|  |  |  |
| --- | --- | --- |
| Project titleTEd@S (The Education Administration System) | | |
| Student name | Student number | Supervisor |

# Project statement

Explain what your project is in 150 words or less

## Purpose of Application:

Integrated administration application that will allow staff with limited IT skills carry out user administration tasks. These include front end tasks such as timetabling, scheduling, attendance and performance tracking and background tasks such as server user account creation, folder creation and group and permission setting. The application will be designed to ensure data integrity, security and availability at all times.

## Case study:

Adult Education Centres

## Audience:

Administration staff

Tutors

## What they require:

Administration staff:

* Simple data input mechanism
* Secure storage of staff and student data
* Central, secure file storage
* Ability to generate timetables and schedules
* Ability to track student attendance
* Ability to track student progression
* Ability to produce reports regarding centre and student performance

Tutors:

* Accessing timetables
* Accessing student details
* Accessing attendance lists
* Adding student performance data

# What research has been done and what are the outputs?

## Background research

Small education centres and businesses to store and manipulate data belonging to their users for various different reasons. These users may be staff or external users such as students or customers. The challenge these organisations face is to being able to store this data in a secure manner while maintaining a high level of availability. Although they may have some IT infrastructures in place many of these organisations lack the knowledge and skills in-house to utilise these structures properly and achieve their IT goals. Due to cutbacks in funding over the past few years such centres have been forced to severely trim back their staffing levels to a point were some staff are carrying out duties which had been carried out by multiple members of staff in the past. Many administrative duties such as timetabling, scheduling, reporting and student/ tutor participation are carried out manually as it is felt there is a lack of affordable integrated software on the market [1].

The following are the eight golden rules of data protection [2]. Education centres need to store sensitive data regarding staff and students including amongst other sensitive data PPS numbers. Any application developed will need to adhere to these rules in order to ensure the centre remains compliant.

|  |
| --- |
| 1. [Obtain and process information fairly](http://www.dataprotection.ie/docs/A-Guide-for-Data-Contollers/696.htm#1)  2. [Keep it only for one or more specified, explicit and lawful purposes](http://www.dataprotection.ie/docs/A-Guide-for-Data-Contollers/696.htm#2)  3. [Use and disclose it only in ways compatible with these purposes](http://www.dataprotection.ie/docs/A-Guide-for-Data-Contollers/696.htm#3)  4. [Keep it safe and secure](http://www.dataprotection.ie/docs/A-Guide-for-Data-Contollers/696.htm#4)  5. [Keep it accurate, complete and up-to-date](http://www.dataprotection.ie/docs/A-Guide-for-Data-Contollers/696.htm#5)  6. [Ensure that it is adequate, relevant and not excessive](http://www.dataprotection.ie/docs/A-Guide-for-Data-Contollers/696.htm#6)  7. [Retain it for no longer than is necessary for the purpose or purposes](http://www.dataprotection.ie/docs/A-Guide-for-Data-Contollers/696.htm#7)  8. [Give a copy of his/her personal data to an individual, on request](http://www.dataprotection.ie/docs/A-Guide-for-Data-Contollers/696.htm#8) |

The centre chosen as a case study for this project has provided a useful insight to the struggles they face to in managing data and network resources. Due to financial constraints it has been left to tutoring/ administration staff and volunteers to carry out these duties. Considering the lack of time, resources and specialist skills available to the centre a gallant effort has been made in this respect. Having said that the collective feeling is that some new technology could help provide compliance with data protection rules, provide automation of existing tasks whilst improving the efficiency of carrying out others and providing automatic and manual network administration options.

