# 11. Evaluation and Testing

## 11.1. Evaluation

### 11.1.1. Plan, scope, and validity

All main features of the software were put through testing in order to ensure output matched expectation. Various visualizations we recorded as a means of observing the way with which the system handled a given task.

## 11.2. Test data

### 11.2.1. User Accounts

The following user accounts are seeded into the database or restored from the provided test data backup.

|  |  |  |
| --- | --- | --- |
| Username | Password | Roles |
| connor.sant.17@um.edu.mt | FYPsRule! | Student |
| jabela@um.edu.mt | FYPMaster! | Supervisor |
| secretary@um.edu.mt | FYPMaster! | Administrator |

### 11.2.2. Students

The below table shows students that are uploaded by means of the system UI using the provided csv file, or by means of restoring the database backup also provided.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Student Id | Student Name | Student Surname | Student Email Address | Student Average Mark |
| 123400L | Joe | Borg | JB@um.edu.mt | 68 |
| 123498M | Ian | Grech | IG@um.edu.mt | 68 |
| 123499M | Nicole | Cassar | nicole.cassar.18@um.edu.mt | 88 |
| 132400L | Jane | Said | JS@um.edu.mt | 65 |
| 143200L | Joshua | Spiteri | joshua.spiteri.18@um.edu.mt | 70 |
| 432100L | Caoimhe | Camilleri | caoimhe.camilleri.18@um.edu.mt | 78 |
| 432199M | Link | Vella | LV@um.edu.mt | 90 |
| 313699M | Connor | Sant Fournier | connor.sant.17@um.edu.mt | 74 |

## 11.2.3. Supervisors

Below is a table of test supervisors that may be imported through the UI using the provided csv file or restored by means of the backup file also provided.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Supervisor Id | Supervisor Name | Supervisor Surname | Supervisor Email Address | Supervisor Total Quota |
| 123456M | Peter | Xuereb | pxuereb@um.edu.mt | 1 |
| 123460M | Michel | Camilleri | michcam@um.edu.mt | 3 |
| 123462M | Ernest | Cachia | ecachia@um.edu.mt | 3 |
| 123463M | Tony | Spiteri Staines | tspiteristaines@um.edu.mt | 1 |
| 123464M | John | Abela | jabela@um.edu.mt | 10 |
| 123465M | Clyde | Meli | cmeli@um.edu.mt | 2 |
| 123468M | Joseph | Vella | jvella@um.edu.mt | 6 |
| 123471M | Colin | Layfield | clayfield@um.edu.mt | 2 |
| 123472M | Joshua | Ellul | jellul@um.edu.mt | 3 |
| 123474M | Matthew | Montebello | mmonte@um.edu.mt | 1 |
| 123475M | Lalit | Garg | lgarg@um.edu.mt | 8 |
| 123476M | Chris | Porter | cporter@um.edu.mt | 1 |
| 123480M | Joseph | Bonello | jbon@um.edu.mt | 3 |
| 123482M | Conrad | Attard | conatt@um.edu.mt | 7 |

## 11.2.4. Areas

The following areas are conventionally entered into the system by each supervisor. However, for testing purposes, the database backup provided contains test data for areas tied to each lecturer. Due to the data being of significant quantity, the below table is only a partial representation of the areas table. The data displayed pertains to supervisor id, description, and quota for each area.

|  |  |  |  |
| --- | --- | --- | --- |
| Supervisor Id | Area Title | Area Description | Area Quota |
| 123460M | ICT in Smart Wheelchairs | As our society evolves the proportion of ageing citizens is increasing. A significant proportion of this sector is dependent on human care - often in an institutionalized environment. One of the biggest costs is that of the human carers themselves. A large proportion of the carers time is spent tending to individual resident needs - among which is transportation rom one part of a nursing institution to another. Various ICT based solutions are being studied to alleviate this problem - help use the time of human carers more efficiently and improve the quality of life of residents through increased mobility and interaction. his may involve the use of different technologies in the fields of IOTs - multi-modal input - location sensing - Smart Wheelchairs as well as algorithmic approaches to navigation - route finding and behaviour modelling. The specific focus of the study will be determined through discussion with the interested student. | 4 |
| 123465M | Computer Security and Communications Issues in Automobiles | Investigation into computer car security issues using various techniques. CAN bus message injection can be used to perform actions like setting the speedometer or wipers. | 2 |
| 123476M | Human-Computer Interaction in Software Testing | This body of work investigates software testing practices - toolsets and processes from an HCI perspective evaluates the impact of work practices on human testers as well as associated outputs. HCI techniques are applied to study human activity - performance and perceptions tackling issues such as information anxiety - workload - efficiency - error rates and knowledge management amongst others. Software testing consists of various disciplines including test automation - reporting - build pipelines - tooling - non-functional testing (e.g. accessibility - security - performance) - bug-management - team management - remote testing - communication - and so forth. | 1 |
| 123464M | AI and Machine Learning | The use of AI & Machine Learning to provide a system that automatically learns and improves from experience without being explicitly programmed to do so. | 2 |
| 123464M | Natural Language Processing | Deciphering the Voynich manuscript - Sentiment Analysis | 3 |
| 123464M | Bioinformatics (DNA & Protein Sequence Classification - Medical Imaging) | DNA & Protein Sequence Classification. Object Detection - Recognition - & Classification in Medical Imaging. | 1 |
| 123464M | Machine Vision | Image Deconvolution (de-blurring) - Image Classification | 1 |
| 123464M | Space Science | Optimal Antenna Placement for the SKA - Deconvolution of Cosmological Images | 2 |
| 123464M | Search and Optimisation | Optimisation and Scheduling (manufacturing - airlines - timetabling - etc.) - Constraint Satisfaction Solving - Logistics and Routing | 2 |
| 123472M | Blockchain and/or Cryptocurrency and/or Smart Contracts | I am interested in supervising projects related to development of tools that enable easier programming models for Blockchain systems. How Blockchain can be used for common/social good uses: Connecting Blockchain/DLTs to the Internet of Things (IoT) Inter-blockchain communication models - and macro-programming models of systems including the Internet of Things (loT) | 1 |
| 123462M | High-Level Description Languages for IoT Programming | Software Engineering. System modelling - High-level description languages software and System Architectures - System Integration - Smart Systems | 2 |
| 123462M | Automated Portal Creation Tools | Software Engineering - System Modelling. Web Programming - Software and System Architectures - Data Processing. Interface Design | 4 |
| 123474M | Autonomous Drone Navigation | This project investigates the notion of have one or more drones automatically following a person to be utilised for a variety of application that include surveillance/security applications. but could also be employed to track/record an athlete/cyclist or even a field researcher. Additional applications include leisure application as well commercial application as in guides for tourists. | 2 |
| 123468M | Database Data Modelling | Data modelling is the basis with which data requirements - of a computer information system - are met. The more complete the constructed model is - with respect to these requirements - the easier is the system development required Furthermore - it was now become very common that a set of data requirements are met with more than one data model for example using relational tor most requirements and then use specialized spatial data modelling to capture GIS data requirements. Since the relational data model - it has become apparent that the issues are dependent on the chosen data modelling language. Some are quality example what type of query languages operate on the data modelled - and others actually affect aspects of system performance. Previously we have used Open-sourced DBMS (e.g. PostgreSQL MongoDB Neo4j) - Oracle - MS SQL Server - DB2 | 1 |
| 123468M | Database Setup - Tuning and Administration | DBMS is a complex software system. Nonetheless DBMSs cater for a wide array of data requirements and corresponding operational requirements (e.g. query response and transactional throughput claims). | 2 |
| 123468M | Digital Forensics | Digital forensics is the science of identifying evidence from digital sources and which provides the forensic experts with robust tools and techniques to solve complicated digital-related crimes (B Camier - 2002). We have had a good number of FYPs in this area: creating a song recognition system (e.g. requested to handle song royalties collection) file carving of image files with a novel process; relating crime occurrences with geo - social and road networks and create a visualization of a company's email corpus (usually required in fraud investigations). Previously we have used Open-sourced DBMS(e.g. PostgreSQL) and other freely and publicly available packages. | 3 |
| 123468M | Data Mining & Data Warehousing: With Massive Datasets | Digital forensics is the science of identifying evidence from digital sources and which provides the forensic experts with robust tools and techniques to solve complicated digital-related crimes (B Camier - 2002). | 2 |
| 123468M | Scalability Issues over Distributed Data | The ACID principles in database transaction processing has met many demanding requirements with great success but with recent data requirements of new applications - e.g. social and mobile - ACID is not the only option. Yet meeting theACID principles in centralized databases was never an easy task either for DBMSs or database designers. Adding data partitioning and replicating partitions across data servers have strained what demand ACID based transactional systems can provide and Scale up to. | 1 |
| 123471M | Voynich Word Prediction | The Voynich Manuscript - often called ‘the most mysterious manuscript in the world’ - resides in the Beinecke Rare Book & Manuscript Library at Yale University (Manuscript MS 408). Since its purchase from the Jesuit order in 1912 by Wifrid Voynich over a century of efforts to decode its mysterious (and only example of its kind) text has proved unsuccessful. In the last few decades the contents of this mysterious text have been transcribed into a computer readable format. There are some challenges with the transcriptions in that - not surprisingly - there is not 100% consensus as to what some of the text transcribes to (if you view images of the pages - even with a magnifying glass - you can see why) The various versions of the transcriptions encode within them alternative transcriptions for the words in question or other indicators of uncertainty. One possible solution to this is to process the text and - using statistical techniques. | 1 |
| 123463M | Real Time Systems | RT computing - cloud computing software development. Design and implement a non-trivial real time system that solves a particular problem in the business domain or a financial problem. | 1 |
| 123475M | Hospital Admission Pattern Analysis - Bed Resource Requirement Forecasting - Allocation & Management | The project would develop novel applications of Al/ML methods in the healthcare management problems such as hospital admission pattern analysis - bed resource requirements forecasting - allocation and management. It would provide students an excellent opportunity to understand how Al/ ML methods can be used for developing solutions to real life problems and also assessing effectiveness of such Al/ML tools. | 2 |
| 123475M | Traffic Congestion Problems in Malta | The project will utilize the modelling - simulation - resource allocation and scheduling methods solve a problem. Also - they would be able to develop skills of problem analysis - solution design - data collection and solution evaluation. Further they would be able to learn developing and implementing algorithms and logic to develop innovative solutions. | 1 |
| 123475M | Digital Footprint Analytics | It is an application of Al techniques (especially machine learning) to social media data analytics. Machine learning - digital footprints - data analytics. social media analytics behavioural health - mental health - neurological disorders - sentiment analysis - opinion mining - natural language processing. | 1 |
| 123475M | Smart Sensor for EEG Acquisition and Epileptic Seizure Detection | The project would assess and develop novel applications of AI/ML methods in EEG analysis and epileptic seizure detection. It would provide students an excellent opportunity to understand how Al/ ML methods can be used for real Iife problems and also developing skills in assessing and developing Al/ML tools. | 2 |
| 123475M | Machine Learning Methods for Handling Missing Data in Medical Questionnaires | The project would assess and develop novel application of AI/ML methods in missing data handling in medical questionnaires. It would provide students and excellent opportunity to understand how AI/ML methods can be used for real life problems and also developing skills in assessing and developing AI/ML tools. | 1 |

