Award Submission Summary

for

SuperTest

v1.0

1/10/2022

Prepared by

Filipe Lagrenade,

Sanjay Williams,

Jia Son Pow, and

Isaac Ellis

Capstone Group I3

**Table of Contents:**

[**Presentation Video**](#_uqgv6urtfjoh) **3**

[**Introduction**](#_o7qok8em3tbl) **3**

[**Organisation/personal Overview**](#_ebl8v120eahl) **3**

[**Product Overview**](#_25nxeqxbyqkb) **4**

[**Quality of Solution**](#_vjdgyo70qeww) **4**

[**Innovation**](#_ym6xulj2kt02) **5**

[**Benefits Realisation**](#_3t1b7xhk4sg1) **6**

# 

# [Presentation Video](https://drive.google.com/file/d/1rM1lAi17V1W7WAs4XGlTndJwhICHPjGt/view?usp=sharing)

# Introduction

SuperTest, an all-in-one solution created for in-class examination invigilation in the Super Lab, providing various well-thought-of features that are focused to help users achieve what’s most important (the test!),on a simple and intuitive web application interface for examination invigilators and students alike.

# Organisation/personal Overview

Our team, Capstone Group I3, was formed through a selection process where our group placement is determined by factors such as project preferences, general academic performances and individual skills. Our team consists of 4 members. We are Filipe, Isaac, Jay and Sanjay.

Filipe is in his penultimate year of his computer science degree. He works full-time as an IT support and administration worker looking to pursue employment as a software developer. For this project he has been primarily a front-end developer.   
  
Isaac is a final year computer science student with over 5 years of industry experience. For the project he has worked on the full stack.  
  
Jay is a final year student who majors in Cyber Security. He is currently a full-time student and working towards acquiring a role as a Cyber Security Advisory Specialist Intern under the DTS team at Curtin University. For this project, he has primarily worked on the documentation and front-end.

Sanjay is unable to fill in his information as he is currently sick. For this project, he has worked on the middleware and collaborated on some parts of the backend with Isaac.

# 

# 

# Product Overview

The main focus of our software application is to provide a brand new platform for examination invigilators/unit coordinators to create, plan and administer assessments in the newly established SuperLab computer lab. The test paper distribution and backing-up of students’ results tackle the issue of contract cheating by creating an equitable testing environment for students. This ensures that test papers are released in unison and the students’ work is adequately recorded.   
  
Additionally, the software application will also allow both students and invigilators to send messages to each other during examinations. Autosaving and an intuitive submission system is built into this application to allow students to effortlessly submit their work at the end of the exam without using too much time as they will no longer need to access an external learning management system like Blackboard. An on-screen timer is provided for time management.

# Quality of Solution

The student client for this application will be running on a linux distribution that will be installed on the computers in the newly built SuperLab.Through the use of Electron.js, the software can be bundled into any native binary which means that if the client would like to expand to having the application working on Windows or MacOS, it will be simple to do so. As the computers are running on a Windows operating system, we will have our application be run on a Linux Virtual Desktop Infrastructure.

On the other hand,The UC Client is being created with portability in mind. This means that the software will most likely be accessed through a web browser, accessible through a tablet, laptop or computer without needing any software installed. This is to fulfil the invigilators’ need to monitor the examinations and create/view/manage assessments outside the SuperLab.

Both clients are very similar in design with a few differences. On the Invigilator/UC client, there is a “Create Assessment” option available where an invigilator can input the relevant details needed to create a new assessment to be undertaken.

As for the security configuration for the application, we have decided to go with JSON Web Token (JWT) as our choice of user authentication method. This is because the JWTs are commonly signed using either a secret hashed algorithm (HMAC) or a public/private key pair (RSA/DSA). This ensures data integrity as when a token is signed with a key pair, it signifies that the party with the private key is the one who signed it. Any alteration to the content will break the digital signature and the server will reject the token.

# Innovation

The UC and student client both follow an architecture with two sections. The first is the view layer which contains the dashboard (which contains details about past and future assessments), login, active assessments, the ability to create an assessment (for UC’s only) and the ability for the students to ask questions to the UC’s.

This then connects to the second layer which contains all of the services. This handles all of the API calls between the clients and the backend. For the UC Client this includes a question/notification API to receive student questions and post notifications, an API to retrieve assessment details and user details for the current active user. For the student client, there are API’s which handle receiving and sending questions/answers, getting assessment details, files and user details.

One of the major innovations is that the student client will also be handling periodic RSync updates which will sync the data in the students working directory to the backend server so they don’t lose their data if something goes wrong (computer crashes etc.). When a student launches an assessment, it will create an RSync connection between the server and the client. The student client will then push the files to the server after a certain interval has elapsed.

On the backend, a three layer architecture design is used. The Controller layer handles API requests by calling functions from the service layer. The service layer functions by interacting with the DAO layer, other services and performs business logic on data. The DAO layer provides functions that allow interaction with databases. User management is handled by embedding a KeyCloak server inside the backend application. This is configurable whilst providing java libraries and an admin interface. The backend allows clients to send files, via rsync, by dynamically creating ssh users during an assessment.

Last but not least, the database will be the storage location for relational tables created and/or used by the application. The student/UC logs and files will be stored on the hard drive of the computers in the SuperLab.

# 

# Benefits Realisation

<Understanding of business environment, financial benefits, social benefits and efficiencies delivered, effectiveness of solution in delivering its stated outcomes (350words max)>  
[Client completes this, or we complete after consultation with client.]

# 