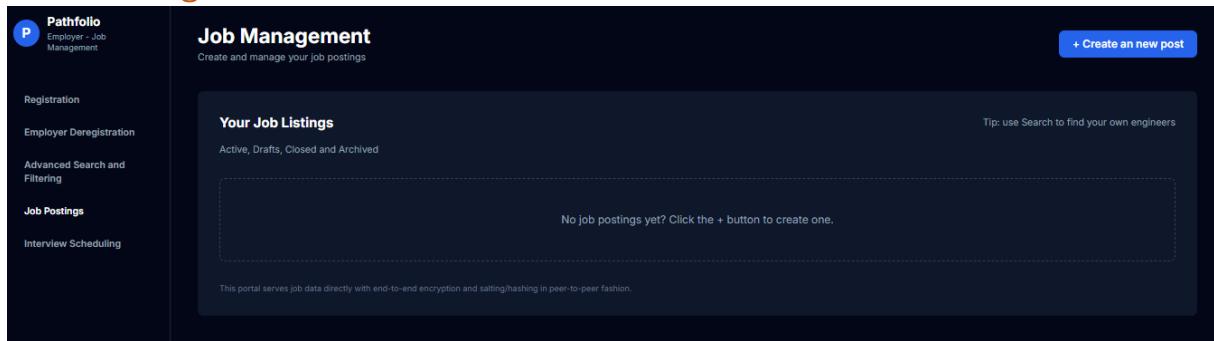


Figure 26: Job Posting UI

Job Posting UML

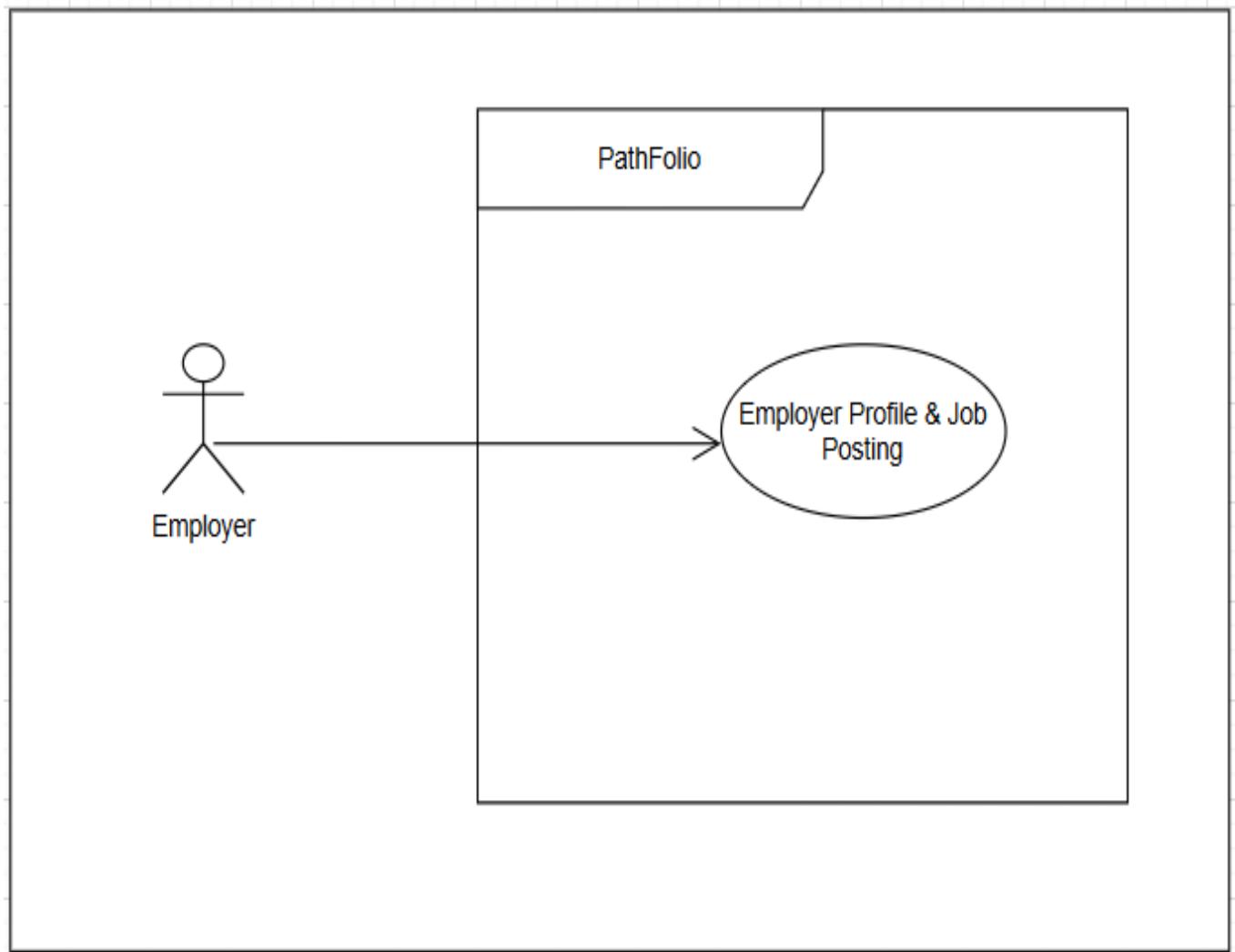


Figure 27: Job Posting UML

Step-by-step

Step 1: Accessing the Job Posting interface

- 1.1 The Employer logs into the system and is directed to the dashboard.
- 1.2 The employer clicks on the toggle to access the sidebar to access “**Job Management**”.
- 1.3 The system redirects the employer to the **Job Management** page, which includes options such as “**Post a Job**,” “**My Listings**,” and “**Applications**.“
- 1.4 The employer clicks the “+ Post a Job” button.
- 1.5 The system opens a **modal form (Job Creation Interface)** where job details can be entered.

Step 2: Entering Job Details

- 2.1 The employer fills in the following mandatory fields:

- **Job Title**
- **Job Description** (including roles and responsibilities)
- **Employment Type** (Internship / Part-Time / Full-Time)
- **Required Qualifications & Skills**
- **Application Deadline**

2.2 The employer may optionally add additional details such as:

- **Location** (On-site / Remote / Hybrid with city & region)
- **Salary Range (optional)**

Step 3: Configuring Job Visibility and Status

3.1 The employer selects the **Job Status** using a dropdown:

- **Open / Published** — immediately visible to eligible students. The posting becomes available in the **Application Portal** and appears in **student job recommendations**.
- **Draft / Hidden** — saved privately and not visible to students the job remains hidden while the employer finalizes the posting.

Step 4: Saving the Job Posting

4.1 The employer clicks the “**Publish**” button within the modal form.

4.2 The system validates the entered job details.

4.2.1. If any required fields are missing or invalid, the system highlights them and displays an error message.

4.2.2. If all details are valid, the system saves the new job posting.

4.3 The system closes the modal form, and the **Job Listings** page updates to show the new job entry.

4.4 The system provides the user with the pop-up **confirmation message** indicating the job has been successfully post and fades after a few seconds.

Step 5: Editing an Existing Job Posting

5.1 On the **Job Listings** page, the employer locates the job they wish to modify.

5.2 The employer clicks the “**Edit**” button associated with that posting.

5.3 The system responds by opening the same modal form used for creating the job postings, pre-filled with existing job details.

5.4 The employer updates any of the job information (title, description, requirements...)

5.5 The employer clicks “**Save Changes.**”

5.6 The system validates the updated information.

5.7 If valid, the system applies the changes, updates the listing, and displays a **success confirmation message**.

Step 6: Managing Job Status (Open / Closed / Archived)

6.1 On the **Job Listings** page, each job posting includes a **Status Dropdown**.

6.2 The employer clicks the dropdown to change the job’s status.

6.3 The available status options include:

- **Open** – The job remains active and visible to students in the Application Portal.
- **Closed** – Applications are disabled, and the job is hidden from the Application Portal.
- **Archived** – The job is permanently hidden from the Application Portal but kept for record-keeping or analytics.

6.4 When the employer selects “**Closed**” or “**Archived**”, the system immediately removes the posting from the Application Portal.

6.5 The Job Listings page refreshes to reflect the updated status.

6.6 The system displays a **confirmation message** (“*Job status successfully updated.*”) to confirm the change.

Best case scenario

1. Thabo clicks the “**Job Management**” button on the main sidebar.
2. The system redirects to the **Job Listings** page.
3. Thabo clicks the “**Post a New Job**” button.
4. The system opens a **job posting modal form**.
5. Thabo fills in all required job details (title, company info, description, requirements, application deadline, etc.) and sets the **status to “Open / Accepting Applications.”**
6. Thabo clicks the “**Publish Job**” button.
7. The system validates the input, saves the job posting, closes the modal, and updates the **Job Listings** page.
8. The system displays a **confirmation notification**: “*Job successfully posted and now visible in the Application Portal.*”
9. Later, Thabo decides to make an update and clicks “**Edit**” on that job posting.
10. He adjusts the details and clicks “**Save Changes.**”
11. The system updates the job posting, refreshes the listing, and displays a **confirmation message**: “*Job successfully updated.*”

Worst case scenario

1. Thabo clicks “**Job Management**” on the main sidebar.
2. The system redirects to the **Job Listings** page.
3. Thabo clicks “**Post a New Job.**”
4. The system opens the **job posting modal form**.
5. Thabo fills in the job details **but forgets to set the status to “Open”** or keeps it as “**Draft / Hidden.**”
6. Thabo clicks “**Publish Job.**”
7. The system saves the job, closes the modal, and updates the Job Listings page.
8. A confirmation message appears, but **the job does not appear in the Application Portal**, leaving Thabo confused as to why no students are applying.

Alternative Scenario 1

1. Thabo clicks “**Job Management.**”
2. The system redirects to the **Job Listings** page.

3. Thabo clicks “**Post a New Job.**”
4. The job posting modal appears.
5. Thabo fills in **only some fields** and leaves mandatory fields like **Job Title and Application Deadline** empty.
6. Thabo clicks “**Publish Job.**”
7. The system displays an **error message**: *“Please complete all required fields before publishing.”*

Alternative Scenario 2

1. Thabo opens the **Job Posting form** via the **Job Management** section.
2. The system displays the modal form.
3. Thabo enters **invalid characters or excessively long text** in fields like **Job Title**
4. Thabo clicks “**Publish Job.**”
5. The system displays an **error message** such as: *“Please remove special characters — only letters, numbers, and punctuation are allowed.”*

Interview Scheduling System

Description:

The Interview Scheduling System enables employers to invite applicants to interview directly through the platform. Employers can propose interview times, specify format (in-person, virtual, or phone), and send invitations. Students can accept, decline, or reschedule. All confirmed interviews sync automatically with both parties' calendars and trigger notifications.

Interview Scheduling System UI

The screenshot displays the Pathfellow Employer Portal interface. On the left sidebar, there are links for Employer Registration, Employer Deregistration, Advanced Search and Filtering, Job Postings, and Interview Scheduling. The main content area is titled "Job Management" and shows a search bar for "Select job to view applicants". A dropdown menu indicates "Software Engineer Intern". Below this, a table lists "Applicants" with columns for Candidate, Applied Date, GPA / Course, Status, and Actions. The candidates listed are John Doe (Interviewing), Jane Smith (Pending Review), and Sam Wilson (Shortlisted). In the "Interview Overview" section, it shows scheduled interviews for employer and candidates. One entry is for John Doe with Emily Carter on 2025-10-22 at 10:00 AM. Another entry is for Sarah Lee with Michael Chen on 2025-10-23 at 02:00 PM.

CANDIDATE	APPLIED	GPA / COURSE	STATUS	ACTIONS
John Doe	2025-10-15	3.8 / CompSci	Interviewing	View Message
Jane Smith	2025-10-14	3.9 / Data Science	Pending Review	View Message
Sam Wilson	2025-10-12	3.5 / IT	Shortlisted	View Message

Interview Overview	
Scheduled interviews for employer and candidates (demo)	
John Doe with Emily Carter For: Software Engineer Intern	2025-10-22 10:00 AM
Sarah Lee with Michael Chen For: Data Analyst Graduate	2025-10-23 02:00 PM

Figure 28: Interview Scheduling System UI

Interview Scheduling System UML

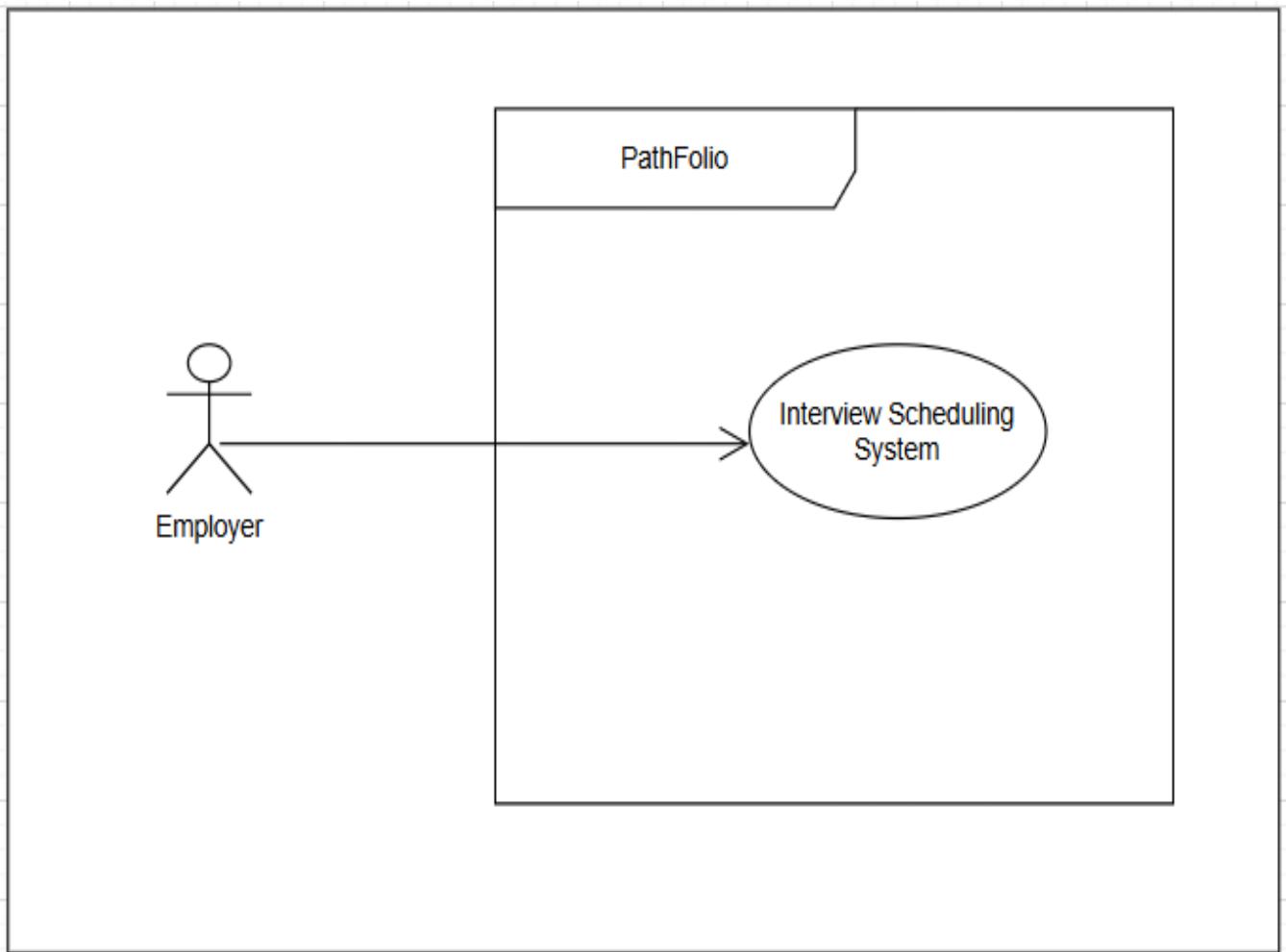


Figure 29:Interview Scheduling UML

Step-by-step

Step 1: Accessing the interview scheduler

- 1.1 The employer clicks “**Job Management**” on the main sidebar.
- 1.2 The system redirects the employer to the Job Management page, displaying all active job postings.
- 1.3 The employer selects a specific job and clicks “**Applicants**” from the secondary options.
- 1.4 The system displays a list of candidates who have applied for that job, along with their application details.

Step 2: Selecting a Candidate for an Interview

- 2.1 The employer reviews the list of applicants and identifies the candidate they want to invite for an interview.
- 2.2 The employer clicks the “**Schedule Interview**” button associated with that candidate.
- 2.3 The system opens the scheduling modal form with pre-populated candidate information.

Step 3: Scheduling the Interview

- 3.1 The employer enters the following interview details in the form:

- Interview date
- Interview start and end time

- Interview mode (e.g., online, in-person)
- Optional notes or instructions for the candidate

3.2 The employer clicks “**Confirm Interview**” to schedule the interview.

3.3 The system validates the input, saves the interview details, and updates the candidate’s status in the job application list.

3.4 A professional confirmation message is displayed: “*Interview successfully scheduled for [Candidate Name] on [Date] at [Time].*”

Step 4: Notifications

Feature User: System (Automated)

4.1 The system sends an interview invitation notification to the selected candidate via email and in-platform alerts.

4.2 The system logs the scheduled interview in both the employer’s and candidate’s interview overview pages.

Step 5: Editing or Cancelling an Interview

Feature User: Employer

5.1 The employer navigates to the scheduled interview in the Applicants list.

5.2 The employer clicks “**Edit**” to modify interview details or “**Cancel**” to withdraw the invitation.

5.3 The system validates changes, updates the interview status, and notifies the candidate automatically.

Best-Case Scenario

1. David (Employer) clicks “**Job Management**” on the main sidebar.
2. The system displays all active job postings.
3. David selects a job and clicks “**Applicants**”.
4. The system shows a list of candidates who applied for that job.
5. David reviews the applicants and clicks “**Schedule Interview**” for Sarah, a promising candidate.
6. The system opens the interview scheduling form with Sarah’s information pre-populated.
7. David selects the interview date, time, and mode (online), adds a note with instructions, and clicks “**Confirm Interview**”.
8. The system validates the details, saves the interview, updates Sarah’s application status, and displays: “*Interview successfully scheduled for Sarah on [Date] at [Time].*”
9. The system automatically sends Sarah a notification via email and in-platform alerts about the interview.

Worst-Case Scenario

1. David (Employer) clicks “**Job Management**” on the main sidebar.
2. The system displays all active job postings.
3. David selects a job and clicks “**Applicants**”.
4. The system shows a list of applicants.

5. David accidentally clicks “**Schedule Interview**” for the wrong candidate (Tom).
6. The system opens the interview form with Tom’s information pre-populated.
7. David fills in the interview date and time but forgets to add a mode or instructions and clicks “**Confirm Interview**”.
8. The system schedules the interview but the missing mode leaves Tom unsure whether it’s online or in-person, causing confusion.
9. David leaves the page without notifying Tom directly, so Tom sees only the automated notification and is left unsure about the exact details.

Alternative Scenario 1

1. David (Employer) clicks “**Job Management**” on the main sidebar.
2. He selects a job and clicks “**Applicants**”.
3. David wants to schedule multiple interviews for several candidates at once.
4. The system allows selecting multiple candidates, but David accidentally selects overlapping time slots.
5. When he clicks “**Confirm Interview**”, the system schedules all interviews but some overlap, leaving the candidates confused about their actual interview times.
6. David realizes the overlap but cannot easily fix it from the bulk scheduling interface, causing partial rescheduling via separate messages.

