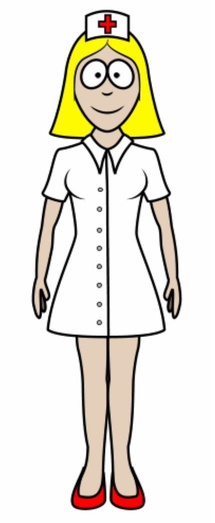
Interim Report



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 to create a proto-type system and manage to help the district nurses within the Bargoed offices allocate their calls in the most efficient way possible. This will allow for more time to be spent with patients. I have focused on the main aspect of allocating calls which allows for a larger scope for expansion in the future. This system will be a desktop application which can be accessed from secure offices in the Bargoed district. The system will allow for interaction between itself, the district nurses and administrator.

Acknowledgements

- Carol Lippiett – Mother, client and district nurse.

Contents

Figures

Tables

Introduction

Initial Description:

At the moment the team of district nurses allocates their calls amongst staff with no particular structure and then they have to individually organise their call list. This is a time consuming task for the staff as this needs to be done the day before as many calls require a certain level of staff to attend the call. Therefore many calls need to be swapped with other members of staff ahead of time in order for all calls to be covered. This project aims to implement a system that can distribute calls amongst staff based on 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.

In order to gain this initial Description I had to look into background research and find out exactly what is required.

Background and Requirements Gathering

Questions:

In order to gather the information required to get a clear understanding of what the aims and objectives of my project are I asked my client a number of questions:

What type of calls is there on a daily basis?

- Diabetics (Insulin injections) – once or twice a day according to dosage required
- Wound management – many need daily dressing however each wound if different and needs to be treated accordingly
- Palliative care – care of dying patient administrating medication
- Vene puncture (bloods)
- Enteral feeding – feeding regime for those who are unable to feed themselves
- Administration of medication (for example injections)

Which calls need to be done first?

- Insulin as this needs to be done before a patient can have breakfast
- Enteral feeding – needs a feed to be put up in the morning and then taken down later that day therefore requires second call
- Fasting bloods – bloods taken before a patient has anything to eat
- Vene puncture samples – Daily collection picked up at 12 to go to pathologist for processing
- Timed medication – Morning medications which are prescribed and patients are unable to administer themselves. Syringe driver medication (to be replenished every 24 hrs via the syringe driver). Eye drops as prescribed can be up to 4 times daily.
- Wound management - Complex dressings/simple dressings – If a wound needs a swab then this would need to be known in the morning as a sample would need to be taken in the morning and sent off with the blood samples
- Palliative care
- Administration of medication

Are the bloods dropped off locally?

- Dropped off locally at the health centre and then picked up by the health authority van who takes it to the local hospital.

Is there one member of staff who collects all the bloods and then takes them all at once?

- Each member of staff is responsible for delivering their own bloods samples/ wound swabs safely and on time to the health centre.

How long will each type of call take?

- Insulin – 15 mins (check blood sugars and insulin)
- Enteral feeding – 45 mins in the morning along with medication –30 mins to take the feed down.
- Fasting bloods – 15 mins
- Vene puncture samples – 15 mins
- Timed medication – Syringe driver 45 mins – eye drops 15 mins
- Wound management – complex wound 1hr – simple wound 15 mins
- Palliative care – 1hr
- Administration of medication – 15 mins (all according what medication)
- Alongside calls for patients who need regular care there are also calls that involve assessments – incontinent assessments, nursing assessments (nursing or residential) and continuing care assessments (who will provide the care).
- Often members of staff have to attend a meeting at the hospital before a discharging a patient into the community.

What equipment is needed for each type of call?

- Insulin – Patient has own medication, syringes and sharps box in house but nurse carries own adrenalin
- Enteral feeding – nurse carries own gloves, apron and dressing packs all other equipment is prescribed individually for the patient at their home
- Fasting bloods – Nurse carries all equipment including needles, vacutainer, blood sample bottles, blood forms and sharps box
- Vene puncture samples – Nurse carries all equipment including needles, vacutainer, blood sample bottles, blood forms and sharps box
- Timed medication – Syringe driver medication is at the patients house which has been individually prescribed by the doctor, nurse carries syringes and needles – eye drops, nurses carries gloves, apron and dressing pack
- Wound management – patient has own dressings prescribed by doctor. Nurse carries hand sanitizer, gloves, apron and dressing pack
- Palliative care – nurse carries apron, gloves, syringes, needles and sharps box. Patients have own drugs prescribed by doctor.
- Administration of medication – nurse carries adrenalin, needles, syringes and sharps box
- For each patient the nurse fills in details on care sheets that are in patients homes but carry spare ones if they need changing.

Who distributes the calls amongst the staff?

- Manager (band 7 nurse) and deputy Manager (Band 6 nurse – e.g. sister)

What system do you use to distribute calls? How do you store patients details?

- Excel is used to distribute the calls.

How are the calls worked out once distributed?

- Calls need to be distributed to the right member of staff based on their current skills (for example band 6, band 5 or auxiliary nurse.)
- Once a member of staff has been given their list of calls they manually go through the list and work their day out based on what type of calls they are

given. Prioritising bloods and insulin's and calls which need to be done in the morning.
- Work day out the day before.

How does staff pick up the equipment?

- There is a large store room at base (health centre) where nurses can collect equipment as required.

Typical Day of a District Nurse:

An example of the type of day a district nurse currently has:
(Call sheet has already been collected for this day)

- Leave the house 8:30am and head to first call (fasting bloods).
- Then head to remaining fasting bloods calls.
- Then move on to diabetic insulin's (am).
- Then move onto routine bloods.
- Once all bloods have been completed take the blood samples to the surgery for collection at 12pm.
- Then continue with remaining calls.
- End of the day head back to the office for office work/paperwork and to collect calls ready for the next day.
- Follow call sheets and collect requirement equipment for the next day.

Type of Documents:

A district nurse is required to fill out many different documents for both themselves and patients, here are some examples:

- Mileage forms each month,
- Update patients health records daily,
- Time sheets weekly.

Examples of these documents are as an appendix

Following on from the information given from the client (Questions, Typical Day and Type of Documents) I now have a detailed understanding of the initial problem.

Initial Problem:

From the first client meeting and the in depth look into the background research the initial problem became clearer:

Calls are divided out amongst those that are working therefore leaving many nurses with calls that need to be done in a short space of time. (For example numerous bloods which need to be done before 12 in order to be sent off on time)
Once calls are distributed amongst the staff then each staff member has to individually take the time to order their own calls into bloods before 12, syringe drivers, dressings etc. They try and make sure they fit in all calls and leave enough time to complete each call. Different calls need different allotted times set. Also emergencies or situations occur which require an extended amount of time which then back logs the rest of the calls for the day.
Each member of staff needs to make sure they have all the equipment needed for all their calls for a particular day and usually there is no time to come back for resources.

They need to have the correct equipment for each individual call for example syringes and different dressing packs.
Work phones are used to call one another with any problems; for example if a member of staff needs to stay at a house longer than predicted and cannot manage all their calls they can ring round and see if another member of staff can cover some calls for them.

Following on from finalising details for the initial problem I researched a number of items in order to understand how best to go about this problem. I looked into existing systems to understand what has already been created and what ideas I could potentially use for my project. Then I looked at a number of methodologies in order to discover a clear structure I could use in order to make the project run smoothly and efficiently. Based on the data needed for the project (patient and staff details etc) I reached the best ways of handling and storing this data, and also the best programming language to a create a system suitable for my client's problem. I looked at a number of algorithms to see what way was best to deal with the data whilst allocating calls. I also looked in to the legal rights of the NHS as the data I am required to store is very sensitive and needs to be handled correctly. Here is the research:-

Research:-

Existing Systems:

The problem of having different types of calls and allocating them amongst staff can be done in many different ways and therefore I looked into similar systems which have this same basic idea:

Recall system for diabetic patients based in a Clinic (My Surgery Website, 2012)**:-**

- Computerised recall system to keep contact with patients throughout each year.
- System will deal with sending out information to patients who require a blood test before a main appointment. This needs to be ordered so as to be completed before main appointment letter is sent out.
- System will store personal data which will need to be secure.

  For my system:-

  - This would be useful for my project as it is a system which is already in place which requires information to be stored securely but readily accessed to make appointments (in my case call appointments.)
  - This system also is set up to send out paper work to patients however this paperwork will only be required for staff members.
  - Ordering of bloods calls before main appointment will be similar to the lists I create. Where bloods will need to be done before the main set of calls each shift.

Prescriptions within a Pharmacy:-

- Computerised system to make sure there are enough tablets for the prescriptions coming into to Pharmacy.
- System to store information about which drugs are going out to Patients and how many are left in stock. This information is stored so that the Pharmacy can calculate when they need to re-order certain items and the Pharmacy is kept well stocked.
- In a pharmacy they are require to have a number of different drugs and allocate them to a number of patients who require it. This can sometimes be one drug on

a prescription or numerous and therefore data will need to be taken from different areas but given to one person.

- These drugs are connected to a database which keeps track of drugs being given out and automates a re-order form once a particular drug is low. This is done via a computer but also checked manually.

For my system:-

- This would be useful for an additional feature where equipment could be stored and logged. Once an item is taken for a call then a note could be made automatically on the system and once number of items left becomes low and list could be produced (Re-order form.)