## 11.2.5. Student Preference

Seeing as the student preferences table contains a large number of rows, sample rows will be displayed a means of representing the test data contents.

|  |  |  |  |
| --- | --- | --- | --- |
| Preference Id | Student Id | Area Id | Time Submitted |
| 1 | 123400L | 30 | 2020-05-19 09:38:09.0313331 |
| 2 | 123498M | 7 | 2020-05-19 09:39:48.6361068 |
| 3 | 132400L | 35 | 2020-05-19 09:45:11.7805281 |
| 4 | 210199M | 17 | 2020-05-19 09:47:36.5423667 |
| 5 | 432199M | 21 | 2020-05-20 16:29:21.5917367 |
| 6 | 593898M | 2 | 2020-05-21 09:29:29.8551704 |

## 11.3. Test Cases

### 11.3.1. Importation of data

Assumption: The main assumption taken for this part of testing is to determine if the system is able to import uploaded csv files and store them on the database.

#### Test case 1: Importing a list of students

The following test was conducted to check if the importation feature can accept student data as a csv file and subsequently save the given data.

To test this feature, a csv file composed of test data for students was selected. An import operation was carried out and proceed as follows:

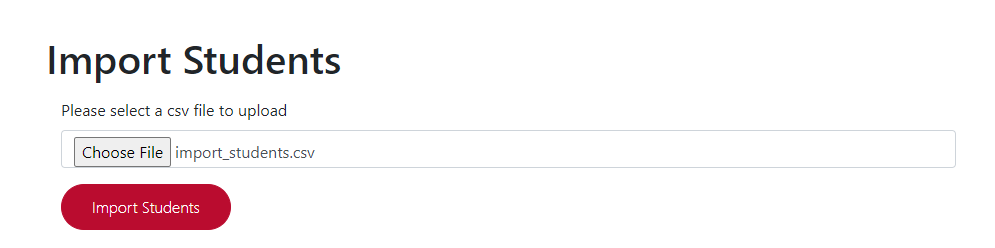


Figure : csv file upload with student data

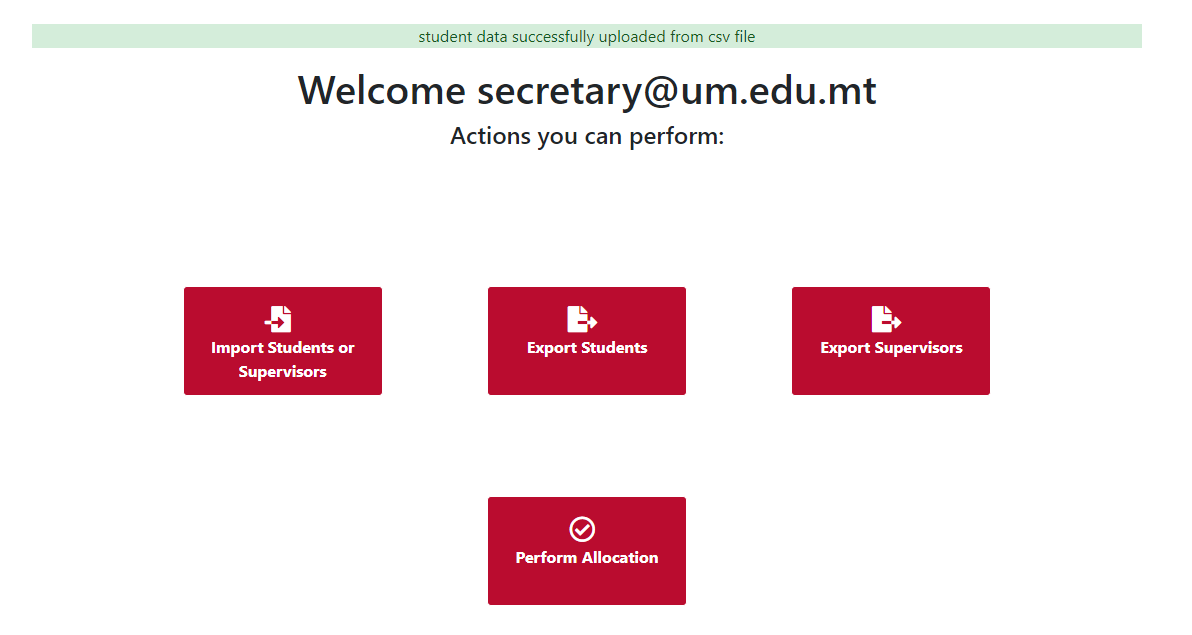


Figure : Prompt displayed as a means of notifying administrators of a successful upload

The displays above show the upload field being populated and a success notification being display, and therefore, the test was successful.

#### Test case 2: Importing a list of supervisors

The following test was carried out as a means of checking if the importation feature can accept supervisor data as a csv file and subsequently save the given data.

To test this feature, a csv file composed of test data for supervisors was utilised. An import operation was carried out and proceed as follows:



Figure : Uploading of csv file with supervisor data

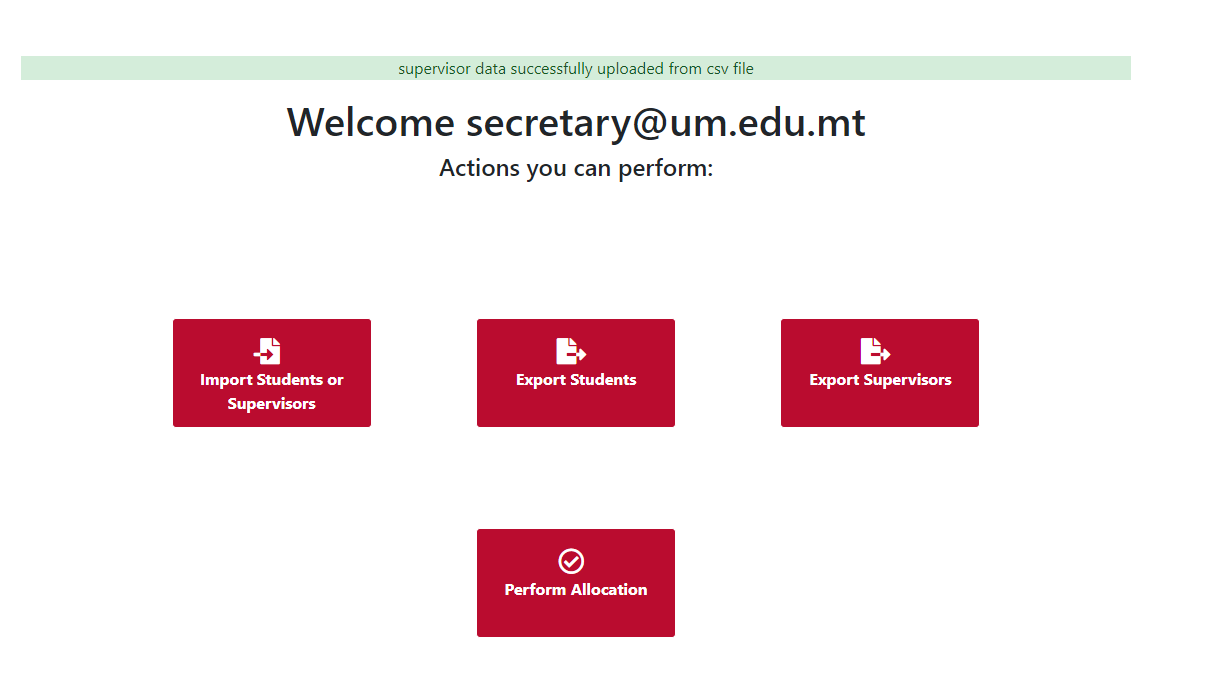


Figure : Notification of successful upload and storage of data

The displays above show a file being specified within the upload filed and a notification being display prompting success, therefore, the test has proven to be successful.

