Nicky Temperley

Nicky Temperley

E6155906

TMA02



Table of Contents

[A Lone Worker System to Improve Staff Safety 2](#_Toc6331997)

[1. Preparation and Planning 2](#_Toc6331998)

[1.1 Title and Scope 2](#_Toc6331999)

[1.2 Resources 5](#_Toc6332000)

[1.3 Future Plan 5](#_Toc6332001)

[2. Project Work Completed 7](#_Toc6332002)

[2.1 Informationsources 7](#_Toc6332003)

[2.2 Project Work 7](#_Toc6332004)

[3. Review and Reflection 17](#_Toc6332005)

[3.1 Ways of Working 17](#_Toc6332006)

[3.2 Evaluating Project Management 17](#_Toc6332007)

[3.3 Legal, social, ethical and professional issues 18](#_Toc6332008)

[4. References 20](#_Toc6332009)

[5. Appendices 21](#_Toc6332010)

[Appendix 5.1 – A screenshot showing a solution employed by a neighbouring Council 22](#_Toc6332011)

[Appendix 5.2 – The initial enquiry email from the client 23](#_Toc6332012)

[Appendix 5.3 – Initial Meeting with the Client 25](#_Toc6332013)

[Appendix 5.4 – My Interpretation of the main user interface 27](#_Toc6332014)

[Appendix 5.5 – Staff details collection form 28](#_Toc6332015)

[Appendix 5.6 – Proposed designs for the Java GUI 29](#_Toc6332016)

[Appendix 5.7 – Agreed designs for the Java GUI 34](#_Toc6332017)

[Appendix 5.8 – Agreed designs for the Mobile App 36](#_Toc6332018)

[Appendix 5.9 – Downloads used 37](#_Toc6332019)

[Appendix 5.10 – JSwing pages 38](#_Toc6332020)

[Appendix 5.11 – Java Code 44](#_Toc6332021)

[Appendix 5.12 - Tutor Contact Log 59](#_Toc6332022)

[Appendix 5.13 – Project Log 60](#_Toc6332023)

[Definitions 61](#_Toc6332024)

# A Lone Worker System to Improve Staff Safety

|  |  |
| --- | --- |
| **Learning outcomes** | **Mark**  (out of 20) |
| You should be able to develop and demonstrate the ability to:  - LO2. identify and refine the goals and content of your project;  - LO10. identify and address the legal, social, ethical and professional issues that may arise during the development and use of computing and IT systems.  **Comment:**  Goals refinement (LO2):-  Describes and justifies several refinements to the project aims and objectives.  Legal, social, Ethical and professional issues (IT systems) (LO10):-  Comprehensively identifies the relevant LSEPIs and reflects on how they relate to their project work and their professional behaviour as they work on this project. | 18 |
| You should be able to develop and demonstrate the ability to:  - LO4. gather, analyse and evaluate relevant information to complete the project successfully;  - LO6. make effective use of a variety of information sources including the internet, demonstrating awareness of the credibility of the source; and  - LO7. communicate information, ideas, problems and solutions clearly  **Comment:**  Analyse/Evaluate all relevant information (LO4&6) :-  You find relevant material, from a range of sources, clearly showing an understanding of the credibility and significance of the source of information.  Communication (LO7):-  You present effectively and succinctly how you have and can contribute to the work to be done. A very clear, readable piece of work. You make good use of language, diagrams, tables, etc. where necessary. | 18 |
| You should be able to:  - LO1. demonstrate and apply a systematic understanding of the fundamental technical concepts and principles relevant to your project  - LO11 analyse a practical problem and devise and implement a solution, building on the knowledge and skills developed throughout your earlier OU studies.    **Comment:**  Technical underpinning (LO1):-  You make use of several concepts and principles from the literature and/or prior OU study.  Problem identification and on-going solving (LO11):-  You successfully demonstrate work towards a project that addresses one or more of the core aspects of the problem and its solution. | 18 |
| LO8. You should learn independently and reflect on what has been done, with a view to improving skills knowledge.    **Comment:**  Independent learning, self-evaluation and improving (LO8):-  A clear and purposeful reflection on work achieved so far and/or on things that impeded your work was presented. | 18 |
| You should be able to develop and demonstrate the ability to:  - LO3. identify, list and justify the resources, skills and activities needed to carry out the project successfully;  - LO9. plan and organise your project work appropriately, and keep systematic records of plans, progress.  **Comment:**  Resource/Skills planning & management (LO3):-  You have kept good control over the resources that you have access to and that are needed for your project. You also managed well the risks as they presented.  Project work planning and monitoring (LO09):-  Honest and accurate review of own work was shown. You also recognise what you have/have not yet achieved and the reasons behind not being able to do certain things so far. | 18 |

**Overall mark = Total of learning outcome marks = 90%**

## 1. Preparation and Planning

### 1.1 Title and Scope

#### 1.1.1 Background

I have been working for Northumberland County Council for 9 years now, and within it there are over 6000 employees across dozens of departments, each providing a role to the Council and the Community. One of the departments in the council is the planning department, who receive planning applications.

Planning is a customer facing Service. They deal with Planning applications from members of the Public, land owners and Developers. The type of planning applications can vary from small scale such as a simple householder extension, through to large scale residential developments, new commercial developments, opencast sites, etc. Planning Officers are given 'delegated authority' by the Chief Planning Officer which means they have authority to determine planning applications. Most applications are determined by a Planning Officer however some less straightforward or contentious applications may be referred to a Planning committee.

Planning also write Policies through consultation to guide future development of the County. This sets the framework for all planning officers to adhere to for all future applications for developments of towns and villages within the County.

Conservation areas also fall under the remit of Planning. The Conservation officers may be involved in a planning application to give opinion and guidance to a planning officer where a planning application/development is in a Conservation area. Likewise advise is also provided to residents who may wish to make changes to a historic building such as a grade II listed building.

Planning also deal with a number of planning breaches. This may be that building works are carried out without planning permission or not in compliance with their approved planning approval. The Enforcement officers will also get involved in reports from members of the public of issues such as untidy land, non-approved advertising, etc.

#### 1.1.2 The Problem

The planning department have about 100 users who work out on site. Fiona Charlton, the client, is looking for the development of a system that will allow her and other managers to record which officers are on site, which site they are at, the time that they arrived and the time they estimated they will leave. The officer will time to leave the site, and if they haven’t checked in when they leave or before this time an alert needs to be sent to their line manager. See [Appendix 5.2](#_Appendix_2_–) for a brief description in the client’s words.

Enrich the body of the report by taking quotes and using them here.

Currently there is no standardised reporting procedures for this to happen, and all teams use a different way of recording the information. There is a system already available called Guardian24, but at over £5 per user per month, this isn’t a viable solution for the whole workforce and is only used for the most at-risk officers. Without a standardised solution it could be the case that a site officer has an accident and isn’t able to contact the office, and no-one would be aware that there was a problem.

The officers will need to be able to submit the check in and out times from their mobile phone, and the system must be able to handle a manual input by office staff in the case they are unable to use their phone for whatever reason.

Clear enough description of the problem, use it in the final version too.

#### 1.1.3 My Proposed Solution

To solve the problem, I am planning on designing and implementing a system to record all the information required. I will use a Java GUI to make calls the SQL to the database, with a mobile app created in Cordova using HTML, CSS and JavaScript to communicate via a REST web service to the database.

The Java system will record the managers details, the officer’s details, the estimated and actual check in and out times (with the actual times being submitted through the mobile app or manually entered), the site location and any notes required for the site. Officers, or admin staff on their behalf, will selected the officer’s name, manually enter a site location, then enter estimated check in and out times for when they arrive and leave site. The actual times will be entered later through the mobile app. The GUI will check for users that have checked in at a site and haven’t checked out by the time they have estimated; in this instance, an alert will show on every running instance of the system. This will be a prompt for a manager to contact them and take further action if needed or update the system on their behalf.

The mobile interface will be simple, with only a dropdown box to select the site, and buttons for check in and check out. The user will be authenticated by the Google account that they are logged into on the phone, which is linked to their staff domain account, and the site dropdown will only show sites that they have previously typed into the system.

You may wish to create a feature where sites are already entered from the backend after allocation.

#### 1.1.4 Alternatives to my proposal

|  |  |
| --- | --- |
| **Alternative** | **Justification** |
| Carry on the way they are currently working | Not really an option as it could be a health and safety risk if site operatives have an accident on site and no-one notices that they haven’t checked in |
| Use the app Guardian24 for ever officer | Not feasible due to the cost, it would work out over £500 per month |
| Ask a third party to develop a solution | Another cost implication, high initial cost although there shouldn’t be a monthly cost apart from maybe an annual licensing fee |
| Develop a system with different technologies to the one proposed | This is achievable, and up until recently it was a strong possibility. I have decided against this as Java is my strongest language and I feel I could deliver the best results with a language I am comfortable, rather than trying to learn a new language to achieve a suitable outcome. |
| The development team at Northumberland County Council to develop a solution | The development team are unable to develop a solution as it is outside the scope of the services they provide to the council. Resources may become available later, but currently this isn’t an option |

#### 1.1.5 Associated Risks

Below is a list of all risks that may occur, with their likelihood, their impact on my project, and my efforts to mitigate them:

|  |  |  |  |
| --- | --- | --- | --- |
| **Risk** | **Impact** | **Chance of Occurrence** | **Mitigation** |
| Finished product does not work as intended | M | M | I can only try as hard as I can to complete the project. If there is a part that doesn’t work I’ll document what it is, and why it doesn’t work, and try and find another method to achieve the same result. |
| Client rejects the finished product | H | L | Meet with the client at each step to make sure that the product meets their expectations |
| Users are unable to access through mobile | L | M | This would only happen if they were in an area with low internet signal. In this case, if they ring the office staff can update the system on their behalf |
| Project becomes unmanageable | H | L | I have defined the scope to be within my control. If anything unexpected comes up, I’ll speak to the client and come to a mutual arrangement so the project can be completed on time |
| Lack of co-operation from users | H | L | The client is trying to get approval from management to make it a disciplinary action if they don’t use the system as intended |
| Unable to complete project in the timeframe given | L | M | As mentioned earlier, I’ll speak with the client to see if we can reduce the functionality, which can then be included in a future iteration. |
| Change in hardware available | L | L | There are no plans to change the hardware used at Northumberland County Council. We have just recently deployed new equipment which should be suitable for at least the next 10 years |

Risk assessment has been done well.

#### 1.1.6 My Lifecycle Model

In TMA01 I compared several lifecycle models, and in the end, I decided that an iterative model would work best for this project. After completing the first iteration, I am convinced that that was the correct choice for me, it allowed me to build earlier working programs for the client to use and test, and it allowed any problems in the design to be identified early so they can be rectified before going on to the next iteration.

### 1.2 Resources

For this project, I anticipated I will need the following resources:

|  |  |  |
| --- | --- | --- |
| **Resource** | **Why is it needed? / What does it do?** | **How to acquire** |
| Java v8u11 | Enables use of the Java programming language, and the ability to compile/run programs | Downloaded for free from the Oracle website. |
| Netbeans v8.2 | An IDE – for writing the code | Included with the above download. |
| DB Browser for SQLite | Software to allow me to create a database for my GUI to connect to, as well as seeing the state of the database at any given time. | Downloaded from the portable apps website. |
| Java ODBC Drivers v3.27.2.1 | To allow my Java code to communicate with the SQLite database. | Downloaded from a BitBucket repository, as informed by |
| Data from users | This will be collected by sending the form in Appendix 5.5 to each of the users intending to use the system. For the purpose of this iteration, I have generated random information from [Mockaroo](https://mockaroo.com/) to build my database | Form has been sent to users to complete for next iteration. Random test data was created using the website GUI. Any users that have responded will be prompted by their manager to fill it in |
| Microsoft Forms | To create the form to collect the data. MS Forms is part of the Office 365 suite, which I have access to as both part of my Student Membership and my work account. | The simple interface is used to create the form, with several options available for each question type |
| UMLet | My chosen application for drawing and editing UML diagrams. | Downloaded from the UMLet website |

Please see [Appendix 5.9](#_Appendix_5.9_–) for more details

### 1.3 Future Plan

At this point in my project, I have partially implemented a working GUI and have added functionality to add edit and delete officers and managers. It seems that when I try to delete a manager, the database is locking, and I am unable to proceed past this point; after further investigation I think that there is an open connection from another method that is causing this, so my first port of call for TMA03 is to find this error and rectify it. After this I will finish the other edit and delete methods for both officers and managers. Overall though, I’m happy with my progress so far.