Methodologies:

In order to understand how I will approach this type of project I researched a number of methodologies. These helped to structure my work on a day to day basis in order to make sure all work was completed by its due date.

- Waterfall Model

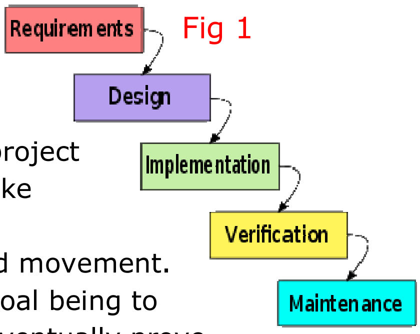


Fig 1

The waterfall model is a simple model that consists of five distinct phases. These are, "Requirements", "Design", "Implementation", "Verification" and "Maintenance". Due to my project being a proto-type system the "Maintenance" phase would not take place.
These five phases follow sequentially and do not permit backward movement. The aim is to perfect a phase before moving onto the next; the goal being to minimize resources wasted on designs or implementations that eventually prove to be unusable.
This project will need for the waterfall model to be carried out twice in order to cover core functionality and then additional features.

- Scrum

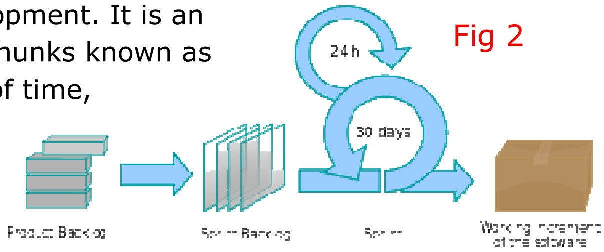


Fig 2

Scrum is a specific example of agile software development. It is an iterative framework that breaks development into chunks known as "sprints". These sprints each last for a set amount of time, generally between a week and a month. For this project these will be the weekly meeting with the supervisor where set amounts of work are required to be shown. The framework details tasks in the "product backlog", which is a set of high level requirements created at the outset of the development process.
Before a sprint begins, a planning meeting takes place (for this project; with the supervisor), during which the goals for the sprint are established and placed into a "sprint backlog", this is detailed in the Gantt chart produced for the initial plan. The Gantt chart includes the expected time taken for each task.
After the set duration of the sprint has elapsed, any uncompleted requirements are returned to the product backlog.

- Extreme Programming

Extreme Programming (XP) is an iterative development methodology, similar to other agile methodologies such as scrum. It takes its name from the idea that if a practice is beneficial, then taking this practice to extreme levels will provide extreme benefits.

The core tenet of XP is to hold short and frequent development cycles, releasing updates after each one. Other key features are programming in pairs, communicating frequently with the client and other programmers, conducting extensive unit testing of all code and only designing and implementing features once they are actually required.

Chosen Methodology

After considering the methodologies discussed above and taking into account the structure of my project, I am going to use a mixture of the waterfall model and the scrum methodology.

This project is broken down into very distinct parts that precisely follow the water fall model's steps. For my project each step is then broken down into smaller sprints which are explained in the Gantt chart (attached appendix.) Weekly meetings are set with the supervisor to deliver work from each sprint.

Once a section of the project is completed and submitted, changes would be difficult to make. I am going to take the waterfall model act as a guide throughout the project. This waterfall model and scrum methodology mixture will be carried out twice based on core functionality first and then being re-done from the beginning for additional functionality.

SQLite vs MySQL:

SQLite

- Single-file based database.
- More portable.
- Don't have to connect to an oracle server whenever you want to use it.
- Wide range of tools available to create lite databases.
- Useful for showing relationships.
- Stores data in structured files with indexes.
- Good for embedding in an application.
- Not fit for production.
- Open source library.
- Requires no server.
- Great for temporary and test purposes.
- Good for rapid development.

MySQL

- Normal database.
- Difficult to set up.
- Need a database server and client already set up to be beneficial for temporary and test purposes.
- Not suitable for embedding in some hardware.

Decision:-

After comparing both databases I have decided to go with SQLite for a number of reasons;

- If the system was ever released the database which coincides with it would be able to be accessed without needing a server connection.
- Already have code to connect to java program from last year.
- Due to my product not going into production as it will be a proto-type.
- Rapid development could be completed easily once core functionality is complete to create additional functionality.

- Since this system will be a proto-type the database will be good for test purposes.

Why Java?

Fig 3

Reasons for why I have chosen java:-

- Simple grammar.
- Object Oriented – program can be broken down into "objects" and those objects have their own attributes and methods which define them.
- Re-uses code very easily.
    - For example for my project I can re-use the code to link java to the database.
- Portability – java runs well on all popular platforms such as windows and Linux.
    - The operating system for my project will be Windows.
- Speed.
- Standard API's – comes with a large standard library. The library contains packages for a platform-independent GUI, threads, sockets, encryption etc.
    - For my project the GUI is a big part of my system and needs to be user friendly for all users of the proto-type system. The GUI has to be clear and precise in order to fit the purpose for all levels of staff's computer literacy skills.
- Don't have to worry about memory most of the time due to garbage collection.
- Can handle inner and anonymous classes.
- Controlled environment such as a VM means that there is no chance of corrupting the memory. Therefore when a bug appears it usually provides enough evidence about what caused it than in a typical native program.
- Has great support for exceptions, or issues that occur during runtime. These exceptions disallow hackers and malicious code form causing an issue on the system level.
    - This is a great aspect to have for my program as it will be handling data which needs to be kept secure at all times.
- Standardizes on a naming convention that is clean and readable.
- Fast edit, compile and run times.
    - Will be ideal my project as I will need to edit, compile and run over and over in order to make sure all core functionality is covered. Once this is done the fast edit will be good for when I look into additional functionality.
- Commonly used to write and run applications that run pretty much the same way independent of the system on which they are running.
    - This will be needed if this program were to be released as it would need to be run on Bargoed office computers.

Algorithms:

I have researched numerous algorithms in order to construct the most efficient algorithm for allocating calls.

Selection Sort (Virginia Polytechnic Institute and State University, 2007)**-**

Sorting n integers into ascending order:

1) Receive a set of n integers to sort into ascending order.

2) Set a marker for the unsorted section at the front of the set of integers.
3) Repeat steps 4 through 7 until one integer remains in the unsorted section.
4) Compare all unsorted integers.
5) Select the smallest unsorted integer.
6) Swap this integer with the first integer in the unsorted section.
7) Advance the marker to the right one card.
8) Stop, list is now sorted.

- This is a recursive algorithm.
- **For my system** I would need to:
    - Sort my lists into Bloods and Patients, so ideas from this algorithm can be used however it will not need to be in ascending order.

Divide and Conquer Algorithms (Wiseman, R, 2012)**-**

These types of algorithms follow the structure:-

1) Dividing the data into two parts (recursively.)
2) Finding subsolutions to each part (simple enough to solve directly.)
3) Constructing the final answer from answers to the subsolutions by combining them to a give a solution to the original problem.

An example of a divide and conquer algorithm is **merge sort** (XoaX, 2012)-

1) Divide: If S has at least 2 elements (nothing needs to be done if S has zero or one element), remove all the elements from S and put them into 2 sequences, S1 and S2, each.
2) Recurse: Recursively sort sequences S1 and S2.
3) Conquer: Put back the elements into S by merging the sorted sequenced S1 and S2 into a unique sorted sequence.

- **For my system** I would need to:
    - Divide: Patients into Bloods and Other calls.
    - Sort: Based on Town.
    - Split lists between selected members of staff.
    - Conquer: once each divided list (Bloods and Other calls) has been divided amongst selected members of staff each list will create a merged list. Each merged list will contain a number of Bloods and Other calls.

Another example of a divide and conquer algorithm is **quick sort** (Jitomirsky, E, 2012)-

1) Pick an element, called a pivot, from the list.
2) Reorder the list so that all elements with values less than the pivot come before the pivot, while all elements with values greater than the pivot come after it. After this partitioning, the pivot is in its final position.
3) Recursively sort the sub-list of lesser elements and the sub-list of greater elements.

Vehicle Routing Problem (Beasley, J.E, ())**-**

1) Scheduling of vehicles from one or more central depot to a number of delivery (or pick up) points.
2) Time windows for delivery.
3) Legal hours.

- **For my system** I would need to class these points as:-
  1) Central depot would be district base and then delivery points would be calls route.
     + These calls will be grouped based on town name and this will take into account **geographical location**.
     + Calls will be grouped together as best as possible in order to limit driving times between calls to allow for more time to be spent with patients.
     + For this proto-type system (core functionality) the geographical location will be taken into account for Patients based on the town they live in. This seems the most efficient way to organise calls based on how the calls can be completed to a high standard. District nurses will not be as rushed trying to get from one call to the next if calls are in close proximity. This can help all district nurses complete their call lists smoothly and optimise the district nurse and patient interaction.
     + For additional functionality geographical location will be taken into account for district nurses.
  2) Time windows would be bloods being done before 12pm.
  3) Legal hours would be the length of shifts 9am – 5pm.

Capacitated Vehicle Routing Problem (Neo, ())**-**
Made up of 2 subproblems that is broken down to vehicle scheduling and vehicle routing.

- **For my system** I would need to class these points as:-
  1) Staff scheduling
  2) Patient routing
  - This would be done to optimise time that is spent in patient houses.

