SOFTWARE DESIGN

Assignment 1 – Group report

Group name: Cyber ATTAK

Awaiz Solkar (L39220265)
Toseef Ashraf (L39220303)
Tapiwa Matukutire (L39220273)
Arini Rajes (L39222543)
Katie Powl (L39220257)

Contents

List of figures

Introduction

What is SDLC?

The Software Development Life Cycle (SDLC) is a structured process that allows a cost-effective and time efficient way of designing and building high quality software. “It is an industry recognised procedure comprising a sequence of activities or phases designed for creating a new enterprise software application/product or modifying existing software” (Senthilvel, Khan, & Qureshi, 2017). The life cycle involves all aspects from getting the requirements from the client to the planning stage to the deployment of the product.

What is Software Development?

According to IBM, Software Development “refers to a set of computer science activities dedicated to the process of creating, designing, deploying and supporting software” (IBM, 2019). The process is based on several phases within the development life cycle. Some of these phases include but are not limited to: collaboration from the client via elicitation techniques of their requirements; project planning which is based on the clear and concise requirements; risk management throughout the development process; version control and defensive development; quality control of the software product and finally, presentation and delivery of the final software iteration. Successful software development requires comprehensive planning complimented by effective process management (Pilone and Miles, 2007).

Introduction to the project

The project was conducted on behalf of a client, Trafford Pharmaceuticals. The objectives were to collaborate with the client on their requirements for a functional management system that oversees inventory management, syncs sales to current inventory levels in real time and finally ensures that dangerous drugs are administered appropriately. The Medical sector is tightly regulated and thus, the management system needs to assist the client to meet their operational requirements as they expand reliably. Currently, Trafford Pharmaceuticals are running outdated systems that operate as separate procedures. These procedures must be manually synchronised which is inefficient and leaves the firm exposed to risks of human error. The proposed management system seeks to solve these issues and improve efficiency (Koch, 2005).

Introduction to Waterfall method

Trafford Pharmaceuticals are an SME business. According to the British Government Website, a small to medium sized business (SME) can be defined as “any organisation that has fewer than 250 employees and a turnover of less than €50 million or a balance sheet total less than €43 million” (UK Government, 2023). The client falls within this criterion and therefore the Waterfall model was deemed the most suitable approach for this project. The Waterfall methodology “follows a chronological process and works based on fixed dates, requirements, and outcomes” (Adobe, 2022). Typically, one phase completes before proceeding to the next phase, as illustrated by Figure 1:

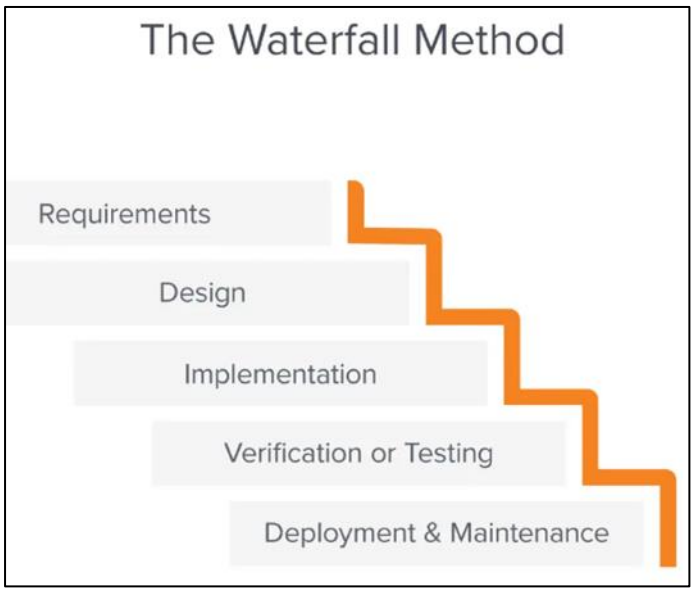

Figure 1: Waterfall method. (Adobe, 2022)

Requirement Engineering

Introduction to Requirement Engineering

Requirements engineering (RE) is a systematic process in software engineering which involves gathering, analysing and documenting requirements for a system/project. RE is one of the critical phases of the Software Development Life Cycle as it ensures that the end product meets the needs and expectations of stakeholders and users. In context of the pharmacy management system scenario, RE is key for defining the functionalities and features for the to-be system.