### 11.3.2. Submission of student preferences

Assumption: The main assumption made in the case of this part of testing is to ensure that the system is able to allow students to submit final year project preference forms.

#### Test case: Submitting preferences form as a student

This test was done in order to check that the system will accept a given choice of preference submitted by a student and subsequently store the submitted data within the database.

To test this feature, a student login was utilised, with the preferences form being filled in for the student, and submitted into the system as follows:

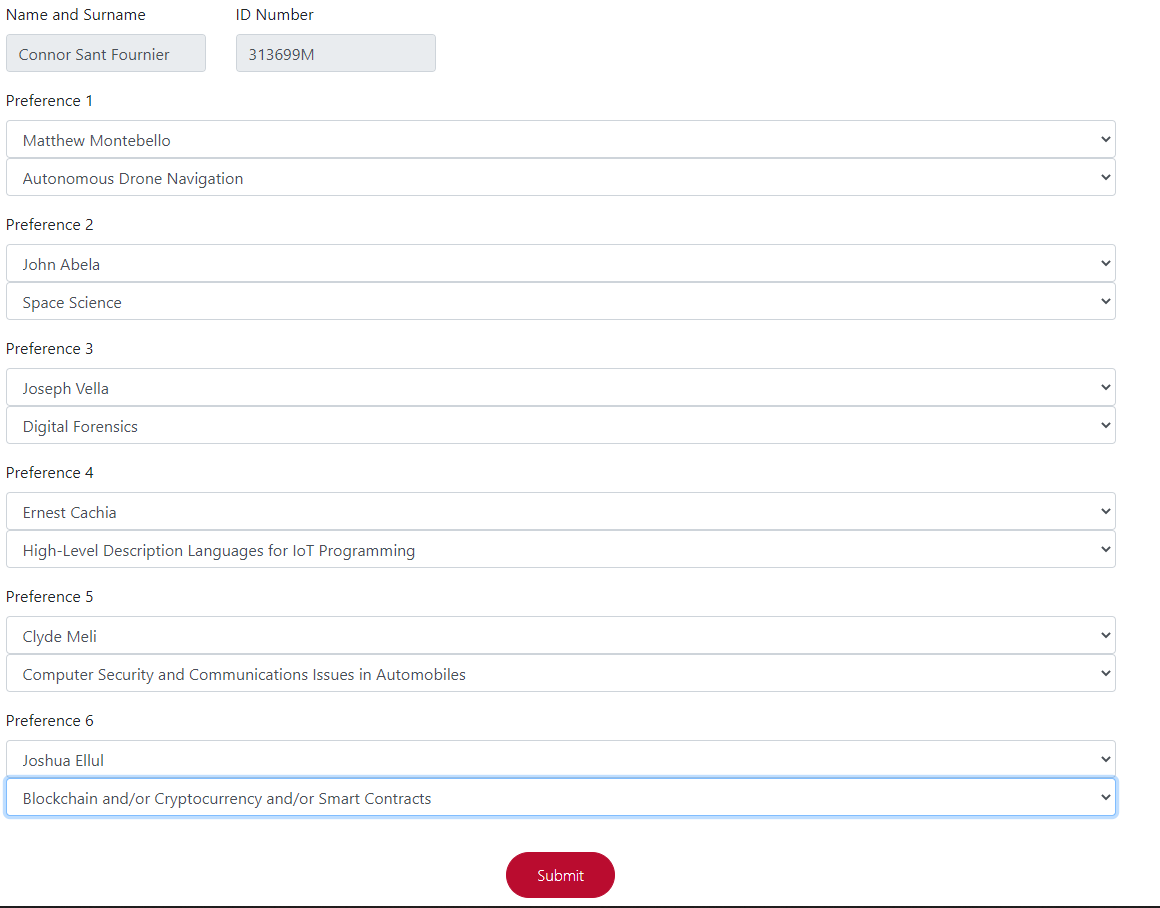


Figure : Complete preferences submission form for students

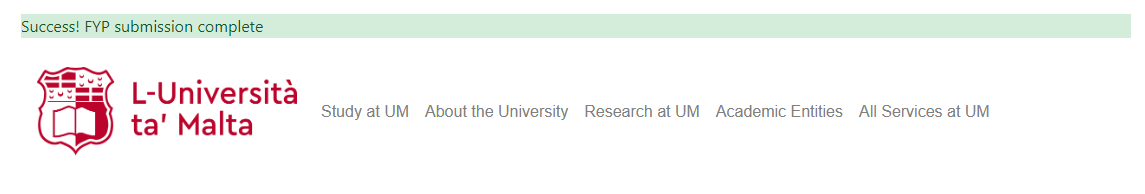


Figure : Notification of successful submission

As may be observed in the above displays, a completed form is submitted, with a success message notifying that data has been uploaded and save. By means of such a result, it is possible to state that the test was completed successfully.

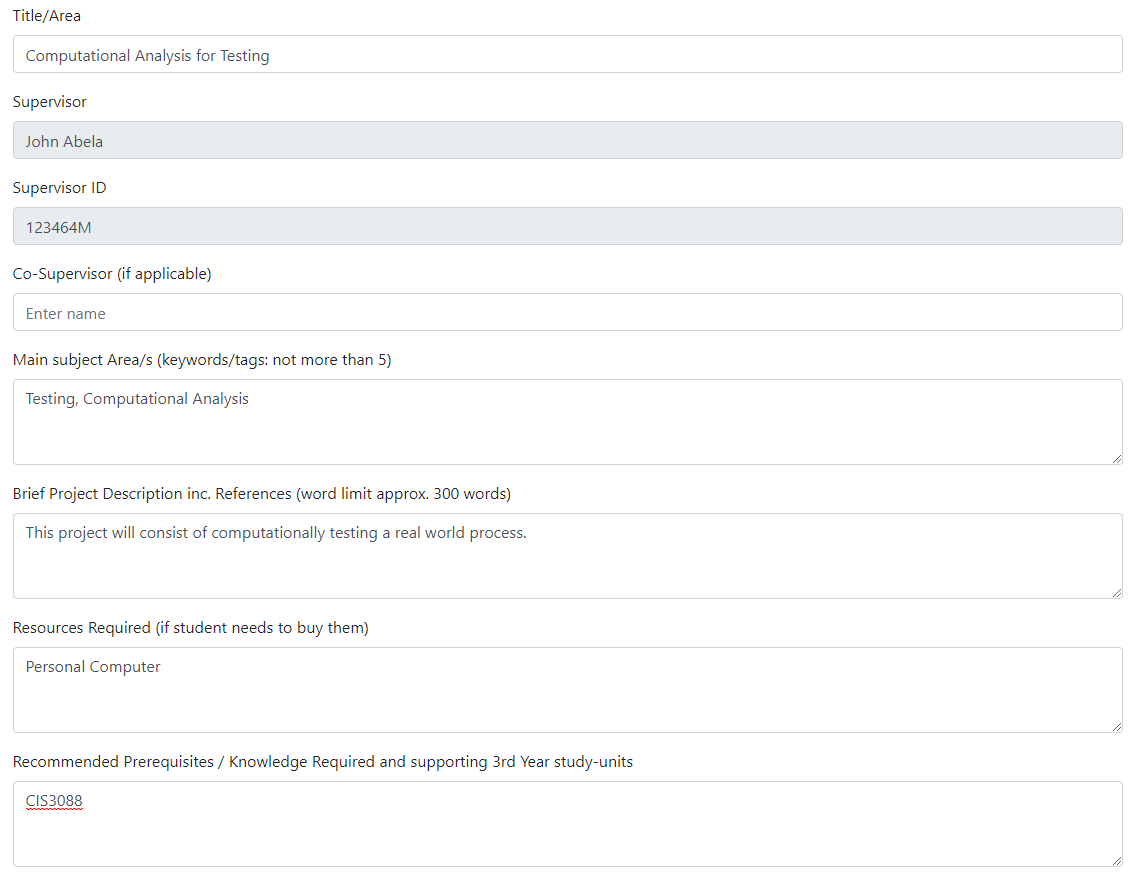
### 11.3.2. Supervisor actions

Assumption: The main assumption for this segment of testing is to determine whether or not the system provides complete services for supervisors to add, edit, and delete an area.