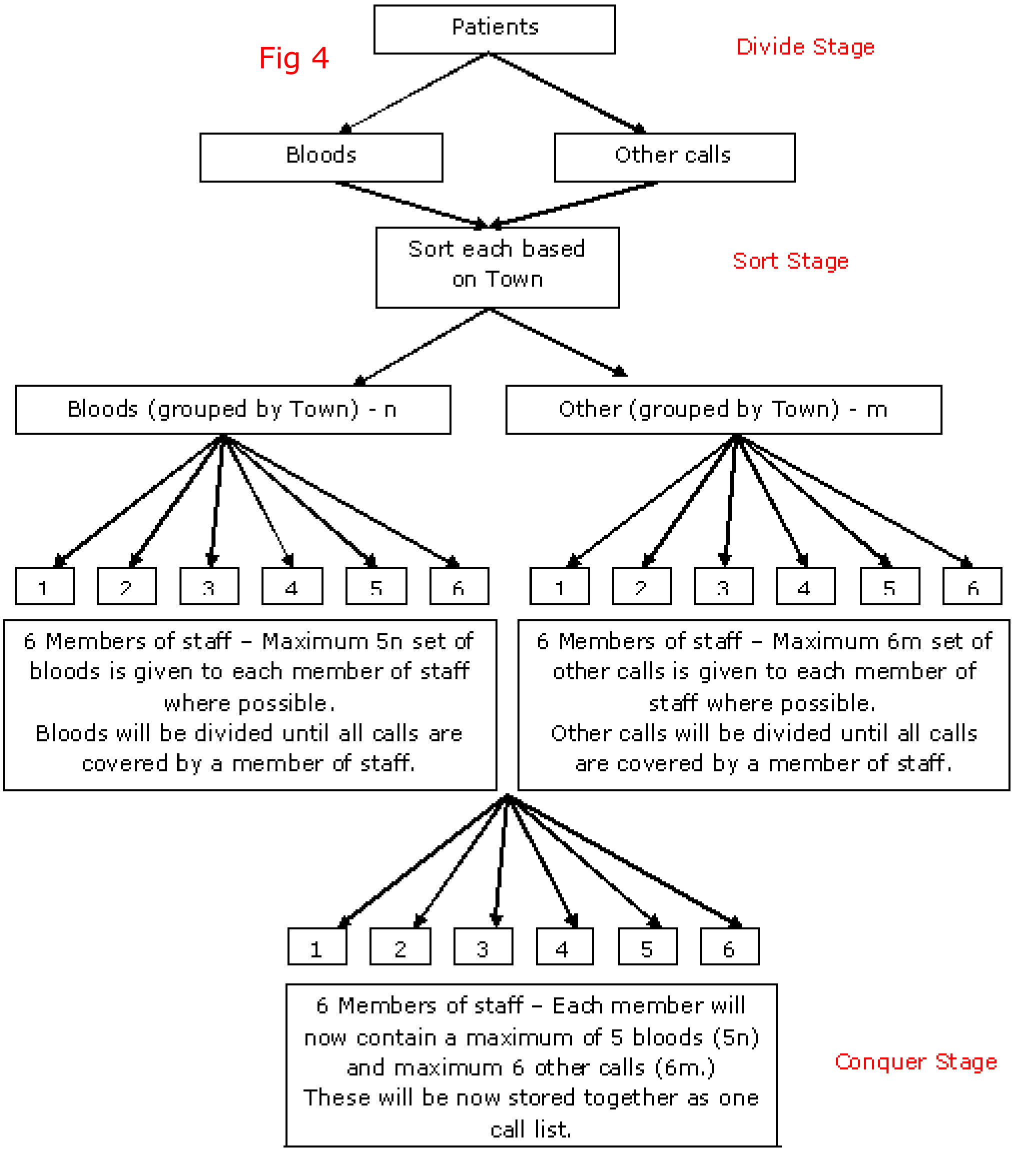
Clarke and Wright Algorithm (Neo, ())**-**

1) Initialise routes so that each route serves one customer.
2) Calculate Sij for all customers pairs (i,j) and list savings in descending order.
3) Beginning at the top of the list with the largest saving, test combination of 2 routes via edge ij to produce potential saving Sij.
4) Check feasibility of savings Sij with respect to the constraints of the problem. If feasible, merge routes along edge ij, else reject.
5) Until the bottom of the list is reached, move to next saving in the list and go to stage 4).

- **For my system** I would need to:-
  1) Make sure a staff member is only required at one patient at any one time.
  2) Test that calls are grouped to small areas where possible in order to save on travelling time. This time can then be spent with patients and used to catch up on lengthy paperwork.
  3) Every patient needs to be covered by a member of staff each shift.

Here is a diagram of the structure of an algorithm I will follow to create my proto-type system for allocating calls into call lists:-

Fig 4


Patients
Divide Stage
Bloods
Other calls
Sort each based on Town
Sort Stage
Bloods (grouped by Town) - n
Other (grouped by Town) - m
1
2
3
4
5
6
1
2
3
4
5
6
6 Members of staff – Maximum 5n set of bloods is given to each member of staff where possible. Bloods will be divided until all calls are covered by a member of staff.
6 Members of staff – Maximum 6m set of other calls is given to each member of staff where possible. Other calls will be divided until all calls are covered by a member of staff.
1
2
3
4
5
6
6 Members of staff – Each member will now contain a maximum of 5 bloods (5n) and maximum 6 other calls (6m.) These will be now stored together as one call list.
Conquer Stage

- I will begin by using a **divide and conquer algorithm**. I will start with the **divide** stage where I will separate Patients in to two groups; Bloods and Other calls.
- Once separated I will sort each list based on the Town name stored; Bloods n and Other m.
- I will then link in the **vehicle routing problem** where the delivery points in this algorithm will be seen as the patient houses. Since each group (Bloods and Other) are grouped based on town this should allow for the most efficient route to be allocated. This will eliminate long driving times between calls as they should be in close proximity.
- The **time windows** will be important as the bloods need to be carried out first of all. So when bringing back the lists back together into one using the **conquer** stage the bloods group will be put at beginning of call list followed by other calls.
- Each nurse will work 9am-5pm shift pattern for the purpose of this proto-type system.

Legal Rights – NHS:

Using the **NHS Direct Confidentiality Policy** I have researched the guidelines of the Policy and come to a conclusion for what is needed for my project.

Information for the NHS Direct Confidentiality Policy:-

NHS Direct:

- Is governed by legislation to process personal data, sensitive personal data and corporate data legally and ethically. The main United Kingdom legislations are:
    - Human Rights Act 1998;
    - Data Protection Act 1998;
    - Common Law Duty of Confidentiality;
    - Freedom of Information Act 2000.

NHS has a set of principles which need to be followed:

- Principle 1
    - Personal data shall be processed fairly and lawfully and, in particular, shall not be processed for a number of reasons.
- Principle 2
    - Personal data shall be obtained only for one or more specified and lawful purposes, and shall not be further processed in any manner incompatible with that purpose or those purposes.
- Principle 3
    - Personal data shall be adequate, relevant and not excessive in relation to the purpose or purposes for which they are processed.
- Principle 4
    - Personal data shall be accurate and, where necessary, kept up to date.
- Principle 5
    - Personal data processed for any purpose or purposes shall not be kept for longer than necessary for that purpose or those purposes.
- Principle 6
    - Personal data shall be processed in accordance with the rights of data subjects under this Act.
- Principle 7
    - Appropriate technical and organisational measures shall be taken against unauthorised or unlawful processing or personal data and against accidental loss or destruction of, or damage to, personal data.
- Principle 8
    - Personal data shall not be transferred to a country or territory outside the European Economic Area unless that country or territory ensures an adequate level of protection for the rights and freedoms of data subjects in relation to the processing of personal data.

Conclusion:-

Using the information from the NHS Direct Confidentiality Policy I have come to the conclusion that when carrying out my project I need to change some aspects in order to stick within these guidelines set by the NHS Direct.

- The personal data I will store on both patients and staff will need to be made up data.
- The personal data that will be stored in my system will need to follow the same structure as the NHS guidelines:-
    - Stored securely – for example only the administrator should have access to the main area where details are stored.

  - Only processed/accessed when needed – Only administrator has access to these details and therefore should only be accessed when data needs to be updated or deleted from the system.
  - Be kept up to date – for example when patients move home the address needs to be updated. This should only be able to be changed by the administrator.
  - Be kept for no longer than necessary – for example when a patient no longer requires care.
- Personal data should not be transferred outside of the NHS direct system.
- Personal data should only be stored for lawful purposes – for example to keep a clear record of staff working on a set of patients just in case something goes wrong with a patient and the situation needs to go through the courts.
- Only relevant personal data should be stored – for example home address, telephone number. Information that would be counted as irrelevant would be for example; 'what was your first pet's name.'
- Data should be organised and kept safe in order to prevent corruption and loss of data.
- Any printed data should be disposed of correctly; for example when a patient's personal details are printed once finished with they should be shredded.
- Any personal information based on staff or patients should be kept safely within the office environment.

Specification and Requirements

Based on the extensive background research I have come up with a proposed solution.