You may wish to add screen shots of what has been achieved within the main body of the report to enhance the readers experience of the report.

The second iteration will see me creating a mobile app for users, which will cover UC9, UC10 and UC11 ([see 'The project so far'](#_2.2.1_The_project)). This will allow them to check in and out when they arrive and leave sites, allowing office staff to see how they are progressing in their day, and if any emergencies come up. The mobile app will authenticate against the Google profile that the user has logged in, so all the user needs to do is select the site they are at, click check in when they arrive at the site, and click check out again when they leave. I already have a design approved ([see appendix 5.8](#_Appendix_5.8_–)) by the client; so, it’s just a case of implementing it.

The resources I think I’ll need for this iteration are:

1. Cordova – a cross platform mobile app creator. I can create the app using HTML, CSS and JavaScript and Cordova will translate this into the code needed to run on a specified OS, in this case, Android.
2. A mobile phone so I test that the app runs as expected.
3. The form ([see appendix 5.5](#_Appendix_5.5_–)) completed from around 5 managers and 20 officers so I can use some real data in the system
4. Clarification of what data is stored about the sites, or is it just a site name? Will arrange another meeting with the client to discuss this and anything else that comes up in the meantime.
5. A list of around 30 active sites, to use for testing
6. A Rest Service to collect the location and pass the information to the database

The rough timeline for implementing the app is:

|  |  |
| --- | --- |
| w/c | Activity |
| 22/04/2019 | Correct the coding in the GUI to allow the edit and delete queries to function correctly |
| 29/04/2019 | As above |
| 06/05/2019 | Break due to other commitments |
| 13/05/2019 | Create the app in Cordova, and get the code looking correct in HTML and CSS |
| 20/05/2019 | Start TMA02 and write up my project work so far |
| 27/05/2019 | Design the Restful Web Service |
| 03/06/2019 | Incorporate the web service into the app |
| 10/06/2019 | Break due to other commitments |
| 17/06/2019 | Incorporate the submitted times into the GUI, and allow them to be edited if needed |
| 24/06/2019 | Testing |
| 01/07/2019 | Spare week to handle any unexpected problems |
| 08/07/2019 | Hand in TMA03 |

## 2. Project Work Completed

### 2.1 Informationsources

As my project seemed to have a problem with the database locking, I wanted to find more information on Java Threads. A thread would allow requests to be stored and accessed in order, so if two people tried to update a record at the same time, the second would happen as the first finished. Using the search facility in the OU library, I came across a book called Java Threads, second edition (Oaks et al. 1998); which from the title seemed to offer exactly what I needed. I quickly discounted this book as I realised it was from 1999 and related to Java 2, so any information or techniques in there will be outdated.

After further searching I came across an article on the IBM website called Introduction to Java Threads (Goetz, 2002). Although this is still quite outdated, I can’t find many resources on the matter, or not many newer ones anyway. I decided to read it and not only has it given me a better understanding of what threads are, but when to use them, when *not* to use them, and how to implement them. As of writing this TMA I haven’t implemented them, but I think to get around the database locking issue as mentioned in my project work, I think I need to introduce them. Having processes running on threads will stop the database locking as methods won’t run concurrently, so they won’t be accessing the database at the same time.

As also mentioned in my project work, a major issue I was having was connecting my Java program to my SQLite database, something I haven’t had to do as part of my OU studies thus far. After searching on the internet, I managed to find an article on SQLite Tutorial (n.d.), which gave me not only the download link for the driver, but also step by step instructions on how to add the driver as a library to my java project and is also the basis for the “Test Connection” method in my “DataConnect.java” class. There were other sources available for this technique, but I found these the most straightforward and user friendly.

My next issue was that I have never used SQL and Java together, so I didn’t know how to do SQL calls within the Java code. There were ample resources available, but to start with I went to the Java Docs homepage, specifically relating to the java.sql package. These guides are the ultimate resource, everything you could need is there; but personally, I found all of the information a bit overwhelming. I didn’t spend too long looking at the docs, but I did refer to them about specific methods if I came across something in my other research that I didn’t understand. For the bulk of my database interaction, I used a website I’m going to refer to as “programming notes” titled “Java Database (JDBC) Programming by Examples with MySQL“ (Hock-Chuan, n.d.). I found the notes here relevant, easy to use, and they were helpful in creating my own classes and adapting to suit my needs.

### 2.2 Project Work

At the end of the last TMA I had a list of questions that I wanted to ask my client. I have reproduced the list below with the response:

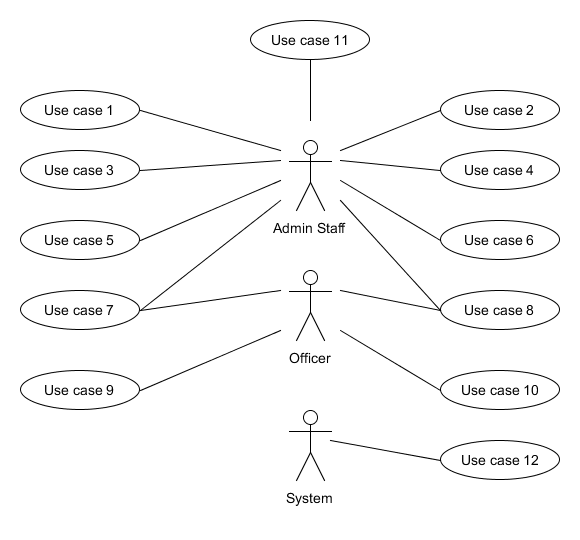
1. What is the role of the admin user? What do they do that the managers don’t and vice versa?  
   *There is no difference between the two users in regards to the function of the system, all staff will use the system, but admin staff wont be mentioned within the system. Site officers are admin staff that also perform site visits*
2. What happens in the event that the site user cant get a mobile signal, or use their phone for some reason.  
   *In this case, they tend to call or text their manager, or someone else in the office if they aren’t available. Even if a mobile signal cant be used for contact, they normally use a landline at whatever location they visit.*You may wish to create a message in your app that tells the user to call using a landline when data / mobile signal is not good.
3. What happens if a manager is not in the office when one of their team members go to site?  
   *If a manager is not available, due to annual leave etc, then another manager is appointed to that team in their absence. We’d quite like a way to move people teams in the system you’re developing to handle this if that’s at all possible.*Your requirements suggest that there is an important admin role, therefore site allocation may be added as suggested before.
4. How would you expect to be able to manually edit the times of the site officers?  
   *We’d only need to edit the “actual” check in and out times, but I’d expect them to be available within the system that’s used on a desktop machine.*
5. If an estimated check out time is not specified, would it be helpful to use a default value?  
   *No, I’d like the officers to be forced to enter an estimated time please.*
6. What information do you store about the sites that the officers visit?  
   *We need to store their name, mobile number, car registration, next of kin name, and a contact number for the next of kin.*Is there any unique information you store about the users?  
   *Not at the moment, but we have the staff ID number that could be used as a unique identifier if that’s helpful?*
7. Is a site deleted once the officers have finished all their visits? Is it made inactive?  
   *The site details don’t need to be kept. We visit thousands of sites per year, so its not feasible to store all that data. We’re just using this system to view officers current visits, once they check back in, there no need to store the data about that visit.*

#### 2.2.1 The project so far

After reviewing all the information, and speaking with the client, I have developed to following use cases:

|  |  |  |
| --- | --- | --- |
| Reference | Name | Description |
| UC1 | Add a manager | Add a new manager to the system |
| UC2 | Edit a manager | Edit a currently stored manager from the system |
| UC3 | Delete a manager | Delete a currently stored manager from the system |
| UC4 | Add an officer | Add a new officer to the system |
| UC5 | Edit an officer | Edit a currently stored officer from the system |
| UC6 | Delete an officer | Delete a currently stored officer from the system |
| UC7 | Set estimated times | An officer sets the estimated visit times using the GUI |
| UC8 | View more details | Anyone can click “more details” on the GUI and see more details about the selected officer |
| UC9 | Check in from app | An officer uses the mobile app to check in when they get to site |
| UC10 | Check out from app | An officer uses the mobile app to check out when they get to site |
| UC11 | Manually edit actual times | If a user rings the office rather than use the app, the actual times need manually updated |
| UC12 | Alert the manager | If an Officer hasn’t checked in by the time the estimated check out time arrives, an alert will present on the screen |

From the use case, I have created a use case diagram denoting the three actors; Admin Staff, Officers and the System itself the show the responsibilities concerned:



#### 2.2.2 First Iteration

##### 2.2.2.1 The Protype Designs

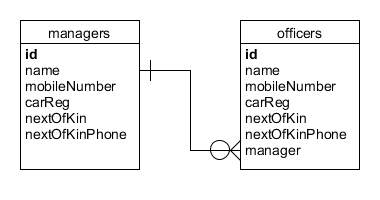
For the first iteration I have chosen to implement part of the GUI, specifically the part that allows new users to be added, edited or deleted; this will deal with UC1-UC6. I have initially developed the GUI as per [appendix 5.6](#_Appendix_5.6_–) and she has replied with the following comments:

* Could we get a button on the main screen, so admin staff can easily view all other details about the selected officer without having to go through the system looking for them
* The date field is irrelevant as we are only concerned with visits that are ongoing or happing on that day
* Could we have separate times on the page so we can see both the expected time on site and the actual time, that way we can see if someone has actually arrived on the site. This could be used in case someone has had an accident on the way to site. i.e. if they haven’t checked in around the time they were expecting, we can contact them to make sure everything is ok

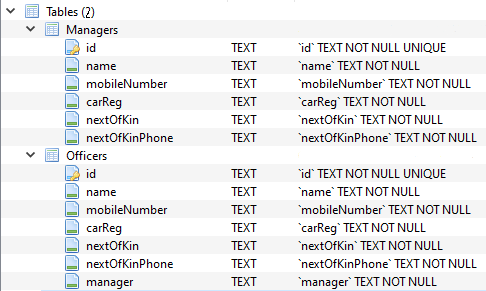
In response to the above points I have created the prototypes in [appendix 5.7](#_Appendix_5.7_–) which include an amended front screen, and a new screen for the “more details” button. After showing these new designs to my client, she’s agreed to all the changes and has given the approval to develop them.