## Alternative existing solutions to the problem you are solving

(Focussing on the architecture and user/functional requirements

As TEd@S is intended to be an all in one administration system that will incorporate functionality which might usually be provided in separate applications no one application reviewed provides all the functionality proposed in TEd@S and vice versa. Instead applications that provide much of the functionality proposed and those that provide specific functionality have been chosen. There are many different software tools on the market targeting the different academic markets. Through research it has been found such tools are quite limited in this country particularly those intended for the adult and community education market. Most applications on the market provide functionality more aimed at delivering course material rather than providing the resources to set up and manage courses.

In order to provide as wide a selection as possible of applications that provide similar functionality to that proposed for TEd@S applications have been split into categories depending on the functionality provided. Where possible trial versions of the different applications have been tested using Nielson’s Heuristics as a guide. Jakob Nielson proposes 10 general principles concerning user interaction design which are as follows

1. Visibility of system status

User should know where they are in the system and should be kept informed of what is going on.

1. Match between system and real world

The system should match how the user communicates and approaches tasks as much as possible

1. User control and freedom

Should provide easy to find undo and exit options for the user

1. Consistency and standards

Language and wording should be consistent throughout the system and comparable with is already available in the same category on peer systems

1. Error prevention

Should prevent errors as much as possible providing users with confirmation messages before completing actions

1. Recognition rather than recall

Options, objects and actions should be visible to the user. Help should be visible or easy to find

1. Flexibility and efficiency of use

System should be useable from a novice and an expert’s point of view. Expert users should be provided with options to accelerate use

1. Aesthetic and minimalist design

Only provide information that is needed. Keep as uncluttered as possible.

1. Help users recognise, diagnose and recover from error

Provide clear concise error messages and solution suggestions

1. Help and documentation

Should be easy to find, short and concise

Education Solutions

MIT’s suite of provides much of the functionality proposed in TEd@S albeit in a suite rather than one application. Unfortunately a trial couldn’t be secured for this product so no review was possible.

Sales Pulse

Adult education centres are required by “” to use the sales pulse adult education application. This is a web application which is cloud based with a client side written in HTML, ASP and JavaScript. It is mainly used by “” to collect information in order to sanction funding and tutor payments. The application allows the addition of course categories, courses, tutors, students and the allocation of students and tutors to specific courses. Timetables and reports can also be generated. [[1]](#endnote-1)

Once logged into SalesPulse the user interface is aligned to the left hand-side and appears cluttered and cramped with small font. Once the user has adjusted to the appearance the user interface is quiet intuitive and easy to use. SalesPulse Adult Education offers resources to users depending on which projects they are involved with. The main screen offers a choice of all projects across the top of the screen with access to those the user is offering courses in. On the main screen options for entering and retrieving data are provided depending on what project has been selected. IFrames provide forms for adding data and for presenting data retrieved from the database. a menu bar provides tabs for each sub category which when selected allows the user to tailor search queries and enter data depending on what the category is.

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

## Figure is a print screen of the interface provided when the learner tab has been selected. Numerous fields are provided enabling the user to search using different criteria. Learners can also be searched alphabetically by clicking the appropriate letter. Along the bottom of a screen a simple two line instruction is provided on how to use the various search options. Also provided on this page are options to add new learners and to add details of users who have enquired about future courses.

Once a learner has been selected many options are available to the user including adding the learner to a course, updating learner’s details viewing a learners profile amongst others. Within a learners profile the user can view and update information. Some options that are available but seem a little out of place are the options to transfer a learner to another course and to add a learner. It is not clear what the purpose of these two options is within the student profile and help is not readily available. Selecting the other menu options such as tutors and courses presents much the same options as those available in the learners sections just with different search fields. Once you get further into these sections the format is much the same as in the leaner section with different options and tasks available to the user

In general the help provided by this application is sufficient. Instructions on the various screens are quite limited although user guides are available for most sections and tasks on the main screen. The application is a little inconsistent in informing the user exactly where they are in the system. In some categories the user is kept informed at the top of the screen to exactly what section and subsection they are in. Other sections either don’t provide this at all or only in part. After a period of use it became clear the most efficient way to get back to the main menu which can involve up to three steps depending what section the user is in.