Alternative Scenario 2

1. Sarah (Candidate) logs into the portal and clicks “**Applications**” on her sidebar.
2. She sees a new interview invitation from David (Employer).
3. The notification includes date, time, and mode.
4. Sarah tries to select a preferred time slot from the available options but accidentally clicks a time that conflicts with another commitment.
5. The system accepts her selection and confirms the interview, but she realizes later that she cannot attend that time.
6. Sarah is left needing to contact the employer separately to request a reschedule, creating some uncertainty and extra effort.

Administrators Features

Analytics Dashboard

Description

The Analytics Dashboard provides university administrators with focused insights into job placement outcomes, employer engagement, and skill demand trends. It helps admins monitor placement rates, evaluate how employers interact with the platform, and identify in-demand skills so they can make data-driven decisions. Access is restricted to Admin users.

Analytics Dashboard UI



Figure 30: Analytics Dashboard UI

Analytics Dashboard UML

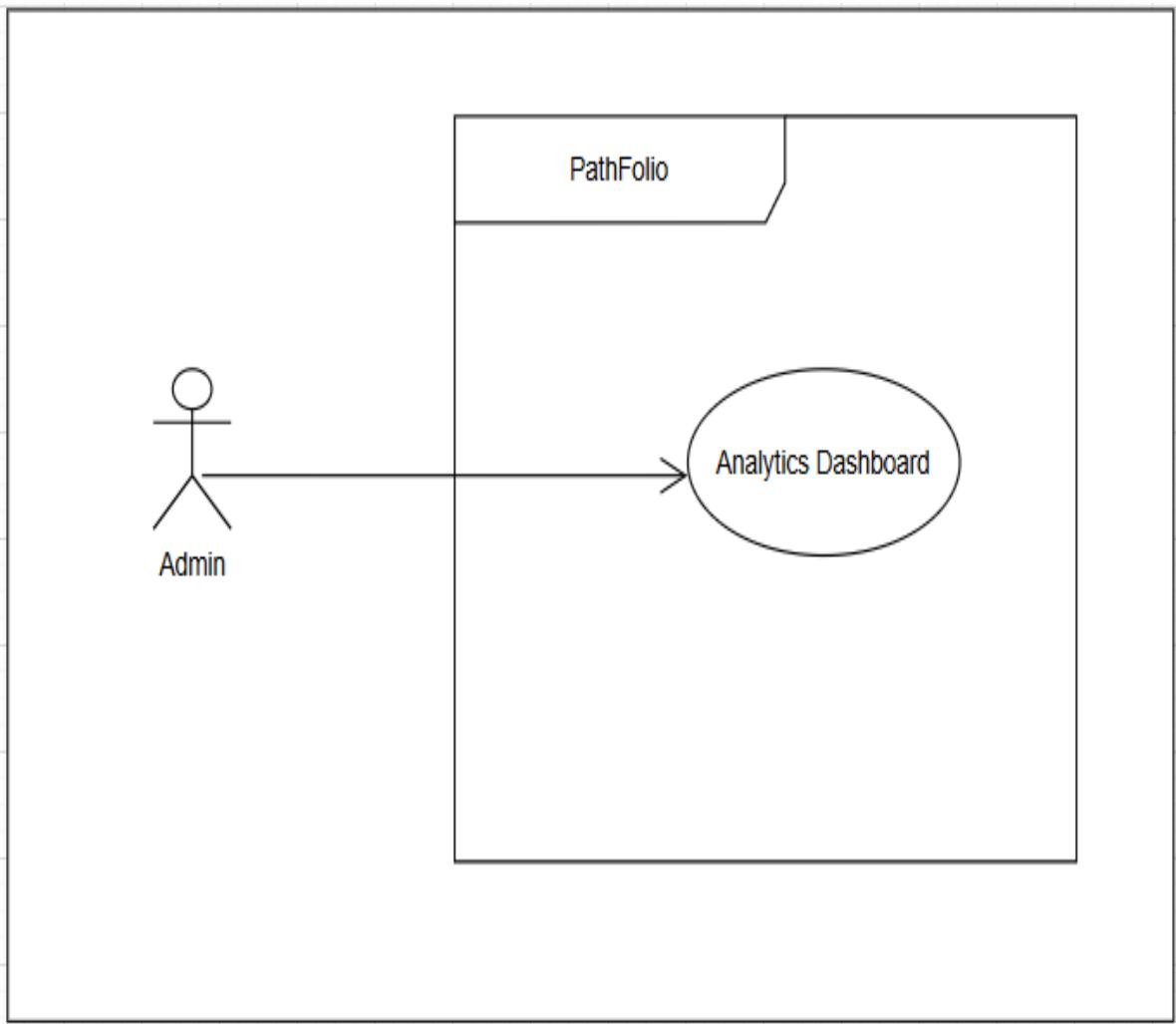


Figure 31: Dashboard UML

Step-by step

Step 1: Accessing Analytics

- 1.1 The admin clicks “**Analytics**” on the main sidebar.
- 1.2 The system redirects the admin to the **Analytics Dashboard**.
- 1.3 The dashboard displays a summary header and prompts the admin to choose one of three analytics categories: **Job Placement Rates**, **Employer Engagement**, or **Skill Trends**.

Step 2: Selecting Analytics Category and Options

- 2.1 The admin selects the desired analytics category (one of: **Job Placement Rates**, **Employer Engagement**, **Skill Trends**).
- 2.2 After selecting a category, the system reveals related filter controls. Available filters include:
 - **Date range** (preset ranges and custom ranges)
 - **Program / Faculty / Course**
 - **Employer type** (industry, public/private — verified employers only)

Step 3: Applying Filters and Generating the Report

- 3.1 The admin chooses a date range and any program or employer filters required.
- 3.2 The admin clicks “**Generate**” (or “**Apply Filters**”) to run the report.
- 3.3 The system queries the data sources, computes the requested metrics, and renders the report for the selected category.

Step 4: Viewing Reports and Key Metrics

- 4.1 The admin reviews the **summary statistics** displayed at the top of the report (e.g. total placements, number of active employers, or most requested skills — depending on selected category).
- 4.2 The admin scrolls down to view the **detailed breakdown** in table or chart format.
- 4.3 The admin clicks on **column headers** (e.g. “Highest Placement Rate” or “Most Active Employer”) to **sort the results**.
- 4.4 If needed, the admin adjusts filters (e.g. date range or program) and the **system refreshes the report automatically**.

Best case scenario

1. Alex (Admin) clicks “**Analytics**” on the main sidebar.
2. He selects **Job Placement Rates** and sets the date range to the last 12 months and filters for the Engineering faculty.
3. Alex clicks **Generate** and the dashboard shows a clear conversion funnel and the placement rate by program.
4. He identifies a healthy increase in hires for a particular program and exports the summary as a PDF to share with faculty leads.
5. Alex also schedules a monthly placement report to be emailed to relevant administrators automatically.

Worst case scenario

1. Candice (Admin) opens the Analytics Dashboard.
2. She chooses **Employer Engagement** but does not set a date range or program filter.
3. Candice immediately generates the report and shares it with stakeholders.
4. The report contains all historical employers (including inactive accounts and old postings), producing an overwhelming set of results.
5. Stakeholders are confused by the unfiltered data and struggle to extract current, actionable trends.

Alternative scenario 1

1. John (Admin) needs to compare placement trends year-over-year.
2. He selects **Skill Trends** but accidentally sets the date range to a single month instead of one year.
3. The system returns a short-term spike in one skill that looks significant on the chart.
4. John misinterprets the short-term spike as a long-term trend and schedules an unnecessary skills workshop based on that single-month view.

Alternative scenario 2

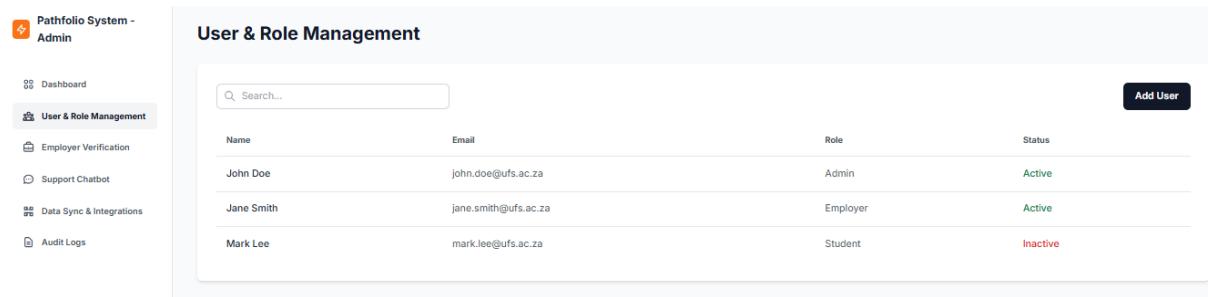
1. Sandra (Admin) selects **Job Placement Rates** and views the dashboard.
2. She sees a metric labelled “**Response Rate**” and assumes it means “percentage of applicants hired.”
3. Acting on that assumption, Sandra prepares a report recommending programmatic changes to improve hiring.
4. Later it is clarified that **Response Rate** measures employers’ reply rate to applications, not hires, so Sandra’s initial recommendations were based on an incorrect interpretation of the metric.

User Access Control

Description

The User Access Control feature enforces role-based access for all platform users. Users are automatically assigned a role (Student, Employer, or Administrator) based on their registration path. This role determines which features and sections of the platform a user can access, ensuring that sensitive actions are restricted to authorized users only. Once assigned, roles are permanent and cannot be changed.

User Access Control UI



The screenshot shows the 'User & Role Management' page. On the left is a sidebar with icons for Dashboard, User & Role Management (selected), Employer Verification, Support Chatbot, Data Sync & Integrations, and Audit Logs. The main area has a search bar and a 'Add User' button. A table lists three users:

Name	Email	Role	Status
John Doe	john.doe@ufs.ac.za	Admin	Active
Jane Smith	jane.smith@ufs.ac.za	Employer	Active
Mark Lee	mark.lee@ufs.ac.za	Student	Inactive

Figure 32: User Access Control UI

User Access Control UML

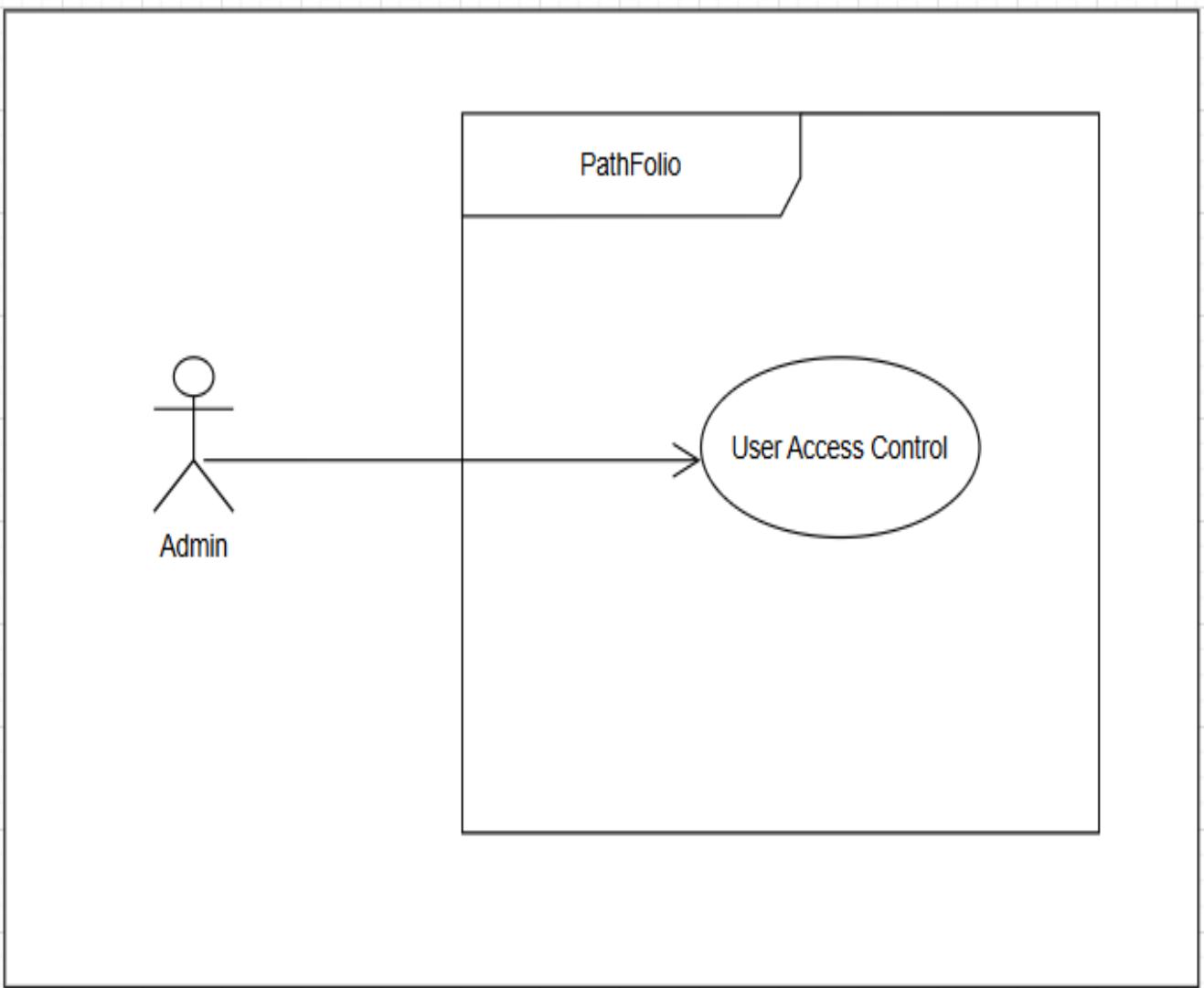


Figure 33: User Access Control UML

Predefined Roles:

- **Student:** Profile management, job applications, academic integration
- **Employer:** Candidate search, job posting, interview scheduling
- **University Admin:** Analytics access, user management, employer verification

Step by step

Step 1: Access Restrictions Based on Role

1.1 The system checks the user's role on every page or feature request.

1.2 Permissions per role include:

- **Student:** Access to student dashboard, profile, job search, applications, CV builder, interview scheduling (for self), analytics of own activity.
- **Employer:** Access to employer dashboard, job posting, applicant management, interview scheduling (for posted jobs), analytics of their posted jobs and applicants.
- **Administrator:** Access to admin dashboard, user management, platform-wide analytics, and reporting tools.

Step 2: Denying Unauthorized Access

- 2.1 If a user attempts to access a feature outside their role, the system blocks the request.
- 2.2 The system displays an informative message: "You do not have permission to access this section."
- 2.3 The user remains on the last accessible page or is redirected to their role-specific dashboard.

Step 3: Login and Role Validation

- 3.1 Each time a user logs in, the system checks their role from the database.
- 3.2 Based on the role, the system displays the appropriate dashboard and menu options.
- 3.3 Unauthorized menu items or buttons are hidden from the user interface.

Best case scenario

1. David registers as an Employer via the registration page.
2. The system validates his information and assigns the Employer role.
3. David logs in and sees the employer dashboard with options to post jobs, manage applicants, and schedule interviews.
4. All student-only features are hidden; David can navigate the platform efficiently without encountering errors.

Worst case scenario

1. Sarah registers as a Student but tries to access the Employer job posting section directly via URL.
2. The system detects that her role is Student.
3. The system blocks access and displays the message: "You do not have permission to access this section."
4. Sarah remains on her student dashboard, unable to access employer features.

Alternative scenario 1

1. Michael (Employer) logs in and attempts to access admin analytics.
2. The system validates his role and restricts access.
3. Michael sees the message: "You do not have permission to access this section" and remains on the employer dashboard.

Alternative scenario 2

1. Emily (Student) logs in and tries to view analytics for other students.
2. The system validates her role and restricts access.
3. Emily is only able to see her personal activity analytics and is prevented from viewing others' data.

Support Chatbot

Description

The Support Chatbot provides guided help specifically for administrators to manage administrative tasks such as profile oversight, job posting management, and CV guidance for students. It delivers step-by-step instructions, automated responses, and suggestions tailored to the administrator's tasks.

Support Chatbot UI

The screenshot shows the 'Support Chatbot' interface within a web-based application. The left sidebar is titled 'Pathfolio System - Admin' and includes links for Dashboard, User & Role Management, Employer Verification, Support Chatbot (which is selected and highlighted in grey), Data Sync & Integrations, and Audit Logs. The main area has a title 'Support Chatbot' and a message input field: 'Hello! I'm here to help. How can I assist you today?'. A button 'I need help with my profile setup' is visible. The chatbot responds with: 'Certainly. To set up your profile, navigate to the "User & Role Management" section from the sidebar. You can add new users and assign roles there. Is there a specific part of the profile setup you're having trouble with?'. Below this, another message asks: 'How do I verify a new employer that has registered on the platform?'. The chatbot replies: 'Employer verification can be done from the "Employer Verification" tab. You will see a list of pending verification requests. You can review their details and approve or reject them from that screen.'. A third message asks: 'The data sync with the university's main database seems to be failing. What should I check?'. The chatbot responds: 'I'm sorry to hear you're having issues. Please check the "Data Sync & Integrations" page for the status of the connection. You can also view the "Audit Logs" to see if any specific errors are being reported during the sync attempts. If the problem persists, I can escalate this to our technical team.' At the bottom, there is a text input field 'Type your message...' and a 'Send' button.

Figure 34: Chatbot UI

Support Chatbot UML

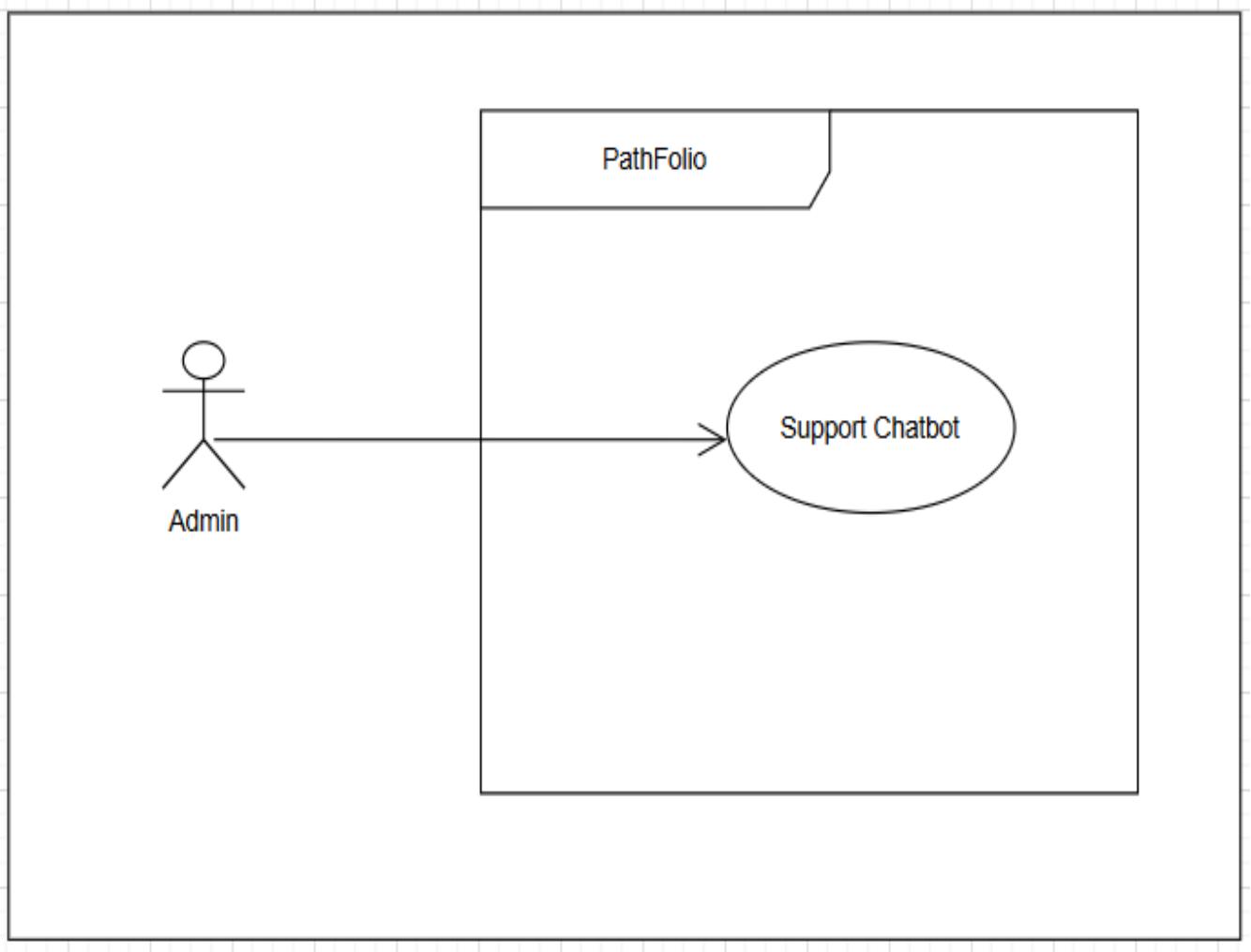


Figure 35: Chatbot UML

Step-by step

Step 1: Accessing the Chatbot

- 1.1 The admin navigates to the main sidebar and clicks the “**Support Chatbot**” button.
- 1.2 The system opens the chatbot interface.

