Technical Specification

for

Cooperative Education Portal

Version 1.4

Prepared by Arir Allana

University of Central Asia, Naryn

25th October 2021

Table of Contents

Table of Contents ii

1. Introduction 1

1.1 Purpose 1

1.2 Intended Audience 1

1.3 Product Scope 1

1.4 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

3. Project Management 3

3.1 Work Plan 3

3.2 Staffing 3

3.3 Risk Management 3

3.4 Monitor and Control 9

4. System Features 9

4.1 Data Flow 9

4.2 User Classes and Characteristics 10

4.3 Use Case Descriptions 12

5. Appendices 14

5.1 Use Case Tables 14

# Introduction

## Purpose

This document describes technical specifications for developing the first release of the Co-operative Education portal for University of Central Asia (UCA).

The Co-operative Education Program is a work integrated learning program where students apply in-class learning to real life professional experiences within the field of study across private, public, and community-based sectors. University of Central Asia (UCA) Co-op Program has been developed in partnership with the University of Victoria, Canada. The major characteristics of a Co-operative education programs include (1) a sequence of work and academic periods, (2) monetary compensation for the work done, (3) an evaluation of the student’s and employer’s performance over the course of the work term; and (4) faculty engagement in the cooperative education program. In cooperative education programs, the work of the student is supervised by an employer and monitored by the academic institution.

## Intended Audience

This document is intended for all stakeholders involved in the implementation and/or use of the website including but not limited to UCA Co-operative Education Team and Development Team.

This document provides agreement on how the website is to be implemented and function. Specifically, it provides a means of:

* Communication between the sponsors and Development Team
* Supporting website design and testing
* Supporting system testing activities
* Supporting validation and verification activities
* Supporting a controlled system evolution

## Product Scope

The e-portal is designed for interaction between students, co-op department staff (organization of students' internships), academic teaching staff and employers. The portal should cover only the processes of the organization of students' internships, without covering the academic processes, the processes of accounting students by department and group, etc. The portal should be publicly available on the Internet with the delimitation of sections in accordance with the rights of the role of the user. Languages of the portal: English and Russian.

## References

The portal will be implemented according to the example of the University of Victoria <https://learninginmotion.uvic.ca/home.htm>.

# Overall Description

## Product Perspective

This is a new product being developed specially to automate manual routine tasks of the co-op department, and provide a platform for students, employers and co-op staff for easy and accessible communication and collaboration.

The Career Portal enables students to search for suitable jobs and match make them with Employers

The main objectives for the portal are:

1. Connecting Alumni with Employers
2. Enabling Co-Op’s to gain work experience while studying
3. Facilitator Students finding & participating in Co-Op’s
4. Counselling for Career Development

## Product Functions

### General:

1. Users (staff/ students) can login using their UCA email credentials
2. Users can change password

|  |  |
| --- | --- |
| **Students** | **Co-op Staff** |
| 1. Can add relevant documents to their profiles i.e. CV, cover letters, transcripts etc. 2. Can view relevant co-op, on-campus, and conference opportunities 3. Can apply for posted jobs 4. Can send approval requests to co-op staff i.e. co-op initial acceptance requests and grade requests 5. Can submit forms i.e. WRT, CAF, and timesheets 6. Can maintain their co-op details on their profiles 7. Can receive comments and grades from co-op staff 8. Can view profile summaries i.e. what documents are missing or how many co-op hours are left | 1. Post job opportunities 2. Receive approval request from students and accepts and rejects it 3. Can view student profiles 4. Can view student submitted documents 5. Can provide feedback and grades to students 6. Can upload individual documents i.e. offer letter and evaluation forms |

# Project Management

## Work Plan

This web application will be developed using agile software development model. Agile model is a form of incremental model which is used when the application must be delivered to the market as quickly as possible, and the requirements are constantly changing depending on the user interaction with the product. New builds can be successfully implemented without losing the original functionality because each release is well tested to ensure all requirements are being fulfilled. Since emphasis is placed on volatility from the start, it requires less effort on the part of the programmers which is both time and cost effective.

Using aforementioned model and a dedicated group of programmers the first release can be delivered in 3 months from the start on project implementation. Later on, updates will be made to the application on a weekly basis based on reviews from end users.