Although SalesPulse contains functionality that is the most comparable to that proposed in TEd@S and is designed for adult education centres it does present some shortcomings for centres that provide courses other than those that fall under the adult education umbrella. Data collected and stored by Sales pulse is relevant only to courses affiliated to “”. Timetables generated in this application are produced for tracking of course hours rather than as use as a reference for tutors and students. Reports generated in Sales Pulse are rendered in excel format only and do not represent all data required for end of year reports. Sales Pulse also does not allow for data integration and bulk data importation although these options have been explored in the past[ref sales pulse]

Moodle

Whilst developing a project idea providing functionality to facilitate the deliverance of course material in TEd@S was considered. This led to research of applications that currently provide this type of functionality. One of these is Moodle and even tough this type of functionality will not be included it was decided to include the review in this report as the application does provide some of the functionality proposed for TEd@S. Some of this functionality provided by Moodle which has been proposed for TEd@S includes:

* adding users
* groups
* courses
* adding tutors and students to courses
* adding grades
* reporting
* user administration
* backups

Moodle uses HTML, PHP and JavaScript to present content to users. Its main page is uncluttered with large font and provides all options available to the user on the left hand-side and any relevant content is displayed in the centre of the screen. Along the top of the screen is a status bar which keeps the user informed exactly what part of the system they are in at any given time. This type of layout is used throughout the application giving the user the impression that they it is the page content that is changing and not the actual page. This provides for easy navigation and clear system status but it does result in cluttered displays at in particular when generating reports.

Users can be added to Moodle either individually through a form or by bulk upload using csv delimited files. Categories and course can be added and updated through a course and category screen. This screen offers options to sort and move categories as well as adding courses to categories. In truth this screen is slightly confusing and requires some trial and error before figuring out what each option is for. A nice feature in this section is the ability to restore courses and categories from backups. Backups can be run manually or be configured to run automatically. The user can choose to back-up the whole of the application specific parts.

As a demo version of Moodle was used to review Moodle some of the criticism aimed at it may be unfair but there did seem to be a lack of easy accessible help although plenty of documentation is available from the Moodle website. All icons used throught Moodle also provides information regarding their use if hovered over. Error prevention is lacking slightly in places particularly when adding users. As stated previously TEd@S is not intended to be a Moodle type application but ideas on features and layouts have been provided by the application as well as how the application should perform overall.

E-Front

This application provides similar functionality to that available in Moodle and so only a short review has been carried out on it. This application provides a friendly user interface which incorporates images with text to present user options. Forms for adding content are also visually appealing and laid out simply.

|  |
| --- |
|  |

|  |
| --- |
|  |

During testing this application scored badly in some of the heuristics being used as a testing guide. Error prevention and detection is inconsistent throughout the application. Once a task has been completed either successfully or not there is no clear exit/ return path or progression path. An interesting feature throughout the application is the use of type ahead lists instead of drop dropdown lists for certain fields. Depending on the amount of options available this can be significantly more efficient. There is little or no help available in this application but as this is a demo version it may not be available.

|  |  |  |  |
| --- | --- | --- | --- |
| Heuristic | Moodle | SalesPulse | EFront |
| Visibility of system status | 10 | 8 | 7 |
| Match between system and real world | 7 | 8 | 8 |
| User control and freedom | 8 | 8 | 5.5 |
| Consistency and standards | 8 | 8 | 6 |
| Error prevention | 8 | 7 | 5 |
| Recognition rather than recall | 8 | 8 | 6.5 |
| Flexibility and efficiency of use | 8 | 8 | 6 |
| Aesthetic and minimalist design | 8 | 7.5 | 8.5 |
| Help users recognise, diagnose and recover from error | 8 | 7 | 5 |
| Help and documentation | 7[ii] | 9 | 0[iii] |

As it is proposed that TEd@S will provide automated network administration programs that already provide this functionality were investigated. Below is a table of applications available for Windows. No programs for Linux were found during this research through research of network administrator forums it seems scripts are the main method of carrying out such tasks.