Why is Requirement Engineering important?

It can be argued that requirement engineering is “the most important aspect of a software project since it establishes the foundations for all project tasks” (Wiegers, 2010). It helps with the understanding of objectives, constraints and functionalities of a software system which then minimises the risk of project failure. By using requirement engineering techniques, potential issues can be identified very early in the project which will then lead to a cost-effective solution and increase the chances of success.

How Requirement Engineering was used to create Requirements Document

The requirement engineering process played a crucial role to complete the analysis and gather all the user/stakeholder needs by using two requirement gathering methodologies. The detailed information acquired from the requirement gathering influenced the content of the requirements document (refer to appendix 1).

The first methodology that was used for the requirement gathering was interviews. “The interview method stands out as the most effective and efficient approach for collecting requirements. Using this strategy allows direct engagement with stakeholders and key users to understand their needs and goals for implementing a software system” (Pal, 2020).

Justification: Interviews allowed direct communication which provided an opportunity to dive deeply into the users’ needs and expectations. A thorough understanding was gained through in-depth discussions which may not have been possible through other requirement gathering methods. Compared to user observation which may take an extensive time, interviews are time efficient as it involves a focused discussion. This

was great for the pharmacy management system project which has time constraints. Although interviews required pre-planning and availability, an advantage of this was being able to plan a set of open-ended questions and documentation in order to capture the verbal communication for reference through the project.

The next methodology that was conducted for the requirement gathering was user observations. This method is “the act of physically watching a process being performed and is a powerful tool for gathering information about an existing system enabling the analyst to see the reality of the situation” (Dennis, Wixom, & Tegarden, 2012).

Justification: An active observation method allowed the team to get a deeper understanding of an existing pharmacy management system and its process in real time. Despite the fact the observation was time consuming, an advantage is that it checked the validity of information that was given through interviews and meetings. In relation to this it also allowed implicit requirements to be uncovered as stakeholders were not aware of every need or did not communicate it through the interview. This method also allowed the team to see real-life problems with the management system as they occurred through the observation which gave a better understanding of the issues.

Justification of sections in the Requirements Document

The requirement document is vital for the project as it helps the team understand the needs of the client. Therefore, each section of the requirements document serves a crucial purpose. The introduction sets the context, functional requirements define system capabilities, non-functional requirements specify constraints, and use cases provide detailed scenarios. These sections collectively ensure clarity, completeness, and alignment with the pharmacy management system's unique needs and goals.

Figure 2 shows the justification for each section in more detail:

Requirement document sections	Justification
Document revisions	This section is essential for keeping an organised record of any changes that are made to the requirements document through the pharmacy management system project. It ensures stakeholders are able to keep track of the modification history as the document provides a clear audit trail
Approvals	This section is crucial for establishing a formal agreement between the relevant parties involved in the project and making sure the documentation accurately represents the requirements/needs. It ensures validation and is proof that the document has undergone review.
Project intro/scope	This section is one of the most important as it sets the stage for the whole pharmacy management system project. It gives a brief overview of the project as it outlines the purpose, goals, objectives and constraints
Business process overview	This section is critical for outlining the current process of the existing system and the future process of the system. It helps contextualise all the requirements and serves as a base for understanding how the to-be system will integrate and impact the current business process
Functional requirements	This section is vital for the actual development and testing of the project as it specifies the system features, capabilities and behaviours. It is the main section of the document which highlights what the pharmacy management system is expected to do using "shall" statements. When these requirements are fulfilled, it would mean user/stakeholder needs have been met.
Non-functional requirements	Going hand in hand with functional requirements, this section is also vital for the development and testing of the project. This section addresses the qualities and attributes the pharmacy management system should have. The non-functional requirements can guide the system designers in making important decisions about the performance and quality of the system.
Risks and issues	This section is important for risk management as it identifies potential risks and social, ethical, professional and legal issues in the requirement gathering stage. By highlighting the risks and issues it enables the team to mitigate them and have contingency plans to minimise the impact of potential issues during the requirements gathering.