Proposed Solution:

Here are proposed salutations to the individual problems highlighted in the initial problem:

Calls are to be categorized into bloods etc and then given out equally amongst staff in order to give each member of staff enough time to cover the bloods and move onto other calls. A program needs to be created to allow for entry of the members of staff working on a particular day and then divide out the calls accordingly. The program should take into account the grade of staff working for example band 5 and the length of shift for each staff member and divide calls accordingly. For example bloods would not be given to those on the afternoon shift. (Additional functionality)

A program needs to be created in order to enable the addition of client and staff into the system. For each client all medical details need to be entered along with the equipment needed for a particular call. For each staff all details need to be entered along with the band they are on. (Additional functionality)

Create a system so that if a member of staff needs someone else to covers some calls then they can call the office and a staff member who is at base can look into the system and see who is available to take the calls and call them with details of the calls that need to be taken. (Additional functionality)

Create estimated time needed within each call in order to estimate the work load given to each member of staff.

Create a system to allow for when a member of staff calls in sick then their calls can be immediately reallocated amongst staff in on that day, or if this is not possible then management can authorise a use of a bank nurse to simple take over the calls. (Additional functionality)

Patient location needs to be taken into account to prevent staff driving from one place to another and then back again. Cluster calls into a smaller area of a calls to limit the amount of time driving from one call to the next and this can help lead to the efficiency of calls being completed on time. This can help with health and safety due to the fact staff can spend less time on the roads and more time with the patients.

System:

This system will:

- Be a proto-type therefore won't have an option to translate to welsh.
- Be a basic system to perform core functionality.
- Use substitute data for NHS data in order to comply with NHS rules and regulations.
- For the purpose of this proto-type system annual leave, sick days and holidays will not be taken into account for members of staff.
- The administrator will change details for staff and patients from the main database for core functionality however within additional features this will be done using the system interface.
- Each member of staff will complete a 9-5pm shift only and will be assumed to all be on the same band level.

Users of the System:

There will be two main users of the system:-

- Administrator
    - Administrator will look after all the details stored within the system.
    - Administrator will have direct access with the database in order to edit details for both patients and nurses. This will only be due to the system being proto-type, however if additional features are added then this function will be done directly through the system interface instead.
    - Administrator will be the only user who can allocate calls.
    - Calls need to be allocated each day ready for the coming day. These need to be allocated before 12pm each day.
- District Nurse
    - Nurse will have access to the system to view their personal details, patient's details and call lists.
    - Nurse will use the system when more detailed version of patient's details are needed as the call lists only display a short version of patient's information.
    - Nurse will be able to log into the system after 12pm each day to collect their calls for the coming day. This list can be printed so they can have easy access to list while on calls.

User Requirements:

From the information gathered while asking questions to my client (district nurse) I pin pointed the important aspects:

- 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.

Call Lists:

Call lists will be created based on each selected staff member having a 9-5 shift.
For the purpose of this project and using the fact that this system will be a proto-type each call list will be made up of (insulin calls will be included within other calls):-

- 5 Bloods
- 6 Other Calls

Bloods:-

- 15 minutes to complete treatment
- 5 minutes to complete paperwork
- 10 minutes to travel to each call
- OVERALL = 30 minutes

Bloods will be completed between 9am to 11:45 am.
From 11:30am to 11:45am staff members will drop off bloods samples to the clinic before the 12pm deadline.
Then from 11:45am to 12:15pm this will be allocated for a dinner break.

Other Calls:-

- 20 minutes to complete treatment
- 5 minutes to complete paperwork
- 10 minutes to travel to each call
- OVERALL = 35 minutes

Other calls will be completed between 12:15pm to 4:30pm.
Extra half hour is allocated to allow for:-

- Unforeseen problems such as traffic.
- Calls taking longer than allowed for.
- Time to collect calls and equipment ready for next shift.

I have taken an average amount of time set for each call and if I have time to complete additional functionality I will include correct time sets for each particular call.
I have included the constraint that there will be 6 District Nurse staff for each shift with a maximum of 30 Bloods patients and maximum 36 Other Patients in order for all calls to be covered.

Core Functionality:-

Functional Requirements:

- 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.
- A system is required to sort through patient's details comparing criteria with district nurse staff levels and then divided amongst staff accordingly.
- After 12pm call lists can be collected.
- 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
  - 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.

Table 1

Functional Requirement	Solution
Nurses/Users are able to view and download/print their call lists	System will allocate an appropriate set of calls for each member of staff based on the data stored in the patient's databases.

Non - Functional Requirements:

- The system is required to differentiate between personnel and therefore have secure access. For example have a log in.
- Another security requirement s that the system should keep its integrity at all times. Personal data for both patients and staff are stored and this data needs to be protected from those trying to hack the system. When invalid data is entered into the system then the system should handle this and not crash.
- The system is required to have a search property in order for the logging in process to function correctly.
- The system is required to be as quick and as efficient as possible, then the system will need a sorting method and a link to the relevant database involved.
- System will require numerous databases to store patients and staff details and this data will need to be held securely and be inaccessible to any without authorisation.
- To enable our program to be maintained easily, we should self-document all the code we produce and complete a set of system documentation.
- Security
  - Our system will be storing details which are private and confidential and therefore are at risk of copyright and therefore the database will need to be protected. This should be done by password protecting the relevant database and only allowing an administrator to access this information. Only administrator will have access to the database. The system will have to have

necessary protection to stop unauthorised people changing important information.

- In order to protect the system, it shall require the user to successfully complete a log in process, with details provided to them. These details will be stored in a password protected database only allowing administrator or staff access.

Table 2

Non-Functional Requirement	Solution
Database Security	Password protected database, keep regular back-ups in case of corruption. Restricted access.
Access	Only allow access to staff and admin. All users must log in before accessing the system.

- Legal Requirements
  - As with any piece of software that has a general release, our system will have to comply with various rules, regulations and Acts of Parliament. Such an example is that material published in Wales should have a Welsh-language version available. Due to this software working so close with the NHS, it should comply with these regulations also.

Table 3

Non-Functional Requirement	Solution
Regulatory Compliance	Cross-check functions & usage of system with a list of relevant regulations our software should satisfy for example NHS.

- Usability
  - As for any piece of software that is to be used by the general public, the system needs to be easy to use, preferably without the need for a manual.
  - System needs to be accessible to all users including those with disabilities.
  - Another requirement would be to make the software easy-to-use as some potential users may not be computer literate. Therefore, the user interface should be simple to navigate through and show information clearly. The software should be executed via a minimum of user input.

Table 4

Non-Functional Requirement	Solution
Usability	All buttons will contain relevant labels with text in a suitable size so that it can be read and understood easily. Program should run by clicking an icon (or in a browser window via a link for a web application version).

- Extensibility, Maintainability & Documentation
  - I need to make sure it is possible to add extra functionality but in a discrete way. I need to try and make sure it isn't time consuming to add or modify functionality in order to improve the system. The system needs to be easily maintained. This is important because if any bugs are detected after release it will be necessary to solve them quickly as the system will be relied upon by the nurses.

- In order to satisfy the requirements for maintainability and extensibility, we would need to provide some technical documentation and self-document code. This helps any additional developers that may make changes to the system.

Table 5

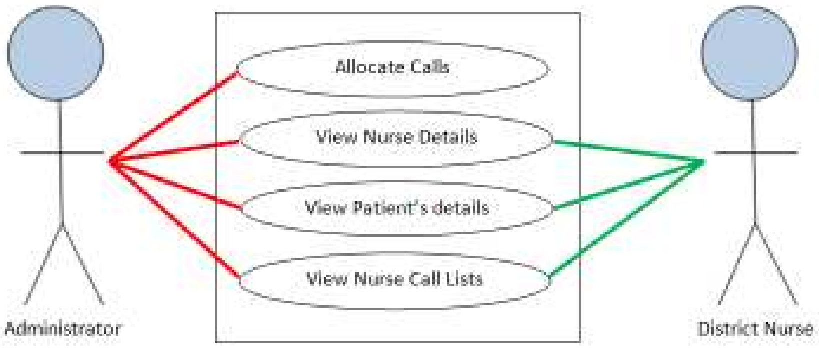
Non-Functional Requirement	Solution
Extensibility	Self-document to allow the maintainer to quickly locate code needing to be changed. Provide technical documentation in the design as well as information that the client will understand.
Maintainability	Self-document to allow the maintainer to quickly locate code needing to be changed. Provide technical documentation in design as well as information that the client will understand. Undertake thorough testing so that test data can be entered into the system to produce the same result that can then be verified by another developer in the team.
Documentation	Provide technical documentation containing a list of all the algorithms used in the software and provide detailed pseudocode and flow charts.

Architecture and Design:-

Use Cases:

Call Lists:-

Fig 5



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.

1.2 Use Case – View Nurse Details

Administrator

Brief Description

This allows the administrator to view all staff details.

District Nurse

Brief Description

This allows the nurse to view their personal details.

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. 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.

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.

District Nurse

Brief Description

This allows the nurse to view their personal call lists for the upcoming day. These can be printed in order to allow for easy accessibility between calls during a shift.