Step 2: Selecting a topic

- 2.1 The chatbot presents a list of administrative help topics:

- Profile Setup Oversight
- Job Posting & Applicant Management
- CV Builder Monitoring

- 2.2 The admin clicks on the desired topic to receive guidance.

Step 3: Receiving Guidance

- 3.1 The chatbot provides step-by-step instructions for the selected topic.
- 3.2 The system may include:

- Suggested next administrative actions
- Links to relevant pages or forms

- Quick tips for efficient management

Step 4: Asking Questions

- 4.1 The admin can type specific questions about administrative processes.
- 4.2 The chatbot responds with context-specific guidance or directs the admin to relevant resources.

Step 5: Completing the Task

- 5.1 The admin follows the instructions provided by the chatbot to complete the administrative task (e.g., managing job postings, verifying student CVs, or overseeing profiles).
- 5.2 The system logs interaction history for reference or future analytics.

Best case scenario

1. David (Admin) clicks the **Support Chatbot** button on the sidebar.
2. The chatbot interface opens immediately.
3. David selects **Job Posting & Applicant Management** as the topic.
4. The chatbot provides clear, step-by-step instructions for posting a job and reviewing applications.
5. David follows the instructions successfully and completes the task efficiently.
6. The system logs the interaction and confirms that the guidance session is complete.

Worst case scenario

1. David (Admin) clicks the **Support Chatbot** button on the sidebar.
2. The chatbot interface opens.
3. David selects **CV Builder Monitoring** as the topic.
4. David types a vague question like “CV issues.”
5. The chatbot provides generic instructions that are unclear to David.
6. David is left unsure how to proceed with verifying student CVs, delaying administrative tasks.

Alternative scenario 1

1. David (Admin) clicks the Support Chatbot button.
2. He selects Profile Setup Oversight as the topic.
3. The chatbot provides guidance and includes links to example student profiles.
4. David partially follows the instructions but skips reviewing some fields.
5. The system logs that the review was incomplete, leaving David aware that additional follow-up is needed

Alternative scenario 2

1. David (Admin) clicks the Support Chatbot button.
2. He selects Job Posting & Applicant Management as the topic.
3. The chatbot provides step-by-step guidance for reviewing applicants and updating job statuses.
4. David attempts to follow the steps but enters incorrect data (e.g., wrong application status).
5. The chatbot provides reminders about correct procedures but does not correct the input, leaving David to manually fix the issue.

Chapter 3: Software Project Management Plan

Introduction

The Software Project Management Plan (SPMP) outlines the key processes, objectives, and deliverables for the development of the Pathfolio System. It provides a framework for project organization, timeline, resource allocation, and risk management to ensure the successful delivery of the project. The plan details the scope, objectives, assumptions, and constraints involved in building a comprehensive system for managing academic and administrative processes.

Software Project Management Plan

1. Overview

1.1. Project Summary

The Software Project Management Plan will outline the details of the management plan that will be followed meticulously to develop the Pathfolio System. It entails the development cycle, organization, specific roles, projected timelines and testing procedures.

1.1.1.1.Purpose, scope, and Objectives

The Software Project Management Plan relays all the details regarding the development plan and the development cycle. Each member of the development team is assigned roles and duties necessary for project delivery, together with the procedures they will employ to finish their respective tasks. It will also outline deadlines so that the project will be delivered on time. The first prototype will include a UI design, which will be responsive and be able to navigate through different tabs and pages, contain the logo and the corresponding prompts according to the client's specifications.

1.1.1.2.Scope

- User Login and Registration
 - Login: All the users including UFS students, administrators and employers log into the system using their credentials.
 - Registration: The system allows new users to be able to register their credentials in the database, by making up all the credentials.
- Communication and Announcements
 - Announcements: The Administrator and the employer can create announcements. The administrators can create emergency announcements that are sent via email to alert the students of any imminent announcements.
 - The administrators can delete and monitor irrelevant and inappropriate announcements.
 - The students can only view the announcements.
- Collaboration tools:
 - Support Chatbot: Enables the user to query the system for help, and it is used by the students and employers.

- Resource management
 - Uploading: The students and employers can upload supporting documents to validate whatever content they uploaded. For students they will provide documents to prove the skills they have acquired. The employer will upload all the documentation that validates the legitimacy of their corporation.
 - The material uploaded by the students will be verified by the administrator and only then will it be shown as a badge earned by the students which will be visible on their profile.
- Analytics and Tracking:
 - Progress Tracking: Analytics tools track placement rate, total students registered, and the total number of vacancies posted. The admin views this metrics on their dashboard to gauge the effectiveness of the system.

1.1.1.3. Objectives

The main objective of the Pathfolio system is to help students equip themselves with skills and accolades that are necessary and important to the real-world industries. The employers get to have only the students who are work ready which helps the organizations spend even less time and resources on training the new workforce.

1.1.2. Assumptions and Constraints

The following is a list of all assumptions and constraints:

- Team members will attend all the meetings and on time.
- Team members will meet all the deadlines.
- Team members will deliver high quality code throughout the development of the project.
- Team members will follow all the requirements meticulously

The following are a list of constraints:

Delivery Time:

The system is expected to be ready and fully operational by 19 October 2025 at 17h00 (CAT), meeting all the specified milestones and requirements.

Budget:

The total internal budget for the development of the project is capped at R2 500 000, 00. The estimated budget should cover development costs, resources, testing deployment and initial maintenance.

Project Personnel

The project team includes only 5 members: Vuyo Nzimande (Project Manager), Dube Naledi, Mokhotsoa Sepinare, Seipone Amogelang, Lutu Mpukuco. No consultations are expected to be utilized unless stated otherwise.

Testing and Deployment Timeline:

The timeline for testing must be strictly followed, which cannot exceed beyond the final delivery date of the project. This includes system testing, user acceptance testing, and resolving critical bugs or performance problems

Programming Environment:

All the technology used for the development of the system will be in Microsoft Technologies, which includes C# as the core language if development, Microsoft SQL Server for the database management system, and ASP.NET Core MVC for full stack.

1.1.3. Project Deliverables

- Pathfolio system Software:
 - Description: A fully functional software, including all the key features.
- User Manual and Training Guide:
 - A user manual outlining how each user is supposed to interact with the system, including how to navigate the system, perform key functions and troubleshoot common problems. This ensures that the user will utilize the system efficiently and effectively.
- Detailed Source Code:
 - Full access to the project's source code includes all the documentation related to the code.

1.1.4. Schedule and Budget summaries

The project has an estimated 4-month schedule. This does not include the weekends. To finish the project within the allocated time, team members need to work additional working hours. This requires dedication of about 16 hours every week to stay on schedule.

Estimated Costs and Workflows:

Workflow	Hours	Hourly Rate (R)	Estimated Cost(R)
Requirements	106	350	37,100.00
Analysis	122	350	42,700.00
Design	126	350	44,700.00
Implementation	296	350	103,600.00
Testing	250	350	87,500.00
Total Development Cost Estimated	900	350	315,000.00

General Expenses:

Software Related:

- Microsoft Visual Studio Enterprise R 4,750 per user/month.
- Microsoft SQL Server Essentials R9,515 lifetimes.
- Microsoft 365 Business Premium R418 per user/month

Expense	Cost (Rands)	Estimated Overall Cost
Rent	15,600 p/m * 4.5	70,200
Utilities (water, internet, insurance, legal counsel, bank fees)	5,600	25,200
Travel and food	6,000	27,000
Software	-	147,797

Miscellaneous	-	68500
Total		338697

1.2 Evolution of the project management plan

2 Reference Material

- Schach, S.R., 2011. Object-Oriented and Classical Software Engineering. 8th ed. New York: McGraw-Hill.
- AVIATE Solutions(2024) Software Engineering Project: [### 3 Definitions and acronyms.](https://learn-eu-central-1-prod-fleet01-xythos.content.blackboardcdn.com/6195b5fbe2480/18456906?X-Blackboard-S3-Bucket=learn-eu-central-1-prod-fleet01-xythos&X-Blackboard-Expiration=1760194800000&X-Blackboard-Signature=m4Z3RPnuDfGXiYXsSTAXmRrmnVYhwy8ZLi0SxOMP5dI%3D&X-Blackboard-Client-Id=309416&X-Blackboard-S3-Region=eu-central-1&response-cache-control=private%2C%20max-age%3D21600&response-content-disposition=inline%3B%20filename%2A%3DUTF-8%27%27Final%2520Team%2520Project%2520submission_Best%2520Project%25202024.pdf&response-content-type=application%2Fpdf&X-Amz-Security-Token=IQoJb3JpZ2luX2VjEGoaDGV1LWNlbnRyYWwtMSJIMEYCIQDaLUkHTY7bkvY3OtW4t6vlgrYHgj94ewd8B8uE8aiuqglhAN5qMTnyzCvQHnUP%2FILd6rT6HUzw11Tc11AbDVTIQZZGKr4FCBMQBB0MNjM1NTY3OTI0MTgzb9uOPwhOZkMDCEbZMqmwmXeHwlGBiqCmxqpYKXZ2F%2B4LFQH7nW3yGGjYUmE9kclq3sB5D6pMP0H17VQ3P8rHYIuJJ2n9d23PtHioDWRy19U%2FmYJ44swPkwiAmuMXCs2FWADAOcpL3RKOpMVcZow9m28sEX0SCBpVsOHrBPTcDv49TVeZFWBSTJ%2FnRGdBAkiG1u4Lmgn7ogC%2BB5I0u05WqzjjrThNANnPyUnyQJ8FJICpPC8TPO1ilo4YfVjkmgQHzCPzbChqcYEeeJ2d6s%2BQZ%2Bbz%2FiSUXvvgTCC7IJN77mjLQYVIHe%2FzzvrZDdOu1SR9pZR%2FZuOwUzf1edZGYg304xne%2ForWa%2F099106aYDDiqoV9k2tT1gcaZoajgGI%2BWDP3td0YhQ8Ac%2B0RFA2u%2FI0j2Qf2R44k7Eu3RWgKZe%2BJAIlZ8fqJw25rg78wacm%2BpzNV8WvuP0TpxkQ3UbUTWOrWSVAAxtpQJ%2F%2BLzad60VJnGv534Vuugl9hV4RSgr7DgmXYtkW3V%2FXBERtrwrGGalrieNeaj1zk7RCUxCsNgQ%2Blyj3%2B1WCBN8tX0Pe2hUahBwsQKngc17PGvO1Or7u0pMTi55VYbNb%2FdN6%2BwvOBz1Q4rvvbPzNaf%2FRbnzjoyzH6qf5039hXNi2anfaVU1%2BnUWiQ7IIJitwnllaglHvTzxwlG5jzQI354CK2COADQqpV1921Zwl05fXQPjK1vsFg6Mgos62FnTpupXwQNOrQP8esmyDh2X1rWpHvKWCb2cjTsCscPMJw4aiHZauyKsF6J7ipJ0fzpRcoePqZr%2FoUWx37ysViwaOa0o5nG8BMkp7H3YuhNhRIFZC%2B0IAMex4goz%2BaDyW%2BdN374DwGfT8ZpydNHRPO0LEiME2E8rNovDpm6Nfazc9GdksdNML7IqMcGOrABr1gFKR7IPbZbKqETkhQbbKTS8G1loVfBGgE729flJ8SEVtDsbQ6MDQT0GD2%2BNaijlqkTRKaCC0gisonkajuJzjNcW8qeNjGNAoTUQ1gyE%2FJrb6iv0TEI8rSftjBJfLgS0qnllypu9ZDbm5VHifm%2FEobRL21TgXICiDTWlJcWk7Z9qPEsscJP49mBVN%2B%2Btvu9ErEMsBy1b2kqVyrAEXCet1TMVkuRy0MwCQ%2FV9pzo%3D&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20251011T090000Z&X-Amz-SignedHeaders=host&X-Amz-Expires=21600&X-Amz-Credential=ASIAZH6WM4PLQSNCDARX%2F20251011%2Feu-central-1%2Fs3%2Faws4_request&X-Amz-Signature=5f7c6f431b1781c5615f6c45d011ffd74a8d67542378a6adef769581b21f9643

</div>
<div data-bbox=)

3.1 Definitions

- **Client:** An individual or entity that the project is created for. The client provides the project's requirements, feedback and approves the final project
- **Pathfolio System:** A platform created and designed to help students equip themselves with skills that are necessary for work and employers to stipulate exactly what is expected from the soon to be work force.
- **Admin:** The system administrator is the UFS staff that are responsible for managing, validating user roles, access permissions and systems configurations.
- **Employer:** All validated corporations that are in need of skilled personnel.

3.2 Acronyms:

- QA: Quality Assurance
- SPMP: Software Project Management Plan
- POPI: Protection of Personal Information
- UI: User Interface
- PM: Per Month
- OS: Operation System
- VS: Visual Studio
- PU: Per User
- UAT: User Acceptance Testing
- IDE: Integrated Development Environment
- DBMS: Database Management System
- PM: Project Manager
- LD: Lead Developer
- UX: User Experience

4. Project Organization

4.1 External interface

Client Organization

The client organization, which oversees the development and delivery of the system, will provide all the requirements which are used in the requirement phase of the development cycle if the project as a first step. Meetings and updates are conducted throughout the development of the project to ensure that the development team does not deviate from the needs of the client. This ensure that the system is aligned with the latest goals and expectations of the client.

4.2 Internal structure

Development Group:

- Developers (All the team members): Responsible for writing code, implementing features and integrating subsystems.
- UI/UX Designers (S. Mokhotsoa, N. Dube): Concentrate on the design of the UI and ensures that it is user friendly and meets the usability goals.
- Database Administrator: (A. Seipone): Handle data storage and management, ensuring the embedded database is properly designed and optimized for efficient storage of data.
- Project Manager (V. Nzimande) Oversees the development of the entire project, ensuring that the independent modules are working well after integration and that the development of the project is on schedule.
- System Architect (M. Lutu): Is responsible for the architecture of the entire system.
- Quality Assurance (Dr R. Fouche): Ensures that the system works as stipulated and that everything is well integrated.

4.3 Roles and responsibilities

4.3.1 Project Manager (V. Nzimande)

The PM oversees the entire project, performs coordination duties between all the members of the team as well as the members of the client entity. Ensuring that all the project milestones are achieved.

Lead Developer (M. Lutu)

The LD is responsible for managing the technical aspects of the project, ensuring that the code is well structured and adheres to the standards of the organization.

UI/UX Designer (V. Nzimande, N. Dube)

Responsible for designing the UIs in a way that facilitates maximum efficiency and effectiveness and ensuring that all the usability goals and user's needs are met.

Database Administrators:

Ensure that the DBMS is embedded correctly and the tables are well structured, and that the database is configured optimally and data retrievals are done efficiently.

System Architect:

Analyses business requirements to ensure that they align with the organizational goals and structure. They are responsible for the entire lifecycle of the system, from initial concepts, design, implementation, maintenance and optimization, ensuring that all the components like hardware, software and network work together seamlessly.

Quality Assurance:

Ensures that all the Pathfolio system is tested thoroughly, which will include functional testing, performance testing and security testing, ensuring that all the standards are met meticulously.

4.3.2 Responsibilities

V. Nzimande (PM)

- Ensures the project is on time and within budget
- It is a medium of communication between development team and relevant stakeholders.
- Oversees resources allocation and risk management.
- Facilitate meetings and provide regular project status updates to the client

S.Mokhotsoa, N. Dube(UI/UX Designers)

- Design the user interface and layout of the system
- Creates responsive prototypes for client feedback
- Ensures usability goals are met
- Ensures design feasibility and implementation

Amogelang Seipone (Database Admin)

- Design and implement the database schema for the system
- Optimizes the database performance
- Manage data migration and backup procedures
- Ensures data integrity and security

M. Lutu (Lead Developer)

- Oversee technical design decision
- Guide the development team
- Ensures code quality
- Aligns the backend and front end of the system.

System Architect:

- Design and oversee complex IT system
- Analyses business requirements to align technology with organizational goals.
- Responsible for the entire lifecycle of the system.
- Creates technical specifications, managing cost and ensuring security and performance of the system.

5. Managerial Process Plans

5.1 Startup Plan

5.1.1. Estimation Plan

- Excessive iterations of going through the specifications document to determine all the client's specifications for the system.
- Research of all the required technologies to be used in the system and research for any optimum alternatives of the technologies to meet the specifications.
- Outline of how much can be achieved in the given period.
- Create guidelines for Prototype 1 and 2
- Create estimated deadlines and task completion deadlines to create these prototypes on time.

5.1.2. Staffing Plan

The project staff was picked according to evaluations of strengths and weaknesses amongst the group members. With individuals placed to roles according to their strengths and freedom of duties, to duties that would include using their weakness.

5.1.3. Resource acquisition Planning

The software required for this project requires subscriptions to Microsoft's Environment Visual Studio, SQL Server, and 365 for each member of the staff, of which the cost of these has been outlined in section 1.1.4 of this document.

5.1.4 Project Staff Training Plan

Every group member undergoes discussion classes twice a week with Dr Fouche and completes 1 tutorial quiz a week based on the work that has been done that week. Each group member has undergone Microsoft 365 training.

5.2 Work Plan

5.2.1 Work activities

Work Units:

- UI/UX for prototype 1: The first prototype contains a working UI/UX which would be developed by V. Nzimande, S. Mokhotsoa, and N. Dube.

- UFS databases integration: The UFS database integration will be done by M. Lutu
- Prototype 2: This version of the system will include prototype 1 integrated with the UFS database and receive data from the Database. The version of the system will also include responsive UI which increases the UX. All team members are involved in this part.
- Documentation: All team members make sure that any additions, removals and changes to the system are well documented.