Figure 2: Justification of document sections

Justification of choices made, in relation to specific criteria

Every choice in the requirements document aligns with the pharmacy management system scenario. Security measures address patient data protection, inventory management choices optimise operations, user authentication ensures data integrity, usability focuses on staff with varying technical expertise, integration meets healthcare ecosystem demands, and scalability anticipates future growth. Each choice strategically fulfils specific scenario criteria.

Figure 3 shows the justification of choices in more detail:

Choices made	Justification
Security requirements	Security measures were considered as the pharmacy management system deals with sensitive data such as patients' personal details and medical information. Access control and encryption were needed to ensure that authorised personnel can access and manage the data. This aligns with the pharmacy's critical need for safeguarding patient confidentiality and complying with legal standards.
User authentication	Similar to security requirements, user authentication was considered due to the nature of sensitive data. Role-based access control was needed so that different functionalities of the system can be accessed by the relevant personnel such as the pharmacists and admins. This choice aligns with the pharmacy's focus on data integrity.
Scalability	Scalability was considered after anticipating the potential growth of the pharmacy. This called for certain choices to be made in the system design to ensure it allowed easy adaptation for increased data and additional patients. The scalability aligns with the pharmacy's long-term vision for growth.
Product/stock management	Efficient management methods were needed to handle the pharmacy's products. The choice to include stock related requirements was to streamline processes. This aligns with the pharmacy's aim to minimise stockouts, be cost-efficient, reduce wastage of supplies and overall enhance the stock management.
User-friendly experience	After realising the pharmacy staff have different technical expertise, the user interface of the management system required a user-friendly design. This aligns with the pharmacy's desire to have minimal training time and have less errors occur. The requirements focused on things such clear text/labelling, simple navigation and helpful prompts on the pharmacy management system to enhance user experience.
External system integration	The pharmacy management system needs to seamlessly integrate with external systems such as the GP and healthcare databases. This allows smooth communication and operations within the business. This aligns with the pharmacy's requirement for efficient collaboration and information exchange with external entities which will help with the patient's journey.

Figure 3: Justification of choices made

System Modelling

Introduction to the types of diagrams used

The chosen diagrams for the pharmacy management system are Flowcharts and Entity Relationship Diagrams (ERD). “Flowcharts enable users to visualise the sequential progression of instructions within an application, illustrating the path from start to finish” (Balti & Galloway, 2021). “An Entity Relationship (ER) Diagram is similar to a flowchart, offering a visual representation that showcases the connections and associations among different "entities" such as individuals, objects, or concepts within a database system” (Lucidchart, n.d.).

Justification of diagram choices

Flowcharts were chosen to illustrate the sequential flow of processes within the pharmacy management system and help in understanding prescription handling, inventory management, and other operational workflows. “Flowcharts offer a comprehensive overview of a process, enabling stakeholders to pinpoint bottlenecks, communication deficiencies, and highlight opportunities for improvements” (Miro, n.d.). ER diagrams were selected to portray the database structure, facilitating the design of efficient data storage and retrieval operations tailored to pharmacy-specific requirements.

Explanation of Entity Relationship Diagram

The Entity Relationship model (appendix 2) for Trafford Pharmaceuticals details the entities (or objects) and the relationships they share. In this instance, the central object consists of Trafford Pharmaceuticals as shown in figure 4. This entity shares binary relationships with the following entities: Medicines, Suppliers, External NHS Hospitals, Reports (user generated), Pharmacy assistants, Pharmacists and most importantly, Customers. The crow’s foot defines the type of relationship between the entities.