Class Diagram:

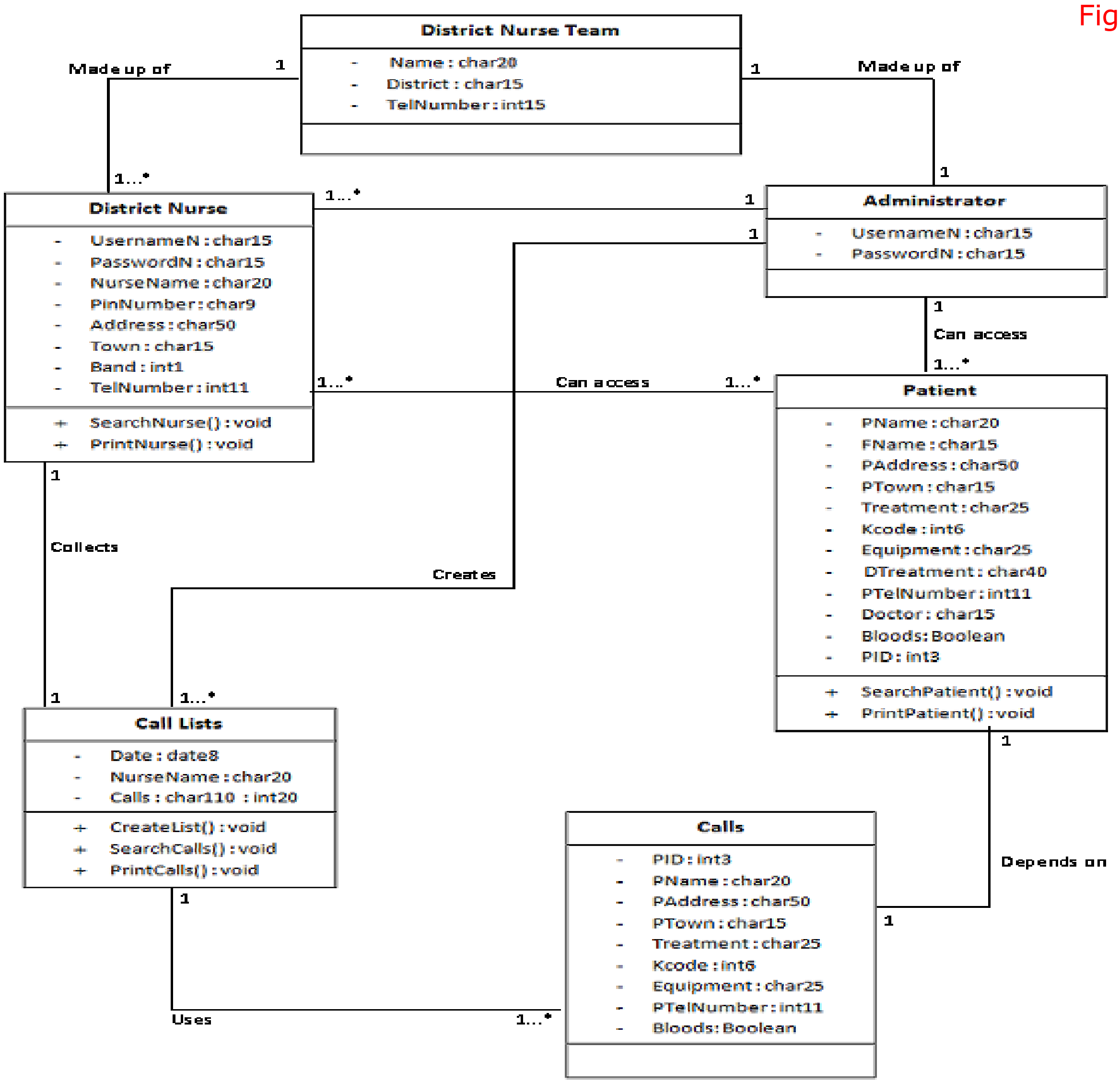


Fig 6

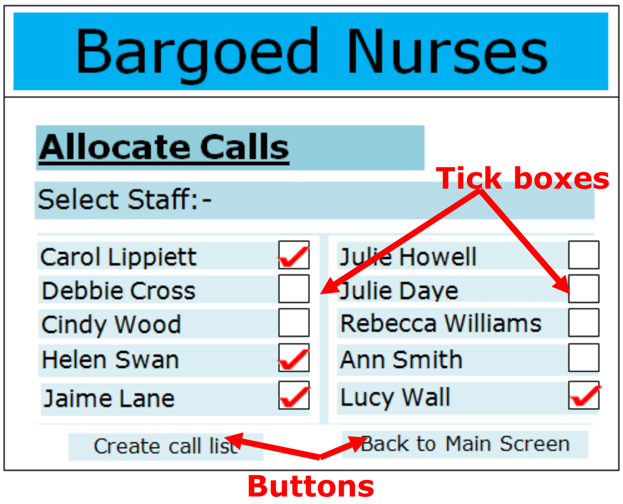
Glossary of Terms for Class Diagram:

- UsernameA and PasswordA – Username and Password assigned to Administrator.
- UsernameN and PasswordN – Username and Password assigned to District Nurse.
- PName – Patient Name.
- FName – Family Name.
- DTreatment – Detailed description of Treatment.
- Treatment – short description of Treatment.
- Bloods – Patients who require bloods to be taken.

Sample Templates:

I have created a sample set of templates for what the system could potentially look like. I worked directly with the client (district nurse) in order to create these templates to cover all aspects they required from the system.
Here is a sample few of the templates created.
The administrator will have rights in order to be able to create a call list. They will use the call duty sheets which are created 2 months in advance to select the members of staff working each shift. Here is the screen they will be displayed in order to select staff members:-

Fig 7



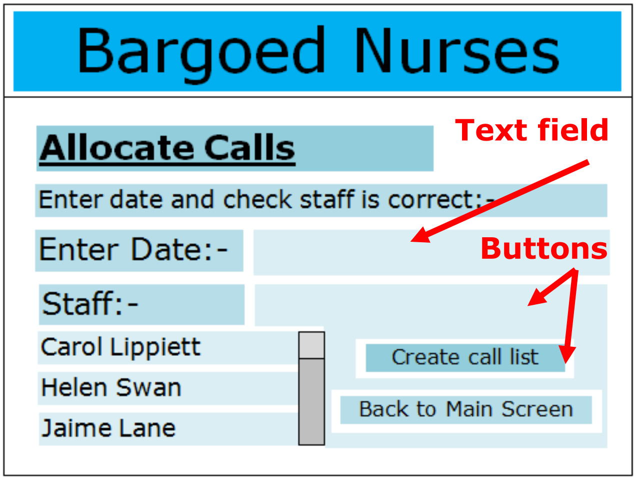
Carol Lippiett	☑	Julie Howell	□
Debbie Cross	□	Julie Daye	□
Cindy Wood	□	Rebecca Williams	□
Helen Swan	☑	Ann Smith	□
Jaime Lane	☑	Lucy Wall	☑

Tick boxes

Buttons

Once the administrator has selected a number (max number 6) of staff and selected 'Create Call List' they will be displayed a new screen. This new screen will require the administrator to manually enter the date and check selected staff members from scroll bar list. Once checked they can select 'Create Call List:-'

Fig 8



Enter Date:-

Staff:-

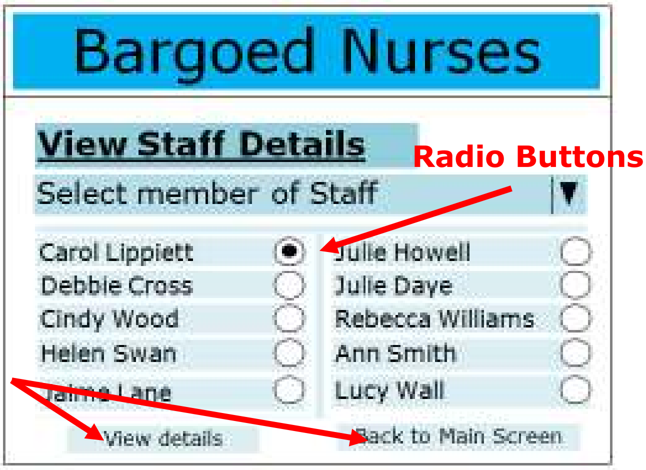
Carol Lippiett
Helen Swan
Jaime Lane

Text field

Buttons

The administrator will have rights in order to be able to view details about each member of staff. They can select a staff member from the list and then select 'View details.' The selected screen will look like:-

Fig 9

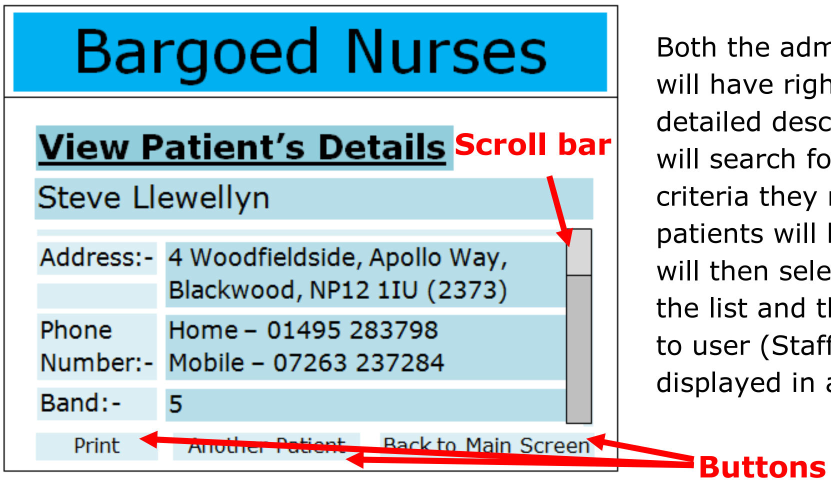


Carol Lippiett	●	Julie Howell	○
Debbie Cross	○	Julie Daye	○
Cindy Wood	○	Rebecca Williams	○
Helen Swan	○	Ann Smith	○
Jaime Lane	○	Lucy Wall	○

