



## **ASSIGNMENT - COVER PAGE**

**CT046-3-2-SDM**

**SYSTEMS DEVELOPMENT METHODS**

**APU2F0506IT(FT)/APU2F0506IT(CE)/ APU2F0506IT(BIS)/**

**APU2F0506IT**

**HAND OUT DATE: 21 AUGUST 2025**

**HAND IN DATE: 18 OCTOBER 2025**

**WEIGHTAGE: 50%**

---

### **INSTRUCTIONS TO CANDIDATES:**

- 1 Students are advised to underpin their answers with the use of references (cited using the 7<sup>th</sup> Edition of APA Referencing Style).**
- 2 Late submission will be awarded zero (0) unless Extenuating Circumstances (EC) are upheld.**
- 3 Cases of plagiarism will be penalized.**

## **GROUP ASSIGNMENT**

### **GROUP 35**

**Module Name: System Development Methods**

**Module Code: CT046-3-2**

**Lecture Name: Ainul Hazmin Bin A Hamid**

**Student Name & TP Number:**

<b>No</b>	<b>Name</b>	<b>TP Number</b>
1	Chan Min Huey	TP083261
2	Cheong Pei Enn	TP081926
3	Ng Xiang Ling	TP082046
4	Woo May Eng	TP082001
5	Yap Li Shan	TP080968
6	Kang Hong Qian	TP081205

# Table of Contents

1.0 Planning Phase – Product Backlog & System Requirements .....	6
1.1 Introduction .....	6
1.1.1 Stakeholders.....	6
1.1.2 Users .....	7
1.1.3 Current Business Process.....	7
1.2 Problem Statement .....	8
1.3 Proposed System .....	10
1.4 Scopes and Objectives.....	12
1.4.1 Project Scopes.....	12
1.4.2 Project Objectives .....	13
1.5 Project Scheduling.....	14
2.0 Analysis Phase – Task Engineering & System Analysis .....	15
2.1 Product Backlog Creation .....	15
2.1.1 Product Backlog .....	15
2.1.2 Kanban Board .....	19
2.2 Functional and Non-Functional Requirements .....	20
2.2.1 Functional Requirements .....	20
2.2.2 Non-Functional Requirements.....	21
3.0 Design Phase – System Design.....	22
3.1 Use Case & System Modeling .....	22
3.1.1 Use Case Description – Apply for job.....	22
3.1.2 Use Case Description – Post job vacancy .....	23
3.1.3 Use Case Description – CSC Verifies and Approve Employer Account.....	24
4.0 Development Phase – Execution & Prototype Development .....	26
4.1 Prototype .....	26

4.3 Develop an initial Test Plan .....	40
5.0 Deployment Phase – System Deployment Strategy & Final .....	42
5.1 Deployment Strategy.....	42
5.2 Final Sprint Review & Retrospective .....	44
6.0 User Story Creation & Processing .....	46
6.1 Chan Min Huey TP083261 .....	46
6.1.1 Student – Student Apply for Job.....	46
6.1.2 Student – Student Receive Appointment Notification .....	46
6.2 Cheong Pei Enn TP081926 .....	47
6.2.1 CSC – Verify Employer Account.....	47
6.2.2 CSC – Request Additional Employer Documents.....	47
6.3 Ng Xiang Ling TP082046 .....	48
6.3.1 Student – Manage Profile and CV .....	48
6.3.2 Student – View Application Status.....	48
6.4 Woo May Eng TP082001.....	49
6.4.1 Employer – Filter job application.....	49
6.4.2 Employer – Schedule Interviews for shortlisted candidates.....	49
6.5 Yap Li Shan TP080968 .....	50
6.5.1 Employer – Post a Job .....	50
6.5.1 Employer – Manage Posted Jobs.....	50
6.6 Kang Hong Qian TP081205 .....	51
6.6.1 CSC – Authorize Job Posting .....	51
6.6.2 CSC – Suspend or Remove Position Advertising .....	51
7.0 Agile Principles & Reflection .....	52
7.1 Agile Principles .....	52
7.1.1 Chan Min Huey TP083261 .....	52
7.1.2 Cheong Pei Enn TP081926.....	54

7.1.3 Ng Xiang Ling TP082046 .....	56
7.1.4 Woo May Eng TP082001 .....	58
7.1.5 Yap Li Shan TP080968 .....	60
7.1.6 Kang Hong Qian TP081205 .....	62
7.2 Information System Methodology Reflection.....	64
7.2.1 Chan Min Huey TP083261 – Waterfall vs Scrum.....	64
7.2.2 Cheong Pei Enn TP081926 – Hybird Model vs. Agile .....	66
7.2.3 Ng Xiang Ling TP082046 – Kanban vs Prototyping .....	68
7.2.4 Woo May Eng TP082001 – Scrum vs Rapid Application Development RAD) ....	70
7.2.5 Yap Li Shan TP080968 – V-model vs. RAD .....	72
7.2.6 Kang Hong Qian TP081205 – Extreme Programming (XP) vs Waterfall .....	74
8.0 System Design .....	76
8.1 Design Model .....	76
8.1.1 Chan Min Huey TP083261 – UML Sequence Diagram (Apply Job) .....	76
8.1.2 Cheong Pei Enn TP081926 – UML Class Diagram (Employer Account Verification).....	78
8.1.3 Ng Xiang Ling TP082046 – UML Activity Diagram (Manage Profile and CV)....	81
8.1.4 Woo May Eng TP082001 – Use Case Diagram (Scheduled Interview for Shortlisted Candidates).....	83
8.1.5 Yap Li Shan TP080968 – Use Case Diagram (Post a Job) .....	85
8.1.6 Kang Hong Qian TP081205 - Sequence Diagram (Approve Job) .....	87
9.0 Appendix.....	89
10.0 References.....	90

# **1.0 Planning Phase – Product Backlog & System Requirements**

## **1.1 Introduction**

The Asia Pacific University of Technology and Innovation (APU) is an educational institution specializing in technology-based education. In line with their commitment to improve student work readiness and industry collaboration, APU is building its TalentLink System. This centralized service, which cuts down drastically the manual and decentralized process currently experienced in the university's Career Services Centre (CSC), is built to enhance job-internship matchmaking for students.

### **1.1.1 Stakeholders**

<b>Stakeholder</b>	<b>Role and Responsibilities</b>
<b>Career Services Centre (CSC)</b>	Project sponsor and administrator. Ensure employer accounts are active, posts jobs, monitor system usage and solve problems.
<b>Students</b>	Primary end users. Sign up, upload resumes, have a look around, and apply for jobs.
<b>Employers (Internal &amp; External)</b>	Internal: Jobs on campus by department. External: Private industry that provides internship or off-campus jobs. Post open positions, screen applicants, and schedule interviews.
<b>IT Administrators</b>	Technical assistance and system maintenance. Updates, monitoring system security, and availability.

### **1.1.2 Users**

User Category	Roles and Usage
<b>Students</b>	Log in to register, post your resume and search for jobs; receive emails about newly posted part-time/internship positions; apply for the position using one click.
<b>Employers (Internal/External)</b>	Organizations can sign up, post job listings, browse candidates online to shortlist them, and schedule interviews between suitable candidates and recruiters.
<b>Career Services Centre (CSC)</b>	Confirm employers and advertise, supervise system use, support and ensure adherence to the recruitment process.
<b>IT Administrators</b>	System maintenance and updated Security monitoring Smooth running of the system with data protection Support.

### **1.1.3 Current Business Process**

Process Step	Current Practice
<b>Job Posting</b>	Companies send job information to CSCJob details posted on noticeboards or by email.
<b>Application Submission</b>	Students apply by emailing their resume or providing a paper copy.
<b>Screening &amp; Shortlisting</b>	Hiring managers and recruiters read resumes by hand, with no support from the digital world, which means filtering is not very effective.
<b>Interview Scheduling</b>	Employers and students schedule interviews by email or phone, causing hassles and misunderstandings.
<b>Tracking</b>	Students cannot know about their application status, which increases the administrative burden on CSC by manual tracking and recording.

## 1.2 Problem Statement

### Problem 1

<b>Title</b>	<b>Manual Job Posting</b>
<b>Description</b>	Employers these days submit job information to CSC in paper form or through emails. This delays since the staff should upload and authenticate all the information manually prior to publishing.
<b>Cause / Source</b>	1. No digital job posting platform. 2. Employers are dependent on traditional communication techniques.
<b>Effects</b>	Job openings are advertised late, decreasing promotion for students and causing undue workload for CSC staff.

### Problem 2

<b>Title</b>	<b>Non-Digital Applications</b>
<b>Description</b>	Students send in applications in the form of e-mailed resumes or in physical copies, which cause missing files and difficulties in tracking.
<b>Cause / Source</b>	1. Lack of centralized online application system. 2. Manual resume collection by CSC and employers.
<b>Effects</b>	Lost applications, slow employer responses, and reduced chances for students to secure interviews.

### Problem 3

<b>Title</b>	<b>Ineffective Screening</b>
<b>Description</b>	CSC and employers manually go through each resume to identify qualified candidates, which is inefficient and prone to human error.
<b>Cause / Source</b>	1. No automated filtering or search tools. 2. Manual review process.
<b>Effects</b>	Candidate selection takes longer, leading to delays in shortlisting and slower recruitment cycles.

#### **Problem 4**

<b>Title</b>	<b>Uncoordinated Interview Scheduling</b>
<b>Description</b>	Interviews are scheduled through email or phone calls, which often results in overlapping appointments or miscommunication.
<b>Cause / Source</b>	1. Absence of integrated scheduling system. 2. Reliance on manual coordination.
<b>Effects</b>	Students may miss interviews, and employers waste time rescheduling.

#### **Problem 5**

<b>Title</b>	<b>No Application Tracking</b>
<b>Description</b>	The students cannot trace their job application status and are required at times to contact the CSC manually for the status.
<b>Cause / Source</b>	1. Absence of tracking facility in the existing system. 2. Update in status is manually conducted through CSC staff.
<b>Effects</b>	Causes frustration for students and increases administrative workload.

#### **Problem 6**

<b>Title</b>	<b>Weak Data Management</b>
<b>Description</b>	Attended and student records are manually kept without due security and backup provision, which can lead to loss of information or access.
<b>Cause / Source</b>	1. No centralized database. 2. Poor data security practices.
<b>Effects</b>	Potential breaches in the data, inefficient processes, and lack of vital recruitment files.

### 1.3 Proposed System

Problem	Solution
<b>Manual Job Posting</b>	Employers can post job vacancies directly via the online platform. Each post is reviewed and approved by the Career Services Centre (CSC) before publication, reducing manual work and improving efficiency.
<b>Non-Digital Applications</b>	Students can submit job applications digitally via the TalentLink system. All applications are stored in a central database, making it easier for employers and CSC staff to access and manage records.
<b>Ineffective Screening</b>	Automated search and filtering tools can help employers and CSC staff to identify qualified candidates more quickly based on keywords, skills, or qualifications in student profiles and CVs.
<b>Uncoordinated Interview Scheduling</b>	The system provides an online interview scheduling feature through which employers can send interview invitations directly to students. Notifications are generated automatically to prevent overlapping appointments or communication errors.
<b>No Application Tracking</b>	Students can track the progress of their job applications in real time through the system. The student dashboard displays status updates such as 'Applied', 'Submitted' and 'Reject'.
<b>Weak Data Management</b>	A central database is used to securely store all data, including student profiles, employer records and job postings. This database is protected by proper access control and backup features, which prevent data loss and ensure reliability.

## Context Diagram

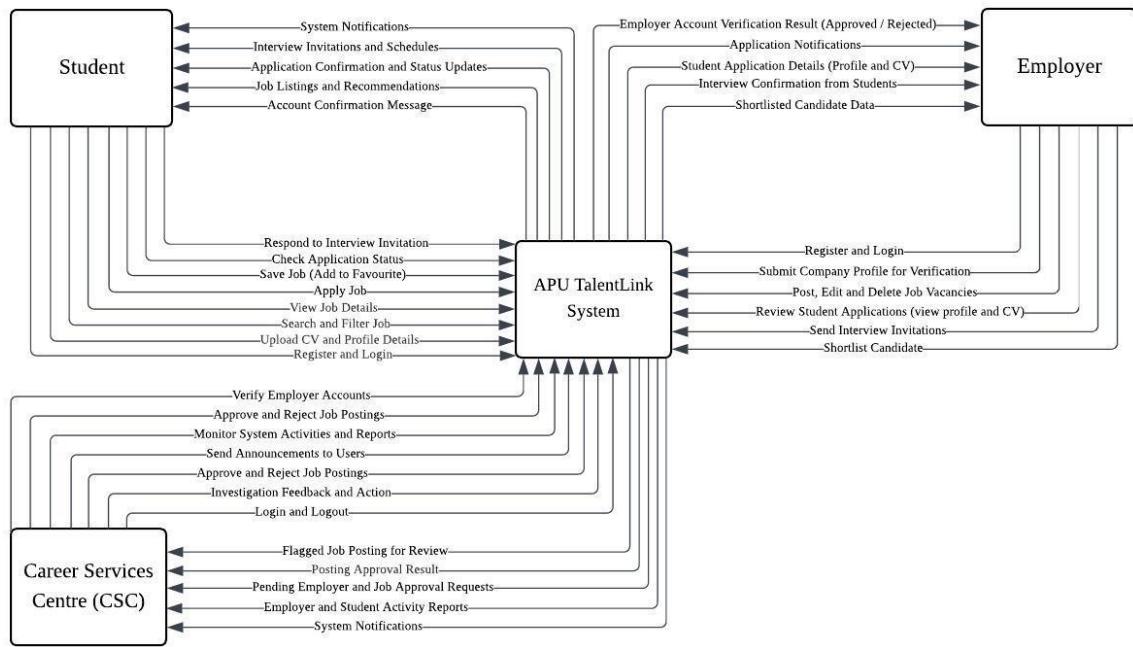


Figure 1.3 Context Diagram of APU TalentLink System

This context diagram shows how the APU TalentLink System interfaces and exchanges information with three main users: Students, Employers and Career Services Centre (CSC). The system enables students to register and update their profiles and CVs, apply for jobs, and receive notifications such as interview invitations and application updates. Employers can register company accounts, post and manage job vacancies and review student applications to shortlist suitable candidates for interviews. The CSC verifies employer accounts and approves or rejects job postings. It also reviews any listings that are flagged by the system to ensure accuracy and reliability. The diagram provides an overview of the main communication between users and the system.

## **1.4 Scopes and Objectives**

### **1.4.1 Project Scopes**

The APU TalentLink System is a centralized online platform that connects students, employers and the Career Services Centre (CSC). It replaces the manual job posting and application process with a digital and efficient system. The main scopes of the system are:

1. Students can register and create profiles and upload their CVs for employers to view.
2. Students can browse verified job postings, apply for positions and check the status of their applications.
3. Employers can create accounts, post job vacancies and update or remove their listings as required.
4. The CSC can review and approve employer registrations and job postings before making them visible to students.
5. Employers can use the system to review applications, shortlist suitable candidates and schedule interviews.
6. Students can receive notifications about application updates, shortlisting results and interview invitations.
7. CSC staff can monitor activity, review flagged job postings and generate reports for record purposes.
8. All information is stored securely in a centralized database with proper backup and access control.
9. The system is designed for internal use by the APU and does not integrate with external recruitment platforms such as LinkedIn or JobStreet because its focus is on managing job opportunities within the university environment.
10. The system does not support real-time communication features such as live chat or video conferencing, because its main purpose is to manage job postings and applications rather than hosting direct interactions.

## **1.4.2 Project Objectives**

The APU TalentLink System uses digital technology to improve the job placement process for students, employers and CSC. The objectives are:

1. To digitalize job posting, application and approval processes within APU.
2. To reduce manual work and human errors by introducing an automated system.
3. To give students easier access to verified job opportunities and real-time updates.
4. To help employers manage job listings and applications more efficiently.
5. To ensure data accuracy, security and reliability through central management.

## 1.5 Project Scheduling

The Gantt Chart above represents the general schedule of the project APU TalentLink System, based on the principles of an Agile approach in the form of timeboxed sprints. The total project duration is nine weeks where it starts on 14 July 2025 and completes on 10 September 2025. Its workflow is divided into six major steps: Planning, Analysis, Design, Implementation, Testing and Deployment where a single sprint of work contributes to the completion of the system.

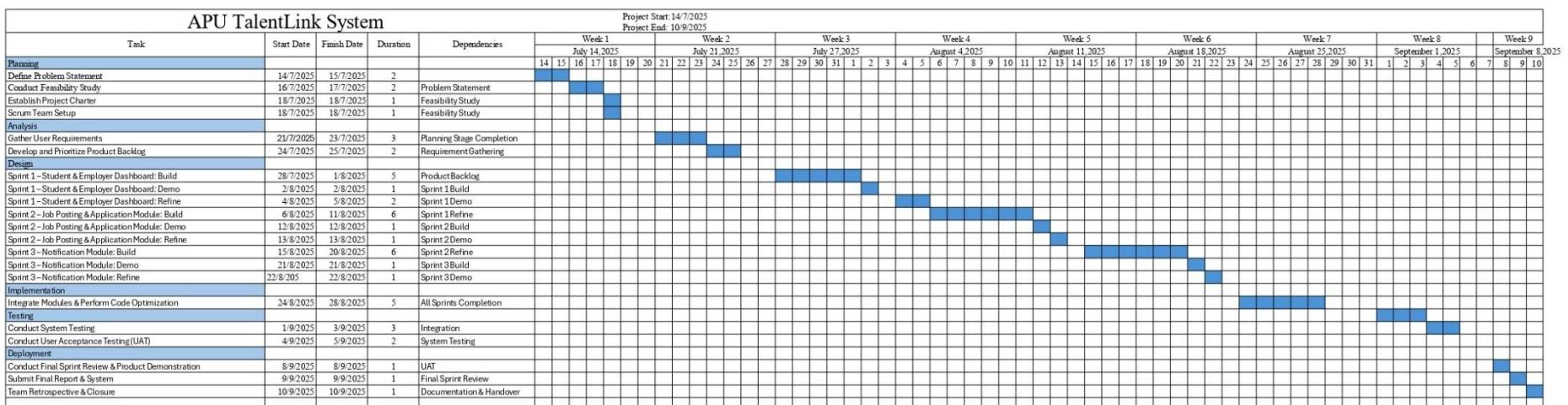


Figure 1.5 Project Scheduling Gantt Chart

## 2.0 Analysis Phase – Task Engineering & System Analysis

### 2.1 Product Backlog Creation

#### 2.1.1 Product Backlog

ID	User stories summary			Acceptance Criteria	Priority	Status
	As a ...	I want to ...	So that ...			
US01	Student	Apply for a job through the TalentLink system	I can conveniently submit my application to multiple employers without visiting different websites	1. By clicking the “Apply” button, I can submit my job application. 2. The applied job is stored in my application history. 3. I receive a confirmation notification after successful submission. 4. I should not be able to apply for the same job again.	Must	Done
US02	Student	Receive a notification when an employer schedules an interview	I can prepare and attend the interview on time	1. The system sends an automatic notification when an interview is scheduled. 2. The notification includes date, time, and meeting details. 3. The scheduled appointment is marked on my calendar. 4. I should be able to view all past and upcoming interviews.	Must	Done
US03	Student	Manage my personal profile and upload my CV through the TalentLink system	I can keep my information accurate and presentable to employers	1. On the “Profile” page, I can update my personal details such as contact info and education background. 2. The system allows me to upload my latest CV to ensure my records are complete. 3. After saving, the updated details and uploaded CV are displayed	Should	Work in Progress

				immediately on my profile.		
US04	Student	View my updated profile and uploaded CV within the TalentLink system	I can confirm that my latest information has been successfully saved	1. The system displays my updated information and CV after editing. 2. The details shown match the most recent changes I made. 3. My updated data remains accessible under the “Profile” page for verification.	Should	Work in Progress
US05	Employer	Post a new job vacancy including title, job description, salary range, and work type on APU TalentLink	I can attract suitable and outstanding applicants in less time	1. Employers must complete all mandatory fields before posting. 2. Incomplete forms display an error and block submission. 3. Successfully posted jobs appear in the student dashboard’s “Recent Job List.” 4. System shows a confirmation message upon successful posting. 5. Employers can edit or delete published jobs via the “Job Listing” page.	Must	Done
US06	Employer	View, shortlist, and manage job applications for my posted vacancies	I can efficiently review profiles, schedule interviews, and make hiring decisions within the system	1. Employers can view applicant profiles and download CVs. 2. Filtering and sorting features are available for specialization and qualifications. 3. Buttons such as “Shortlist,” “Reject,” and “Schedule Interview” are functional. 4. System provides confirmation messages after each operation.	Should	Work in Progress
US07	Employer	View and filter job applications on the platform	I can easily identify qualified candidates for shortlisting	1. System shows all valid applications for my job posting(s). 2. I can filter by qualification, GPA, skills, and submission date. 3. Results refresh dynamically without full-page reload. 4. I can mark one or more candidates as Shortlisted. 5. Shortlisted candidates are saved and visible under the Shortlist tab. 6. System confirms the action and logs it for records.	Should	Work in Progress

<b>US08</b>	Employer	Schedule interviews for shortlisted candidates	I can progress hiring efficiently within TalentLink	<ul style="list-style-type: none"> <li>1. I can select shortlisted candidates to open the interview form.</li> <li>2. System checks for time conflicts before confirming date/time.</li> <li>3. Interview details (time, mode, link/location) are saved to the database.</li> <li>4. Student receives an in-system notification/confirmation.</li> <li>5. I can reschedule or cancel if needed; records are updated.</li> <li>6. Application status auto-updates to Interview Scheduled.</li> </ul>	Should	Work in Progress
<b>US09</b>	CSC Officer	Verify new employer accounts and authenticate company records	Only legitimate organizations can post jobs on TalentLink	<ul style="list-style-type: none"> <li>1. System alerts CSC when a new employer registers for verification.</li> <li>2. CSC can review company details and upload documents in the admin console.</li> <li>3. CSC can Approve or Reject the account after validation.</li> <li>4. Approved employers gain access to the Job Posting module; rejected employers receive reasoned notification.</li> <li>5. All verification actions are logged for audit.</li> </ul>	Must	Done
<b>US10</b>	CSC Officer	Request additional documents from employers when submissions are incomplete or unclear	Employer accounts are accurately verified before activation	<ul style="list-style-type: none"> <li>1. CSC can set status to Additional Documents Pending with comments on what is missing/corrective actions.</li> <li>2. Employers are auto notified to re-submit required files.</li> <li>3. After uploading and reviewing, CSC can Approve the account and update status accordingly.</li> <li>4. All requests, uploads, and status changes are logged for traceability.</li> </ul>	Could	To be started
<b>US11</b>	CSC Officer	Screen and approve job listings provided by employers	Only valid and appropriate vacancies appear in the TalentLink system	<ul style="list-style-type: none"> <li>1. CSC can review job listings after verifying employer validity.</li> <li>2. Approved jobs automatically appear in student listings.</li> <li>3. Employer receives confirmation upon approval.</li> </ul>	Must	Done
<b>US12</b>	CSC	Suspend or remove job postings that violate	The platform remains	<ul style="list-style-type: none"> <li>1. CSC can suspend or remove non-compliant postings.</li> <li>2. Suspended posts are hidden from student view.</li> </ul>	Could	To be started

	Officer	APU policies or contain false information	professional and credible	3. Employers receive automatic notifications with the reason.		
--	---------	---	---------------------------	---	--	--

## 2.1.2 Kanban Board

DO	DOING	DONE
<b>US10</b> – CSC: Request additional employer docs	<b>US06</b> – Employer: Manage/review applications	<b>US09</b> – CSC: Verify employer account
<b>US12</b> – CSC: Suspend / remove posting	<b>US07</b> – Employer: Filter applications	<b>US11</b> – CSC: Authorize job posting
—	<b>US08</b> – Employer: Schedule interviews	<b>US05</b> – Employer: Post job
—	<b>US03</b> – Student: Manage profile & CV	<b>US01</b> – Student: Apply for job
—	<b>US04</b> – Student: View updated profile & CV	<b>US02</b> – Student: Receive interview notification

## **2.2 Functional and Non-Functional Requirements**

### **2.2.1 Functional Requirements**

#### **Registration and Authentication of the user**

- Before being granted access, new employer accounts should be verified by CSC administrators.
- The system should authenticate the input of the user (email address, password strength) and create the account.

#### **Profile & Resume Management**

- Students would have to create and maintain personal profiles, such as academic background, skills, and posted resumes.
- The officers in charge of CSC should be able to view and grant approval to the student and employer profiles.

#### **Job Posting & Application**

- Students should be in a position to search and apply to the available jobs or internships through a keyword or filters (e.g. category, duration, company).
- The CSC officers will have to read and give their consent to the job postings before they are posted in line with the APU policies.

#### **Notifications and communication**

- Student and employer have to be notified about the updates on their applications or invitation to the interview or feedback.
- The users should have the capability of looking at all previous notifications in their message history section.

#### **Tracking & Reporting**

- The students need to see and monitor the status of each of their job applications (applied, shortlisted, interviewed, hired, rejected).
- The employer should have the capability of keeping track of how many applications are received by the advertisement.

## **2.2.2 Non-Functional Requirements**

### **Performance & Reliability**

- The maximum time required to load a page should be not more than three seconds given normal conditions of the server.
- It should not go offline except when the CSC administrators are on scheduled maintenance.
- The results of job search and updating application have to be accurate in less than five seconds.

### **Security & Privacy**

- Encryption of all user information including resumes and login credentials should be done both in transit and rest.
- The data should be secured by automatically ending user sessions after an inactivity period.
- CSC administrators can only give approval or cancel job listing and accounts.

### **Usability & Accessibility**

- The interface should be mobile, tablet, and desktop browsers friendly.
- Every action taken by a user should have clear feedback messages (success, warning, or error).

### **Maintainability & Scalability**

- System architecture should enable the easy addition of new modules, event registration or analytics.
- It should be automated to do a regular data backup every 24 hours.

## 3.0 Design Phase – System Design

### 3.1 Use Case & System Modeling



Figure 3.1 Use Case Diagram

#### 3.1.1 Use Case Description – Apply for job

Use Case	Apply for job
Brief Description	This use case allows student registered and logged in to submit job application. The system validates student's profile and CV, records the application, and sends a notification to the employer.
Actors	Student
Precondition	<ul style="list-style-type: none"> <li>• Student is registered and logged in.</li> <li>• Job posting is active and open for applications.</li> </ul>
Main Flow	<ol style="list-style-type: none"> <li>1. The use case begins when the student selects an available</li> </ol>

	<p>job vacancy to apply for.</p> <ol style="list-style-type: none"> <li>2. The system displays the job vacancy details.</li> <li>3. The student clicks “Apply”.</li> <li>4. The system verifies that the student had completed profile and uploaded CV.</li> <li>5. The system checks that the job vacancy is still available for applications.</li> <li>6. The system prompts the student to confirm the application submission.</li> <li>7. The student confirms the application.</li> <li>8. The system records the application in the database.</li> <li>9. The systems sends a notification to the employer about the new application.</li> <li>10. The system displays a success message and updates the student’s application history.</li> </ol>
Alternative Flows	<p>4(a) If the CV file unsupported, the system rejects it and requests a valid file.</p> <p>5(a) If the job vacancy is closed or expired, the system displays an error message indicating the position is unavailable.</p> <p>8(a) If system error during submission, the system displays “submission pending” and prompts the student to retry later; system logs the failed transaction.</p>

### 3.1.2 Use Case Description – Post job vacancy

Use Case	Post job vacancy
Brief Description	This use case allows a verified employer to create and submit a new job vacancy in the system. The system validates the job details and then forwards the vacancy for Career Service Centre (CSC) approval before it becomes visible to students.
Actors	Employer
Precondition	<ul style="list-style-type: none"> <li>• Employer has registered and logged in account.</li> </ul>

	<ul style="list-style-type: none"> <li>• Employer account is verified by the Career Service Centre.</li> <li>• Employer has access rights to create job vacancies.</li> </ul>
Main Flow	<ol style="list-style-type: none"> <li>1. The use case begins when the employer selects “Post Job Vacancy” from the system dashboard.</li> <li>2. The system displays a job posting form.</li> <li>3. The employer enters required job details</li> <li>4. The employer submits the job vacancy for approval.</li> <li>5. The system validates all required fields.</li> <li>6. The system records the job vacancy in the database with status “Pending Approval”.</li> <li>7. The system notifies the Career Service Centre of the new vacancy.</li> <li>8. The system displays a confirmation message to the employer.</li> </ol>
Alternative Flows	<p>1 (a) If the employer account not verified, the displays a message that verification is required before posting jobs.</p> <p>5 (a) If the system detects missing or invalid job details, it highlights the incorrect fields and prompts the employer to correct them.</p> <p>6 (a) If the CSC rejects, employer receives rejection reasons and able to edit and resubmit.</p>

### 3.1.3 Use Case Description – CSC Verifies and Approve Employer Account

Use Case	CSC Verifies and Approve Employer Account
Brief Description	This use case allows the Career Service Centre (CSC) to review newly registered employer accounts, verify their legitimacy, and either approve or reject the account. Only approved employers gain access to job posting features.
Actors	Career Service Centre (CSC staff)
Precondition	<ul style="list-style-type: none"> <li>• Employer had verified and submitted job posting.</li> </ul>

	<ul style="list-style-type: none"> <li>• CSC user is logged into the CSC management interface.</li> </ul>
Main Flow	<ol style="list-style-type: none"> <li>1. The use case begins when the CSC staff accesses the list of pending employer accounts.</li> <li>2. The system displays all employer accounts awaiting verification.</li> <li>3. The CSC staff selects an employer account to review.</li> <li>4. The system displays company details and supporting documents.</li> <li>5. The CSC staff verifies the accuracy and legitimacy of the information.</li> <li>6. The CSC staff approves the employer account.</li> <li>7. The system updates the employer status to “Active.”</li> <li>8. The system notifies the employer of successful approval.</li> </ol>
Alternative Flows	<p>5 (a) If the employer information is incomplete or suspicious rejected by the CSC staff, the system notifies the employer with reasons and required corrections.</p> <p>5 (b) If the CSC staff denies access due to policy violation or fraud. The CSC staff marks the account as “Blacklisted.” The system permanently blocks the account. The system notifies administrators for further action.</p>

## 4.0 Development Phase – Execution & Prototype

### Development

#### 4.1 Prototype

##### 1. User-centred Design

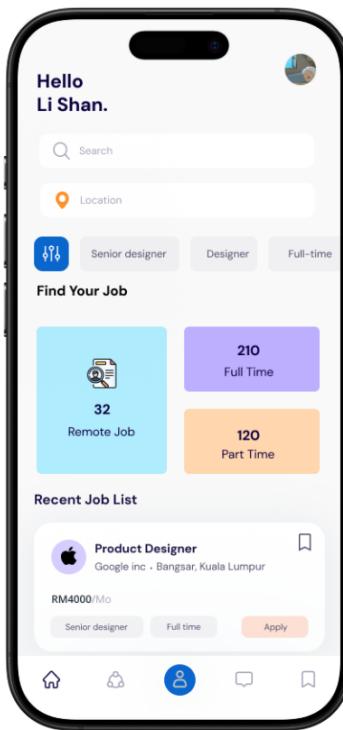


Figure 4.1.1 Student - Dashboard

The student dashboard is the first interface that students will see after logging into the application. Upper area contains actionable interface components, such as a search bar and various filtering functions, allowing users to search for career opportunities by keywords and/or locations. The "Job Search" area on the screen contains three types of job, which is "Remote Positions", "Full-time" and "Part-time", and displays the current number of positions in each field. The application is also with a "Recent Job List" function. This curated list will automatically push the most matching job information based on the user's browsing history. Lastly, the navigation bar at the bottom of the screen features five core functions which is "Home", "Application Record", "Personal Profile", "Notifications" and "Favorites". Users can quickly access frequently used function with just one click.

## Design Principle - User-Centered Design

This interface is made for the students' central goal, allowing them to efficiently find and apply for a suitable job. All elements in the interface, such as the search bar, category cards, and recommended positions list are all focused on the user's task to ensure that all operations are simple to understand and easy to accomplish. Additionally, with the specific content area to indicate user preference and the structure of hierarchy of content, the system allows students to easily explore and select suitable positions, thereby lessens the cognitive load of the system. The bottom navigation bar also provides access to the frequency function needed for the user in one click, improving usability and interaction experience.

## 2. KISS (Keep It Simple and Straightforward)

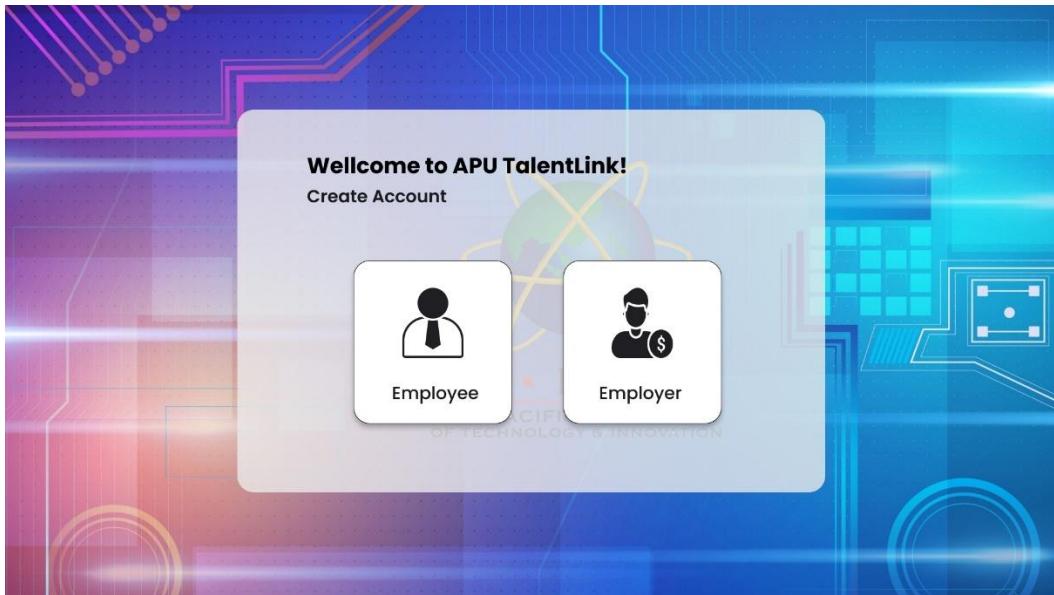


Figure 4.1.2 Registration Page

The system will take the user to this selection page after they click "Create Account." Users can register by simply choosing between the two options: "Employee" or "Employer."

### Design Principle – KISS (Keep It Simple and Straightforward)

This page follows the "Less is more" design philosophy and the KISS design principle. The system prevents users from being bothered by complicated forms or redundant information when they first register by keeping only two basic options. In addition to improving registration efficiency, the simplified layout successfully lowers the likelihood of user operational errors.

### 3. Consistency

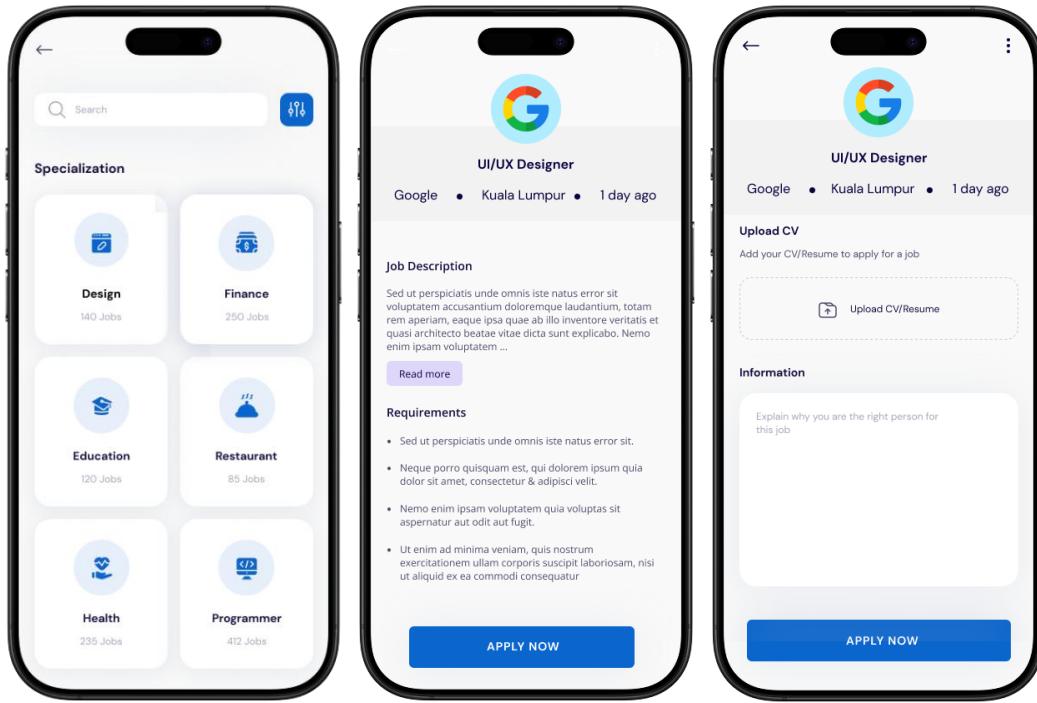


Figure 4.1.3 Student - Search, Job Details, Application Page

This section presents the complete user journey from searching for a job to applying. When the user clicks the "Search" button on the dashboard page, the system will display a category menu for students to filter positions by areas of interest. After clicking the "Apply Now" button for the selected position, the system will redirect to the position details page, which displays company information, job description and requirements. After clicking "Apply Now" again on the details page, the user will enter the application page. Then students can fill in supplementary information, upload resumes and submit applications here.

#### Design Principle – Consistency

Multiple functional pages maintain a consistent visual design, including color, font, icon and button layout, demonstrating a strong visual unity. This visual consistency reduces the burden on users to relearn interface navigation, as users can rely on this sense of familiarity at different operation steps. Whether it is searching, viewing details or submitting applications, users can quickly identify the button area, thereby enhance usability and improve efficiency.

#### 4. Visibility of System Status

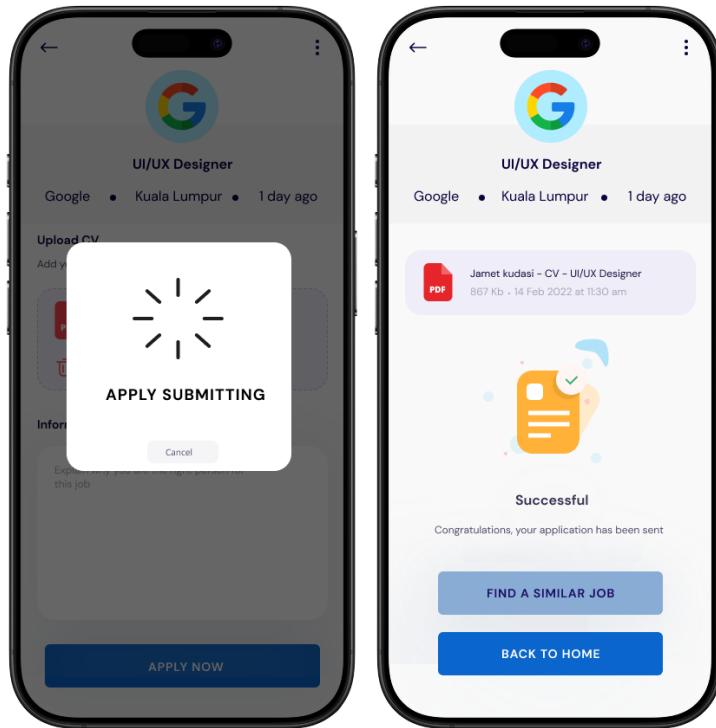


Figure 4.1.4 Student - Application Page

When the user clicks the "Apply Now" button on the application page, the system will display a loading animation indicating that the application is being processed. After loading is complete, the application page will switch to display the status of successful application. The "Return Home" button and the "Find Similar Positions" button located below the successful application notification allow users to return to the home page or continue searching for other positions similar to the one they just applied for.

#### Design Principle – Visibility of System Status

This principle emphasizes that the system must inform users as soon as possible whether its operating status is normal. The system clearly conveys the information that requests are being processed and have been successfully submitted through animations and notifications, effectively alleviating the anxiety caused by waiting and uncertainty. After the task is completed, the design of the status box helps maintain the user's exploration process, as clear and immediate feedback not only boosts user confidence but also makes the entire interaction process more coherent.

## 5. Visual Hierarchy

The screenshot displays the 'Pending Verification (8)' section of the APUTalentLink Admin interface. On the left, a sidebar menu includes 'Dashboard', 'Verify Employer Account' (which is selected), 'Job Monitoring', 'User Management', and 'System Settings'. The main area shows five rows of pending accounts:

Account Status	Action	Action	Action
Super Kitchen Sdn Bhd John • john@superkitchen.mail.com Registered: 2025-09-20 • Industry: F&B	Review	Approve	Reject
Jo.Jo Kitchen Sdn Bhd Amy • amy@jojo.mail.com Registered: 2025-09-20 • Industry: Marketing	Review	Approve	Reject
BCC Sdn Bhd Kenny • kenny@bbc.mail.com Registered: 2025-09-20 • Industry: Business	Review	Approve	Reject
Mario Kart Sdn Bhd Hills • hills@mariokart.mail.com Registered: 2025-09-20 • Industry: Tech	Review	Approve	Reject
McD Sdn Bhd Nugget • nugget@mcd.mail.com Registered: 2025-09-20 • Industry: Design	Review	Approve	Reject

At the bottom, there are navigation arrows for page 01 and 02.

Figure 4.1.5 Admin - Verify Employer Account Page

This page is the Employer Account Verification page for administrators. This page is used to review pending company accounts that awaiting verification. The system displays pending verification account details in a table format. The left-hand side of the table presents company logo and fundamental company information. Besides, three action buttons are positioned on the right-hand side of the table, they are review, approve, and reject.

### Design Principle – Visual Hierarchy

By varying the font weight, color intensity, and layout placement, this interface creates a distinct visual hierarchy. Users can quickly identify important elements because crucial information, like the company name, is displayed in bold font on the top line. Lighter palettes are used for supplemental information such as the application date and username, which successfully separate primary from secondary content. To draw the administrator's attention and direct further actions, the action buttons on the right use different colors based on their purpose. The user experience and management efficiency are greatly improved by this well-organized visual hierarchy, which makes page information instantly understandable.

## 6. Timely Feedback and Responsiveness

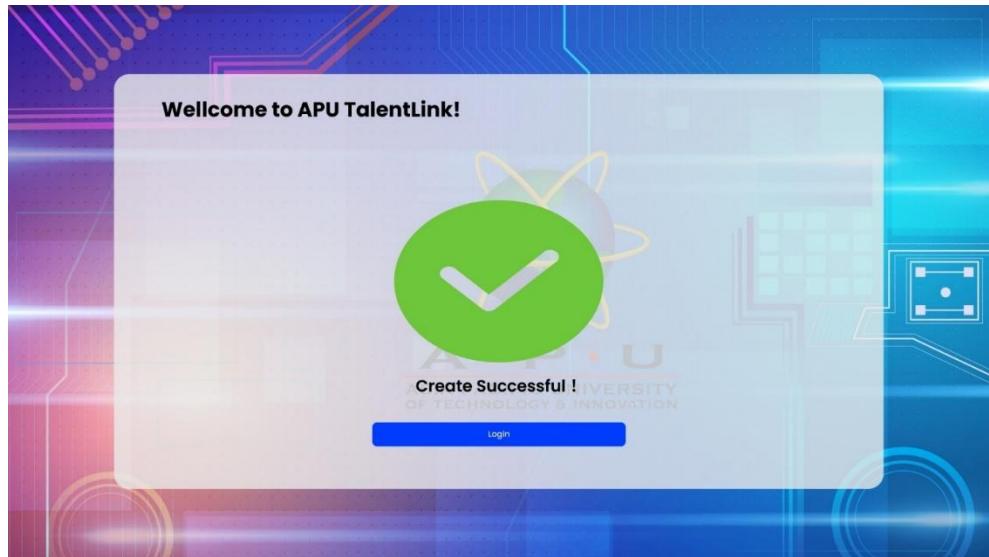


Figure 4.1.6 Registration Page

The system will notify the user with green checkmark and the icon with the words “Create Successful” after the registration form has been successful submitted. A login button at the bottom of the page directs the user backs to the login page so they can start to use the system.

### Design Principle – Timely Feedback and Responsiveness

This page uses timely and clear visual cues to inform users of operation outcomes, adhering to the design principles of feedback and responsiveness. In addition to improving the psychological cue of "task completion," the combination of tick icons and successful messages allows users to instantly verify the system's response. Concurrently, the interface offers straightforward buttons for subsequent actions, which quickly direct users to the next phase and avoid process problems or user confusion. This immediate feedback improves the system's dependability and builds user trust in addition to increasing the flexibility of the interaction experience.

## 7. Accessibility

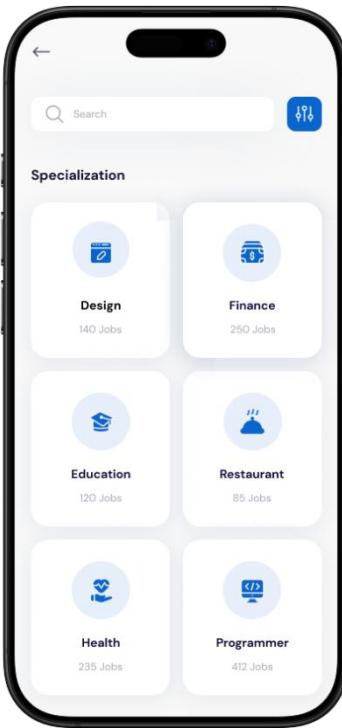


Figure 4.1.7 Student – Search Page

When students click the search bar on the student dashboard, the user could continue entering the key word to search or use the category cards below for quick access to different job types. Each type of category-like design and finance is presented as a card displaying the category name, number of available positions and corresponding icons. This enables users to visually identify content across categories.

### Design Principle – Accessibility

This page follows the accessibility principle to ensure the system can be easily used by the user with different backgrounds and abilities. By using text and icons on the category card, the system will provide an additional perceptual clue for users that with visual limits or language comprehension. To guarantee readability across all devices and under different lighting conditions, the spacing between cards with clear typography and high-contrast design are used. This design enhances information accessibility and interaction convenience for users ,and further demonstrates the system commitment to inclusivity and diversity.

## 8. Error Prevention

The screenshot shows the 'Post a job' form on the APUTalentLink platform. The left sidebar lists 'EMPLOYERS DASHBOARD' with 'Dashboard', 'Job Listing' (selected), 'Application Management', 'Recent Activity', and 'Settings'. The main area has a header 'Post a job'. It includes fields for 'Job Title' (with an error message 'Please enter a Job Title!'), 'Tags' (AI, Computer Sains), 'Job Role' (Engineer), 'Salary' (Min: 3500 MYR, Max: 6000 MYR, Type: monthly), 'Advance Information' (Education: Degree, Experience: -, Job Type: -), 'Vacancies' (1), 'Expiration Date' (06/10/2025), and 'Job Level' (-). Below these are sections for 'Apply Job on:' ('On APUTalentLink' checked, 'On Your Email' uncheckable), 'Description & Responsibility' (description: 'We are looking for an AI Engineer to design, build, and deploy AI-driven solutions. The role involves developing machine learning models, optimizing algorithms, and integrating AI capabilities into existing systems. You will work closely with data scientists and software engineers to deliver scalable applications that solve real-world business challenges.'), and 'Responsibilities' (list: 'Develop and train machine learning and deep learning models.', 'Preprocess and analyze large datasets to extract insights.', 'Deploy AI models into production and monitor performance.', 'Collaborate with cross-functional teams to integrate AI solutions into products.', 'Stay up to date with the latest advancements in AI/ML.'). At the bottom are 'Post Job' and 'Cancel' buttons.

Figure 4.1.8 Employer – Post a Job Page

When the employer wants to post a job, the system will display a form interface for them to fill in. System will require the employer to fill up all the key job details, including the job title, tags, salary requirements and others. As shown in the figure 4.1.8, if the employer clicks the “Post Job” button without filling the “Job Title”, the system will display an error message. This message will highlight the missing content and prompt the employer to complete it. Furthermore, a warning message will appear below the form to let user identify and correct the error quickly.

### Design Principle – Error Prevention

This page is designed with error prevention principles to reduce the user from submitting incorrect information through input validation and real-time feedback. Before the form is

submitted, the system will automatically check the conditions and completions of all required fields and provides instant feedback when error is detected. This design not only helps users avoid repeated actions but also ensures the accuracy and consistency of the information. Other than that, clear error messages and visual indicators could let user identify issues quickly, further enhance the user experience and system reliability.

## 9. Affordance

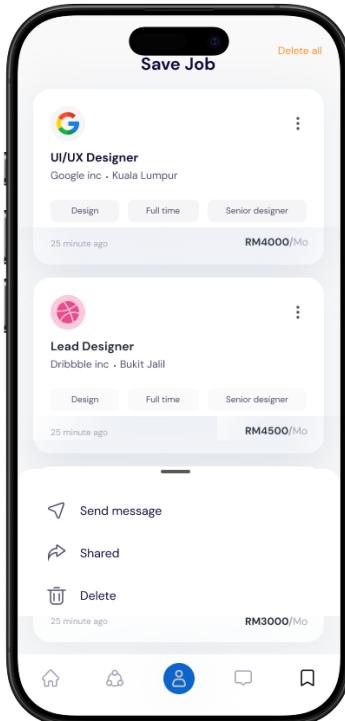


Figure 4.1.9 Student – Saved Jobs Page

This page shows all the job listings saved by the student. Each saved job card has an icon at the upper right corner. Click on the icon, then it will pop up a function menu with three options: "Send Message", "Share" and "Delete".

### Design Principle – Affordance

This design follows the principle of affordance. By using visual hints, users are able to identify the interactive elements and the functionality. The three-dot icon on the upper-right corner of the job card clearly shows that this area is clickable to expand more options. User could visually distinguish which area is draggable or clickable, further enable them to complete their task without additional instructions. This clear interactive feedback reduces the learning curve of the system and enhances the system's smoothness and visualization. Overall, the system operation is more aligned with user habits and delivered a more natural and seamless experience.

## 10. Responsive Design

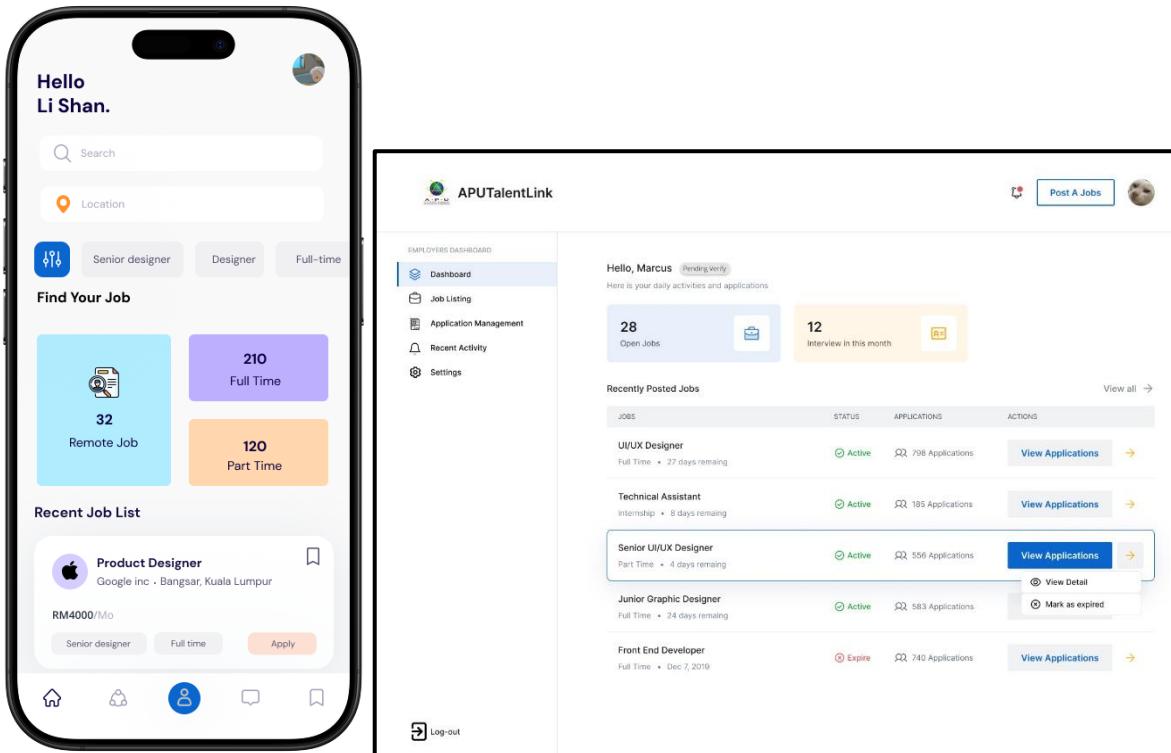


Figure 4.1.10 Student & Employer Dashboard

This section presents two different layouts of the system on different device, the desktop dashboard is mainly for employers while the mobile dashboard is mainly for students.

### Design Principle – Responsive Design

The responsive design principle could ensure the seamless operation of the system across different devices and resolutions through the dynamic element adjustments and adaptive layouts. To provide consistent interaction experience on various devices, the application interface will automatically fit the components based on the screen size. Follow by the responsive design principles, the desktop interface will focus more on comprehensive information presentation while the mobile interface focuses on convenience and operability. This feature ensures users can access the system anytime and anywhere. Overall, this design enhances the accessibility and flexibility of the system while the system also aligns with modern user's cross-platform usage habits. These features will further increase the value of the system.

## 4.2 Testing Strategy

Test Strategy	Unit Testing
Description	<p>Unit testing involves independently testing the smallest parts of your code within a system such as functions, modules, or classes to verify that their functionality operates as intended. During testing, testers need to write multiple tests for a single function to cover various possible scenarios and these all-name test cases. The logical correctness of each module is ensured by testing it independently of the rest of the system, which means it is not dependent on databases or networks.</p>
Advantages	<ol style="list-style-type: none"> <li>1. <u>Can detect errors at an early stage</u> Unit testing enables developers to identify and recovery issues during the early stages of development, thereby preventing the issue becoming more challenging and difficult to resolve in future, while simultaneously also can reduce the recovery costs.</li> <li>2. <u>Enhancing code quality</u> By conducting independent testing for each part, developers can ensure that every unit of code functions as intended and meets requirements, thereby significantly improving code readability and maintainability.</li> <li>3. <u>Enhance Developer Confidence</u> Once system code is all done with unit testing, developers will gain greater confidence when modifying or extending functionality. This is because unit testing helps to confirm that existing features operate as intended and will not be inadvertently disrupted in subsequent stages.</li> </ol>
Example	<p>Within the APU TalentLink system, unit tests may be employed to validate whether the login functionality correctly processes user input such as usernames and passwords. When incorrect credentials are entered, the system should return a login failure prompt. Conversely, users should successfully login when correct input.</p>

Test Strategy	Integration Testing
Description	<p>Integration testing is a testing technique employed to verify the correctness of interactions and data exchange between different components or modules within a system. Once all modules have passed unit testing and been assembled, integration testing is conducted to examine the data flow between them, interface compatibility, and overall operational logic.</p>
Advantages	<ol style="list-style-type: none"> <li>1. <u>Ensuring the correctness of interactions between modules</u> Integration testing verifies that data transmission and communication logic between different modules function correctly when collaborating, thereby guaranteeing the overall stability of the system's operation.</li> <li>2. <u>Can detect interface compatibility issues</u> By executing integration tests between modules, problems like potential interface mismatches, data type conflicts, or dependency problems can be identified, thereby preventing system failures during future deployment phases.</li> <li>3. <u>Simulating real-world system operating environments</u> Integration testing can replicate actual user workflows, helping developers identify potential defects that may come out in real-world operational environments. This enables defects can be rectification early, thereby enhancing the system's overall stability.</li> </ol>
Example	<p>Within the APU TalentLink system, integration tests can be employed to validate the complete job application process from students. Upon a student clicking ‘Apply Now’, the system should correctly initiate interactions between the application module, database module, and notification module. Furthermore, testing must ensure application details are successfully recorded into database, and that the notification module automatically send confirmation notifications to the student's account after a successful application.</p>

## 4.3 Develop an initial Test Plan

TEST DATE/ TIME: To be determined

PROJECT ID: APU TalentLink Prototype

TEST BY: Development Team

SYSTEM TITLE: APU TalentLink System

TEST CASE	TEST TYPE	TEST STEPS	EXPECTED RESULT	ACTUAL RESULT	REMARKS
1. User Sign-in	Unit Test	<ol style="list-style-type: none"> <li>1. Navigate to login page</li> <li>2. Enter valid username and password</li> <li>3. Click “login” button.</li> </ol>	Users are successfully authenticated and redirected to their respective dashboard like Student or Employer.	-	-
2. Students Apply for Job	Integration Test	<ol style="list-style-type: none"> <li>1. Student logs in.</li> <li>2. Students search and select a job.</li> <li>3. Student clicks “Apply Now”</li> <li>4. Student confirms application.</li> </ol>	<ol style="list-style-type: none"> <li>1. Jobs details are displayed.</li> <li>2. Application is recorded.</li> <li>3. Confirmation message is shown to the student account.</li> <li>4. Employer receives a notification of the new application.</li> </ol>	-	-
5. Employer Post a Job	System Test	<ol style="list-style-type: none"> <li>1. Employer logs in.</li> <li>2. Employer clicks “Post a</li> </ol>	<ol style="list-style-type: none"> <li>1. System validates all required fields.</li> <li>2. Job vacancies is</li> </ol>	-	-

		<p>Job.”</p> <p>3. Employer fills in all mandatory fields</p> <p>4. Employer submits the form.</p>	<p>saved with status “Pending Approval.”</p> <p>3. A confirmation message is displayed.</p> <p>4. CSC receives a notification for a new job for approval.</p>		
--	--	--	---	--	--

## 5.0 Deployment Phase – System Deployment Strategy & Final

### 5.1 Deployment Strategy

Deployment Strategy	Parallel	Phased	Pilot
Project Size	Suitable for small-sized systems	Suitable for medium-sized systems	Suitable for large-sized or high-risk systems
Advantages	<ul style="list-style-type: none"> <li>• Old and new systems run at the same time</li> <li>• Easy to check and compare performance</li> <li>• Lower risk of data loss</li> </ul>	<ul style="list-style-type: none"> <li>• Modules are released step by step</li> <li>• Easier to fix errors in smaller parts</li> <li>• Users can adjust gradually</li> </ul>	<ul style="list-style-type: none"> <li>• System tested in a limited environment</li> <li>• Useful for feedback and improvements before full rollout</li> </ul>
Limitations	<ul style="list-style-type: none"> <li>• Need to run two systems together, more workload</li> <li>• Higher maintenance effort</li> </ul>	<ul style="list-style-type: none"> <li>• Take longer overall</li> <li>• Some users may have to wait for access</li> </ul>	<ul style="list-style-type: none"> <li>• Limited users may not show all system issues</li> <li>• Take time to reach full operation</li> </ul>
Cost	High, because both systems must be maintained at the same time	Moderate, as modules are released one by one	Moderate to high, depending on test duration
Time Consumed	Shorter overall because both systems operate together	Longer, since each phase must be completed before moving to the next	Moderate, depending on the pilot testing duration

The Phased Deployment approach was chosen for the APU TalentLink System based on the comparison. This method is considered suitable as the system comprises just three connected parts: Student, Employer and CSC. It helps to keep the system stable and gives users time to familiarize themselves with it. It also enables the development team to identify and resolve any issues early on, before moving to the next stage.

## **Deployment Steps**

- Phase 1: CSC and Employer Modules

The first phase will focus on the CSC and Employer modules. During this stage, CSC staff will verify employer accounts and approve job postings, enabling real job data to be added to the system.

- Phase 2: Student Module

The student module will then be launched. Students will then be able to register, explore available jobs and submit applications via the system.

- Phase 3: Notification and Reporting Modules

Once the main modules are operational, the notification and reporting features will be activated for all users. The system will start sending automatic updates and generating reports to support communication and tracking.

- Phase 4: Full System Integration

This is the final stage of the process, where all the modules are combined to form one complete platform. A final round of testing will be conducted to ensure the system runs smoothly before its official release.

## **5.2 Final Sprint Review & Retrospective**

### **1. Early Planning is important**

The first lesson learned is a proper startup planning could minimize rework and maximize overall project efficiency. With a problem statement clearly identified, scope and objective during the project initiation phases, it may effectively direct entire development process. A Gantt chart is developed to ensure task progress in an order manner and avoid the conflict of schedules or work overload during some stages.

### **3. Agile Methodology Strengthens Team Collaboration**

The second lesson learnt is agile methodology could help our team to improve task clarity and communication. Our team could handle workflow more effectively and be more transparent in task progress by using the Scrum or Kanban approaches. Daily sprint meetings and inspection of the backlog can put the team on the same page and also allow quick responsiveness to any changes. Overall, it increases productivity and efficiency of our teams.

### **3. User Stories Keep the Focus on Real Needs**

The third lesson learnt is writing user needs could help our team to know the requirements of user. By using the format “As a [user], I want to [action], so that [goal]”, our team could gain clear insight into user expectations and prioritizes the critical features. This method helps to enhance the relevance of system functionality and ensure the system develop closely with user needs.

### **4. Prototyping Improves Design Clarity**

The fourth lesson learnt is that building a prototype can help stakeholders and team members to visualize the system clearly before actual development begins. By using tools such as Figma, the misunderstood of stakeholder minimized and the expectations are aligned, making the design process became more structured. This early-stage visualization help to identify issue and make adjustments.

## 5. Testing and Deployment Strategies Are Key to Success

The last lesson learnt is a well-planned testing and deployment is essential for a system to launch successfully. By selecting an appropriate deployment method like phased or parallel deployment and establish a comprehensive test plan, it could help to enhance system reliability and stability. Issue identified in early testing phase not only build user trust but also reduce the risk of failure occur during post-deployment.

## **6.0 User Story Creation & Processing**

### **6.1 Chan Min Huey TP083261**

#### **6.1.1 Student – Student Apply for Job**

As a student, I want to apply for a job through the TalentLink system so that I can conveniently submit my application to multiple employers without visiting different websites.

##### **Acceptance Criteria:**

By clicking “Apply” button, I should be able to apply my ideal job. Once the job application has been submitted, the job applied should store in my application history. After the job has successfully applied, I should receive a confirmation notification from system. Lastly, I should not be able to apply for the same job again.

#### **6.1.2 Student – Student Receive Appointment Notification**

As a student, I want to receive a notification when an employer schedules an interview so that I can prepare and attend the interview on time.

##### **Acceptance Criteria:**

When the employer schedules an interview, I should be able to receive an automatically generated notification from the system. The notification should include the date, time, location or online meeting links of the interview. Furthermore, the appointment scheduled should be marked on my calendar. Lastly, I should be able to view all past and upcoming interviews.

## **6.2 Cheong Pei Enn TP081926**

### **6.2.1 CSC – Verify Employer Account**

As a CSC officer, I would like to confirm new accounts of employers and authenticate their company records so that only those that are legitimate organizations post their job opportunities in the TalentLink system.

#### **Acceptance Criteria:**

The system is expected to alert CSC when a new employer registration is being created to undergo verification. The officers at CSC should be able to check the company data and uploaded files via the administrator interface. Once the accuracy and authenticity of the information are verified, CSC will be able to accept or decline the account. The employers who have been approved will be given access to the TalentLink job posting module and rejected employers will be given a notification as to why they are rejected. Any kind of verification should be documented in the system log as a result of auditing.

### **6.2.2 CSC – Request Additional Employer Documents**

I would like to inform employers as an officer of the CSC that I may request them to provide additional documents in case the information they provide is incomplete or unclear so that all accounts of employers can be accurately verified before they are activated on the TalentLink system.

#### **Acceptance Criteria:**

The system must enable the CSC officers to reference an employer verification status to "Additional Documents Pending." CSC is able to include comments as to what documents are missing or require correction. The employer will be notified automatically asking to re-submit the necessary files. After the documents have been uploaded and checked, the status may be set to the Approved one. The requests and updates should be logged in order to trace them and audit.

## **6.3 Ng Xiang Ling TP082046**

### **6.3.1 Student – Manage Profile and CV**

As a student, I want to be able to manage my personal profile and upload my CV through the TalentLink system, so that I can ensure that the information I provide to employers is accurate and up to date.

#### **Acceptance Criteria:**

On the Profile page, I should be able to update my personal details, such as contact information and educational background. The system should also allow me to upload my CV in PDF format only, which will be stored securely in my account. Once the updates are saved, the revised information and the uploaded CV should be clearly displayed on the Profile page. The system should also show a confirmation message, such as “Profile and CV updated successfully,” to confirm that the update has been completed.

### **6.3.2 Student – View Application Status**

As a student, I want to be able to view the status of my job applications in the TalentLink system so that I can stay updated on the progress of my submissions.

#### **Acceptance Criteria:**

After submitting a job application, I should be able to see its current status displayed on the Application History page. Depending on the employer’s review, the system should automatically show one of the available statuses, such as 'Applied', 'Submitted' or 'Rejected'. Whenever the status changes, I should receive a notification from the system to keep me informed. This function allows me to easily check the outcome of my applications without needing to contact CSC manually for updates.

## **6.4 Woo May Eng TP082001**

### **6.4.1 Employer – Filter job application**

As an employer, when I want to view and filter job applications on the platform, so that I can easily identify qualified candidates for shortlisting.

#### **Acceptance Criteria:**

When the system displays all valid applications submitted for each employer's job posting. Employer able to filter (qualification, GPA, skills, submission date) to narrow results. System dynamically refreshes the list according to selected filters without full page reload. Employer can mark one or more candidates as "Shortlisted." The shortlisted candidates are saved to the database and accessible under the "Shortlist" tab. Finally, the system confirms successful shortlisting and logs employer actions for record tracking.

### **6.4.2 Employer – Schedule Interviews for shortlisted candidates**

As an employer, I want to schedule interviews for shortlisted candidates so that I can proceed with recruitment efficiently within the TalentLink platform.

#### **Acceptance Criteria:**

Employers select shortlisted candidates to create the interview scheduling form. System checks for existing conflicts before confirming interview date/time. Interview details (time, mode, link/location) are stored in the database. The student receives notification and confirmation via TalentLink system. Employers can reschedule or cancel interviews if needed. System automatically updates the candidate's application status to "Interview Scheduled."

## **6.5 Yap Li Shan TP080968**

### **6.5.1 Employer – Post a Job**

As an employer, I want to post a new job vacancy including title, job description, salary range, and work type at APUTalentLink so that I can attract suitable and outstanding applicants in less time.

Acceptance criteria:

1. The system should require employers to complete all mandatory fields such as job title, work type, salary range and location.
2. When the form information is incomplete, the system should display an error message and prevent users making submission.
3. Upon successful posting a new job, the job vacancy should immediately appear in the student dashboard's 'Recent Job List'.
4. The system should display a confirmation message to employer upon successful posting.

### **6.5.1 Employer – Manage Posted Jobs**

As an employer, I want to view, edit and delete my posted job so that I can manage position information up-to-date and remove jobs that are already closed or filled efficiently.

Acceptance criteria:

1. The system should display a list of all positions created by the employer, including both active and inactive vacancies.
2. Employers should be able to edit job details directly within the job listing, such as the job description, application deadline, or requirements.
3. The system should provide a deletion function for employers to remove completed or closed positions.
4. Following any operation, the system should provide timely feedback and response, such as displaying 'Job Updated Successfully' or 'Job Deleted Successfully'.

## **6.6 Kang Hong Qian TP081205**

### **6.6.1 CSC – Authorize Job Posting**

As a CSC employee, I would like to screen and approve the job listings provided by the employers in such a manner that valid and suitable job openings are reflected in the TalentLink system.

#### **Acceptance Criteria:**

The system should allow CSC officers to view the job postings and confirm them after it has verified that the employer is valid and all the details in the postings meet APU's specifications. The job is auto listed in student postings once confirmed and the employer receives the confirmation notification.

### **6.6.2 CSC – Suspend or Remove Position Advertising**

As a CSC member, I am to suspend or remove the job postings that violate APU's policies or give false information for the website to be professional and credible.

#### **Acceptance Criteria:**

CSC authorities should be in position to suspend or remove job listings that conflict with policies or provide false information. Once in suspended status, the listing is made temporarily unavailable to the students and the employer's account is notified along with the reason.

# **7.0 Agile Principles & Reflection**

## **7.1 Agile Principles**

### **7.1.1 Chan Min Huey TP083261**

#### **1. Customer satisfaction through early and continuous delivery of valuable software**

##### **Description:**

This agile principle emphasizes delivering functional modules of the system to user as early as possible to collect user feedback and enable continuous improvement. The development teams could ensure that user receive immediate value while enabling the system to improve based on real user needs. Early delivery allows stakeholders to test the features and build confidence in the system's progress.

##### **Example:**

In APU Talent Link project, this principle can be implemented through prioritize the development of functional modules like “Applying Jobs”. Functional modules like “Applying jobs” give direct benefits to the stakeholders like students and employers so the functional modules should be first implemented in the prototype system. By implementing these functional modules early, students and employers could first interact with the system and provide valuable feedback while the development teams could continuously refine and iterate other functionality and system updates.

This approach creates a continuous feedback loop between students and employers, enabling the development team to align with user's actual needs and continuously refine the system.

## **2. Welcome changing requirements, even late in development**

### **Description:**

This agile principle encourages the development team to maintain flexibility to changes in user needs or addressing new requirements, even though at the last stage of development. The agile teams don't adhere rigidly to fixed plans, but they do adjust the system based on feedback to ensure the final system closely aligns with what user needs to maximize values.

### **Example:**

New feedback and requirements may appear during the implementation of functional modules such as "Apply Jobs". For example, students might request additional features like bulk resume upload during the system implementation. As an agile development team, the team should accept it, adjust the backlog and prioritize the changes rather than rejecting them.

This flexibility ensures that the system implemented remains user-centric and continuously adapt changes to the growing needs of stakeholders like students, employers, and CSC service team even after the prototyping phase.

## **7.1.2 Cheong Pei Enn TP081926**

### **1. Projects are built around motivated individuals, who should be trusted**

#### **Description:**

This agile principle points out that agile development relies on motivated and accountable people that are provided with trust and free to accomplish their work. Team members will tend to be more accountable, creative and become more initiative when they believe they are the owners of their work. Trust allows making decisions quicker and promoting cooperation without having to watch each other.

#### **Example:**

In APU TalentLink project, the developers were each allocated with ownership of one module depending on their area of specialization. In the case of CSC verification feature, the designated body member was given the responsibility to design and implement the logic of employer approval by himself and liaise with others to integrate it. Instead of checking on them, regular stand up meetings were conducted to present progress and troubleshooting together instead of constantly checking on them. Enabling the members of the team to make decisions and hold responsibility, the project became more efficient and had a better sense of team ownership that directly influenced the final verification module quality.

## **2. Face-to-face conversation is the best form of communication (co-location)**

### **Description:**

This agile principle is to focus on the face to face in order to be able to communicate best and share ideas and solve problems in agile development. Discussion Verbal discussion can make requirements clearer in a shorter time period, eliminates misunderstanding, and builds teamwork unlike written communication, which can take a long time.

### **Example:**

In the process of creating the CSC verification system, developers and CSC representatives discussed data fields, document forms, and rules of approval in the flesh quite often to ensure that the system was correct. Through these meetings, instant feedback and rapid changes to the verification process without having to wait until a reply was sent by email were achieved. Through their constant face-to-face communication, the team did not have to worry as much about technical implementation and administrative requirements not aligning so that the process of verification in the TalentLink system was more accurate and user-friendly.

### **7.1.3 Ng Xiang Ling TP082046**

#### **1. Working software is delivered frequently (weeks rather than months)**

##### **Description:**

This agile principle focuses on delivering smaller and functional parts of the system at regular intervals instead of waiting until the whole project is completed. This enables the team to obtain feedback early, make necessary adjustments quickly and ensure that the development remains on track. It also enables identification and correction of potential issues earlier, which makes the overall stability and quality of the end product even greater.

##### **Application in APU TalentLink Project:**

In the APU TalentLink project, this principle was used in early sprints by creating important modules such as "Manage Profile" and "Upload CV." It allows students to update their personal details and upload their CV before the complete system is ready. Releasing these components early allowed students and staff of CSC to test and provide feedback, which assisted the team in making timely improvements.

Frequent deliveries ensured that the project maintained steady progress and increased user confidence since students and CSC could review the real updates and improvements throughout development.

## **2. Close, daily cooperation between businesspeople and developers**

### **Description:**

This agile principle emphasises the importance of close and regular collaboration between developers and business stakeholders to make sure that the system can cater to real business needs. Frequent communication helps to avoid misunderstanding and allows the team to respond to changes in requirements easily.

### **Application in APU TalentLink Project:**

In the APU TalentLink project, the development team collaborated with the Career Services Centre (CSC), which served as the business representative. Meetings were scheduled regularly to ensure that profile fields, CV format and data handling procedures in the "Manage Profile" module were correct. In this interaction, developers were able to make improvements to the prototype given a review by the CSC so that the system could satisfy both technical goals and expectations from the end-users.

This collaboration helped the team to maintain a practical design that met the requirements of both the CSC and the institution, as well as the needs of students.

## **7.1.4 Woo May Eng TP082001**

### **1. Working software is the principal measure of progress**

The objective of using this principle is to ensure that the system delivers a minimal but fully functional filtering interface in the first sprint: list view, basic filters, and shortlists flag during the shortlisting module. This working increment becomes the measures of progress rather than extensive documentation. Employers can use the features immediately as their feedback will guide enhancements. This principle emphasizes that they measure their progress by measuring the amount of software that is meeting the user's need. By delivering the shortlist capability early allows the project to validate assumptions and avoid building unused features.

#### **Application in APUTalentLInk project:**

- Incremental Feature Delivery: The team focused on implementing a system which functions job application viewing page with filter options. This allowed users to experience a usable feature within the early stages of development.
- Feedback Improvement: After the sprint, the development team collect feedback and comments from the users like students, employers, and CSC staff to improve the filter criteria, layout, and shortlist confirmation module. This feedback helps to ensure the system aligns with real operational needs.
- Progress Measurement through Functionally: Instead of lengthy documentation, progress was demonstrated through working functions such as “View Applications”, “Shortlist Candidates”, and “Schedule Interview”, which indicates the main functionality is working.

Through APU TalentLink project working, ensured that development outcomes were measurable, usable and continuously validated by actual system performance.

## **2. Sustainable development, able to maintain a constant pace.**

The objective of using this principle is to ensure that the development will run fast, but sustainable sprint with a realistic scope for each sprint like filter refinement, shortlist persistence, scheduling. Automated tests and quality checks will be integrated to prevent technical debt. This helps teams to avoid overtime crunches and ensures continuous delivery on time. This principle highlights sustainable pace and technical excellence as essential for long-term agility. A steady development reduces defects and supports iterative improvements to the workflow.

### **Application in APUTalentLink project:**

- Realistic Sprint Planning: Each sprint focused on the specific user stories, such as “Filter Applications” or “Schedule Interview,” ensuring they are completed within two weeks while maintain a balanced workload.
- Technical Excellence and Automation: Regular test runs were performed to detect integration issues early, thus maintaining the quality of APU TalentLink system.
- Team Well-Being and Efficiency: Workloads were evenly distributed and avoiding overtime so as to ensure continuous improvement. To promote sustainability development, the team reduced defect rates, maintain the highest quality standards for the duration of the project, and motivate all team members.

Through continuous integration and automated testing practices ensured stable progress without sacrificing quality. Maintaining a constant rhythm across sprints allowed the team to adapt quickly to feedback while avoiding excessive workload.

## **7.1.5 Yap Li Shan TP080968**

### **1. Continuous attention to technical excellence and good design**

This principal highlight that throughout the entire development process, agile team members should be dedicated to concentrating on technical excellence and good design. This is because well-written code has many advantages, including increased dependability and ease of maintenance and expansion. To guarantee continued high performance and sustainability as the system develops, agile team members should also continuously pledge to write clear, organized, and reusable code.

Application in APUTalentLink project:

- Modular Design: We divide the system into distinct modules such as Application Management, Job Listing Management, and Notification Modules, ensuring clearer code structure and facilitating future maintenance.
- Continuous improvement of code quality: Following the completion of each Sprint, agile team members will conduct code reviews to ensure all codes follow naming conventions, logical clarity, and optimization of database queries for enhanced operational efficiency.
- Adopting consistent UI/UX design: Throughout the prototyping phase, we maintain consistent visual design and interaction logic to minimize user learning curves and enhance the user's overall experience.

Through continuous attention to excellent technical and good design, the APU TalentLink has seen significant improvements in both performance and stability.

## **2. Simplicity — the art of maximizing the amount of work not done is essential**

This principle highlights that one of the main goals of agile development is "simplicity." It highlights that teams should avoid adding superfluous or overly complicated features instead of concentrating on the system's primary goal and requirements. Teams can produce high-quality deliverables while saving time and money by choosing the most straightforward solution that satisfies requirements. This increases project flexibility and maintainability.

Application in APUTalentLink project:

- Focus on Core Features: During the initial planning phase, the team developed only core functionalities such as "Job Search", "Job Application" and "Account Registration". Features like video interviews and personalized recommendations place on a future expansion plan to ensure rapid prototype development.
- Simplified User Flow: Interface design follows the KISS principle, minimizing unnecessary clicks. For example, students could complete a job application in just two steps.
- Efficient Work Allocation: Within each Sprint, we prioritized tasks with the highest impact, reducing redundant work and the development of low-value features.

By following the principle of simplicity, we ensure that the project team can deliver high-quality project prototypes within a limited timeframe.

## **7.1.6 Kang Hong Qian TP081205**

### **1. Best architecture, requirements, and designs emerge from self-organizing teams**

The best solutions come from self-organizing teams: team members take ownership together, make decisions together, and govern by themselves. With this liberty, teams will also boost their motivation, quick problem solving, and production that satisfies real needs with minimal top-down management.

Application in APUTalentLink project:

- Domain teams: Job Posting, Application Management, and CSC Monitoring take charge of quality and make local tech/design choices.
- Tight UI-backend integration: The Job Posting team established its own schema and validation rules and worked closely with the UI team to ensure everything worked smoothly.
- Outcome: Quicker decisions, increased clarity on accountability, and more aligned architecture to the stakeholders (which are students, employers, CSC) needs.

Less rework and handoff delay; design changes within a sprint and more stakeholder satisfaction with shipped features.

## **2. Regularly, the team reflects on how to become more effective and adjusts accordingly**

Continuous improvement by periodic retrospectives—list what worked and what did not and make concrete commitments to action—such that efficiency and quality increase with time.

Application in APUTalentLink project:

- Sprint retrospectives: Brief retrospectives at the end of each sprint identified latency and integration problems (such data incompatibilities between the student and employer modules).
- Specific fixes: Version-control processes were streamlined by the team and task accountability was established to avoid replicated mistakes.
- Outcome: Fewer defects, smoother handovers, and more consistent momentum across iterations.

Reduced defect rate and cycle time; simplified ownership minimized blockers and on-time delivery improved through the sprints.

## 7.2 Information System Methodology Reflection

### 7.2.1 Chan Min Huey TP083261 – Waterfall vs Scrum

Methodology	Waterfall	Scrum
<b>Development Speed</b>	Slower development, as the waterfall phases (requirements, design, implementation, testing and deployment) must be completed in order before proceeding to the next phase.	Faster than waterfall, as it develops system with iterative sprints that deliver small but working increments of the system regularly.
<b>Project Size</b>	Suitable for large-scale projects with clear and stable requirements.	Suitable for small to medium-sized projects, or those with dynamic and frequently changing requirements.
<b>Flexibility</b>	Very low flexibility, as any change in requirements requires to visit back earlier stages, causing high rework costs.	High flexible as the requirements may change during the project, and feedback is included in each iteration cycle.
<b>User Involvement</b>	Low user involvement because the user only participates in the requirements gathering stage and the final stage.	High user involvement because the users are continuously involved in the sprint planning, review, and feedback cycles stages
<b>Risk</b>	Higher risk, since the testing only occurs in the last stage, the issue of the system might not be discovered beforehand.	Lower risk, as early and frequent testing in every sprint allows issue to be detected and corrected in earlier stage.
<b>Quality Control</b>	Quality control is conducted near the end of the development phase, which may cause defects to be detected late.	Quality is always validated through sprint reviews, daily stand-ups, and testing regularly after each iteration.

<b>Focus</b>	Focus on completing all project phases in order, ensuring documentation and process compliance.	Focus on delivering functional, usable software rapidly while adapting to user needs and priorities.
<b>Example in APU TalentLink</b>	Suitable to use at the beginning stage of the system planning to establish a clear understanding of system requirements, database structure, and documentation.	Suitable for prototype and enhancement stages, where team collaboration, flexibility, and feedback help to refine user experience and functionality.

Both Waterfall and Scrum methodologies have their own strategies that are suitable for different ideal scenarios. Waterfall is more suitable for project with clear and stable requirements that require extensive validations like the earlier stages of the APU TalentLink System. It has structured, linear processes and comprehensive documentations that ensure the earlier stage of APU TalentLink System have a well-defined architecture design and accurate data modelling. While Scrum are more suitable for project where the requirements may have changes in the later stages like the later development of APU TalentLink's user interface, job posting and student application features. It emphasizes collaboration, rapid iteration, and agility during the implementation of the system.

Therefore, I recommend combining both approaches, which is to apply Waterfall methodology during the early phase of system development for solid planning and stable foundation and Scrum methodology during the later stage for enhance user engagement and iterative improvement. This hybrid approach let the APU TalentLink system to create a dynamic platform by combining both technical stability and user-oriented flexibility.

## 7.2.2 Cheong Pei En TP081926 – Hybird Model vs. Agile

Methodology	Hybird Model	Agile
<b>Development Speed</b>	Slower than Agile, as Hybrid Model involves structured planning, documentation, and milestone approvals before each sprint is executed.	Faster than Hybrid, as Agile focuses on short, iterative sprints that deliver working system components frequently.
<b>Project Size</b>	Suitable for medium to large-scale projects that require detailed documentation and controlled progress.	Suitable for small to medium-sized projects that evolve quickly and depend on frequent feedback from users.
<b>Flexibility</b>	Provides moderate flexibility since most requirements are fixed during the planning stage, with limited adjustments allowed in later sprints.	Highly flexible because requirements can be changed, reprioritized, and refined during any sprint cycle.
<b>User Involvement</b>	Low to medium user involvement, as users mainly participate during sprint reviews or final testing.	High user involvement, as users are consistently engaged in sprint planning, review, and feedback sessions.
<b>Risk</b>	Lower risk compared to traditional sequential models, as iterative sprints help detect and resolve issues earlier, but still depends on accurate initial documentation and proper coordination between stages.	Lower risk because problems can be identified and corrected quickly through frequent testing, sprint reviews, and continuous feedback from users.
<b>Quality Control</b>	Quality control is maintained through milestone reviews and combined documentation checks at the end of each sprint.	Quality is ensured through continuous testing, peer reviews, and sprint retrospectives after every iteration.

<b>Focus</b>	Focuses on balancing structured control with adaptive improvement to maintain system reliability and documentation consistency.	Focuses on adaptability, teamwork, and continuous improvement to deliver working software faster.
<b>Example in APU TalentLink</b>	Used in the CSC verification module at the planning stage to establish structured employer validation rules and documentation before moving into iterative sprints for refinement.	Suitable during the prototype and testing stages, where frequent feedback from CSC officers helped refine the verification dashboard, user interface, and notification accuracy.

In conclusion, both Hybrid Model and Agile methodology have their own strengths and areas where they can be used. Hybrid Model is more focused on planning and documentation and controlled progress so that it suits the initial stage of development of the APU TalentLink system, particularly developing the CSC verification rules, workflow, and data validation processes to be reliable and consistent. Agile, on the other hand, facilitates flexibility, high-speed iteration, and constant user engagement, which is why it is best suited to the improvement and testing stages, where frequent feedback is used to revise the verification dashboard and make the entire usability more seamless.

I would suggest using the combination of both methodologies and using the Hybrid approach in the first stages of development to avoid instability and a clear system structure, and Agile approaches in the next stages of improvement to make it more flexible and responsive. Such an integration allows the APU TalentLink project to be reliable and flexible at the same time, which leads to a stable, but user-friendly recruitment environment.

### 7.2.3 Ng Xiang Ling TP082046 – Kanban vs Prototyping

Methodology	Kanban	Prototyping
<b>Development Speed</b>	Progress is continuous but moderate, with tasks moving through stages rather than fixed sprints.	Development starts quickly, as prototypes are built early for user testing and feedback.
<b>Project Size</b>	Suitable for medium to large projects that require regular updates and maintenance.	Suitable for small to medium projects requiring early design validation and frequent feedback.
<b>Flexibility</b>	Highly flexible, as changes can be made anytime based on user needs or project progress.	Moderately flexible, mainly during the design phase but becomes more stable once the prototype is approved.
<b>User Involvement</b>	Users mainly contribute by providing feedback on system updates.	Users, such as students and CSC staff, are directly involved in reviewing and improving prototypes.
<b>Risk</b>	Lower risk because continuous tracking helps detect and fix issues early.	Moderate risk if user feedback is unclear or changes are not implemented correctly.
<b>Quality Control</b>	Quality is maintained through visual task tracking and workflow limits on the Kanban board.	Quality depends on regular user evaluation and improvement of each prototype version.
<b>Focus</b>	Focuses on improving process efficiency and maintaining workflow balance.	Focuses on system design, usability and user satisfaction.
<b>Example in APU TalentLink</b>	Suitable to use after the system deployment stage to manage ongoing updates, monitor workflow progress and ensure continuous improvement.	Suitable for the early design and development stages to quickly build, test and refine system interfaces based on feedback from students and CSC staff.

In conclusion, both Kanban and prototyping have their own strengths and are suitable for different stages of system development. Kanban focuses on visualising tasks, improving workflow and maintaining steady progress, making it suitable for later stages such as system updates and maintenance. Conversely, prototyping focuses on creating early versions of the system, collecting user feedback and improving features through testing. This makes it suitable for the early design and development stages of the APU TalentLink system.

Therefore, I recommend combining both methods in the APU TalentLink project. Prototyping could be employed initially to design and test key functions, while Kanban could be used subsequently to manage progress and facilitate continuous improvement. This combination will help the system to remain flexible, efficient and user-friendly throughout the development process.

## 7.2.4 Woo May Eng TP082001 – Scrum vs Rapid Application Development (RAD)

Methodology	Scrum	Rapid Application Development (RAD)
<b>Development Speed</b>	Steady and controlled sprint within 2-4 weeks, progress steady and measurable.	Very fast, emphasizing urgent projects within days or weeks.
<b>Project Size</b>	Suitable for small to medium-sized projects that require continuous improvement.	Best for short-term and urgent projects where the goal is immediate development and early visualization of system functions.
<b>Flexibility</b>	High flexibility within Sprint boundaries as new requirements are added in the next iteration.	Very high flexibility as users can suggest changes during prototype cycles, but frequent revisions may cause inconsistencies.
<b>User Involvement</b>	Continuous involvement through product owner, scrum master, and development team work closely in each sprint.	During the design, JAD brings users into the development process as active participants.
<b>Risk</b>	Lower risk. Every sprint includes testing and review ensuring that progress is stable and early problem detect.	Medium risk. The rapid pace may mask architectural or integration problems leading to technical debt.
<b>Quality Control</b>	Emphasize continuous quality, integration testing to improve development.	Quality control relies heavily on prototype testing and user feedback, lack of formal testing and documentation limited in initial cycles.
<b>Focus</b>	Focus on sustainable development and continuous	Focus on speed and user satisfaction.

	development.	
<b>Example in APU TalentLink</b>	Suitable for modules like Employer Shortlisting and Scheduling, where each sprint refines and adds working features with CSC feedback.	Suitable for initial prototyping of student dashboards or job search interfaces to collect feedback on the system usability.

In conclusion, Scrum and Rapid Application Development (RAD) are user-centered, but they are two different development philosophies. RAD prioritizes speed over structure using quick design and prototype cycles to produce urgent projects. It targeted to gain early approval from stakeholders but may risk technical problems if applied for integration project. In contrast, Scrum emphasized control overall progression. Through defined sprints, regular progression, Scrum ensures sustainable rhythm of progress and maintains project quality.

In the APU TalentLink project, adopting Scrum is more suitable for the long-term system development. This is because Scrum delivers a steady, controlled rhythm with low risk and high-quality control to ensure sustainable progress and continuous stakeholder collaboration.

## 7.2.5 Yap Li Shan TP080968 – V-model vs. RAD

Methodology	V-Model	Rapid Application Development (RAD)
Development Speed	Slower, each phase must experience thorough validation and documentation before proceeding to the next stage.	Faster, employing short-cycle iterations to deliver results swiftly through prototyping.
Project Size	Suitable for large and structurally complex systems with stable requirements.	Suitable for small to medium-sized projects with frequently changing requirements.
Flexibility	Low flexibility, if client's requirement changes it require multiple stages of redesign.	High flexibility, allowing modifications at any stage based on user feedback and requirement.
User Involvement	Lower, users primarily participate during system testing or user acceptance testing phases.	Higher, users have been continuously involved in the project evaluation and feedback process since the early stages of development.
Risk	Low risk, fault and failure are mitigated through phased testing and documentation control.	Medium risk, although documentation is less than V-model, direction can be quickly adjusted via rapid feedback.
Quality Control	Quality control is stringent, with formal validation and testing at every stage to ensure system reliability.	Quality control relies on user feedback and continuous improvement, rather than formal validation processes.
Focus	Focus on the system's accuracy, stability and compliance.	Focus on development speed, usability and user satisfaction.
Example in APUTalentLink	Suitable for the early stages, ensuring the stability of database design, system validation, and core functionality.	Suitable for the later stages, optimizing interfaces and user experience through rapid prototyping.

To sum up, the V model and Rapid Application Development (RAD) each have their own advantages and are applicable in different scenarios. The V model is applicable to the initial development stage of the APU TalentLink system because it emphasizes a systematic and structured methodology and a high-quality verification process. These elements are crucial for ensuring system reliability, data accuracy and long-term functional soundness. However, during the system iteration and optimization stage, RAD is more applicable because it emphasizes rapid delivery, short iteration cycles, and deep user participation, focusing on the continuous improvement of user interfaces, interaction experiences, and functional responsiveness.

My suggestion is to adopt both methods simultaneously. In the early stage of system development, the V model will help establish a solid infrastructure. In the later stage, the Rapid Application Development (RAD) method is employed to achieve a rapid iterative cycle of user feedback and optimization. This approach will help APU TalentLink strike the best balance between a high-quality system and an efficient development process, ultimately creating a reliable, efficient and user-friendly online recruitment platform.

## 7.2.6 Kang Hong Qian TP081205 – Extreme Programming (XP) vs Waterfall

Methodology	Extreme Programming (XP)	Waterfall
<b>Development Speed</b>	Rapid growth through short cycles and continuous integration.	Slower because each stage must be complete before it's time to move ahead.
<b>Project Size</b>	Able to accommodate small to medium-sized projects for regular updates.	Able to accommodate large projects that have definite fixed requirements.
<b>Flexibility</b>	Extremely adaptable; modifications are possible anywhere by using the iterative cycle.	Low flexibility: modifications after design are time-consuming and expensive.
<b>User Involvement</b>	High involvement: the users provide feedback throughout the iteration.	Low involvement: the users are involved mostly during requirement and test phases.
<b>Risk</b>	Low risk due to the continuous verification and checking by TDD (Test-Driven Development).	High risk because testing will have to wait until final development.
<b>Quality Control</b>	Quality ensured by continuous testing and pair programming.	Quality verified later, greater likelihood of discovering defects late.
<b>Focus</b>	Place a strong emphasis on teamwork, feedback, and early delivery of quality code.	Focus completing each structured stage properly documented.
<b>Example in APUTalentLink</b>	Suitable for ongoing improvement of system modules like job application and student notification features.	Suitable for early system planning, defining architecture, and setting database design.

Conclusion:

Extreme Programming (XP) and the Waterfall model also have distinct advantages for the APUTalentLink system. Waterfall provides a clear, structured format that is ideal for the upfront planning and system design phases to clearly delineate the database and requirements. XP permits rapid construction, frequent testing, and feedback by the users and is highly applicable to the construction and refinement phases where rapid response and flexibility are key.

Therefore, a hybrid approach is proposed—with Waterfall in early development phases for establishing a stable foundation, and XP in later phases for iterating and building functionality incrementally from sustained end-user feedback. This hybrid retains reliability for the system but allows adaptive growth for APUTalentLink platform.

## 8.0 System Design

### 8.1 Design Model

#### 8.1.1 Chan Min Huey TP083261 – UML Sequence Diagram (Apply Job)

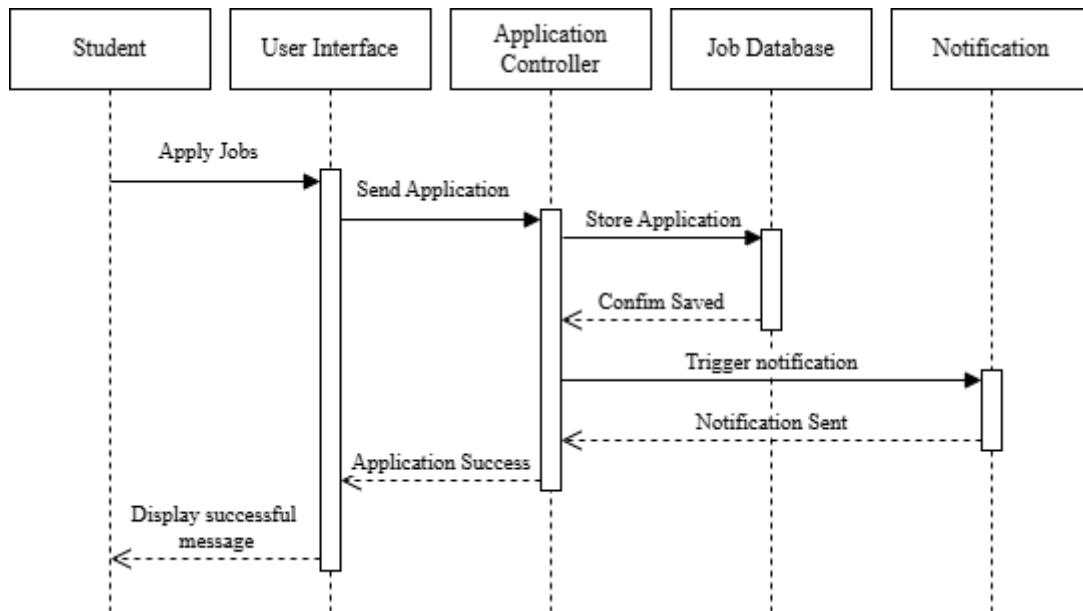


Figure 8.1.1 UML Sequence Diagram (Apply Job)

This sequence diagram shows how the APU TalentLink system manages the student job application process, with the interaction among the actors: students, user interface, application controller, job database and notification. Student act as the starting of the job application process, then the user interface handles the student's action and provide outputs, application controller processes the application request and manage communications with job database and notification, and lastly the notification send confirmation to the student after successful submission.

## Sequence Flow Explanation

### 1. Apply Jobs (Student → User Interface):

Students selected a job from the job list via the user interface and click on the “Apply” button to start the application process.

### 2. Send Application (User Interface → Application Controller):

The user interface forwards the student’s job application details like CV to the application controller and requests the application controller for further processing.

### 3. Store Application (Application Controller → Job Database):

The application controller records and stores the student’s application information into the job database.

### 4. Confirm Saved (Job Database → Application Controller):

The job database confirms that the application has been successfully saved and sends a confirmation message back to the application controller.

### 5. Trigger Notification (Application Controller → Notification):

After receiving confirmation from the job database, the application controller triggers the successful notification message to inform the students.

### 6. Notification Sent (Notification → Application Controller):

The notification sends back a message to indicate that the notification has been successfully delivered.

### 7. Application Success (Application Controller → User Interface):

The application controller sends a successful response to the user interface to indicate that the job application process was completed.

### 8. Display Successful Message (User Interface → Student):

The user interface displays a successful message to the students to confirm that the job application has been successfully submitted.

## 8.1.2 Cheong Pei Enn TP081926 – UML Class Diagram (Employer Account Verification)

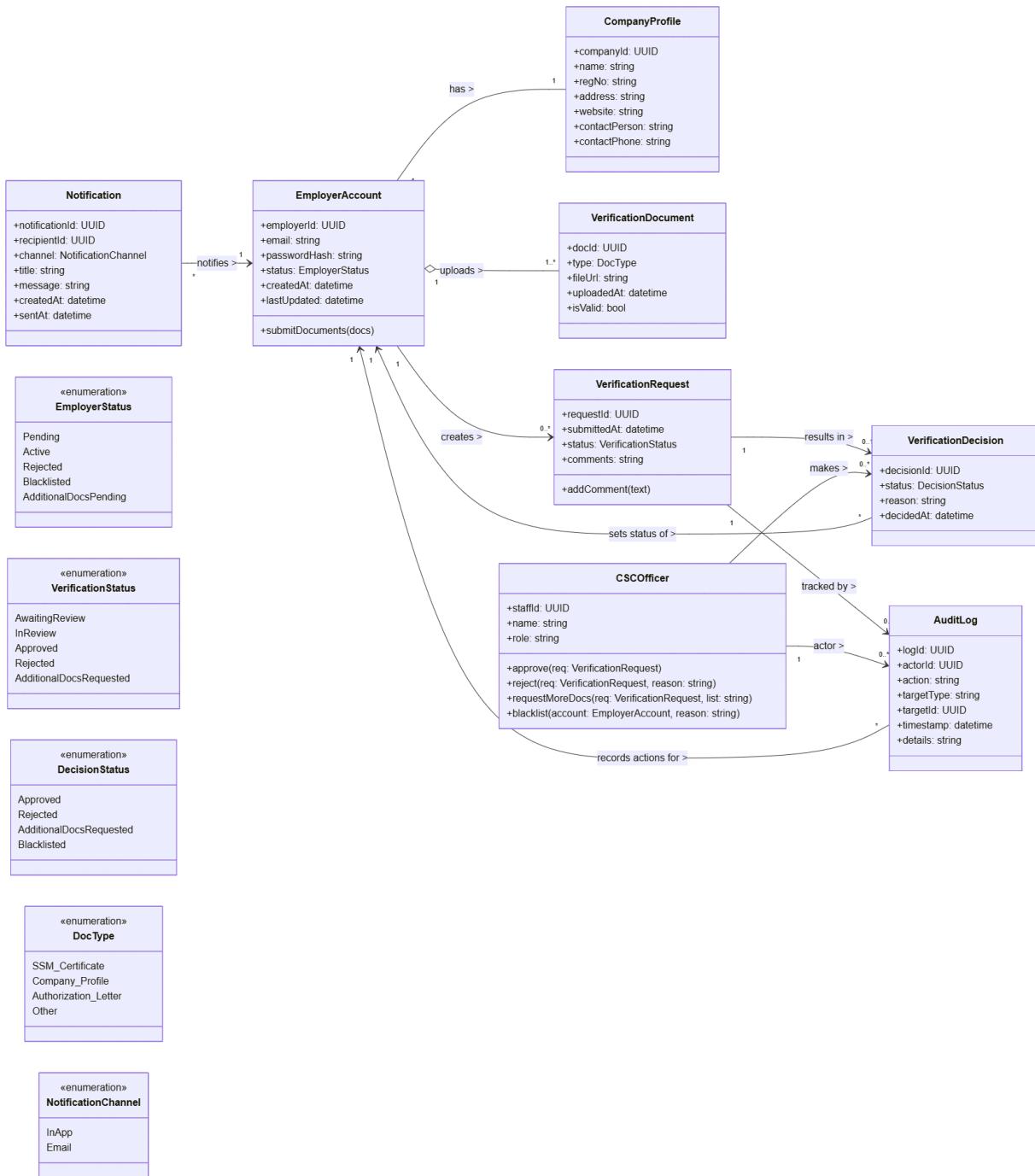


Figure 8.1.2 UML Class Diagram (Employee Account Verification)

The following class diagram shows how the APU TalentLink system manages the verification of employer accounts. It determines the hierarchy and relationship between major classes: EmployerAccount, CompanyProfile, VerificationDocument, VerificationRequest, VerificationDecision, CSCOfficer, Notification and AuditLog. The diagram displays the

process in which an employer registers and uploads the company documents, and makes a verification request, which is subjected to the CSC Officer. The CSC Officer goes through the documents and makes a decision on a verification and updates the status of accounts of the employer. The system also sends notifications to notify the employer about the results and logs all activities to the audit log to be transparent and traceable. The data consistency between the various phases of the process is maintained by a variety of enumeration classes like EmployerStatus, VerificationStatus, DecisionStatus, DocType and NotificationChannel which are fixed sets of values. In general, this class diagram illustrates the rational organization and interrelationship that guarantee safe and orderly employee verification process in the TalentLink system.

### Class Relationship and Process Explanation

#### 1. Employer Account Creation (EmployerAccount → CompanyProfile):

The employer registers a new account and forms a company profile on the same, where business information like name, registration number, and contact information is stored.

#### 2. Document Submission (EmployerAccount → VerificationDocument):

The employer posts necessary papers (SSM certificate, authorization letter). Such files are saved as the instances of the VerificationDocument class.

#### 3. Verification Request (EmployerAccount → VerificationRequest):

The employer placed a request of verification. This request captures the submission details, referencing documents, and verified or not.

#### 4. Request Processing (VerificationRequest → CSCOfficer):

The CSC Officer then checks the pending requests and verifies the documents presented and changes their status to “In Review”.

#### 5. Decision Making (CSCOfficer → VerificationDecision):

The officer endorses, declines or demands more documents. The decision, rationale and the date are stored in the VerificationDecision class.

#### 6. Account Status Update (VerificationDecision → EmployerAccount):

Depending on the decision made by the officer, the account of the employer is

changed (Approved, Rejected, or AdditionalDocsPending).

7. Notification Trigger (VerificationDecision → Notification):

Upon making a decision, a notification about it is automatically forwarded to the employer by the system via the Notification class in the format of an in-app message or email.

8. Audit Logging (CSCOfficer → AuditLog):

All the activities performed throughout the verification process are logged in the AuditLog class, which is accountable and provides traceability of the system.

### 8.1.3 Ng Xiang Ling TP082046 – UML Activity Diagram (Manage Profile and CV)

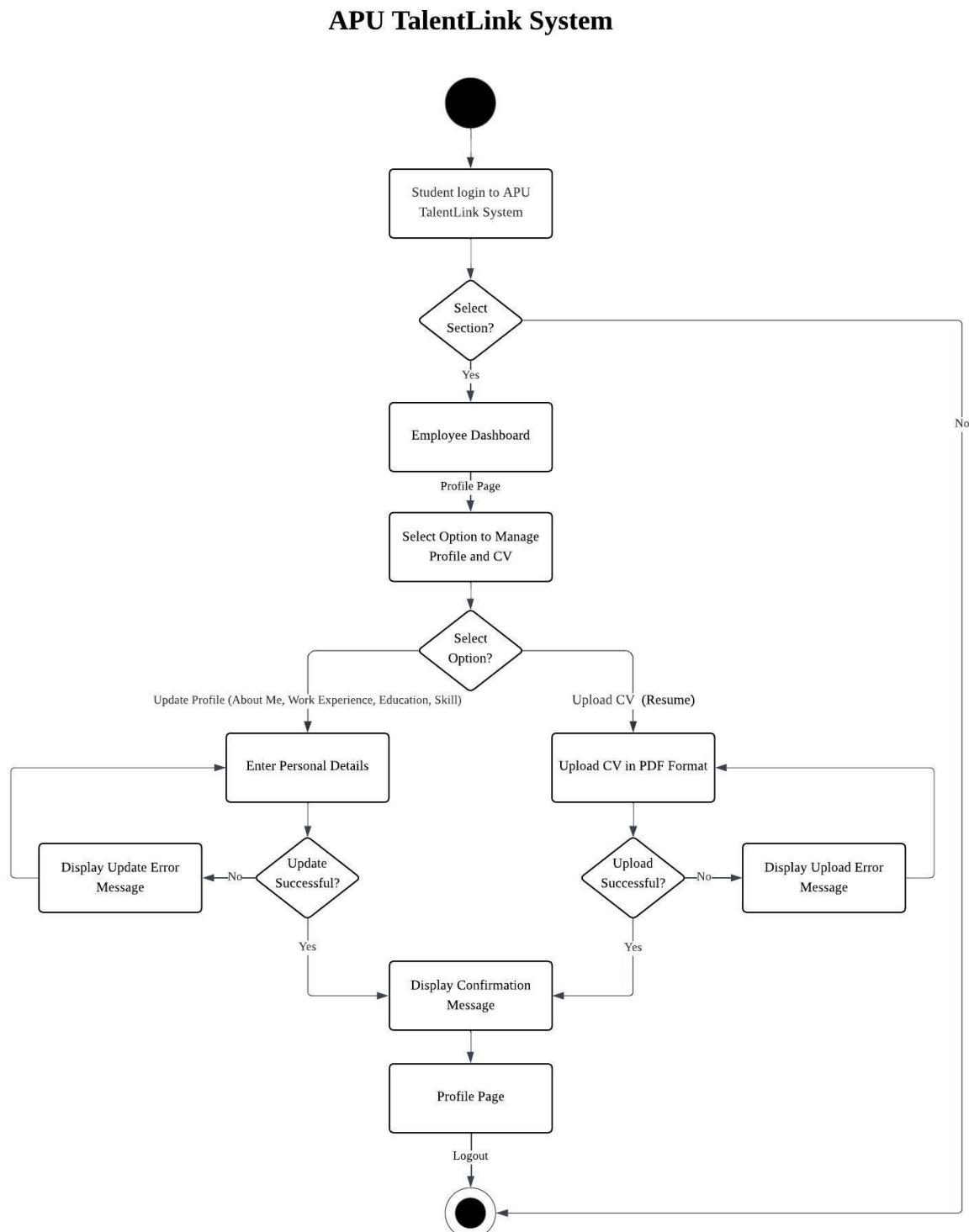


Figure 8.1.3 UML Activity Diagram (Manage Profile and CV)

This activity diagram illustrates the workflow of the “Manage Profile and CV” function in the APU TalentLink system. The process begins when a student logs in to the system and accesses the profile management section. The student can choose to update their personal details, such as About Me, Work Experience, Education and Skills. They can also upload a new CV in PDF format on the same page. The system will verify the entered data or check the uploaded file format and size. If the update or upload is successful, the system will display a confirmation message to the student. Otherwise, an error message will be shown. This diagram demonstrates how the system supports students in managing their profiles efficiently and ensures data accuracy through validation steps.

#### Process Flow:

1. The student logs in to the APU TalentLink system.
2. If login is successful, the system displays the Student Dashboard, and the student can navigate to the Profile Page.
3. The student can select the option that they want to update.
4. If students want to update their profile, they can select any of the options: About Me, Work Experience, Education and Skills to modify the information they want to update.
5. The system will verify the input data.
6. If the update is successful, the system will display a confirmation message to the student. Otherwise, an error message will be shown.
7. If students want to upload a new CV, they can select the Resume option to upload a new CV file in PDF format.
8. The system will check the file format and upload successfully.
9. If the file is valid, the CV will be saved in the system and the system will display a confirmation message to the student. If not, an error message will be shown.
10. Finally, the student is redirected to the Profile Page and may log out of the system.

### 8.1.4 Woo May Eng TP082001 – Use Case Diagram (Scheduled Interview for Shortlisted Candidates)

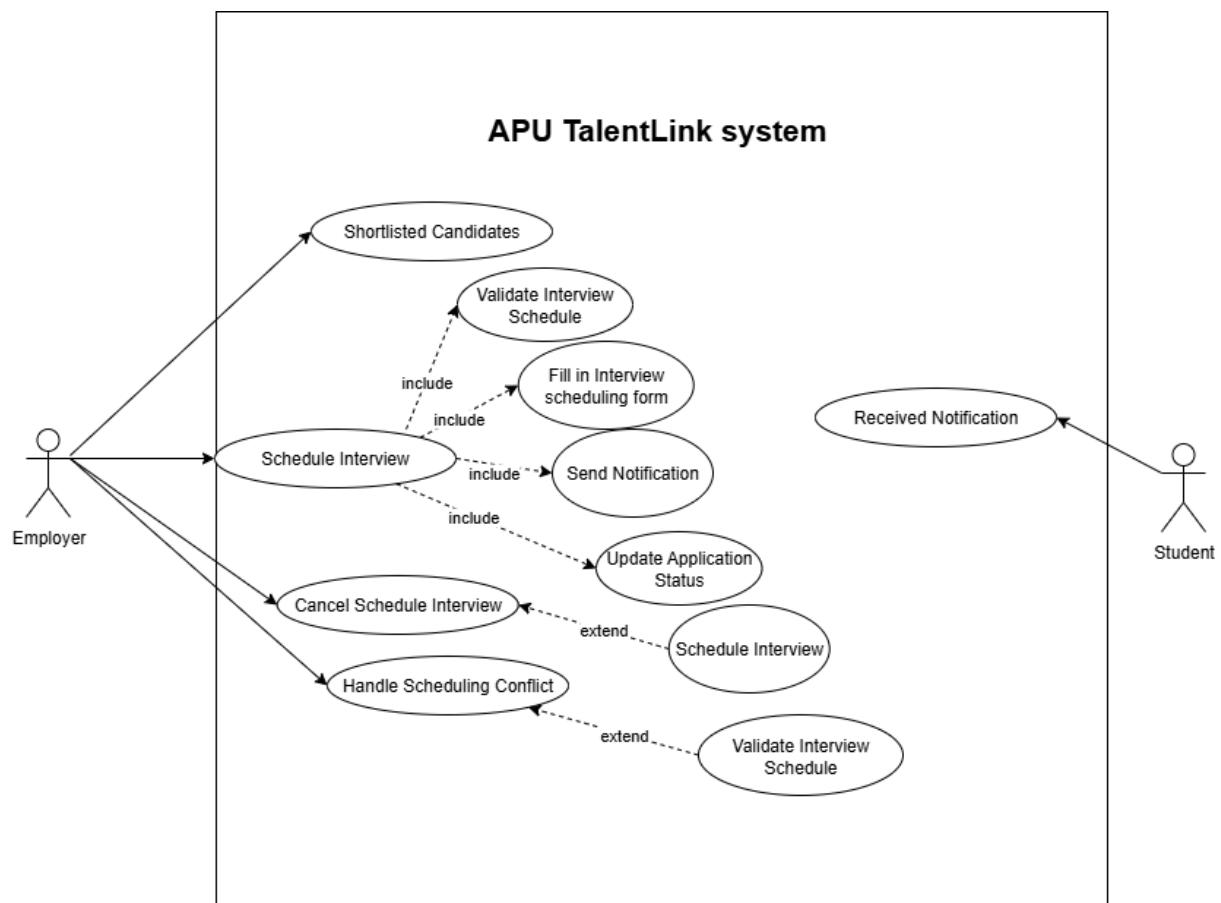


Figure 8.1.4 Use Case Diagram (Scheduled Interview for Shortlisted Candidates)

#### Use Case Description

Use Case Name	Scheduled Interview for Shortlisted Candidates
Actor	Employer
Description	Allow employers to schedule interviews for shortlisted candidates within the APU TalentLink system.
Precondition	The employer has successfully logged into the system and has shortlisted candidates.
Main flow	<ol style="list-style-type: none"> <li>1. The employer clicks on the “Shortlisted Candidates” button.</li> <li>2. The employer selects candidates and click on “Schedule Interview” button.</li> <li>3. The system displays the interview scheduling form with</li> </ol>

	<p>details such as date, time, mode (online/physical), and location/link.</p> <ol style="list-style-type: none"> <li>4. Employer fills in the interview form and submits it.</li> <li>5. The system validates form and checks conflicting interview schedules.</li> <li>6. The system updates the candidate's application status to "Interview Scheduled."</li> <li>7. The system sends notifications to both the candidate and the employer regarding the scheduled interview.</li> </ol>
Alternative flow	<p>5(a). If a scheduling conflict is detected (e.g., overlapping date/time), the system displays a warning message for the employer to select a new slot.</p> <p>6(a). If the employer cancels the scheduling, the system returns to the "View Shortlisted Candidates" page without any changes.</p>

The diagram illustrates the use case relationships of the Employer and the Candidates within the APU TalentLink system when scheduling interviews for shortlisted candidates. The main flow "Schedule Interview" function by employer is monitored by the system. When the scheduling is validated, the system automatically sends notification to shortlisted candidates and the student's application job status are updated.

In conclusion, this Use Case Diagram effectively demonstrates the structured interaction between the employer and the APU TalentLink system during the interview scheduling process. The system operating by automatically checking for scheduling conflicts, updating candidate status and generating real time notifications to all relevant parties.

### 8.1.5 Yap Li Shan TP080968 – Use Case Diagram (Post a Job)

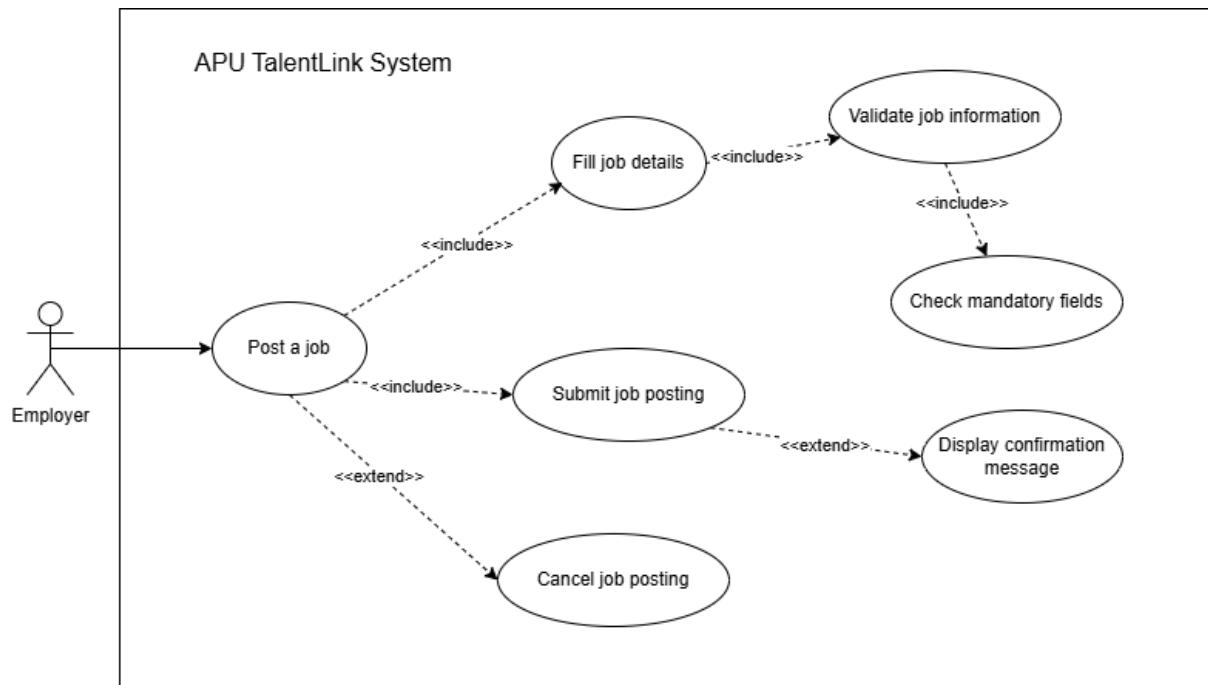


Figure 8.1.5 Use Case Diagram (Post a Job)

#### Use Case Description

Use Case Name	Post a Job
Actor	Employer
Description	Allow employers to create and post new job vacancies within the APU TalentLink system.
Precondition	The employer has successfully logged into the system and done account verification.
Main flow	<p>8. The employer clicks the 'Post a Job' button.</p> <p>9. The system displays the job posting form.</p> <p>10. The employer fills in the job details (e.g. job title, salary, type, description).</p> <p>11. The system validates the information and checks for mandatory fields.</p> <p>12. The employer submits the job posting.</p> <p>13. The system displays a confirmation message stating 'Job Posted Successfully'.</p>
Alternative flow	4(a). If any mandatory fields are incomplete, the system will

	display an error message and require employer to complete. 5(a). If the employer cancels the operation, the system will return to the employer's job listing page.
--	---

This diagram shows the use case association when employers use the "Post Job" function in the APU TalentLink system. The employer interacting with the system is a participant in this use case relationship. When the employer clicks the "Post Job" button, they need to fill in the job information, verify the input content, and then submit the form. After the form is filled out, the system will automatically execute the verification process to ensure that all required fields are complete, and the input format is correct. If the verification is successful, the system will inform the employer that the job Posting submission is successful. If the employer cancels an operation at any stage of the process, the extended use case named "Cancel Job Posting" will call the first participant, causing the system to return to the job list interface.

In summary, the use case diagram demonstrates how the system ensures the integrity of job information and the reliability of system operation during the job Posting process through steps such as verifying data, alerting users to errors, and providing immediate responses.

### 8.1.6 Kang Hong Qian TP081205 - Sequence Diagram (Approve Job)

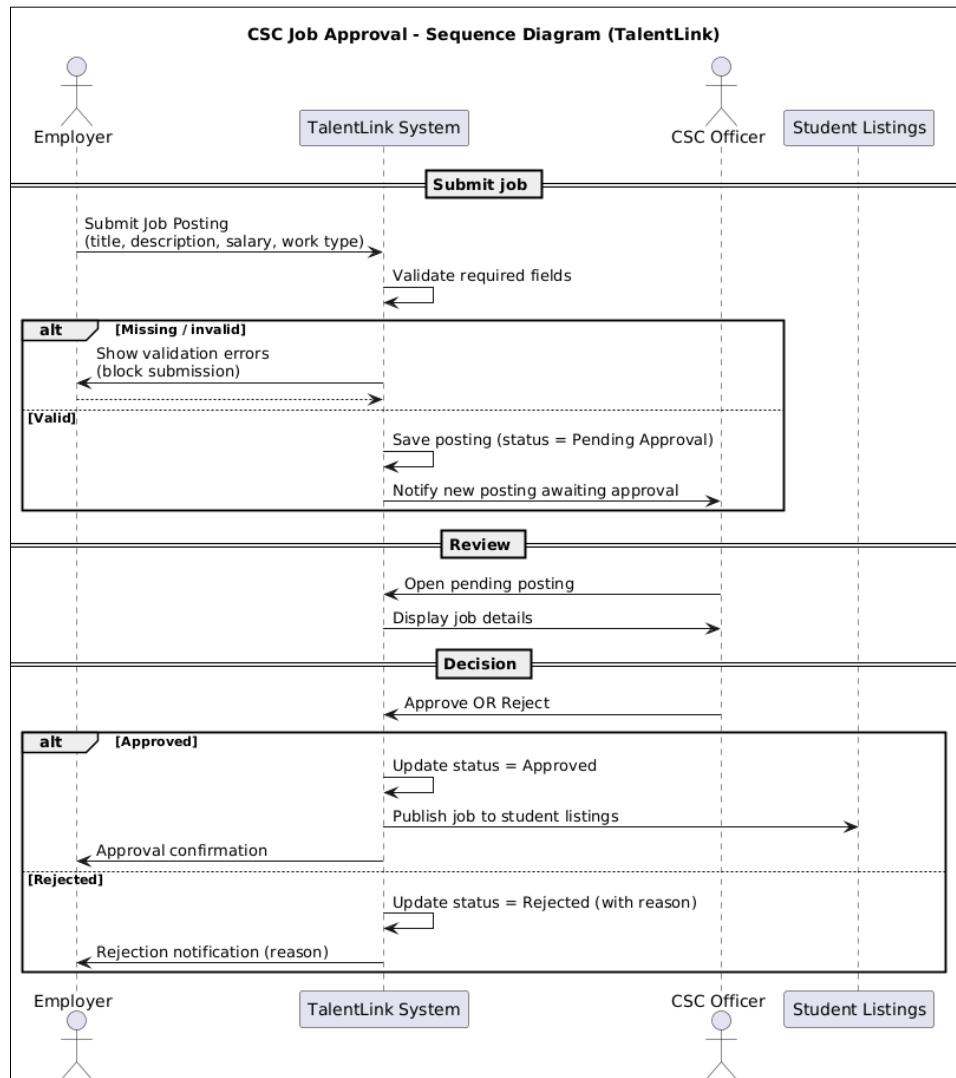


Figure 8.1.6 Sequence Diagram (Approve Job)

This sequence diagram shows the CSC Job Approval process in TalentLink. Employer, TalentLink System, CSC Officer, and Student Listings act as actors. An employer submits a job, then the system will check required fields. If information is missing, validation errors are returned, and the submission is not allowed to proceed. Otherwise, posting is saved with status "Pending Approval" and notify the CSC officer. The CSC officer gets the pending record and sees the job details then makes the decision. On Approve, the system alters status to "Approved", publishes the job to the Student Listings and sends approval confirmation to the employer. On Reject, the system records the reason, changes the status to "Rejected" and notifies the employer accordingly. A straightforward, auditable path—submit → validate → pending → review → approve/reject → publish/notify—is delineated to ensure merely

validated listings are accessible to the students but allow the CSC to retain full control over the job integrity and compliance.

## 9.0 Appendix

**ASIA PACIFIC UNIVERSITY OF TECHNOLOGY AND INNOVATION**  
**CT046-3-2-SDM**  
**Student Coursework Workload Matrix - Grades and Feedback Attachment**

INTAKE: APU2F2506IT, APU2F2506IT(FT), APU252506IT(BIS)		STUDENT NAME	CHAN MIN HUEY	CHEONG PEI ENN	KANG HONG QIAN	NG XIANG LING	WOO MAY ENG	YAP LI SHAN	
System Name:		TP NO.	TP083261	TP081926	TP081205	TP082046	TP082001	TP080968	
<b>A. Group Component</b>									
	ASSIGNMENT COMPONENT	ALLOCATED MARKS	CONTRIBUTION PERCENTAGE	TOTAL %					
Q1	Planning Phase	20	0.00	20.00	40.00	40.00	0.00	0.00	100
Q2	Analysis Phase	10	0.00	50.00	50.00	0.00	0.00	0.00	100
Q3	Design Phase	5	0.00	0.00	0.00	0.00	100.00	0.00	100
Q4	Development Phase	25	25.00	0.00	0.00	0.00	25.00	50.00	100
Q5	Deployment Phase	10	50.00	0.00	0.00	50.00	0.00	0.00	100
	Documentation & Report Presentation	5	16.66	16.66	16.66	16.66	16.66	16.66	100
	Total Marks and Contribution	75	16.66%	16.66%	16.66%	16.66%	16.66%	16.66%	
		Signature	HUEY	ENN	KANG	LING	WOO	SHAN	

## 10.0 References

GeeksforGeeks. (2018, April 5). *Prototyping Model Software Engineering*. GeeksforGeeks.

<https://www.geeksforgeeks.org/software-engineering/software-engineering-prototyping-model/>

GeeksforGeeks. (2018, July 13). *SDLC VModel Software Engineering*. GeeksforGeeks.

<https://www.geeksforgeeks.org/software-engineering/software-engineering-sdlc-v-model/>

GeeksforGeeks. (2018, March 18). *Waterfall Model Software Engineering*. GeeksforGeeks.

<https://www.geeksforgeeks.org/software-engineering/waterfall-model/>

GeeksforGeeks. (2018, October 10). *Integration Testing Software Engineering*.

GeeksforGeeks. <https://www.geeksforgeeks.org/software-testing/software-engineering-integration-testing/>

GeeksforGeeks. (2019, May 30). *System Testing Software Engineering*. GeeksforGeeks.

<https://www.geeksforgeeks.org/software-testing/system-testing/>

GeeksforGeeks. (2022, March 24). *Kanban Agile Methodology*. GeeksforGeeks.

<https://www.geeksforgeeks.org/software-engineering/kanban-agile-methodology/>

GeeksforGeeks. (2023, December 13). *What is Scrum? Understanding the Agile Framework for Project Management*. GeeksforGeeks. <https://www.geeksforgeeks.org/software-engineering/what-is-scrum/>

GeeksforGeeks. (2025, July 11). *Rapid Application Development Model (RAD) Software Engineering*. GeeksforGeeks. <https://www.geeksforgeeks.org/software-engineering/software-engineering-rapid-application-development-model-rad/>

Hartshorne, D. (2020, September 30). *How to incorporate the 12 Agile principles in your development projects*. Monday.com Blog. <https://monday.com/blog/rnd/agile-principles/>

Katalon. (2023, January 31). *What Is Integration Testing And Its Types?* Katalon.com; Katalon. <https://katalon.com/resources-center/blog/integration-testing>

Oppermann, A. (2023, April 6). *What Is the V-Model? (Definition, Examples) | Built In.*

Builtin.com. <https://builtin.com/software-engineering-perspectives/v-model>

ProductPlan. (2021). *What are the 12 Agile Principles? | Definition and Overview.*

Www.productplan.com. <https://www.productplan.com/glossary/agile-principles/>

Schneider, A. (2024, November 25). *What Is Rapid Application Development (RAD)? The Complete Guide for 2025.* Monday.com Blog. <https://monday.com/blog/rnd/rapid-application-development-rad/>