

# UB JOB PORTAL

Report #1



CMPS4131 - Software Engineering

Manual Medina

March 5, 2023

# User Effort Estimation

		Team Member Name				
		Jahmur L	Jaheim L	Josias M	Tadeo B	Daniel M
R e s p o n s i b i l i t y  L e v e l	<b>Project management (10 points)</b>	10%	5%	5%	75%	5%
	<b>Sec.1: Customer Statement of Requirements (9 points)</b>				100%	
	<b>Sec.2: System Requirements (6 points)</b>	45%			10%	45%
	<b>Sec.3: Functional Requirements Specification (30 points)</b>	19%	19%	19%	19%	24%
	- Stakeholders					100%
	- Actors and Goals				100%	
	- Use Cases	100%				
	- Use Case Diagram			100%		
	- Traceability Matrix					100%
	- Fully Dressed Descriptions		100%			
	- Sequence Diagrams				75%	25%
	<b>Sec.4: User Interface Specs(15 points)</b>	20%		37%	43%	
	- Preliminary Design	25%		25%	50%	
	- User Effort Estimation			50%	50%	
	<b>Sec.5: System Architecture (15 points)</b>		16.6%	33.3%	33.3%	16.6%
	- UML Package Diagram				100%	
	- Architecture Styles				100%	
	- Mapping Subsystems to Hardware					100%
	- Connectors and Network Protocols			100%		
	- Global Control Flow			100%		
	- Hardware Requirements		100%			
	<b>Sec.6: Plan of Work (5 points)</b>			100%		

# Table of Contents

<b>CUSTOMER STATEMENT OF REQUIREMENTS</b>	<b>4</b>
Problem Statement	4
Glossary of Terms	7
<b>SYSTEM REQUIREMENTS</b>	<b>9</b>
Functional Requirements	9
Non-Functional Requirements	10
On-Screen Appearance Requirements:	11
<b>FUNCTIONAL REQUIREMENTS SPECIFICATION</b>	<b>12</b>
Stakeholders:	12
Actors and Goals	12
Use cases	13
Use Case Diagram	15
Traceability Matrix	16
Fully-Dressed Description	17
Sequence Diagrams	22
<b>Interface Specification</b>	<b>26</b>
Preliminary Design	26
User Effort Estimation	30
<b>System Architecture</b>	<b>32</b>
UML Package Diagram	32
Architecture Styles	33
Mapping Subsystems to Hardware:	34
Connectors and Network Protocols:	34
Global Control Flow	34
Hardware Requirements:	35
Plan of Work	36
Gantt Chart for the development of the System:	37
<b>References</b>	<b>40</b>

# CUSTOMER STATEMENT OF REQUIREMENTS

## Problem Statement

### **Finding a job is hard. The problems students face**

In Belize, it is common knowledge that finding a job after university, given your experience, is quite hard. To find a job, students must scour social media, newspapers, and company websites to find a vacancy in an organization. In addition, some students rely on asking friends, family, and others to learn about available jobs. This old way of finding a job is time-consuming and unproductive. Given the right tool, the time it takes to find a job vacancy can be used to apply for appropriate jobs. Other popular sites like Indeed or LinkedIn are great for individuals to post their resumes and more information about themselves. Also, these applications have integrated features to allow individuals to learn more about posted job vacancies easily. Unfortunately, many organizations in Belize still rely on older methods to get out their job applications and hire new candidates. Henceforth, a more modern but “Belizean” solution must be implemented to encourage organizations to use these new technologies further to enhance their search for the best student candidates. Furthermore, it helps students find jobs that best suit them.

### **Keeping track of applications (student perspective)**

There is no guarantee of success in one job application in the current job market. Therefore, it is in their best interest that a student applies for as many jobs as possible. Remembering and revisiting the numerous applications and tracking their status can be unmanageable. Such tracking can be done using Excel spreadsheets. The drawback comes with the limiting factor of manually updating new rows of information, which could get numerous and frustrating. An integrated tool can make this feature seamless and more intuitive.

### **Keeping track of applications (employer perspective)**

Places companies post vacancies include their websites, newspapers, or social media such as Facebook. In today's market, youth unemployment is 21.81% of the workforce aged 15 - 24 in 2022 (*Belize Youth Unemployment Rate 1991-2024*, n.d.). Therefore, we can deduce that a job application on social media can have numerous applicants. Usually, these applicants would have to submit their resumes through email, as this is the norm for professional communication in Belize. The employer must then examine the resumes and CVs provided through email. A short list is created with the individuals to interview. The employer is then obliged to respond to all the other applicants about the status of their resumes. In some cases, applicants are not told the status of their application and take the initiative to reach out, requesting an update. This whole process can become efficient for both applicants and employers by using a centralized job vacancy application. In such an application, employers can manage all job vacancy posts with related applicant information shortlists and issue application updates to candidates not considered. This speeds up the process of employers finding a suitable candidate and does not leave other candidates guessing if their resume was considered.

### **Companies may enjoy the organization of speaking with multiple candidates for a posted job**

Another advantage of a centralized system is that a company can organize conversations between numerous shortlisted candidates about a particular vacancy. This streamlines the process of communication between employers and potential candidates without the stress of email conversations. As mentioned above, candidates not considered for the job can be easily updated through an employer's click of a button, removing the time to update each candidate individually.

### **Companies can reach more people and different kinds of people much faster**

Companies sometimes need to find a suitable candidate quickly to fill an important role. Not all companies have the reach or the following to discover the best candidates. With the advent of a new job portal application for UB students, companies can leverage this tool to get their job

vacancies directly in the eyes and interests of hundreds of students. Job posts can instantly reach candidates with different fields of expertise, degrees, and experience. This, in turn, minimizes the time an employer spends searching for a candidate. This also increases the chances of finding the perfect candidate for the role. The system puts the employer and candidate in direct contact with each other. An employer can converse with potential candidates, only taking the conversation to other applications if a video call is required.

### **Sending and receiving documents become faster and easier**

Handling numerous documents and conversations through Gmail (the most common communication method in this process) can become messy, especially when handling numerous attachments spread across multiple messages in an email thread. Henceforth, an organized way to view all sent and received documents between parties can make processes for employers more intuitive. Such a feature is incredibly useful and helpful for employers, especially in limited time.

### **Advantages for UB students**

As a student and alumni of the University of Belize, these students automatically have the advantage of reach when it comes to getting their skills, experiences, and resumes out to potential employers. This may improve the likelihood of UB students getting hired.

In conclusion, an application built to address these problems can ultimately impact Belize's youth unemployment rate. It allows employers to reach UB candidates faster and allows students to find and apply for multiple jobs easily.

