

Final Report



Title: District Nurse Call scheduling problem

Author: Charlotte Lippiett

Supervisor: Helen Phillips

Moderator: Yukun Lai

Module Number: CM0343

(40 Credits)

Abstract

This project is aimed at working with district nurses to create a proto-type system that will be beneficial for them. This system will manage the allocation of calls in the most efficient way possible to allow for staff's time to be used spending extra time with patients. This will allow for staff's time to be exploited in the best way possible. For this project I looked into the many aspects that the district nurses required and concentrated on the main function; allocating calls. This allows for vast options for expansion in the future in order to improve the system.

This system will be a desktop application which can be accessed from secure offices in the Bargoed district. This system will allow for easy interaction between itself, the district nurses and the administrator.

Contents

Abstract	2
Acknowledgements	3
Figures	3
Tables	4
1 Introduction	5
1.1 Initial Description (from Initial and Interim reports):	5
1.2 Description:	5
1.3 Summary of User Requirements:	5
2 Design	6
2.1 Activity Diagrams:	6
2.2 User Stories:	10
2.3 Use Cases:	11
2.4 Database Design	17
2.5 Algorithms	19
3 Implementation	20
3.1 Templates:	20
3.2 Implementation:	28
3.3 Creating Interfaces:	28
3.4 Low level solutions to high level problem:	29
3.5 Main aspects of solution:	37

4	Results and Evaluation	40
4.1	Results:	40
4.2	Testing system with real users (User Testing):	46
4.3	Evaluation:	50
5	Future Work	56
5.1	Additional requirements:-	56
5.2	Card Reader:-	57
5.3	Making system online:-	60
5.4	Cannot be portable:-	60
5.5	Translate to welsh:-	61
6	Reflection on Learning	61
	Appendices	65
	References	65

Acknowledgements

- Carol Lippiett – Mother, client and district nurse.

Figures

Figure 1.	Login - Administrator & Nurse	6
Figure 2.	Allocating calls - Administrator	7
Figure 3.	View personal details - Nurse	7
Figure 4.	Viewing staff details – Administrator	8
Figure 5.	View nurse call list (personal) – Nurse	8
Figure 6.	View Patients’ Details – Administrator and Nurse	9
Figure 7.	Patient details found or not found – Administrator and Nurse	9
Figure 8.	View nurse call lists – Administrator	10
Figure 9.	Use case	11
Figure 10.	Temporary Tables included in class diagram	17
Figure 11.	Updated class diagram	18
Figure 12.	Log in screen	20
Figure 13.	Main menu - Administrator	20
Figure 14.	Allocate calls – main screen	21
Figure 15.	Allocate calls - check	21
Figure 16.	Allocate Calls – date entry	21
Figure 17.	Allocate calls - confirmation	22

Figure 18. view staff details - main screen	22
Figure 19. view staff details - selection	22
Figure 20. view staff details - individual staff details	23
Figure 21. view patient's details - main screen	23
Figure 22. view patient's details - criteria entered	24
Figure 23. view patient's details - matched criteria	24
Figure 24. view patient's details - patient selected	24
Figure 25. view patient's details - patient's details	25
Figure 26. view calls - main screen	25
Figure 27. view calls - staff selected	25
Figure 28. view calls - staff call list	26
Figure 29. main screen - Nurse	26
Figure 30. personal details main	26
Figure 31. view personal call list - main screen	27
Figure 32. Administrator functionality	40
Figure 33. Main log in screen	42
Figure 34. Main administrator screen	42
Figure 35. Main screen for allocation of calls	42
Figure 36. Check staff have been selected correctly and enter date	43
Figure 37. Confirmation message that the call list has been created	43
Figure 38. Nurse functionality	43
Figure 39. Main nurse screen	45
Figure 40. Individual call list	45
Figure 41. In depth details on selected patient	45
Figure 42. Front view of lone worker - badge	58
Figure 43. Back view of lone worker - badge and buttons	58

Tables

Table 1. Call Allocation Functionality	43
Table 2. Viewing call list functionality	46

1 Introduction

1.1 Initial Description (from Initial and Interim reports):

Currently the team of district nurses assigns one member to allocate their calls amongst staff with no particular structure, and then each staff member has to individually organise their call list. This is a time consuming task for each individual staff member as this needs to be done the day before, as many calls need to be allocated to specific staff members based on the skill (band) level. Often when this is done some patients are assigned to the wrong members of staff and therefore need to be swapped in order for all calls to be covered. This project aims to implement a system that can distribute calls amongst all staff members taking into account their band staff level, the length of shift and type of call. This system should optimise staff within patient's houses and allow for more time to fill in important paperwork.

1.2 Description:

This project will see the creation of a prototype system in order to assist a team of nurses to create the most efficient call lists. They have a current manual system in place which takes up a lot of their valuable time which is clearly stated in the questions discussed in the Interim report. Their time is currently used allocating calls to their team of nurses and then each nurse needs to manually sort through their own lists. This problem was considered in the Interim report and following extensive background research a proposed solution was created. This proposed solution takes into consideration all the requirements directly from the district nurses along with following ethical and legal guidelines set by NHS. The prototype system aims to demonstrate such a system that provides a solution to the district nurse's main problem and a system that can be used for future development.

Following on from the background research carried out in the Interim Report, I looked more in depth into the current problem in order to understand what aspects exactly to implement.

1.3 Summary of User Requirements:

These were concluded in the Interim Report:

Core Functionality:-

- Provide a set of Calls for each district nurse that is required to work. This call list will be created using the 'Call List' section from the interim report.

Additional Functionality:-

- Staff and patients details can be changed directly from the system. Without this the information for staff and patients will be changed by the administrator directly from the database.
- Time constraints for calls. Without this an average time for each call is taken into account and calculated accordingly.
- Call Reallocation. This could be needed should a staff member who is allocated to work a particular day in unable to work for unforeseen circumstances.
- Take into account the band level of staff as certain members of staff are able to complete certain types of calls. Some calls require a higher level of staff than others and more skills are needed in order to treat patient correctly.

2 Design

2.1 Activity Diagrams:

(Oestererich, 2002)

(Bennett, Skelton, Lunn, 2004)

I decided to use activity diagrams as a way to represent the activities and actions performed at each step. They help break down a complex structure into step-by-step workflows which I can use as a guideline during implementation. Each diagram can display the activity states, transitions and decisions to show the overall control of the system.