#### Test case 1: Submitting an area proposal

By means of testing, this feature was evaluated to ensure that data submitted from the form was successfully stored within the database.

For this feature, testing was to be done by means of filling in and submitting a new area for a test user being in the role of a supervisor.



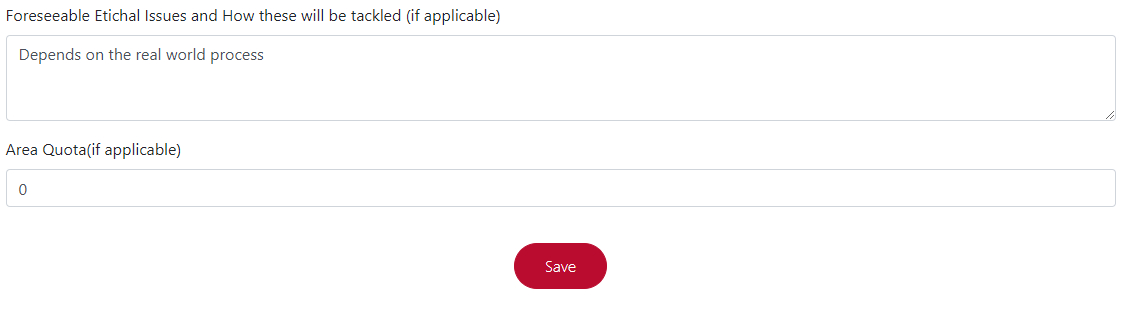


Figure : Filled in area proposal form for supervisors

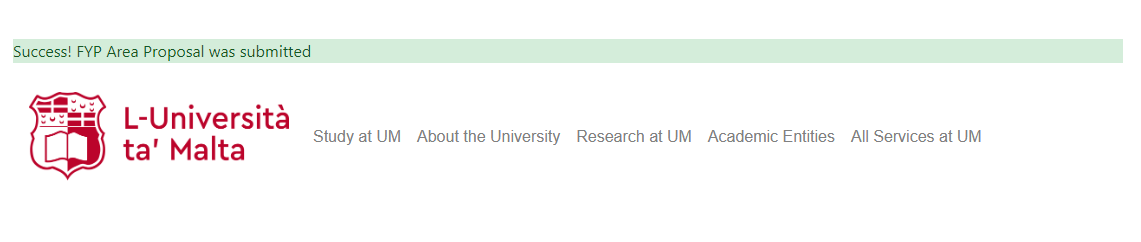


Figure : Notification of successful area upload

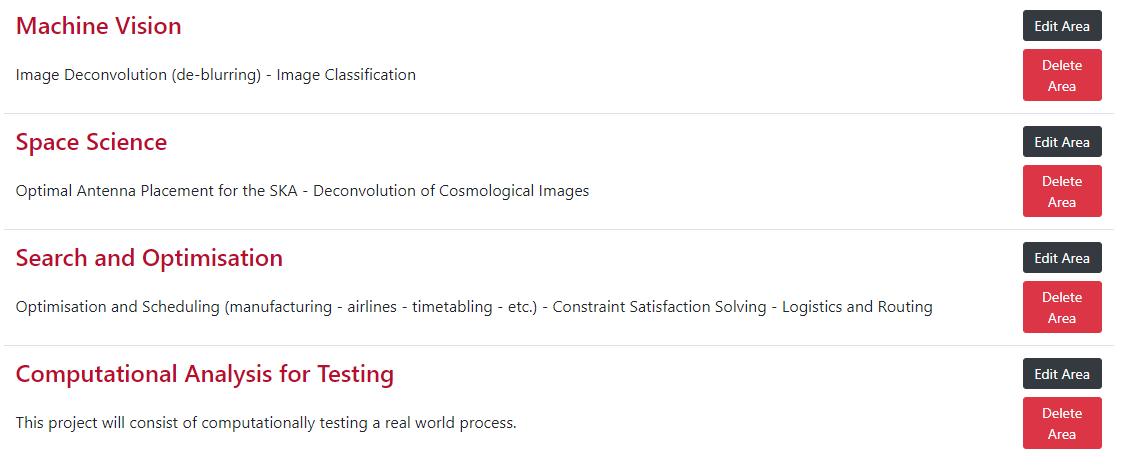


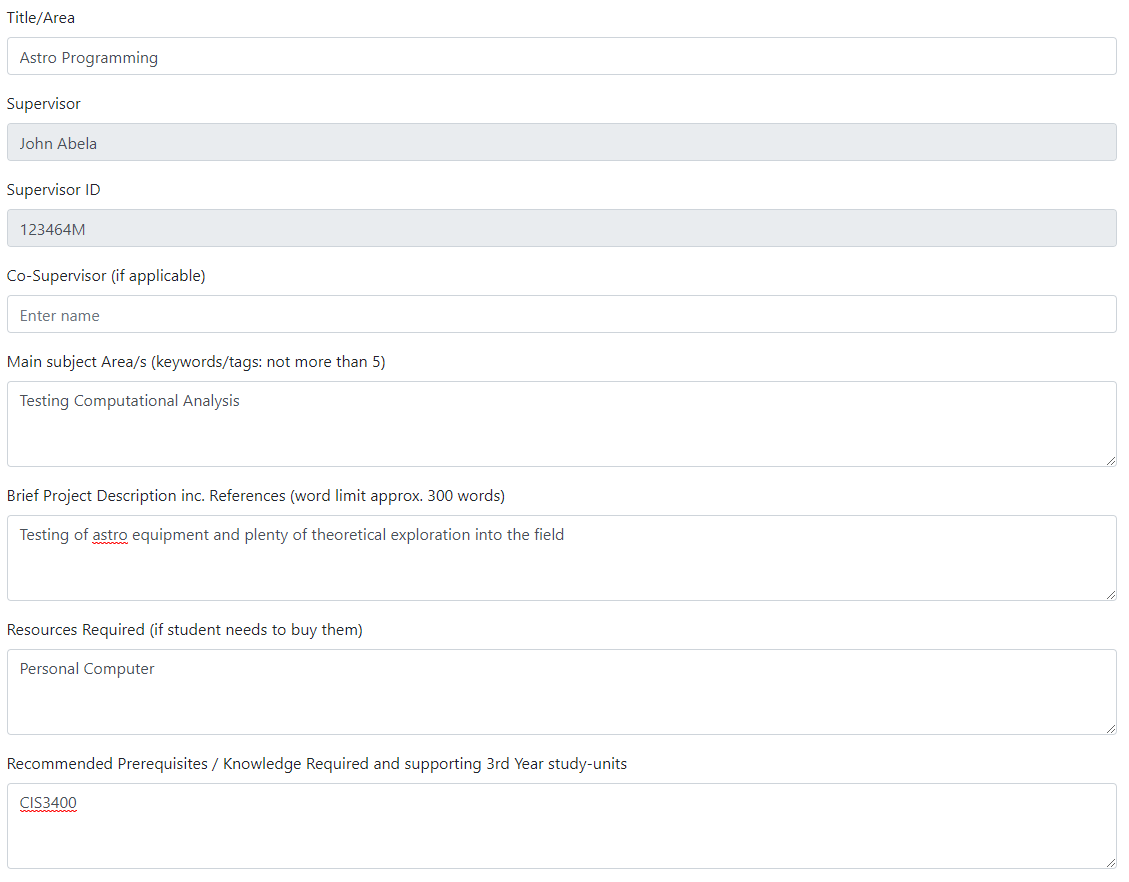
Figure : Evidence of the submitted area now forming part of the areas list for the given test user

Through the above display, it is possible to observe that the submitted area ‘Computational Analysis for Testing’ was successfully added to the list of areas for the test user. As a result, it may be stated that this test was complete successfully.

#### Test case 2: Editing a submitted area

With regards to this feature, users are to be allowed to edit submitted areas in the case of any amendment that may be required.

This test was carried out by means of editing the area ‘Computational Analysis for Testing’, having been used in the previous test.