##### 2.2.2.2 SQLite Database

Since the first iteration is only concerned with the data concerning the site officers and the managers, my ERD is displayed below to reflect this:

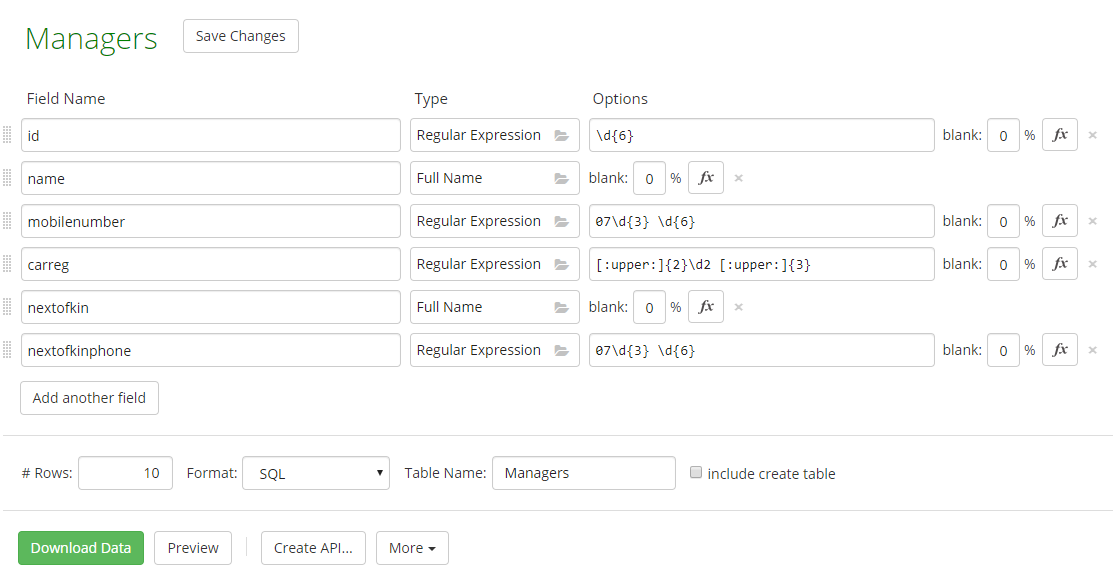


I have then used DB Browser for SQLite to create the database. This was done using the GUI, there is a create database button, from then it’s a simple as creating the tables using the buttons shown. My final table when created looks like this:



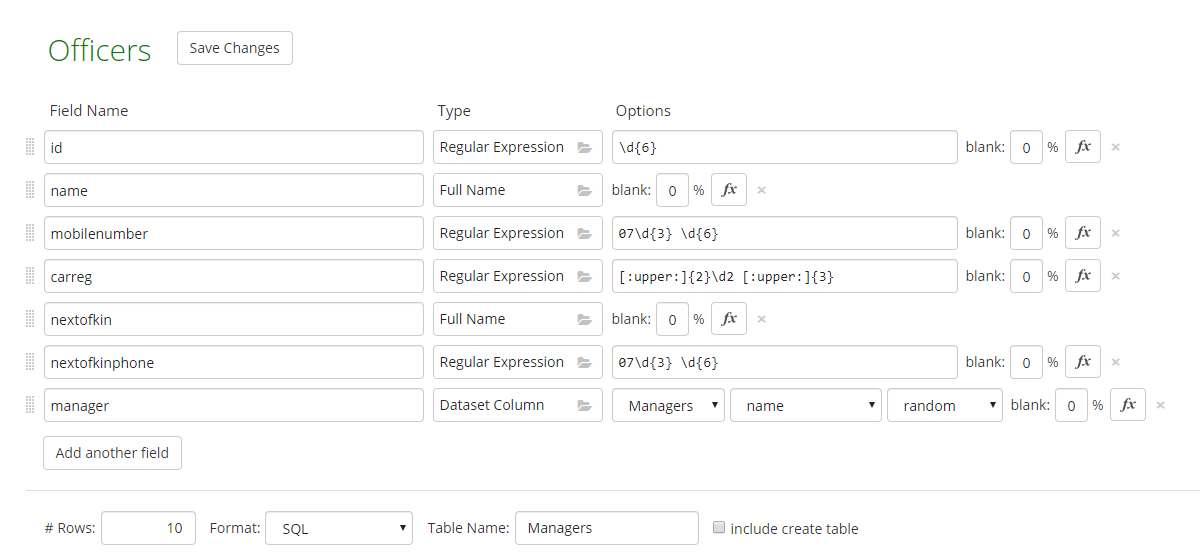
To populate the tables with test data I have used a website called Mockaroo. This website has allowed me to specify the types of data I want to create, and then provides the SQL for me to write to insert the data into the tables. The parameters I have used on the site to create the data are:

**Managers**



This gives me a 6 digit ID number (which matches our staff ID’s), a random name, a mobile number in the correct format, a car reg in the correct (post 2011) format, a random name for the next of kin and a phone number for them in the correct format.

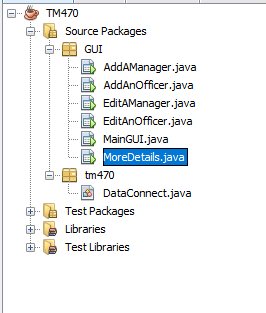
**Officers**



This gives me the same data as above, but with addition of a manager field, which randomly selects one of the previously created managers.

##### 2.2.2.3 Creating the GUI

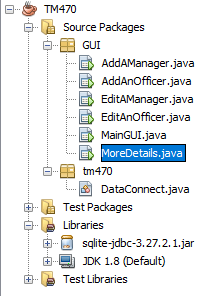
To create the GUI I have decided to use the JSwing elements of Java, as taught in M256. This will give me a drag-and-drop interface, which should allow me to fine tune the design to how its been agreed. I started by placing the elements on the page to recreate the main page and I’ll concentrate on the coding behind the objects later. I have created one main page, then an additional page to correspond to each of the buttons along the bottom of the page. All of these pages represent a Java Class, so that’s a total of 6 classes, one for each page. I have also created another class called ‘DataConnect.java’ which handles the connections to the database to make a total of 7 classes.



All the pages within the GUI package are available within [appendix 5.10](#_Appendix_5.10_–). I have used the appendix to detail how the pages work together, and any other relevant notes are below the pictures.

##### 2.2.2.4 Problems so far

The only main problem I’ve faced up until this point was that I didn’t know how to connect my java pages to my SQLite database. Although my studies have taught me Java and SQL individually, there has never been any cross over between the two. After a bit of research on the internet, I found a tutorial (SQLiteTutorial.com, n.d.), and after downloading the SQLite JDBC driver I managed to add it to my project and this will allow me to connect my Java GUI with my SQLite database. My directory now looks like this:



##### The code behind the pages

All of the classes I have created need code added to provide the required functionality. The below table lists the methods I will need are:

|  |  |  |  |
| --- | --- | --- | --- |
| **Class** | **Method** | **Activated by** | **Description** |
| MainGUI | btnTestConnectionMouseClicked() | Clicking on the ‘Test’ Connection’ button | Tests the connection to the database for testing purposes. To be removed in a later iteration |
| MainGUI | btnExitMouseClicked() | Clicking on the ‘Exit’ button | Displays a confirmation box asking if they want to close the system, if they click yes it closes, if they click no it returns to MainGUI |
| MainGUI | btnAddManagerMouseClicked() | Clicking on the ‘Add A Manager’ button | Displays the AddAManager screen |
| MainGUI | btnAddOfficerMouseClicked() | Clicking on the ‘Add An Officer’ button | Displays the EditAManager screen |
| MainGUI | btnEditManagerMouseClicked() | Clicking on the ‘Edit A Manager’ button | Displays the AddAnOfficer screen |
| MainGUI | btnEditOfficerMouseClicked() | Clicking on the ‘Edit An Officer Button’ button | Displays the EditAnOfficer screen |
| MainGUI | cmboOfficerMouseClicked() | Clicking on the ‘Officer’ dropdown menu | This will do an SQL call in the format “SELECT name FROM Officers” to populate the list of officers available |
| MainGUI | btnMoreDetailsMouseClicked() | Clicking on the ‘More details’ button | Displays the MoreDetails window |
| MainGUI | cmboOfficerActionPerformed() | Selecting a new officer from the officer drop down menu | Performs an SQL call in the format “SELECT \* FROM Officers WHERE name = ‘xxxx’”, where xxxx is the officer name selected in the list. |
| AddAManager | btnReturnMouseClicked() | Clicking on the ‘Return’ button | Returns to MainGUI |
| AddAManager | btnSaveMouseClicked() | Clicking on the ‘Save’ button | Performs an SQL call in the form “INSERT INTO Officers(id, name, mobileNumber, carReg, nextOfKin, nextOfKinPhone) values (?,?,?,?,?,?)", where the question marks refer to the values the user has entered on the screen |
| AddAnOfficer | btnReturnMouseClicked() | Clicking on the ‘Return’ button | Returns to MainGUI |
| AddAnOfficer | btnSaveMouseClicked() | Clicking on the ‘Save’ button | Performs an SQL call in the form “INSERT INTO Officers(id, name, mobileNumber, carReg, nextOfKin, nextOfKinPhone, manager) values (?,?,?,?,?,?,?)", where the question marks refer to the values the user has entered on the screen |
| EditAManager | btnReturnMouseClicked() | Clicking on the ‘Return’ button | Returns to MainGUI |
| EditAManager | btnDeleteMouseClicked() | Clicking on the ‘Delete’ button | Performs an SQL call in the form "DELETE FROM Managers WHERE id = \" + managerId + \" where the managerId is the manager’s ID that the user has selected in the form |
| EditAManager | btnUpdateMouseClicked() | Clicking on the ‘UPDATE button | Performs an SQL call in the form "UPDATE Managers SET name = xxxx. mobileNumber = xxxx, carReg = xxxx, nextOfKin = xxxx, nextOfKinPhone = xxxx WHERE id = \" + managerId + \" where the managerId is the manager’s ID that the user has selected in the form and xxxx are the details the user has entered on the form. |
| EditAnOfficer | btnReturnMouseClicked() | Clicking on the ‘Return’ button | Returns to MainGUI |
| EditAnOfficer | btnDeleteMouseClicked() | Clicking on the ‘Delete’ button | Performs an SQL call in the form "DELETE FROM Officers WHERE id = \" + id + \" where the id is the officers ID that the user has selected in the form |
| EditAnOfficer | btnUpdateMouseClicked() | Clicking on the ‘UPDATE button | Performs an SQL call in the form "UPDATE Officers SET name = xxxx. mobileNumber = xxxx, carReg = xxxx, nextOfKin = xxxx, nextOfKinPhone = xxxx, manager = xxxx WHERE id = \" + id + \" where the id is the officers ID that the user has selected in the form and xxxx are the details the user has entered on the form. |
| MoreDetails | btnReturnMouseClicked() | Clicking on the ‘Return’ button | Returns to MainGUI |

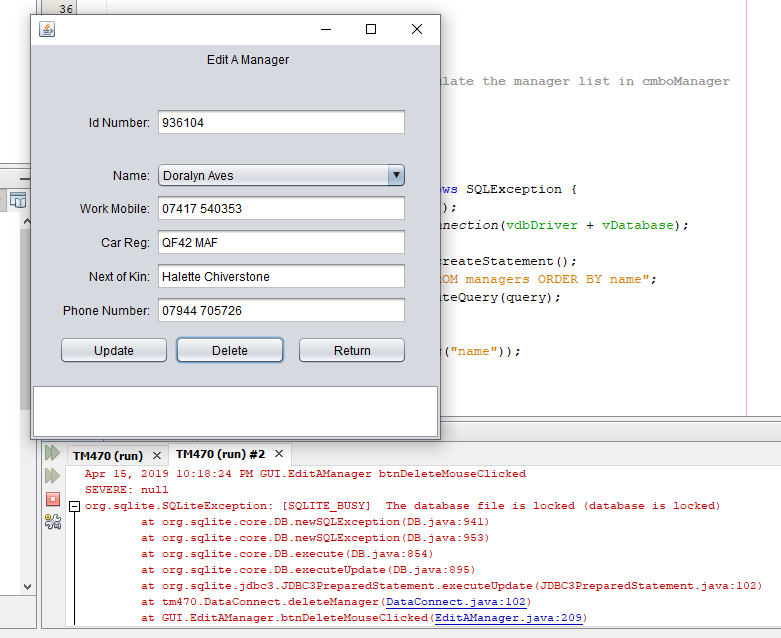
To help with the above methods, I will also create helper methods in the Data Connect class. These, and the full versions of my code can be found in [appendix 5.11](#_Appendix_5.11_–).

##### 2.2.2.6 UAT – User acceptance testing

To test my GUI I will ask my client Fiona to run the folling tests:

|  |  |  |  |
| --- | --- | --- | --- |
| **Reference** | **Name** | **Description** | **Outcome** |
| Test1 | testAddNewManager | To test the addition of a new manager, valid data entered by the user, expected success message | Success message generated, all fields cleared for next input |
| Test2 | testAddNewManager | To test the addition of a new manager, invalid data entered by the user, error message expected | Error message generated, as expected |
| Test3 | testAddNewOfficer | To test the addition of a new officer, valid data entered by the user, expected success message | Success message generated, all fields cleared for next input |
| Test4 | testAddNewOfficer | To test the addition of a new manager, invalid data entered by the user, error message expected | Error message generated, as expected |
| Test5 | testEditManager | To test deleting an existing manager, current manager selected, expected success message | Error message, database file has locked |
| Test6 | testEditManager | To test editing an existing manager, valid data entered by the user, expected success message | Not implemented yet due to error on previous method |
| Test7 | testEditManager | To test editing an existing manager, invalid data entered by the user, expected success message | Not implemented yet due to error on previous method |
| Test8 | testEditOfficer | Not implemented yet due to error on previous method | Not implemented yet due to error on previous method |
| Test9 | testEditOfficer | Not implemented yet due to error on previous method | Not implemented yet due to error on previous method |
| Test10 | testMoreDetails | After selecting a valid officer on MainGUI, all of their details should be available when the user clicks the ‘More Details’ button | Success. As expected, all the details of the currently selected officer are available when clicking on ‘More Details’ |

The tests above in Red failed, and I will be fixing this first as part of the next iteration. The error message showing on Test5 (with a random manager) is:



## 3. Review and Reflection

### 3.1 Ways of Working

I think on a whole, the way that I’m working suits my project well. At the moment, because of my family and work life, I tend to do most of my work in the evenings; which suits me as I seem to better with a quiet environment. Unfortunately, though this normally means late nights (and early mornings for work), so I can only manage a couple of nights before I need a night off. If at any point I fall behind in my studies, I can take time off work to help me catch up, as well as finish early or start late. Even after a few years of studying now, I am still trying to find what works best for me, I think it depends on my mood. Sometimes I like light, others dark; sometimes I like music, others silence; sometimes I can only work on my desk, others only on the sofa. My way of working adapts to the situation and the mood at the time.

My contact with my tutor has been minimal, I’ve only really contacted him when I’ve been having serious doubts about my project, normally right before a TMA is due. He has given me the advice and reassurances that I’ve needed ([see appendix 5.12](#_Appendix_5.12_-)).

Looking back at TMA01, my tutor gave me the following advice:

* identify list of resources and do their quality evaluation
* Suggest mitigation techniques for the risks associated with the tasks for your project

I have tried to give a more substantial list of resources, but I tend to read more articles on websites than the formal books that I think this is asking for? Maybe listing websites like I’ve done is enough, I just need o give more of a detail’s explanation of them and how they benefit my project? To address the second point, I have amended my risk table to include how I am mitigating the risks, and I hope this more what is expected. Bringing forward from TMA01, I have also included more information in the background and the proposed solution; I hoped a better understanding of the problem would lead to a better understanding of my thoughts towards the solution. Either way, I look forward to seeing the feedback provided by my tutor, hopefully the way I am currently working can help me to achieve

### 3.2 Evaluating Project Management

Generally, my project management skills have been as I expected. I have tried my best to stick to the plan I made in TMA, but I failed to consider my commitments from other modules; I’ve amended this in my TMA02 plan and I’ve allocated 2 weeks before TMA03 is due to concentrate on other assignments.

One thing I have neglected is my project log; I must’ve overlooked the advice to complete one. I have now started a project log as of half way through this TMA, and I have included my first two entries as appendix 5.13, and I will continue to fill this in every time I work towards my project.

I think the size of the subtasks I’ve set me are about right, although some are larger than others. I’ve set them in the way I have as they represent progress; at the end of each task I can look back and see something that I’ve accomplished. I could’ve broken them down further, but I don’t think the progress would’ve been as recognisable as it is at the moment. I also think the timeline that I’ve created now will help me to minimise the risk of not completing the project on time; if I complete all of the tasks as intended, I should have ample time to get everything written up in time for the EMA. One thing that I haven’t considered though, is the fact that I study 4 modules currently, and 3 of them are due to finish in July. This means that if anything, I’ll be able to dedicate more energy

### 3.3 Legal, social, ethical and professional issues

The solution I am implementing does touch on several legal, social, ethical and professional issues, and I’ll use the questions posed in the OU guidance to show how I am dealing with them:

Q. Who will use the results of your project and what impact could it have on them and on others?

A. The system will initially be trialled by the planning department, but depending on their results, it will be rolled out to other departments across Northumberland County Council and certain parts of the NHS that we work closely with.

Q. What could happen if the outcome of your project becomes extremely successful/influential? Could it have any impact upon society or the way people work? Would this be a good or potentially bad thing? How could any bad impacts be avoided?

A. If the project became extremely successful, it wouldn’t have a major impact upon society. The individuals using the product would be better protected, if there was an emergency, then it would be more easily identified, but there wouldn’t be any wide-reaching consequences. One big benefit would be cost; my product would be distributed for free, where similar products have a charge per user.

Q. Could private data be collected? Could this data conceivably be misused in some way? How could this be avoided?  
A. Private data is collected before the user begins using the product ([see appendix 5.5](#_Appendix_5_–)). This is stored in a secure cloud storage facility (In our case Google Drive or OneDrive). The only way this data can then be accessed is through a Northumberland County Council device with an active account on our domain; all our company data is secured in the same way. The only other private data that is collected is the user’s location, and as stated by the ICO

“You can only process location data if you are a [public communications provider](https://ico.org.uk/for-organisations/guide-to-pecr/key-concepts-and-definitions/#publicprovider), a provider of [a value-added service](https://ico.org.uk/for-organisations/guide-to-pecr/communications-networks-and-services/location-data/#valueaddedservice), or a person acting on the authority of such a provider, and only if:

* the data is anonymous; or
* you have the user’s consent to use it for a value-added service, and the processing is necessary for that purpose.”

As previously stated, the user agrees to any terms I’ve set, which will include the collection of personal data, when they submit their details to use the system.

Q. Could people be tracked? – are there privacy issues? Does this matter? How could this be avoided?

A. Yes, people could be tracked, but only by Council employees who already have authorisation to do so. Staff have previously agreed to this when they were issued with a work mobile phone, which will be used to run the app. The phone itself can be tracked without this app being installed.

Q. Could the outcome of your project conceivably be misused in some way, perhaps in conjunction with something else which may not even have been developed yet?

A. Not that I’m aware of, but any security implications of any unreleased software will be dealt with in future updates to the system.

## 4. References

Goetz, B (2002). *Introduction to Java Threads* [Online]. Available at <https://www.ibm.com/developerworks/java/tutorials/j-threads/j-threads.html> (Accessed 10/04/2019)

Hock-Chuan, C (n.d.). *Programming Notes* [Online]. Available at <http://www.ntu.edu.sg/home/ehchua/programming/java/jdbc_basic.html> (Accessed 05/04/2019)

Oaks, S., Wong, H., Loukides, M., (1999) *Java Threads: Second Edition*[Online], O’Reilly Media. Available at <https://learning-oreilly-com.libezproxy.open.ac.uk/library/view/java-threads-second/1565924185/pr01.html> (Accessed 29/03/2019).

Oracle Docs (n.d.) *Package java.sql* [Online]. Available at <https://docs.oracle.com/javase/8/docs/api/index.html?java/sql/package-summary.html> [Accessed 04/04/2019).

SQLite Tutorial (n.d.) *SQLite Java: Connect To The SQLite Database Using SQLite JDBC Drive*r [Online]. Available at <http://www.sqlitetutorial.net/sqlite-java/sqlite-jdbc-driver/> (Accessed 04/04/2019).

## 5. Appendices

[Appendix 5.1 – A screenshot showing a solution employed by a neighbouring Council](#_Appendix_1_–)

[Appendix 5.2 – The initial enquiry email from the client](#_Appendix_2_–)

[Appendix 5.3 – Initial meeting with the Client](#_Appendix_3_–)

[Appendix 5.4 – My interpretation of the main user interface](#_Appendix_4_–)

[Appendix 5.5 – Staff details collection form](#_Appendix_5_–)

[Appendix 5.6 – Proposed designs for the Java GUI](#_Appendix_5_–)

[Appendix 5.7 – Approved designs for the Java GUI](#_Appendix_6_–)

[Appendix 5.8 – Approved designs for the Mobile App](#_Appendix_7_–)

[Appendix 5.9 – Downloads used](#_Downloads)

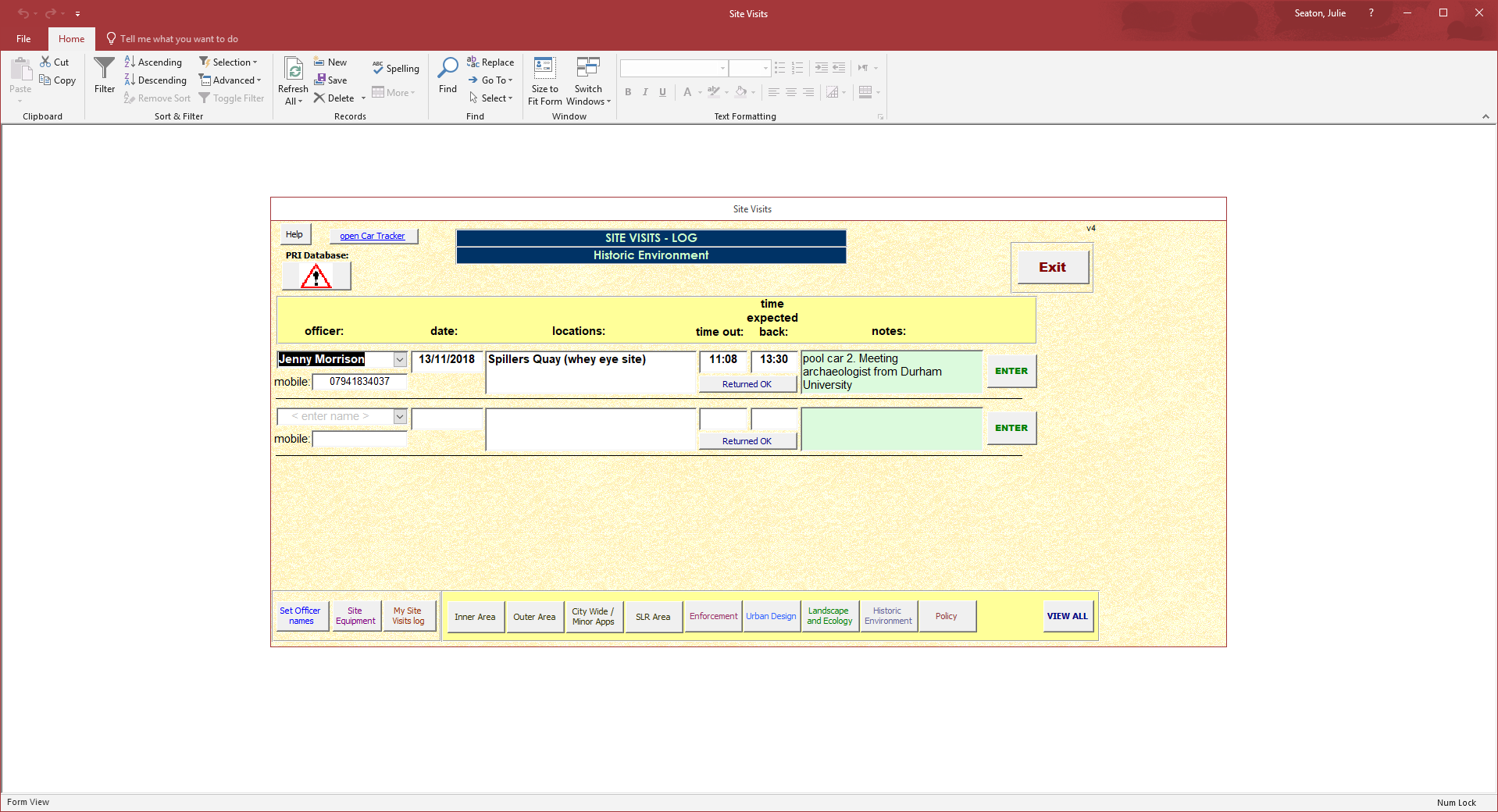
[Appendix 5.10 – JSwing pages created](#_Appendix_5.10_–)

[Appendix 5.11 – Java Code](#_Appendix_5.11_-)

[Appendix 5.12 – Tutor Contact](#_Appendix_5.12_-)

Appendix 5.13 – Project Log

### Appendix 5.1 – A screenshot showing a solution employed by a neighbouring Council



### Appendix 5.2 – The initial enquiry email from the client

|  |  |
| --- | --- |
| Northumberland County Council Mail | **Nicky Temperley <nicky.temperley@northumberland.gov.uk>** |

|  |
| --- |
| **FW: Lone Working Policy** 1 message |

|  |  |
| --- | --- |
| **Michelle Williams**<michelle.williams@northumberland.gov.uk> | 13 February 2019 at 12:11 |
| To: Nicky Temperley <nicky.temperley@gs.northumberland.gov.uk> | |
| |  | | --- | | Just checking with Joe Murphy/ Joanne Southern that anything has been done yet?  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ From: Fiona Charleton <[fiona.charleton@northumberland.gov.uk](mailto:fiona.charleton@northumberland.gov.uk)> Sent: 15 November 2018 16:32 To: Michelle Williams; Harri Bellizzi Subject: Fwd: Lone Working Policy  Hi Michelle/Harri  Sorry I know this is quite a bad time for you! I have been speaking with John Froud today about looking at the options of building a lone worker system for our Planning Officers who frequently go out on site. The system we use currently does not work so we need to review how we monitor our lone workers movements so we know where they are, who they are meeting and highlight when they haven't returned as expected.  I have spoken with Julie Seaton at Newcastle City Council. She is a former member of NCC staff and was very complimentary of the Newcastle system when she moved over 3 years ago. She has sent me a screenshot and brief overview as to how their system works. I have forwarded this on to IT for a call to be raised as to whether we can build something similar for our Officers. John has suggested I send this to both of you to look at the possibility of developing this. He is also keen to maybe widen this to other Services if we can build something that would work. Newcastle have been using this for a few years now and it does seem to work for them so we would be foolish to not investigate it when we are struggling with our current process.  Thanks Fiona  ---------- Forwarded message --------- From: Fiona Charleton <[fiona.charleton@northumberland.gov.uk](mailto:fiona.charleton@northumberland.gov.uk)<mailto:[fiona.charleton@northumberland.gov.uk](mailto:fiona.charleton@northumberland.gov.uk)>> Date: Thu, 15 Nov 2018 at 16:21 Subject: Fwd: Lone Working Policy To: John Froud <[john.froud@northumberland.gov.uk](mailto:john.froud@northumberland.gov.uk)<mailto:[john.froud@northumberland.gov.uk](mailto:john.froud@northumberland.gov.uk)>>   Hi John  Further to our discussion. Please see the below email which I mentioned I received from Newcastle City Council which details the home built system that they have been using with success for their Planning Officers.  I will update you when I know more from the call I have raised with IT. I will also send it separately to Michelle and Harri for them to look at.  Thanks for your help.  Fiona  ---------- Forwarded message --------- From: Seaton, Julie <[julie.seaton@newcastle.gov.uk](mailto:julie.seaton@newcastle.gov.uk)<mailto:[julie.seaton@newcastle.gov.uk](mailto:julie.seaton@newcastle.gov.uk)>> Date: Tue, 13 Nov 2018 at 12:26 Subject: RE: Lone Working Policy To: Fiona Charleton <[fiona.charleton@northumberland.gov.uk](mailto:fiona.charleton@northumberland.gov.uk)<mailto:[fiona.charleton@northumberland.gov.uk](mailto:fiona.charleton@northumberland.gov.uk)>>   Hi Fiona  Things are going OK here  We have an IT sytem that all officers must use when going out on site. They need to say where they are going and what time they will be back. It flashes an alert if they are not back by the time they stated and a manager must be notified. We monitor usage and it is a disciplinary offence if it’s not used. There are the odd occasion where people may go home straight from site and then we get them to call a ‘buddy’ back in the office when they are leaving the site – but it must be at a reasonable time when we know there will still be someone in the office who could raise the alarm if they don’t call.  All our pool cars (which staff use) have trackers on them so we could trace a car if we needed to.  This is a screen shot of the site visit database. Simon (our IT guru) put it together so he could possibly do something similar again.  Julie    -- Fiona Charleton Senior Technical Planning Officer Planning Northumberland County Council (01670) 622687   -- Fiona Charleton Senior Technical Planning Officer Planning Northumberland County Council (01670) 622687 | | |

### Appendix 5.3 – Initial Meeting with the Client

Q. What are you expecting from this system as a minimum requirement

A. I would like a system to record who’s out on site, when they got there, when their expected to leave, and receive an alert if they haven’t left by this time. Id like to be able the add or delete staff as they change too, although that will be mainly be down to the admin staff.

Q. What information do you store about the officers

A. We store their first name, surname, a mobile number, and their date of birth in case we have two people with the same name.

Q. Do you need the users to log in for security, or are their Windows Credentials enough to verify their identity?

A. No, theres no need for them to log in really.

Q. Do you need managers to have a different level of access to the admin staff? If so, what is the difference?

A.

Q. Rather than tracking pool cars, do all the staff have company issued mobile phones?

A. Yes, all staff have a company issued mobile phone and they all have a voice and data package.

Q. How would you like to see the alert to managers? By email, screen notification, text message etc.

A.

Q. Do you need to record historic information, or once the site officer has confirmed they’ve left the site can the information be discarded?

A. Yes, we would need to save the information for a minimum of one year.

Q. On the screenshot you’ve provided, would the user complete the site they are visiting, or is that selected from a list of sites?

A.

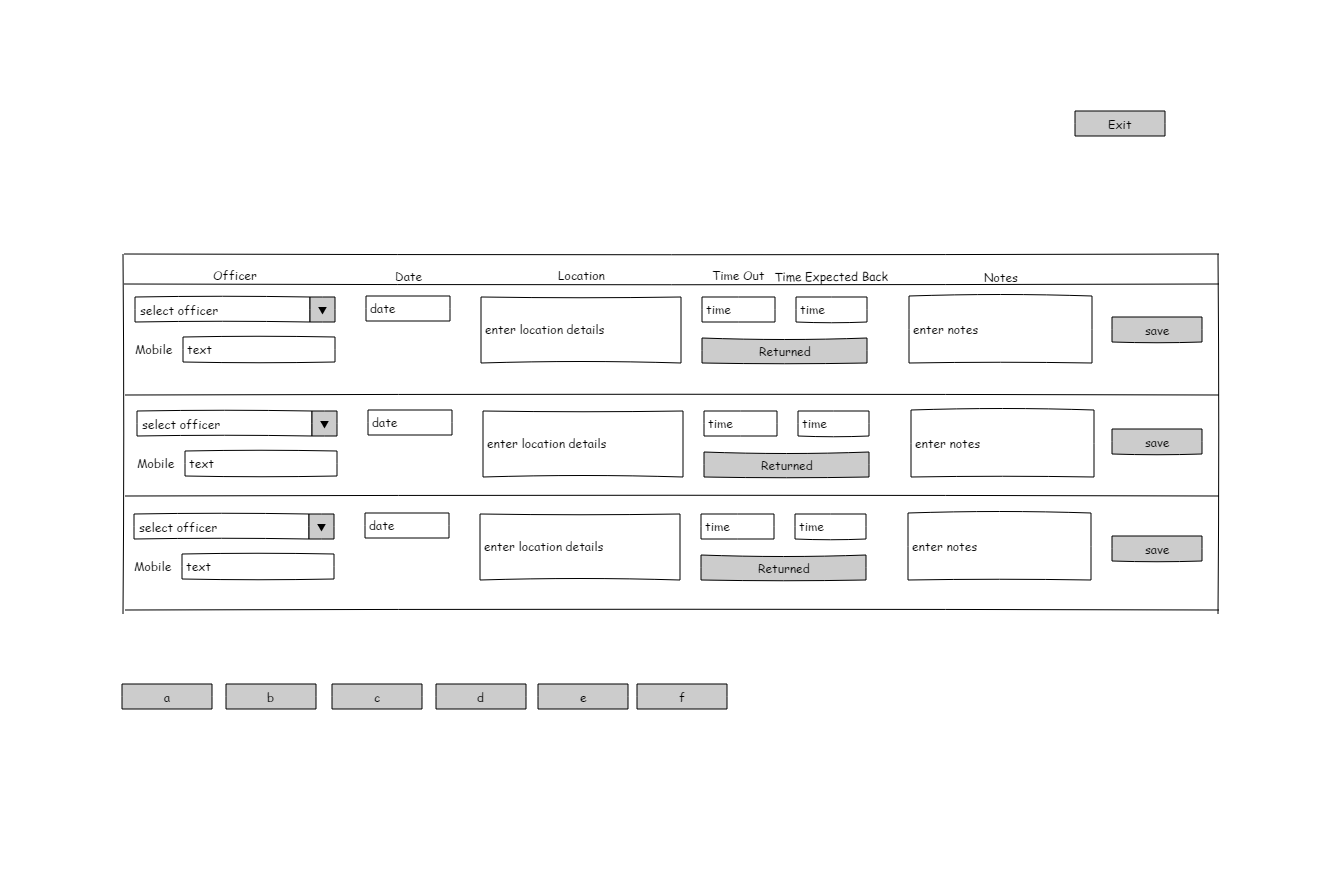
Q. Do you need to import any previous information into the system, or will you be using this as a fresh start?

A. No, we’re going to use this system as new, no previous data will be imported into it, it’ll all be stored in its original state. If we need to refer back to it we’ll do it manually, there’s no need to link this new system to it.

Q. Do you a unique identifier for Officers and locations, or would a generic ID be enough?

A. We use the staff payroll number as a unique ID for staff members, but the site ID isn’t relevant to anything we hold in our systems.

### Appendix 5.4 – My Interpretation of the main user interface



This file is saved as ‘My interpretation of the Main User Interface.epgz’ which can be opened with Pencil, or as an image as ‘My interpretation of the Main User Interface.png’

### Appendix 5.5 – Staff details collection form

Lone Work App - Account Creation

Please complete the following information to create your account

Hi, Nicky, when you submit this form, the owner will be able to see your name and email address.