## Staffing

|  |  |  |
| --- | --- | --- |
| **Role** | **Description** | **Assigned To** |
| Analyst | Evaluates technical aspects of project and assigns deadlines  Advises programmers on system and interface design considering requirements of stakeholders | Arir Allana |
| Programmer | Codes the application | To be hired |
| Programmer | Codes the application | To be hired |
| Programmer | Codes the application | To be hired |

All team members report to supervisor

## Risk Management

The are several risks that the project lead must be cognizant of during the software development cycle. The approach to handling risks is to identify the likelihood of their occurrence and create contingencies to tackle them.

Unaddressed risks may eventually cost the project team valuable time and resources and incur displeasure of the stakeholders. They might even cause serious emergencies that can result in conflicts. The project lead must have multiple plans in place so that is a risk cannot be eliminated completely, its effects are minimized so as to not cause hindrance to the end user.

Various risks and their effects are assigned probabilities in order to prioritize the worst and deal with them quickly. This keeps the development process running smoothly.

### Process

The diagram below summarizes the risk management procedure (Satya Narayan Dash, 2019):

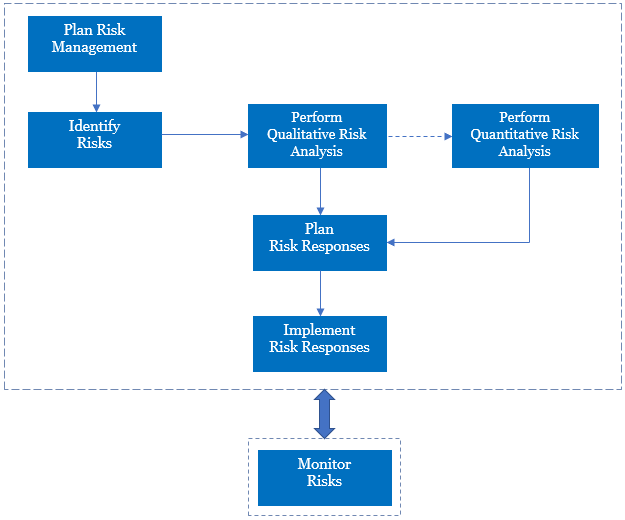


Image Source: <https://www.mpug.com/articles/fundamentals-of-project-risk-management-framework/>

### Risk Identification

The following are the risk associated with developing and implementing the system:

|  |  |
| --- | --- |
| **Type of Risk** | **Description** |
| Technical | * Hardware and software resources used for development are not available * Issues when integrating off the shelf packages and converting data * Hardware and software resources used for implementation are not optimal |
| Personnel | * The development team members are absent when required * Lack of qualified personnel to fulfil job requirements |
| Estimation | * Project delivery will exceed its schedule * The cost of software development was underestimated * The end user work environment was not well understood |
| Tools | * Code is overly lengthy and/or takes too much time to execute |

### Risk Analysis

Probabilities will be assigned to risks based on their likelihood of occurrence. Scores will be calculated to rank risks.

#### Qualitative Risk Analysis

The following probability ranges have been decided but may be subject to change as the software life cycle progresses:

* High Risk: Probability of occurrence greater than or equal to 0.75
* Medium Risk: Probability of occurrence greater than 0.25 and less than 0.75
* Low Risk: Probability of occurrence less than or equal to 0.25

Risks can be further grouped into how serious they are. High impact risk can cause major hindrance to performance, deadlines, and loss of resources. On the other hand, medium and low impact risk are less threatening and do not waste as many resources. The following risk register shows the probabilities of various risks identified previously as well as their impacts.

|  |  |  |  |
| --- | --- | --- | --- |
| **Risk** | **Probability** | **Impact** | **Color Code** |
| Issues when integrating off the shelf packages and converting data | Medium | High |  |
| Code is overly lengthy and/or takes too much time to execute | Medium | High |  |
| Project delivery will exceed its schedule | Medium | High |  |
| Hardware and software resources used for development are not available | Low | High |  |
| Hardware and software resources used for implementation are not optimal | Low | High |  |
| The cost of software development was underestimated | Low | High |  |
| The end user work environment was not well understood | Medium | Medium |  |
| Lack of qualified personnel to fulfil job requirements | Medium | Medium |  |
| The development team members are absent when required | Low | Medium |  |

High and medium impact risks must be eliminated or contained at all costs as will be explained further.

