Report

Contents

[Introduction: 1](#_Toc159485721)

[Aims and objectives: 1](#_Toc159485722)

[Objectives: 2](#_Toc159485723)

[Literature Review: 2](#_Toc159485724)

[Tools and methodologies: 2](#_Toc159485725)

[Current and new methods and technologies: 3](#_Toc159485726)

[Related Systems: 3](#_Toc159485727)

[MIMIC-II 3](#_Toc159485728)

[NHS - SystmOne 4](#_Toc159485729)

[Requirements: 5](#_Toc159485730)

[Project Planning & Team Roles: 6](#_Toc159485731)

[Implementation & Testing: 7](#_Toc159485732)

[Evaluation & Lessons Learned: 7](#_Toc159485733)

[References: 7](#_Toc159485734)

[Appendices: 8](#_Toc159485735)

# Introduction:

# Aims and objectives:

This project aims to develop a hospital dashboard which takes in patient data and flags the patients who are most in need of seeing a dietitian.

## Objectives:

1. We need to research similar projects so we can see what we can do differently.
2. We must assign specific roles to different team members to distribute the workload.
3. We need to come up with the design and test plan which will be figuring out the technical requirements and evaluating the use cases.
4. We must implement and test the project to ensure functionality and reliability.
5. We then need to evaluate which requirements we have met and or missed to see whether this project was successful.

# Literature Review:

There is currently a need for the critical care unit to start prioritising patients who need to see a dietitian because there are insufficient resources for every patient to see a dietitian and the patients who need to see a dietitian the most may currently miss out. This is because it is very difficult for the critical care unit staff to efficiently prioritize patients.

So, developing a feeding dashboard which will flag the patients who need to see the dietitian, will aid the staff significantly, optimising and increasing healthcare resources due to less time needed to prioritize patients this will make sure that patients with a greater need get the help required.

## Tools and methodologies:

The tool we will use to plan this project is Astah UML. Astah UML is a program which allows for the creation of UML diagrams like the use case diagram. Using UML helps to define the scope of the project abstracting it into easily digestible sections. (Fernández-Sáez, Chaudron and Genero, n.d.). We have chosen Astah UML it’s written in Java so it will run on any device, allowing team members on a Mac to contribute to creating UML diagrams.

The tool we will use to manage this project is Gitlab - a version control software which stores a project in a repository. Using Gitlab will allow multiple team members to simultaneously work on the project at the same time while avoiding conflicts because changes to the same file are merged (Perez-Riverol *et al.*, 2016). This will make collaboration easier between team members and make developing the project much easier.

The software development methodology we are going to use to manage this project is a modified version of Scrum, which is an agile software development methodology. Using an agile software development methodology allows for easier collaboration between team members and stakeholders (Karrenbauer, Wiesche and Krcmar, 2019), which means that if the requirements change during the development life cycle we can adapt. We are using a modified version of Scrum because the Scrum methodology includes daily meetings called ‘daily Scrum’ (Schwaber and Sutherland, 2020), which we will modify into 2 weekly meetings instead. This will help the team collaborate while easily fitting it in with our timetables. A similar agile methodology we could have used is Xtreme Programming (Beck, K. and Andres, C. (2004)). This methodology is a more intensive process, with vigorous testing and revaluation taking place at every stage of development. While this would’ve provided the benefit of continuous bug checking to make robust code, this would not have been feasible to implement, as the group is not able to meet often enough to have the required amount of reflection and discussion. In contrast, Scrum still functions well on a less frequent basis.

## Current and new methods and technologies:

One technology we will use to implement this project is the Python programming language. Python is a cross-platform, multi-paradigm programming language (‘Python (programming language) ~ Information Technology ~ 2420 ~ kelas-karyawan-bali.kurikulum.org’, n.d.) meaning that it supports procedural and object-oriented programming. Therefore, choosing Python for programming this project will allow for the program to run on any platform without having to write multiple versions, reducing the time to develop. Furthermore, Python’s support for object-oriented programming will allow for the encapsulation of multiple complex datatypes, improving maintainability.