Required

1.Staff ID



2.Phone Number



3.Next of Kin



4.Next of Kin Phone Number



5.Car Reg



6.Please select below to show your consent to use the information provided to create a new account in the NCC Lone Worker app. The information you provide will be stored and used in accordance to the Data Protection Act and GDPR. You can view full terms and conditions, or request to remove or update your information at any time by emailing [loneworker@northumberland.gov.uk](mailto:loneworker@northumberland.gov.uk)

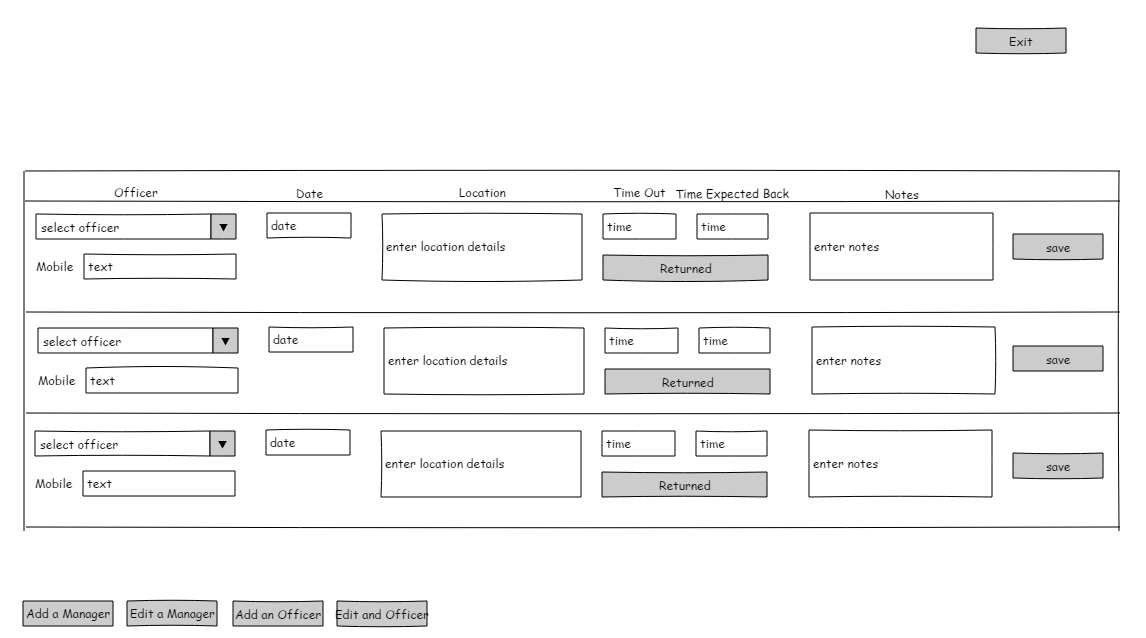
I Agree

I Do Not Agree

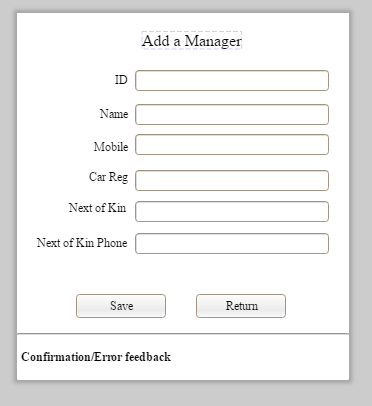
Submit

### Appendix 5.6 – Proposed designs for the Java GUI

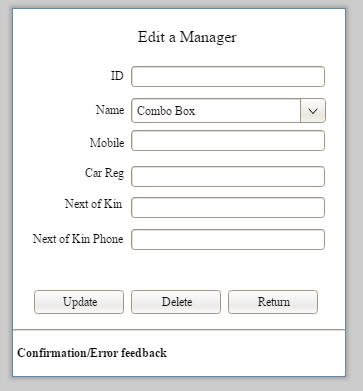
#### 5.6.1 Main User Interface



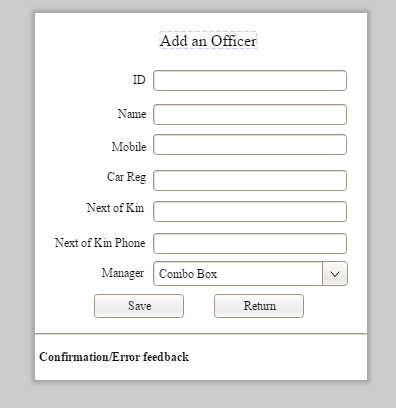
#### 5.6.2 Add a manager



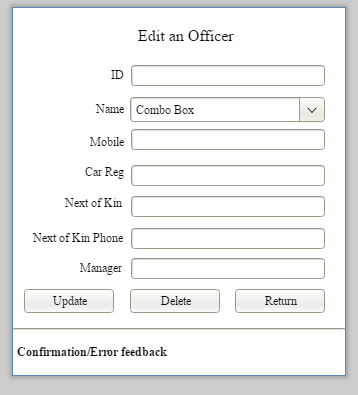
#### 5.6.3 Edit a Manager



#### 5.6.4 Add An Officer

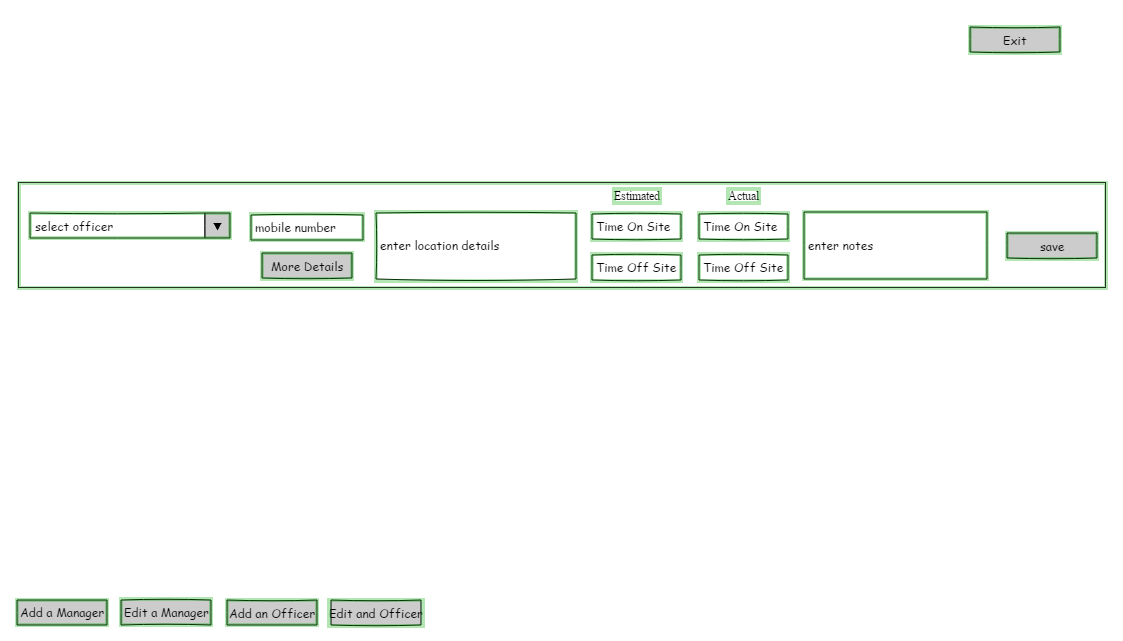


#### 5.6.5Edit An Officer

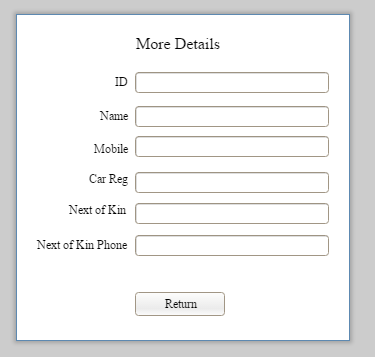


### Appendix 5.7 – Agreed designs for the Java GUI

#### 7.1 Main User Interface

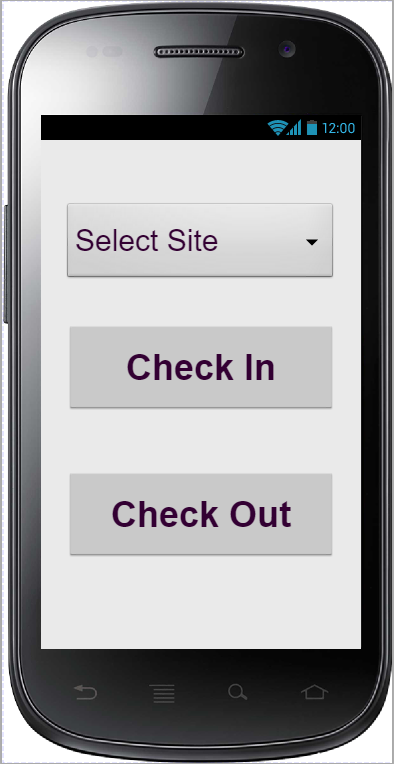


#### 7.2 More Details



**All other pages for the GIU were agreed as proposed**

### Appendix 5.8 – Agreed designs for the Mobile App



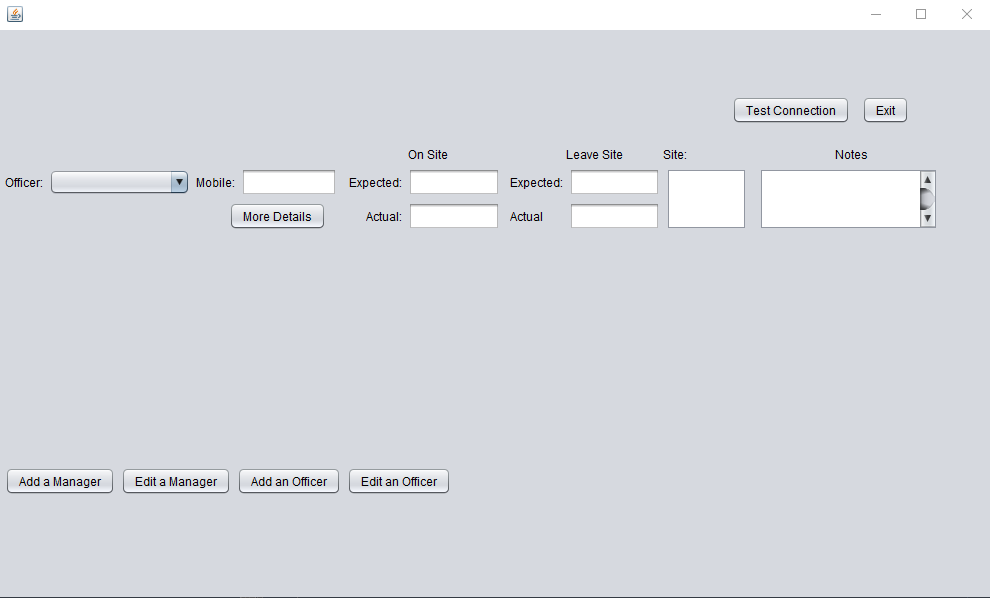
### Appendix 5.9 – Downloads used

|  |  |
| --- | --- |
| **Software** | **Download Link** |
| Java | <https://www.oracle.com/technetwork/java/javase/downloads/jdk-netbeans-jsp-3413139-esa.html> |
| Netbeans | <https://www.oracle.com/technetwork/java/javase/downloads/jdk-netbeans-jsp-3413139-esa.html> |
| DB Browser for SQLite | <https://portableapps.com/apps/development/sqlite_database_browser_portable> |
| Java ODBC Drivers | <https://bitbucket.org/xerial/sqlite-jdbc/downloads/> |
| UMLet | <https://www.umlet.com/changes.htm> |

### Appendix 5.10 – JSwing pages

#### 5.10.1 MainGUI,java

Exit the system



EditAManager.java

AddAManager.java

Only here for testing purposes

EditAnOfficer.java

AnnAnOfficer.java

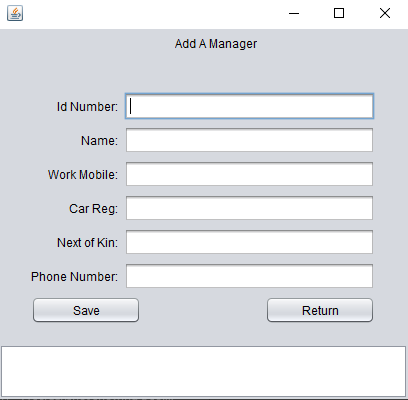
MoreDetails.java

For the purpose of this iteration, the users can:

* 1. Select an officer from the drop down box. This will populate the box with the officers name and the mobile field with their mobile number.
  2. Once an officer has been selected, the user can click on ‘More Details’ and see all the details held for that officer.
  3. Click ‘Add a Manager’ to go to the AddAManager.java page.
  4. Click ‘Edit a Manager’ to go to the EditAManager.java page.
  5. Click ‘Add an Officer’ to go to the AddAnOfficer.java page.
  6. Click ‘Edit an Officer’ to go to the EdditAnOfficer.java page.
  7. Click ‘Exit’ to close the system.