5.2.2 Schedule Allocation

Start Date	End Date	Artifact
01/11/2025	15/11/2025	Prototype 1 Development and Continuous testing
16/11/2025	17/11/2025	Final testing of Prototype 1
18/11/2025		Delivery of Prototype 1
20/11/2025	07/01/2026	Prototype 2 development and Continuous testing
09/01/2026		Delivery of Prototype 2

5.2.3 Resource Allocation

Each member has access to the same resources as all the other group members. The resources include Microsoft Visual Studio, Microsoft SQL Server, and Microsoft 365 as outlined in sections 1.1.4 and 5.1.3 of these documents.

5.2.4 Budget allocation

No budget has been yet allocated for this project.

5.3 Control Plan

5.3.1 Requirements Control Plan

Each member is required to attend the scheduled group meetings. The meetings and important notes of each meeting are documented. Each member must Follow the guidelines in the Specifications document and meet each deadline. Any unexpected issues, technical difficulties or requests by the clients will be assessed by the team and decided upon.

5.3.2 Schedule Control Plan

Except for the scheduled meeting times, the team might be required to meet outside of the scheduled times to finish and deliver the system on time. The Project manager will make sure each member of the team completes their task on time.

5.3.3 Budget Control Plan

There is no budget allocated to PathFolio Systems, but any future budget provided will be efficiently used according to section 1.1.4

5.3.4 Quality Control Plan

The team will constantly perform a quality check on the system at least once a week to make sure that the project meets all expectations. The client will be present during the development of the system and therefore will be informed of the quality of the system.

5.3.5 Reporting Plan

The project manager will generally note the attendance of the group members in meetings and make sure that the system is up to date.

5.3.6 Metrics Collection

The project manager will make sure that each week every development task is on track and progressing as expected. The project manager will make sure that the code is efficient, and meets all the standards.

5.4 Risk Management Plan

Development

- The team will meet regularly to make sure that the production is not stopped at one point.
- There will be scheduled deadlines that everyone will be following to make sure the project is not delayed
- Each team member will be kept up to date, and inform the project manager of any changes or difficulties that might affect their ability to complete the task they are given on time.

Project Failure

- If the technology does not exist or is not viable the project manager will discuss it with the client and make sure that there is another route that will result in a satisfactory outcome for the client.

Server failure

- Server failure can only be experienced when UFS Servers are down.

5.5 Close-out Plan

The team will submit all the deliverables to Dr Fouche.

6. Technical Process Plan

6.1.1 Process Model

Agile Software Development



6.2 Methods, Tools and Techniques

- Method: Incremental and Iterative Development Plan.
- Tools: Visual Studio, SQL Server.
- Techniques: Regular meetings with client.

6.3 Infrastructure Plan

There is no need for an infrastructure plan since UFS backend infrastructure is adopted.

6.4 Project Acceptance Plan

The client will be kept in loop throughout the development. After which the team will test the system and make sure that the prototypes are working and acceptable.

7. Supporting Process Plans

7.1 Configuration Management Plan

The document will include a section of changes made to the document and code and a request for change process that will allow for smooth configuration.

7.2 Verification and Validation Plan

Verification and Validation are done throughout every phase not only at the end of a phase or at the end of development.

7.3 Documentation Plan

All the team members are responsible for the Specifications Requirement Document, the SPMP, and the design and architecture document.

7.4 Quality Assurance Plan

The project manager will make sure of the quality of the system.

7.5 Reviews and Audits

During development, every member of the team will test and report any deficiencies in the system.

Design flaws or bugs will be reported and documented for immediate fix or future review.

7.6 Problem Resolution Plan

Each member of the team will keep the team members up to date on any issues that they might encounter. Afterwards the team will decide on how to handle those issues and make sure that the project is completed. They will also make any changes necessary to make sure that the project runs as smoothly and as efficiently as possible.

7.7 Subcontractor Management Plan

We have no subcontractors

7.8 Process Improvement Plan

To enhance the software's development process, it is essential to produce clear and comprehensive documentation for the source code provided. Certain features (Some of the optional features) could not be completed within the current timeframe and have therefore been postponed to the next phase of development. By writing clean, well-structured code, thoroughly documenting it, and preparing a detailed maintenance manual, the future development team will be better equipped to implement any necessary updates or additions. Additionally, the client has been kept informed about the deferred features, ensuring they can assist the future team in determining the next steps.

Chapter 4: Quotation

PROJECT BUDGET QUOTATION

QUOTE NUMBER: 00821

QUOTE DATE: 16/10/2025

BILLED TO: Dr R. Foucher
University of the Free State
Bloemfontein, Free State, SA,9301
082-456-7810

PathFolio Systems
8 Rindl Street, Universitas
Bloemfontein, Free State, SA 9301
quotations@pathfoliosys.co.za
051-653-9090

DESCRIPTION	UNIT COST	QTY/HR RATE	AMOUNT
Requirements Gathering	R37,100.00	1	R37,100.00
Analysis	R42,700.00	1	R42,700.00
System Design	R44,100.00	1	R44,100.00
Implementation	R103,600.00	1	R103,600.00
Testing & QA	R87,500.00	1	R87,500.00
Software Support	R147,797.00	1	R147,797.00
General Expenses	R260,300.00	1	R260,300.00
Office Rent	R70,200.00	1	R70,200.00
Utilities	R25,200.00	1	R25,200.00
Consultation	R9,800.00	10	R98,000.00
Travel & Accommodation	R27,000.00	1	R27,000.00
Salaries & Contractors	R2,065,000.00	1	R2,065,000.00
Miscellaneous	R68,500.00	1	R68,500.00
SUBTOTAL			R3,076,997.00
DISCOUNT			R255,000.00
TAX RATE			15%
TAX			R423,299.55
TOTAL			R3,245,296.55

Prepared by: PathFolio Systems

Contact: quotations@pathfoliosys.co.za

Chapter 5: Design Documentation

Noun Extraction

Description Summary

The **University of the Free State (UFS)** is initiating the **development** of a comprehensive **recruitment** and student **profile platform** aimed at improving graduate **employability** and facilitating employer **engagement**. The **platform** will function as a centralized digital **bridge** between academic **performance** and career **opportunities**, integrating securely with **UFS** academic **databases** for real-time student **data updates**.

The **system** will be delivered as both a web **application** and a mobile **application**, enabling **students** to create professional **profiles**, build **AI-assisted CVs**, showcase **skills** and **portfolios**, receive job **recommendations**, and apply directly through an **application portal**. **Employers** will benefit from advanced candidate **search**, job posting **tools**, interview **scheduling**, and optional **access** to verified academic **records** with student **consent**.

Administrators will access **analytics dashboards**, manage **user roles**, and utilize **tools** such as **skill gap analysis** and a support **chatbot**. The project's **scope** includes core **feature development**, academic **system integration**, initial **deployment**, and **training**, with future **enhancements** planned for **alumni engagement** and **AI-driven career insights**.

The **platform** supports **UFS's** strategic **mission** to prepare **students** for the **workforce** and provide **employers** with qualified **talent**, establishing a scalable, secure, and future-ready digital **recruitment ecosystem**.

Table

NOTE: Greyed out nouns are short-lived (Not candidate entities)

Nouns	
Concrete Nouns	Abstract Nouns
<ul style="list-style-type: none">● University Of the Free State● Platform● Bridge● Databases● Updates● System● Application● User● Students● Profiles● CVs● Skills● Portfolio● Portal● Employers● Tools● Records● Consent● Administrators● Dashboards● Roles● Chatbot● Interview	<ul style="list-style-type: none">● Development● Recruitment● Employability● Engagement● Performance● Opportunities● Recommendations● Scheduling● Access● Analysis● Scope● Integration● Deployment● Training● Enhancements● Mission● Workforce● Talent● Ecosystem● Insight

CRC Cards

Class name: User
Responsibilities
<ul style="list-style-type: none">• Represent a generic system user (student, employer, or admin).• Handle login, authentication, and access rights.
Collaborator
<ul style="list-style-type: none">• Student• Employer• Administrator

Figure 36: User CRC Card

Class name: Student
Responsibilities
<ul style="list-style-type: none">• Manage personal profile, CV, and skills.• Apply for jobs and receive notifications.• Build a portfolio to showcase projects.
Collaborator
<ul style="list-style-type: none">• CV• Portfolio• Skills• Application• Interview• Chatbot

Figure 37: Student CRC Card

Class name: Employer
Responsibilities
<ul style="list-style-type: none">• Post jobs and internships.• Review student applications and schedule interviews.
Collaborator
<ul style="list-style-type: none">• Application• Interview• Administrator

Figure 38: Employer CRC Card

Class name: Administrator
Responsibilities
<ul style="list-style-type: none">• Monitor analytics and job placement rates.• Ensure compliance and manage platform settings.
Collaborator
<ul style="list-style-type: none">• Analytics Dashboard• Employer• Student• Chatbot

Figure 39: Administrator CRC Card

Class name: Application

Responsibilities
<ul style="list-style-type: none"> • Record job applications submitted by students. • Track status (pending, shortlisted, rejected, hired). • Link to job postings and interviews.
Collaborator
<ul style="list-style-type: none"> • Student • Employer • Interview • CV

Figure 40: Application CRC Card

Class name: CV
Responsibilities
<ul style="list-style-type: none"> • Store structured resume details. • Auto-populate applications. • Update and reflect academic/skill data.
Collaborator
<ul style="list-style-type: none"> • Student • Application • Employer

Figure 41: CV CRC Card

Class name: Skills
Responsibilities
<ul style="list-style-type: none"> • Store and categorize competencies (technical, soft skills, certifications). • Match skills to job requirements.
Collaborator
<ul style="list-style-type: none"> • Student • CV • Job/Opportunity

Figure 42: Skills CRC Card

Class name: Portfolio
Responsibilities
<ul style="list-style-type: none"> • Showcase student projects, certifications, and achievements. • Enhance job applications by demonstrating capabilities.
Collaborator
<ul style="list-style-type: none"> • Student • Employer

Figure 43: Portfolio CRC Card

Class name: Tools

Responsibilities
<ul style="list-style-type: none"> • Provide system functionality such as CV builder, job matching, and scheduling. • Enhance student and employer interactions with the platform.
Collaborator
<ul style="list-style-type: none"> • Student • Employer • Chatbot • Application

Figure 44: Tools CRC Card

Class name: <i>Chatbot</i>
Responsibilities
<ul style="list-style-type: none"> • Provide user assistance and FAQs. • Guide students and employers through processes.
Collaborator
<ul style="list-style-type: none"> • Student • Employer • Administrator

Figure 45: Chatbot CRC Card

Class name: <i>Interview</i>
Responsibilities
<ul style="list-style-type: none"> • Store interview details (date, time, mode). • Notify participants and update application status.
Collaborator
<ul style="list-style-type: none"> • Student • Employer • Application

Figure 46: Interview CRC Card

Pseudo Code

```
// -----  
// BASE CLASS  
// -----  
  
CLASS User  
  
    // Attributes  
  
    id : String  
  
    name : String  
  
    role : String      // 'Admin' | 'Student' | 'Employer'  
  
    email : String  
  
  
    // Methods  
  
    login() : Boolean  
  
    logout() : void  
  
END CLASS
```

```
// -----
```

```
// USER SUBCLASSES
```

```
// -----
```

```
CLASS Admin INHERITS User
```

```
    // Attributes  
  
    adminId : String  
  
    permissions : List<String>  
  
    systemLogs : List<String>
```

```
    // Methods
```

```
        manageUser(user : User) : void  
  
        generateReport(params : Map) : List<String>  
  
        enforceCompliance() : void
```

```
END CLASS
```

```
CLASS Student INHERITS User
```

```

// Attributes

studentId : String
skills : List<Skill>
preferences : List<String>
applications : List<Application>
cv : CV
portfolio : Portfolio
scheduledInterviews : List<Interview>

// Methods

viewApplications() : List<Application>
viewProfile() : Map
updateProfile(profileData : Map) : void
applyToJob(job : Job) : Application
withdrawApplication(appId : String) : void
viewScheduledInterviews() : List<Interview>

END CLASS

CLASS Employer INHERITS User

// Attributes

employerId : String
companyName : String
jobListings : List<Job>

// Methods

postJobListing(job : Job) : void
updateJobListing(jobId : String, updates : Map) : void
reviewApplications(jobId : String) : List<Application>
scheduleInterview(applicationId : String, date : Date) : Interview

END CLASS

// -----

```

```

// CORE SYSTEM ENTITIES

// -----
CLASS Job

// Attributes
    jobId : String
    title : String
    description : String
    employer : Employer    // association
    location : String
    salary : Double
    deadlineDate : Date
    requiredSkills : List<Skill>
    postDate : Date
    status : String      // 'Open' | 'Closed'

// Methods
    addApplication(application : Application) : void
    closePosting() : void
    updateDetails(updates : Map) : void
    matchCandidates(students : List<Student>) : List<Student>

END CLASS

CLASS Application

// Attributes
    applicationId : String
    submissionDate : Date
    status : String      // 'Submitted' | 'Under Review' | 'Shortlisted' | 'Interview' | 'Rejected' |
    'Hired' | 'Withdrawn'
    job : Job
    applicant : Student
    cvSnapshot : CV

// Methods

```

```
updateStatus(newStatus : String) : void  
withdraw() : void  
getDetails() : Map  
END CLASS
```

```
CLASS Interview  
// Attributes  
interviewId : String  
dateScheduled : Date  
interviewer : Employer // association  
interviewee : Student // association  
mode : String // 'Online' | 'In-Person'  
status : String // 'Scheduled' | 'Completed' | 'Cancelled'
```

```
// Methods  
scheduleInterview(date : Date) : void  
sendInterviewInvite() : Boolean  
reschedule(newDate : Date) : void  
updateInterviewStatus(status : String) : void  
END CLASS
```

```
CLASS Skill  
// Attributes  
skillId : String  
name : String  
category : String // e.g., 'Technical', 'Soft', 'Certification'  
level : String // 'Beginner' | 'Intermediate' | 'Expert'  
  
// Methods  
validateSkill() : Boolean  
matchToJob(job : Job) : Boolean  
END CLASS
```

```
// -----
// SUPPORT & AUXILIARY ENTITIES
// -----
CLASS Chatbot

// Attributes

sessionId : String

status : Boolean

conversationHistory : List<String>

toolAccess : List<String>

// Methods

startConversation(user : User) : void

sendMessage(user : User, message : String) : String

suggestHelpTopics(user : User) : List<String>

endConversation() : void

END CLASS
```

```
CLASS Tools

// Attributes

toolId : String

toolName : String

toolType : String      // e.g., 'CVBuilder', 'Scheduler', 'Matcher'

capabilities : List<String>

// Methods

callTool(user : User, params : Map) : Map

configureToolSettings(settings : Map) : void

accessTool(user : User) : Boolean

END CLASS
```

CLASS CV

```
// Attributes  
  
cvId : String  
  
owner : Student      // association  
  
education : List<String>  
  
experience : List<String>  
  
skills : List<Skill>  
  
achievements : List<String>  
  
  
// Methods  
  
updateEducation(records : List<String>) : void  
  
updateExperience(records : List<String>) : void  
  
addSkill(skill : Skill) : void  
  
export(format : String) : Blob // e.g., 'PDF', 'DOCX'
```

END CLASS

CLASS Portfolio

```
// Attributes  
  
portfolioId : String  
  
owner : Student      // association  
  
projects : List<Map>    // each project: {title, description, uri, date}  
  
certifications : List<String>  
  
  
// Methods  
  
addProject(project : Map) : void  
  
removeProject(projectId : String) : void  
  
displayPortfolio() : Map
```