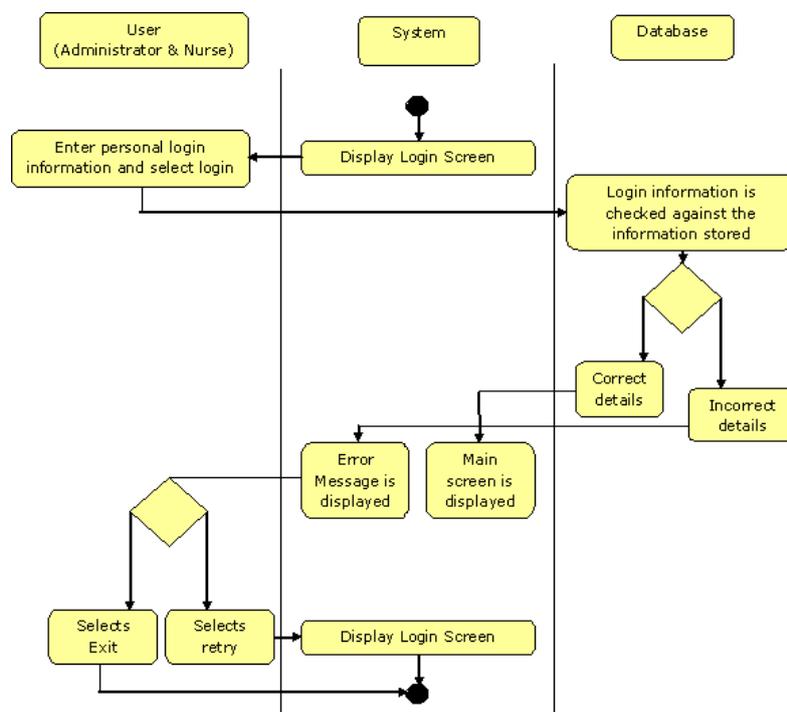


Figure 1. Login - Administrator & Nurse

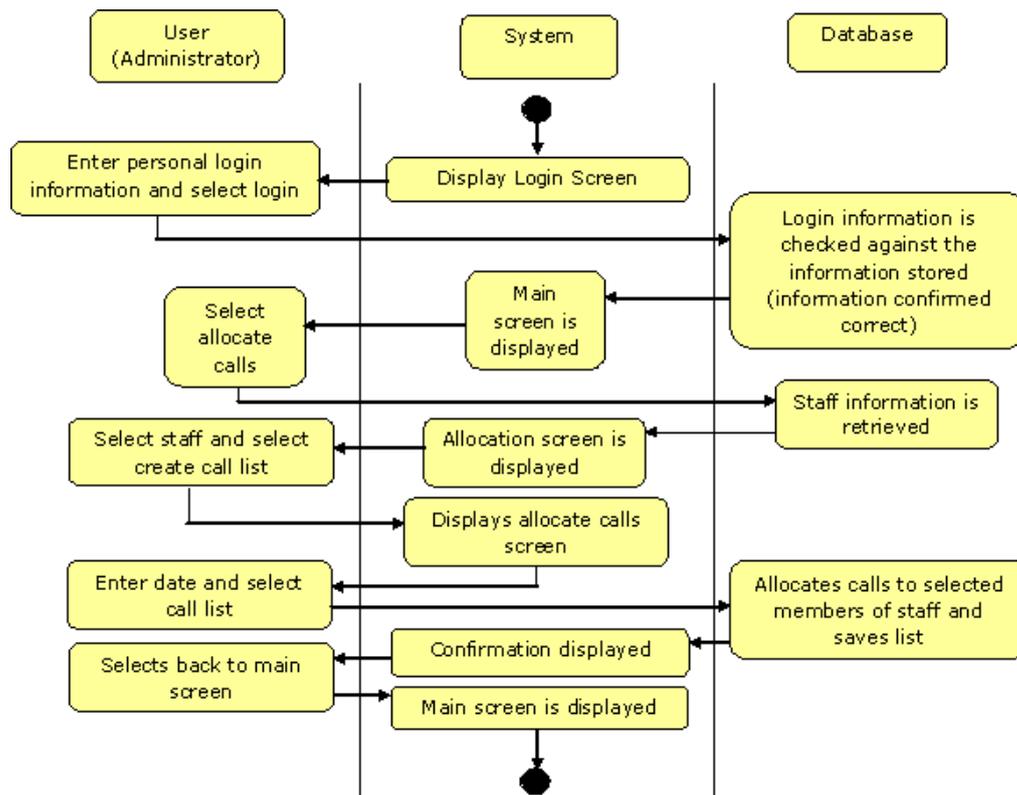


Figure 2. Allocating calls - Administrator

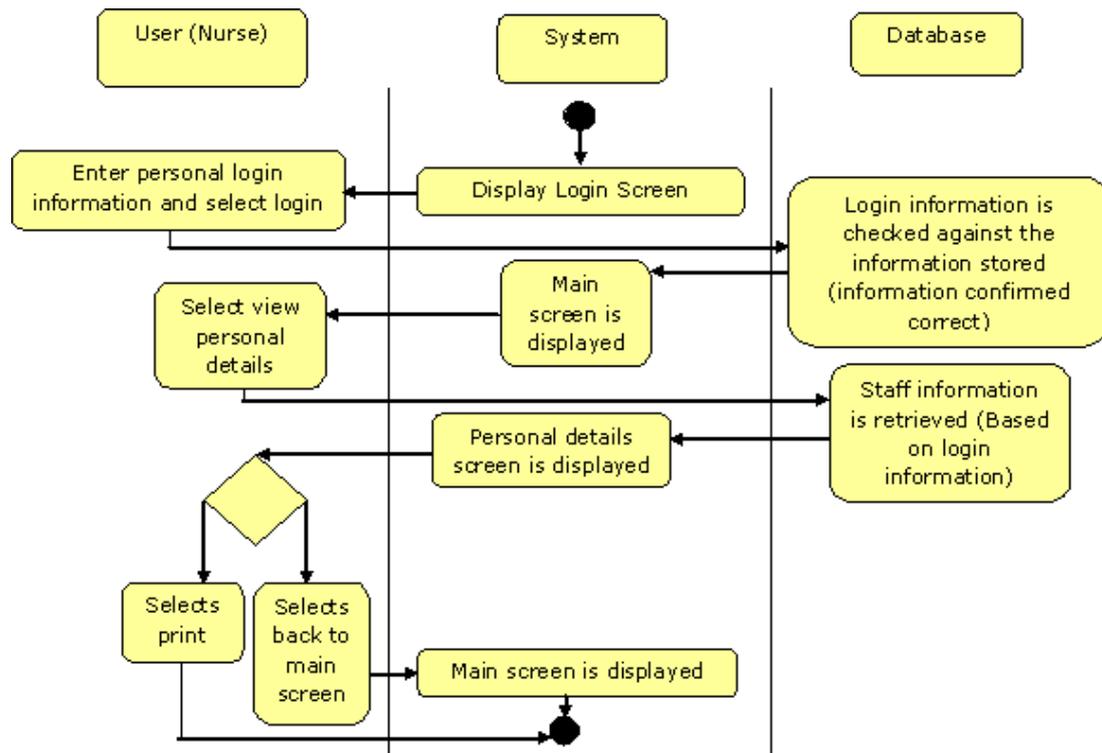


Figure 3. View personal details - Nurse

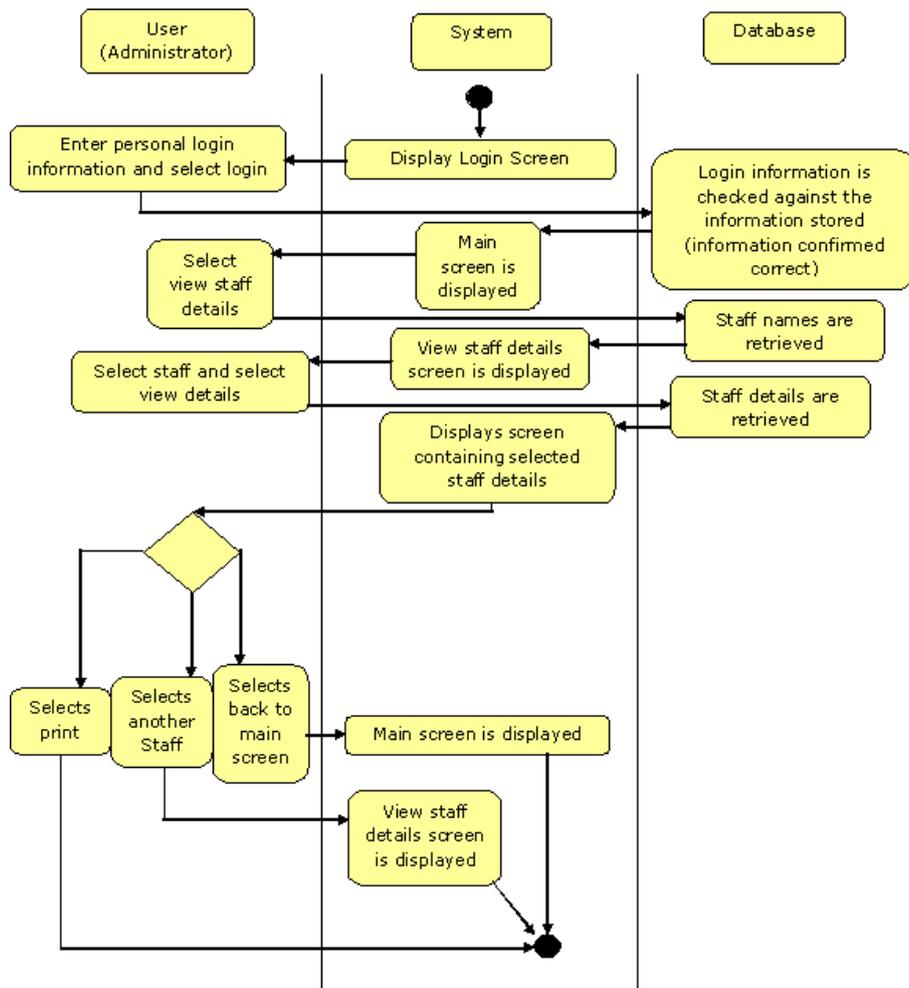


Figure 4. Viewing staff details – Administrator

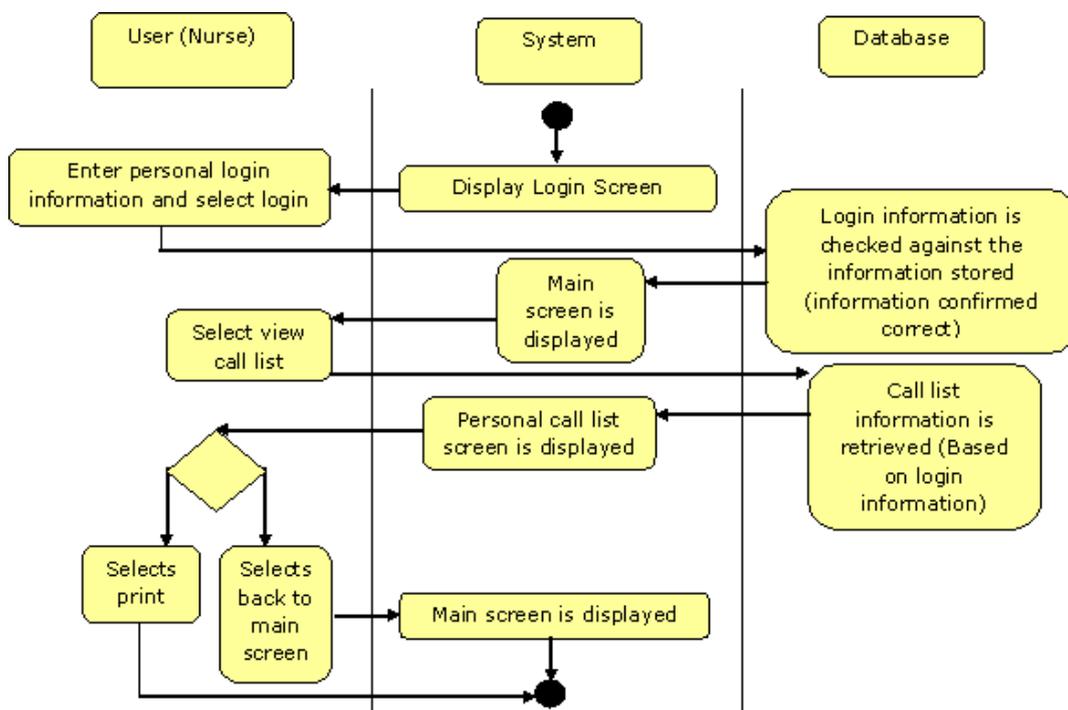


Figure 5. View nurse call list (personal) – Nurse

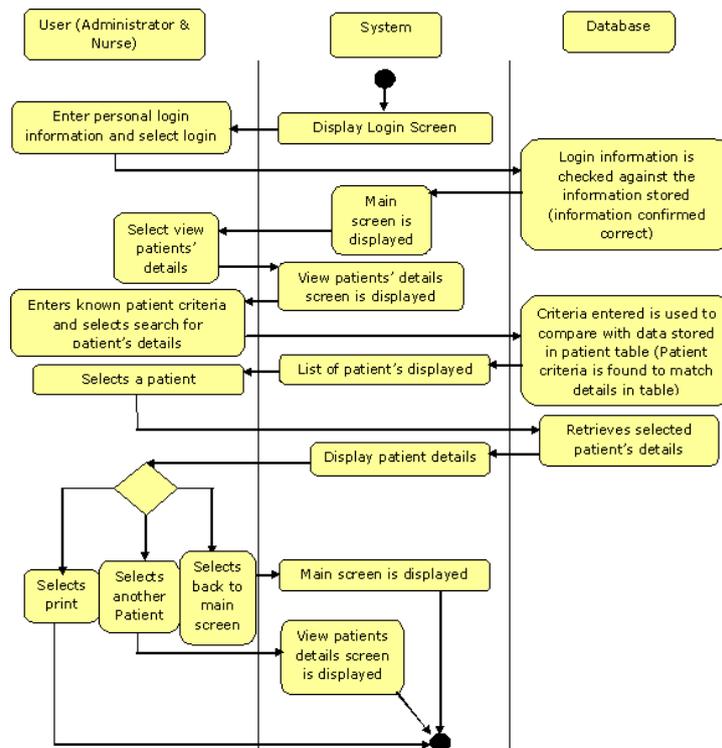


Figure 6. View Patients' Details – Administrator and Nurse

Within the *view patients'* part of the system, the system is required to allow for user to input patient criteria. This criterion is then compared to information stored in the patient database. This aspect is part of Figure 6. This activity diagram shows the route if a patient is found or not found within the database.

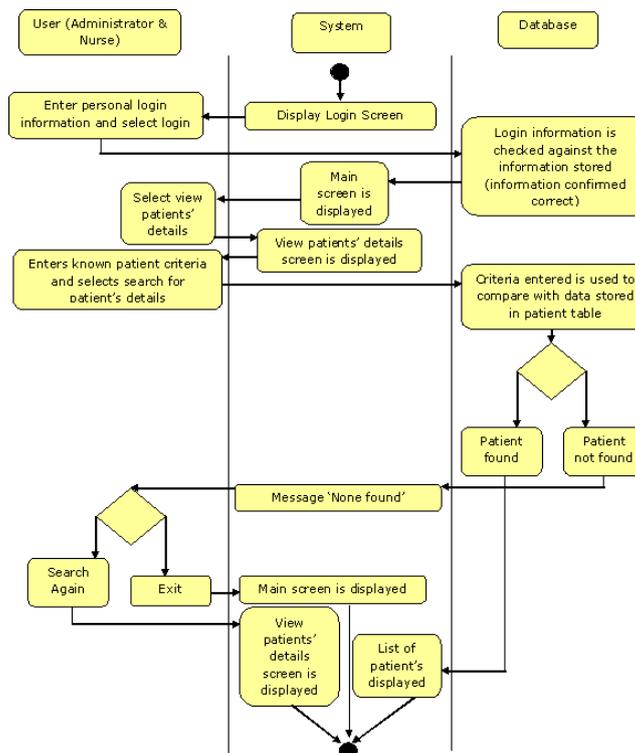


Figure 7. Patient details found or not found – Administrator and Nurse

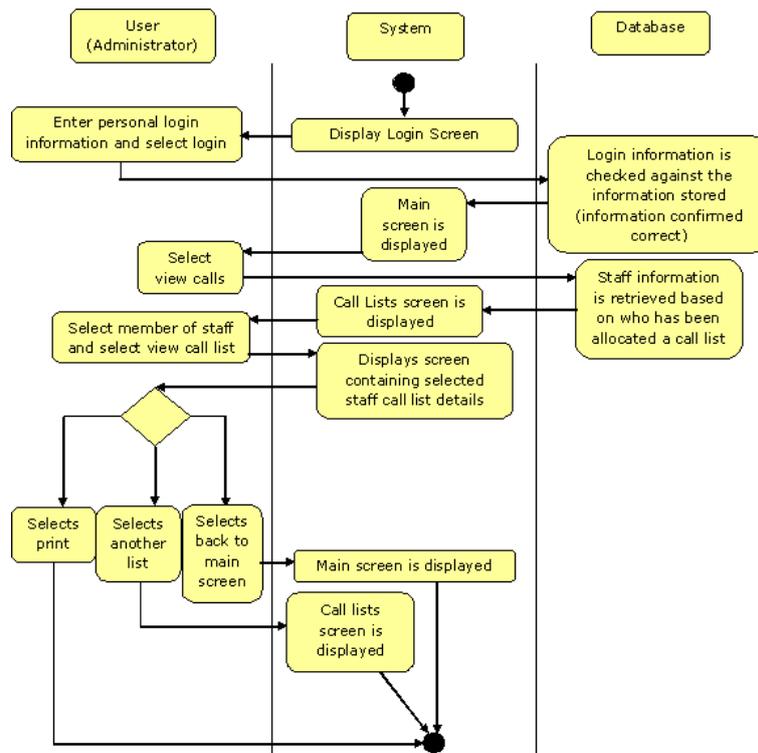


Figure 8.View nurse call lists – Administrator

2.2 User Stories:

(Oestererich, 2002)

I decided to use user stories as part of my design stage for a number of different reasons. They are short descriptions of functionality that are told from the perspective of a user that are valuable to either a user of the software or the client. They will help as a reminder for when implementing the system. User stories will help capture what my client needs and wants from the system in a way they understand.

District nurse:-

Amy is a district nurse and has been for the past year. Amy aims to print off her call list for tomorrow. She has come into the office as 12pm in order to have time to view list. Amy likes to have her call list early as she can get equipment she needs and be organised for the day ahead.

Amy is confident whilst using computers and is used to a Windows operating system. Amy graduated from Cardiff University School of nursing a year and a half ago.

Once Amy was given her personal log in details she found it straight forward to log in. She located 'View Call List' and selected this button. She was then displayed her call list. Amy used the scroll bar to look down to the bottom of her call list. She clicked 'Print' in order to have a copy handy whilst she was driving around.

Amy read the paper copy of her calls and realised she had a new patient (George Green) and wanted to gain more information on the treatment required. Therefore Amy clicked 'Back to Main Screen' and then from the main screen 'View Patient's Details.'

Using the short description given to her on her call sheet she entered criteria into the text boxes (George Green) provided. She selected 'Search for Patient's details' and was immediately displayed with 2 entries. Using criteria from sheet she had not entered she was able to select the correct patient and select 'View details.' Amy was displayed with a detailed list of information for George Green and decided she wanted to print this information and therefore selected 'Print.'

Amy was happy with the information and call list she had gathered and decided she wanted to log out. Therefore she selected 'Back to Main Screen' and then from there 'Log out.'

2.3 Use Cases:

(Oestererich, 2002)

(Bennett, Skelton, Lunn, 2004)

I decided to create use cases as they can clearly separate the processes needed and breakdown each functional area ready for implementation. Each use case explains the main functions for each actor in order to achieve their individual goals of completing each task. Use cases can also display where some components can be re-used (can be seen in Figure 9 below where both actors share 'View Patient's details.')

Use cases are a great way to present my functional requirements in an easy to follow format. They can give an overall system scope which can be referred back to during the implementation stage. These can also be looked back at during the testing and evaluation stage in order to decide whether the main goals and functionality set here are actually met once the system is fully implemented.

Call Lists:-

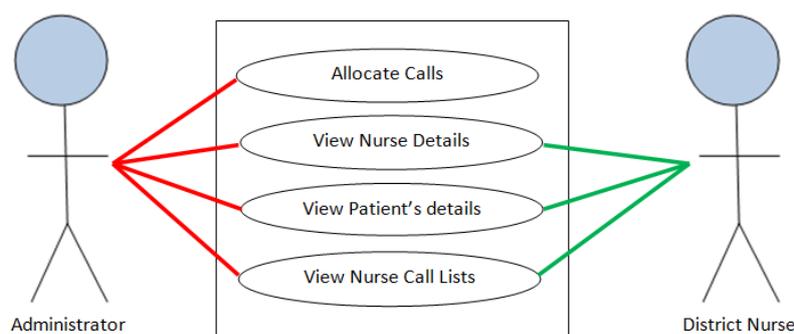


Figure 9. Use case

1.1 Use Case – Allocate Calls

Brief Description

This is the core use case of the project which allows the administrator to allocate calls to individual nurses. These allocated call lists are stored ready to be viewed later.

Actors

Administrator

Basic Flow

- 1) This use case starts after the administrator has logged in to the system.
- 2) A new screen will be displayed to user (administrator main screen), and then the user selects 'Allocate calls.'
- 3) A new screen will be displayed to user displaying instructions needed to allocate calls.
- 4) Allocate calls:-
 - 4.1) User selects nurses on duty using the tick boxes provided.
 - 4.2) User selects 'Create call list' and a new screen will be displayed.
 - 4.3) User enters date of call list.
 - 4.4) User checks members of staff are correct using the scroll bar.
 - 4.5) User selects 'create call list' and a new screen is displayed.
 - 4.6) A confirmation that the call list has been created is displayed to user.
 - 4.7) User can then go back to main screen by selecting 'Back to Main Screen.'

Priority

This is high priority because the program cannot work without this use case. (Core functionality wouldn't be completed without this.)

Pre-Conditions

The administrator (user) must be logged in.

Post-Conditions

If the user case is successful then the call lists database will be updated.

1.2 Use Case – View Nurse Details

Administrator

Brief Description

This allows the administrator to view all staff details.

Actors

Administrator

Basic Flow

- 1) This use case starts after the administrator has logged in to the system.

- 2) A new screen will be displayed to user (administrator main screen), and then the user selects 'View staff details.'
- 3) A new screen will be displayed to user displaying a list of district nurses to select from.
- 4) View staff details:-
 - 4.1) User selects a nurse from the list displayed using the radio buttons.
 - 4.2) User selects 'View details' and a new screen will be displayed.
 - 4.3) The selected staff member's details are displayed to user and user can use scroll bar to see entire details.
 - 4.4.1) User can view another staff's details by selecting 'Another staff' and screen is directed back to 4.1.
 - 4.4.2) User can also print these details by selecting 'Print.'
 - 4.4.3) User can go back to main screen by selecting 'Back to Main Screen.'

Priority

This is an important part of the system as all staff details can be viewed and therefore can be contacted if need be.

Pre-Conditions

The administrator (user) must be logged in.

Post-Conditions

If the user case is successful then all staff details can be viewed and printed.

District Nurse

Brief Description

This allows the nurse to view their personal details.

Actors

District Nurse

Basic Flow

- 1) The use case starts after the nurse has logged in to the system using their personal log in details.
- 2) A new screen will be displayed to user (nurse main screen), and then the user selects 'Personal Details.'
- 3) A new screen will be displayed to user displaying the users' personal details. These details can all be seen by using the scroll bar.
- 4) These details can be printed by selecting the 'Print' button.
- 5) The user can go back to main screen by selecting 'Back to Main Screen.'

Priority

This is an important part of the system as each member of staff can view their own personal details and can see clearly if any details need changing.

Pre-Conditions

The nurse (user) must be logged in.

Post-Conditions

If the user case is successful then all staff can view their own personal details and print them if required.

1.3 Use Case – View Patient’s Details

Brief Description

This allows the nurse and administrator to view their patient’s personal details. These details are more detailed than the information in the call lists.

Actors

District Nurse and Administrator

Basic Flow

- 1) The use case starts after the nurse or administrator has logged in to the system using their personal log in details.
- 2) A new screen will be displayed to user (either nurse or administrator main screen), and then the user selects ‘View Patient’s details.’
- 3) A new screen will be displayed to user instructing them to input details where required.
 - 3.1) User enters details for Patient Name, Family Name, Town and/or Treatment.
 - 3.2) The user selects ‘Search for Patient’s details.’
 - 3.3) A new screen will be displayed to user showing whether a patients’ details have been found or not.
 - 3.3.1) If details have not been found then an error message will be displayed and then user can use ‘Another Patient’ to search again by going back to 3.1.
 - 3.3.2) If details have been found then a list of those found are displayed to user.
 - 3.3.2.1) User can then use scroll bar to look through list found and select particular patient using radio button.
 - 3.3.2.2) User then selects ‘View details.’
 - 3.3.2.3) A new screen is displayed to user displaying the chosen patient’s details. These details can be seen by using the scroll bar.
 - 3.3.2.4) User can print these details by selecting ‘Print.’

3.4) User can choose to search for another patient by selecting 'Another Patient' and will be directed back to 3.1.

3.5) User can go back to main screen by selecting 'Back to Main Screen.'

Priority

This is an important part of the system as both administrator and nurse can view patient details. This information is important as it contains a more detailed version of descriptions found on call lists. For example a key code may not have been included on call list but is needed to get entry to a property to treat a patient.

Pre-Conditions

The nurse or administrator (user) must be logged in.

Post-Conditions

If the user case is successful then all staff and administrator can view personal patient's details.

1.4 Use Case – View Calls

Administrator

Brief Description

This allows the administrator to view call lists for each member of staff working one particular day; the following day.