Please note, the ‘Test Connection’ button confirms connection to the database, and is only there for testing purposes. This will be remove in a later iteration.

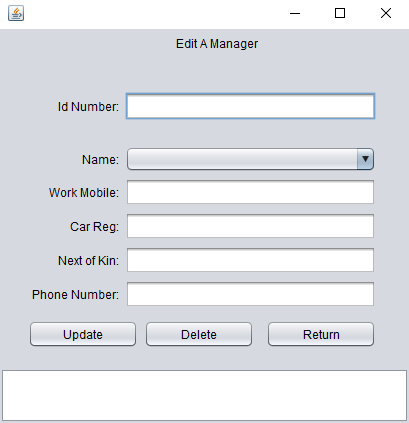
#### 5.10.2 AddAManager.java



MainGUI.java

The user can either enter all the required details here, or click return to go back to the main page. Once all the details have been entered and the user clicks save, either a confirmation message or an error message is displayed in the white bar at the bottom.

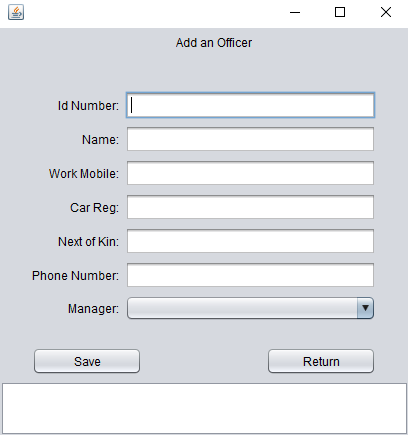
#### 5.10.3 EditAManager.java



MainGUI.java

On this page the user has to select the Manager they wish to amend from the drop-down list. Once selected they can either change any of the details and click update to save; click delete to remove the user from the database or return to the main screen. Once all the details have been entered and the user clicks save, either a confirmation message or an error message is displayed in the white bar at the bottom.

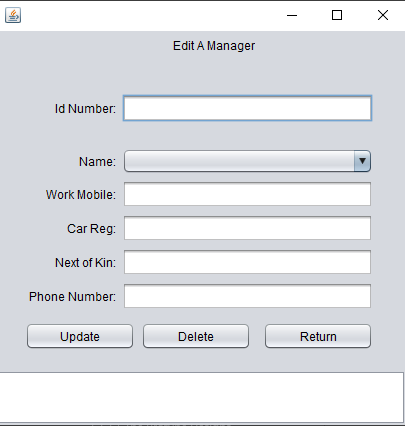
#### 5.10.4 AddAnOfficer.java



MainGUI.java

This form has the same functionality as AddAManager.java with the addition of the Manager drop down box; this needs to be selected when adding a new Officer. The manager drop-down provides a collection of all managers who are currently on the system. Once all the details have been entered and the user clicks save, either a confirmation message or an error message is displayed in the white bar at the bottom.

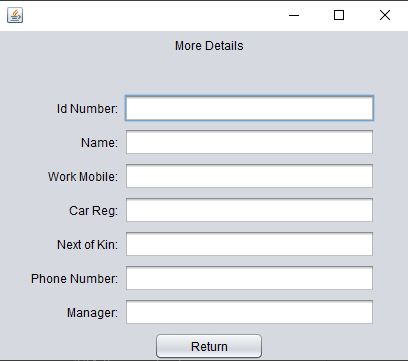
#### 5.10.5 EditAnOfficer.java



MainGUI.java

Again, this form has the same functionality as EditAManager.java with the addition of the Manager drop down box.

#### 5.10.6 More Details

  
  
This page has no functionality, the only option the user has is to click ‘Return’ and go back to the main page. The details cant be edited from this view either.

MainGUI.java

### Appendix 5.11 – Java Code

#### 5.11.1 MainGUI.java

**I have removed all system generated code. For the full code please see the attached program in the zip directory**

|  |
| --- |
| package GUI;  import java.sql.Connection; import java.sql.DriverManager; import java.sql.PreparedStatement; import java.sql.ResultSet; import java.sql.SQLException; import java.sql.Statement; import java.util.List; import java.util.logging.Level; import java.util.logging.Logger; import javax.swing.JOptionPane; import tm470.DataConnect;  /\*\*  \*  \* @author zy395907  \*/ public class **MainGUI** extends **javax**.**swing**.**JFrame** {   private void **btnTestConnectionMouseClicked**(java.awt.event.MouseEvent evt) {   DataConnect a = new DataConnect();   a.testConnection();  }   private void **btnExitMouseClicked**(java.awt.event.MouseEvent evt) {   int dialogButton = JOptionPane.YES\_NO\_OPTION;   int dialogResult = JOptionPane.showConfirmDialog(this, "Are you sure?", "Close the system", dialogButton);   if (dialogResult == 0) {    System.exit(0);   } else {    System.out.println("No Option");   }  }   private void **btnAddManagerMouseClicked**(java.awt.event.MouseEvent evt) {   AddAManager aManager = new AddAManager();   aManager.setVisible(true);  }   private void **btnAddOfficerMouseClicked**(java.awt.event.MouseEvent evt) {   AddAnOfficer anOfficer = null;   try {    anOfficer = new AddAnOfficer();   } catch (SQLException ex) {    Logger.getLogger(MainGUI.class.getName()).log(Level.SEVERE, null, ex);   }   anOfficer.setVisible(true);  }   private void **btnEditManagerMouseClicked**(java.awt.event.MouseEvent evt) {   EditAManager aManager = null;   try {    aManager = new EditAManager();   } catch (SQLException ex) {    Logger.getLogger(MainGUI.class.getName()).log(Level.SEVERE, null, ex);   }   aManager.setVisible(true);  }   private void **cmboOfficerMouseClicked**(java.awt.event.MouseEvent evt) {   cmboOfficer.removeAllItems();   DataConnect dbc = new DataConnect();   List < String > officers = null;   try {    officers = dbc.getOfficers();   } catch (SQLException ex) {    Logger.getLogger(MainGUI.class.getName()).log(Level.SEVERE, null, ex);   }   officers.forEach((officer) -> {    cmboOfficer.addItem(officer);   });  }   private void **btnMoreDetailsMouseClicked**(java.awt.event.MouseEvent evt) {   MoreDetails more = null;   try {    more = new MoreDetails();   } catch (SQLException ex) {    Logger.getLogger(MainGUI.class.getName()).log(Level.SEVERE, null, ex);   }   String officerName = (String) cmboOfficer.getSelectedItem();   more.setVisible(true);   try {    more.getDetails(officerName);   } catch (SQLException ex) {    Logger.getLogger(MainGUI.class.getName()).log(Level.SEVERE, null, ex);   }  }   private void **cmboOfficerPropertyChange**(java.beans.PropertyChangeEvent evt) {   System.out.println("Changed");  }   private void **cmboOfficerActionPerformed**(java.awt.event.ActionEvent evt) {   String vdbDriver = "jdbc:sqlite:";   String vDatabase = "D:/Program Files/SQLiteDatabaseBrowserPortable/Data/TM470.db";   Connection vConnection = null;   String name = (String) this.cmboOfficer.getSelectedItem();   if (name != null) {    try {     vConnection = DriverManager.getConnection(vdbDriver + vDatabase);     Statement stmt = vConnection.createStatement();     String query = "SELECT mobileNumber FROM Officers WHERE name = \"" + name + "\"";     ResultSet results = stmt.executeQuery(query);     txtMobile.setText(results.getString(1));    } catch (SQLException ex) {     Logger.getLogger(MainGUI.class.getName()).log(Level.SEVERE, null, ex);    }   }  } |

#### 5.11.2 AddAManager.java

**I have removed all system generated code. For the full code please see the attached program in the zip directory**

|  |
| --- |
| package GUI;  import java.sql.\*; import java.util.logging.Level; import java.util.logging.Logger;  /\*\*  \*  \* @author zy395907  \*/ public class **AddAManager** extends **javax**.**swing**.**JFrame** {       private void **btnReturnMouseClicked**(java.awt.event.MouseEvent evt) {                                               this.setVisible(false);     }                                        private void **btnSaveMouseClicked**(java.awt.event.MouseEvent evt) {                                               try {             String id;             String name;             String mobileNumber;             String carReg;             String nextOfKin;             String nextOfKinPhone;             String vdbDriver = "jdbc:sqlite:";             String vDatabase = "D:/Program Files/SQLiteDatabaseBrowserPortable/Data/TM470.db";             Connection vConnection;              id = txtId.getText();             name = txtName.getText();             mobileNumber = txtMobileNumber.getText();             carReg = txtCarReg.getText();             nextOfKin = txtNextOfKin.getText();             nextOfKinPhone = txtNextOfKinPhone.getText();             vConnection = DriverManager.getConnection(vdbDriver + vDatabase);             PreparedStatement st = vConnection.prepareStatement("insert into MANAGERS(id, name, mobileNumber, carReg, nextOfKin, nextOfKinPhone) values (?,?,?,?,?,?)");             st.setString(1, id);             st.setString(2, name);             st.setString(3, mobileNumber);             st.setString(4, carReg);             st.setString(5, nextOfKin);             st.setString(6, nextOfKinPhone);             int a = st.executeUpdate();             if (a > 0) {                 txtAddManagerOutcome.setText((String)name + " has been added, you may \n"                     + "now add another manager or close this screen by clicking Return");                 txtId.setText("");                 txtName.setText("");                 txtMobileNumber.setText("");                 txtCarReg.setText("");                 txtNextOfKin.setText("");                 txtNextOfKinPhone.setText("");             }         } catch (SQLException ex) {             Logger.getLogger(AddAManager.class.getName()).log(Level.SEVERE, null, ex);             txtAddManagerOutcome.setText("There has been an error \n"                     + "please check the data above and try again");         }      } |

#### 5.11.3 EditAManager.java

|  |
| --- |
| **I have removed all system generated code. For the full code please see the attached program in the zip directory**  package GUI;  import java.sql.Connection; import java.sql.DriverManager; import java.sql.PreparedStatement; import java.sql.SQLException; import java.util.List; import java.util.logging.Level; import java.util.logging.Logger; import tm470.DataConnect;  /\*\*  \*  \* @author zy395907  \*/ public class **EditAnOfficer** extends **javax**.**swing**.**JFrame** {      /\*\*      \* Creates new form AddAnOfficer      \* @throws java.sql.SQLException      \*/     public **EditAnOfficer**() throws SQLException {         initComponents();         cmboName.removeAllItems();         DataConnect dbc = new DataConnect();         List<String> managers = dbc.getManagers();         managers.forEach((manager) -> {             cmboName.addItem(manager);         });        }       private void **btnReturnMouseClicked**(java.awt.event.MouseEvent evt) {                                                this.setVisible(false);     } |

