**Software Documentation**

**Software Title:** Credibility Interview Question Randomiser

**Current Date:** April/May, 2021

**Authors:**

* James Matcham
* Awais Hussain

**Functional Description & Overview**

To effectively carry out credibility interviews.

The application allows the interviewer to conduct the interview in an orderly manner with each set of questions being randomly generated from a set of predetermined questions. Each interviewer therefore has a separate set questions for the interview allowing for variation in the questions being asked.

## Software Design Considerations

### **Assumptions**

### **Dependencies**

* Cloud Server (AWS)
* NodeJS
* Express
* JsPDF
* MYSQL

### **Constraints**

* Zoom recordings currently being downloaded manually, and stored locally. There is a branch which isn’t live that will **allow** users to ‘upload’ recordings. Recording size per interview is based on quality and time of interview. Best estimate is 100MB per interview, 100 interviews. 10GB, I would recommend that we will need 15GB of storage for a buffer. But that would be under revision after more information.

### **General Guidelines**

* Security to be handled by Active Directory, with access to admin page for only specific people from that group.

### **Objectives**

* To allow the interviewer to call upon a random set of questions each time an interview is conducted
* To provide the interviewer with a guide in which manner to conduct the interviewer.
* Ensure that all the relevant information is provided to both the interviewer and interviewee.
* Ensure all information and responses are logged for later review.
* To save a ‘report’ as a pdf of file that they can come back to review at a later date

### **Methodology**

Express is the main endpoint for the user and their method of interacting with the application. The base page directs the user to the interview form where the fields should be filled out and can be submitted in order to access the other pages.

To generate the sets of questions an SQL statement is executed that returns questions based upon their question type (open or closed) which are then displayed on the form accordingly. As this SQL statement runs upon landing on the page, the questions generated are different each time.

Upon submission, the data gets passed through to the report page where the user can review their submitted information.

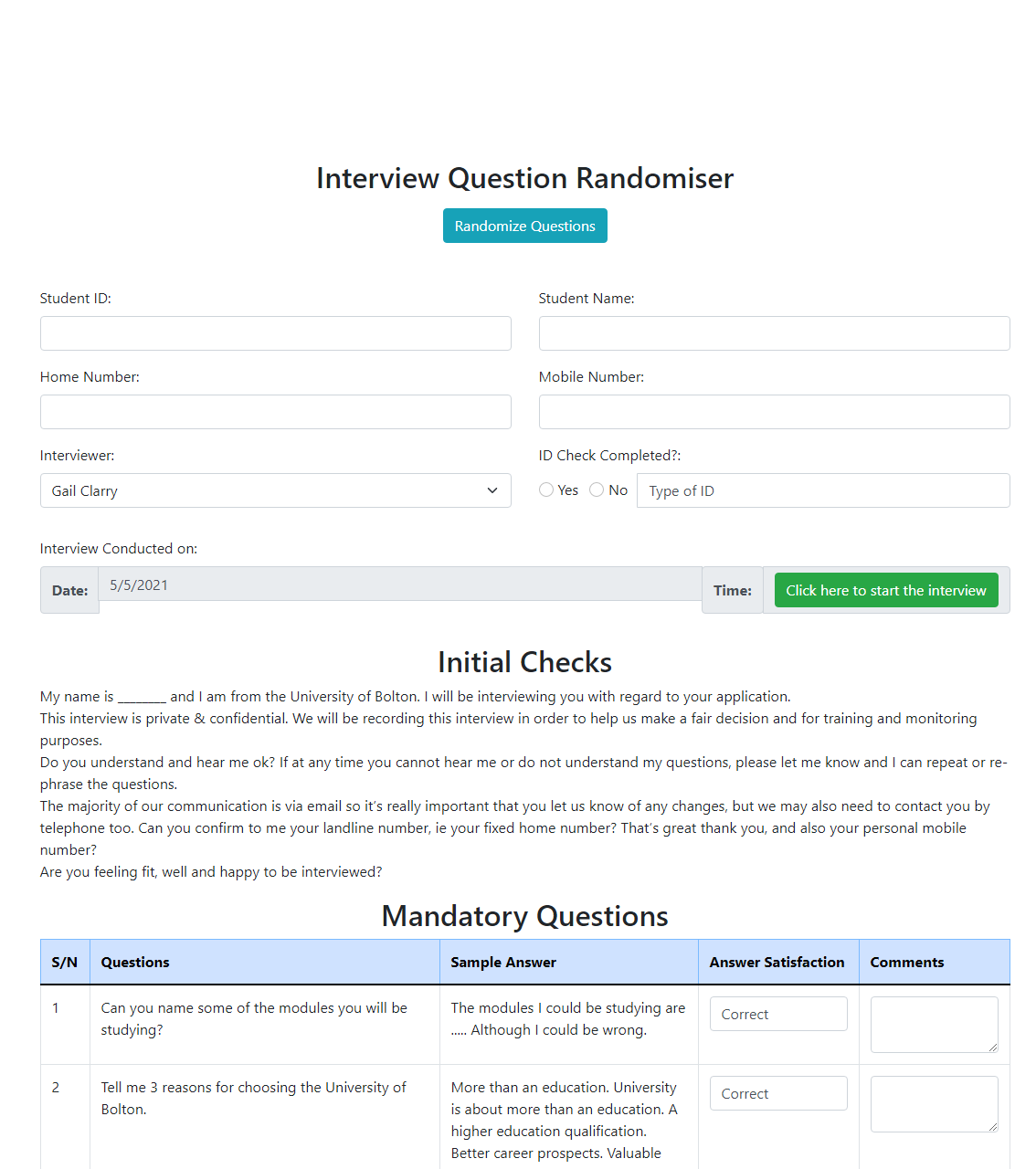
Using JsPDF, the report is generated upon completion of the interview and stored with the interviewees information and interview contents being positioned on the page manually. This allows the user to save the interview data in case the interview is required at a later stage to be printed or simply reviewed.