END CLASS

Class Diagram

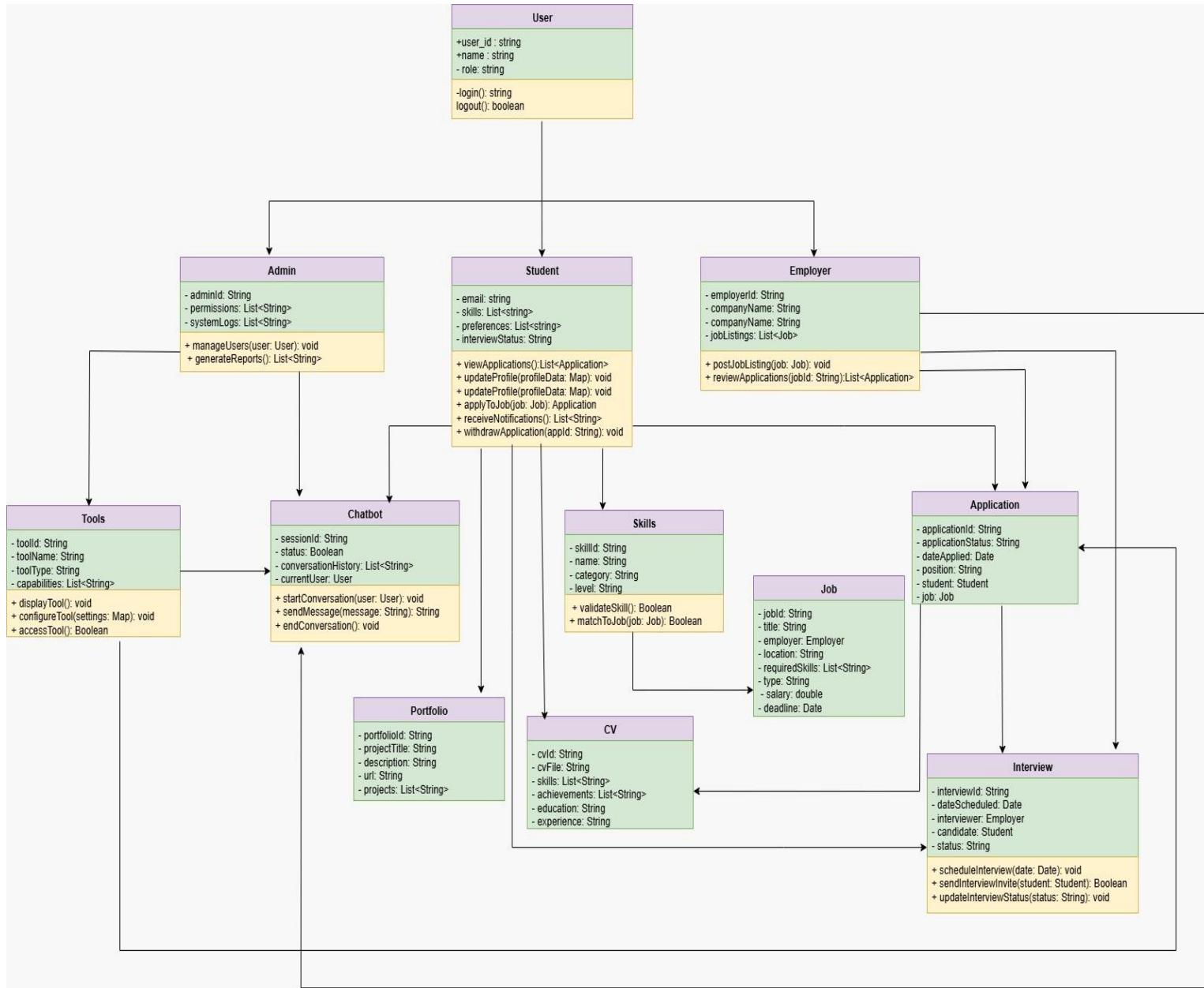


Figure 47: Class Diagram

State Charts

State Charts: Student

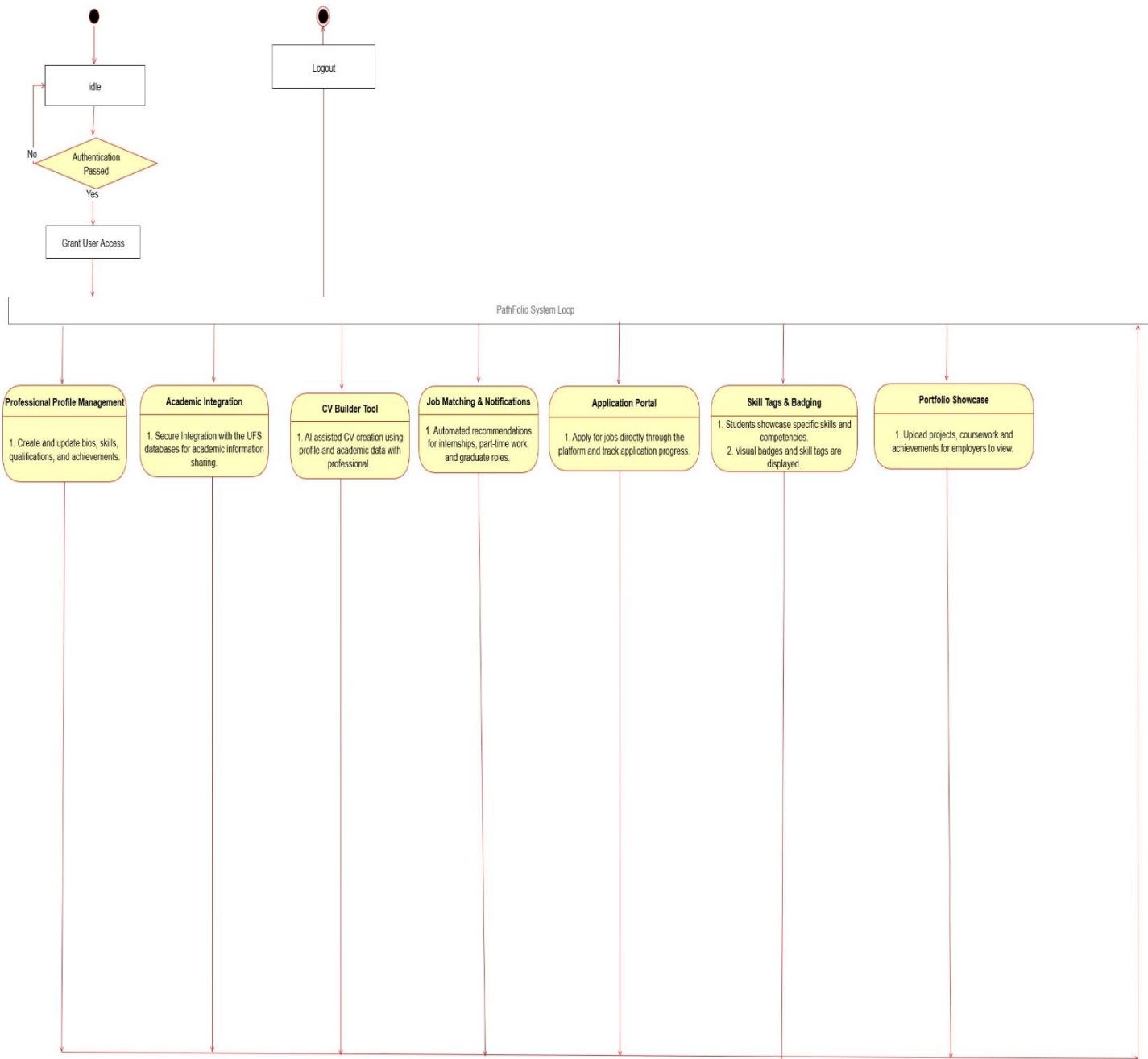


Figure 48: Student State Chart

State Charts: Employer

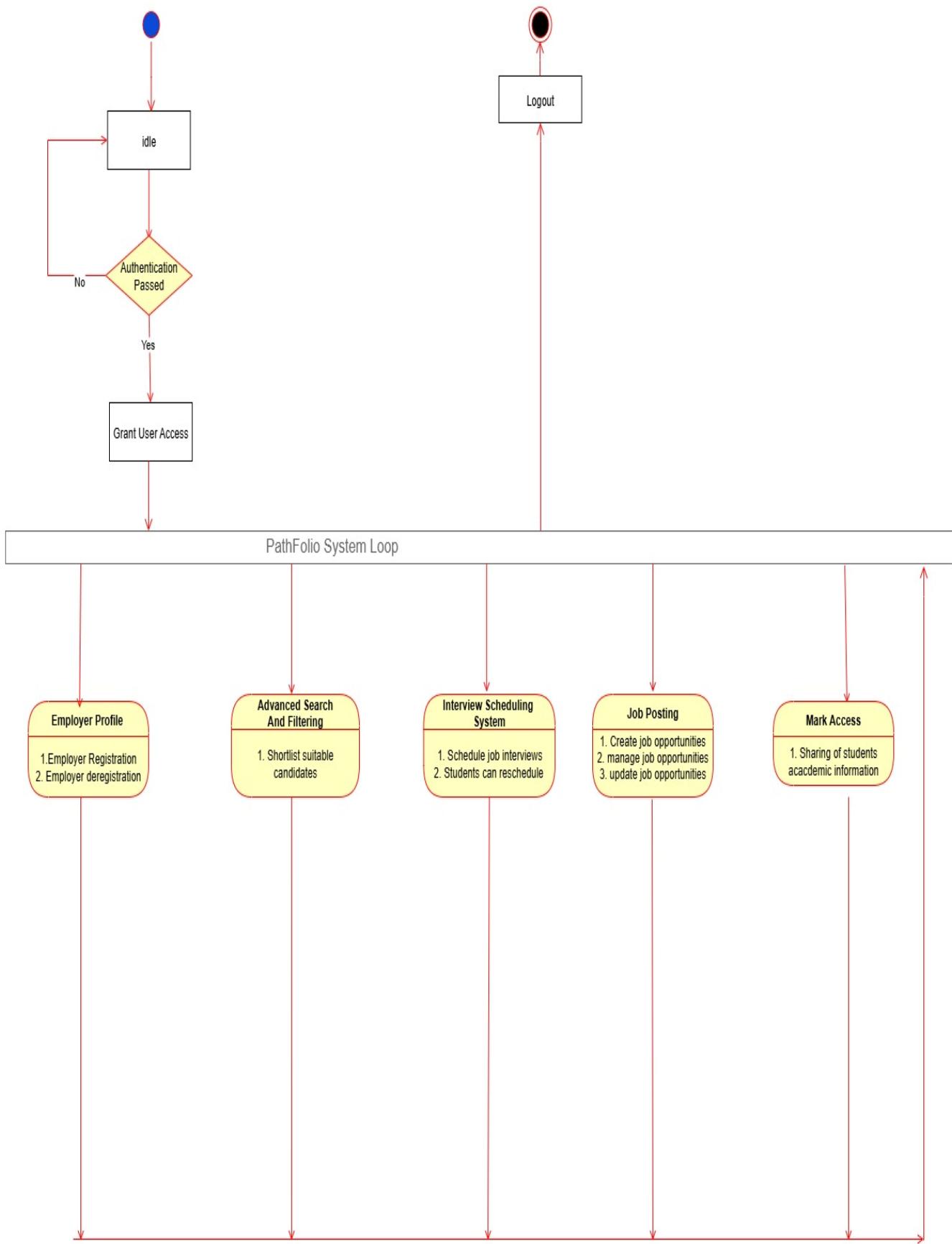


Figure 49: Employers State Chart

State Chart: Administrator

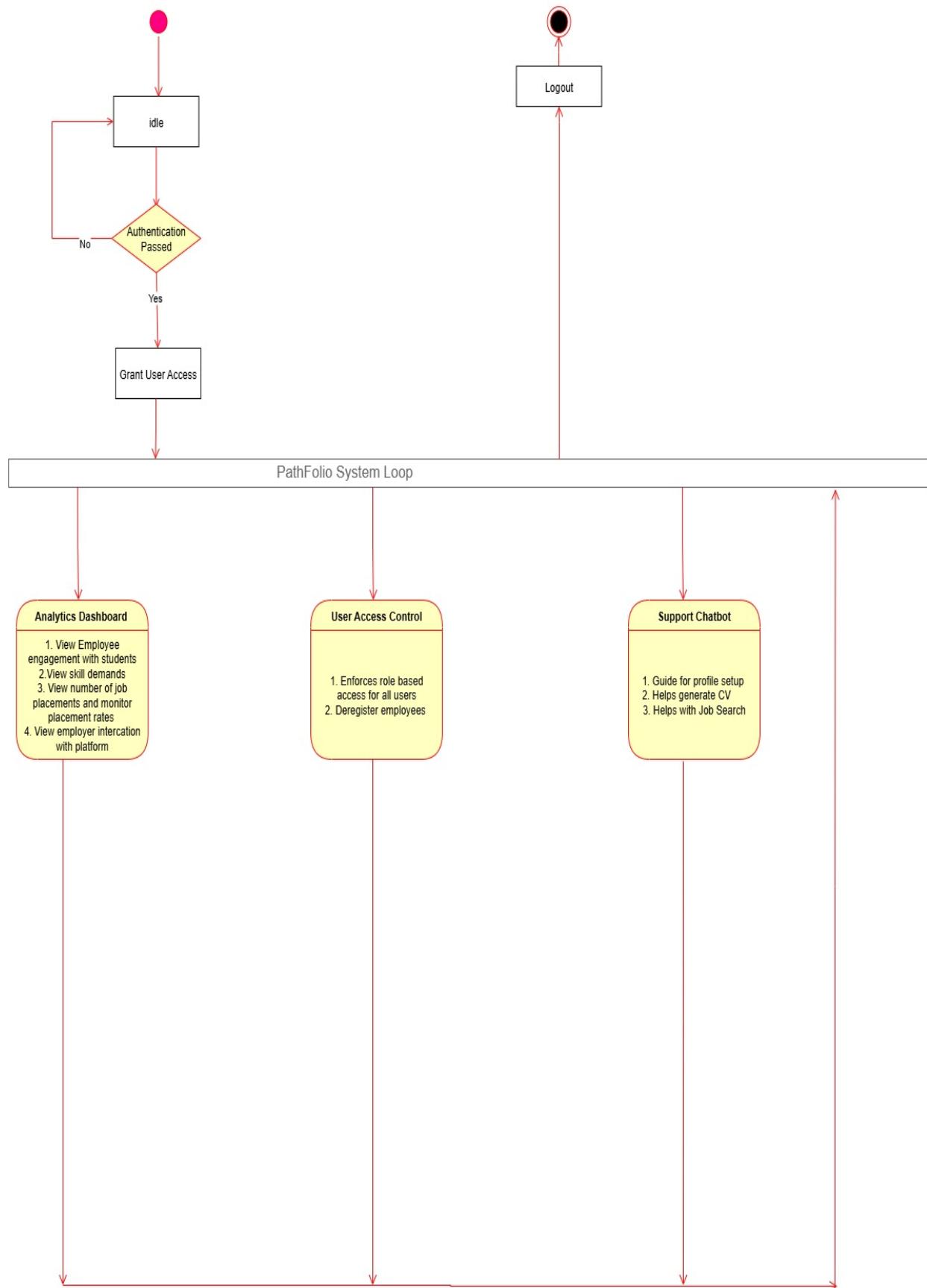
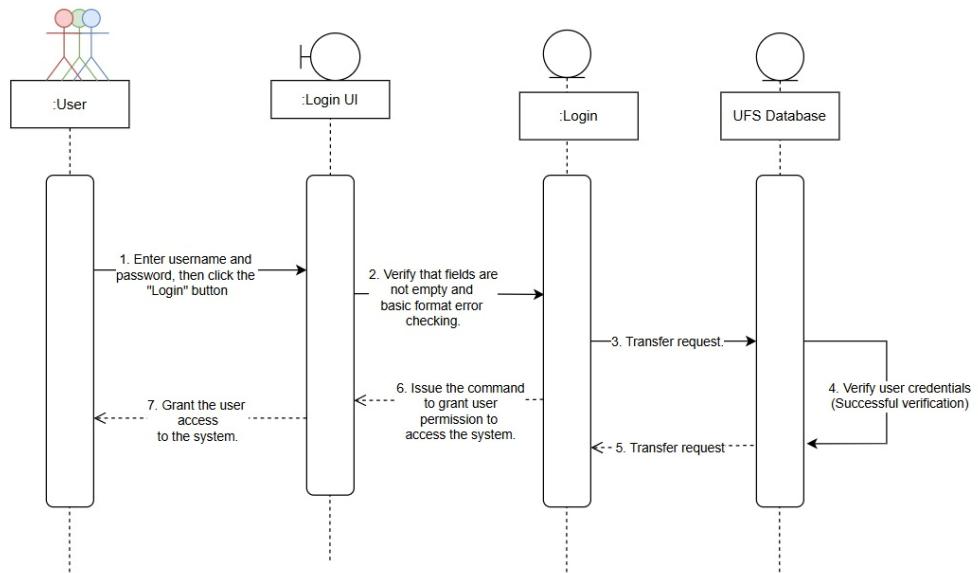


Figure 50: Administrators State Chart

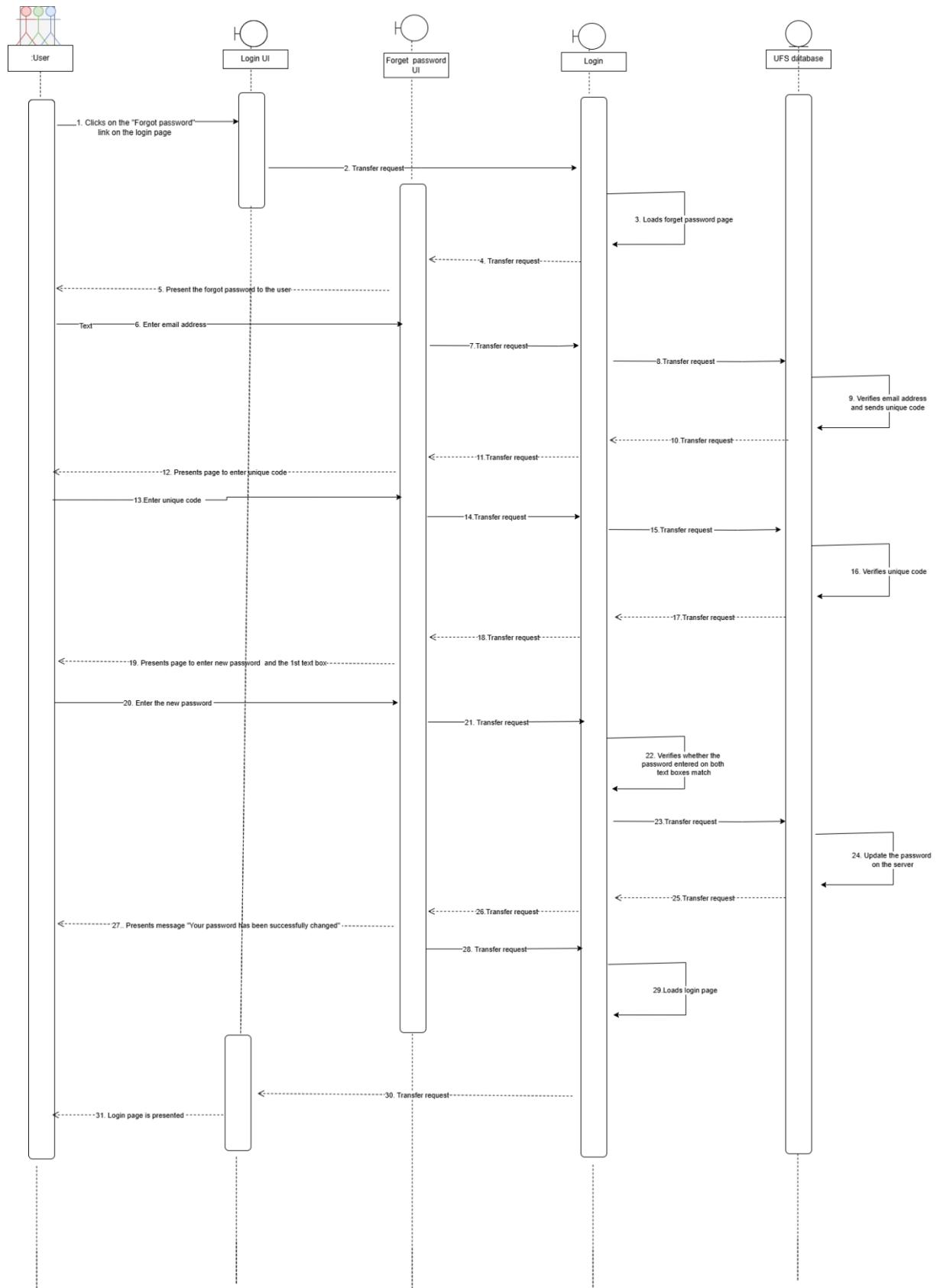
Sequence Diagrams

System Users		
Disclaimer: The extent to which users can interact with system features varies according to their designated access levels and administrative rights.		
Administrator	Employer	Student
		