#### 5.11.4 AddAnOfficer.java

|  |
| --- |
| **I have removed all system generated code. For the full code please see the attached program in the zip directory**  package GUI;  import java.sql.Connection; import java.sql.DriverManager; import java.sql.PreparedStatement; import java.sql.SQLException; import java.util.List; import java.util.logging.Level; import java.util.logging.Logger; import tm470.DataConnect;  /\*\*  \*  \* @author zy395907  \*/ public class **AddAnOfficer** extends **javax**.**swing**.**JFrame** {      /\*\*      \* Creates new form AddAnOfficer      \* @throws java.sql.SQLException      \*/     public **AddAnOfficer**() throws SQLException {         initComponents();         cmboManager.removeAllItems();         DataConnect dbc = new DataConnect();         List<String> managers = dbc.getManagers();         managers.forEach((manager) -> {             cmboManager.addItem(manager);         });        }       private void **btnReturnMouseClicked**(java.awt.event.MouseEvent evt) {                                                this.setVisible(false);     }                                         private void **btnSaveMouseClicked**(java.awt.event.MouseEvent evt) {                                              try {             String id;             String name;             String mobileNumber;             String carReg;             String nextOfKin;             String nextOfKinPhone;             String manager;             String vdbDriver = "jdbc:sqlite:";             String vDatabase = "D:/Program Files/SQLiteDatabaseBrowserPortable/Data/TM470.db";             Connection vConnection = null;                                       id = txtId.getText();             name = txtName.getText();             mobileNumber = txtMobileNumber.getText();             carReg = txtCarReg.getText();             nextOfKin = txtNextOfKin.getText();             nextOfKinPhone = txtNextOfKinPhone.getText();             manager = (String)cmboManager.getSelectedItem();             Connection con = DriverManager.getConnection(vdbDriver + vDatabase);             PreparedStatement st = con.prepareStatement("insert into officers(id, name, mobileNumber, carReg, nextOfKin, nextOfKinPhone, manager) values (?,?,?,?,?,?,?)");             st.setString(1, id);             st.setString(2, name);             st.setString(3, mobileNumber);             st.setString(4, carReg);             st.setString(5, nextOfKin);             st.setString(6, nextOfKinPhone);             st.setString(7, manager);             int a = st.executeUpdate();             if (a>0)             {                 txtAddManagerOutcome.setText((String)name + " has been added, you may \n"                     + "now close this screen by clicking Return");                 txtId.setText("");                 txtName.setText("");                 txtMobileNumber.setText("");                 txtCarReg.setText("");                 txtNextOfKin.setText("");                 txtNextOfKinPhone.setText("");;                 cmboManager.setSelectedIndex(-1);                              }         } catch (SQLException ex) {             txtAddManagerOutcome.setText("There has been an error \n"                     + "please check the data above and try again");             Logger.getLogger(AddAnOfficer.class.getName()).log(Level.SEVERE, null, ex);         }          } |

#### 5.11.5 EditAnOfficer.java

|  |
| --- |
| **I have removed all system generated code. For the full code please see the attached program in the zip directory**  package GUI;  import java.sql.Connection; import java.sql.DriverManager; import java.sql.PreparedStatement; import java.sql.SQLException; import java.util.List; import java.util.logging.Level; import java.util.logging.Logger; import tm470.DataConnect;  /\*\*  \*  \* @author zy395907  \*/ public class **EditAnOfficer** extends **javax**.**swing**.**JFrame** {      /\*\*      \* Creates new form AddAnOfficer      \* @throws java.sql.SQLException      \*/     public **EditAnOfficer**() throws SQLException {         initComponents();         cmboName.removeAllItems();         DataConnect dbc = new DataConnect();         List<String> managers = dbc.getManagers();         managers.forEach((manager) -> {             cmboName.addItem(manager);         });        }       private void **btnReturnMouseClicked**(java.awt.event.MouseEvent evt) {                                                this.setVisible(false);     } |

#### 5.11.6 MoreDetails.java

**I have removed all system generated code. For the full code please see the attached program in the zip directory**

|  |
| --- |
| package GUI;  import java.sql.Connection; import java.sql.DriverManager; import java.sql.PreparedStatement; import java.sql.ResultSet; import java.sql.SQLException; import java.sql.Statement; import java.util.List; import java.util.logging.Level; import java.util.logging.Logger; import tm470.DataConnect;  /\*\*  \*  \* @author zy395907  \*/ public class **MoreDetails** extends **javax**.**swing**.**JFrame** {      /\*\*      \* Creates new form AddAnOfficer      \* @param name      \* @throws java.sql.SQLException      \*/     public **MoreDetails**() throws SQLException {         initComponents();              }       private void **btnReturnMouseClicked**(java.awt.event.MouseEvent evt) {                                                this.setVisible(false);     }                                         private void **txtNameMouseExited**(java.awt.event.MouseEvent evt) {                                                  }                                          /\*\*      \* Get the details for the officer sselected in MainGUI      \* @param officerName      \* @throws SQLException       \*/     public void **getDetails**(String officerName) throws SQLException {         String vdbDriver = "jdbc:sqlite:";         String vDatabase = "D:/Program Files/SQLiteDatabaseBrowserPortable/Data/TM470.db";         Connection vConnection = null;                  txtName.setText(officerName);         vConnection = DriverManager.getConnection(vdbDriver + vDatabase);         Statement stmt = vConnection.createStatement();             String query = "SELECT \* FROM Officers WHERE name = \"" + officerName + "\"";             ResultSet results = stmt.executeQuery(query);              txtId.setText(results.getString(1));             txtMobileNumber.setText(results.getString(3));             txtCarReg.setText(results.getString(4));             txtNextOfKin.setText(results.getString(5));             txtNextOfKinPhone.setText(results.getString(6));             txtManager.setText(results.getString(7));                             } |

#### 5.11.7 DataConnect.java

|  |
| --- |
| package tm470;  import java.sql.\*; import java.util.\*;  public class **DataConnect** {     String vdbDriver = "jdbc:sqlite:";  String vDatabase = "D:/Program Files/SQLiteDatabaseBrowserPortable/Data/TM470.db";  Connection vConnection = null;                           /\*\*          \* method to test database connection which returns a boolean value          \*/  public boolean **testConnection**(){    //try connecting to database, if successful show success message and return true boolean value  try {    vConnection = (Connection)DriverManager.getConnection(vdbDriver + vDatabase);    System.out.println("Connection to database successful!");    return true;      //catch will show unsuccessful message and return false boolean value     } catch (Exception ex) {    System.err.println ("Connection unsuccessful\n" + ex.toString ());    return false;  }     }                                                         /\*\*      \* return the list of managers to populate the manager list in cmboManager      \* on AddAnOfficer.java      \*      \* @return List of managers      \* @throws java.sql.SQLException      \*/     public List<String> **getManagers**() throws SQLException {         List<String> list = new ArrayList();         vConnection = DriverManager.getConnection(vdbDriver + vDatabase);         try {             Statement stmt = vConnection.createStatement();             String query = "SELECT name FROM managers ORDER BY name";             ResultSet results = stmt.executeQuery(query);             list.add(null);             while (results.next()) {                 list.add(results.getString("name"));             }                     } catch (SQLException e) {             System.out.println("Exception = " + e);         }         return list;     }          public List<String> **getOfficers**() throws SQLException {         List<String> list = new ArrayList();         vConnection = DriverManager.getConnection(vdbDriver + vDatabase);         try {             Statement stmt = vConnection.createStatement();             String query = "SELECT name FROM officers ORDER BY name";             ResultSet results = stmt.executeQuery(query);             list.add(null);             while (results.next()) {                 list.add(results.getString("name"));             }             stmt.close();         } catch (SQLException e) {             System.out.println("Exception = " + e);         }         return list;     }              public ResultSet **getManagersDetails**(String name) throws SQLException {         List<String> list = new ArrayList();         vConnection = DriverManager.getConnection(vdbDriver + vDatabase);         try {             Statement stmt = vConnection.createStatement();             String query = "SELECT \* FROM managers WHERE name = \"" + name + "\" ORDER BY name";             ResultSet results = stmt.executeQuery(query);                          return results;                      } catch (SQLException e) {             System.out.println("Exception = " + e);         }         //System.out.println(list);         return null;     }               public void **deleteManager**(String managerId) throws SQLException{             vConnection = DriverManager.getConnection(vdbDriver + vDatabase);             PreparedStatement st = vConnection.prepareStatement("DELETE FROM Managers WHERE id = \" + managerId + \"");             st.executeUpdate();                   }    } |

### Appendix 5.12 - Tutor Contact Log

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Contact via | Date | Time | Duration | Topic |
| Email | 02/02/2019 | 22:35 |  | Initial ‘Welcome to the course’ email |
| Email | 03/02/2019 | 22:19 |  | I replied telling Manish a bit about me and what I was planning on doing for TM470 |
| Email | 05/02/2019 | 21:46 |  | Manish replied to arrange a telephone appointment |
| Email | 06/02/2019 | 21:40 |  | I confirmed the appointment is acceptable |
| Telephone | 08/02/2019 | 20:31 | 9 minutes | Initial call to discuss my project ideas. At the time I was going to make a stock system for work, but Manish explained that this project wasn’t deep enough for me to gain the marks needed for this course, so he asked me to reconsider my project |
| Email | 13/02/2019 | 21:40 |  | After considering Manish’s advice about my project, I emailed him about developing a ‘Lone Worker System’ for the Planning Department and explained how I envisioned it. |
| Email | 20/02/2019 | 22:05 |  | Manish replied to say this idea sounded more feasible, but warned me about developing a real project as their expectations could be a hindrance |
| Email | 21/02/2019 | 20:32 |  | I emailed after the first meeting with the client, and I’ve agreed that although the IT department at work will develop the real application for her, she’s agreed to act as a fictitious client for me in the same regards. |
| Email | 27/02/2019 | 20:52 |  | Manish sent a group email with a marking scheme. |
| Email | 27/02/2019 | 00:19 |  | I emailed Manish requesting an extension due to a late meeting with the client |
| Email | 27/02/2019 | 20:13 |  | Manish agreed to my extension |
| Email | 03/03/2019 | 21:50 |  | I emailed to try and set up another telephone appointment to discuss concerns with my project |
| Email | 04/03/2019 | 16:48 |  | Emailed my availability for the telephone call |
| Email | 09/03/2019 | 10:33 |  | Manish apologised for not getting back to me |
| Email | 09/03/2019 | 11:19 |  | I emailed Manish my concerns |
| Email | 09/03/2019 | 12:56 |  | He replied to confirm that I’d mentioned I knew what I was doing now |
| Email | 11/04/2019 | 23:20 |  | Emailed to set up another telephone appointment |
| Email | 13/04/2019 | 19:32 |  | Manish replied suggesting a time |
| Email | 13/04/2019 | 21:00 |  | I replied saying I wasn’t available and suggested another time |
| Email | 13/04/2019 | 23:17 |  | And he confirm this was acceptable |
| Telephone | 14/04/2019 | 16:55 | 13 minutes | Explained my doubts about my assignment, Manish reassured me and advised me on what kind of things I should include, what to include from TMA01, and where I can make up marks where I lost them last time |
| Email | 15/04/2019 | 13:27 |  | I emailed asking for another short extension |

### Appendix 5.13 – Project Log

|  |  |
| --- | --- |
| Log Sheet Number: | 1 |
| Subject: | TM470 |
| Focus: | TMA02 |
| Time Spent: | 14 hours |
| Work | Inserted all appendices, finished section 1. Started Project Log!!! |
| Problems | Struggled to work out what to include in my project work. Very Time conscious! |
| Comments: | Happy with the results of my work. I wish I’d read the information at the start of the module, I didn’t realise I should be keeping a log. I will start one now and update it every time I study.  Still need to complete literature sources and section 3. I have asked for an extension, but hopefully I’ll only need a day |

|  |  |
| --- | --- |
| Log Sheet Number: | 2 |
| Subject: | TM470 |
| Time Spent: | 8 hours |
| Work | Finished TMA02 |
| Problems | Struggled with what to write for section 3 |
| Comments: | Finally finished my TMA, although I’m not happy with my literature sources and section 3; I don’t know if I’ve included enough data to get the marks I need. I feel that with literature sources I’m expected to read books and books on the subject, when in reality I’m only reading relevant pages or online articles that are beneficial to my project.  Section 3 seemed to be about reflecting on my work so far and although I’m pleased with my project, I always find myself asking “Is it enough”. I hope so |

## Definitions

Admin Staff – anyone who uses the developed system

Check In – When a site officer arrives to site

Check Out – When a site officer leaves site

Site Officer – An employee of Northumberland County Council that participates in site visits on behalf of the Planning Department