#### Quantitative Risk Analysis

The following table assigns values to probabilities and impacts and ranks risks based on their scores. All values are arbitrary and are subject to change as the development process progresses.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk Name** | **Type** | **Probability (P)** | **Impact (I)** | **Score = P\*I** |
| Issues when integrating off the shelf packages and converting data | Technical | 0.5 | 0.4 | 0.2 |
| Code is overly lengthy and/or takes too much time to execute | Tools | 0.5 | 0.4 | 0.2 |
| Project delivery will exceed its schedule | Estimation | 0.5 | 0.4 | 0.2 |
| Hardware and software resources used for development are not available | Technical | 0.25 | 0.4 | 0.1 |
| Hardware and software resources used for implementation are not optimal | Technical | 0.25 | 0.4 | 0.1 |
| The cost of software development was underestimated | Estimation | 0.2 | 0.4 | 0.08 |
| The end user work environment was not well understood | Estimation | 0.5 | 0.3 | 0.15 |
| Lack of qualified personnel to fulfil job requirements | Personnel | 0.15 | 0.3 | 0.045 |
| The development team members are absent when required | Personnel | 0.25 | 0.3 | 0.075 |

### Risk Response Planning

Risks of tools type will be assigned to programmers, those related to estimation and technology will be delegated to analyst. The analyst will also provide advice to other members on handling risks in their own areas.

Risks will be labelled as following:

* **Eliminate:** High impact risks that must be dealt with at the earliest
* **Contain:** High or medium impact risks that must be avoided or whose effects must be mitigated in short run
* **Accept:** Low impact risks which pose negligible threat. Must be addressed in long run
* **Outsource:** Risks that must be assigned to someone not on the team

The table below shows the classification of identified risks as per the aforementioned labels as well as their possible solutions:

|  |  |  |  |
| --- | --- | --- | --- |
| **Risk Name** | **Score** | **Response** | **Solution** |
| Issues when integrating off the shelf packages and converting data | 0.2 | Contain | Versatile programming languages that can integrate various packages must be used. Appropriate tools for data conversion must be found |
| Code is overly lengthy and/or takes too much time to execute | 0.2 | Accept | Languages with comprehensive documentation and large developer community should be utilized |
| Project delivery will exceed its schedule | 0.2 | Eliminate | Project timeline must be confirmed with sponsor and made available to all team members |
| Hardware and software resources used for development are not available | 0.1 | Contain | Optimal resources must be purchased as soon as possible |
| Hardware and software resources used for implementation are not optimal | 0.1 | Contain | Optimal resources must be purchased as soon as possible |
| The cost of software development was underestimated | 0.08 | Eliminate | Detailed research must be done on development costs before providing quotation to sponsor |
| The end user work environment was not well understood | 0.15 | Contain | Market research must be done to understand usage |
| Lack of qualified personnel to fulfil job requirements | 0.045 | Contain | Train available personnel in required skills given sufficient time exists or hire foreign nationals provided budget is enough |
| The development team members are absent when required | 0.075 | Accept | Organize team building activities and make schedules more adaptable |

## Monitor and Control

In order to keep a check on the risks, the team will hold a meeting every week where members will discuss the risk management plan and analyst will provide input where necessary. Furthermore, communication can also be done via platforms like Microsoft Teams where short video conferences can be held to seek urgent consultation on a pressing matter.

Risk scores may be revaluated based on feedback from user, sponsor, or other valid reasons like budget cuts. Progress reports will be sent to the analyst every month where decision will made on whether the risk has been successfully tackled or that it requires better solutions.

Lastly, meetings will also be held frequently with sponsor to debate the performance of the application and update any functionality requirements from users.

# System Features

## Data Flow

View DFD\_Admin.jpeg, DFD\_Co-op.jpeg, DFD\_Student.jpeg, DFD\_Staff.jpeg, DFD\_Employer.jpeg.

## User Classes and Characteristics

View ClassDiagram.jpeg

|  |  |  |
| --- | --- | --- |
| Class Name: Registered User | ID: 1 | Type: domain |
| Description: Sign in Page | | Associated Use Cases: |
| Responsibilities:   * Allows registered user to access their account by entering email/StudentID and password * Validates email/StudentID and password * Store personal information of users * Allows registered user to access their account and send messages to other registered users | | Collaborators: |

|  |  |  |
| --- | --- | --- |
| Class Name: Student | ID: 2 | Type: sub-domain |
| Description: Student’s account | | Associated Use Cases: |
| Responsibilities   * Verifies student login * Allows students to view jobs, edit CV, apply for a job, accept/reject a job, and submit documents * Store personal information of students (StudentID, password, major, year). * Verifies whether a student accepted or rejected a job | | Collaborators: |

|  |  |  |
| --- | --- | --- |
| Class Name: Co-op | ID: 3 | Type: sub-domain |
| Description: Co-op Staff’s account | | Associated Use Cases: |
| Responsibilities   * Verifies Co-op staff login * Allows Co-op staff to view jobs, verify jobs, view job applications by students, view feedback submitted by employer/staff, view documents submitted by student * Store personal information of Co-op staff ( password, position). * Allows Co-op staff to verify whether a student has submitted required documents and received feedback * Allows Co-op staff to complete an active job of a student | | Collaborators: |

|  |  |  |
| --- | --- | --- |
| Class Name: Staff | ID: 4 | Type: sub-domain |
| Description: Staff’s account | | Associated Use Cases: |
| Responsibilities   * Verifies faculty/staff login * Allows staff to add activities, process an application by accepting/rejecting it and provide feedback on active jobs * Store personal information of staff (password, position, department). * Verifies whether staff accepted or rejected an application | | Collaborators |

|  |  |  |
| --- | --- | --- |
| Class Name: Employer | ID: 5 | Type: sub-domain |
| Description: Employer’s account | | Associated Use Cases: |
| Responsibilities   * Verifies employer login * Allows employer to add jobs, process an application by accepting/rejecting it and provide feedback on active jobs * Store personal information of employer (password, position, company). * Allows employer to view student profiles * Verifies whether employer accepted or rejected an application | | Collaborators |

## Use Case Descriptions

View UseCaseDiagram.jpeg

**Actors:**

* Registered User
* Student
* Employer
* Co-op
* Staff
* Admin
* Server

**Use cases:**

1. Sign in (include):
   1. Authenticate (extend)
2. Edit CV:
   1. Sign in (include):
3. View Available Jobs (extend)
   1. Sign in (include)
   2. Verify Job (include)
4. View Applications (include)
   1. Sign in (include)
5. Apply for Job (include)
   1. View Available Jobs (include)
   2. Add Application Doc (extend)
6. Accept Reject (include)
   1. Apply for Job (include)
   2. View Applications (include)
7. View Feedback (include)
   1. Sign in (include)
   2. Provide feedback (extend)
8. Submit Doc (extend)
   1. Sign in (include)
9. View Submissions (include)
   1. Sign in (include)
   2. Submit Doc (extend)
10. Add Job (extend)
    1. Sign in (include)
11. Provide Feedback (extend)
    1. Sign in (include)
12. Verify Job (include)
    1. View Jobs (include)
13. Complete Job
    1. Verify (include)
14. Edit Profile
    1. Sign in (include)
15. Send Message
    1. Sign in (include)
16. View Student
    1. Sign in (include)
    2. Add Student (extend)
17. View Staff
    1. Add Staff (extend)

Sign in (include): the user enters username and password to login

Authenticate (extend): username and password are verified by server

Edit CV: student can make changes to their CV fields

View Available Jobs (extend): student can view unfilled jobs approved by Co-op

Verify Job (include): Co-op can verify whether a posted job meets required standards

View Applications (include): student can view statuses of all their applied jobs

Apply for Job (include): student can apply for an unfilled job

Add Application Doc (extend): student can add supporting documents such as cover letter and transcripts on their job application

Accept Reject (include): student can accept or reject a job offer

View Feedback (include): student can view feedback provided by their employer

Provide feedback (extend): employer staff can provide feedback

Submit Doc (extend): student can submit work term report and reflections

View Submissions (include): Co-op can view student submissions

Add Job (extend): registered staff/company can add new job

View Jobs (include): Co-op can view all posted jobs

Complete Job : Co-op can complete a student’s job (having viewed submissions and feedback)