Another technology that we will use to implement this project is Tkinter. Tkinter is a library that comes with Python, which is used to create graphical user interfaces (Moore, 2018). Using Tkinter means that fewer extra libraries will need to be installed to run the program, making it simpler to deploy. Furthermore, being able to create a Graphical User Interface will make the program easier to use, meaning that less training will be required.

## Related Systems:

### MIMIC-II

The Multiparameter Intelligent Monitoring in Intensive Care II (Saeed, M *et al.,* 2011) was a system developed by academics to produce diagnostic and therapeutic data from a large population of adult Critical Care Unit patients. Patients in a CCU had data values associated with them, such as medication, test results etc. These values were used to create a public-access database for use in various medical research.

MIMIC-II’s functionality very closely resembles the nutrition dashboard that we are looking to develop. We are also looking to use many metrics of patient data, but rather than simply develop a database, we are looking to create an interactive GUI. Therefore, the likes of reports and summaries don’t have to be outsourced to another application, as we plan to have these functionalities built into our system. We can do this as the data we are going to use is already available to us – much of the purpose of MIMIC-II was the initial data collection, having to synchronise data from many different databases throughout the healthcare system.

### NHS - SystmOne

The UK’s National Health Service has a vast amount of data equating to over 80 million patient records (NHS, 2024). Many efforts have been made in recent years to digitalise the handling of their records, to maintain accuracy and consistency, as well as produce reports and summaries which can be used to aid future research into healthcare.

The NHS’ data is stored in a Personal Demographic Service (NHS, 2024). This acts as a bulk database for every patient on record. However, this does not contain any form of user interface for data handling. The PDS is instead interacted with from several solutions. For example, a hospital receptionist using a *Patient Administration System*, and a citizen using the NHS app will both be transferring data to and from the PDS.

GP surgeries need to often interact with the PDS. To do this, they will have a standardised system in their network. The three most used are *EMIS Web, SystmOne* (TPP, 2024)and *Vision* (GP Training Support, 2016)*.* These products bear a visual resemblance to the dashboard which we will be looking to develop.

*SystmOne:*

SystmOne facilitates access to both patients and medical staff. Patients can use the service to register or change their details, as well as being able to order prescriptions.

The staff can use the service to view patient records and add any updates about recent appointments or changes to medication. There is also plenty of functionality to produce reports on the data:

* Clinical Reports
  + Reports created for local use, using whatever data is required.
  + i.e. A report on one GP surgery’s new admissions during February.
* National Reports
  + Reports intended to be viewed by many entities nationally.
  + Will uphold national standards of report structure.
* Strategic Reporting
  + Bulk data extract for use in another application, or to be sent to another organisation.

These features allow medical staff to produce any required reports quickly and effectively. This is especially true when compared to before digitalisation, where the NHS would have to collate data from however many surgeries/hospitals throughout the UK.



Fig 1.0 – A snapshot of the Clinical Record Viewer in SystmOne (Adewunmi, 2014).

In Fig 1.0, we can what the SystmOne application looks like while running on a staff computer. The user is presented with a Graphical User Interface, which has many features that we can use as inspiration for our dashboard:

* Side menu containing different functions.
* Main window which changes depending on the side menu selection.
* Ribbon of options displayed across the top of the screen.
  + Remain consistent regardless of function selection.
* Effective use of contrasting colours for easy visibility.

# Requirements:

# Project Planning & Team Roles:

* Show patients admitted to the CCU

This task involves one of the team members writing Python code to open and read a CSV (comma-separated values) file and display its contents. Ideally, the code allows the user to choose a specific file they want to read and display as opposed to the file being hardcoded. Also, the contents of the file should be displayed in a Tkinter window in the form of a table. Realistically one person should be able to complete this task on their own in a short timeframe as this task does not have any complex requirements.

* Make an intuitive GUI using Tkinter

This task involves the creation of a User Interface which would be used to navigate between the different functions of our software. We’ll be using the tkinter framework that comes pre-packaged with Python. This will likely be one of the harder tasks and as such more members of the team would have to be allocated to it and it might take longer to complete.