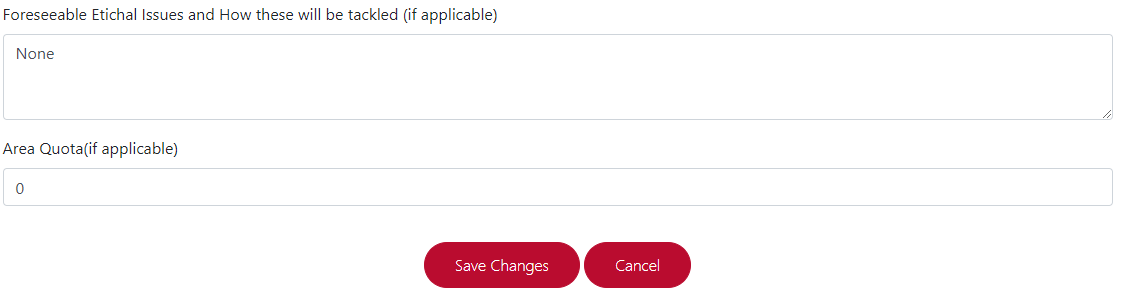


Figure : Edit area form for supervisors

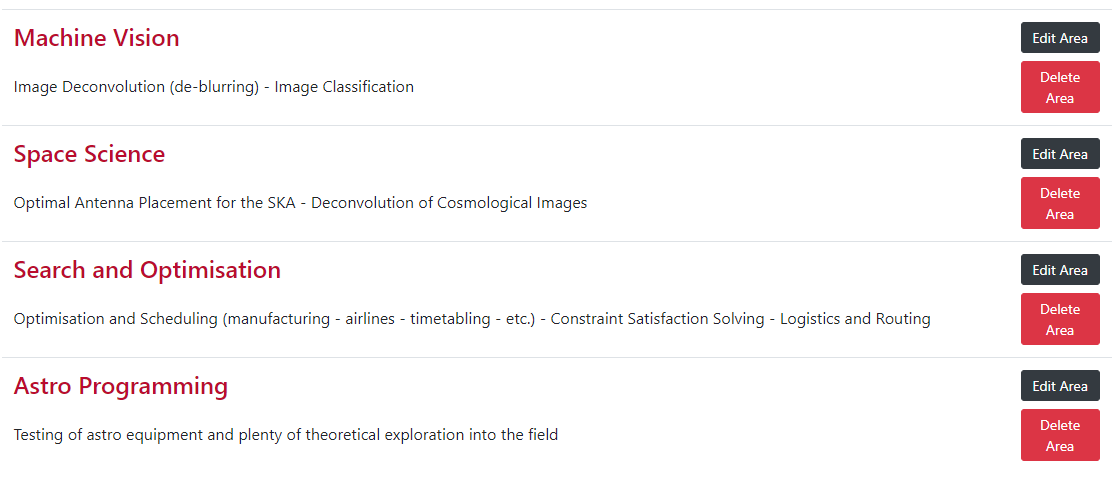


Figure : Evidence for changes made to the edited area

As may be observed, ‘Computational Analysis for Testing’ was changed to ‘Astro Programming’ through the edit form, having all submitted changed being saved. Seeing as the area had changes properly updated, this test may be considered as successful.

#### Test case 3: Deleting an area

For this test, the area submitted and edited during testing was to be deleted. This was to be done as means of showing the features available to a supervisor boasted completeness.

This feature was tested by means of selecting the ‘Delete Area’ area, and subsequently observing the list of areas to ensure that the given area for testing was no longer present.

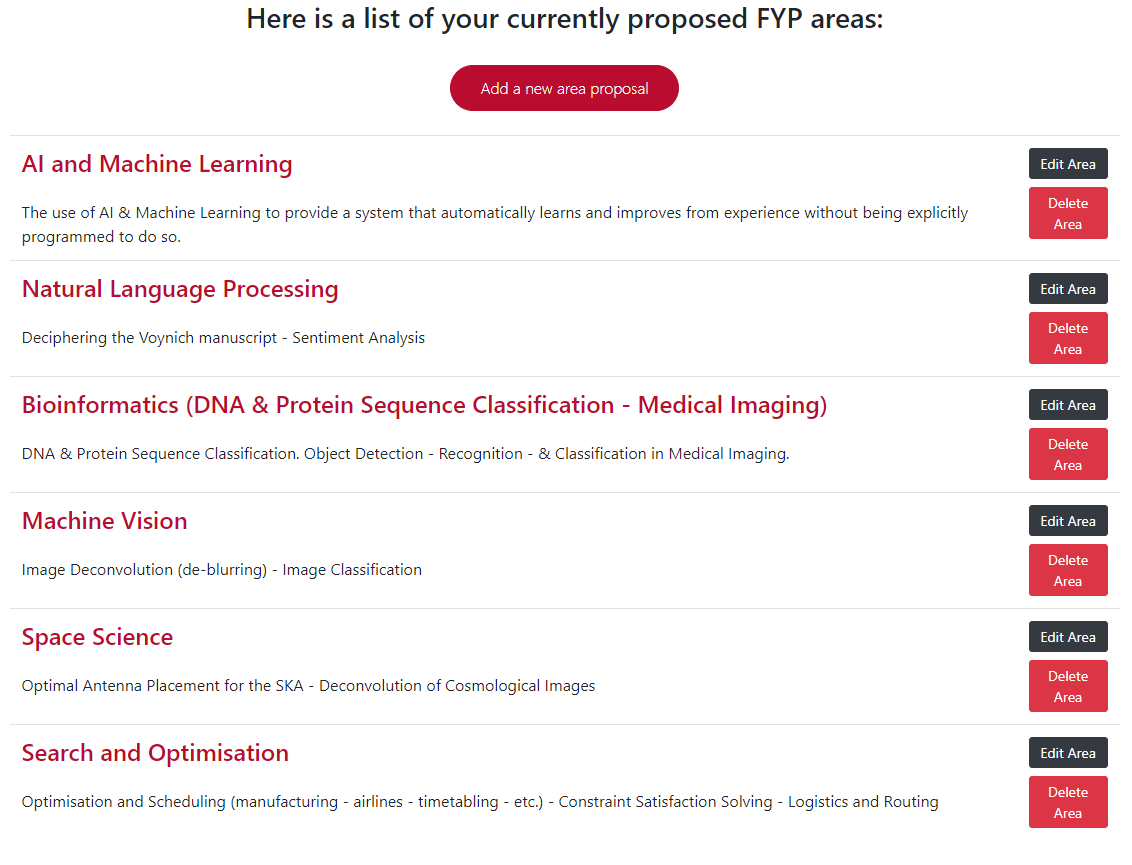


Figure : Full list of areas for user, showing that 'Astro Programming' has been removed

Through the above display, it is evident that the testing area ‘Astro Programming’ is no longer present. An apparent lack of presence serves as evidence of functionality for the delete feature, therefore, testing may be assumed to have been successful.

### 11.3.3. Exportation of details

Assumption: The main assumption made for this part of testing was to determine if the system was able to export relevant student and supervisor information that is used during allocation.

#### Test case 1: Exporting student data

With regards to this feature, the exportation of students was to be selected, to ensure that the correct file is downloaded.

This was to be done by means of selecting the ‘Export Students’ feature, commencing the download of a csv file as a result



Figure : Proof of download of csv file

The above download notification provided by the browser serves as evidence as to the download of the expected csv file. Due to the given result, testing may be considered successful.

#### Test case 2: Exporting supervisor data

For this feature, supervisor data was to be exported through the available ‘Export Supervisors’ feature.

Testing was carried out by selecting the feature mentioned and ensuring the commencement of download of a csv file containing supervisor data.



Figure : Evidence showing download of supervisor data as a csv file

As may be observed, the download of a csv file with relevant supervisor data commences successfully. As a result, testing for this feature is successful.

### 11.3.4. Performing Allocation

Assumption: Testing for this part of the system was prioritized as a means of being certain that the main goal of the system upheld complete working order.

#### Test case 1: Missing student submission

This instance of the performance of allocations consisted of attempting to perform allocations with a number of students having not yet submitted preferences.

Testing was done by accessing the ‘Perform Allocation’ page with test users being in the role of student having no preferences submitted.

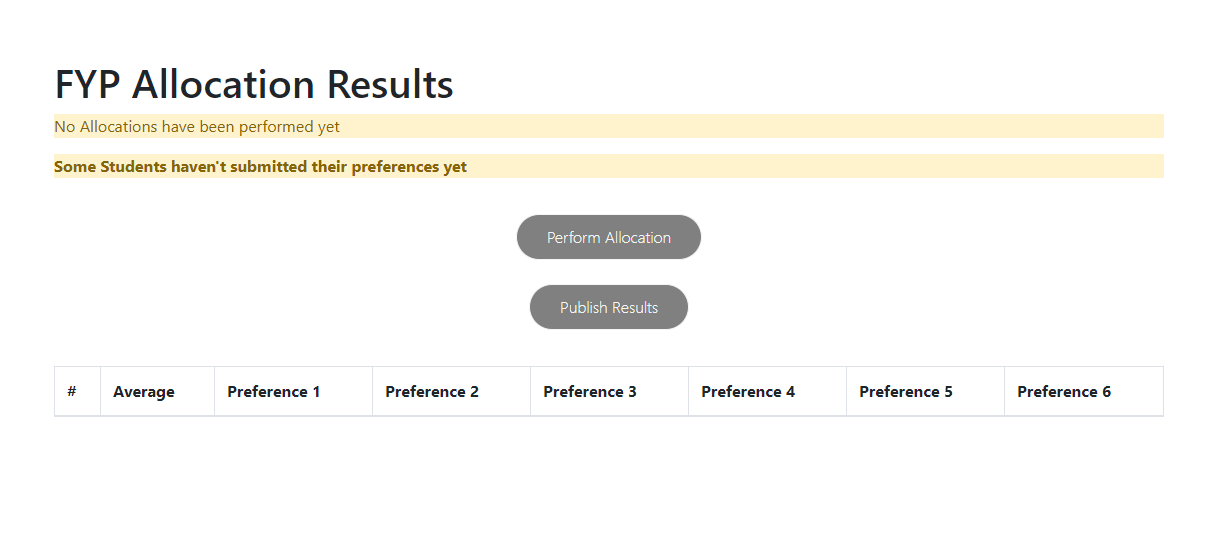


Figure : Some students having not yet submitted preferences

Upon entering the desired page, the ability to perform allocations is disabled as data would not be complete, having certain students with no preferences submission. By means of such measure taken to prevent allocation, testing may be assumed to have been carried out successfully.