## Technologies researched

What candidate technologies did you examine? What criteria did you use for selection? What experience you have in the chosen technologies? If new to you, what have you done to familiarise yourself with the new technology?

## Other relevant research done

Brainstorming Session

An initial brainstorming session was held with the centre coordinator and two members of staff who carry out both administrative and teaching duties. Some ideas were discussed as to what the application may include. The main points that came from the meeting are as follows:

* The centres current database was developed using legacy software which is now outdated
* Problems with data storage such as duplication, corruption and loss have occurred increasingly over the past few months
* It is feared requirements for data protection are not being met
* Data access is limited is limited as is how it is presented
* Many tasks are being carried out manually taking up staff hours which could be better spent on other duties
* There is no structured way to record and monitor student and tutor participation
* The same data is required to be entered for different organisations
* There is no way of tracking student progression
* There is a lack of knowledge regarding how to create and manage user accounts and how to create and manage folders and their permissions
* The centres IT resources are either being underutilised or not used correctly including:
* Duplication of files on the file server
* Files saved in incorrect folders
* Files being saved on PCs rather than the file server

Interview

The brainstorming session had provided an outline of what the user requirements for TEd@S might be. A set of questions was sent to the centres coordinator with the view of holding an interview a short time later. Through investigation and research a clear picture of what IT resources were available at the centre and how they were being utilised had been developed, clarification was now needed regarding data in particular the following:

* What data regarding what parties is being stored locally
* What data is required by third party organisations
* How was data was currently inputted and retrieved
* What reporting was required and how it was currently been carried out

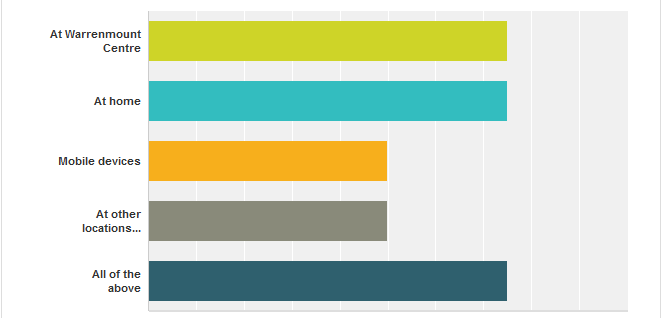
Clarification was also sought on what type of learners attended the centre with a particular focus on their IT literacy skills.

Surveys

The interview helped clarify issues surrounding data and eliminate some provisional requirement particularly those that were being considered for learners. Three separate surveys were released during the requirement gathering phase, the purpose of which was to further clarify what the user requirements should be. One survey was created for tutors, another for administration staff and a third designed for other centres that provide similar education services.

|  |
| --- |
|  |

Results from the tutors survey help clarify what features would be useful to them for course delivery. From the table above it is evident that there is support for many features laid out in a provisional list of functionality which was under consideration. Other suggestions were also provided through the survey such as the provision of individual learning plans for students and the ability for tutors to upload links to QA surveys.



The survey also provided an insight into where tutors would like to be able to use the application. This will aid the developed in deciding which frameworks should be used and what platforms it should be aimed at. Other questions were designed to ascertain how tutors currently carry out tasks such as roll call and the recording and tracking of student performance.

|  |
| --- |
|  |

The survey also provided an indication of the level of support for such an application and the willingness to utilise it if developed.