Actors

Administrator

Basic Flow

- 1) This use case starts after the administrator has logged in to the system.
- 2) A new screen will be displayed to user (administrator main screen), and then the user selects 'View Calls.'
- 3) A new screen will be displayed to user displaying a list of the district nurses working the following day. (The calls have to be allocated beforehand so the right day is visible to user.)
- 4) View call lists:-
 - 4.1) User selects a nurse from the list displayed using the radio buttons.
 - 4.2) User selects 'View Call List' and a new screen will be displayed.
 - 4.3) The selected staff member's call list is displayed to user and user can use scroll bar to look at entire call list.
 - 4.3.1) User can view another call list by selecting 'Another List' and screen is directed to 4.1.
 - 4.3.2) User can also print these details by selecting 'Print.'
 - 4.3.3) User can go back to main screen by selecting 'Back to Main Screen.'

Priority

This is an important part of the system as call lists are the main functionality of the system.

Pre-Conditions

The administrator (user) must be logged in and must have allocated the calls for the next day if the user wants to view correct calls for next day.

Post-Conditions

If the user case is successful then all staff working on a particular day has a call list assigned to them which can be viewed and printed if required.

District Nurse

Brief Description

This allows the nurse to view their personal call lists for the upcoming day.

Actors

District Nurse

Basic Flow

- 1) The use case starts after the nurse has logged in to the system using their personal log in details.
- 2) A new screen will be displayed to user (nurse main screen), and then the user selects 'View Call List.'
- 3) An new screen will be displayed to user displaying the users' personal call list for the next day. These details can be seen by using the scroll bar.
 - 3.1) These details are set into two parts one for bloods and one for remaining calls.
 - 3.2) User can print this list by selecting 'Print.'
 - 3.3) User can go back to main screen by selecting 'Back to Main Screen.'

Priority

This is an important part of the system as this is the main aspect of the core functionality. A call list is provided for each member of staff in advance.

Pre-Conditions

The nurse (user) must be logged in.

Post-Conditions

If the user case is successful then all staff can view their own personal call lists and these can printed if required.

2.4 Database Design

(Bennett, Skelton, Lunn, 2004)

I used the Class diagram from the Interim Report to create a database structure to include a number of tables to handle all the data required. Whilst implementing the system I required a number of temporary tables to handle data. Here is a new class diagram including the temporary tables I used:-

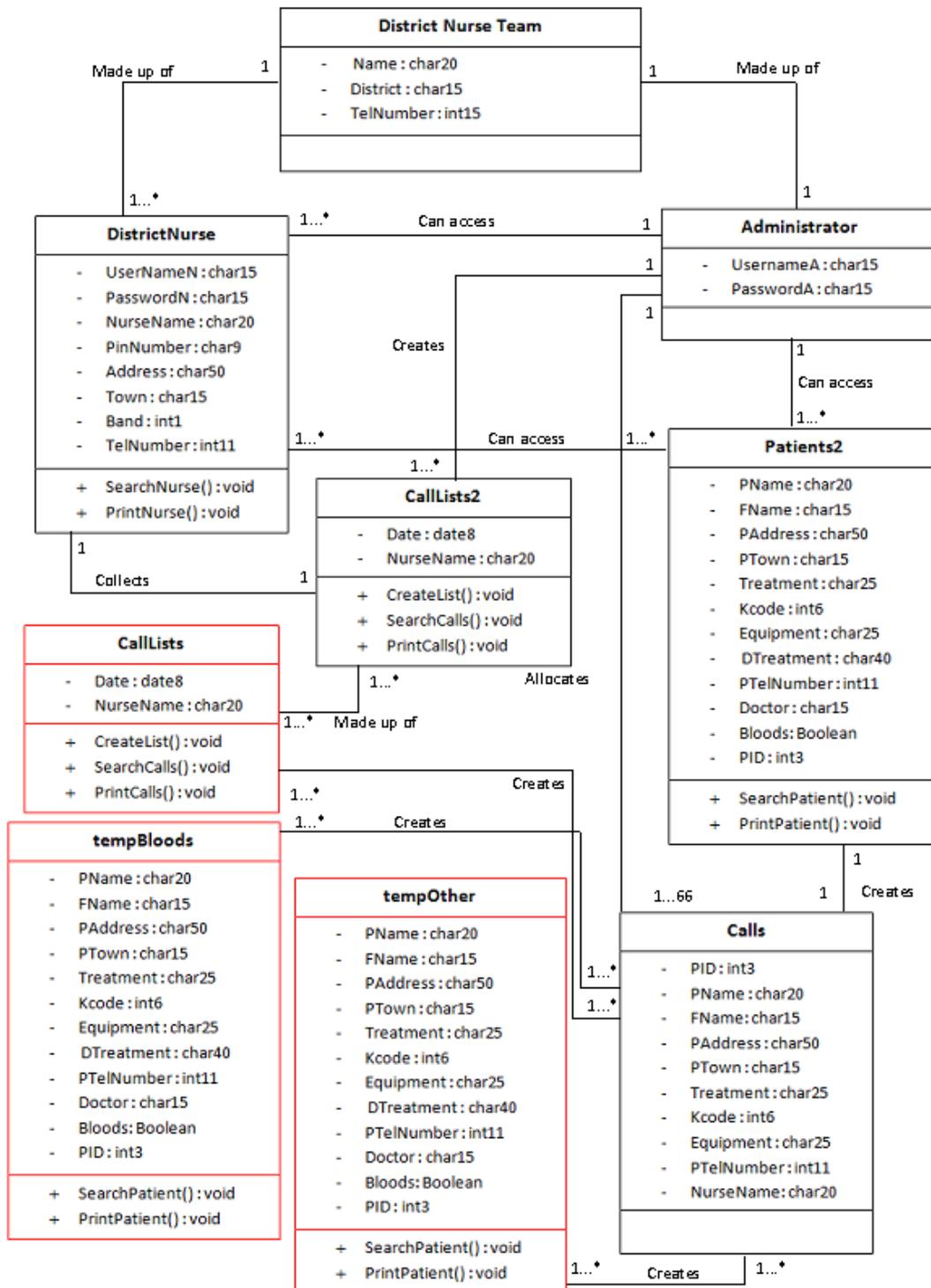


Figure 10. Temporary Tables included in class diagram

I also needed to adapt my original class diagram so it now looks like:-

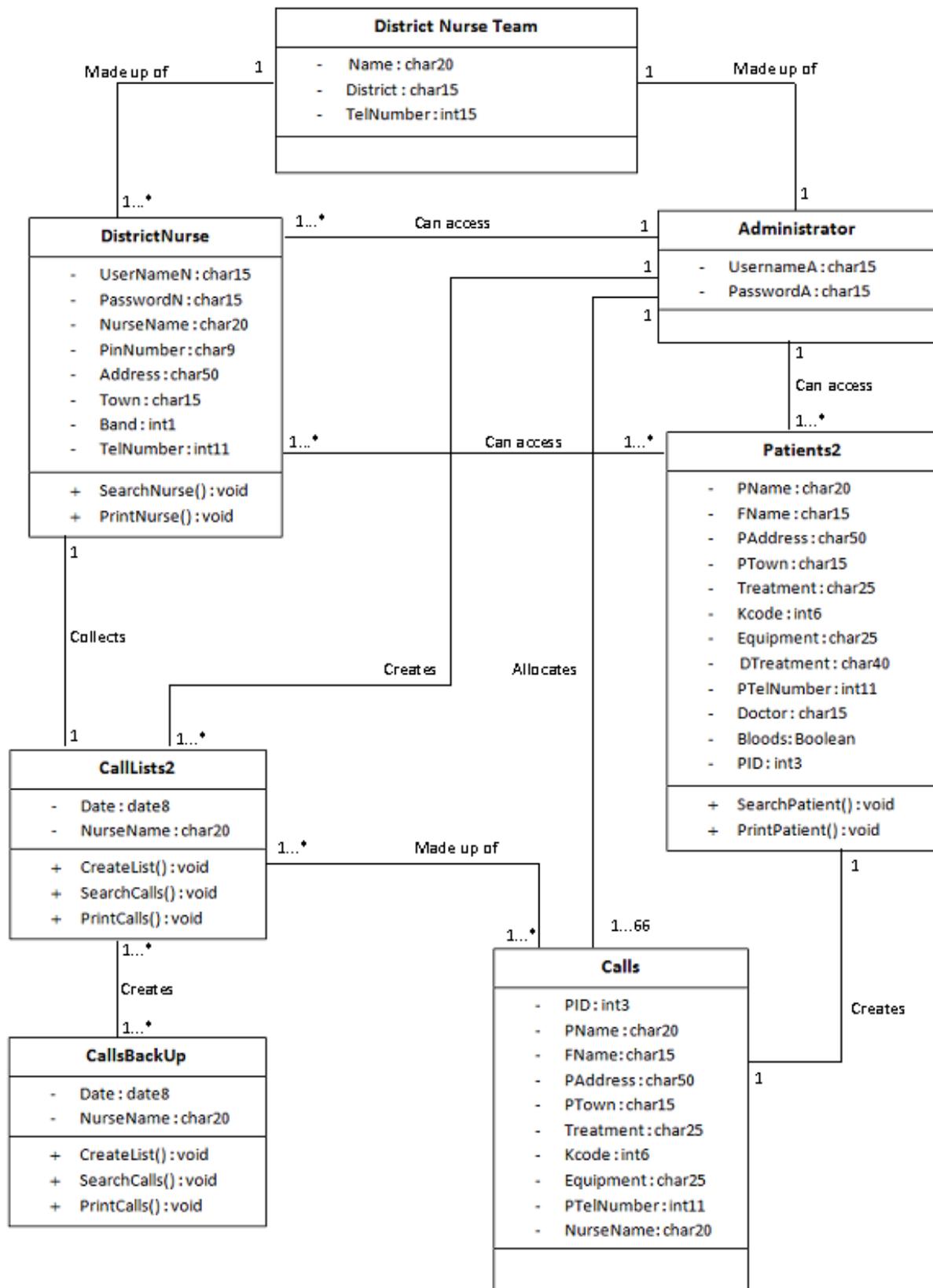


Figure 11. Updated class diagram

2.5 Algorithms

(Using Figure 4 from Interim Report)

In the interim report I looked into a number of different algorithms which would be useful within my system. I needed to use a number of different algorithms in order to create an accurate call list, based on the requirements collected during the early stages of my project. Here are the numerous algorithms I selected as being the most efficient to fit the purpose of my challenge:-

Vehicle (capacitated) Routing Problem (*Beasley, J.E, ()*) is a useful algorithm for the part of my system where calls are allocated to individual members of selected staff. This algorithm will be useful for sorting a number of lists in my database which need to be sorted based on their geographical location. For instance:-

- The lists of patients will need to be sorted based on the town saved for each patient. This will allow for members of staff to have their calls grouped together based on their location and limit the time spent driving between calls. This will eliminate staff having to drive from one area to another and back; for example having one call in Blackwood then having one in Ystrad and then having to drive back to Blackwood for another call.
- The list of selected working staff will need to be sorted based on their town in order to allocate calls most efficiently. Once staff list is sorted then they can be matched to the sorted patient list and this should lead to each staff member having a list which is closely matched to their geographical location as possible. This will be done as best as possible, as sometimes there may not be a member of staff that lives in an area covered by the District Nurse Team. For example a member of staff could live in Rhymney but travel to work in Bargoed and therefore the patches Bargoed covers with not cover that nurses' personal geographical location.

Divide and Conquer Algorithms (*Wiseman, R, 2012*) is a useful algorithm for the part of my system where calls need to be separated and sorted and then re-joined to create one call list. For instance:-

- The list of patients will need to be split into two lists; one *Bloods* and one *Other Calls*.
- Each list then needs to be sorted into ascending order based on their 'Town.'

- Once both lists have been sorted they need to be combined together into one list and each patient is then assigned a member of staff based on the information sorted in the staff working table.

3 Implementation

(Full set of code – Appendix F)

3.1 Templates:

Within the 'Evaluation' of the Interim report I stated I would look into Templates in more detail. I sat with my client (district nurse) and she gave me details of what she wanted from the system and I created these templates to cover all aspects of what she required.

Here is a more detailed version of templates compared to set of examples that can be found within Interim report. Figure 12 can be used by both the administrator and district nurses:-

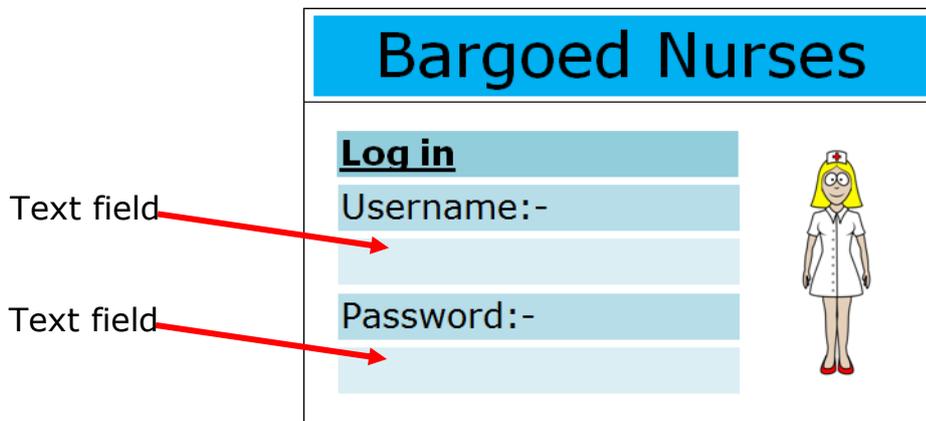


Figure 12. Log in screen

Administrator

Once the administrator has logged in this is the main menu they can see:-

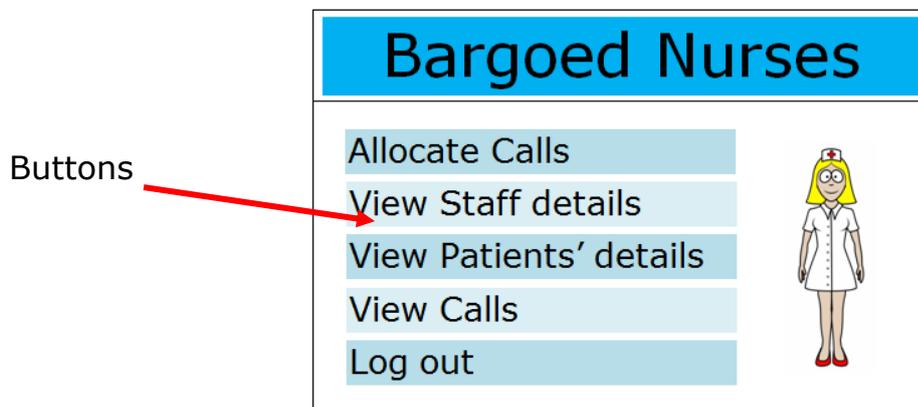


Figure 13. Main menu - Administrator

The administrator can then select from the list of options. For example if the 'Allocate Calls' button is selected this is the screen displayed to administrator:-

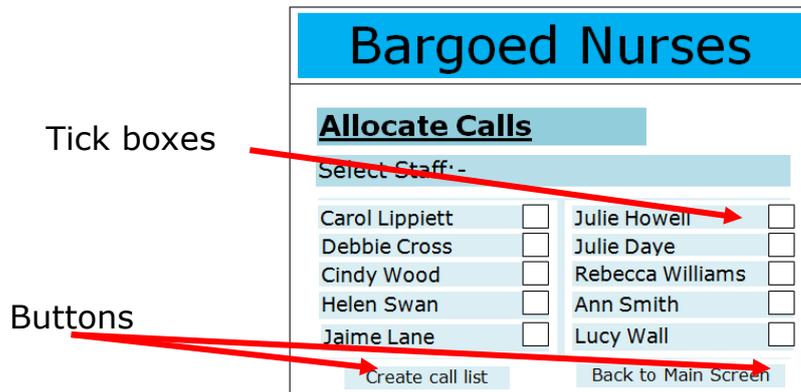


Figure 14. Allocate calls – main screen

The administrator then can look at who is on duty from the call duty sheets which are created 2 months in advance. The administrator can then select these members of staff by selecting the tick boxes:-