## Glossary of Terms

Term	Definition
Employer	The entity that makes job vacancy posts and considers candidates for hire.
Applicant	A UB student who applies for posts seeking an interview for hire.
Application	Submission of one's resume or other personal details to be considered for an interview or job.
Email Thread	A group of emails in an email conversation between two Gmail users.
Administrator	The user that had admin rights to verify job listings, verifying legitimacy of vacancies and users registered.
Analytics	The website will be able to
Client-Server Model	A computing architecture where tasks or services are divided between service providers (servers) and service requesters (clients). Clients initiate requests for services, while servers respond to those requests by providing the requested resources or performing the requested tasks, enabling distributed computing over a network.
Relational Database	A relational database is a type of database that organizes data into tables of rows and columns, with relationships defined between the tables, enabling efficient storage, retrieval, and manipulation of structured data.
REST API	REST API is an architectural style for an API that uses HTTP requests to access and use data, and exchange it securely over the internet. REST API is a way for two computer systems to communicate.
RESTful API	RESTful API is an interface that allows two different systems to exchange information over the internet with tight security. RESTful APIs is the web-service implementation of the

	REST architectural style and offers a scalable and simple method to construct APIs that are applicable to various programming languages and platforms.
CRUD	The CRUD (Create, Read, Update, Delete) paradigm is popular in web application development because it provides memorable foundation for reminding developers how to create com



# SYSTEM REQUIREMENTS

## Functional Requirements

Priority Weight	Description
1	Not important
2	Low importance
3	Normal
4	Important
5	Very Important

Identifier	PW	Requirement
REQ-1	2	The system shall allow applicants and employers to register an account to access the UB Job Portal.
REQ-2	3	The system shall provide applicants with CRUD (Create, Read, Update, Delete) functionality to manage their accounts, including the ability to upload personal information and resume documents, as well as edit their profiles.
REQ-3	5	The system shall allow employers to create vacant job posts.
REQ-4	3	The system shall allow employers to perform CRUD (Create, Read, Update, Delete) operations to manage their accounts, including the ability to upload company information and edit their profiles, including company information and job posts.
REQ-5	5	The system shall allow applicants to search for job posts based on criteria.
REQ-6	2	The system shall allow applicants to filter job post results based on preferences such as full-time/part-time, internship, or remote positions.
REQ-7	2	The system shall allow applicants to apply for job posts.
REQ-8	1	The system shall provide employers access to view applicant information and documents.
REQ-9	2	The system shall facilitate communication between employer and applicants, allowing for instant messaging in regard to a job post.

REQ-10	3	The system shall notify applicants about new and related job posts and their application status as well as notify employers about related prospective applicants.
REQ-11	3	The system shall provide reporting and analytics capabilities to track usage metrics, such as the number of job postings, applications, and user interactions.

## Non-Functional Requirements

### **Functionality**

NONREQ-1: The system should use a method to move user data to the database and back to the user.

NONREQ-2: The system will be responsible for saving data from electronic forms into a database.

NONREQ-3: The platform shall provide companies with the ability to showcase and upload job vacancies.

NONREQ-4: Students can browse, search, and view available job opportunities.

NONREQ-5: Companies and applicants should be able to register, creating individual profiles.

### **Usability**

NONREQ-6: The system should be a clear, user-friendly, and understandable web app.

NONREQ-7: The platform should have an intuitive and user-friendly interface for both companies and applicants.

### **Reliability**

NONREQ-8: The platform should have a high level of reliability, minimizing downtime or service interruptions.

NONREQ-9: Data integrity measures are in place to ensure accuracy, and regular data backups prevent the potential loss, contributing to a reliable platform.

NONREQ-10: High system availability is maintained through robust hosting solutions, redundancy for critical components, and security measures to protect against vulnerabilities and loss of user information.

### **Performance**

NONREQ-11: The platform is optimized for low-latency response times, undergoing regular performance testing to identify and address any bottlenecks.

NONREQ-12: Scalability measures and load balancing techniques are implemented to handle increased user traffic efficiently and maintain optimal performance.

Supportability:

NONREQ-13: Provide documentation and training materials for administrators, companies, and applicants.

NONREQ-14: Offer customer support channels to address user queries and issues promptly.

NONREQ-15: Comprehensive documentation guides administrators, companies, and applicants, with additional troubleshooting guides and FAQs for user support.

NONREQ-16: Training sessions and ongoing support channels, such as email or chat, are provided, and a systematic approach to software updates ensures continuous improvement while minimizing disruptions.

### **Security**

NONREQ-17: Data transmitted and stored should be encrypted to ensure the confidentiality and integrity of user information.

### **On-Screen Appearance Requirements:**

<b>Identifier</b>	<b>PW</b>	<b>Requirement</b>
<b>ONSREQ-1</b>	4	Design an intuitive and responsive layout that adapts to different screen sizes and resolutions. Provide clear navigation paths, ensuring users can easily find and access relevant information.
<b>ONSREQ-2</b>	3	Define a set of consistent icons and symbols for actions, alerts, and other visual cues to enhance user understanding.
<b>ONSREQ-3</b>	3	The platform's visual design should align with the school's brand guidelines, ensuring a consistent and cohesive appearance.

# FUNCTIONAL REQUIREMENTS SPECIFICATION

## Stakeholders:

### 1. UB Students (Internal Users)

- Easily access the UB Job Portal.
- Share personal info for job applications.
- Search for jobs that match their skills.
- Communicate smoothly with potential employers.

### 2. Employers/Companies (External Users)

- Post job openings on the UB Job Portal.
- Check out applicant profiles.
- Communicate with applicants about job opportunities.
- Manage and review job applications.

### 3. Administrators

- Verify and manage user registrations.
- Ensure posted jobs meet eligibility criteria.
- Provide support to both applicants and employers.

## Actors and Goals

Actor	Type	Goals
Student	Initiating	Create an account Upload, delete, and update personal information and documents Apply for jobs Message employers about a job if considered
Employer	Initiating	Post job applications Upload, delete, and update company information Accept potential candidates Message candidates about a job and communicate further steps in the application process.
Administrator	Initiating	See system usage information

## Use cases

Name	Description	Requirement Covered
UC 1 - Registration	Allows the system to create a profile of a user using their personal or business information.	REQ-1 REQ-2 REQ-4 NONREQ-1 NONREQ-2 NONREQ-5 NONREQ-6 NONREQ-7 ONSREQ-3
UC 2 - Authentication	Ensures that ONLY UB students and legitimate businesses are allowed to register and login.	REQ-1 REQ-4 NONREQ-5 NONREQ-17 ONSREQ-3
UC 3 - FilterJob	To allow the system to decipher which listings are legitimate.	REQ-3 ONSREQ-1
UC 4 - ManageProfile	Allows system users to upload, edit, and delete their profile information.	REQ-2 NONREQ-1 NONREQ-2
UC 5 - PostJob	To allow employers to vacant job posts for applicants.	REQ-3 REQ-5 REQ-10 NONREQ-1 NONREQ-2 NONREQ-3 NONREQ-6 NONREQ-7 ONSREQ-3
UC 6 - SearchJobListing	To allow applicants to search through different job listings available.	REQ-5 REQ-6 NONREQ-4 NONREQ-7 ONSREQ-1
UC 7 - ApplyForJob	To allow applicants to apply for jobs by submitting their resume.	REQ-7 NONREQ-1 NONREQ-6 NONREQ-7 ONSREQ-1

		ONSREQ-2
UC 8 - Message	To allow both applicants and employers to communicate with each other via instant messaging.	REQ-9 NONREQ-1 ONSREQ-2
UC 9 - ViewStudentListing	Allows employers to view applicants personal information that they made public, as well their resumes.	REQ-8 NONREQ-7 ONSREQ-2
UC 10 - AnalyzeStatistics	To allow the administrator to view, provide reporting & analytics of usage statistics.	REQ-11 NONREQ-2

### 1. Student Registration and Profile Creation

- Description: Students sign up on the UB Job Portal and create profiles by sharing details about themselves, education, and skills.
- Actors: UB Student, System

### 2. Job Vacancy Posting by Employers

- Description: Employers upload job openings to the UB Job Portal, providing details like job descriptions, qualifications, and deadlines.
- Actors: Employer, System

### 3. Job Search and Application by Students

- Description: Students look for jobs, view details, and submit applications based on their preferences and qualifications.
- Actors: UB Student, System

### 4. Communication between Employers and Students

- Description: Employers and applicants chat through the platform regarding job applications, interview schedules, and application status.
- Actors: UB Student, Employer, System

### 5. Job Application Tracking by Students

- Description: Students track their job applications, receive updates, and efficiently manage multiple applications.
- Actors: UB Student, System

### 6. Job Application Review and Shortlisting by Employers

- Description: Employers review applications, shortlist candidates, and manage the recruitment process within the platform.
- Actors: Employer, System

# Use Case Diagram

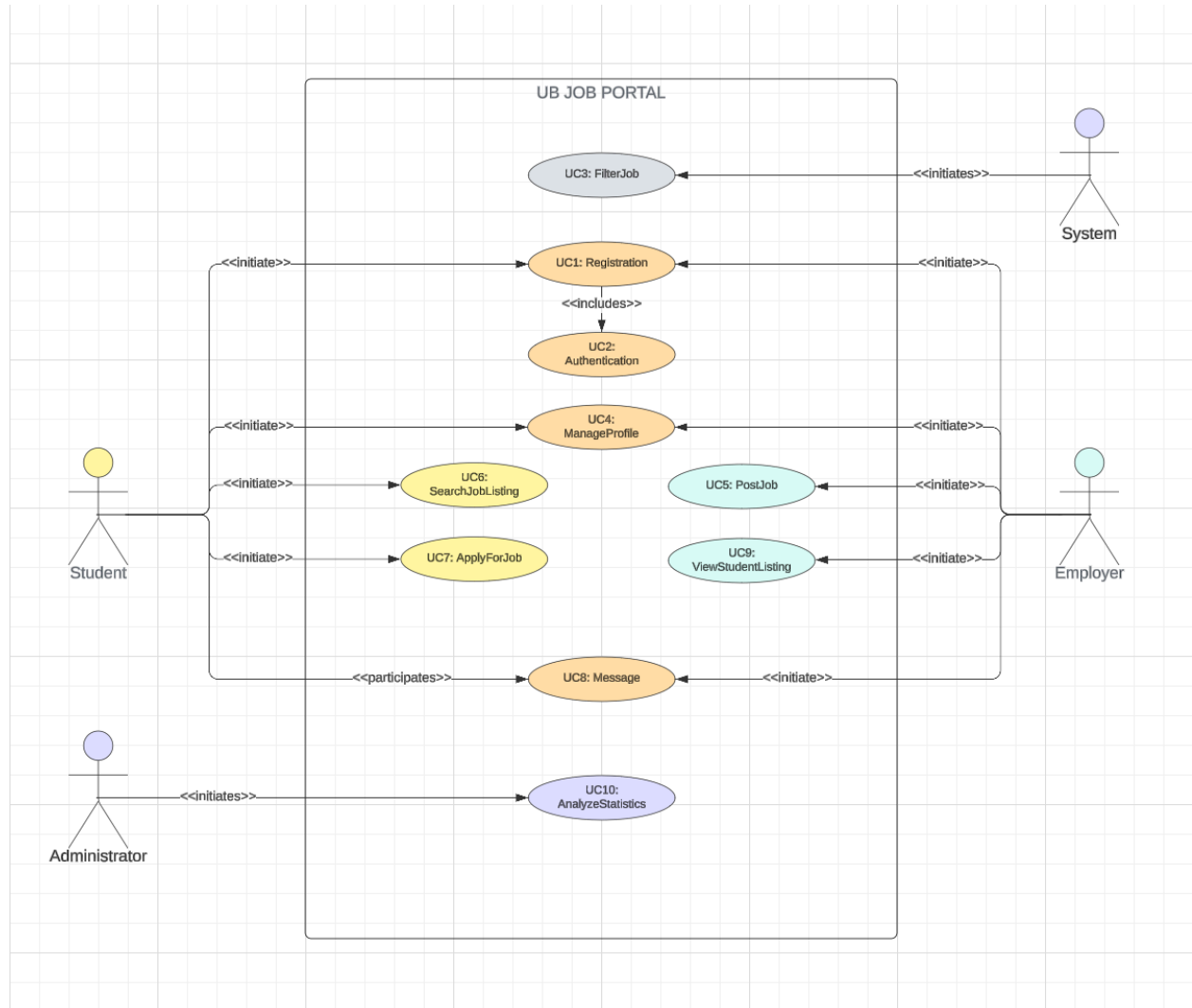


Figure 1 - Use Case Diagram of UB Job Portal

## Traceability Matrix

PW Value		UC 1 - StudentRegistration	UC 2 - StudentAuthentication	UC 3 - EmployerRegistration	UC 4 - EmployerAuthentication	UC 5 - Filter Job	UC 6 - Insert-Info	UC 7 - ChangeStudentInfo	UC 8 - UploadCompanyInfo	UC 9 - ChangeCompanyInfo	UC 10 - CreateJob	UC 11 - SearchJobListing	UC 12 - ApplyJob	UC 13 - Message	UC 14 - ViewDocuments	UC 15 - ViewPersonalInfo	UC 16 - Verify	UC 17 - AnalyzeStatistics
5	REQ-1	X		X														
2	REQ-2	X	X	X	X													
1	REQ-3						X											
2	REQ-4							X										
3	REQ-5																	
5	REQ-6								X		X							
1	REQ-7									X								
5	REQ-8					X						X						
2	REQ-9					X						X						
1	REQ-11												X					
1	REQ-12														X	X		
2	REQ-13													X				
3	REQ-14													X				
1	REQ-15		X		X												X	
3	REQ-16																	X
2	NOREQ-1	X		X		X	X	X	X	X								
4	NOREQ-2							X	X	X								
5	NOREQ-3																	
2	NOREQ-4																	
3	NOREQ-5																	
5	NOREQ-6	X		X														X
2	NOREQ-7	X		X		X	X	X	X	X								
2	NOREQ-8																	
4	ONSREQ-1	X		X			X	X	X	X		X						
3	ONSREQ-2	X		X														
3	ONSREQ-3	X		X														
Total Weight		26	3	26	3	11	9	14	17	13	5	11	2	5	1	1	1	8

Table 1. Traceability Matrix according to system requirements and use cases



## Fully-Dressed Description

<b>UC-1</b>	Registration
<b>Related Requirements</b>	REQ-1, REQ-2, NONREQ-1, NONREQ-6, NONREQ-7
<b>Initiating Actor</b>	Student
<b>Actor's Goal</b>	To allow the system to capture all fields entered when applicants register
<b>Participating Actors</b>	Administrator
<b>Pre-Condition</b>	
<b>Post-Condition</b>	
<b>Flow of Events for Main Success Scenario</b>	<p>→ 1. The applicant accesses the registration site via the web portal or application</p> <p>→ 2. They go to the sign up section of the portal to create a new account</p> <p>→ 3. The applicant inputs all necessary information on the signup page (eg. name, phone number, etc.)</p> <p>→ 4. Once done, the applicant will click the register button.</p> <p>← 5. The system will ask the applicant to confirm the registration. The applicant confirms.</p> <p>← 6. The system updates itself with the new information added by the applicant</p>

<b>UC-5</b>	PostJob
<b>Related Requirements</b>	REQ-3, REQ-5,REQ-10, NONREQ-3
<b>Initiating Actor</b>	Employer
<b>Actor's Goal</b>	Create a post with details of an available job for an applicant to view.
<b>Participating Actors</b>	
<b>Pre-Condition</b>	The employer is registered and have logged in.
<b>Post-Condition</b>	The job post is added to a list of other posts called a job listing. Applicants can then view this job listing.
<b>Flow of Events for Main Success Scenario</b>	→ 1. The employer clicks a button to add a new job post. ← 2. The system displays a form to get the details of the new post. → 3. (a) The employer enters the job details. (b) The employer submits the form. ← 4. (a) The system adds the new post details into the database. (b) The system displays that new post in the applicants' feeds.

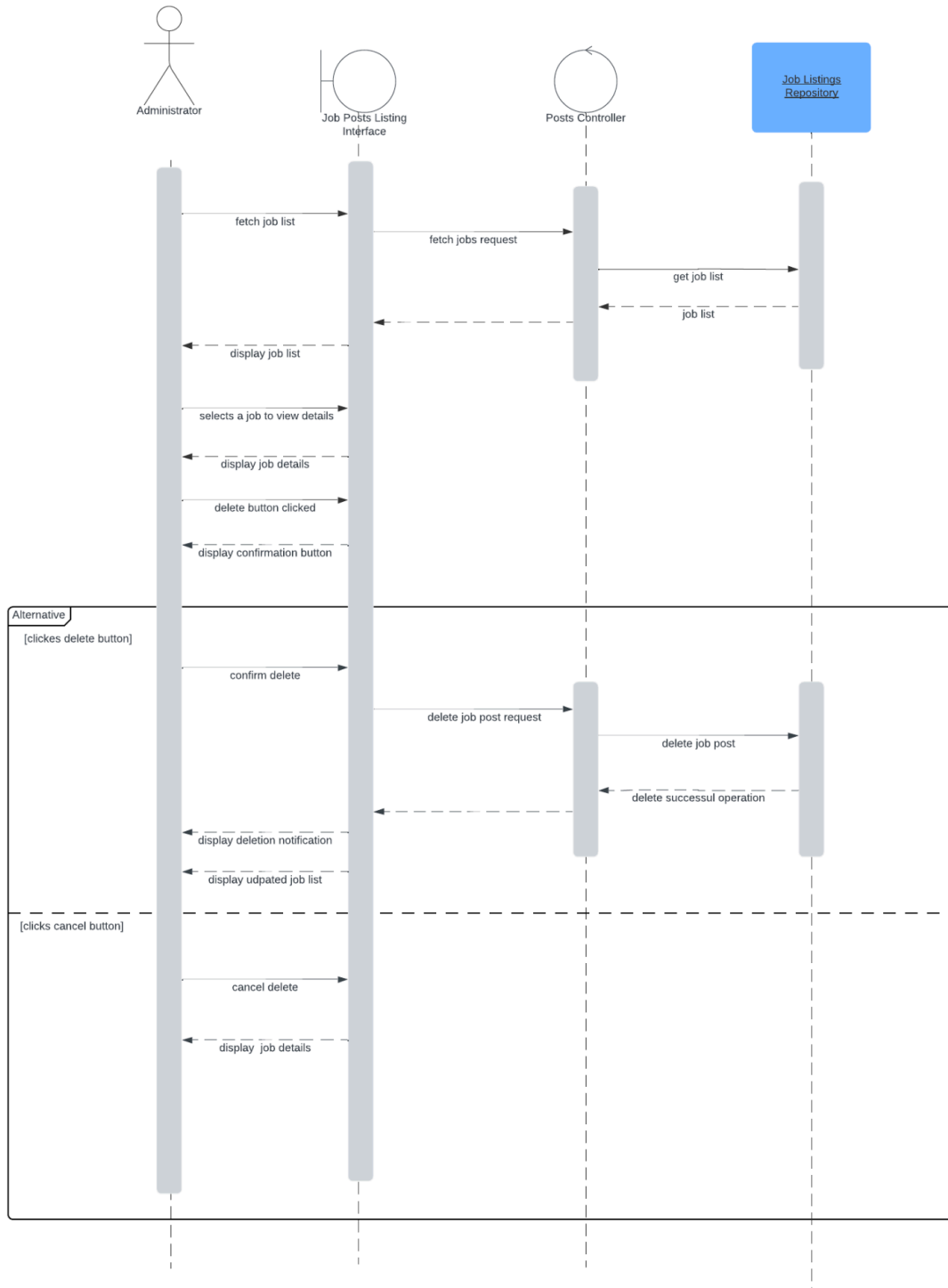
<b>UC-2</b>	Authentication
<b>Related Requirements</b>	REQ-1, REQ-4, NONREQ-5, NONREQ-17
<b>Initiating Actor</b>	System
<b>Actor's Goal</b>	Ensure that applicants are UB students and employers are verified and trusted.
<b>Participating Actors</b>	Administrator
<b>Pre-Condition</b>	The administrator is logged in and can view a list of all users.
<b>Post-Condition</b>	The administrator can disable the account of an employer or applicant.
<b>Flow of Events for Main Success Scenario</b>	<p>Admin checks an employer's account</p> <p>→ 1. The admin selects a button to view all employers.</p> <p>← 2. The system returns the list of employers</p> <p>→ 3. The admin selects to view the employer's details.</p> <p>← 4. The system displays the employer's details</p> <p>→ 5. (a) The admin selects to disable/suspend the employer's account. (b) The admin selects and submits the reason for the suspension.</p> <p>← 6. (a) The system updates the employer's account in the database, not allowing logins for that account. (b) The system displays to the admin that the account is disabled.</p>
<b>Flow of Events for Extensions(Alternate Scenarios)</b>	<p>Admin checks an applicant's account</p> <p>→ 1. The admin selects a button to view all applicants.</p> <p>← 2. The system returns the list of applicants</p> <p>→ 3. The admin selects to view the applicant's details.</p> <p>← 4. The system displays the applicant's details</p> <p>→ 5. (a) The admin selects to disable/suspend the applicant's account. (b) The admin selects and submits the reason for the suspension.</p> <p>← 6. (a) The system updates the applicant's account in the database, not allowing logins for that account. (b) The system displays to the admin that the account is disabled.</p>

<b>UC-7</b>	ApplyForJob
<b>Related Requirements</b>	REQ-7, NONREQ-1, NONREQ-6, NONREQ-7, ONSREQ-1, ONSREQ-2,
<b>Initiating Actor</b>	Applicant
<b>Actor's Goal</b>	To forward an applicant's information to an employer regarding a job post.
<b>Participating Actors</b>	
<b>Pre-Condition</b>	The applicant is logged in and shown a list of posted jobs. The applicant has already uploaded their resumes to their profile.
<b>Post-Condition</b>	The employer is notified of the application and receives the applicant's information.
<b>Flow of Events for Main Success Scenario</b>	→ 1. The applicant selects a button to apply for the job. ← 2. The system prompts to confirm the option. → 3. The applicant confirms the action. ← 4. (a)The system retrieves the applicants' information from a database, along with their resume documents. (b) the system creates this user's application. (c) The system sends the application to the employer.

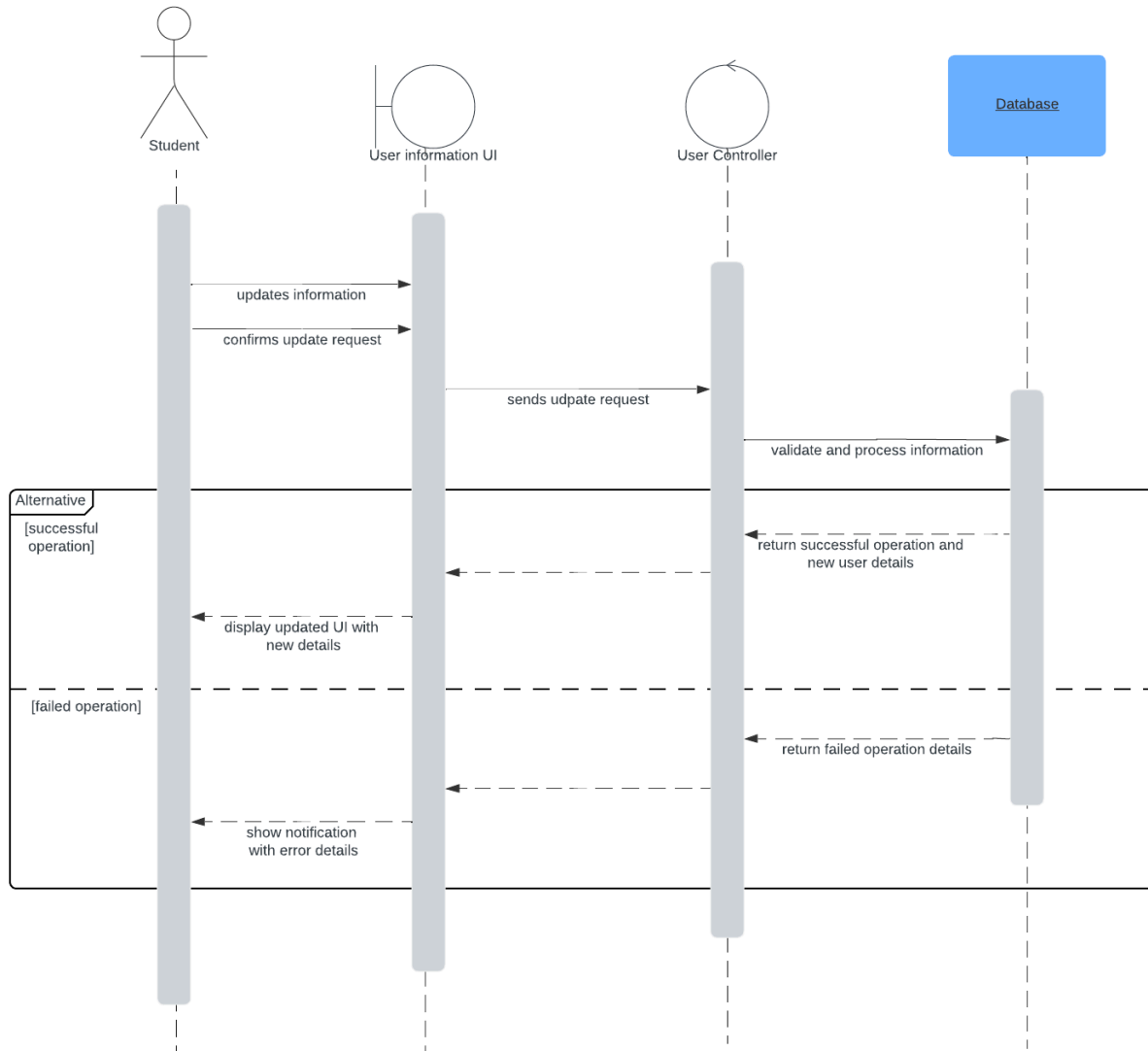
<b>UC-6</b>	SearchJobListing
<b>Related Requirements</b>	REQ-5, REQ-6, NONREQ-4, NONREQ-7, ONSREQ-1
<b>Initiating Actor</b>	Applicant
<b>Actor's Goal</b>	To search and find different jobs based on different criterias.
<b>Participating Actors</b>	
<b>Pre-Condition</b>	The applicant is logged in, and there are multiple job posts.
<b>Post-Condition</b>	A list of job posts that fit the criteria is displayed the applicant applies for those job posts.
<b>Flow of Events for Main Success Scenario</b>	→ 1. The applicant selects a button to see the search criteria options. ← 2. The system displays the options. → 3. The user selects one or multiple filter/search options. ← 4. (a) The system generates a query from the search options. (b) The system queries the database for related job posts.(c) The system displays the job posts to the applicants.

# Sequence Diagrams

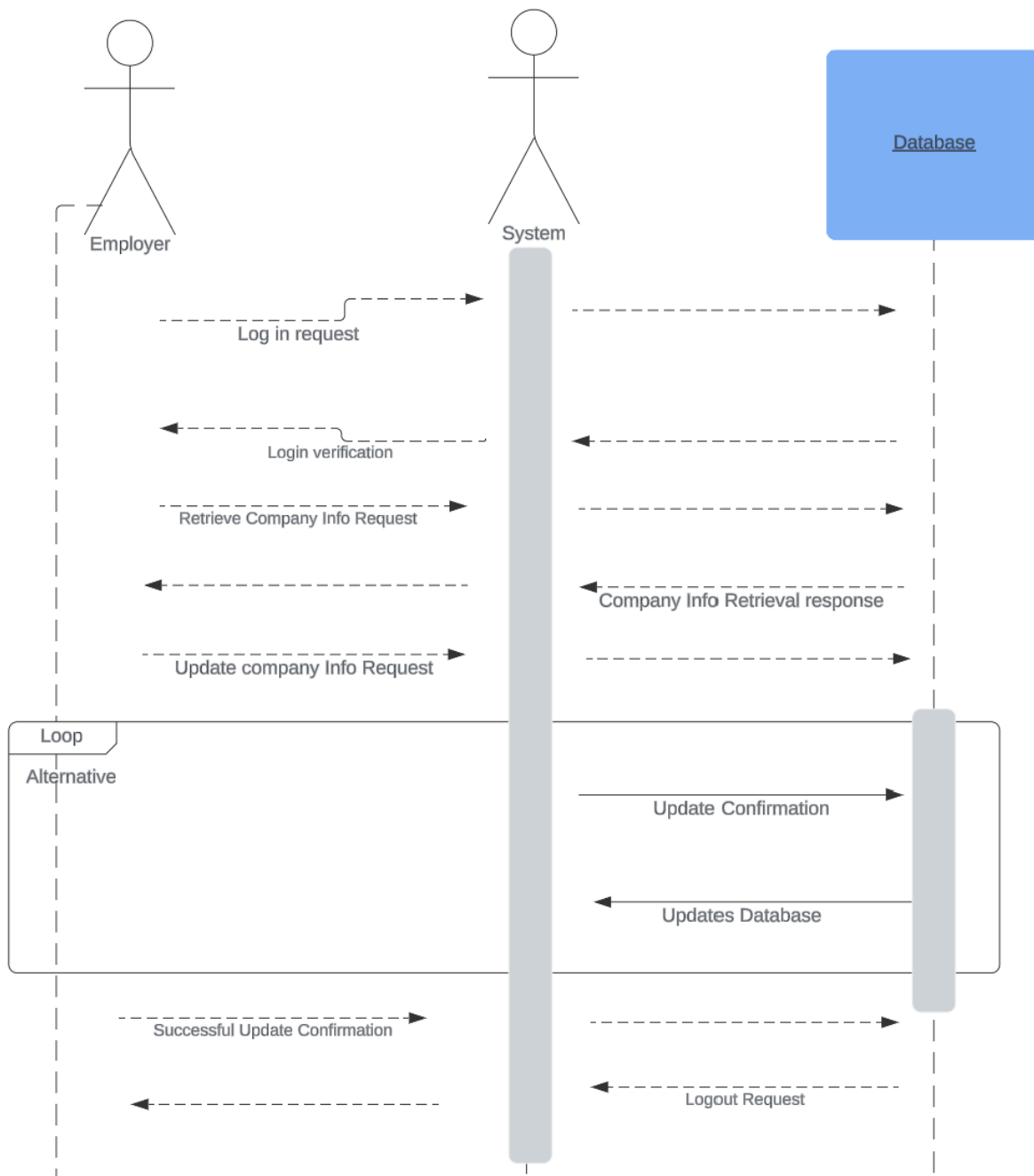
## UC5 - Filter Job



## UC8 Upload Company Information



## UC9 ChangeCompanyInfo





# Interface Specification

## Preliminary Design

### UC 1 - StudentRegistration

To create an account to apply for jobs, an applicant inputs their first name, last name, email address, phone number, and password. With this account, applicants can be able to then log in using their email and password. By creating an account, applicants have access to job listings and have access to apply for jobs.

University of Botswana  
**UB**  
Job Portal

Find Jobs, Apply, Get Hired! Join Now!

First name  Last name

Email Address

Phone number  Field of study

Password

Confirm Password

[Continue](#)

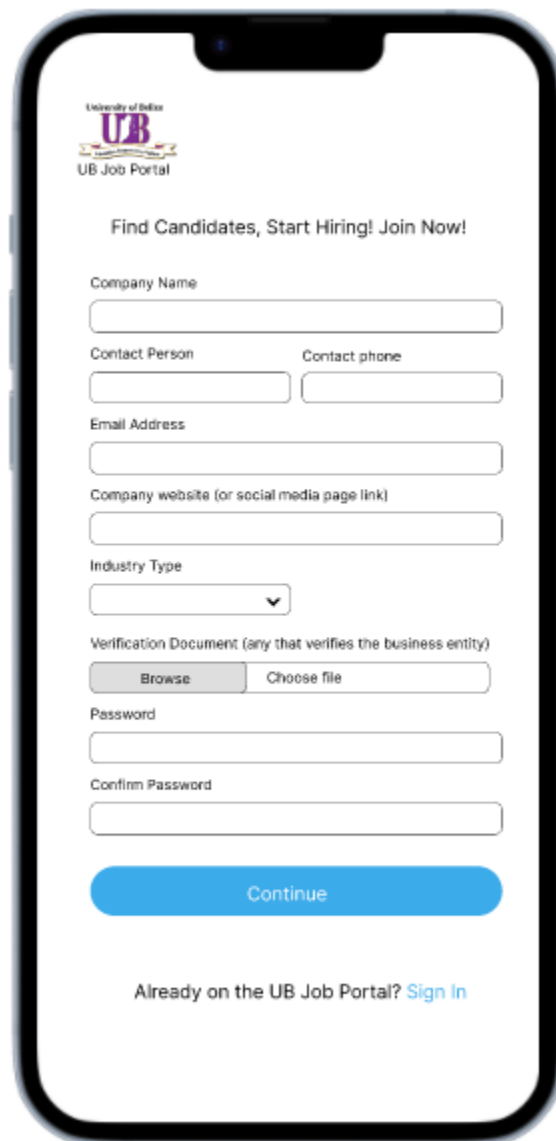
Already on the UB Job Portal? [Sign In](#)

Want to be a registered employer ? [Create an account](#)

Figure 2 - Student registration screen

### UC 3 - EmployerRegistration

To create an account to post jobs, their identity as an official business needs to be confirmed by an administrator. An employer inputs the following information to the sign-up form: company name, a contact person's name, and phone number, email address, company/social media website link, industry type, and any file that helps to verify their business entity. After approval, an employer can log in using their email and password. With this account, employers can create job posts and handle candidates.



The image shows a smartphone screen displaying the 'UB Job Portal' registration form. At the top, the 'University of Botswana' logo and 'UB Job Portal' text are visible. Below this is a heading 'Find Candidates, Start Hiring! Join Now!'. The form contains several input fields: 'Company Name', 'Contact Person', 'Contact phone', 'Email Address', 'Company website (or social media page link)', 'Industry Type' (a dropdown menu), 'Verification Document (any that verifies the business entity)' (with 'Browse' and 'Choose file' buttons), 'Password', and 'Confirm Password'. A blue 'Continue' button is at the bottom of the form. Below the button, there is a link: 'Already on the UB Job Portal? [Sign In](#)'.

Figure 3 - Employer registration screen

## UC 5 - FilterJob

In order for the administrator to verify the credibility of job listings he/she will cycle through each job vacancy that was posted by companies and the administrator has the option to select the specific one that didn't meet the criteria and it can be deleted with the tap of the “delete button”, administrator will be prompted with a warning message indicating that it won't be recovered after deletion..

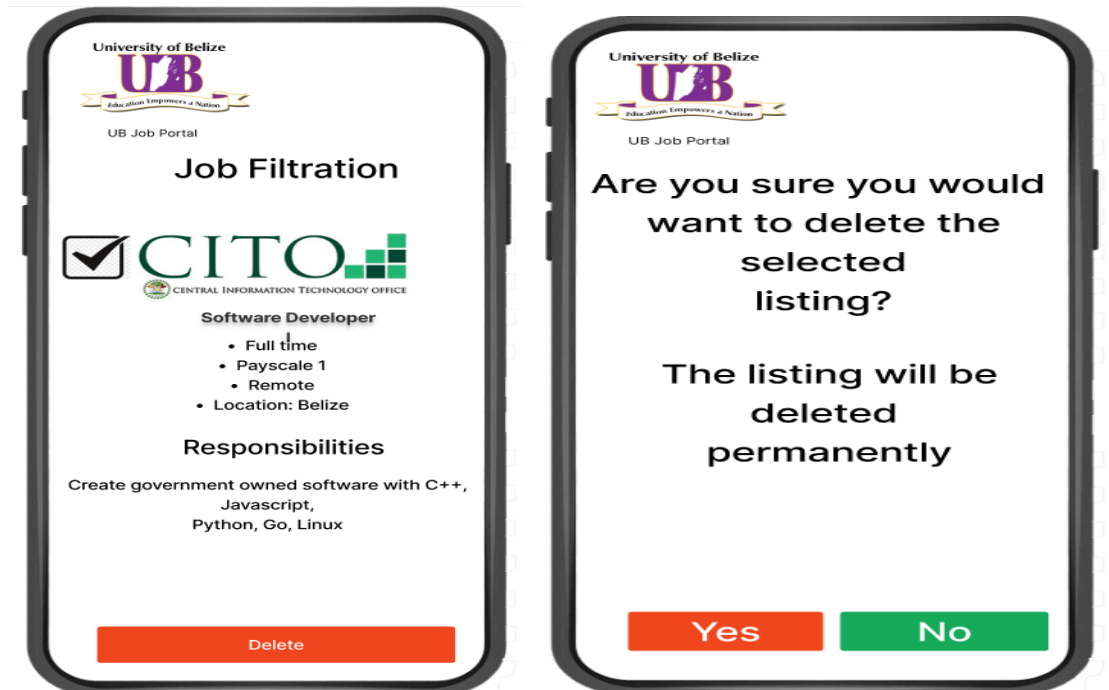


Figure 4 - Job Filtration Screen

## UC9 ChangeCompanyInfo

To update the information so employers can efficiently manage and edit job postings with the latest company information for potential candidates.

The image shows a mobile application interface for the University of Belize Job Portal. At the top, the university's logo and name are displayed. Below this, the title 'Update Company Information' is centered. The form consists of several input fields, each with a label above it: 'Company Name' (with placeholder 'Current Company Name'), 'Contact Person' (with placeholder 'Current Contact Person'), 'Contact Phone' (with placeholder 'Current Contact Phone'), 'Email Address' (with placeholder 'Current Email Address'), 'Company website (or social media page link)' (with placeholder 'Current Company Website'), and 'Industry Type' (with placeholder 'Current Industry Type'). Below these fields is a section for 'Verification Document (any that verified the business entity)', which includes a file input field showing 'CurrentFile.pdf' and a blue 'Edit File' link. At the bottom of the form are two buttons: a blue 'Save Changes' button and a red 'Cancel' button.

University of Belize  
**UB**  
The Belizean University of Research  
UB Job Portal

Update Company Information

Company Name  
Current Company Name

Contact Person  
Current Contact Person

Contact Phone  
Current Contact Phone

Email Address  
Current Email Address

Company website (or social media page link)  
Current Company Website

Industry Type  
Current Industry Type

Verification Document (any that verified the business entity)  
CurrentFile.pdf [Edit File](#)

Save Changes Cancel

Figure 5 - Update Company Information Screen

## User Effort Estimation

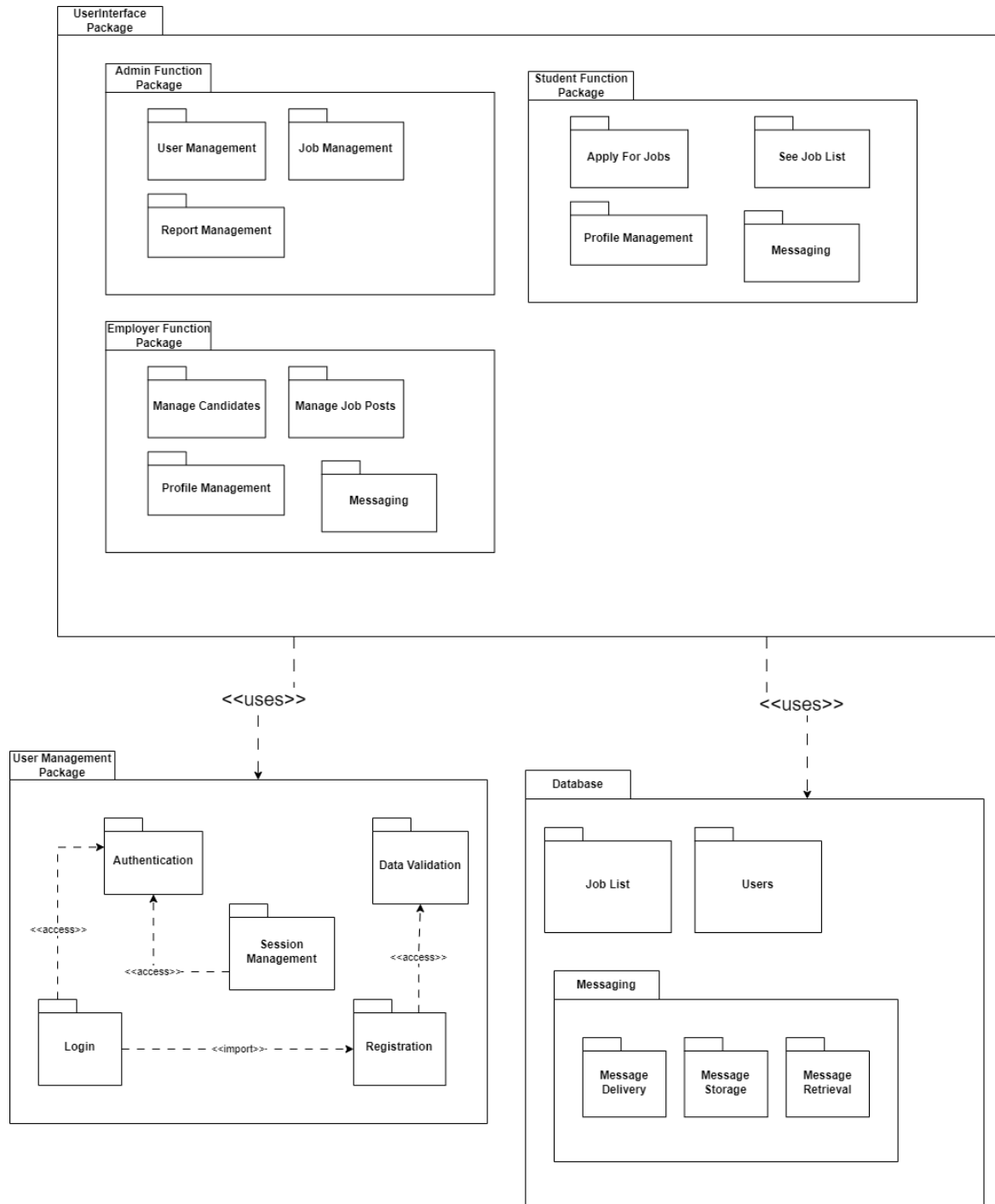
Scenario	Navigation	Data Entry
Student Registration	Click on first name field Click on last name field Click on email address field Click on phone number field Click on field of study dropdown Click field of study from drop-down list Click on password field Click on confirm password field Click continue button	Input first name Input last name Input email address Input phone number Input password Reinput password
Employer Registration	Click on company name field Click on contact person field Click on contact phone field Click on email address field Click on company website field Click on industry type dropdown Click industry type from the drop-down list Click on file input for verification document Click file input selection and confirm button Click on password field Click on confirm password field Click continue button	Input company name Input contact person Input contact phone Input email address Input company website Input password Reinput password
Update Company Info	Click on company name field Click on contact person field Click on contact phone field Click on email address field Click on company website field Click on industry type dropdown Click on file input for verification document Click on Save Changes button	Edit company name Edit contact person Edit contact phone Edit email address Edit company website Edit Industry type Edit file for verification document
Filter Job	Click on invalid job vacancy Click on the delete button	None

	Prompt window appears Click on yes to proceed with deletion Click on no to keep the job vacancy	
--	---	--

# System Architecture

## Identifying Subsystems

## UML Package Diagram



## Architecture Styles

The architecture of the job portal system follows a client-server model, with a web-based user interface for employers, applicants, and administrators interacting with a backend server for data processing and storage. Therefore, employers and applicants can access the service from any device with an internet connection and a web browser. Administrators may view reports and usage statistics on any device, but a desktop computer is most suitable. The UB job portal must handle personal user information over the internet, so reliable protocols such as HTTPS is mandatory.

### **Database**

The system will utilize MySQL as the main relational database technology for all its storage, information processing, and retrieval needs.

### **Data Communication**

The system will utilize a RESTful API so that any authorized requesting system can access and manipulate web resources on the UB Job Portal. All requests are performed using the internet's HyperText Transfer Protocol, and are restricted to GET, PUT, POST, and DELETE requests.

### **Object Oriented**

The system will use object-oriented architecture and design principles to take advantage of modularity, reusability, improved maintainability, and enhanced collaboration.

### **Event Driven**

Our system will use event-driven architecture as users perform actions such as applying for jobs, searching, filtering, and handling personal information. The system must log these events as usage reports are fundamental to the administrator and developers. These events also need to be processed promptly to ensure a good user experience.

### **Asynchronous messaging**

**Real-Time Communication:** Asynchronous messaging can enable real-time communication between users, such as employers and job seekers, allowing immediate responses and quicker interactions during the hiring process.

**Notification System:** Asynchronous messaging can be used to implement a notification system that informs users about important updates, such as new job postings, application status changes, or messages from employers/recruiters.



## Mapping Subsystems to Hardware:

The system primarily operates as a web application, with the client-side subsystem running on users' devices such as laptops, desktops, or mobile phones, utilizing standard web browsers. The server-side subsystem, including the database management system, runs on a centralized server. This server can be hosted either on-premises or on a cloud platform such as Amazon Web Services (AWS) or Microsoft Azure. Communication between the client and server subsystems occurs over the internet, utilizing standard network protocols such as HTTP or HTTPS. Additionally, data transfer between the server and the database requires a stable network connection with sufficient bandwidth to handle concurrent user requests efficiently.

## Connectors and Network Protocols:

To enable companies to hire University of Belize students via a web portal, use HTTPS for secure communication, OAuth 2.0 for user authentication, and ODBC/JDBC for database connectivity. Employ WebSocket for real-time updates, SMTP for emails, and RESTful APIs or GraphQL for integration. Implement TLS/SSL for data security, strong measures against unauthorized access, and consider MQTT for instant messaging.

## Global Control Flow

### Execution Orderliness:

The system adheres to a systematic, procedure-driven approach, ensuring a uniform user experience through a predefined sequence of steps.

It maintains a structured process, wherein each user undergoes the same linear progression when engaging with employers, submitting resumes, and participating in chats.

### Time Dependency:

Notably, the system abstains from the use of timers, devoid of temporal constraints.

The system operates as a non-real-time entity, functioning within an event-response framework, wherein user actions prompt responses without adherence to real-time intervals.

### Event-Response Type:

The system is architected as an event-response model, characterized by its responsiveness to user-initiated actions absent explicit real-time considerations.

It abstains from temporal periodicity, allowing for the natural unfolding of events as users engage in interactions, devoid of predefined time constraints.

## Hardware Requirements:

The hardware requirements will likely depend on many factors, such as the scale of the portal, the number of users it needs to support, and different features that it offers. These are some of the basic hardware requirements

**Storage** → A sufficient amount of storage to store all the data in the portal. A SSD would be a faster choice.

**RAM (memory)** → Minimum of at least 2GB of RAM just for the smooth operation of request.

**Processor (CPU)** → A good CPU (doesn't have to be the latest version) to handle the load of the portal

# Plan of Work

## 1. Requirements Gathering:

- Collaborate with stakeholders to understand their needs and expectations.
- Identify essential features, user roles, and specific integration requirements.

## 2. System Architecture Design:

- Plan the overall structure of the web portal, including database design, server architecture, and technology stack.
- Consider scalability, security measures, and performance optimization.

## 3. Technology Stack Selection:

- Choose appropriate programming languages, frameworks, and libraries based on project requirements.
- Select a database management system that aligns with scalability and data storage needs.

## 4. User Interface (UI) and User Experience (UX) Design:

- Design an intuitive and user-friendly interface that meets the needs of both companies and applicants.
- Implement responsive design for a seamless experience across devices.

## 5. Backend Development:

- Develop server-side logic and functionality to handle user authentication, data processing, and integration with external services.
- Implement RESTful APIs or GraphQL for smooth communication between the front and back end.

## 6. Frontend Development:

- Create the user interface based on the design, ensuring a responsive and visually appealing portal.
- Integrate frontend components with the backend using chosen web protocols.

## 7. Database Implementation:

- Set up the database structure based on the design specifications.
- Implement necessary security measures to protect sensitive information.

## 8. Security Implementation:

- Integrate TLS/SSL for secure data transmission.
- Implement user authentication and authorization mechanisms, utilizing OAuth 2.0 or JWT.
- Employ encryption and secure coding practices to safeguard against potential vulnerabilities.

#### 9. Integration with University Systems:

- Collaborate with the University of Belize's IT department to integrate the portal with existing university systems.
- Ensure compliance with university data policies and standards.

#### 10. Testing:

- Conduct thorough testing, including unit testing, integration testing, and user acceptance testing.
- Identify and resolve bugs, usability issues, and performance bottlenecks.

#### 11. Deployment:

- Deploy the web portal on a secure and scalable hosting environment.
- Implement necessary monitoring tools to ensure system stability.

#### 12. Training and Documentation:

- Provide training sessions for users and administrators.
- Develop comprehensive documentation for ongoing maintenance and future updates.

#### 13. Launch:

- Officially launch the web portal for companies to start hiring University of Belize students.
- Monitor performance and user feedback post-launch for further improvements.

#### 14. Finalize for Final Demonstration:

- Prepare system for final demonstration in class

Gantt Chart for the development of the System:

Task	Timeline													
	March							April						
	1	8	10	16	21	27	31	7	14	17	19	22	26	28
<b>Requirements Gathering</b>	1 Day													
<b>System Architecture Design</b>		7 Days												
<b>Technology Stack Selection</b>			2 Days											
<b>User Interface (UI) and User Experience (UX) Design</b>				6 Days										
<b>Backend Development</b>					5 Days									
<b>Frontend Development</b>						6 Days								
<b>Database Implementation</b>							4 Days							
<b>Security Implementation</b>								7 Days						
<b>Integration with University Systems</b>									7 Days					
<b>Testing</b>										3 Days				
<b>Deployment</b>											2 Days			

[illegible]

# References

*Belize Youth unemployment rate 1991-2024.* (n.d.). MacroTrends.

<https://www.macrotrends.net/countries/BLZ/belize/youth-unemployment-rate>

Terra, J. (2023, October 1). Requirements Traceability Matrix (RTM): *UMass Blog - Know*

*what's trending, get expert tips for your career, and learn from industry*

*experts.* [https://bootcamp.umass.edu/blog/project-management/requirements-traceability-](https://bootcamp.umass.edu/blog/project-management/requirements-traceability-matrix)

[matrix](https://bootcamp.umass.edu/blog/project-management/requirements-traceability-matrix)