The second survey was designed to ascertain what features administration staff would find useful in an application. In hindsight due to the fact staff carries out different administrative tasks the survey could have been designed better to provide more informative results than what it did. The survey did provide useful feedback in relation to perceived strengths, weakness and difficulties associated with applications currently in use at the centre including SalesPulse which was reviewed previously.

|  |
| --- |
|  |

The table above represents the respondent’s opinion of the usefulness of timetabling functionality which will in turn provide the means for student and tutor participation tracking and reporting. The survey also confirmed that it is felt a student and tutors profiles would be made available which would contain personal information as well as any course participation records. It was suggested some sorted of unique identifier be assigned to users in order to eliminate duplicate eateries and the re-entry of data from archives.

|  |
| --- |
|  |

The table above represents the opinion of the respondents when asked if the ability to upload data in bulk using .csv files would be a useful piece of functionality.

The third survey was designed to get a general idea of what other centres were using to carry out their own data storage and handling tasks and to try a general opinion if they felt the functionality proposed in TEd@S would be useful. Although limited responses were received they served to confirm that most of applications used by other centres are bespoke owing to the fact that there are very few applications on the market which are affordable and provide the functionality required by such centres. Respondents also indicated that they felt a lot of functionality proposed would be useful or are already included in the applications they use.

As mentioned previously the survey didn’t provide all the necessary information needed to aid a final decision on requirements so a meeting was arranged a selection of administration staff and management. During this meeting the results of the different surveys were discussed and the level of support the proposed requirements received. The meeting also served to clear up what data was required by the centre itself and third parties such as funding bodies.

## Resultant findings/requirements

Carrying out research allowed the developer gain a clearer picture of the applications requirements and discount several requirements which had been proposed. From completed surveys student note provision and work collection gained favourable support. Through the meeting discussed previously and through discussions with a project supervisor it was decided that both these pieces of functionality would be left for future work. It was conveyed in the meeting that they wouldn’t be seen as a priority for the centre and that they would probably be underutilised due to the student’s low IT literacy levels. It was also suggested that they would probably form part of a separate application. Another proposed requirement was a reporting engine which would auto-generate the various reports required by the centre. Although it is felt that this would be a useful feature it was decided also to add this to future work or if time permits a final phase. It is felt it is more important to provide the correct data for the reports than concentrating on report design. Some suggestions for features were conveyed through the surveys one of which was an individual learning plan for students. Again it is felt although this would be useful it is not a priority and so has been added to future work and would more than likely be implemented in a separate application. The possibility of allowing students to request access to their student profiles for third parties such as potential employers or school registrars. Although this would probably be a good feature it has also been marked for future work.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Req ID | Name of Req | Description | Priority | User Contact | |
| 1 | **Secure Login** | **Login with provided user credentials. Log user visit details** | **High** | **All Users** |
| 2 | **Manual Entering of user data** | **Enter data relating to students or tutors either by manual insertion, bulk upload or restoring from an archive. If required accounts and or folders will be created for the user on the files server.** | **High** | **Administration staff** |
| 3 | **Upload data from files** | **Bulk upload user data from .csv file. If required accounts and or folders will be created for the user on the files server.** | **Medium** | **Administration staff** |
| 4 | **Restore user data from archive** | **Student and tutor data archived at the end of each semester. This feature will restore only required data. If required accounts and or folders will be created/ reactivated for the user on the file server.** | **High** | **Administration staff** |
| 5 | **Update, delete or archive student or staff data** | **Update, delete or archive student or staff data in the database and carry out changes on the file server where** necessary | **High** | **Administration staff** |
| 6 | **Add categories** | **Add new categories of courses for example ICT, literacy** | **High** | **Administration staff** |
| 7 | **Add new course** | **A new course must be added to existing category. One course can belong to different categories** | **High** | **Administration staff** |
| 8 | **Update delete courses and categories** | **Delete or Update course or category data and resources on file server if required** | **High** | **Administration staff** |
| 9 | **Add tutor to a course** | **Create a new instance of a course with a unique course code, tutor and running date range. Create folder on file server** | **High** | **Administration staff** |
| 10 | **Add a student to an instance of a course** | **Add a student to an instance of a tutors course. If necessary create a folder for the student on the file server** | **High** | **Administration and teaching staff** |
| 11 | **View student and tutor profile** | **Generate and view full profile belonging to tutors and students including any courses they are involved in** | **High** | **All users** |
| 12 | **Add, update, delete student performance data** | **Add, update or delete student attendance, results and performance data** | **High** | **Teaching staff** |
| 13 | **Create timetable** | **Select relevant data and create a timetable** | **High** | **Administration staff** |
| 14 | **Generate class schedule** | **From timetables generate daily class schedules including class lists specific to individual teaching staff** | **Medium** | **Teaching staff/ students** |
| 15 | **Network administration dashboard** | **Provide an interface to carry out basic network administration tasks such as creating folders, groups, changing file and folder permissions and backing up different parts of the network** | **Low** | **Administration staff** |