Radio Buttons

Buttons

Fig 10



Bargoed Nurses

View Patient's Details

Steve Llewellyn

Address:- 4 Woodfieldside, Apollo Way, Blackwood, NP12 1IU (2373)

Phone Number:- Home – 01495 283798
Mobile – 07263 237284

Band:- 5

Both the administrator and district nurse will have rights in order to be able to view detailed descriptions are all patients. They will search for a client based on search criteria they manually enter and a list of patients will be displayed to them. They will then select the required patient from the list and this screen will be displayed to user (Staff details and calls lists will be displayed in a similar format):-

Conclusions

Evaluation:

The project was about the prototyping of a system to assist a team of nurses trying to create the most efficient call lists. The current system they use is a member of staff using excel to manually create lists for each individual staff member. This can be very time consuming both for the allocator and the nurses who receive them. Nurses who receive these lists need to manually go through the list and number them based on bloods first and then other calls, trying to keep them in close proximity of each other. The prototype system aims to demonstrate the feasibility of implementing such a system to allow for call lists to be created for each member of staff using specific criteria; i.e. bloods and then other calls.

As a whole the project scope is very detailed and therefore I broke down the initial points into core and additional functionality. For my project I have concentrated on the core functionality in order to produce detailed requirements and designs to allow for a clear focus ready for implementation.

I mainly followed the waterfall model as my software-development methodology to define the aim of the project and to design its prototype – Requirements, Design, Implementation and Testing. I continuously followed my previous Gantt chart from my initial plan and updated it as I went along to ensure the prototype system will be completed on time. I made some changes to the Gantt chart from Initial Plan as due to other work commitments caused a number of weeks to be delayed. Therefore I have constructed an up to date version which is attached.

While conducting Requirements stage, I defined the users (Administrator and District Nurse) and the functional and non-functional requirements of the system. I looked into a number of existing systems such as those working with personal data similar to my project. The requirements and further background information I gathered at the start I used in order to design the prototype of the system. I worked closely with my client by presenting all documents to her and receiving feedback on all aspects of my project. All design templates were designed alongside my client in order to understand exactly what is required ready for implementation stage.

Justification of choice of development platform:

From the beginning I decided that the software to be developed would be (as a prototype) an offline program. This meant that it would not need to communicate with anything else besides the machine it would be installed on; would be installed only on the computers within the secure offices. This was a limitation due to the fact within the modules I have taken at Cardiff University none of which expanded on web based programs but only on single machine programming and databases. However this is not a problem for this project as the system need only be accessed by nurses within the secure offices and not accessed by any unauthorised users. Although I have worked on Linux and Windows machines and therefore the system will need to function on multiple platforms.

Limitations and restrictions:

Throughout the project, I have encountered various limitations. Some of these limitations were due to the skills I have learnt whilst at University which has limited me with my choices.

Another limitation was NHS legal rights. In order to consider legal requirements I will use made up data for both staff and patients however if this system was made for release it would require legality of NHS data. This is thought to be a major limitation for my design however my system with be a proto-type and therefore have looked at it with no constraints.

Database:

The database system for the project was chosen as a way to manage the data handle by the software. A database was necessary but due to the complexity that is required from the specification I decided that a single structure (SQLite) was necessary. SQLite is a relational database management system that would give my programme the necessary flexibility to handle data. This will be useful for creating GUI's in order to create the relations of the database without having to write lines and lines of code. SQLite is very easy to develop with and integrate; I also have the code in wish to integrate with java from previous year's project. It is a great database to prototype a system with, and facilitates local development.

I spent some time this semester familiarising myself with SQLite instead of creating the actual database. I looked back through last year's project to remember the features of SQLite and compare to how I can use them when I begin implementation next semester.

For the final report:

The key deliverables for the Final Report are:

- Summary of user requirements and design aspects since the interim report.
- Evaluation of overall findings and achievements of the project.
- Set of Test Cases to be used in order to test the running of the system.
- Database design.
- Architecture and data design.
- Explanation of the more challenging elements of implementing the system.
- Give an evaluation of the design, project management and methodologies used.
- Provide an up to data time plan and give a more detailed set of ideas for future work.
- If additional functions are to be considered a detailed design and explanation of these functions.

Future work:

Work to be completed by end of term:

By the end of the term I aim to have finished my evaluation of the most efficient diagrams based on my project, such as:

- CRC cards
- User Stories
- Use Cases
- Activity diagrams

- Class diagrams

Along with these diagrams I will aim to create a full set of detailed templates to further conceptualise the system. These can be used as a reference when implementing my system next term.

These diagrams will help to fully understand the connection between all aspects of the system and deduce how the users will interact with it. The diagrams will also help outline the key processes of the system without going into too much detail.

Work to be completed during next term:

For the future term I will begin the implementation stage. During this stage I will concentrate on the core functionality:

- Provide a set of calls (Call List) for each district nurse.

The following are additional functionality that I will look at if I have time after core functionality it completed to a high standard:

- Staff and patients details can be changed.

A tertiary function that will be looked at if previous functionality is completed will be:

- Time constraints.

A quaternary function that will be looked at if all previous functionality is completed will be:

- Call reallocation.

These functions have been agreed with my client.

Glossary of Terms:

Here are a number of terms:

- Patient details – data which relate to a living individual who can be identified from those data. For example, DOB and address etc.
- Staff details – for example telephone/contact number and band level etc.
- Users – District nurses.
- System – The environment which displays information to the user.
- Nurses – for the purpose of this core system I will only have band 5 and band 6 nurses available to take calls.
- Band – the level of skills a staff member has.
- Administrator – for the purpose of hits core system the administrator will be a band 7 nurse.
- Bank Nurse – When a regular member of staff is unable to attend work and a bank nurse is called in to cover the workload.
- Data/Details – personal information stored about patients and staff.
- Bloods – calls which need to be carried out before 12pm.
- Duty List – List of duty 2 months in advance including all members of staff.
- Key code – is the key pad code which enables access to some patient properties.
- Pin Number – the number given to a nurse when they qualify.
- Bargoed office – The office in which this system would be based if it were to be released.
- Nurse – District nurse.

Appendices

Week by Week Plan:

Autumn semester:

There will be a weekly meeting both with the client and moderator.

Week 1:

- Gather information from client in order to have clear view on what the client requires.
- Write up draft initial plan.

Week 2:

- Edit initial plan.

Week 3:

- Create Gantt chart.
- Finalise initial plan and hand in.
- Research background of the day to day running of district nurses.
- Research for project.
- Finalise what is required for the project to meet all requirements.
- Gather fundamental and non-fundamental requirements.

Week 4:

- Research waterfall and other models best suited for this project.
- Research any existing systems.
- Gather acceptance criteria and assumptions for the project.
- Explain the approach how the problem will be solved.
- Work out all users required for the systems.
- UML diagrams for primary functionality.

Week 5:

- Draw some templates of what the system could look like.
- Develop algorithm to allocate calls.
- Research programming languages to use.

Week 6:

- Database design and familiarise with programming languages.

Week 7:

- Familiarise with SQLite.

Week 8:

- Algorithm research.
- Begin to write up interim report.

Week 9:

- Continue to write up interim report.

Week 10:

- Send interim report to supervisor.
- Edit interim report based on feedback.

Week 11:

- Finalise report.
- Submit report.

Spring semester:

Primary functionality:

Week 1:

- Begin Implementation – aim to have interfaces created and algorithm.

Week 2:

- Continue with Implementation – aim to have all interfaces created and begin working on creating databases.

Week 3:

- Continue implementing system making sure basic aspects of the system are on track.
- Create test cases for core functionality.

Week 4:

- Continue with Implementation – Integrate databases and interfaces together.

Week 5:

- Continue implementing system making sure the system is tested based on test case criteria.
- Make sure system is working correctly - Testing of core functionality.

Week 6:

- Test system and fill in test cases.
- Finalise implementation. Make sure the system is complete and ready to be fully tested.
- Move on to additional functionality once core functionality has been completed.

Week 7:

- Edit system if any test cases fail.
- Begin writing up the final report.

Week 8:

- Write up Evaluation.
- Continue writing up the final report.

Week 9:

- Continue writing up the final report.