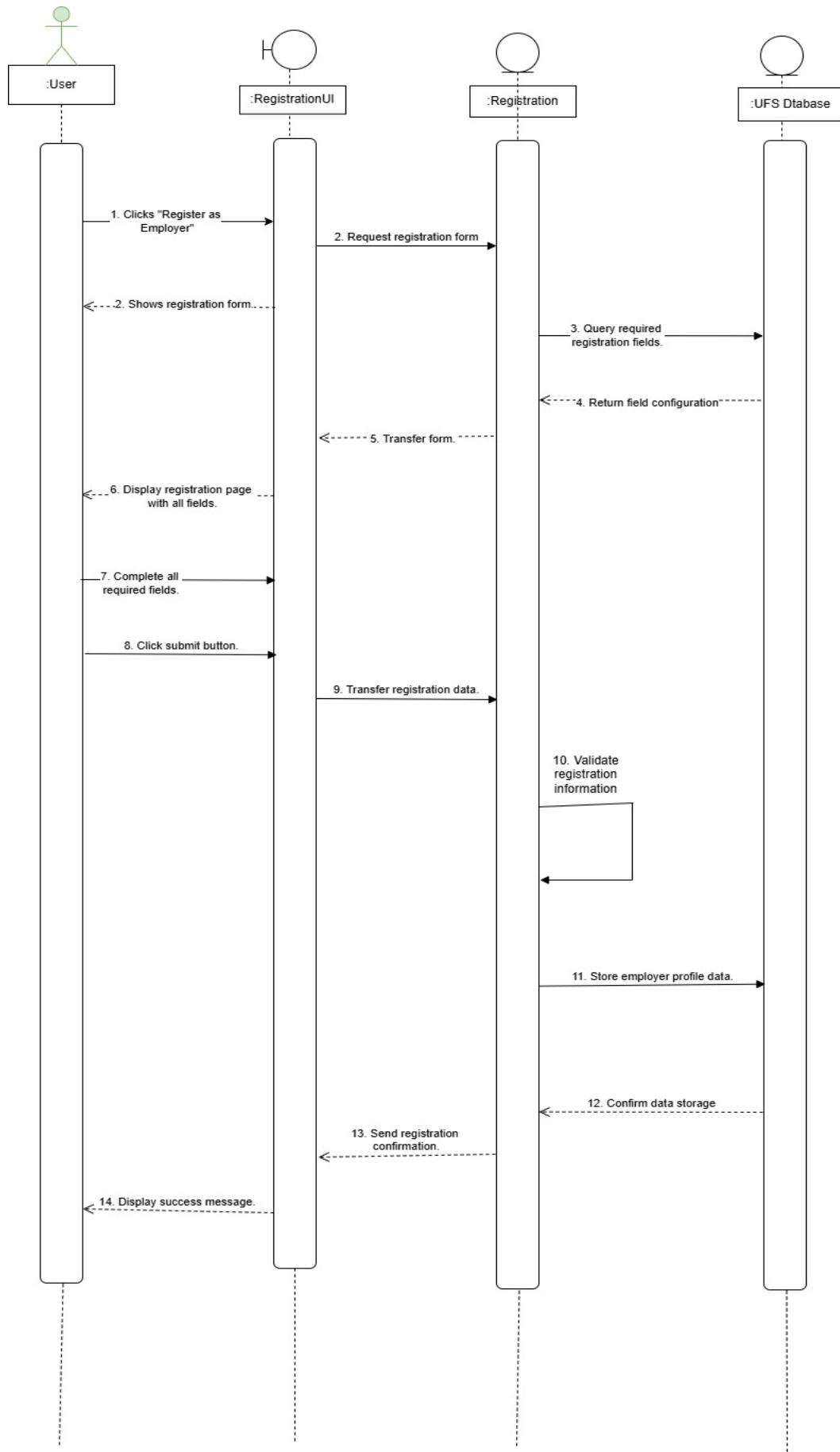
Login



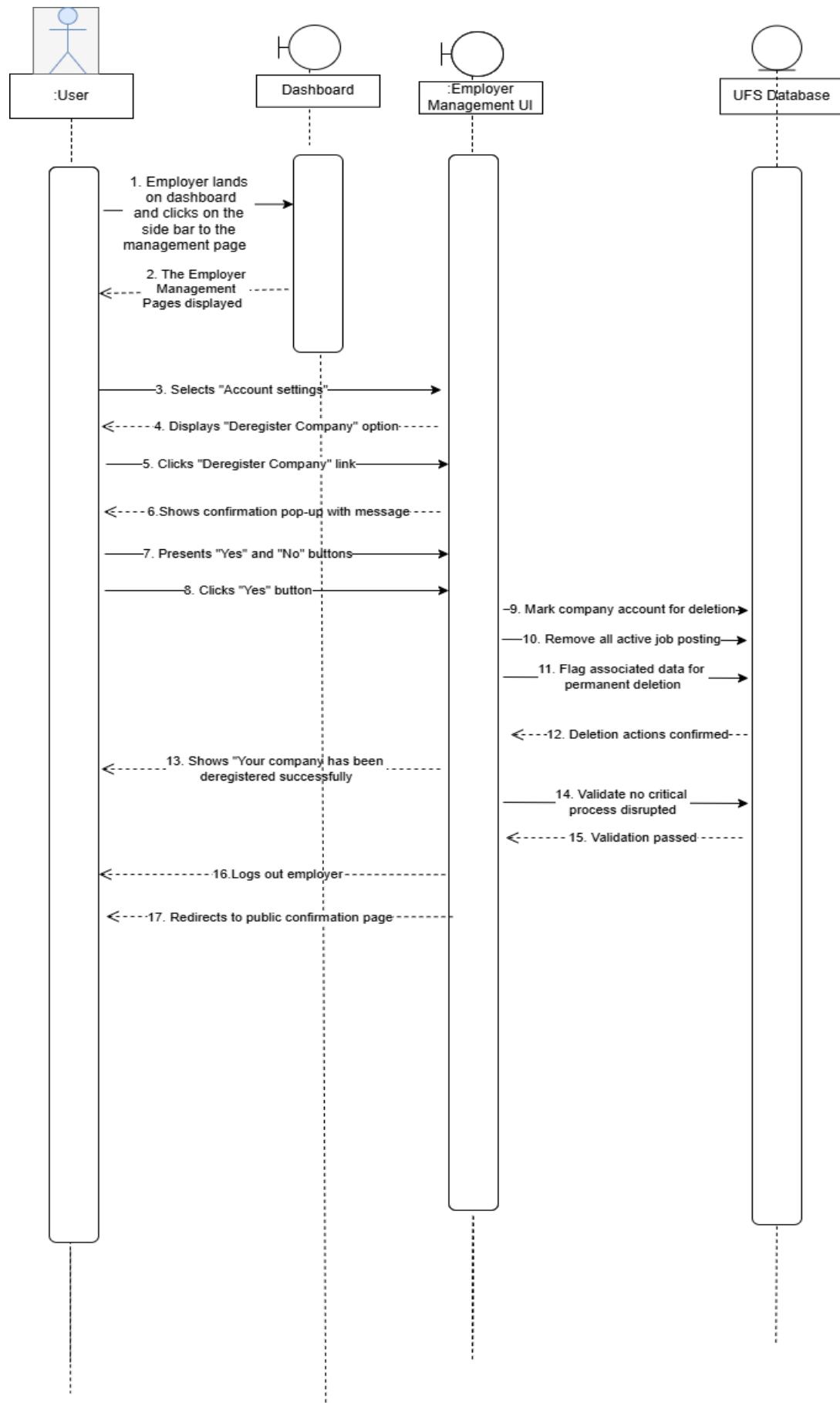
Forget Password



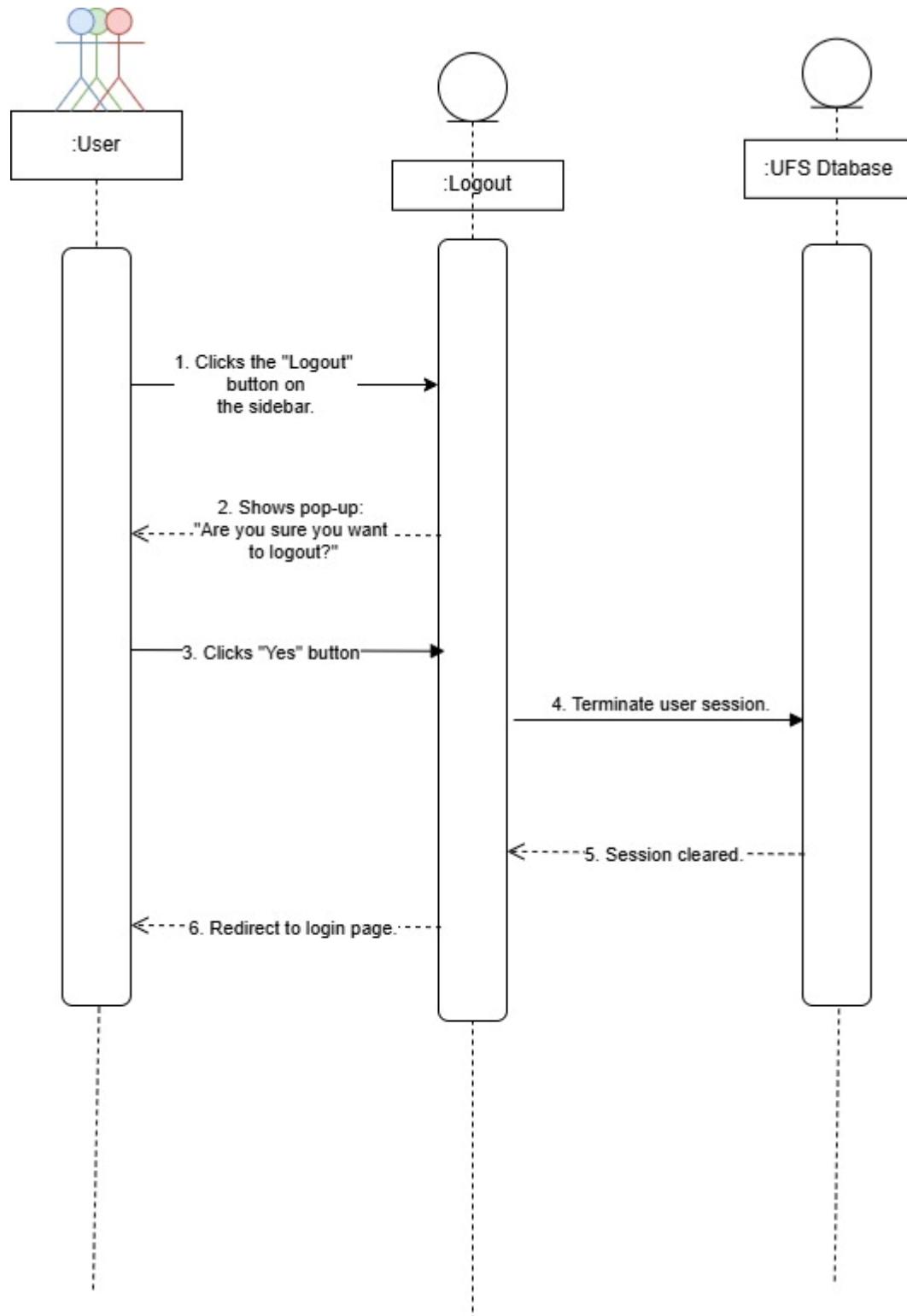
Registration



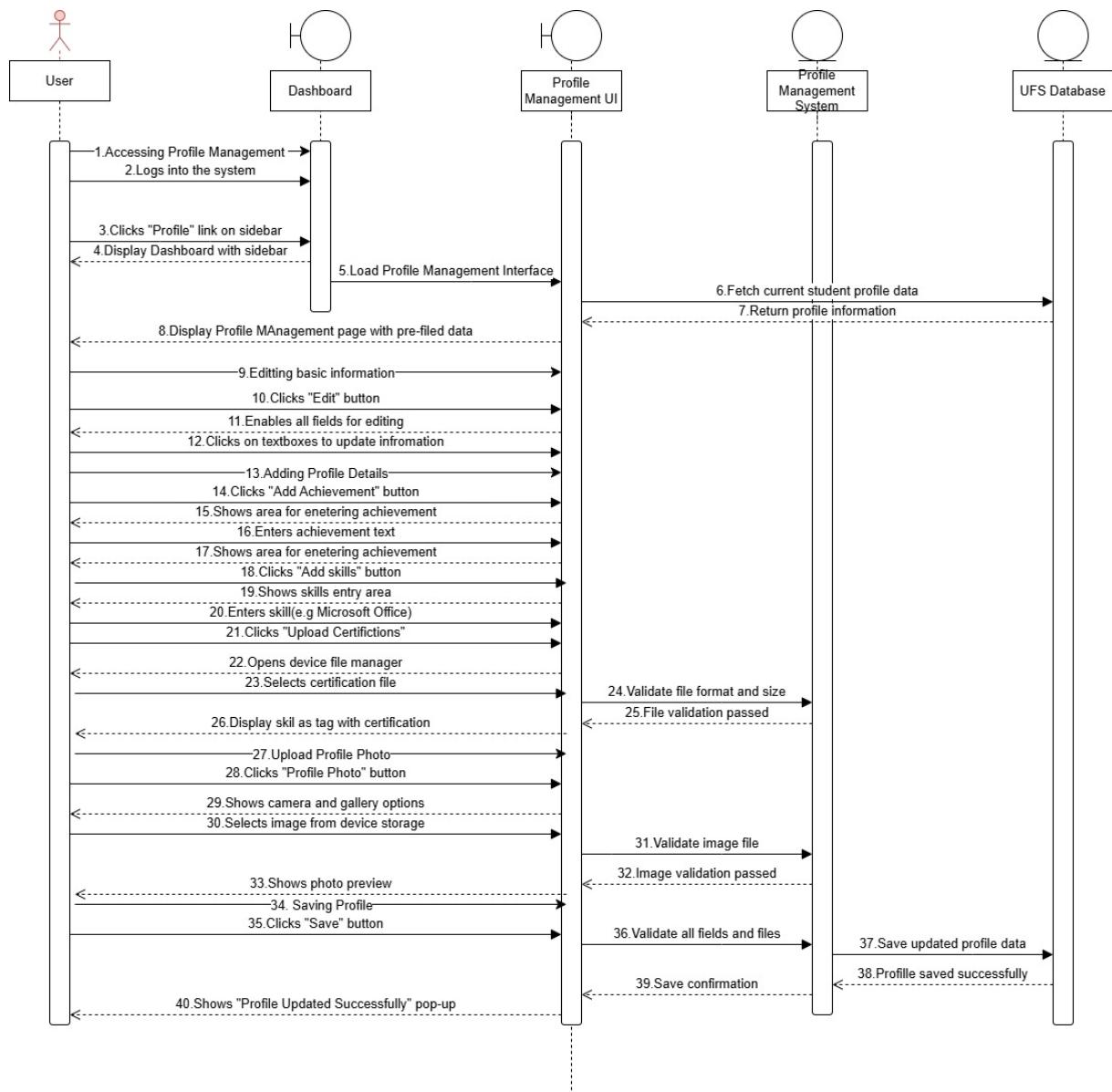
Employer Deregistration



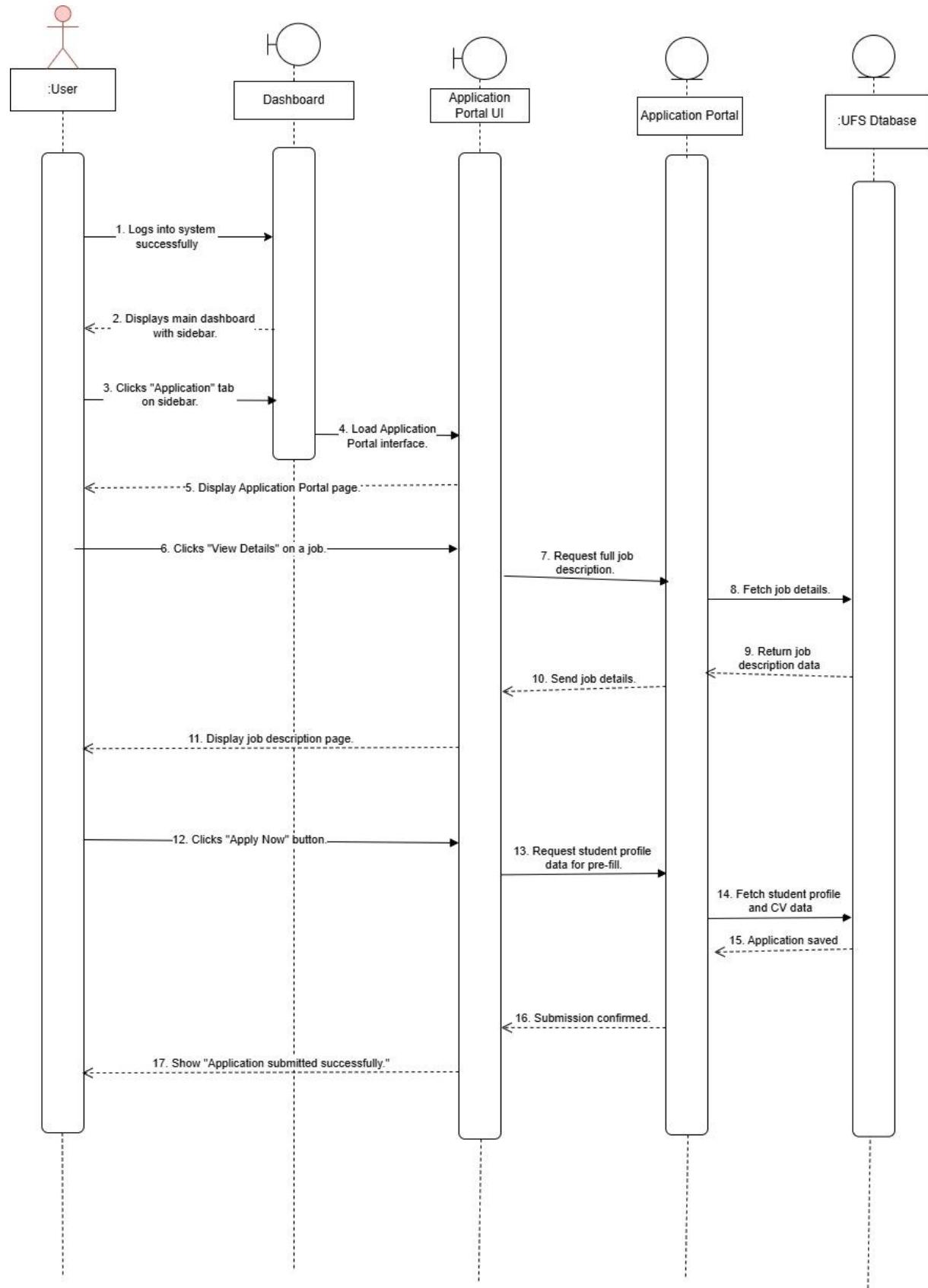
Logout



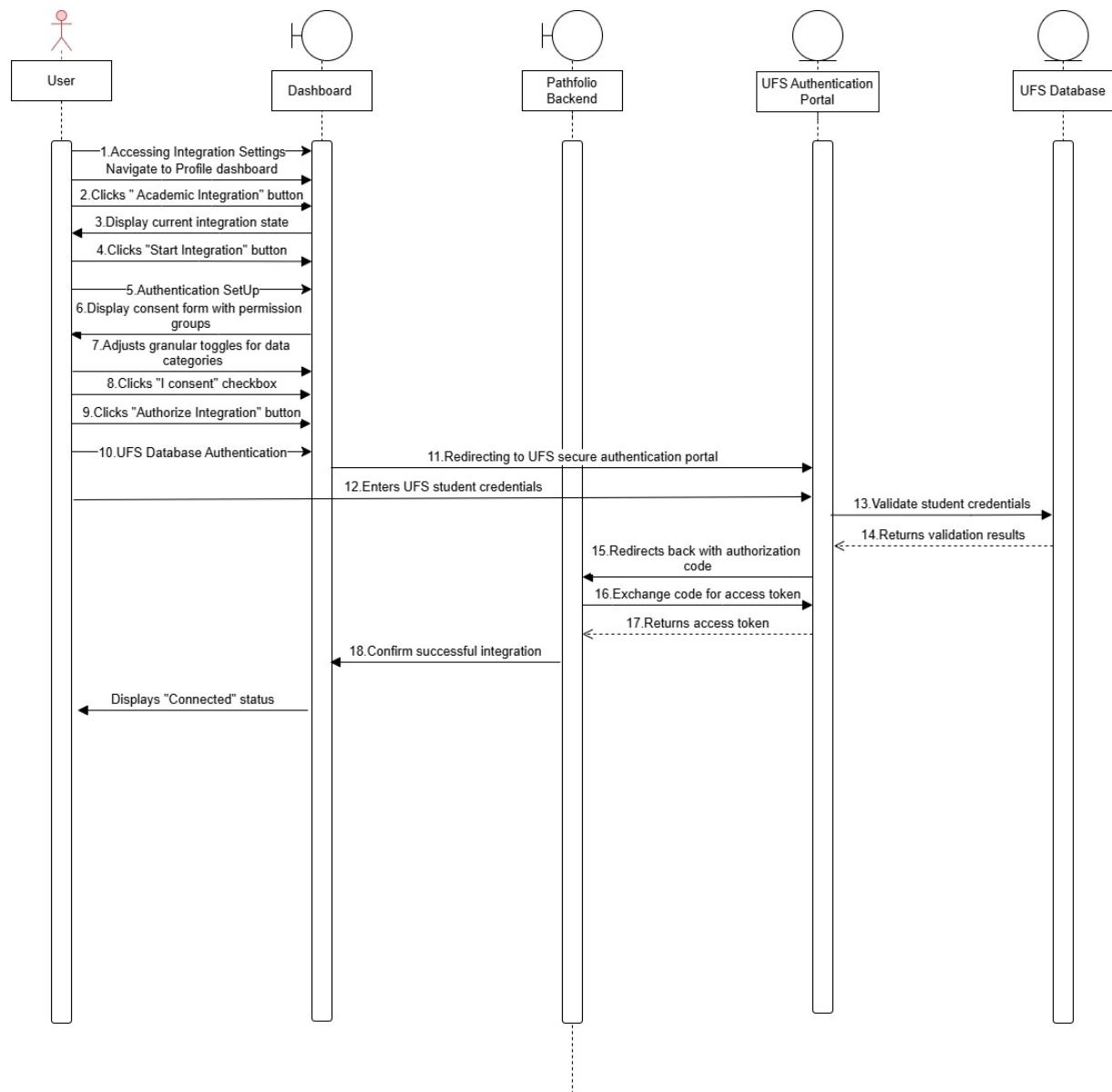
Profile Management



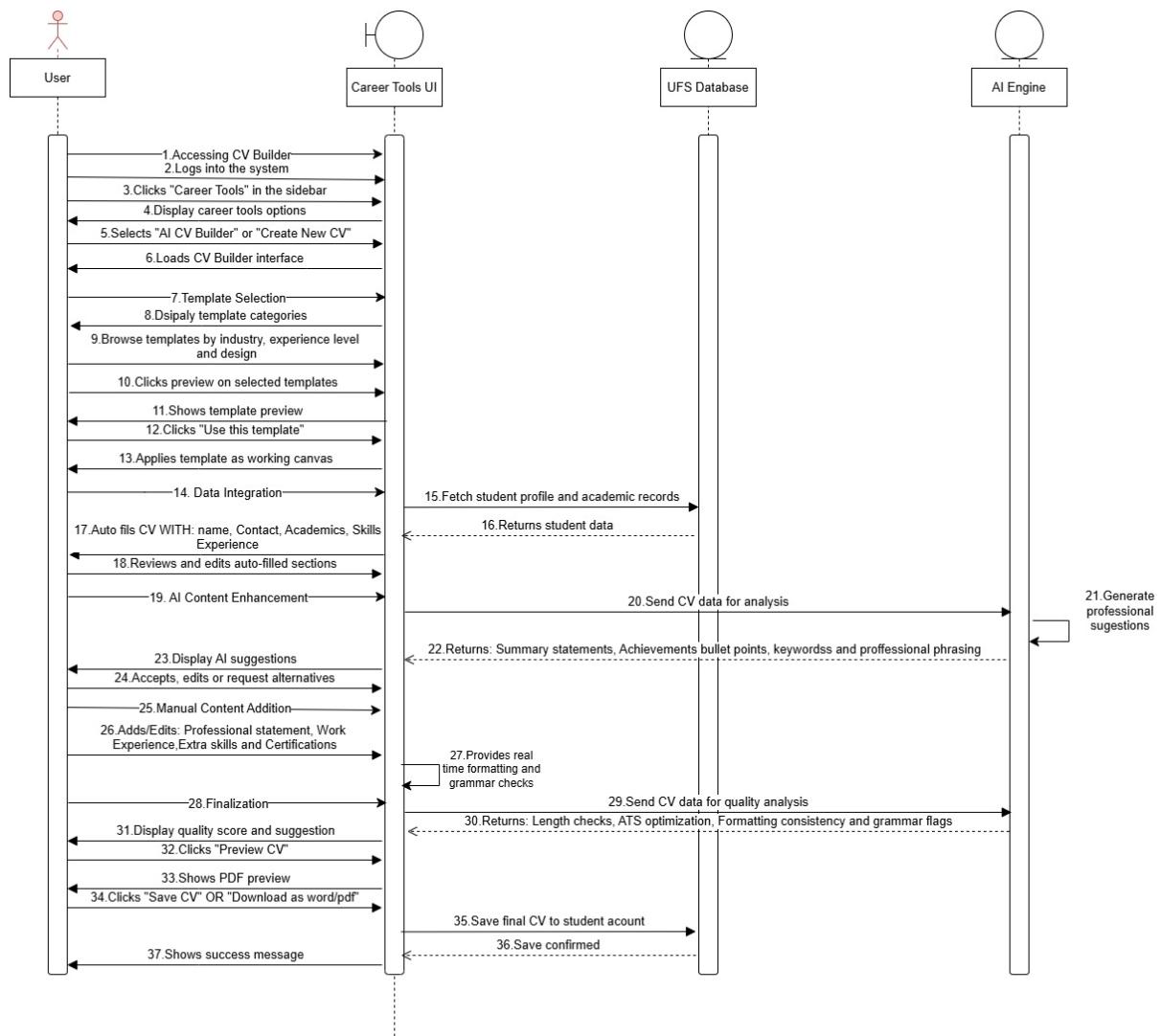
Application Portal



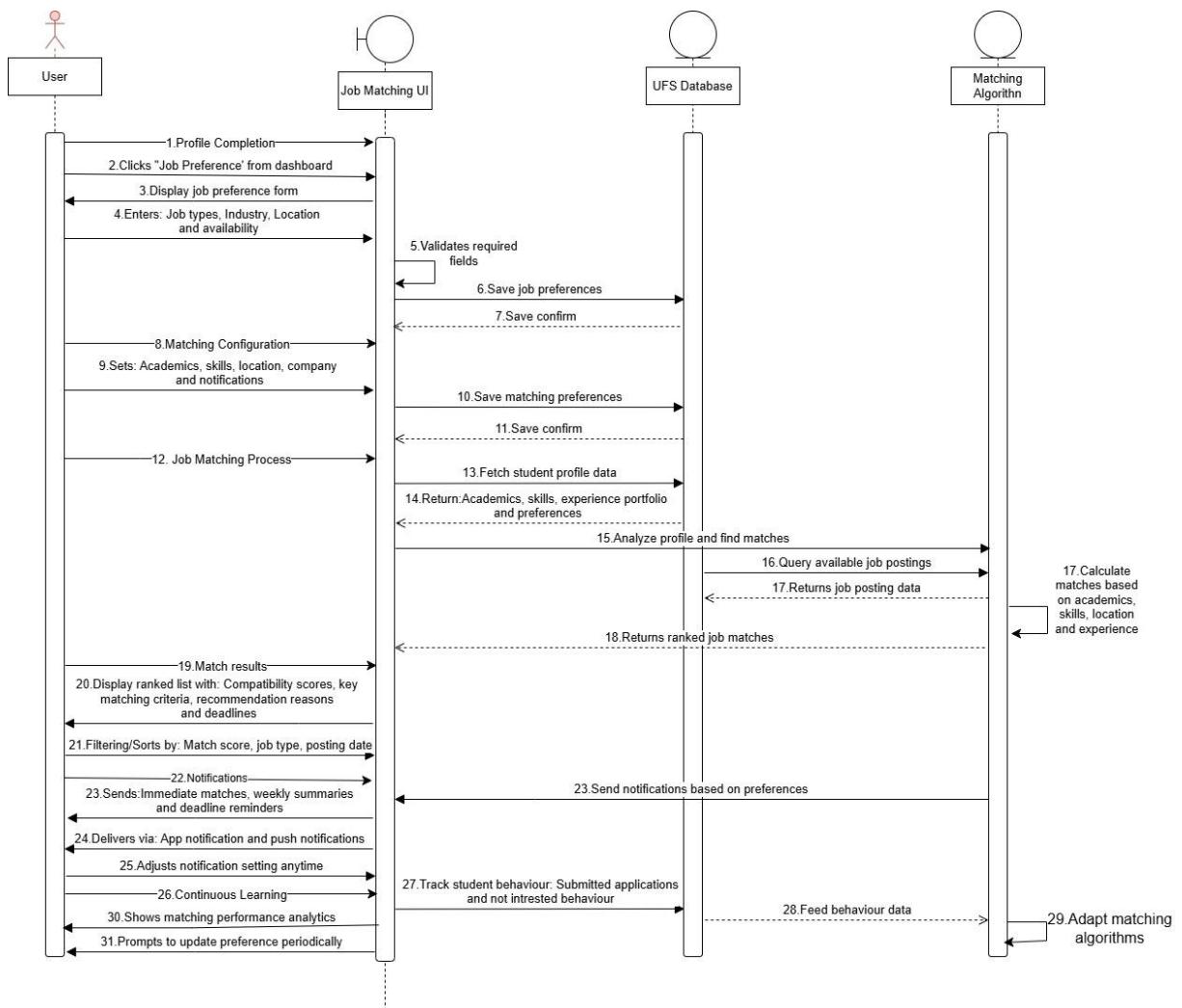
Academic Integration



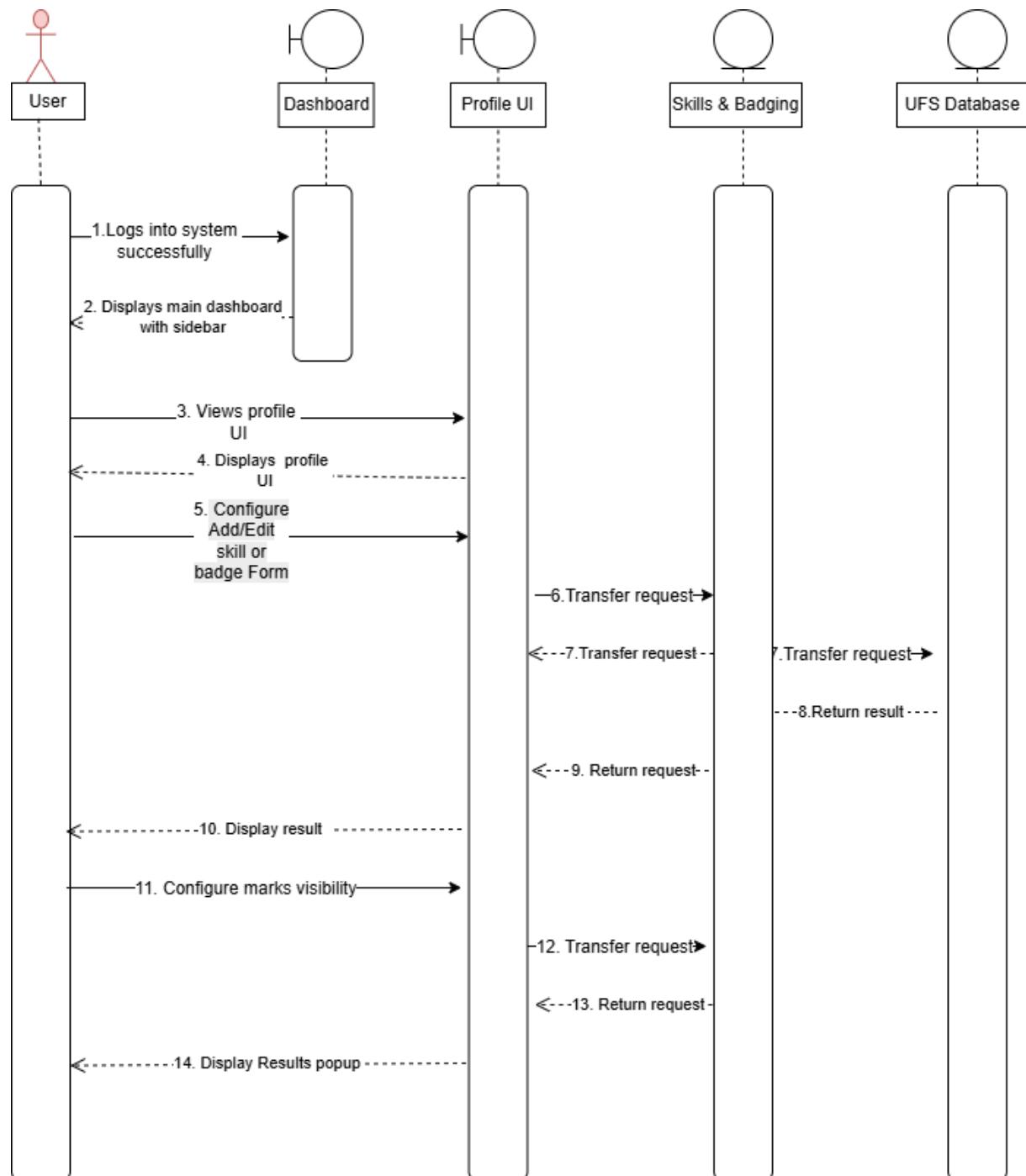