**Secure login**  
As the application will provide access to sensitive information users will be required to login with a unique username and password. Once authenticated a session key will be assigned which will remain valid until the user logs off. All logins and login attempts will be logged in as table for accountability purposes.

**Entering of user data**

If manually entering user information a form will be presented to the user with as many pre-populated and drop down fields as possible in order to allow them to enter user data. An option to restore user data from archives will also be available as will the ability to upload data from .csv files. A template will be downloadable to ensure the correct headings are utilised in order for data to be mapped to the correct fields. Depending on the users role within the organisation network resources such as accounts and folders will be created automatically on the file server. For each method of entering data is chosen the database will checked to ensure that the user hasn’t already been entered. The criteria that will uniquely identify a user have yet to be decided on. Once a user has been entered they will be assigned a unique identifier which they will keep throughout their time at the centre.

**Update, delete or archive student or staff data**

In accordance with data protection laws data is required to be kept up to date and only kept for as long as needed. The option to archive data will also be available. A log of any data manipulation will be stored in the database.

**Add, update or delete courses and categories**

This feature will allow for new categories and courses to be defined. A course can be assigned to one or more categories and tutor can be assigned to one or more courses. When a tutor is assigned to a course this becomes an instance of that course with its own course code and set running period. Students can then be assigned to these course instances. When adding, updating or deleting data these changes will also be reflected on the file server if necessary.

**Student and tutor profiles**

Student and tutor profiles will be created from the data stored. These profiles will include any personal details deemed necessary plus any academic data including any courses currently enrolled in or delivering, any courses completed or delivered in the past, attendance figures plus any results and performance related data. Access to these profiles will be administered depending on the users role within the organisation.

**Add, update or delete student performance data**

This feature will allow teaching staff to enter data such as test results and progression outcomes plus general notes regarding a student’s performance**.**

Create timetable

This feature will allow administration staff to generate timetables on a semester basis. When creating timetables lists of available courses, tutors and rooms will be provided. This function needs to be designed that no room or tutor can become double booked, ensure that rooms are only assigned to classes which contain the facilities that may be required. This feature must allow for updates.

**Generate class schedule**

This feature will provide members of teaching staff with a teaching schedule for the day once they have logged in. Upon clicking into an item of their schedule the user will be presented with any resources required for the specific class such as a list of class participants which is to be used to perform a roll call. Once uploaded to the server these lists will be used to store data concerning student and tutor attendance. In the case of students this data will be used to track their attendance and create alerts if required.

1. Bibliography (research sources)

# Analysis: Describe clearly what your solution will do

# Approach and Methodology

What is your approach to this project? Are you using any particular software methodology? Eg. Are you delivering design/ code in phases, or are you completing all design up front, followed by all coding? Have you some sections lower priority if time runs short?