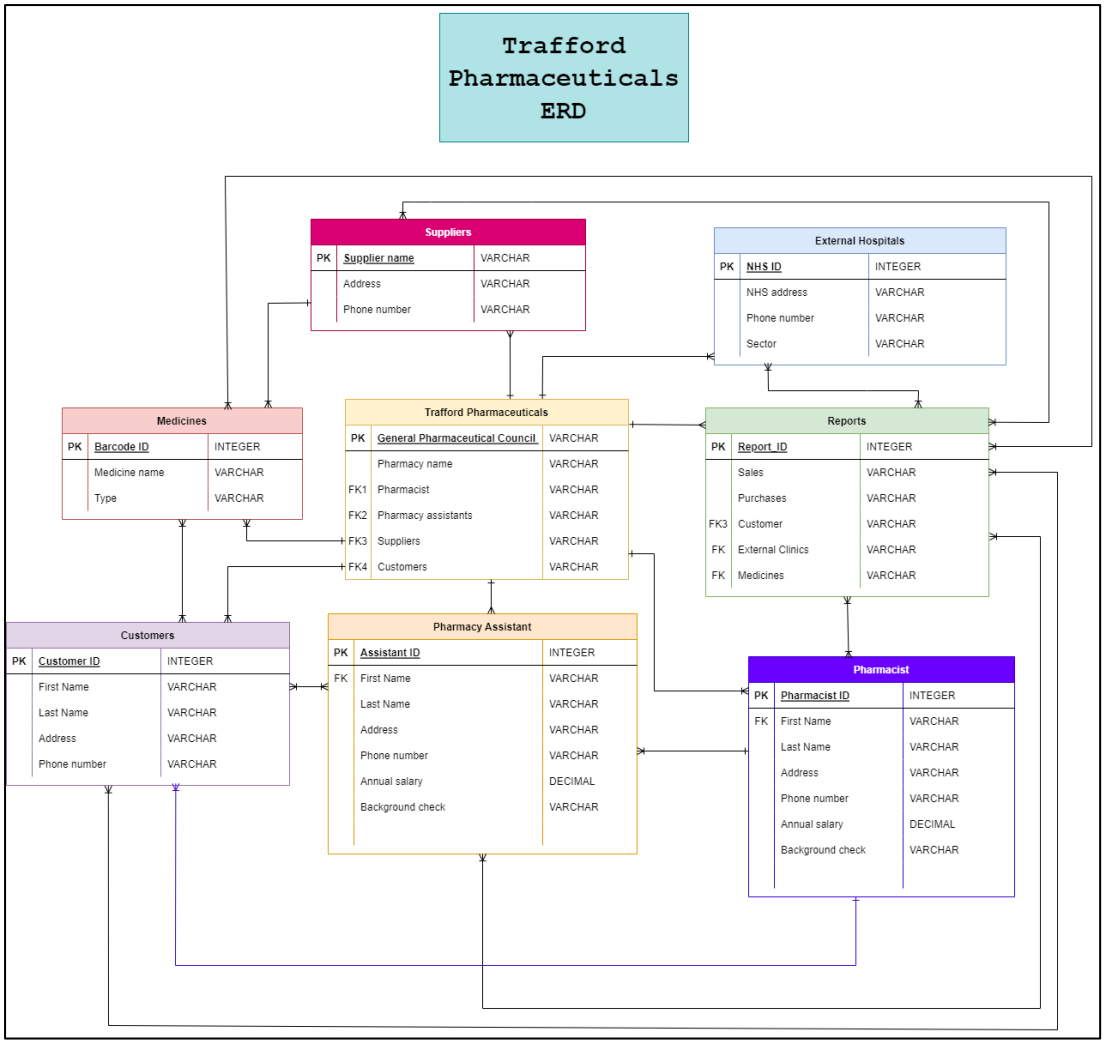

Figure 4, Pharmacy system ERD

To illustrate the cardinality between the entities within the management system's ecosystem, the relationship between Trafford Pharmaceuticals and its staff will be reviewed, as well as the relationship with the customers. Trafford Pharmaceuticals may employ one or many pharmacists; or alternatively one or many pharmacists may work for Trafford PLC. The ER-diagram will be utilised for designing the management system's backend database structure (Gillenson, 2011).