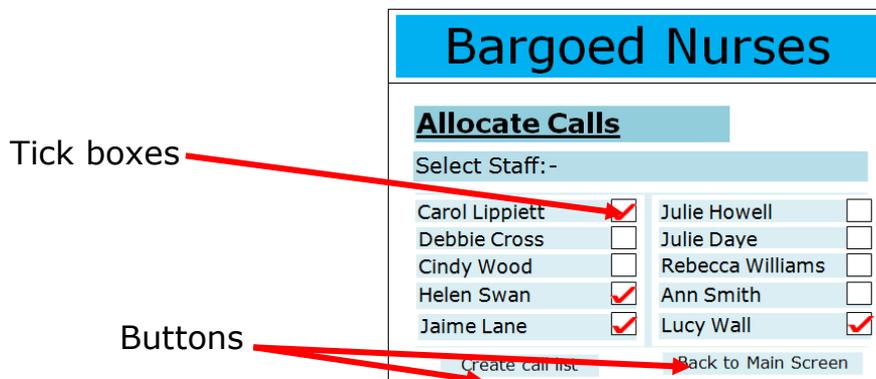


Figure 15. Allocate calls - check

Once staff are selected the administrator can then select 'Create call list':-

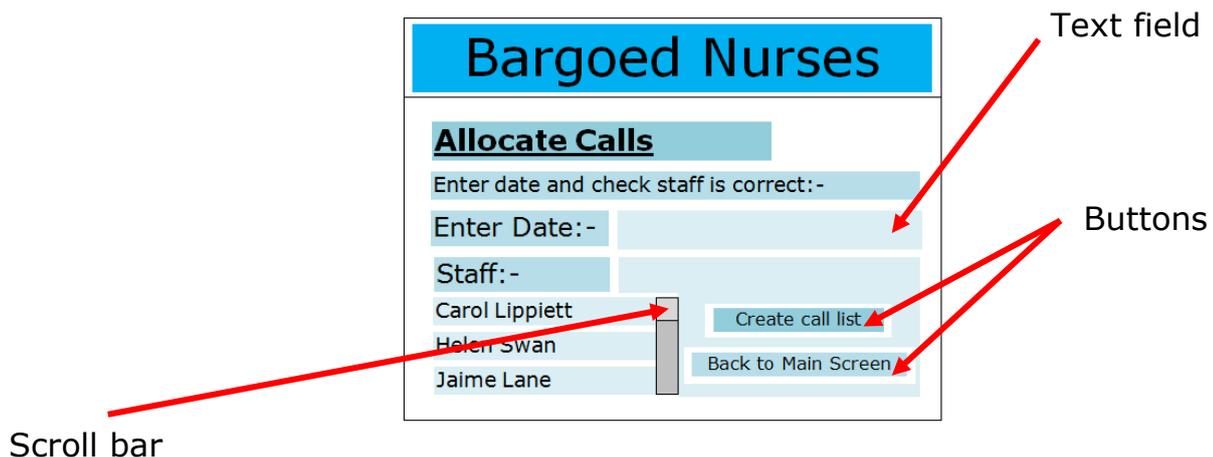


Figure 16. Allocate Calls – date entry

Once date has been entered and 'Create call list' has been selected this screen is displayed to user:-

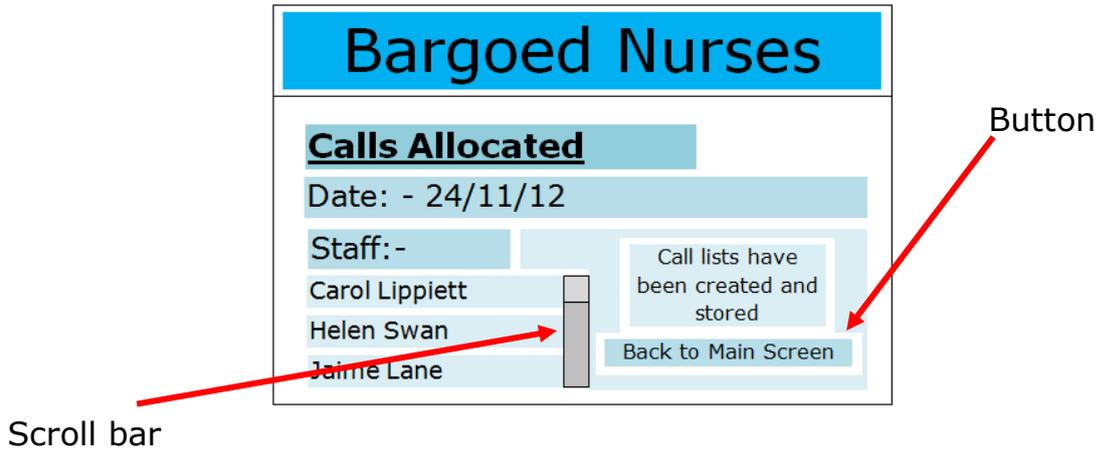


Figure 17. Allocate calls - confirmation

The administrator can then select 'Back to Main Screen.' From the main screen the administrator can select 'View Staff details':-

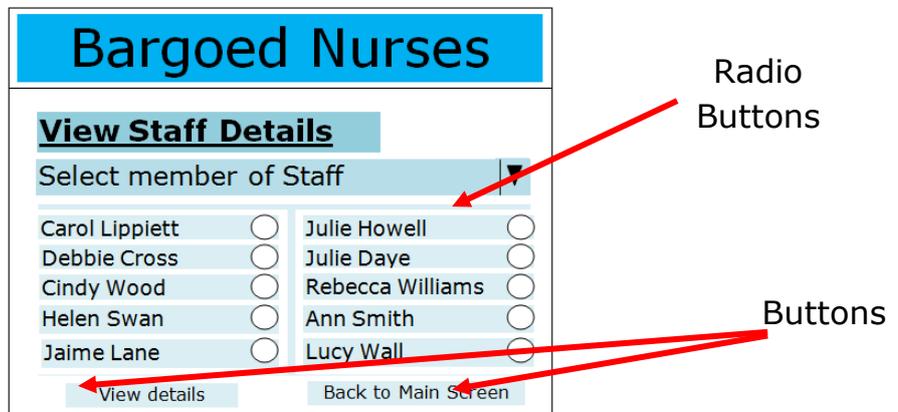


Figure 18.view staff details - main screen

From this list the administrator can select a member of staff:-

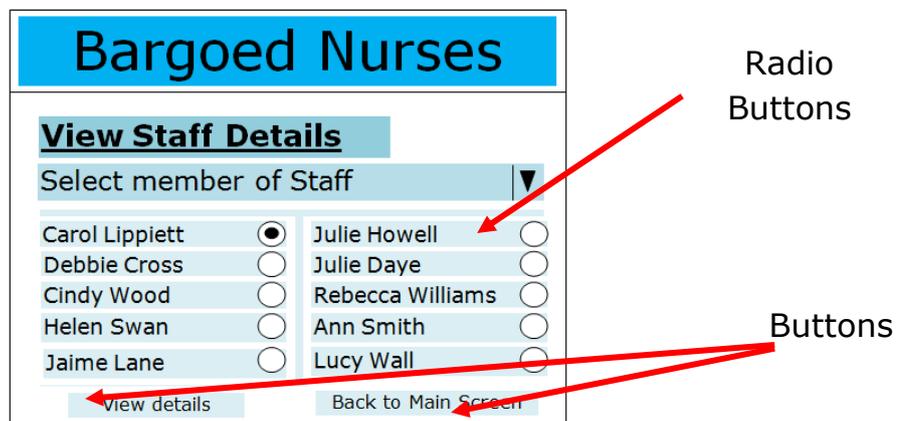


Figure 19.view staff details - selection

The administrator can select 'View details' and this screen is displayed to user:-



Figure 20.view staff details - individual staff details

The administrator can select 'Print' and these details will be printed off and can be easily read. Or the administrator can select 'Another staff' where the previous screen will be displayed to user.

If I manage to move onto additional features of the system this would have an effect on this part of the system. The administrator will have the option to update/delete details from here instead of using the main database.

Or the administrator can be directed back to main screen and can select 'View Patient's details' from there:-

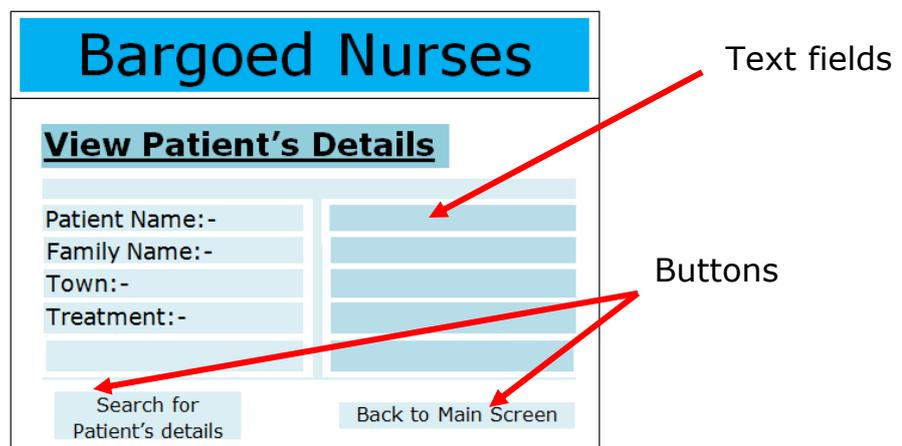


Figure 21.view patient's details - main screen

The administrator can enter known details for a patient and then click 'Search for Patient's details' to see if any of the criteria entered matches information stored in the database:-

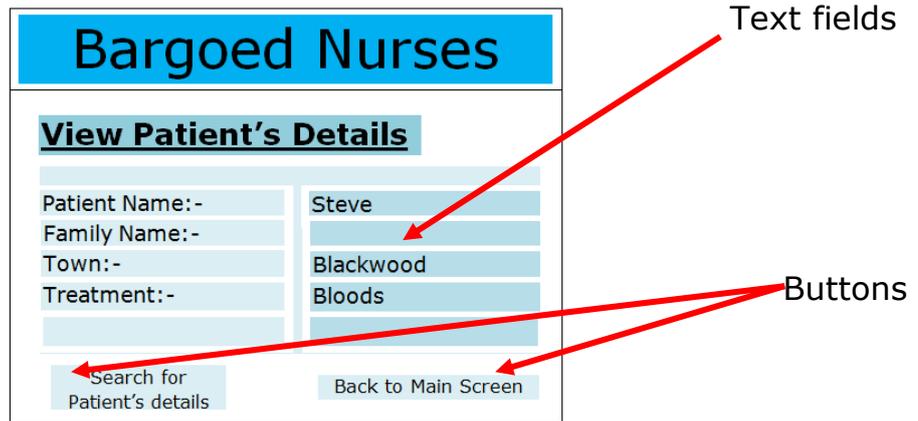


Figure 22.view patient's details - criteria entered

The administrator can then select 'Search for Patient's details':-

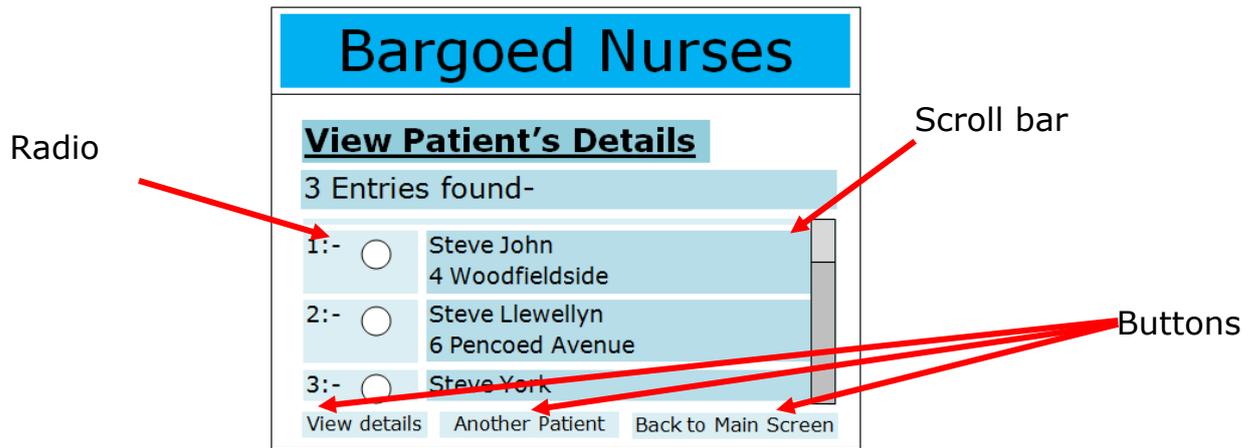


Figure 23.view patient's details - matched criteria

The administrator can then select an entry from the list as shown below or can search for another patient:-

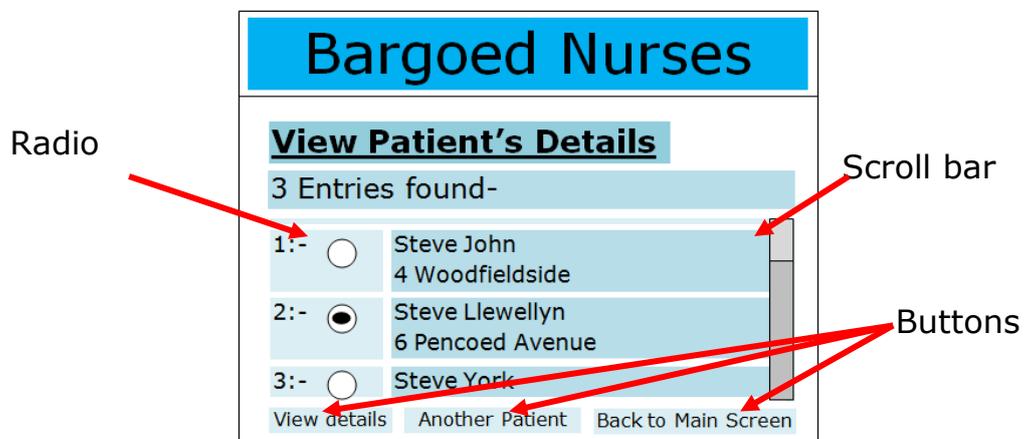


Figure 24.view patient's details - patient selected

The administrator can select 'View details':-



Figure 25.view patient's details - patient's details

These details contain in depth information which is not provided on call lists. The administrator can print these details and also search for another patient.

Or direct back to main screen where they can select 'View Calls':-

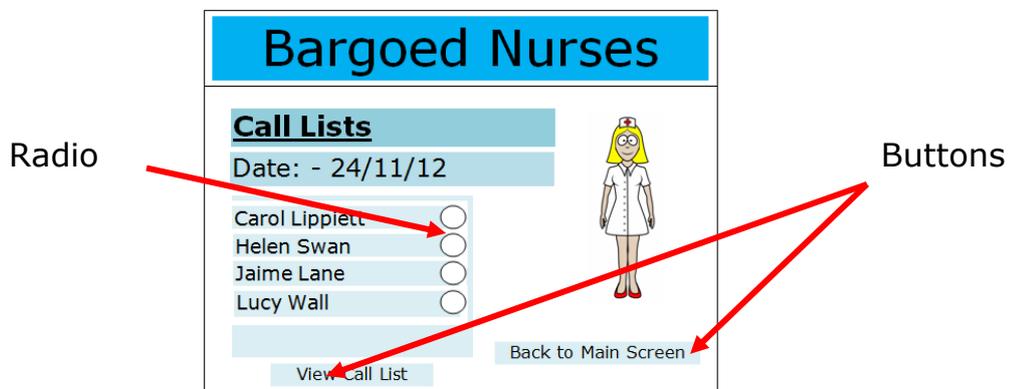


Figure 26.view calls - main screen

Details will only be displayed here if calls have been allocated otherwise the previous days calls will be displayed. The administrator can select from the list of nurses:-

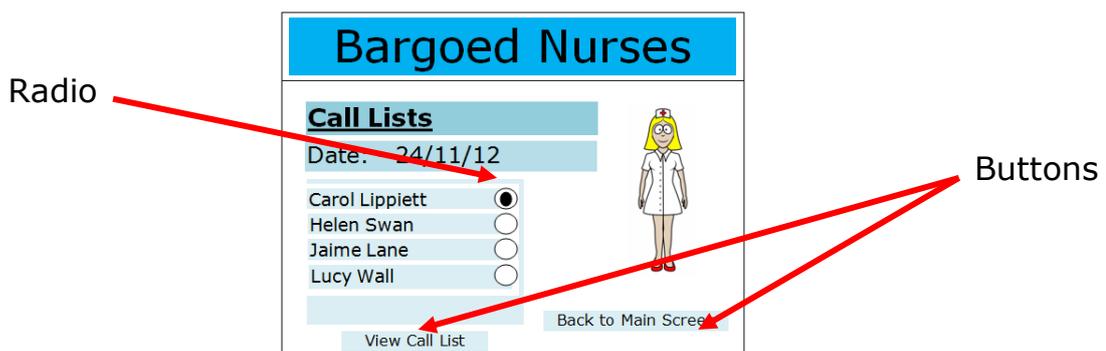


Figure 27.view calls - staff selected

The administrator can select 'View Call List':-

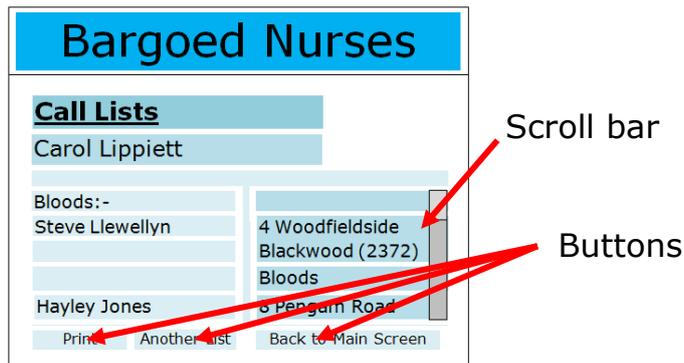


Figure 28.view calls - staff call list

The administrator can print this list or select another staff to view their call list for the next day.