# Design

## Technical architecture diagram:





## Other design documents

Insert other design artefacts that explain your system: e.g. Use cases/ ERDs/ Class diagrams

|  |
| --- |
|  |

|  |
| --- |
| **Use case number: 1** |
| **Use case name:** Add user data through form |
| **Intent:** To allow users to add staff or student data one by one using a form. |
| **Precondition:** User logged in as administration staff |
| **Use case initiation:** Selection of user administration tab |
| **Dialog (Description) :**   1. User: Selects tab for inserting individual user 2. System: Presents a form for data insertion 3. User: Completes form and submits 4. System: Reads in the form and performs checks to ensure user doesn’t already exist 5. System: Populates relevant tables 6. System: Creates accounts and folders on file server if nessesacary   **Alternate Flows:**  4a. System rejects form due to missing/ incorrect fields.  5a. System returns list of users which may be the same of that being entered. |
| **Use case termination:** user selects no when presented with option to enter another user |
| **Post Condition:** User is present in database and relevant network resources have been allocated |

|  |
| --- |
| **Use case number: 2** |
| **Use case name:** Upload user data in bulk |
| **Intent:** To allow users to upload user data from .csv files |
| **Precondition:** Use logged in as administration staff |
| **Use case initiation:** Selection of user administration tab |
| **Dialog (Description) :**   1. User: Selects upload from file 2. System: provides file selection menu which also contains a link to a template 3. User: Chooses a file to upload 4. System: Accepts the upload performs checks for duplication and populates relevant tables. 5. System: Creates accounts and folders on file server if nessesacary   **Alternate Flows:**  3a. User downloads template populates it, saves it then uploads  4a. File fails required field checks, user informed  4b. Possibility of duplication in the database highlighted to the user |
| **Use case termination:** User declines to add another user and returns to user admin screen . |
| **Post Condition:** **:** Users are present in database and relevant network resources have been allocated |

|  |
| --- |
| **Use case number: 3** |
| **Use case name:** Restore data from archive |
| **Intent:** To allow user data be restored from data archived |
| **Precondition:** Use logged in as administration staff |
| **Use case initiation:** Selection of user administration tab |
| **Dialog (Description) :**   1. User: Selects restore from archive 2. System: provides field for unique identifying number 3. User: Enters unique identifier 4. System: searches database for data relating to unique id. 5. System: Presents data to user 6. User: Confirms data is correct and submits 7. System: Enters data into relevant tables 8. System: Creates or reactivates any accounts or folders needed on the file server   **Alternate Flows:**  5a. Fails to find any data, informs user |
| **Use case termination:** User declines to add another user and returns to user admin screen . |
| **Post Condition:** **:** Users are present in database and relevant network resources have been allocated |

|  |
| --- |
| **Use case number: 4** |
| **Use case name:** Update, delete, archive user data |
| **Intent:** To allow user data to be changed, deleted or added to an archive |
| **Precondition:** Use logged in as administration staff |
| **Use case initiation:** Selection of user administration tab |
| **Dialog (Description) :**   1. User: Selects users then user category 2. System: Presents a list of user for that category 3. User: Selects relevant user 4. System: Presents user profile 5. User: performs desired operation on user profile 6. System: Carries out operation   **Alternate Flows:**  5a. Fails to find any data, informs user |
| **Use case termination:** user exits from user profile |
| **Post Condition:** **Database and file server updated** |

|  |
| --- |
| **Use case number: 5** |
| **Use case name: Add categories** |
| **Intent:** To allow new course categories to be added |
| **Precondition:** Use logged in as administration staff |
| **Use case initiation:** Selection of course administration tab |
| **Dialog (Description) :**   1. User: selects add category 2. System: Presents form to collect category details 3. User: Inputs relevant data and submits 4. System: Checks to ensure no other category exists which is the same 5. System: Add new category to the database and creates a folder on the file server 6. System: Confirms to the user that operation was successful and presents the option to add courses to the category 7. User: adds courses or exits   **Alternate Flows:**  5a. Finds duplicate category and informs user |
| **Use case termination:** user exits from course administration |
| **Post Condition:** Database contains new category data and folder exists on file server |