Explanation of Flowchart

The pharmacy management system flowchart (appendix 2) efficiently guides both staff and customers. It starts with user verification, distinguishing between customers and regular staff. Regular staff log in to update/check inventory, process prescriptions, handle payments, and view tasks. Online customers use the website/app for login, account creation, and product purchase. For in-store customers, the system processes prescription submissions, payments for non-exempt individuals, and handles in-store purchases. Arrows in the flowchart show decision points, guiding the sequence based on yes or no responses.

Figure 5 shows the flowchart of the pharmacy management system.

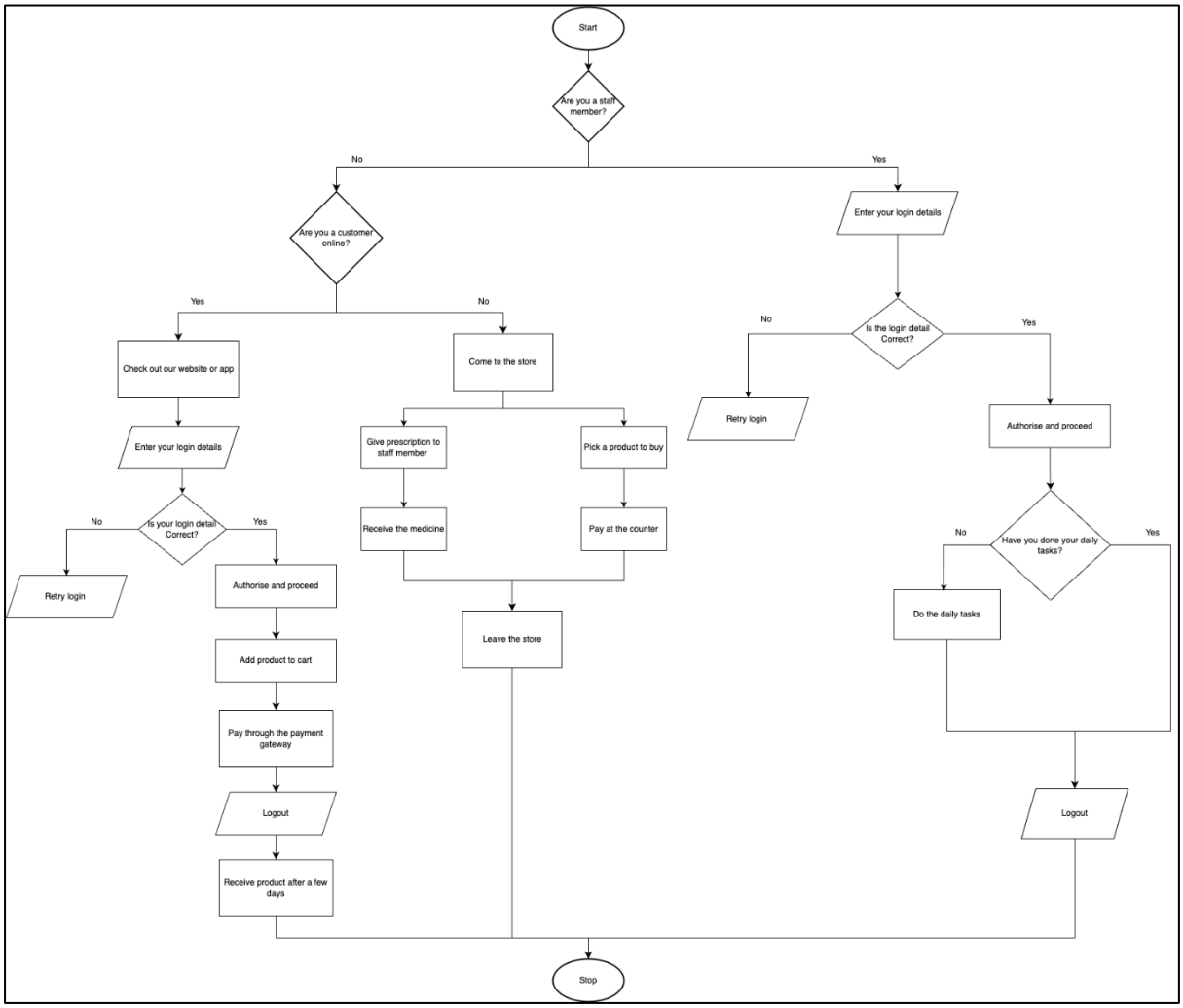

Figure 5, Pharmacy system Flowchart

SWOT analysis of diagrams used

Diagram:	Strengths	Weaknesses	Opportunities	Threats
Flowchart	•Are easy to understand •Gives a simple analysis of the process	•Modifications can be difficult. •Can be difficult to understand if complex	•Can be expanded into new industries	•Competition from other visual methods •Incorrect flowcharts can cause wrong decision making
ER Diagram	•Can be easy to understand •Are easy to scale •Have clarity due to clear and concise relationships	•Can be ambiguous depending on people's understandings	•Businesses can make templates to be time efficient •Can be created through automation based on existing databases	•Risk of misinterpretations •Maintenance can be time-consuming when updating changes

(PlanetTogether, 2020) & (Jaiswal, n.d.)

User Acceptance Criteria

What are User Acceptance Criteria?

“User Acceptance Criteria (UAC) are conditions that are predefined, which need to be met to successfully complete a user story, ensuring it meets their requirements whilst accounting for all outcomes” (Quick, 2023). They define the functional and non-functional expectations from the user's perspective and are used to assess the effectiveness and usability of a system.

Why are they important and used in practice?

Well-outlined User Acceptance Criteria ensures absolute clarity of user expectations of the developed system. By clearly defining acceptance criteria, stakeholders/users can objectively evaluate whether the pharmacy management system meets their needs and requirements. “A user story/requirement lacks completeness if there is no acceptance criteria, as without these criteria there is no way to measure if the project has fulfilled all expectations” (Productfolio, n.d.). It also ensures the project remains user-focused (Rehkopf, n.d.).

Justification of Acceptance Criteria (Scenario-oriented)

For the pharmacy management system, it was decided that the scenario-oriented model would be most appropriate. The scenario-oriented format illustrates the acceptance criteria of the different situations. (Altexsoft, 2023). This ensures that the system's functionality is relevant to practical usage scenarios encountered in a pharmacy environment. Overall, this approach improves usability by directly addressing user needs and workflow requirements through user stories.

Justification of level of detail in Acceptance Criteria

The level of detail in the Acceptance Criteria was justified based on the complexity of the system functionalities outlined in the user stories and requirements document (refer to appendix 1 & 3). Each scenario has the necessary detail to provide clear expectations for the system’s behaviour and performance.

Figure 6 shows a clear and concise sequence of phases in relation to a scenario.

Scenario	As a customer I want to order my prescription online and have it delivered around my busy schedule
GIVEN	The user has an online account
WHEN	The user wants their prescription
AND	The user has a prescription from their doctor showing on their account
THEN	The user logs in and orders the prescription
GIVEN	The user selects delivery method at checkout
WHEN	The user selects a specific date and time
THEN	The user proceeds to payment
AND	The user puts in their payment details
GIVEN	The user has completed the transaction
WHEN	The user receives tracking number
AND	The user waits for delivery
THEN	The user receives their prescription

Figure 6, Scenario-oriented testing (refer to appendix 3)

Justification of choices made, in accordance with Requirements Document

Every choice in defining the acceptance criteria was made in alignment with the requirements document for the pharmacy management system. The acceptance criteria correlates with the user stories that were gathered to ensure each feature is thoroughly tested and validated against the users' expectations.

Team working strategy

Methods of team working strategies

The “Cyber ATTAK” development team consists of five developers who collaborated to produce the Trafford Pharmaceuticals management system. The team have a lean organisation structure (Womack and Jones, 2003):

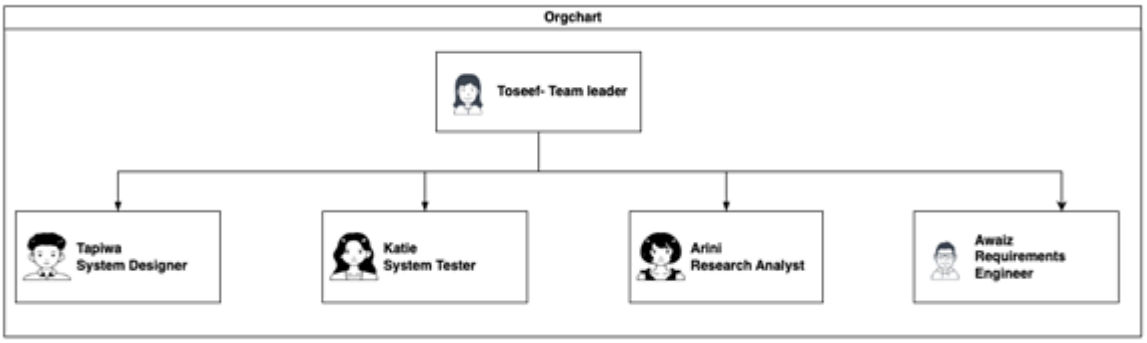

Figure 7, Member roles (Womack & Jones, 2003)

The flat team structure represented by the wiremold figure above, is consistent with “lean techniques”, according to Womack and Jones (2003, pp138) as the team members may be interchangeable depending on the human resource challenges faced. This structure aims to reduce the risk of downtime in the event of a team member being unavailable which enables the team to be adaptable.

The complimentary relationship of each role is illustrated in the workflow:

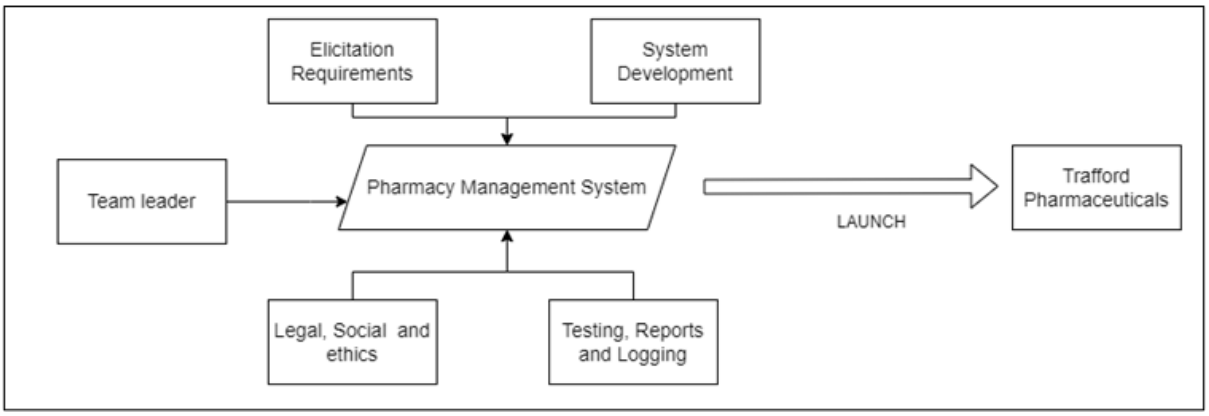

Figure 8, Workflow (Womack & Jones, 2003, pp126)

Justification of choices made in team working strategy

In adherence to the waterfall model for the pharmacy management system, the team adopted a more structured and sequential approach towards the project. Each phase of the SDLC was executed in an orderly manner with defined goals for each week. Different collaborative tools were used such as Monday.com and Notion for task management and Microsoft Teams for regular meetings. This allowed the team to collaborate and keep track of progress through the project lifecycle.