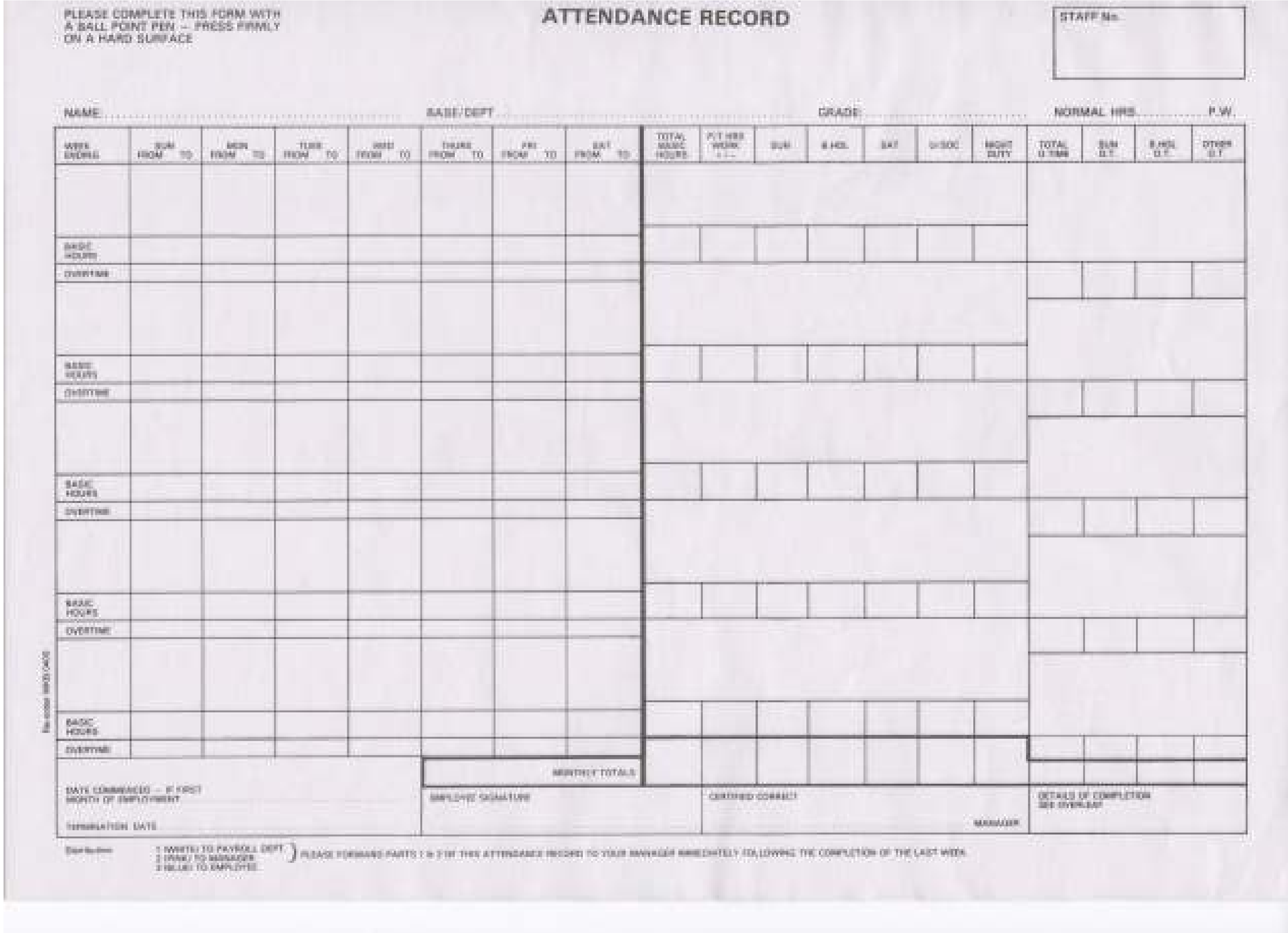
Week 10:

- Finish gathering work in order to finalise final report.
- Send draft to supervisor and moderator.

Week 11 (week ending 5th May):

- Finish final report and hand in.

Time sheet:



PLEASE COMPLETE THIS FORM WITH
A BALL POINT PEN — PRESS FIRMLY
ON A HARD SURFACE

ATTENDANCE RECORD

STAFF No.

NAME .... BASE/DEPT .... GRADE .... NORMAL HRS .... P.W.

WEEK ENDING	SUN		MON		TUES		WED		THURS		FRI		SAT		TOTAL BASIC HOURS	P/T HRS WORK	SUN	B.HOL	SAT	U/SOC	NIGHT DUTY	TOTAL O/TIME	SUN O.T.	B.HOL O.T.	OTHER O.T.
	FROM	TO	FROM	TO	FROM	TO	FROM	TO	FROM	TO	FROM	TO	FROM	TO											
BASIC HOURS																									
OVERTIME																									
BASIC HOURS																									
OVERTIME																									
BASIC HOURS																									
OVERTIME																									
BASIC HOURS																									
OVERTIME																									
									MONTHLY TOTALS																

DATE COMMENCED — IF FIRST MONTH OF EMPLOYMENT

TERMINATION DATE

EMPLOYEE SIGNATURE

CERTIFIED CORRECT

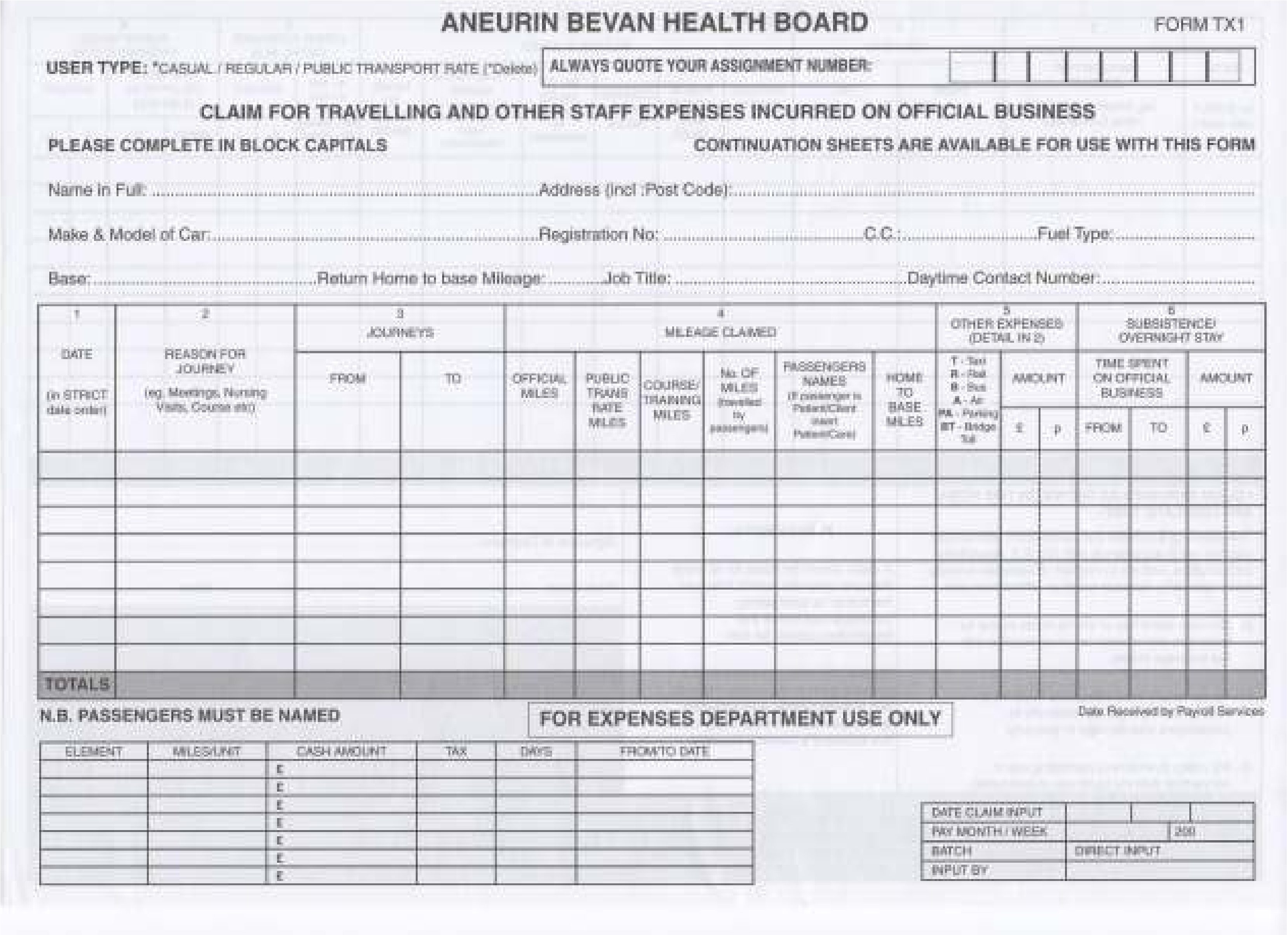
MANAGER

DETAILS OF COMPLETION SEE OVERLEAF

Distribution: 1 (WHITE) TO PAYROLL DEPT
2 (PINK) TO MANAGER
3 (BLUE) TO EMPLOYEE

PLEASE FORWARD PARTS 1 & 2 OF THIS ATTENDANCE RECORD TO YOUR MANAGER IMMEDIATELY FOLLOWING THE COMPLETION OF THE LAST WEEK

Mileage sheets:



ANEURIN BEVAN HEALTH BOARD

FORM TX1

USER TYPE: *CASUAL / REGULAR / PUBLIC TRANSPORT RATE (*Delete)

ALWAYS QUOTE YOUR ASSIGNMENT NUMBER: □□□□□□□□□

CLAIM FOR TRAVELLING AND OTHER STAFF EXPENSES INCURRED ON OFFICIAL BUSINESS

PLEASE COMPLETE IN BLOCK CAPITALS

CONTINUATION SHEETS ARE AVAILABLE FOR USE WITH THIS FORM

Name in Full: .... Address (incl :Post Code): ....