|  |
| --- |
| **Use case number: 6** |
| **Use case name: Add new course** |
| **Intent:** To allow new courses to be added |
| **Precondition:** Use logged in as administration staff |
| **Use case initiation:** Selection of course administration tab |
| **Dialog (Description) :**   1. User: Selects add course 2. System: Presents list of categories to which the course should be added 3. User: Inputs relevant data and submits 4. System: Checks to ensure no other course exists which is the same 5. System: Add new course to the database and creates a folder on the file server 6. System: Confirms to the user that operation was successful   **Alternate Flows:**  5a. Finds duplicate course and informs user |
| **Use case termination:** User exits from course administration |
| **Post Condition:** Database contains new course data and folder exists on file server |

|  |
| --- |
| **Use case number: 6** |
| **Use case name: Update, delete course or categories** |
| **Intent:** To allow courses and or categories be updated or deleted |
| **Precondition:** Use logged in as administration staff |
| **Use case initiation:** Selection of course administration tab |
| **Dialog (Description) :**   1. User: Selects manage courses and categories 2. System: Presents list of categories and list of courses associated with it as well as any options available 3. User: Selects relevant options 4. System: Performs selected actions and updates database and file server where necessary 5. System: Confirms to the user that operation was successful   **Alternate Flows:**  4a. If a category is to be deleted the list of associated courses presented to allow the user reassign them to a category from a list also provided |
| **Use case termination:** User exits from course administration |
| **Post Condition:** Database contains new course data and folder exists on file server |

|  |
| --- |
| **Use case number: 7** |
| **Use case name: Assign a tutor to a course** |
| **Intent:** To allow an instance of a course to be instantiated with a tutor and date range |
| **Precondition:** Use logged in as administration staff |
| **Use case initiation:** Selection of course administration tab |
| **Dialog (Description) :**   1. User: Selects manage courses and categories 2. User: Selects manage courses 3. System: Presents list of courses and tutors and date ranges 4. User: Selects relevant course and tutor option and date range 5. System checks to ensure the same entry doesn’t already exist 6. System: Creates entry in database 7. System: Creates a folder within the file server name using course name, tutor name and date range 8. System: Confirms to the user that operation was successful   **Alternate Flows:**  5a. User informed entry already exists presented with the option to select another |
| **Use case termination:** User exits from course administration |
| **Post Condition:** Database contains data for new course instance and a folder exists on file server |

|  |
| --- |
| **Use case number: 7** |
| **Use case name: Assign a tutor to a course** |
| **Intent:** To allow an instance of a course to be instantiated with a tutor and date range |
| **Precondition:** Use logged in as administration staff |
| **Use case initiation:** Selection of course administration tab |
| **Dialog (Description) :**   1. User: Selects manage courses and categories 2. User: Selects manage courses 3. System: Presents list of courses and tutors and date ranges 4. User: Selects relevant course and tutor option and date range 5. System checks to ensure the same entry doesn’t already exist 6. System: Creates entry in database 7. System: Creates a folder within the file server name using course name, tutor name and date range 8. System: Confirms to the user that operation was successful   **Alternate Flows:**  5a. User informed entry already exists presented with the option to select another |
| **Use case termination:** User exits from course administration |
| **Post Condition:** Database contains data for new course instance and a folder exists on file server |

# Prototyping and Development

Explain exactly what prototyping and development you have completed

# Testing

Explain your planned testing approach: For example: who will be involved, what test scripts are planned, how will the testing be executed.

# Issues and risks

Explain the main issues / challenges that are unresolved on your project. – and your suggested approach to solving them. This is a critical part of your report to show that you understand what is required to complete the project.

# Plan and future work

What are the key deliverables and date for the remainder of the project?

# Conclusions

Identify interim conclusions viz. summary of findings thus far, plausibility of the proposed system and personal development conclusions.

1. Limited to what courses are required by “” [↑](#endnote-ref-1)