Service Specification

# **Introduction**

The purpose of this service specification document is to provide an overview of the Project Digitise and functionalities it provides according to the requirements provided.

The aim is to document the key aspects of Project Digitise at a logical level.

This service specification is intended to be read by system engineers and developers in charge of hosting and maintaining an instance of Project Digitise.

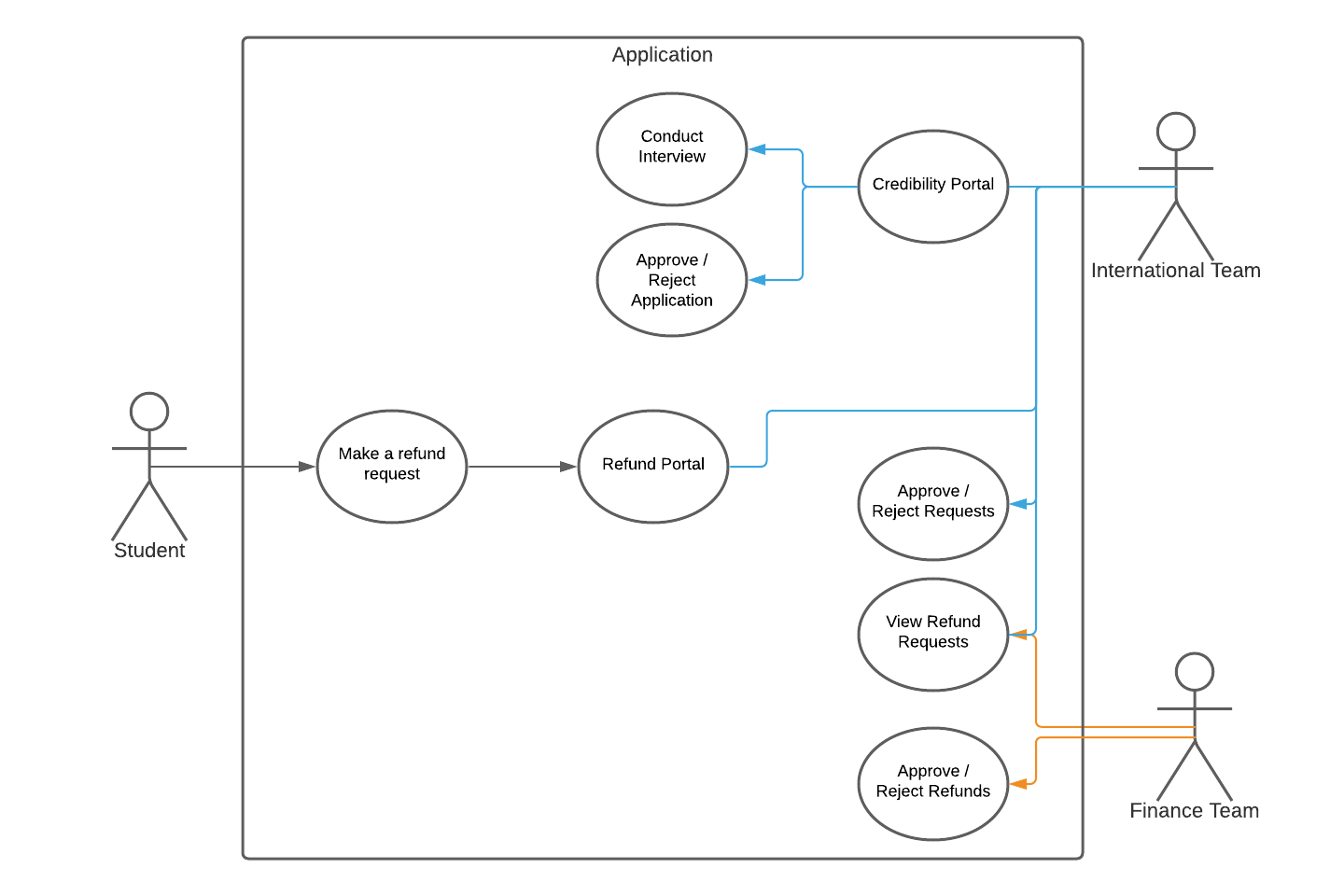
**Inputs from other projects**

This project integrates a previous application (namely the credibility portal) in the aim of merging both applications into a single portal application.

**Application Identification**

|  |  |
| --- | --- |
| Name | Project Digitise |
| Version | 1.0.0 |
| Description | A portal application for both the credibility team and the international team to view/process their respective applications. |
| Developers | Awais, James, *Ahmad, Josh* |
| Status | Provisional (Created but not ready for release) |

# **Software Overview**



The purpose of this software is to provide an endpoint for the student to make a refund request to be sent to the International Team which can later be processed by the finance team to complete the application.

This will also merge the previously mentioned credibility portal which is used to conduct credibility interviews by the international team resulting in an approved/rejected application which would redirect the user to the refund portal.

# **Software Scope**

(Clarify requirements)

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement # | Subsystem? | Requirement Name | Requirement Description |
| 1 | Refund | Submit requests | Allow students to submit refund applications |
| 2 | Refund | Send emails | To send the students a confirmation when the application has been received. |
| 3 | Refund | Review applications | Allow staff to review the applications (through an interface) |
| 4 | Refund | Manage applications | Allow staff to accept or deny refund applications |
| 5 | Refund | Inform management | To send the staff members an email when a new refund application is received. |
| 6 | Credibility | Randomise question set | To allow the interviewer to call upon a random set of questions for each interview. |
| 7 | Credibility | Guided interview | To provide the interviewer with a guide in which manner to conduct the interviewer |
| 8 | Credibility | Informing both parties. | To ensure that all the relevant information is provided to both the interviewer and interviewee |
| 9 | Credibility | Submit responses | Ensure all information and responses are logged for later review. |

# **Infrastructure Requirements**

* SQL
* NODEJS
  + Express
  + Jspdf
  + Body-parser
  + Ejs
  + Cookie-parser

# **Functional Requirements**

Classes, functions, methods)

In tandem with the MYSQL requirement, the application makes use of “pools” to manage the connections between the application and the database. To ensure the application is always available without having the connection break. The connection is only made from the “pool” when a statement is required to be executed. Once the statement is completed the connection is closed and returned to the “pool”.

To serve the individual pages, the application has many GET requests that render their respective pages in HTML for the user to view.

A GET request for the credibility interview would result in a render of the interview page.

When a POST request is received (from submitting a form) the response is parsed and integrated into their respective commands with their respective functions performing the tasks necessary.

A POST request to add a credibility question would result in the question, answer, sample answer being formatted into an appropriate SQL statement to add the information to the database.

A few other functions are executed within the application such as:

* getDate() which simply returns a data value as a string.
* MakePDFFile(data) which takes the submitted information and parses it into a pdf file to be saved in the same directory as the application.

We will need to store the reports that are automatically generated by the credibility portal, this is a requirement, need to find a safe place to store these, that can be accessed by the team.

Database Queries

Get request credibility -> returns a set of questions to be rendered on page.

On form submission of refund request -> send student number -> returns home email

# **Non-Functional Requirements**

As the refund application page itself will be public we will need to separate the rest of the application off. Therefore, will need a log in capability, was suggested to use active directory for this, but as a team we have no experience with this.

As we may hold sensitive data, we will need to encrypt personal details i.e. Bank details etc.

Will need a bolton.ac.uk email that the application can use to send emails from so that the student knows it's from UoB.