Or can direct back to main screen where they can select 'Log out' and be directed back to log in screen.

Staff Nurse

Once the nurse has logged in this is the main menu they can see:-

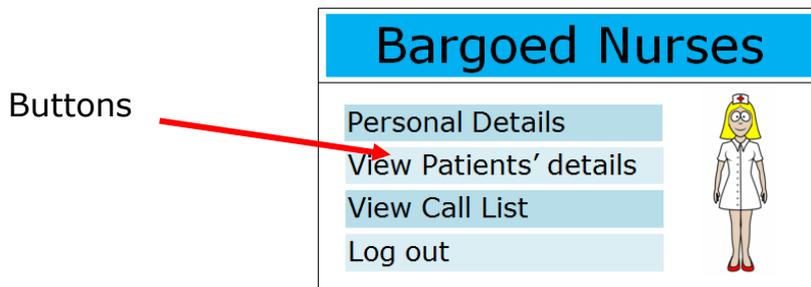


Figure 29.main screen - Nurse

The nurse can then select 'Personal Details.' For example if Carol Lippiett has logged in then when Carol selects 'Personal Details' she will see:-

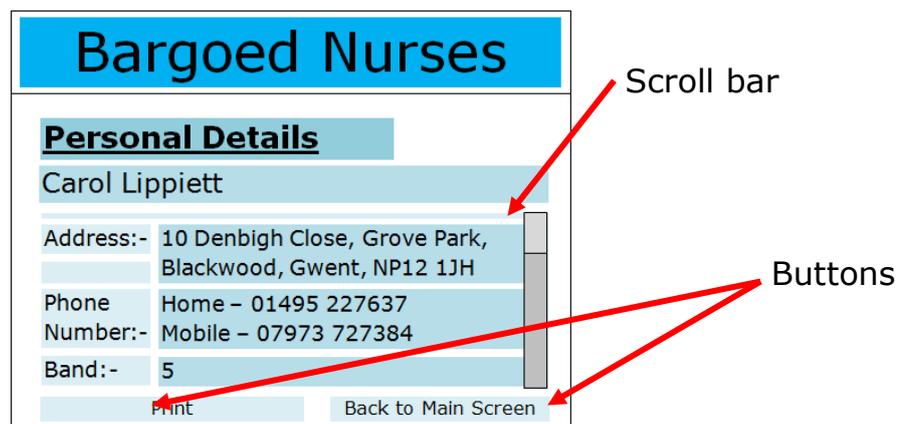


Figure 30.personal details main

The nurse can check their details are correct and can also print these details so they can be given to those who need them. The nurse can then be directed back to main screen where they can then select 'View Patient's details':-

This will be the same as the administrator.

After viewing patients details and then directing back to main screen the nurse can select 'View Call List.' This will be a personalised list based on the nurse that is logged into the system. For example using Carol Lippiett as the nurse logged into the system and based on the nurse logging in after 12pm when the call list should have been created by the administrator:-

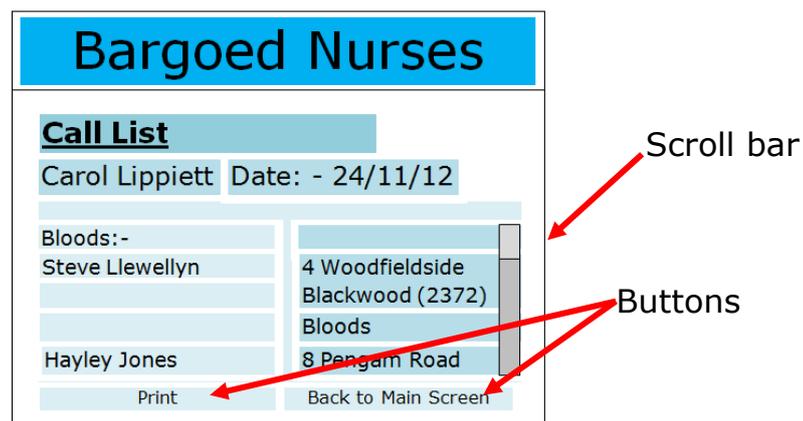


Figure 31.view personal call list - main screen

The nurse can check the date of call list which should be for the next day and then select 'Print' in order to have a printed copy they can use while in the car. The call list will first display bloods calls and then further calls which can be carried out after 12pm. The list should try to be grouped based on areas such as Blackwood calls being carried out one after another in order to save time and optimise time with patients.

If the nurse needs more details than given in the call list; for example a longer home address; they can look into patient's details from the main menu.

Once the nurse has carried out everything they need to in the system then they can direct back to the main screen and log out.

3.2 Implementation:

During the implementation stage I had weekly meetings with both my supervisor and client. Weekly meetings meant I could get important input into the functionality of the system in order to create a system that fulfilled the ideas of my proposed solution set in the interim report. I was able to get feedback as I implemented each part in order to use the ideas of both my supervisor and client to create an overall synthetic looking system along with the important core functionality.

3.3 Creating Interfaces:

Using Appendix E along with the template figures from above (Figure 12- Figure 31) as a guideline I created a number of interfaces; however some changes were made for the benefit of the user. Some of these changes included:-

- 'Allocate calls' interface (Figure 14) no longer has tick boxes to select nurses working, instead has two scroll lists in which nurses can be passed into the other to build a list of nurses. I chose to do this as there is an unknown set size of nurses employed as some nurses stored in database could be bank staff (staff that is called in to work on a temporary basis, for example to cover a nurse which is off sick.) Therefore all nurses can be easily viewed and then selected and passed into second list using a single button. The user can clearly see the new list and if a name is wrong this name can be easily removed in order to make an accurate list using a single button again.
- Instead of a confirmation Frame for 'allocated calls' (Figure 17) I have decided to put a JOptionPane confirmation message, and then user is directed back to the main screen. I think this makes the program more useable instead of the user having to direct themselves back to the main screen after calls have been allocated which seems to be secondary function which can be easily eliminated.
- 'View staff' (Figure 18) based on an admin log in has been changed so that details for staff members can be entered and then based on information stored in database the correct details can be displayed to user. This is another single button function making it clear to user what is required.
- 'View calls' (Figure 26) based on an admin log in has been changed into a scroll list of working staff based on details stored in the database. The user can then double click the staff member they wish to view their call lists and then the list is displayed to user.

- Changes were made to those interfaces where details are viewed such as 'patient details' (Figure 25). Instead of scroll lists I have put them into JTextFields based on the fact the details are clearer to user rather than through scrolling details.
- I have included a sidebar in most interfaces to explain exactly to the user what is required for them to complete each task. This should help each task flow a little easier for the user. This means a user isn't required to remember how to complete each task and the sidebar is there as a reminder to those who become familiar with each task.

3.4 Low level solutions to high level problem: (Algorithms and temporary tables)

Using the algorithms stated in the design section (2.5) of this document it can be seen that I have used a number of algorithms which I found to be the most useful based on my system. The allocation of calls part of my system is a complex task and needed a number of algorithms in order to create the most efficient call list possible. I broke down the main task of call allocation using these algorithms and made the challenge a step by step by step task. Each step enabled me to use an algorithm to adjust my information stored in my database a small step at a time and then I used temporary tables to store the result of each algorithmic step. Here are the algorithmic steps I used and a description of the temporary tables used to store the results:-

Step 1:-

- I used the Vehicle (capacitated) Routing Problem in order to sort the stored list of working staff ('CallLists2') based on their graphical location:-
 - Administrator has selected the working staff and these are stored in 'CallLists2' table.
 - This table is then checked to make sure it contains 6 members of staff before being sorted based on the 'Town' associated with each staff member. For example Bargoed, Blackwood, Cefn Fforest and Ystrad. This result is then stored in the temporary table 'CallLists.'

//This method puts the selected staff in order based on where they live (Town)

```
public void CreateNurseOrder()throwsException
{
```

```

Class.forName("org.sqlite.JDBC");
Connection connt = (Connection)
DriverManager.getConnection("jdbc:sqlite:NurseSchedul
ing.sqlite");
Statement stt = connt.createStatement();
ResultSet rst = stt.executeQuery("SELECT * FROM
CallLists2");

int rowSize1 = 0;
while(rst.next())
{
    rowSize1++;
}

if(rowSize1 == 6)
{
    //If table is already populated then clear label before
    new insertion
    Class.forName("org.sqlite.JDBC");
    Connection connect = (Connection)
    DriverManager.getConnection("jdbc:sqlite:NurseSchedul
ing.sqlite");
    Statement state = connect.createStatement();
    state.executeUpdate("delete from CallLists");

    //Insert into call lists table the list of stored staff based
    on Town
    Class.forName("org.sqlite.JDBC");
    Connection conn = (Connection)
    DriverManager.getConnection("jdbc:sqlite:NurseSchedul
ing.sqlite");
    Statement sttt = conn.createStatement();
    sttt.executeUpdate("Insert into CallLists select
    CallLists2.Date,
    DistrictNurse.NurseName from CallLists2, DistrictNurse
    where CallLists2.NurseName = DistrictNurse.NurseName
    order by Town asc");
}
else
{

```

```
JOptionPane.showMessageDialog(frame, "Call List  
has not been created yet", "Call  
List", JOptionPane.ERROR_MESSAGE);
```

```
}  
}
```

Step 2:-

- I used the idea of the divide and conquer algorithm to firstly sort my 'Patients' table:-
 - I firstly needed to divide my table into *Bloods* and *Other Calls*. (This step will only occur if there are 30 *Bloods* patients and 36 *Other Calls* patients sorted.) These two lists are then stored in two separate temporary tables; 'tempBloods' and 'tempOther.'
 - Each 'tempBloods' and 'tempOther' are each grouped by their 'PTown' in ascending order.

```
//This method checks the amount of blood patients in table  
Patients2 is correct ... need 30
```

```
//Once 30 is found these are stored in temporary table  
tempBloods
```

```
//These are stored in ascending order
```

```
public void CreatePatientList()throwsException
```

```
{
```

```
    Class.forName("org.sqlite.JDBC");
```

```
    Connection connt = (Connection)
```

```
    DriverManager.getConnection("jdbc:sqlite:NurseSchedul  
ing.sqlite");
```

```
    Statement stt = connt.createStatement();
```

```
    ResultSet rst = stt.executeQuery("SELECT * FROM  
Patients2 where Bloods='TRUE'");
```

```
int rowSize1 = 0;
```

```
while(rst.next())
```

```
{
```

```
    rowSize1++;
```

```
}
```

```
if(rowSize1 > 30)
```

```
{
```

```

JOptionPane.showMessageDialog(frame, "There are too
many bloods patients", "Failure to create
list", JOptionPane.ERROR_MESSAGE);
rst.close();
connt.close();
}
else if(30 > rowSize1)
{
JOptionPane.showMessageDialog(frame, "There isn't
enough bloods patients", "Failure to create
list", JOptionPane.ERROR_MESSAGE);
rst.close();
connt.close();
}
else if(rowSize1 == 30)
{
//Clear the temporary bloods table
Class.forName("org.sqlite.JDBC");
Connection connect = (Connection)
DriverManager.getConnection("jdbc:sqlite:NurseSchedul
ing.sqlite");
Statement state = connect.createStatement();
state.executeUpdate("delete from tempBloods");

//Populate temporary bloods with new list of ordered
bloods patients
Class.forName("org.sqlite.JDBC");
Connection conn = (Connection)
DriverManager.getConnection("jdbc:sqlite:NurseSchedul
ing.sqlite");
Statementsttt = conn.createStatement();
sttt.executeUpdate("Insert into tempBloods SELECT *
FROM Patients2 where Bloods='TRUE' order by
PTownasc");
//Call the Create Patient List Other method
//Call the put Bloods in List method
CreatePatientListOther();
putBloodsInList();
}
}

//This method checks the amount of other patients in table

```

```

Patients2 is correct ... need 36
//Once 36 is found these are stored in temporary table
tempOther
//These are stored in ascending order
public void CreatePatientListOther()throwsException
{
    Class.forName("org.sqlite.JDBC");
    Connection connt = (Connection)
    DriverManager.getConnection("jdbc:sqlite:NurseSchedul
    ing.sqlite");

    Statement stt = connt.createStatement();
    ResultSet rst = stt.executeQuery("SELECT * FROM
    Patients2 where Bloods='FALSE'");

    int rowSize1 = 0;
    while(rst.next())
    {
        rowSize1++;
    }

    if(rowSize1 > 36)
    {
        JOptionPane.showMessageDialog(frame, "There are too
        many other patients", "Failure to create
        list", JOptionPane.ERROR_MESSAGE);
        rst.close();
        connt.close();
    }
    else if(36 > rowSize1)
    {
        JOptionPane.showMessageDialog(frame, "There isn't
        enough other patients", "Failure to create
        list", JOptionPane.ERROR_MESSAGE);
        rst.close();
        connt.close();
    }
    else if(rowSize1 == 36)
    {
        //clear temporary other table ready for new insert
        Class.forName("org.sqlite.JDBC");
    }
}

```

```

Connection connect = (Connection)
DriverManager.getConnection("jdbc:sqlite:NurseSchedul
ing.sqlite");
Statement state = connect.createStatement();
state.executeUpdate("delete from tempOther");

//populate tempOther with list of other calls
Class.forName("org.sqlite.JDBC");
Connection conn = (Connection)
DriverManager.getConnection("jdbc:sqlite:NurseSchedul
ing.sqlite");
Statementsttt = conn.createStatement();
sttt.executeUpdate("Insert into tempOther SELECT *
FROM Patients2 where Bloods='FALSE' order by
PTownasc");
}
}

```

- Once two new lists are sorted and stored in two separate temporary tables, then these lists can be combined to create an overall list of sorted *Bloods* and *Other Calls* patients. This new list is stored in the table 'Calls.'

```

//This method checks the amount of calls in Calls..if full then
deletes all
//Then inserts a copy of sorted bloods calls from tempBloods
into Calls
//NurseName is given the value 'Name' for the time being
public void putBloodsInList()throwsException
{
    Class.forName("org.sqlite.JDBC");
    Connectionconnt = (Connection)
    DriverManager.getConnection("jdbc:sqlite:NurseSchedul
ing.sqlite");
    Statementstt = connt.createStatement();
    ResultSetrst = stt.executeQuery("SELECT * FROM
Calls");

    int rowSize1 = 0;
    while(rst.next())
    {
        rowSize1++;
    }
}

```

```

    }

    if(rowSize1 > 1)
    {
        rst.close();
        connt.close();

        Class.forName("org.sqlite.JDBC");
        Connection connect = (Connection)
        DriverManager.getConnection("jdbc:sqlite:NurseSchedul
        ing.sqlite");
        Statement state = connect.createStatement();
        state.executeUpdate("delete from Calls");

        Class.forName("org.sqlite.JDBC");
        Connection conn = (Connection)
        DriverManager.getConnection("jdbc:sqlite:NurseSchedul
        ing.sqlite");
        Statementsstt = conn.createStatement();
        sttt.executeUpdate("Insert into Calls select PID, PName,
        FName, PAddress, PTown, Treatment, Kcode,
        Equipment, PTelNumber, 'Name' from tempBloods");
    }
    else
    {
        Class.forName("org.sqlite.JDBC");
        Connection conn = (Connection)
        DriverManager.getConnection("jdbc:sqlite:NurseSchedul
        ing.sqlite");
        Statementsstt = conn.createStatement();
        sttt.executeUpdate("Insert into Calls select PID, PName,
        FName, PAddress, PTown, Treatment, Kcode,
        Equipment, PTelNumber, 'Name' from tempBloods");
    }
    //Call put others in list method
    putOthersInList();
}

//This method inserts a copy of sorted other calls from
temporary other into Calls
//NurseName is given the value 'Name' for the time being