CV Builder



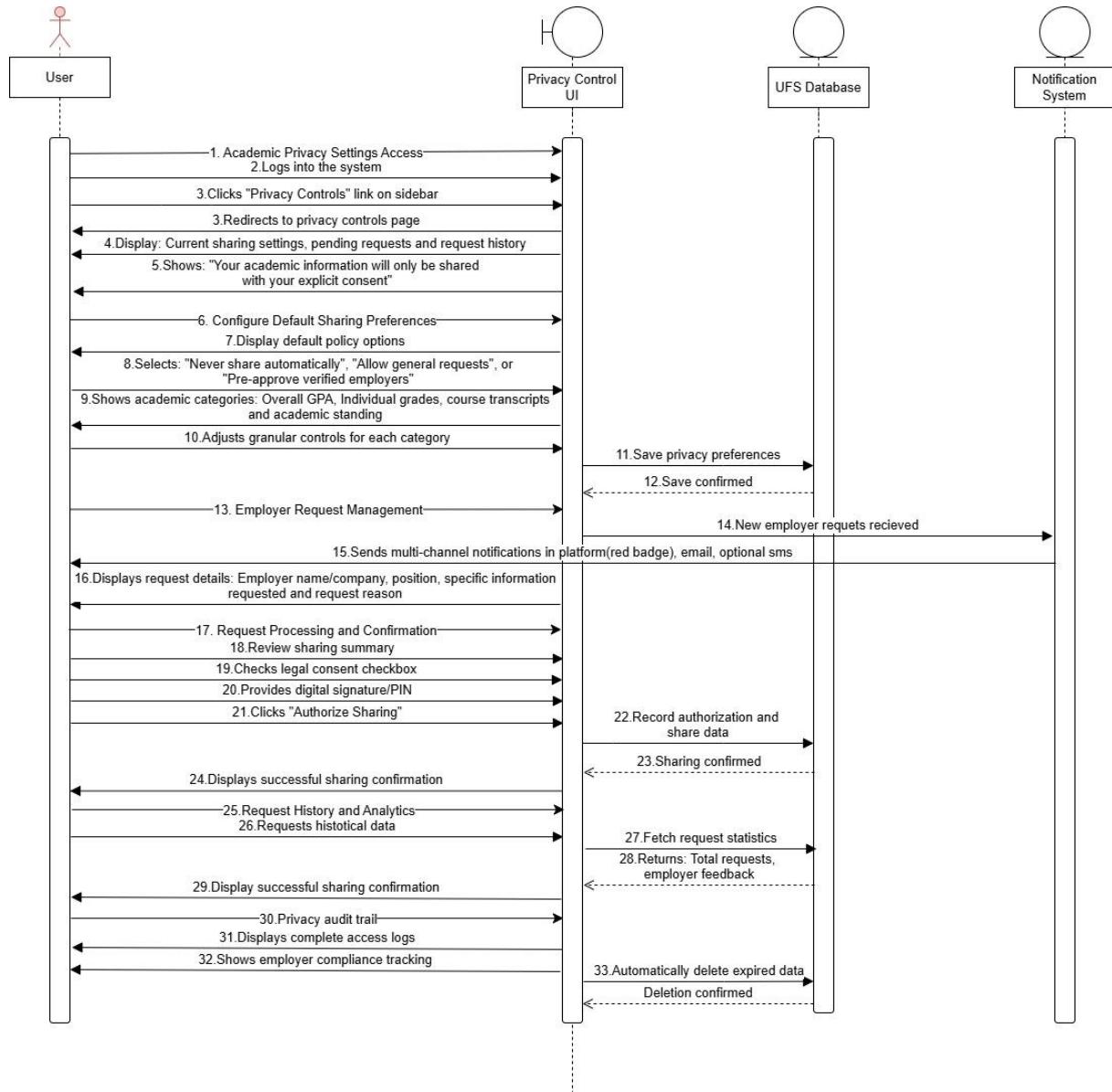
Job Matching



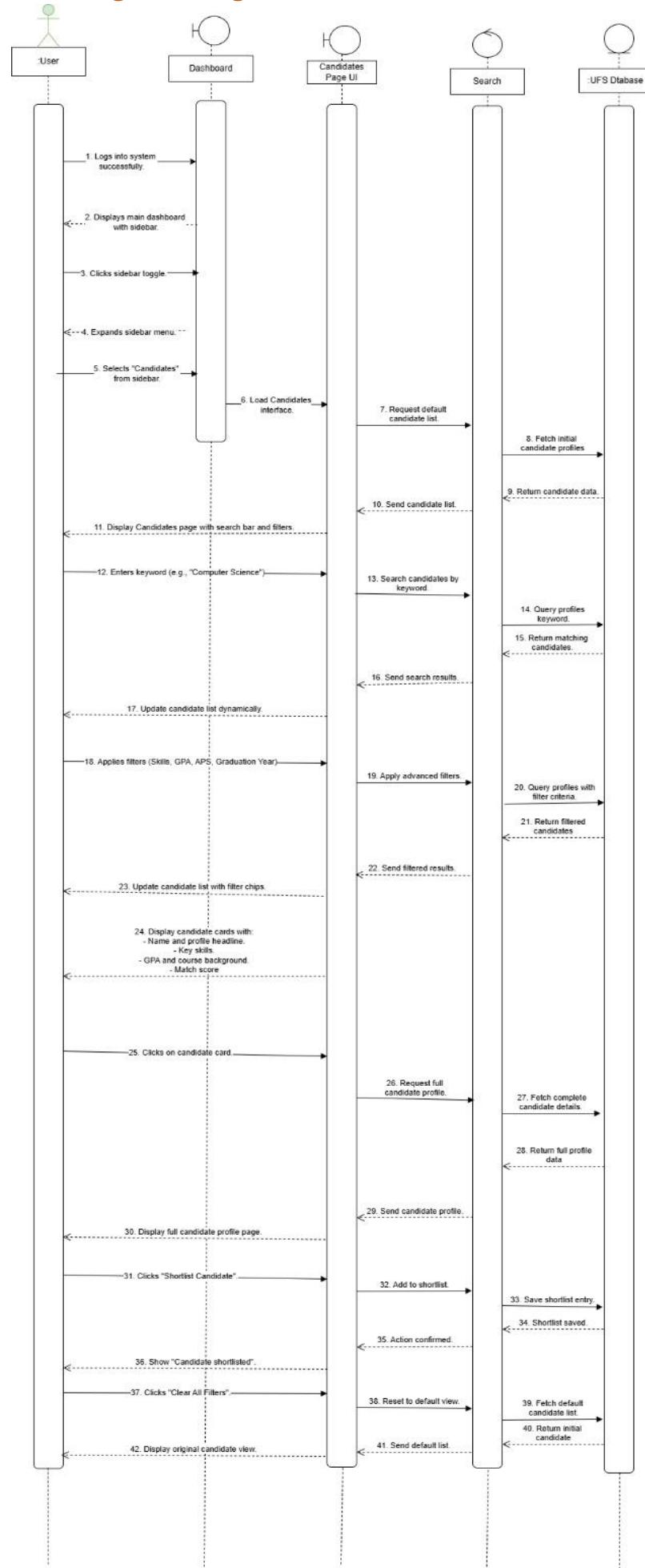
Skills Tags and Badging



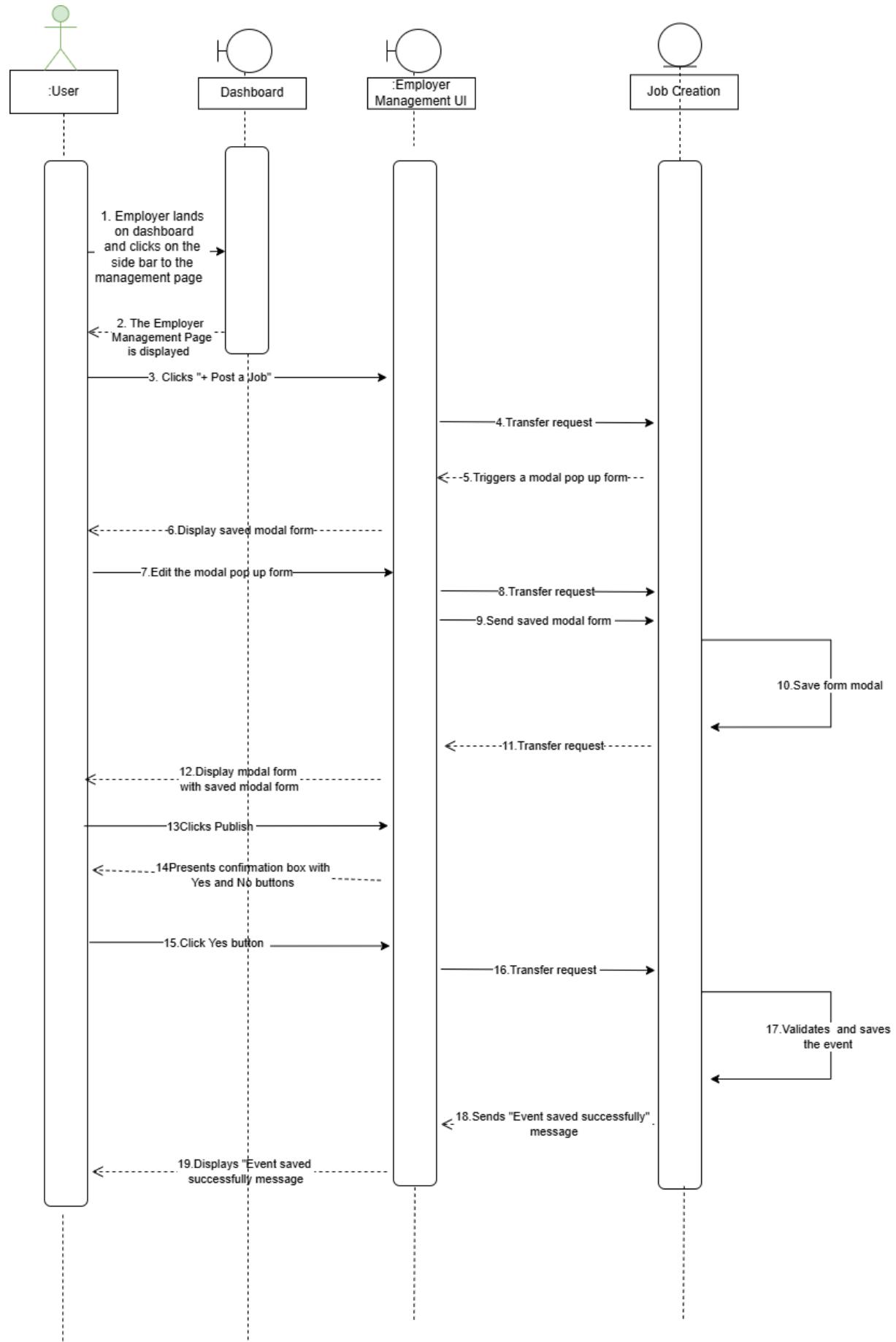
Mark Access



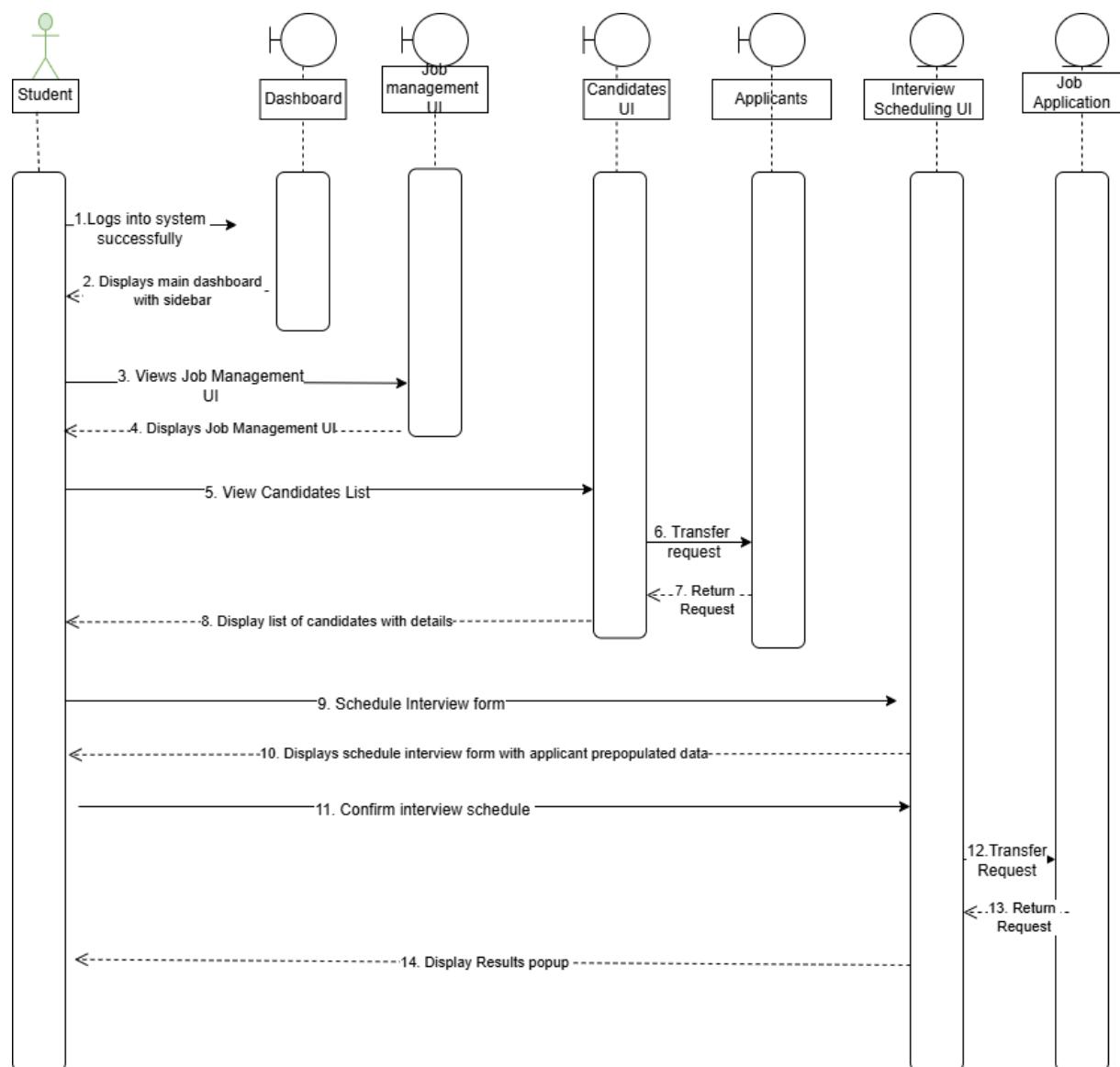
Advanced Searching Filtering



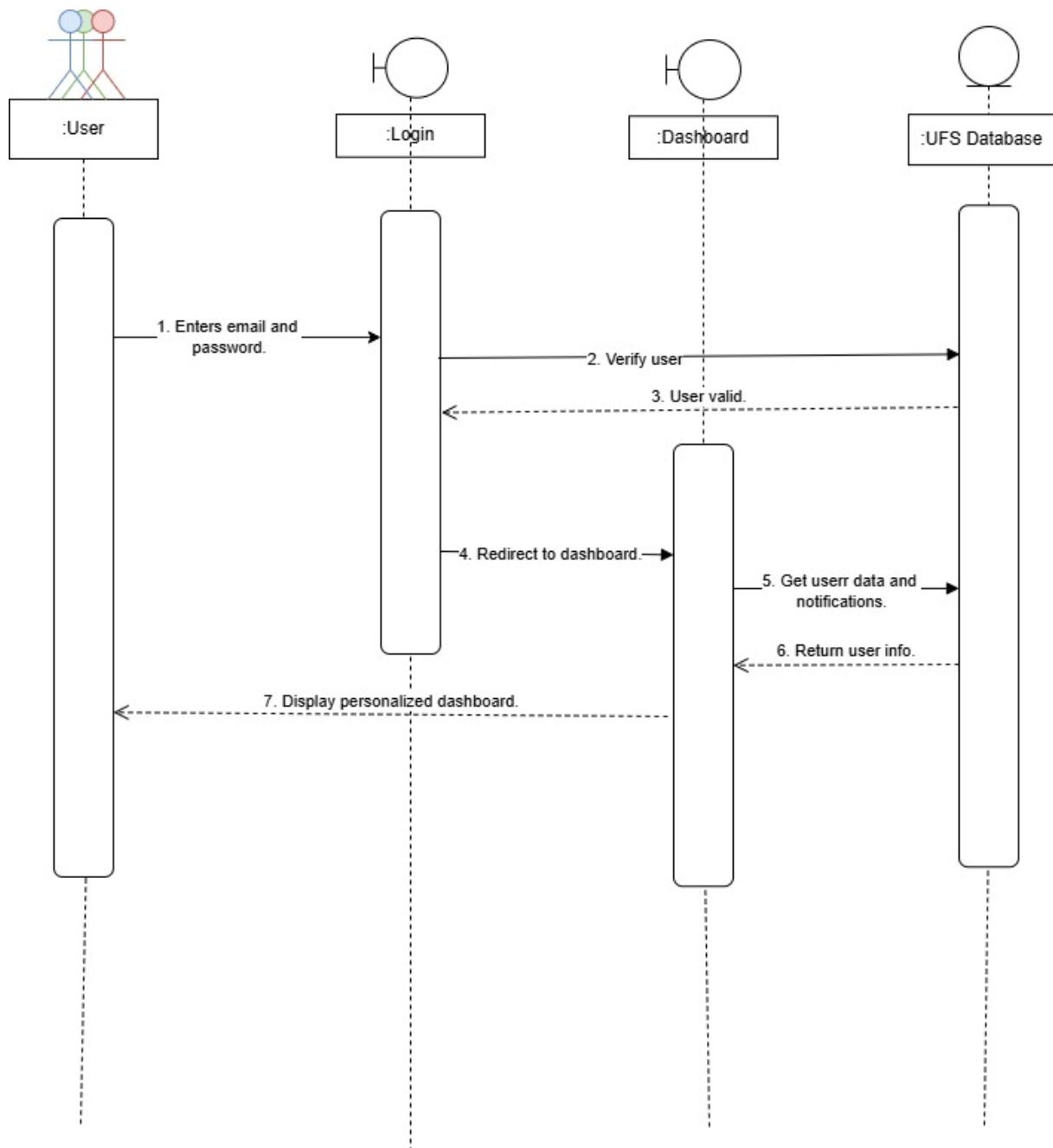
Job Posting



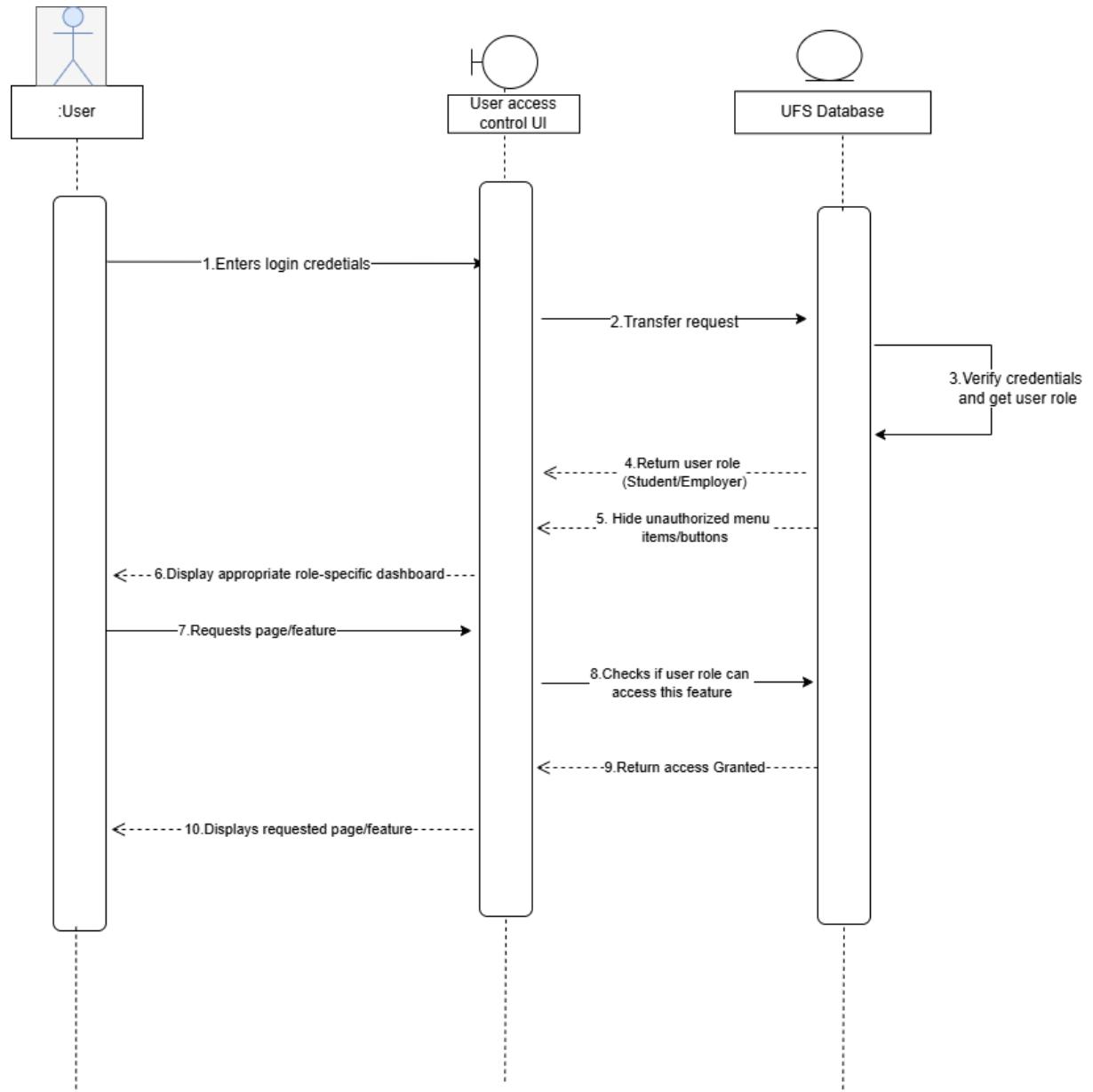
Interview Scheduling



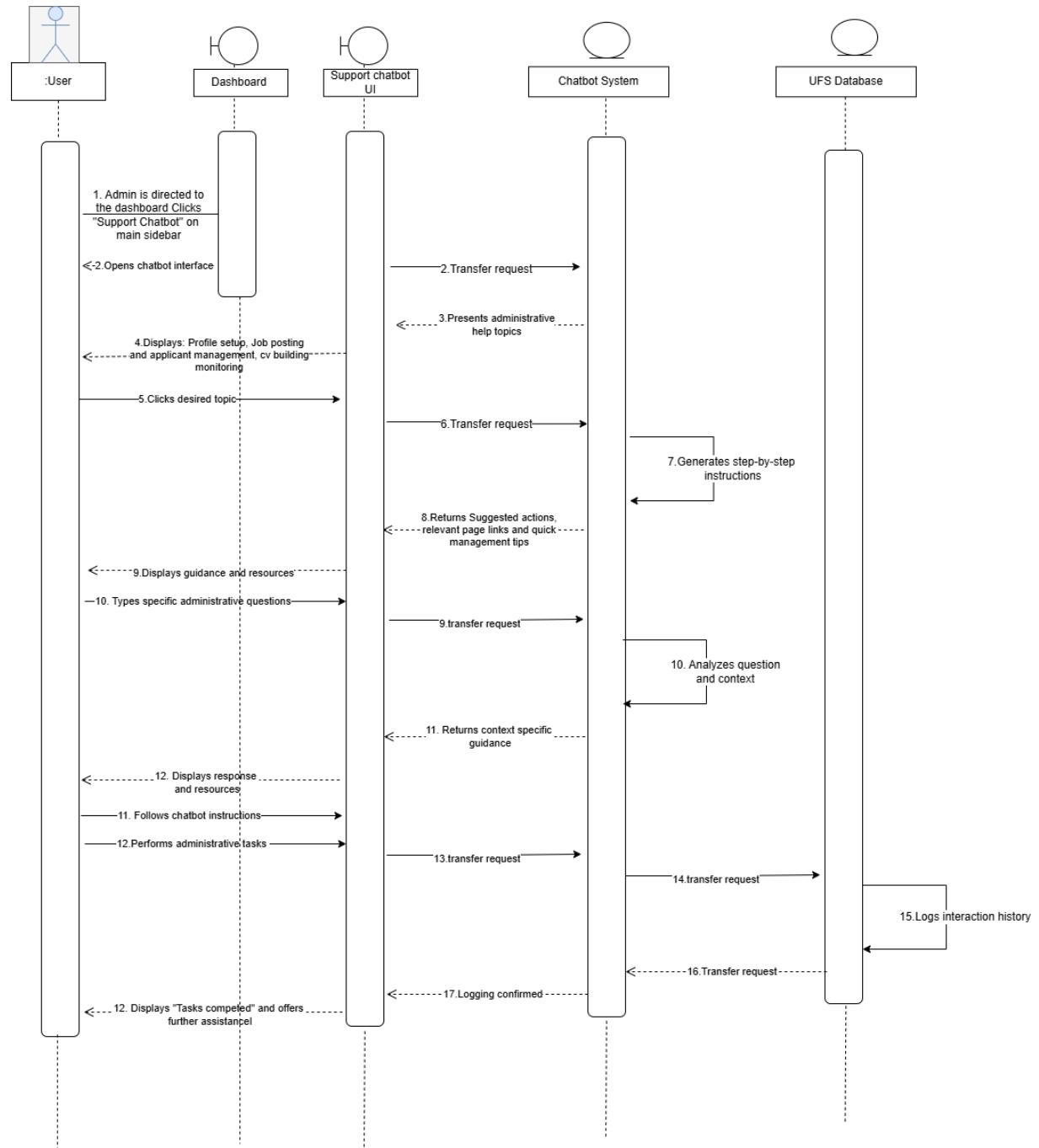
Dashboard



User Access Control



Support Chatbot



UFS Recruitment System - Time Allocation (12-week Project)

Total Duration: 12 weeks (15 hours/week per member × 5 members = 900 hours total)

The table below outlines the distribution of work across all project phases and members.

Names	Requirements	Analysis	Design	Implementation	Testing	TOTAL
Vuyo Nzimande	15	14	17	57	79	182
Mokhotsoa Sepinare	15	14	17	57	79	182
Naledi Dube	14	14	17	56	79	180
Amogelang Seipone	14	13	16	56	79	178
Mpucuko Lutu	14	13	16	56	79	178
TOTALS	72	68	83	282	395	900