Verify (include): Co-op can check a student’s submissions and employer feedback

Edit Profile : user can edit profile info

Send Message : user can send message to another

View Student : admin, Co-op, employer can view all enrolled students

Add Student (extend): admin can add new student to system

View Staff: admin can view registered staff/employers

Add Staff (extend): admin can add new staff/ employer

# Appendices

## Use Case Tables

|  |  |  |  |
| --- | --- | --- | --- |
| Use case name | Sign in | Use case ID | 1 |
| Importance Level | High | | |
| Use case type | Essential | | |
| Primary actor | Registered user | Use case type | Essential |
| Stakeholders and interests | user must login to the system | | |
| Brief description | This case can be used by registered users to login to their accounts and perform activity | | |
| Trigger | Registered user | Type | External |
| Relationships | Association:  Include:  Extend:  Generalization: | Authenticate | |
| Normal flow of events | user must sign in before performing any activity | | |
| Subflows |  | | |
| Alternative/ Exceptional Flows | In case of loss of sign in details; the user may click ‘forgot username or password’ to reset them | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use case name | Authenticate | Use case ID | 1.1 |
| Importance Level | High | | |
| Use case type | Essential | | |
| Primary actor | Registered user | Use case type | Essential |
| Stakeholders and interests | user, sign in, server | | |
| Brief description | Validates if the user has registered on the server | | |
| Trigger | Sign in | Type | Internal |
| Relationships | Association:  Include:  Extend:  Generalization: | Server | |
| Normal flow of events | Sign in details entered are verified | | |
| Subflows |  | | |
| Alternative/ Exceptional Flows | Sign in is declined if either username of password is invalid | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use case name | Edit CV | Use case ID | 2 |
| Importance Level | High | | |
| Use case type | Essential | | |
| Primary actor | student | Use case type | Essential |
| Stakeholders and interests | student must sign in on the system if they need to edit CV | | |
| Brief description | student can make changes to their CV fields | | |
| Trigger | Sign in | Type | Optional |
| Relationships | Association:  Include:  Extend:  Generalization: | Sign in | |
| Normal flow of events | Fields can be saved after editing to make changes permanent | | |
| Subflows |  | | |
| Alternative/ Exceptional Flows | Mandatory fields like name, year of graduation, major etc. must be filled before saving | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use case name | View Available Jobs | Use case ID | 3 |
| Importance Level | High | | |
| Use case type | Essential | | |
| Primary actor | Student | Use case type | Essential |
| Stakeholders and interests | Student must sign in to view jobs | | |
| Brief description | student can view all available jobs | | |
| Trigger | Sign in | Type | Essential |
| Relationships | Association:  Include:  Extend:  Generalization: | Sign in, Verify Job | |
| Normal flow of events | Student may apply to an unfilled job | | |
| Subflows |  | | |
| Alternative/ Exceptional Flows | Student may only view jobs that are not filled and are approved by Co-op | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use case name | View Applications | Use case ID | 4 |
| Importance Level | High | | |
| Use case type | Essential | | |
| Primary actor | Co-op, student | Use case type | Essential |
| Stakeholders and interests | User must sign in to view applications | | |
| Brief description | Student/co-op can view statuses of all their applied jobs | | |
| Trigger | Sign in | Type | Essential |
| Relationships | Association:  Include:  Extend:  Generalization: | View Available Jobs | |
| Normal flow of events | Applications are viewed by employer and accepted / rejected | | |
| Subflows |  | | |
| Alternative/ Exceptional Flows | Applications may not be deleted by user | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use case name | Apply for Job | Use case ID | 5 |
| Importance Level | High | | |
| Use case type | Essential | | |
| Primary actor | student | Use case type | Essential |
| Stakeholders and interests | Student must sign in to apply for jobs | | |
| Brief description | Student/co-op can view descriptions of all available jobs | | |
| Trigger | Sign in | Type | Essential |
| Relationships | Association:  Include:  Extend:  Generalization: | Sign in  Add Application Doc | |
| Normal flow of events | Jobs are made available by co-op and then become visible to student | | |
| Subflows |  | | |
| Alternative/ Exceptional Flows | Only vacant jobs can be applied to | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use case name | Add Application Doc | Use case ID | 5.2 |
| Importance Level | High | | |
| Use case type | Essential | | |
| Primary actor | student | Use case type | Essential |
| Stakeholders and interests | Student must sign in to add application document | | |
| Brief description | student can add supporting documents such as cover letter and transcripts on their job application | | |
| Trigger | Sign in | Type | Essential |
| Relationships | Association:  Include:  Extend:  Generalization: |  | |
| Normal flow of events | Documents may be required to complete certain job applications | | |
| Subflows |  | | |
| Alternative/ Exceptional Flows | Document can only be added after a job application is started | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use case name | Accept Reject | Use case ID | 6 |