* Write code that generates reports and graphs based on patient data

The objective of this task is to write code that interprets the data of individual patients and generates reports based on said data. It should also be able to look at the data of multiple patients and generate reports on them as well. The reports could be either in a table or graphically displayed in the form of multiple graphs. The task itself won’t be the most difficult in all likelihood and as such it won’t have that many resources dedicated to it. Perhaps two people should be able to complete it on time.

* Implement algorithms that evaluate patient data

This task aims to make a functional implementation of the algorithms provided in the case study. These algorithms would take in the data of individual patients and if they deem it necessary, the patient would get flagged to see a dietitian. This task could pose some difficulties and as such more resources would have to be allocated.

(Obviously, I can’t accurately predict how difficult each task would be, how many people would be working on it etc. These are just placeholders in reality we would do it first then edit the plan to match our approach. I based the objectives based on the case study and this is a mostly exhaustive list. If anyone wishes to add anything, by all means, feel free to do so :3)

# Implementation & Testing:

# Evaluation & Lessons Learned:

# References:

Fernández-Sáez, A., Chaudron, M. and Genero, M. (no date) *Exploring Costs and Benefits of Using UML on Maintenance: Preliminary Findings of a Case Study in a Large IT Department* [online]. Available from: <https://ceur-ws.org/Vol-1078/paper4.pdf>.

Perez-Riverol, Y. *et al.* (2016) Ten Simple Rules for Taking Advantage of Git and GitHub. Markel, S., ed. *PLOS Computational Biology* [online]. 12 (7), p. e1004947. [Accessed 1 July 2020].

Karrenbauer, J., Wiesche, M. and Krcmar, H. (2019) Understanding the Benefits of Agile Software Development in Regulated Environments. *Wirtschaftsinformatik 2019 Proceedings* [online]. Available from: <https://aisel.aisnet.org/wi2019/track07/papers/5/>.

Schwaber, K. and Sutherland, J. (2020) *Scrum Guide | Scrum Guides* *Scrumguides.org*. November 2020 [online]. Available from: <https://scrumguides.org/scrum-guide.html>.

‘Python (programming language) ~ Information Technology ~ 2420 ~ kelas-karyawan-bali.kurikulum.org’ (no date) *kelas-karyawan-bali.kurikulum.org* [online]. Available from: <http://kelas-karyawan-bali.kurikulum.org/IT/en/2420-2301/Python_3721_kelas-karyawan-bali-kurikulumngetesumum.html>.

Moore, A.D. (2018) *Python GUI Programming with Tkinter: Develop responsive and powerful GUI applications with Tkinter* *Google Books* [online]. Packt Publishing Ltd. Available from: https://books.google.co.uk/books?hl=en&lr=&id=2kBbDwAAQBAJ&oi=fnd&pg=PP1&dq=tkInter&ots=NgppXWINnl&sig=1G9Rr3C3mGPDv757\_ZZdg1cuk60&redir\_esc=y#v=onepage&q=tkInter&f=false [Accessed 12 February 2024].

NHS (2024) Personal Demographics Service. Available from: https://digital.nhs.uk/services/personal-demographics-service [Accessed 11/02/2024].

GP Training Support (2016) Computer systems in general practice. Available from: <https://gptraining.info/computer-systems-practice/#:~:text=The%20main%20systems%20are%20EMIS%20Web%2C%20SystmOne%2C%20and%20Vision>. [Accessed 11/02/2024].

TPP (2024) SystmOne. Available from: https://tpp-uk.com/products/ [Accessed 11/02/2024].

Adewunmi, A. (2014) Medway Maritime Hospital Case Study. A Critical Appraisal of the Summary Care Record (Scr) Scheme in England [online]., p. 5. [Accessed 12 February 2024].

Beck, K. and Andres, C. (2004) Extreme Programming Explained: Embrace Change. 2nd ed. : Addison-wesley Professional.

Saeed, M., Villarroel, M., Reisner, A.T., Clifford, G., Lehman, L.-W., Moody, G., Heldt, T., Kyaw, T.H., Moody, B., Mark, R.G.: Multiparameter Intelligent Monitoring in Intensive Care II: A public-access intensive care unit database\*: Critical Care Medicine. 39, 952–960 (2011).

# Appendices: