Nicky Temperley

Nicky Temperley

E6155906

TMA03



Table of Contents

[1 Draft Project Report 2](#_Toc13008829)

[1.1 Problem Description 2](#_Toc13008830)

[1.1.1 Background 2](#_Toc13008831)

[1.1.2 The Problem 2](#_Toc13008832)

[1.1.3 My Proposed Solution 3](#_Toc13008833)

[1.1.4 Alternatives to my proposal 3](#_Toc13008834)

[1.2 Account of related literature 4](#_Toc13008835)

[1.3 Account of project work and its outcome 4](#_Toc13008836)

[2 Review 15](#_Toc13008837)

[2.1 Review of the current stage of project work 15](#_Toc13008839)

[2.2 Review Project Management 15](#_Toc13008840)

[2.3 Assess risks to project completion 15](#_Toc13008841)

[2.4 Review of personal development 15](#_Toc13008842)

[3 References 16](#_Toc13008843)

[4 Appendices 17](#_Toc13008844)

# 1 Draft Project Report

## Problem Description

### 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.

### 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 |

## Account of related literature

TODO Literature from TMA01

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.

## Account of project work and its outcome

Enter work from TMA01

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:

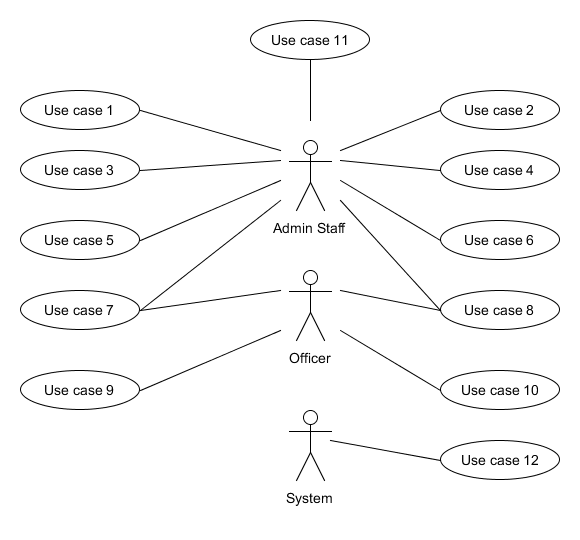
1. What is the role of the admin user? What do they do that the managers don’t and vice versa?  
   *There is no difference between the two users in regards to the function of the system, all staff will use the system, but admin staff wont be mentioned within the system. Site officers are admin staff 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

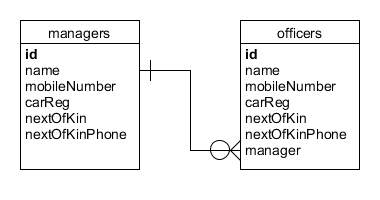
For the first iteration I have chosen to implement part of the GUI, specifically the part that allows new users to be added, edited or deleted; this will deal with UC1-UC6. I have initially developed the GUI as per [appendix 5.6](#_Appendix_5.6_–) and she has replied with the following comments:

* 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

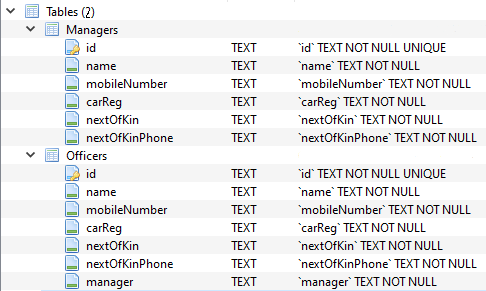
In response to the above points I have created the prototypes in [appendix 5.7](#_Appendix_5.7_–) which include an amended front screen, and a new screen for the “more details” button. After showing these new designs to my client, she’s agreed to all the changes and has given the approval to develop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

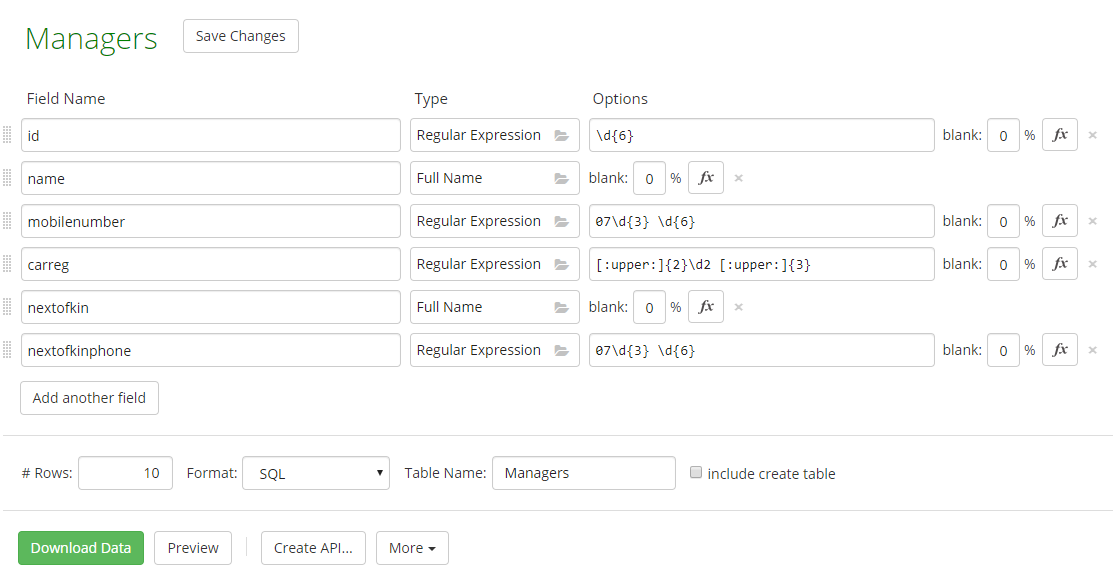


I have then used DB Browser for SQLite to create the database. This was done using the GUI, there is a create database button, from then it’s a simple as creating the tables using the buttons shown. My final table when created looks like this:



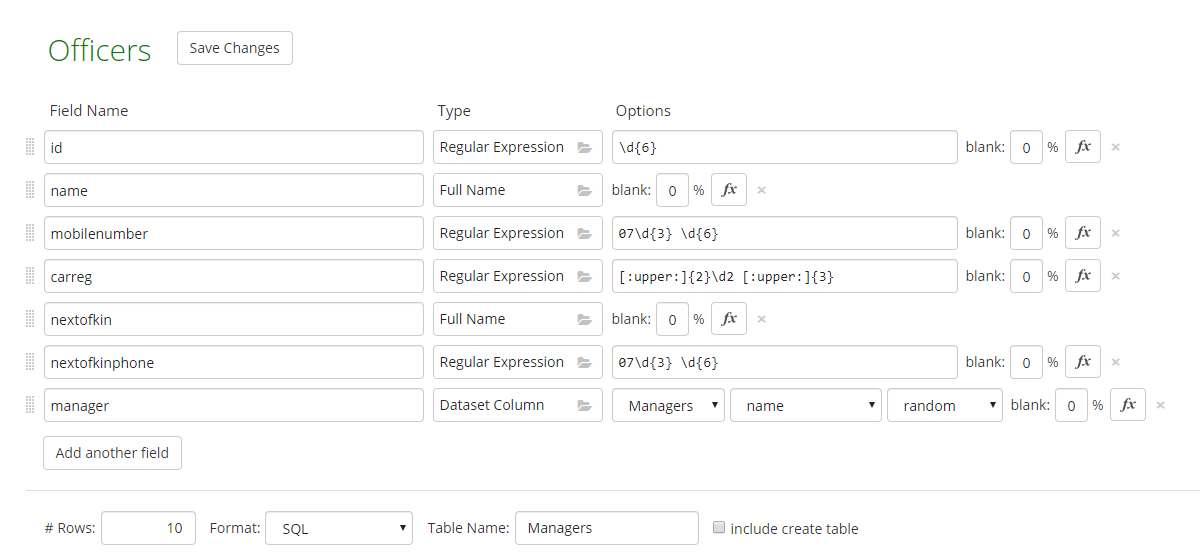
To populate the tables with test data I have used a website called Mockaroo. This website has allowed me to specify the types of data I want to create, and then provides the SQL for me to write to insert the data into the tables. The parameters I have used on the site to create the data are:

**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.

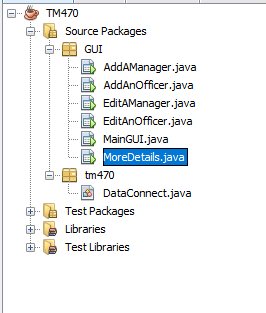
**Officers**



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

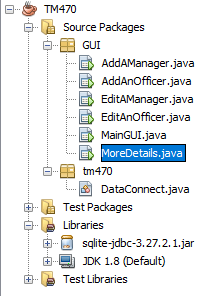
To create the GUI I have decided to use the JSwing elements of Java, as taught in M256. This will give me a drag-and-drop interface, which should allow me to fine tune the design to how its been agreed. I started by placing the elements on the page to recreate the main page and I’ll concentrate on the 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 |

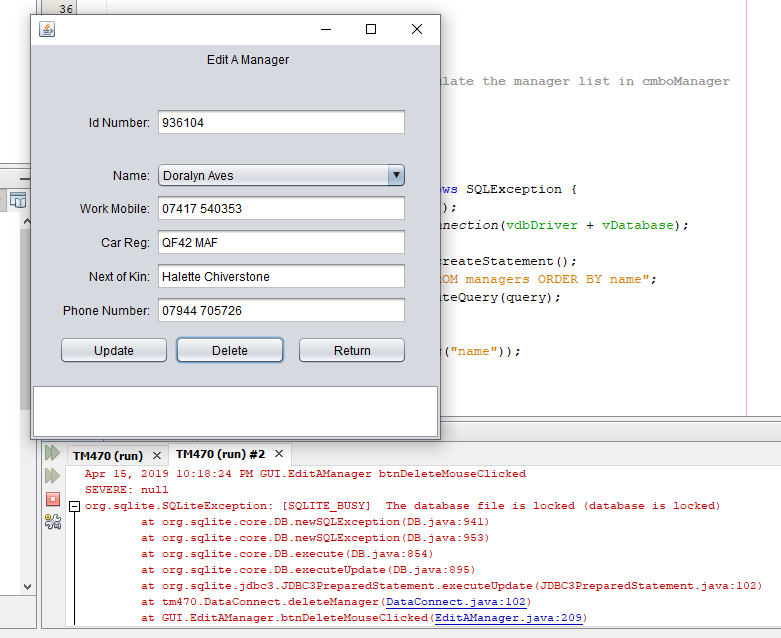
To help with the above methods, I will also create helper methods in the Data Connect class. These, and the full versions of my code can be found in [appendix 5.11](#_Appendix_5.11_–).

##### 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:



# 2 Review



## Review of the current stage of project work

## Review Project Management

## Assess risks to project completion

## Review of personal development

# 3 References

# 4 Appendices

Code files attached to compile solution