#### Test case 2: All students having submitted preferences

This feature comprises of determining whether or not the ‘Perform Allocation’ button is enabled upon all students having submitted preferences.

Testing was done by accessing the ‘Perform Allocation’ page and observing whether or not the appropriate features were made available.

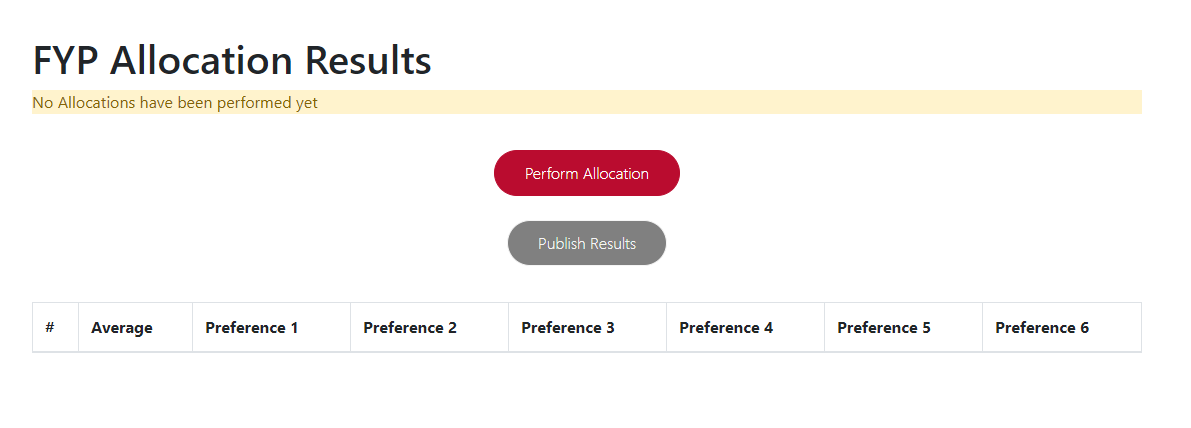


Figure : 'Perform Allocation' page upon all students submitting the preferences form

The ‘Perform Allocation’ button is enabled, with the ‘Publish Results’ button remaining disabled as no results are yet available. As this is the expected result of such a test case, testing has been completed successfully.

#### Test case 3: Performing allocation with student constraints

In order to be able to test the performance of allocations to the furthest extent, two students were given equal average marks as part of the test data. This allows for testing to ensure that equal marks are sorted by date of preferences submission .



Figure : Average Marks for test students ‘Joe Borg’ and ‘Ian Grech’



Figure : Time of submission for test student 'Joe Borg'



Figure : Time of submission for test student 'Ian Grech'

Testing was to be done by means of setting average marks for two given test students. These two students were to have equal average marks, being subsequently sorted by time of submission. These values may be observed in the above images, being extracts from the database.

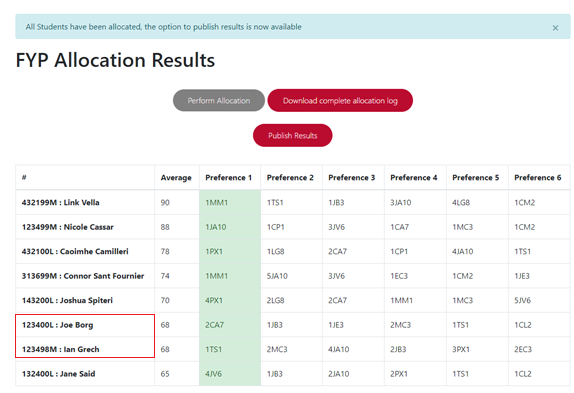


Figure : Allocation result with given test data

Firstly, performance of allocations was carried out successfully, with the provided test data forming the allocation result visible in the above display. Secondly, it may be observed that students with equal marks were sorted into the expected output, having ‘Joe Borg’ being allocated before ‘Ian Grech’, seeing as the former had an earlier submission time. Important to note is that a limitation of the system may be observable, which is to be discussed within the ‘Future Works’ section. Despite such, a valid output was produced as per expectations, and therefore, testing was completed successfully.

#### Test case 4: Performing allocation with supervisor and area constraints

The test to be carried out, consisted of performing allocations with further constraints on the data. These constraints were, allowing supervisors to accept only one student and to disable quotas for each area by setting them all to zero.

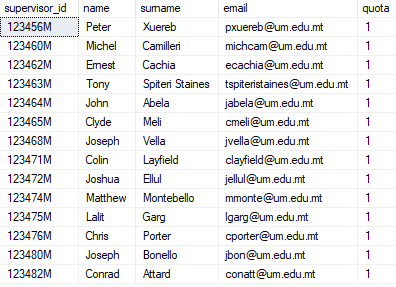


Figure : Setting all total quotas for supervisors to one.

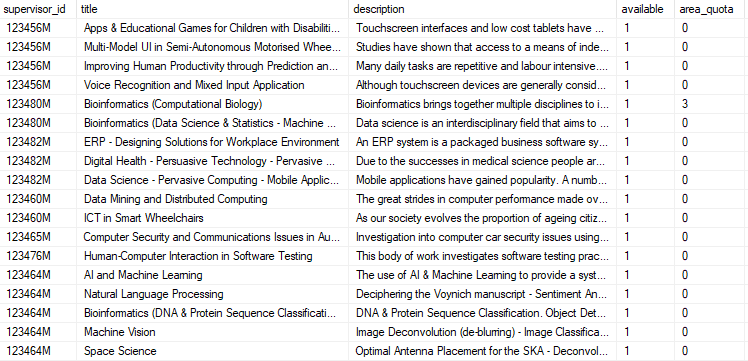


Figure : Disabling area quotas by setting them to zero

Allocation was performed with such additional stress in order to determine whether or not the system will be able to handle the increased chances of students not being allocated due to lack of availability, with regards to the preferences of any given student. Along with the above database samples, students were also given similar preferences as to raise the probability of a lack of availability during allocation.

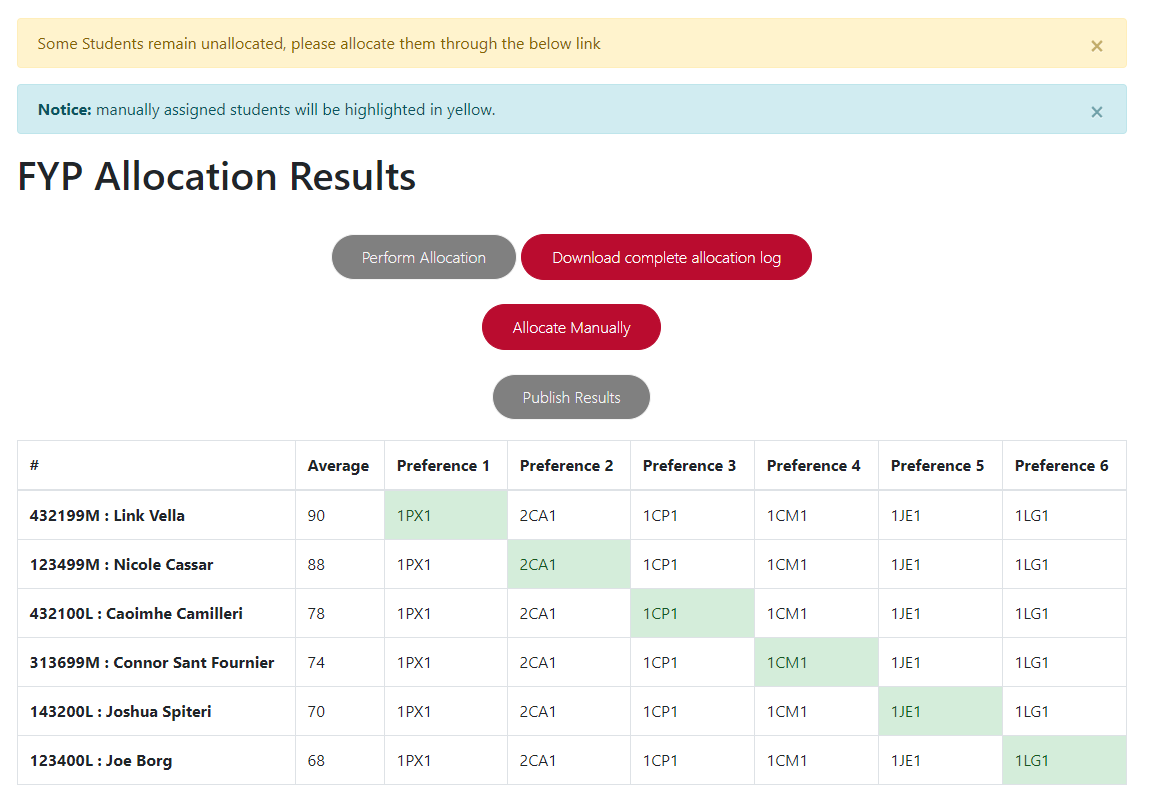


Figure : Performance of allocation, with results returning unallocated students

As was expected through such conditions, certain students remained unallocated. As a result, a notification was displayed, with the enabling of a manual allocation option as a means of allocating remaining students to available supervisors. Having this result be as expected, allows for the confirmation that testing was completed successfully.