```

```

public void putOthersInList()throwsException
{
    Class.forName("org.sqlite.JDBC");
    Connection conn = (Connection)
    DriverManager.getConnection("jdbc:sqlite:NurseSchedul
ing.sqlite");
    Statementsttt = conn.createStatement();
    sttt.executeUpdate("Insert into Calls select PID, PName,
FName, PAddress, PTown, Treatment, Kcode,
Equipment, PTelNumber, 'Name' from tempOther");
//Call the add Nurses method
addNurses();
}

```

Step 3:-

- Then to finish the procedure of allocated calls:-
 - Using the temporary table for sorted members of staff 'CallLists' and the newly created sorted calls 'Calls.'
 - I insert into 'Calls' the name of selected staff members from 'CallLists' into the 'NurseName' column. This will result in each selected staff member having been assigned to 5 *Bloods* patients and 6 *Other Calls* patients.

```

//This method updates calls table with the list of working staff
public void addNurses()throwsException
{
    Class.forName("org.sqlite.JDBC");
    Connection conn =
    DriverManager.getConnection("jdbc:sqlite:NurseScheduling.sqlite");

```

```

String group1 = "Update Calls Set NurseName=? Where rowid
in(1,2,3,4,5,31,32,33,34,35,36)";
PreparedStatement a = conn.prepareStatement(group1);
a.setString(1,firstName);

```

```

String group2 = "Update Calls Set NurseName=? Where rowid
in(6,7,8,9,10,37,38,39,40,41,42)";
PreparedStatement b = conn.prepareStatement(group2);
b.setString(1,secondName);

```

```
String group3 = "Update Calls Set NurseName=? Where rowid  
in(11,12,13,14,15,43,44,45,46,47,48)";  
PreparedStatement c = conn.prepareStatement(group3);  
c.setString(1,thirdName);
```

```
String group4 = "Update Calls Set NurseName=? Where rowid  
in(16,17,18,19,20,49,50,51,52,53,54)";  
PreparedStatement d = conn.prepareStatement(group4);  
d.setString(1,fourthName);
```

```
String group5 = "Update Calls Set NurseName=? Where rowid  
in(21,22,23,24,25,55,56,57,58,59,60)";  
PreparedStatement e = conn.prepareStatement(group5);  
e.setString(1,fifthName);
```

```
String group6 = "Update Calls Set NurseName=? Where rowid  
in(26,27,28,29,30,61,62,63,64,65,66)";  
PreparedStatement f = conn.prepareStatement(group6);  
f.setString(1,sixthName);
```

```
a.executeUpdate();  
b.executeUpdate();  
c.executeUpdate();  
d.executeUpdate();  
e.executeUpdate();  
f.executeUpdate();  
}
```

- A call list will now have been created for 6 individual members of staff.

3.5 Main aspects of solution:

Here are the 'User Requirements' gathered from early research:-

- This project is concerned with prototyping a system to assist district nurses with their call scheduling problem.
- Help deliver clear set of calls for each member of staff.
- Call Lists:
 - Contain small number of bloods for each staff member to complete before 12pm.

- Contain a set of calls that can be completed before the shift ends.
- Even set of calls for each member of staff.
- Call lists should contain a description of the procedure which needs to be carried out and the equipment needed for each call.
- How to gain entry to patient's house. For example whether to use front or back door, whether the door will be open or requires a key or whether a key code is required.
- Whether another member of staff/carer need to be present at the time of call to assist with care.
- Whether a family member needs to be present at the time of call, this family member will need to be contacted previously in order to arrange time with nurse.
- The system is required to allow the nurse's to be able to view their call lists for each coming day.
- The system is required to be able to print/download sets of calls for each member of staff.
- The system is required to search through the database and retrieve call lists for each member of staff on a daily basis. The results are displayed to each user when requested.
- The system is required to search through the database and retrieve information about patients. This information will be more detailed than the information given in call lists. These patients' details are then displayed to user.
- Call lists are the primary function of the system and need to be scheduled for each member of staff on a daily basis. Each list needs to consist of a number of calls the nurse is required to complete each shift.

Using these requirements I have implemented a system to cover these points as best as possible. The proto-type system I have created can help assist Bargoed Nurses with their call scheduling problem. The system allows for interaction from users to complete different tasks based on the user that logs in.

The administrator can manually add information about staff, patients and further administrators directly into the SQLite database. The system then allows the administrator to select staff information which has been drawn from the database ready for call allocation. The system then uses the staff selected to allocate 11 calls to each based on their geographical location compared with each patient.

Each staff member has their own individual call list which they can clearly retrieve by accessing the system and following the correct procedure. They can also use the system to print this information to make the information portable whilst on call. Whilst viewing their call list through the system they can select any patient in order to find out further details should they require this option.

Each call list contains 5 *Bloods* calls followed by 6 *Other Calls*. These each are sorted based on geographical location (Town) and assigned to a staff member based on their Town also. This should mean that a staff member completes all *Bloods* calls in one area before moving to the next without having to return until the afternoon where *Other Calls* are completed. Call lists contain details which staff need whilst on call such as address, town, and a small description of what is required for each call and a keycode if one is available.

However I did not include the requirement where the call list should contain whether another member of staff/carer/family member needed to be present at time of call and I have included this option to be put into the detailed description for each patient rather than a separate option.

The system allows a staff member to log in to system using their personal details and then view their personal call list ready for the coming day. The system allows for each call list to be printed.

If these lists are printed out then it is the district nurses responsibility to bring the list back to the office and place in a 'safe bag.' This bag is then removed by an external company which disposes of this information correctly. This needs to be done in order to protect the information and not reveal this private/personal details to anyone outside the NHS.

4 Results and Evaluation

4.1 Results:

(Full set of Test Cases and Figures are included alongside this document as an appendix) – Appendix G and H respectively.

Test cases and Figures:-

Here is a simple diagram to show the functionality from administrator point of view:-

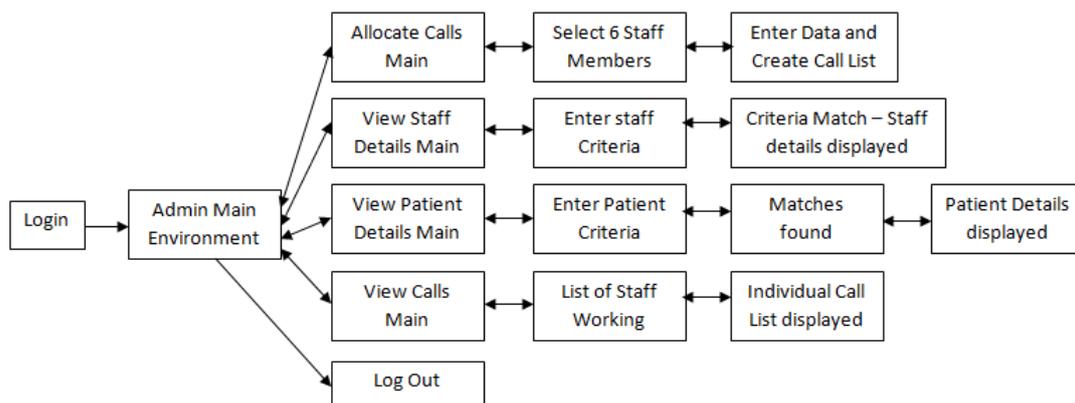


Figure 32. Administrator functionality

There are two main aspects of the system in which I will go into more detail; the allocation of calls by the administrator and then the viewing of call list by the nurse once allocation has occurred. Here is a test case for the main functionality (allocation of calls) following the path detailed above:-

Table 1. Call Allocation Functionality

Test Case Name/Id:	1	Test Purpose:	To test Call Allocation functionality	
Environment:	Windows platform, System			
Preconditions:	Username and password is entered by administrator these details are stored in the database			
User Data Requirements:	Username and Password stored in database and entered into required fields			
System Data Requirements:	Username and Password from Administrator Database tables			
Main Test Case Steps:				
Step No	Procedure	Expected Response	Actual Response	Step No
1	Run System – Log on Screen	Log in window is displayed	(Figure 33)	1
2	Enter Administrator log in details and select ‘Submit’	Main Administrator environment is displayed	(Figure 34)	2
3	Select ‘Allocate Calls’	Main Allocation screen is displayed	(Figure 35)	3
4	Select a member of staff from left hand list and use the ‘Move Nurse’ button to move staff to right and hand and continue to do this until 6 names are reached	None	None	4
5	Select ‘Create Call List’	Date Allocation screen is displayed	(Figure 36)	5
6	Enter date is field provided	None	None	6
7	Select ‘Create Call List’	Confirmation message displayed to user. On selection of ‘OK’ from this message the Main administrator is displayed.	(Figure 37)	7
Comments:				
There is no confirmation message when details are entered correctly				
<i>Data used;For an administrator:- username: becWil, password: becW(These are correct details)</i>				
<i>Members of staff selected to be in right hand list:- Charlotte Lippiett, Jeff Lippiett, Lauren Speicher, Jazz Bartlett, Josh Phillips, Carol Jones</i>				
<i>Data used for date entered: - 09/08/2013</i>				
Related Tests:				
Author:	Charlotte Lippiett	Checker:	Charlotte Lippiett	

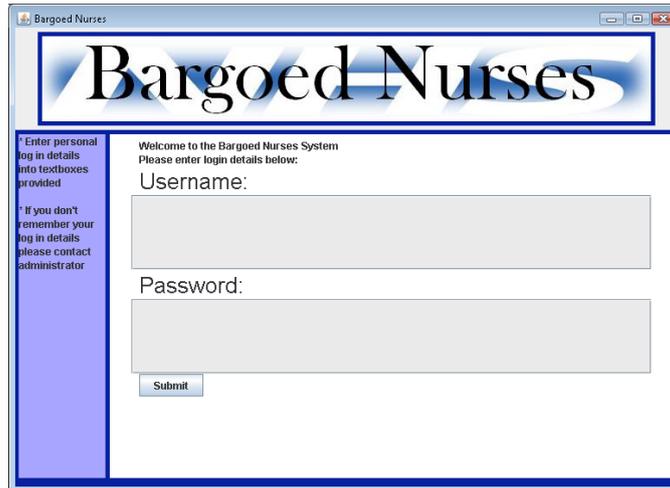


Figure 33. Main log in screen (used by both the administrator and district nurses)

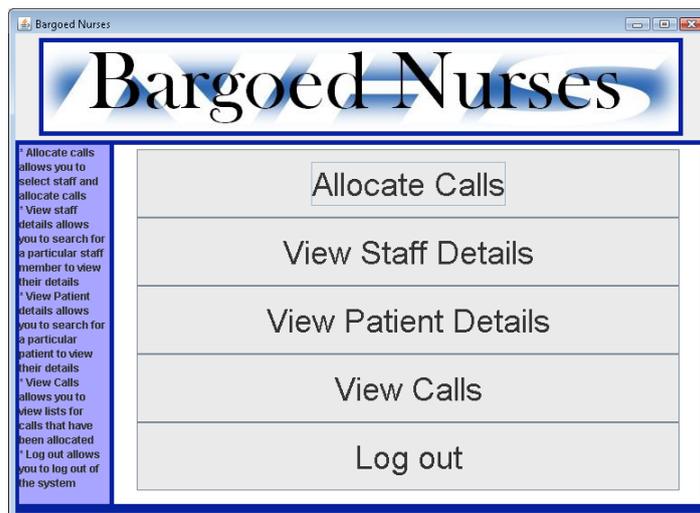


Figure 34. Main administrator screen

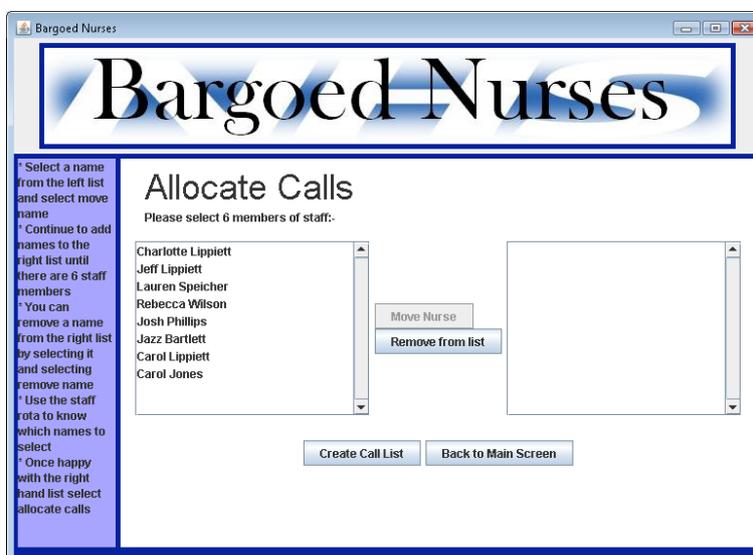


Figure 35. Main screen for allocation of calls



Figure 36. Check staff have been selected correctly and enter date

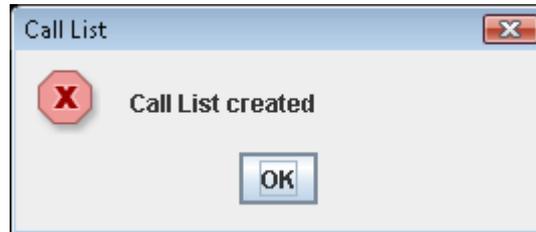


Figure 37. Confirmation message that the call list has been created

Here is a simple diagram to show the functionality from nurse point of view:-

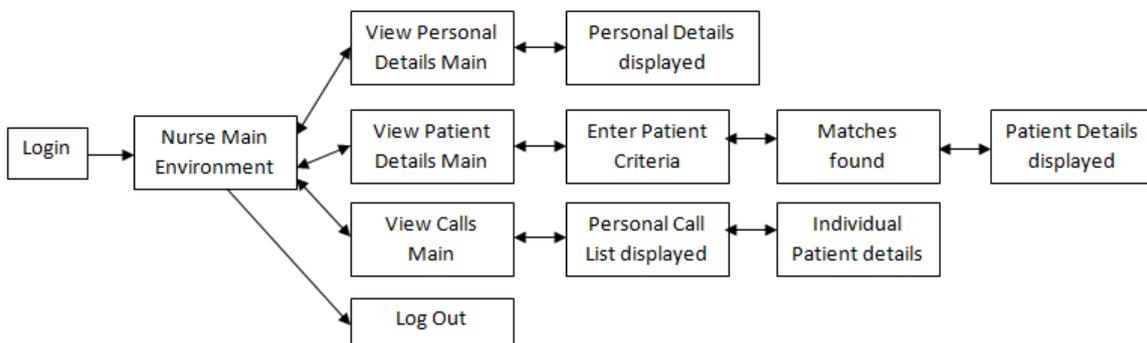


Figure 38. Nurse functionality

Here is a test case for a core function (viewing individual call list) following the path detailed above:-

Table 2. Viewing call list functionality

Test Case Name/Id:	2	Test Purpose:	To viewing call list functionality	
Environment:	Windows platform, System			
Preconditions:	Username and password is entered by District Nurse these details are stored in the database			
User Data Requirements:	Username and Password stored in database and entered into required fields			
System Data Requirements:	Username and Password from District Nurse Database tables			
Main Test Case Steps:				
Step No	Procedure	Expected Response	Actual Response	Pass/Fail
1	Run System – Log on Screen	Log in window is displayed		P
2	Enter District Nurse log in details and select ‘Submit’	Main District Nurse environment is displayed	(Screen shot – Fig 39)	P
3	Select ‘View Calls’	Main View Calls screen is displayed	(Screen shot – Fig 40)	p
4	Double click a patient from the list displayed	Screen displaying a more detailed version of the patient selected	(Screen shot – Fig 41)	P
Comments:				
Test Case 1 has been completed before this test case is valid				
There is no confirmation message when details are entered correctly				
<i>Data used;For an district nurse:- username: jeff, password: jeff(These are correct details)</i>				
<i>Date of call list:- 09/08/2013</i>				
<i>Patient selected from list: - carol Lippiett 10 denbigh close blackwood bloods null</i>				
Related Tests:				
Author:	Charlotte Lippiett	Checker:	Charlotte Lippiett	



Figure 39. Main nurse screen



Figure 40. Individual call list

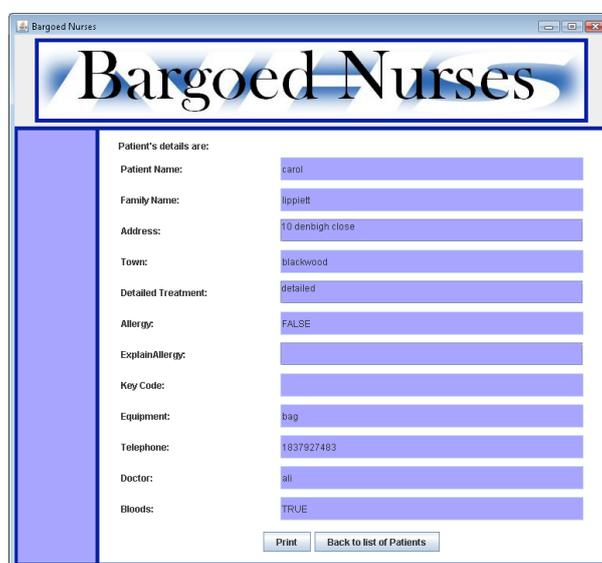


Figure 41. In depth details on selected patient

4.2 Testing system with real users (User Testing):

In order to test the system I used real users so that someone unfamiliar with the system can get their own feel for the system and I gained valuable feedback. I used my client as one real user (district nurse view) as they have background knowledge of what the system should do and have already had some contact with the system during the implementation stage.

I firstly asked my client (district nurse) to use the system from the point of view of a district nurse. My client already has prior knowledge of the system and therefore I simply gave them a document (Appendix A) containing log in details and a number of tasks to complete in order for them to test all aspects of the system. The user followed the tasks indicated to follow and then went on to answer a number of questions. Here are the answers to the questions asked:-

1) Overall look of the system from first sight?

- Clear and colour scheme works well; similar colours used in our uniforms
- Overall feel of the system is friendly and inviting.

2) How did the client find logging into the system?

- Having been given log in details user found it easy to follow sidebar information to input details into the correct text fields provided.

3) How did the client find viewing personal details?

- Labels on buttons were very purpose clear and therefore user found this step very straight forward.

4) How did the client find viewing patient details?

- Confusing to follow the task at first but once first details had been entered user remembered the steps very easily and was able to complete the remaining of the tasks very easily.

5) How the sidebar helps with directions?

- User found the sidebar very useful and consistent throughout the system. User used the sidebar as a helpful reminder when they forgot a step during the completion of a task.

6) How easy is the system to use?

- System provided clear names for buttons and labels throughout which made the system usable.
- Very easy to navigate through the system.

7) How helpful is the system?

- Very helpful, system provided useful notes for the user throughout the tasks. This system flowed well from page to page and showed good linkage.

8) Do you think I can do anything else to make the system better?

- For the purpose of this being a prototype user couldn't think of a way to make it better as it completes the necessary functionality very easily.

I secondly asked a Cardiff university student studying Pharmacy to test the core functionality of the system from the view point of a district nurse with no prior nursing knowledge. This tester also had no contact with the project during the implementation stage. I gave them a document (Appendix B) containing a brief overview of what was required of the system in order for the tester to understand what functionality was required from the system. This overview is followed by a number of tasks for the user to complete to test the functionality of the system. Once tasks were completed I went on to ask the tester a number of questions and here is the feedback:-

1) Overall look of the system from first sight?

- Professional looking system and a colour scheme user associate with NHS and though worked well.

2) How did the nurse tester find logging into the system?

- Log in resembles a log in usually seen on other applications and therefore user found it a straight forward aspects of the system.

3) How did the nurse tester find viewing personal details?

- This task only required one step and user found this very easy to complete.

4) How did the nurse tester find viewing patient details?

- User found the steps given very clear and therefore managed to complete the viewing patient details tasks very easily.

5) How the sidebar helps with directions?

- User found each sidebar very useful as when user lost track they used the sidebar in order to get back on track and complete the task in hand.

6) How easy is the system to use?

- The system is very consistent from page to page and therefore navigating and using the system was very easy as many steps were similar to others.

7) How helpful is the system?

- The system provided everything the system wanted and with ease of use. The user found the overall look of the system helpful as it stayed the same and did not change too much throughout its use.

8) Do you think I can do anything else to make the system better?

- Nothing based on the fact of unknown of nursing and the system seemed to serve the purpose stated in the overview given at the start of the test. The system provided the answers expected in the tasks in a clear format which the user found easy to interpret.

I thirdly asked another Cardiff university student studying English to test the call allocation functionality therefore acting as an administrator with no prior nursing knowledge. The tester had no prior knowledge of the project and did not have contact with the system during implementation stage. I gave the tester a document (Appendix C) containing a brief overview of what was required of the system followed by a number of tasks for the user to complete. Once tasks were complete I went on to ask the tester a number of questions and here is the feedback:-

1) Overall look of the system from first sight?

- From first sight the user expressed that the system looked concise and synthetic. The user has a background in designing for a newspaper and expressed how much they appreciated the overall colour scheme which flowed effectively.

2) How did the client find logging into the system?

- Due to having experience logging into previous systems the user found this step very easy.

3) How did the admin tester find allocating the call?

- Once user understood the concept of moving one selected nurse to the right hand list they were able to repeat this step and easily form a full list of working staff. They found the addition and removal of staff very simple to manoeuvre once completed once.
- The user appreciated the repetition of the created temporary list before creation of real call list. They found the overall complex step a very simple and effective procedure.

4) How did the admin tester find viewing staff details?

- User found steps were made very clear and acknowledged the sidebar information which they could use if they were to forget a step.

5) How did the admin tester find viewing patient details?

- Very similar to viewing staff details with the addition of a scroll list which they found simple to use to access patient details and revert back to list over and over again.

6) How the sidebar helps with directions?

- As each page required many different functions the user found each very useful in explaining each page and the functions they hold. User was grateful for the sidebar as they felt they did not have to remember all aspects of the system off by heart as they could refer to the sidebar for a helpful reminder.

7) How easy is the system to use?

- Throughout using the system the user expressed they found each task shared similar properties such as same name for buttons to return to main screen and therefore found the system easy to use. Due to some similar functionality the user thought the overall system work and flowed well.

8) How helpful is the system?

- The user found the system very helpful and found the best aspect was the sidebars as they could refer to them whenever they liked just to make sure they were completing each task correctly.

9) Do you think I can do anything else to make the system better?

- The user expressed that the system or I should make sure that the user is notified that all textfields are case sensitive.

Lastly I asked another Cardiff university student to test the call allocation functionality therefore acting as an administrator with no prior nursing knowledge however obtains knowledge of programming. I gave the tester a document (Appendix D) containing a brief overview of what was required of the system followed by a number of tasks for the user to complete. Once tasks were complete I went on to ask the tester a number of questions and here is the feedback:-

1) Overall look of the system from first sight?

- Looks unprofessional and needs UI tweaks
- Not like modern GUIs

2) How did the client find logging into the system?

- Had to ask if case sensitive

3) How did the admin tester find allocating the call?

- Had to look twice to find add. Would recommend arrow left and right to add or remove

4) How did the admin tester find viewing staff details?

- Confirmation message on successful find was pointless. Just show details.

5) How did the admin tester find viewing patient details?

- Double click was intuitive and I didn't read the details on the left.

6) How the sidebar helps with directions?

- Did not read once. Too much information

7) How easy is the system to use?

- Very easy, each screen has defined functionality and a specific task. This means only a couple of buttons per page making less clutter

8) How helpful is the system?

- Easy to use and allocate calls

9) Do you think I can do anything else to make the system better?

- Postcodes on address when viewing calls for satnav?

4.3 Evaluation:

The main requirement for this project was to create a system which allowed for selected members of staff to be assigned an individual call list to replace the existing manual system. I have managed to achieve the core functionality by creating a proto-type system to help manage the efficiency of call allocation.

The administrator will create a prerequisite set of data for patients and nurses directly from the database. Only the administrator can access, update or delete this data.

The core functionality consists of an administrator using the rota sheet (This is done two months in advance to show when staff are working) and interact with the system to select the working members of staff for a particular day using the rota sheet. The administrator can then input a date into the system and create a call list for each selected staff member.

The system organises a call list for each member of staff so that the calls are equally distributed. Each call list contains five bloods calls and six other calls; this information was gathered in the interim report. This limited number of calls is calculated in order to take into account travelling from call to call, dinner break and the drop off of blood samples before 12pm. Calls are distributed based on where each staff member lives and the proximity of calls. This is done to create the most efficient list for each member of staff and limit the amount of time spent in the car and optimise the time spent with patients.

Allocating calls is a large part of the proto-system which covers the main part of the core functionality. Another part which is included in the core functionality is that these call lists need to be accessed by all users of the system. Call lists can be viewed by both administrator and district nurse and these lists can be printed out for quick access when staff are unable to get to the office to access the system.

Each call list is made up of the information gathered directly from the client and stated in the initial plan and interim report. Each call list contains brief information on each patient along with any access codes needed to gain entry to a property.

Another part of the core functionality includes both the administrator and district nurses having access to detailed patient information should they require more details than those given in the call lists.

The administrator can also access detailed information on each member of staff. Each district nurse can only access their own personal information.

I worked closely with my client in order to make the processes of the system as smooth as possible. I wanted the system to be clear and usable for everyone. The system displays sidebars in order to help users navigate through the system so that they don't need to memorise steps or they can be used as a simple reminder.

Justification of choice of development language

As stated in the interim report I chose the main programming language to be JAVA. Java was a great platform to use as it provided a great solution to the challenge that the system needed to be user friendly, clear, precise and to fit the purpose. The GUI is a big part of the system in which java allowed myself to create an overall look to suit the client (District nursing team) and all its users. I worked closely with my client (mother/district nurse) in order to create an overall look for the system that would work best on a whole for all members of the team.

As some parts of my system were so similar such as having 'View Patients' for both Admin and Nurse login it allowed me to exploit the good aspects of Java. It allowed me to reuse code and change parts where appropriate for each aspect of the system.

Java enabled me to code small parts of my system and then compiles them to check each section was running accordingly. Each section of code

could be edited; compiled and run very easily which came in handy within my system which required a number of different functions to run independently. Java also enabled me to combine these sections of code so that they can coincide with each other to cover the different aspects which make up the main functionality of the system.

Limitations and Restrictions

Also stated in the interim report I used Java to develop an offline program (a prototype.) As my system is a proto-type the restriction of NHS data never occurred and the only data used within this system is sample data. Therefore no NHS guidelines were broken however if the system were to be expanded and taken online then this would be an issue that would need to be addressed. This will be discussed further in the 'Future Work' section. Due to my system being a proto-type system there are still some small problems with my system such as needed to be re-loaded if one nurse has already logged in and then out, however all the main functionality will run if this is done. My system is a proto-type and therefore the overall look of the system isn't professional and is limited in what I could implement, however it does provide a basis for future improvement.

Database

I stated in the interim report that I would use the relational database management system SQLite. Last semester I spent time familiarising myself with SQLite which was great as I managed to research more in depth the features of the database system and see how it would fit best within my system.

This single structure database was very useful as it was relatively easy to develop and then integrate into my system. It allowed me to create numerous tables and even during implementation when I required temporary tables the database system handled these new additions without a hassle.

Due to having used some aspects of SQLite previously I already had access to some integration code from a previous year's project and therefore this eliminated having to write lines and lines of code over and over. I simply had to adapt the section of code for the purpose of my system at each section.

Methodology Used

As declared in the interim report I decided to use a mixture of the waterfall model and the scrum methodology. This mixture has worked well as the project fell naturally into distinct parts to coincide with waterfall model. These parts were stated in the week by week plan which I attached to the end of the interim report (Appendix I). I found this week by week plan a useful tool to keep track of my project and make sure I was on top of my tasks. It allowed me to include other deadlines from other modules and balance it, so that due to the extra work load at certain times my project would not fall behind. Each week I could see clearly where I was meant to be (small goals based on sprints within the scrum methodology) and what needed to be done for the following week.

Throughout this term I have had weekly meetings with my supervisor which has worked well as it required small goals each week in which I would present to my supervisor my progress. This allowed me to get regular feedback and guidance on the work I had completed and steps I needed to take to reach the next goal.

Salient/Unique features

A salient feature of the system is the administration control where the administrator has full control of the data stored in the database. The administrator manually enters data into the database directly where details can be changed, moved or deleted. The administrator acts as the back end of the system as without the administrator controlling the data input the main functionality (allocation of calls) would not be able to be completed.

A unique feature of the system is where the database holds a copy of all calls in a new table so that all calls are saved and recorded. These can then be accessed at a later date should someone require to review a call. Some times in this career there are incidents which need to be reviewed and this table can hold the details to prove a nurse has attended a call or not. This can be used as evidence if need be in the case of a clinical incident which can lead to court cases.

Achievements

Achievements:-

I have worked alongside my client in order to create a proto-type to suit the needs of a team of district nurses. I used the information gathered in both the initial plan and interim report to create an efficient system that suits all level of computer literacy skills as indicated by my client. I introduced the sidebar on each frame to allow for all users to have clear vision on what was required from them in order to complete each task. I have managed to create a system that complies with the visual needs and practical needs of my client. The system allows for working staff names to be input into the system and a call list to be created for each member (Core functionality.) This call list can then be viewed and printed if required by each working staff member.

Additional functionality:-

(Call reallocation)

Using my created core functionality I looked into additional functionality where incidences occur where some members of staff are unable to work. This could be on a day that has already been allocated and therefore the calls for that staff member would need to be reallocated to another member of staff. Therefore this system allows for re-entry of staff working into the system and reallocate and store calls for a duplicate day. This would be done as soon as the administrator was notified that a member of staff was unable to work and would reallocate calls. Once calls were re-allocated staff would be directed to review their call lists again and note down or print out any changes.

(Geographical location)

During the allocation part of my system I have taken into account the geographical location of both staff and patients. I grouped each list based on the town stored in the database. Each set of patients is assigned to the staff member most suited to the set of patients based on their town locations. This creates call lists which are as efficient as they can be due to the fact that many members of staff may not live in the catchment area for Bargoed calls. Each set of calls assigned to a member of staff consists of calls which are grouped into small sections which represent small areas in the community. This allows calls which are near to each other to be completed together to prevent staff leaving a certain area and

then later having to return. Each call list is automatically grouped as best as possible and there staff no longer needs to manually order their lists like before. Staff can then use their time better by spending the extra few minutes with their patients or completing important paperwork. This can allow for a more efficient work force as it gives more opportunity for staff to keep up to date with paper work, equipment and finish their shifts in time and with time to prepare for the next working day.

Extras:-

Saving data 'Date' and 'Staff name' in a new table in order to keep records of staff working on a particular day. When a new call list is created this information is put into a large table containing all dates saved within the last month. This can become useful when incidences occur within the job and staff is required to prove whether they were working on a particular day. This information can assist with investigations so that they can be handled quick and efficiently should a clinical incident occur.

Conclusion to user testing

From the feedback gained during the user testing stage gathered the knowledge that my system I have created has achieved its core functionality. Each user was able to complete their set tasks very easily and found that they became familiar with the system very quickly. During their time using the system I found they didn't need any prompting to how to use the system as each step they took was similar to the next, and therefore they become accustomed to what the system offered and could handle each task very easily.

Each user stated the ease of the system and how the system flowed from one page to the page. They found that the sidebars helped when they needed a little extra guidance into how each page worked, and many users who are not strongly computer literate found this useful as they didn't need to remember how to complete each page on their own.

Each user was happy with the overall look and functionality of the system and they were pleased to explain that the proto-type fitted its overall purpose for call allocation. They understood that this system is a proto-type and therefore doesn't need to look perfect and the most important aspect was to complete the core functionality. Each user could see the potential in the system and when discussed after testing was complete about how this could work if future work was completed they all agreed

this system was a good stepping stone as a basis ready for a real online system.