For the administration side, the admin page returns ALL of the available questions from the database and returns them in a readable table that can be used to perform CRUD (CREATE, READ, UPDATE, DELETE) operations.

Performing any of the CRUD operations results in the SQL statement being executed that will perform the respective action to the question with the ID specified, or in the case of creating a question a new ID will be generate with the question and stored in the database accordingly.

## User Interface

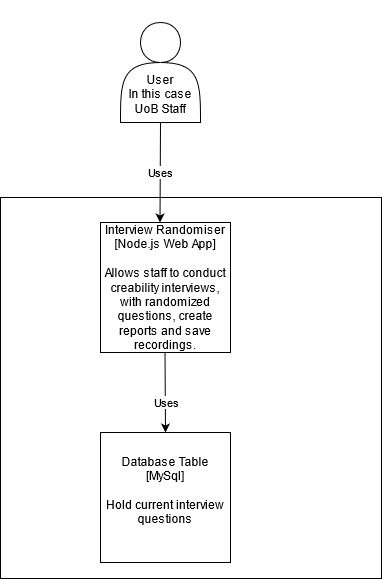
For the interface of the application, the interviewer should expect a straight forward web page with a form that contains information for the interviewer (initial checks) to read and act upon if necessary.



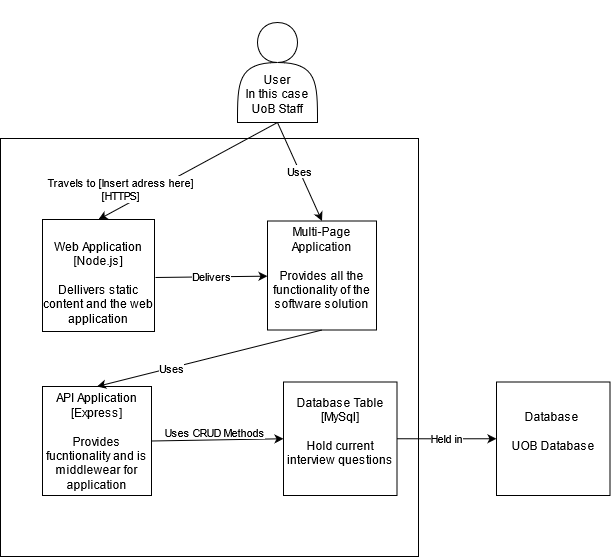
After the interview has been conducted and the form is submitted the user should then be taken to a report page where they can review the information that they have submitted.

## System Architecture

Context View



Container View



## Glossary

Define jargon, terms and concepts as necessary and depending on who will be reading your document.

* Endpoint – Entry point to the service, Point of contact
* SQL Statements – Statements that can be used to perform tasks on the database.

## Additional Comments, Documents or Information

Enclose any key information that was not accounted for in your software design document so far. Feel free to add media, documents, links, videos and the like.

* In the current state of the application, PDF files are generated in the root directory of the application and so far, attempts at storing the reports in a separate file are not working as intended. This could mean the occasional manual organising may be required. **This is still being looked into and does not affect the operation of the application.**