### 11.3.5. Manual Allocation

Assumption: The main assumption to be taken is to ensure whether or not an administrative user is able to use the system in order to manually assign a student, having the given student remain unassigned following the initial allocation.

#### Test case: Manually allocating an unassigned student to an available supervisor

This part of testing serves the purpose of determining if the system can take the submitted student and supervisor and store them as new allocation within the database.

This is done by selecting from a list of unassigned students and available supervisors, subsequently submitting the allocation to be saved and displayed as part of the total allocation result.

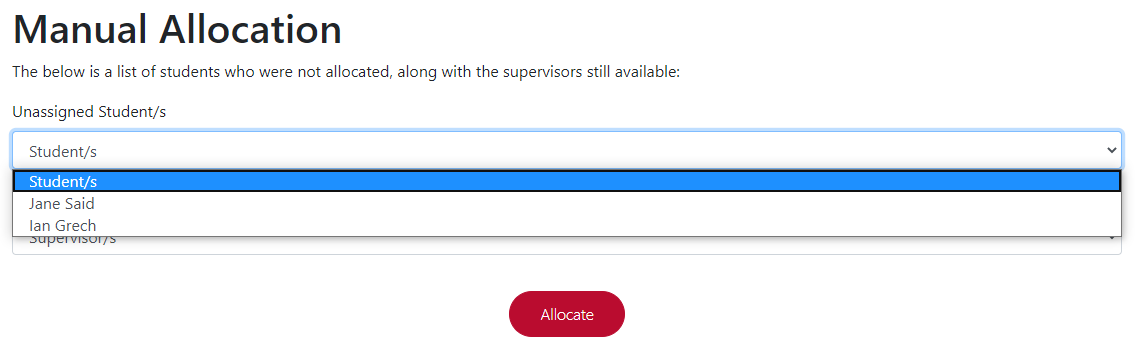


Figure : Choosing from list of unassigned students

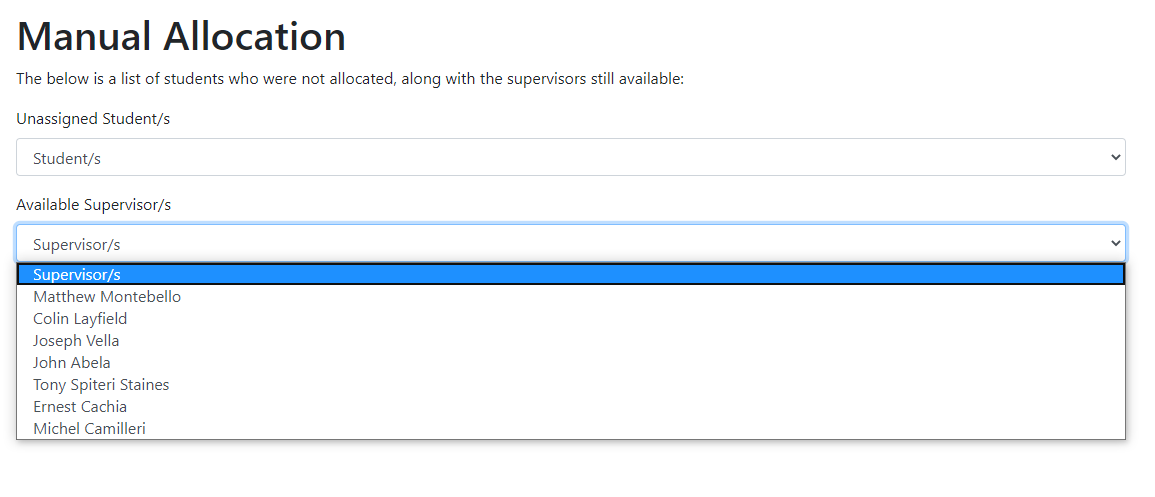


Figure : Choose from a list of supervisors still available

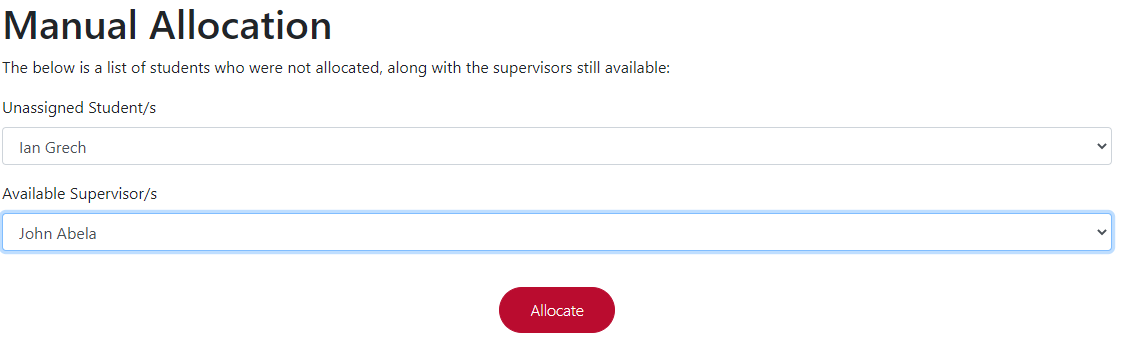


Figure : Selected options for manual allocation

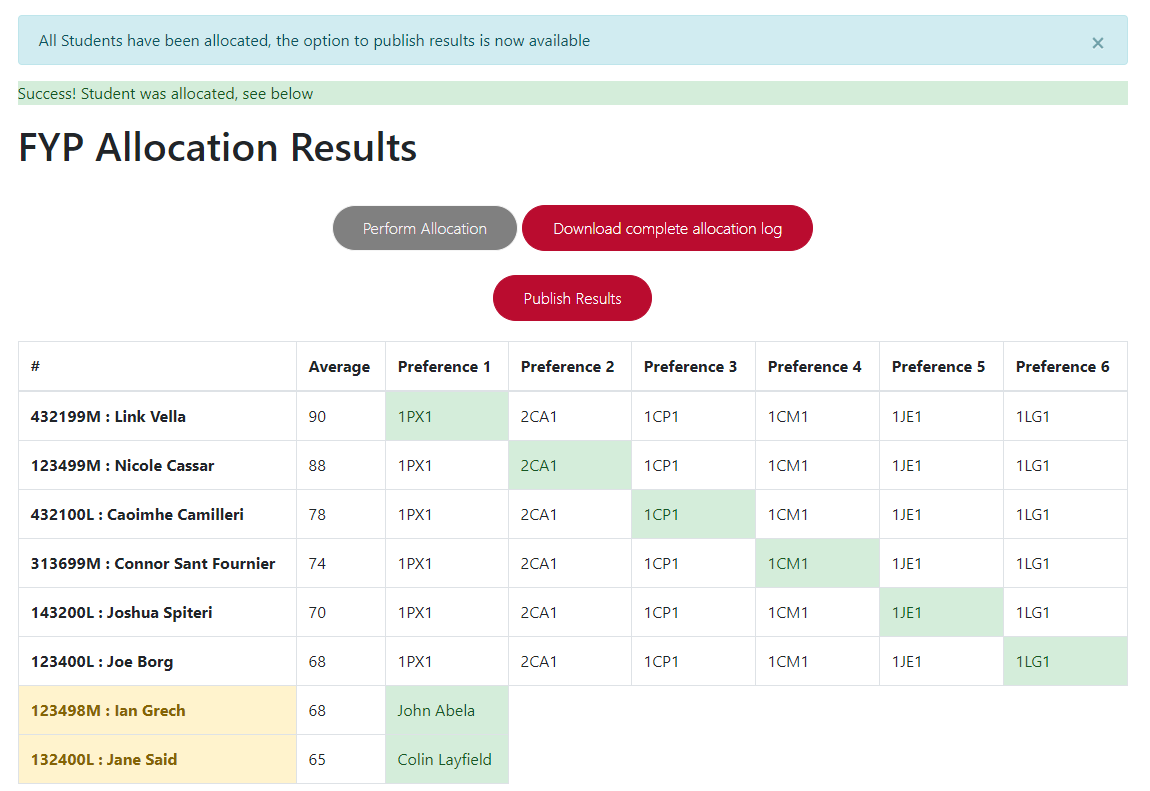


Figure : Students manually inserted highlighted with yellow

Following manual allocation, the output successfully stores and displays manually students with highlighting as a means of signifying manual allocation. This serves as proof of the feature performing as expected, and therefore, testing has been completed successfully.