Make & Model of Car: .... Registration No: .... C.C.: .... Fuel Type: ....

Base: .... Return Home to base Mileage: .... Job Title: .... Daytime Contact Number: ....

1	2	3 JOURNEYS		4 MILEAGE CLAIMED						5 OTHER EXPENSES (DETAIL IN 2)			6 SUBSISTENCE/ OVERNIGHT STAY			
DATE (in STRICT date order)	REASON FOR JOURNEY (eg. Meetings, Nursing Visits, Course etc)	FROM	TO	OFFICIAL MILES	PUBLIC TRANS RATE MILES	COURSE/ TRAINING MILES	No. OF MILES (travelled by passengers)	PASSENGERS NAMES (if passenger is Patient/Client insert Patient/Care)	HOME TO BASE MILES	T - Taxi R - Rail B - Bus A - Air PA - Parking BT - Bridge Toll	AMOUNT		TIME SPENT ON OFFICIAL BUSINESS		AMOUNT	
											£	p	FROM	TO	£	p
TOTALS																

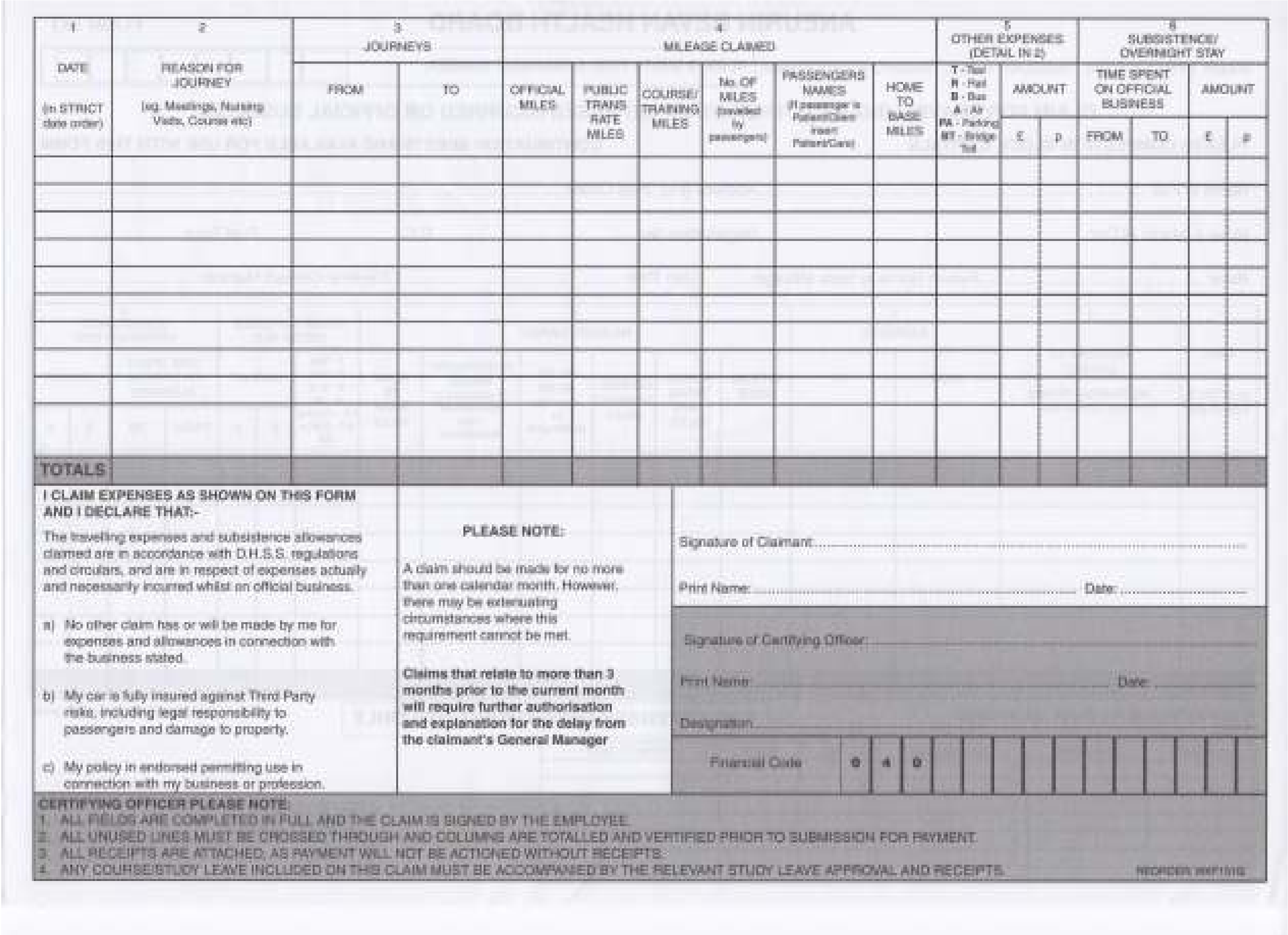
N.B. PASSENGERS MUST BE NAMED

FOR EXPENSES DEPARTMENT USE ONLY

Date Received by Payroll Services

ELEMENT	MILES/UNIT	CASH AMOUNT	TAX	DAYS	FROM/TO DATE
		£			
		£			
		£			
		£			
		£			
		£			
		£			

DATE CLAIM INPUT			
PAY MONTH / WEEK		200	
BATCH	DIRECT INPUT		
INPUT BY			



1	2	3		4						5			6			
DATE (in STRICT date order)	REASON FOR JOURNEY (eg. Meetings, Nursing Visits, Course etc)	JOURNEYS		MILEAGE CLAIMED						OTHER EXPENSES (DETAIL IN 2)			SUBSISTENCE/ OVERNIGHT STAY			
		FROM	TO	OFFICIAL MILES	PUBLIC TRANS RATE MILES	COURSE/ TRAINING MILES	No. OF MILES (travelled by passengers)	PASSENGERS NAMES (If passenger is Patient/Client insert Patient/Carer)	HOME TO BASE MILES	T - Taxi R - Rail B - Bus A - Air PA - Parking BT - Bridge Toll	AMOUNT		TIME SPENT ON OFFICIAL BUSINESS		AMOUNT	
											£	p	FROM	TO	£	p
TOTALS																

I CLAIM EXPENSES AS SHOWN ON THIS FORM AND I DECLARE THAT:-

The travelling expenses and subsistence allowances claimed are in accordance with D.H.S.S. regulations and circulars, and are in respect of expenses actually and necessarily incurred whilst on official business.

a) No other claim has or will be made by me for expenses and allowances in connection with the business stated.

b) My car is fully insured against Third Party risks, including legal responsibility to passengers and damage to property.

c) My policy in endorsed permitting use in connection with my business or profession.

PLEASE NOTE:

A claim should be made for no more than one calendar month. However, there may be extenuating circumstances where this requirement cannot be met.

Claims that relate to more than 3 months prior to the current month will require further authorisation and explanation for the delay from the claimant's General Manager

Signature of Claimant: ....

Print Name: .... Date: ....

Signature of Certifying Officer: ....

Print Name: .... Date: ....

Designation: ....

Financial Code	0	4	0												

CERTIFYING OFFICER PLEASE NOTE:

1. ALL FIELDS ARE COMPLETED IN FULL AND THE CLAIM IS SIGNED BY THE EMPLOYEE.
2. ALL UNUSED LINES MUST BE CROSSED THROUGH AND COLUMNS ARE TOTALLED AND VERTIFIED PRIOR TO SUBMISSION FOR PAYMENT.
3. ALL RECEIPTS ARE ATTACHED, AS PAYMENT WILL NOT BE ACTIONED WITHOUT RECEIPTS.
4. ANY COURSE/STUDY LEAVE INCLUDED ON THIS CLAIM MUST BE ACCOMPANIED BY THE RELEVANT STUDY LEAVE APPROVAL AND RECEIPTS.

REORDER [illegible]

References:

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

Jitomirsky, E, 2012. *The Quick Sort Algorithm, 2012.* [Online] Available at: http://www.mycstutorials.com/articles/sorting/quicksort [Accessed: 4th December 2012]

My Surgery Website, 2012. Clinics and Services, 2012. [Online] Available at: http://www.shaylane.org/page1.aspx?p=1 [Accessed: 3rd December]

Neo, (). Savings Algorithm, [Online] Available at: http://neo.lcc.uma.es/vrp/solution-methods/heuristics/savings-algorithms/ [Accessed: 12th December 2012]

Neo, (). Capacitated VRP, [Online] Available at: http://neo.lcc.uma.es/vrp/vrp-flavors/capacitated-vrp/ [Accessed 12th December 2012]

Virginia Polytechnic Institute and State University, 2007. *Selection Sort Algorithm 2007.* [Online] Available at: http://courses.cs.vt.edu/csonline/Algorithms/Lessons/SelectionCardSort/selectioncardsort.html [Accessed: 4th December 2012]

Wiseman, R, 2012. *Analysis of Algorithms, 2012*. [Online] Available at: http://homepages.ius.edu/RWISMAN/C455/html/notes/Chapter2/DivConq.html [Accessed: 3rd December 2012]

XoaX, 2012. *Lesson 3: Merge Sort, 2012*. [Online] Available at: http://xoax.net/comp_sci/crs/algorithms/lessons/Lesson3/ [Accessed: 4th December 2012]