5 Future Work

5.1 Additional requirements:-

For future work for the system I would firstly look at the additional requirements which were stated in the interim report:-

- Staff and patients details are to be changed directly from the system.
 - The current proto-type system allows only the administrator to complete this function by directly accessing the database. From the database the administrator can change, add or delete details.
 - Due to my current system being a proto-type system I saw this functionality as being less relevant to the core functionality as the system could easily function without it.
 - For future work this functionality could be included within the system to allow for the administrative tasks to be limited and therefore give full control to the system itself. Administrator would make all changes directly from the system of having to access the database each time. The decreased access which would then be needed could be a good thing due to security measures and data protection act.
- Time constraints for calls.
 - The current proto-type system uses an average time for each type of call (*Bloods* and *Other Calls*) and the system calculates the call lists accordingly.
 - For future work *Other Calls* can be broken down into a number of different categories (syringe drivers, dressings, injections etc) and each type given accurate time duration. Then when each call list is created the correct time durations can be taken into account to produce a more accurate call list.
- Call Reallocation; this could be due to a staff calling into the office at the beginning of a shift explaining they are ill and unable to attend work or if a member of staff is unable to complete their entire call lists and needs assistance to complete calls these calls can be reallocated to those available.
 - The proto-type system I have created allows for call reallocation for everyone if one person calls in sick or unable to attend work for other reasons. The system allows for calls to be reallocated for

a duplicate date but not for a single member of staff. Each member of staff would then need to re-enter system and look at their newly created call list as it could have changed due to staff change. This method allows for geographical location to be taken into account for all members of staff even with the change.

- For future work the system would allow for a member of the working staff to be removed from working list and a new one re-added taking over that individuals calls. This would save time staff would not need to review call list but instead use the original and only new member of working staff would need to look for their call list. (This list would be a duplicate of absent staff member and list may not be the most efficient based on geographical location, because new staff member could be from a different town location to absent staff member.)
- Take into account the band level of staff as certain members of staff are able to complete certain types of calls. Some calls require a higher level of staff than others and more skills are needed in order to treat patient correctly.
 - The proto-type system I have created does not take into account a staff members' band level when allocating calls it simple allocates regardless.
 - For future work when allocating calls this band level would be taken into account as only certain members of staff are qualified to complete certain treatments to patient. This therefore would lead to a more accurate list than the one created by the proto-type system.
- Other additional functionality:-
 - Length of shift as during the implementation stage new types of shifts was brought into the District Nurse Team. These new shift patterns include a day shift and a twilight shift where calls are therefore covered through the day and night. This was brought in to provide around the clock care to patients who require it and this takes some stress off other community nurses who used to care for patients during the night.

5.2 Card Reader:-

(Reliance)

During implementation stage of this project the Bargoed District Nurses were introduced to a new tool to enable more security for staff whilst in patients' houses. This tool is called 'Lone Worker' made by the external company Reliance. Each staff member is given a loner worker which looks

like an ordinary badge in order to disguise its real purpose. The lone worker looks like:-



Figure 42. Front view of lone worker - badge



Figure 43. Back view of lone worker - badge and buttons

Each staff member is trained to use this tool in a number of situations in order to increase security for staff. The main objectives of the lone worker are:-

- To provide a greater sense of security
- To look at the nature of abuse towards staff
- Helps to develop plans to combat abuse of staff
- Reduce staff turnover sickness and absenteeism
- Good evidence capturing
- Better working relationship with the Police

There are two main functions that each staff member needs to be aware of:-

- AMBER alert – which a staff member can press and leave information about their whereabouts and stress any concern they may have. A staff member can activate this option by pressing both the static check button along with the amber alert.
- RED alert – which a staff member can press which creates an open voice call straight to an experienced alarm handling operator, at the Reliance ARS, the sooner they can listen the better. A staff member can activate this in one of two ways such by both pressing and holding the large button at the back of device to simply ripping the lanyard plug from the body of the device.

Reliance is an external company which provides 24/7 manned monitoring therefore staff have some point of contact at all times. Staff members carry a large amount of equipment including needles and therefore they require this safety and back up should a member of the public or patient turns on them.

The key service benefits are:-

- Discreet design which ensures lone worker's safety isn't compromised
- Always with the worker, within immediate reach
- 'Status Check' and AMBER alert functions form part of your dynamic risk assessment
- Open voice channel on RED alerts capture evidence

Once a staff member raises the amber or red alert, information about a staffs' location is stored on an electronic management systems solely managed by Reliance Protect. This information will only be used in the event of an incident occurring that requires additional intervention. The

information stored will be made available to the Police and CPS (Crown prosecution service) in the event of an incident occurring that leads to criminal investigation and/or prosecution. Under the Data Protection Act, audio recordings made may be disclosed in connection with legal proceeding.

During future work this device could be used, however in a slightly different way. Each staff member could use the device to state their location before each call and this could be stored in a database back in the central office. Therefore if another member of staff requires knowing where a particular member of staff is then they can use the database to locate the last location stored. The AMBER and RED alerts could still be an external aspect and therefore staff should find themselves feelings safe and secure when in patients' houses. All this information can be used in a court of law as evidence should an incident occur.

5.3 Making system online:-

As the current system is simply a proto-type it is only required to complete the main functional requirements and doesn't have to be perfect. Therefore in future work should this system be taken online it should take design aspects from current system but make the overall look more professional. It should have one professional interface in which all functions of the system operate from. The future system should be able to be downloaded onto a number of secure machines in the District Nurse offices.

Transforming the current proto-type system to an online system would require NHS data to be accessed. Therefore all NHS regulations stated in the NHS Direct Confidentiality Policy legal rights stated in Interim Report need to be met in order for an online system to be used.

5.4 Cannot be portable:-

For the current proto-type system details can be printed off very easily as the data stored is substitute data and not real NHS information. However if the proto-type system was used in order to create an online system, which would follow the legal NHS principles to allow for NHS information to be located, then precautions need to be made.

Staff would only be able to access this online information from a number of secure computers within their offices. Staff could not access this information from portable devices as this would breach guidelines set by the NHS. Therefore the only solution would be to have a print option for each call list within the new system and each staff member could take a

printed list in their car during their shifts. Once their shift would be completed they should bring their printed list back to the office and put in a 'safe sack' which would then be shredded securely at the end of each working week.

5.5 Translate to welsh:-

(Wikipedia, 2013)

(NHS Wales, 2013)

(Welsh Government, 2011)

(Department for Communities and Local Government, 2013)

As this project is being conducted in Wales and based on a Welsh District Nurse Team, then translating the system to Welsh will be a main priority for future work should the system be taken online.

Guidelines from the Welsh Government states that Welsh and English languages should be treated equally in the public sector. These guidelines come from the Welsh language act 1993 and the Government of Wales act 1998. The Welsh Government require public bodies to be prepared for approval of welsh language scheme which indicates commitment to the equality of treatment principle. Should this system be taken online it should comply with the regulations made by the Welsh language board.

Under Welsh language scheme deals with upholding the principle where public businesses in Wales, treat the English and welsh languages equally. This is done where a service is provided to the public in Wales. Since we are under the NHS welsh health board if this project was to be taken online then this would be a principle which would need to be obeyed.

NHS Welsh health board show they are complying with the regulations held by the Welsh language board by having a 'Cymraeg' option on their main and direct websites. Therefore if in the future this proto-type system was to be taken online it would need to be translated into Welsh but still be of an equal high quality as when in English. This option should be clearly available to ever user should they wish to use it.

6 Reflection on Learning

Project formation and Operation

(Stevens, 2000)

(R.Schach, 2002)

(Jalote, 2008)

The main challenge for my project was broken down into specific tasks which would make up 3 main goals; Initial Plan, Interim Report and Final Report which includes a fully implemented system. My role in my project was to complete each task in order to conclude an overall proto-type solution which would suitably complete requirements set by my client. These tasks included gathering research, concluding initial requirements, creating designs in order to visualise and help build (implement) the system. During each task I communicated ideas and pieces of work to both my client and supervisor. Continuous interaction with both my supervisor and client allowed for the project to progress well and complete each goal and then move on to the next. Due to my client living in the same house, I was able to express any ideas freely and would gain feedback on them very quickly. Also living in close proximity harboured an environment of inspiration due to experiencing the main problem on a day to day basis as client would come home from work and express their difficulties with their day. I could then use this frequent information to improve the system on a regular basis to develop the most efficient system possible.

Planning and Progress

The project was planned by analysing the requirements stated by the client and breaking them down into a list of high-level tasks. The high-level tasks were also broken into a week by week plan which allowed me to see where I needed to be on a week by week basis. This list of tasks enabled me to keep up with the tight time frame. These tasks were then included in a Gantt chart, which allowed me to see when exactly particular tasks would be finished. This Gantt chart was revised slightly after the Interim Report and during this Final Report, to reflect where my estimations had proved incorrect, or where there had been delays. The time taken for the design stages of the project were accurate, however, the time taken for integration and testing was underestimated and therefore moved into the last weeks which were meant to be for write up of final report. However, the write up of the final project was over-estimated which meant that the project was completed within the allocated time frame. I attended weekly meetings with my supervisor and client where I gained feedback on any suggestions I made, where appropriate. Particular feedback I gained from my client which I concluded to be very useful was the design of what the system should look like in order to make it as usable as possible for all potential users. An item of feedback I gained from my supervisor which overall improved the system was to use SQLite instead of mySQL as it made the system

more functional and easier to work with. Due to weekly meetings with supervisor and living with my client I was able to gather the main project specifications very early on within the project which enabled for more in depth research to be done at the start. Once this research and specifications were finalised, there was a clear path to follow from beginning to end. Following the waterfall structure stated in 'project formation and operation' once a task was complete I could move onto the next without the need to return and make revisions. Only small aspects of the project was changed and this was mainly in the implementation stage, where working in close proximity with my client enabled me to show any completed part of the system and change parts there and then. These changes included changing tick boxes and radio buttons for clickable scroll bar list in order to make the system more usable.

My role in my project

During the project, I handled all tasks stated in Gantt chart in interim report using skills learnt previously, and those learnt to complete an efficient solution to my clients' problem. The weekly meetings I attended with my client and supervisor (Helen Phillips) helped to discuss progress on each task and what I would aim to be completing for future meeting.

My first task involved structuring my project into smaller steps in order to display a step by step guide to work to be completed within a certain time constraint. This was confirmed with my supervisor and a weekly break down of these steps was also completed, in order for supervisor to have a week by week guideline for where my project should have reached (Appendix I.) Once structure was complete I moved on to gathering research and specifications/requirements. This meant working directly with my client to gain an in depth knowledge of the initial challenge and the extra research needed to fully understand and propose the best solution possible. Using these requirements and proposed solutions I created a number of designs such as templates and activity diagrams (can be seen earlier on in this document; Section 2) to enable implementation to follow on smoothly. Using past knowledge of programming in JAVA I was eager to implement a system which would comply with all functionality and provide a usable environment for my client and other potential users. Using the algorithm designs stated earlier on in the design stage of this document (Section 2) I used their concept to program methods which enabled call lists to be created and saved within the SQLite database. Due to the abstraction of this project this meant that the implementation stage was very complex and required a lot of research even with a full set of designs; research included looking back into

connection of the SQLite database using JAVA programming language. Although this abstraction increased the work-load, it is a decision I think paid off, as it allows the system the freedom for expansion very easily. Proto-type system can easily be used as a basis for expansion to an online system and to be expanded using the additional requirements. The full transcript of current proto-type system can be seen in the Appendix F.

Reflective Evaluation

From year 1 and year 2 modules I gained a number of skills ranging from documentation to programming languages which enabled me to approach this project with a clear and level head. Past modules included project work in which I took ideas from and adapted them to suit this project. These ideas included time keeping, work structure/break down, programming styles, organisation of work, and documentation. Due to the number of different tasks requiring a diverse skill-set I could use previous knowledge to excel easily in some parts, and in others I needed to research and gain a new understanding in particular fields. The project as a whole was large and required all skills I had learnt in previous years along with gaining new knowledge to complete the complex project, and I think I worked well to complete the project to a high standard.

This is the first full project I'd done which meant I had to complete the entire project on an individual basis, rather than from a group point of view which I had been previously used to. This project was probably one of the largest projects I'd undertaken and it has taught me a lot about what is required to complete full documentation and the importance of design. A clear design allowed for clear flow into implementation stage which meant fewer stopping points. The experience of this project will influence how I start projects in the future, in that I will spend more time in creating a well-planned design stage before starting work on implementation. Due to close interaction from the client during the implementation stage, I am confident that the proto-type produced could be used directly in the future to create an online system which could be integrated straight into their offices.

In observation, I think that from the start of the project, communication with both my client and supervisor had been consistent and this has reflected in a precise solution being implemented. Due to good communication on a weekly basis it enabled regular feedback, which I used to input into all aspects of the documentation and implementation stages to create something to fit the purpose of the problem.

Communication included weekly meetings along with numerous emails which I think helped keep the project time span on track.

This was one of the major advantages to my project, and I felt that this was a two way street on behalf of myself, supervisor and client. A good communication channel meant that I did not have to wait a long time to get feedback, and therefore could use any feedback immediately which meant time could be optimised throughout other necessary aspects of the project.

An aspect of the project I would have done differently would be to have a more precise design section. I managed to complete a detailed design section however during the implementation stage many aspects needed to be more specific. Such as more detail on algorithms used and any temporary tables that would be needed. In future projects I would look into researching all aspects of design and cover everything in detail so that during implementation everything is there to follow as guidelines.

Appendices

- Appendix A – Nurse Tester
- Appendix B – Nurse Tester
- Appendix C – Admin Tester
- Appendix D – Admin Tester
- Appendix E – Usability Notes
- Appendix F – Code Listing
- Appendix G – Full Test Cases
- Appendix H – Figures
- Appendix I – Week by week plan

References

Beasley, J.E, (). Vehicle Routing, [Online] Available at: <http://people.brunel.ac.uk/~mastjbb/jeb/or/vrp.html> [Accessed: 10th December 2012]

Ben Shneiderman. (2007). *Eight Golden Rules of Interface Design*. [Online] Available: <http://www.gosoft.co.uk/developerzone/shneiderman/shneiderman.php>. Last accessed 21st April 2013.

Ben Shneiderman, Catherine Plaisant, Maxine Cohen, Steven Jacobs. (2009). Eight Golden Rules of Interface Design. In: Addison Wesley *Designing the User Interface*. 5th ed. London: Pearson. p100-p120.

Bernd Oestererich. (2002). Analysis. In: Addison Wesley *Developing Quality Software with UML*. 2nd ed. London: Addison Wesley. P75.

Bernd Oestererich. (2002). Behavioural diagrams. In: Addison Wesley *Developing Quality Software with UML*. 2nd ed. London: Addison Wesley. P70, p250-p254.

Bernd Oestererich. (2002). Identifying Business Processes. In: Addison Wesley *Developing Quality Software with UML*. 2nd ed. London: Addison Wesley. p69, p161-p170.

Department for Communities and Local Government.(2013). *Welsh Language Scheme*. [Online] Available: <https://www.gov.uk/government/organisations/department-for-communities-and-local-government/about/welsh-language-scheme>. Last accessed 21st April 2013.

NHS Wales. (2013). *Home*. [Online] Available: <http://www.wales.nhs.uk/>. Last accessed 20th April 2013.

Reliance. [no date] *Lone Worker User Training*. Reading: Reliance

Simon Bennett, John Skelton, Ken Lunn. (2004). Activity Diagrams. In: McGraw-Hill *Schaum'soutlines UML*. 2nd ed. London: McGraw-Hill Professional. P234-p239.

Simon Bennett, John Skelton, Ken Lunn. (2004). Class Diagram. In: McGraw-Hill *Schaum'soutlines UML*. 2nd ed. London: McGraw-Hill Professional. P42-p72.

Simon Bennett, John Skelton, Ken Lunn. (2004). Use Cases. In: McGraw-Hill *Schaum'soutlines UML*. 2nd ed. London: McGraw-Hill Professional. p20-p41.

Welsh Government.(2011). *Our Welsh Language Scheme*. [Online] Available: <http://wales.gov.uk/topics/welshlanguage/policy/wls/?lang=en>. Last accessed 21st April 2013.

Wikipedia.(2013). *Welsh Language*. [Online] Available: http://en.wikipedia.org/wiki/Welsh_language. Last accessed 20th April 2013.

Peredita Stevens. (2000). The development Process. In: Booch, Jacobson, Rumbaugh *Using UML, Software Engineering with objects and components*. 2nd ed. London: Addison-Wesley. p47.

Stephen R.Schach. (2002). Software life-cycle model. In: Emily J.Gray *Object-Oriented & classical software*. 5th ed. London: McGraw-Hill. p65-p71.

Pankaj Jalote. (2008). Waterfall. In: Ian Mackie *A concise introduction to software Engineering*. London: Springer. p14-p16.