### 11.3.6. Generation of allocation log

Assumption: This test is to be carried out as a means of ensuring that data generated and written to the allocation log correctly represents the allocation performed.

#### Test case 1: Allocation log for complete allocation

With regards to the allocation carried out in previous test case [Refer to section 11.3.4. Test Case 3], testing shall be carried out to determine if the allocation log generated conforms with the nature of the result outputted.

This will be done by means of downloading the allocation log for the allocation and observing contents in order to identify the expected result.

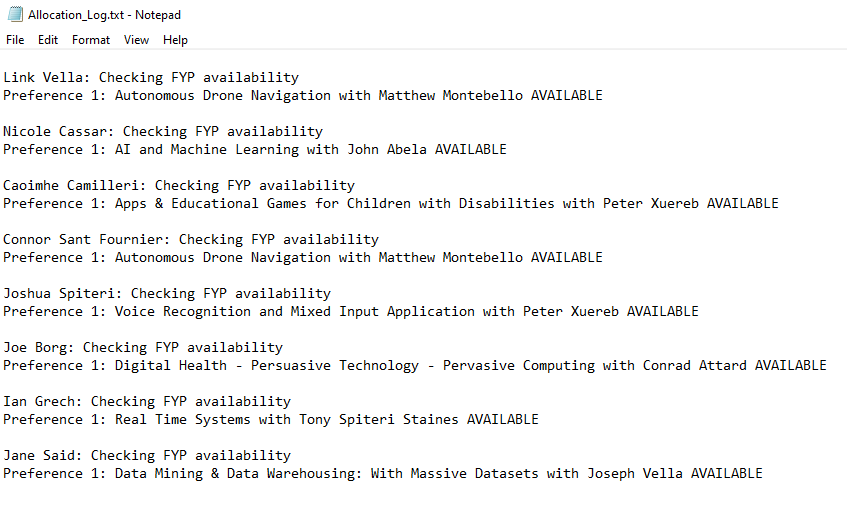


Figure : Allocation log for 11.3.4. Test Case 3

As anticipated the output generated matches the allocation result observable in *figure (SET THIS FIGURE NUMBER UPON COMPILING THE ENTRIE DOCUMENTATION) .* This output, matching the expected result, serves as sufficient proof that testing was completed successfully.

#### Test case 2: Allocation log in the case of student not being assigned

This test serves the purpose of ensuring that the system will be able to generate an allocation log entry that handles the lack of assignment for a given student.

In order to perform this test, the allocation log for allocation results found through the above test case [Refer to 11.3.4 Test Case 4], is to be downloaded. The instance of an unallocated student is to be observed and ensured of proper handling by the system.

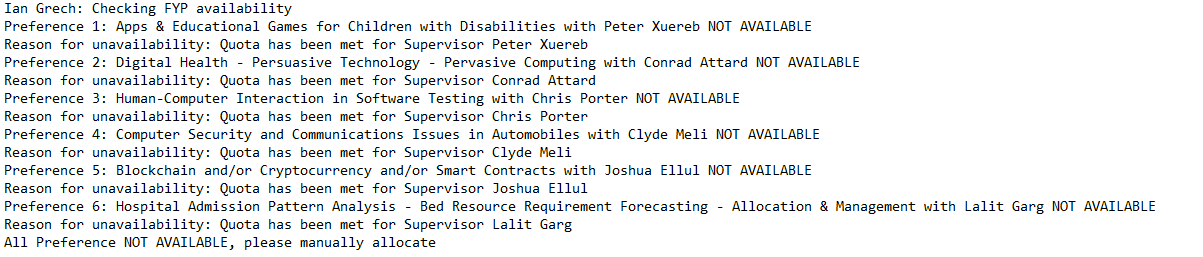


Figure : Allocation log entry for an unassigned student from allocation result 11.3.4 Test Case 4

In accordance with expectations, the generated log entry for an unassigned student shown through the above *figure 29*, handles each preference until none are left to check for availability. As a result, the log generates a message prompting no available supervisors and prompts the need for manual allocation. This output falls in line with expectations, and therefore, testing has been completed successfully.

### 11.3.7. Publishing of results

Assumption: This test aims to determine whether or not the email published by the system is received is correct, both in terms of format and data sent along with the email.

#### Test case 1: Publishing results to automatically allocated students

By means of this test, the email sent to students allocated by the system will be checked to have the correct format. Also, data sent with regards to supervisor and area assigned was checked for accuracy.

This was done by means of adding an existing email address as a test email and checking the received email is in accordance with expectations.

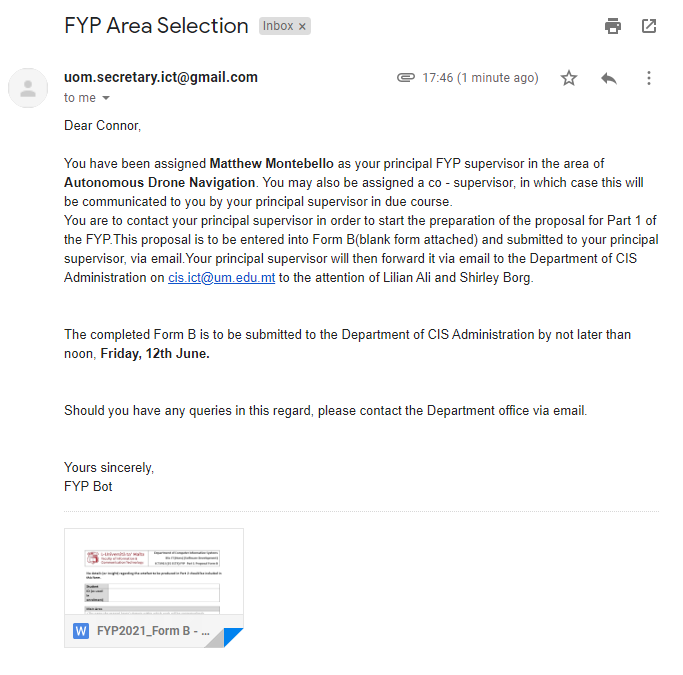


Figure : Email sent to students allocated by the system upon publishing results

As may be observed from the above email, when matched with the allocation performed in the above test case [Refer to 11.3.4. Test Case 3], the data used within the email is in proper accordance with the allocation result. As a result, testing has been successfully completed.

#### Test case 2: Publishing results to manually allocated students

This test is done to determine whether or not manually assigned students, receive a different email template due to the lack of availability for preferences.

An email is to be observed upon publishing results for the above allocation test case [Refer to 11.3.4. Test Case 4], being checked to ensure the template used was as expected.

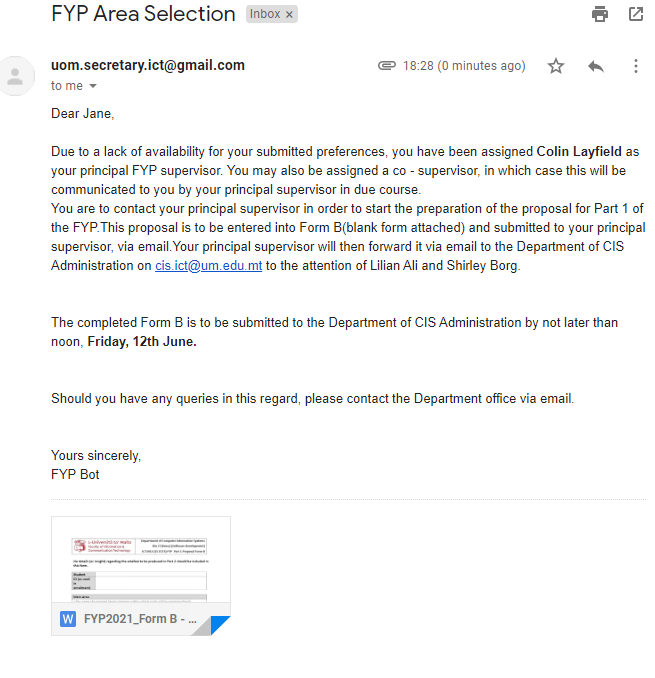


Figure : Email sent to manually allocated students upon publishing results

The template utilised within the above email, follows the specified template to be used with manually allocated student. Therefore, due to the expected result being achieved, testing has been completed successfully.

## 11.4. Results and Comparisons

After conducting various tests, with numerous test cases in each instance, it may be concluded that the system has strengths and limitations for each feature. Exploration of the features developed allowed for assurance of completion with regards to the expected results set towards the beginning of development. It is also agreed upon that testing the system has shown the degree of usability the system upholds, boasting ease of use across the system.

It is evident that certain features have been revealed to have limitations upon testing. These limitations will be recorded in the coming section and should be heavily considered as a means of furthering the system towards a state within which it may be integrated into the faculty of ICT for utilization.

Although testing served the purpose of demonstrating completeness for each feature, possibly unnoticed differences between the existing system and the newly developed system have been highlighted. These differences may include factors such as reliability and ease of use, along with understandability of the system.