| Importance Level | High | | |
| Use case type | Essential | | |
| Primary actor | student | Use case type | Essential |
| Stakeholders and interests | Student must sign in to accept/reject job offer | | |
| Brief description | student can accept or reject a job offer | | |
| Trigger | Sign in | Type | Essential |
| Relationships | Association:  Include:  Extend:  Generalization: | Apply for Job, View Applications | |
| Normal flow of events | The status of job is updated to filled after student accepts | | |
| Subflows |  | | |
| Alternative/ Exceptional Flows | Student may only accept/reject a job once. If they reject, the student may not reapply to the job | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use case name | View Feedback | Use case ID | 7 |
| Importance Level | High | | |
| Use case type | Essential | | |
| Primary actor | student | Use case type | Essential |
| Stakeholders and interests | Student must sign in to view feedback | | |
| Brief description | student can view feedback provided by their employer | | |
| Trigger | Sign in | Type | Essential |
| Relationships | Association:  Include:  Extend:  Generalization: | Sign in  Provide Feedback | |
| Normal flow of events | The employer feedback is made available at the end of job to both the student and co-op | | |
| Subflows | Co-op may complete student job and grant credit after receiving employer feedback | | |
| Alternative/ Exceptional Flows |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use case name | Provide Feedback | Use case ID | 7.2 |
| Importance Level | High | | |
| Use case type | Essential | | |
| Primary actor | Employer/staff | Use case type | Essential |
| Stakeholders and interests | Employer must sign in to provide feedback | | |
| Brief description | Employer/ staff can provide feedback | | |
| Trigger | Sign in | Type | Essential |
| Relationships | Association:  Include:  Extend:  Generalization: | Sign in | |
| Normal flow of events | The employer feedback is made available to student and co-op | | |
| Subflows |  | | |
| Alternative/ Exceptional Flows | Employer may only provide feedback once | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use case name | Submit Doc | Use case ID | 8 |
| Importance Level | High | | |
| Use case type | Essential | | |
| Primary actor | student | Use case type | Essential |
| Stakeholders and interests | Student must sign in to submit doc | | |
| Brief description | student can submit work term report and reflections | | |
| Trigger | Sign in | Type | Essential |
| Relationships | Association:  Include:  Extend:  Generalization: | Sign in | |
| Normal flow of events | The work term report and reflections are made available to co-op | | |
| Subflows |  | | |
| Alternative/ Exceptional Flows | Docs may only be submitted for jobs that have ended | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use case name | View Submissions | Use case ID | 9 |
| Importance Level | High | | |
| Use case type | Essential | | |
| Primary actor | Co-op | Use case type | Essential |
| Stakeholders and interests | Co-op must sign in to view student submissions | | |
| Brief description | Co-op can view student submissions | | |
| Trigger | Sign in | Type | Essential |
| Relationships | Association:  Include:  Extend:  Generalization: | Sign in  Submit Doc | |
| Normal flow of events | Co-op may complete student job and provide credits after receiving submissions | | |
| Subflows |  | | |
| Alternative/ Exceptional Flows | Submissions are only available for jobs that have ended | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use case name | Add job | Use case ID | 10 |
| Importance Level | High | | |
| Use case type | Essential | | |
| Primary actor | employer/co-op | Use case type | Essential |
| Stakeholders and interests | Employer/co-op must sign in before adding job | | |
| Brief description | registered staff/employer or co-op can add new job | | |
| Trigger | Sign in | Type | Essential |
| Relationships | Association:  Include:  Extend:  Generalization: | Sign in | |
| Normal flow of events | Job may now be verified | | |
| Subflows |  | | |
| Alternative/ Exceptional Flows | Job must have unique ID and may only be posted once | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use case name | Verify job | Use case ID | 12 |
| Importance Level | High | | |
| Use case type | Essential | | |
| Primary actor | co-op | Use case type | Essential |
| Stakeholders and interests | co-op must sign in before verifying job | | |
| Brief description | Co-op can verify whether a posted job meets required standards | | |
| Trigger | Sign in | Type | Essential |
| Relationships | Association:  Include:  Extend:  Generalization: | View Job | |
| Normal flow of events | Job is made available to student | | |
| Subflows |  | | |
| Alternative/ Exceptional Flows | Only verified jobs are available to student | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use case name | View Job | Use case ID | 12.1 |
| Importance Level | High | | |
| Use case type | Essential | | |
| Primary actor | co-op | Use case type | Essential |
| Stakeholders and interests | co-op must sign in before viewing jobs | | |
| Brief description | Co-op can view all posted jobs | | |
| Trigger | Sign in | Type | Essential |
| Relationships | Association:  Include:  Extend:  Generalization: | Sign in  Add Job, View available jobs | |
| Normal flow of events | Jobs can be verified to be made available to students | | |
| Subflows |  | | |
| Alternative/ Exceptional Flows | Only jobs posted by employer/staff, co-op or admin are shown | | |