Figure 9 shows the use of Monday.com to create tasks and monitor the progress and contribution of each team member.

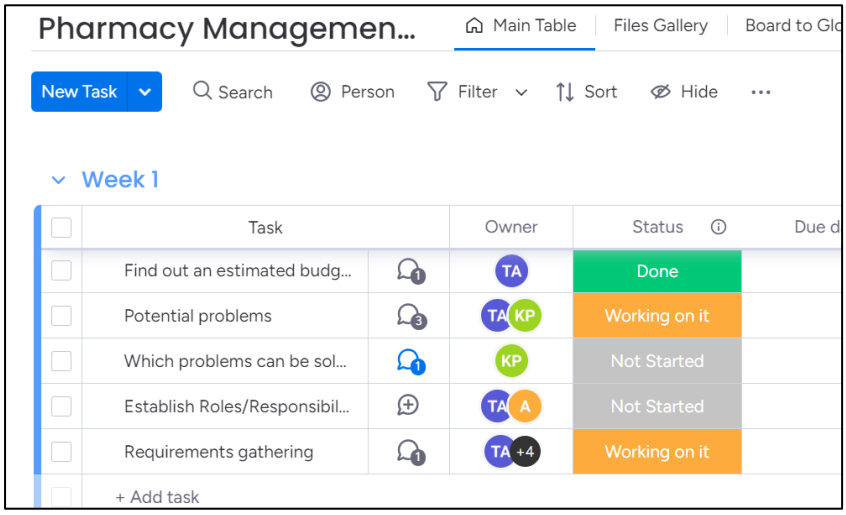

Figure 9, Monday.com collaboration

Team working experience (What went well, what didn't)

The Waterfall model brought clarity to the project by defining distinct phases, each building upon the previous one. Due to the project phases being completed one at a time, the team worked very well together as everybody was on the same page from the beginning till the end. There were issues within the group with members being late and sick however there was clear communication and transparency which resulted in an understanding of members situation. The issue was resolved with other team members being able to handle additional tasks as they were outlined on Monday.com hence resulting in deadlines all being met.

Justification of decisions made in relation to group presentation

The group presentation decisions were influenced by the structured nature of the waterfall methodology. To emphasise the logical progression of each stage in the SDLC, the team decided it was best to give an overview of the stages in a linear fashion. Each team member contributed towards all the different stages of the project which meant everyone could showcase their expertise in different areas ranging from the planning, requirement gathering, system designing and testing.

Conclusion

Following SDLC process and next steps

The project adhered to the Software Development Life Cycle (SDLC) by progressing through planning, identifying problems and risks, requirements analysis, system design, and is now ready for the development phase. If further system development were pursued, then the next steps would be coding the pharmacy management system using relevant languages such as Python. By following SDLC principles, coding will commence with adherence to design specifications, ensuring that requirements are implemented accurately and efficiently.

Testing methods

Testing will include a combination of white box and black box methodologies. White box testing ensures internal logic integrity, while black box testing validates system functionality against requirements. “Black box testing is a software testing technique where the tester assesses the functionality of an application without in-depth knowledge of its internal design. In contrast, white box testing involves testers who have insight into the internal design of the application and analyse it during testing” (Ashtari, 2022). User acceptance testing will also be conducted to ensure user/stakeholder satisfaction. Following these testing methods, the team can aim to deliver a robust and reliable pharmacy management system that meets stakeholder needs and industry standards.

References

Senthilvel, G., Khan, O. M. A., & Qureshi, H. A. (2017), *"Enterprise Application Architecture with .NET Core"* [E-Book format]. Retrieved from https://learning.oreilly.com/library/view/enterprise-application-architecture/9781786468888/ (Accessed 17 January 2024).

IBM (2019). *"What is software development?"* [online] Ibm.com. Retrieved from https://www.ibm.com/topics/software-development (Accessed 27 January 2024).

Pilone, D. and Miles, R. (2007). *"Head first software development."* [E-Book format]. Retrieved from https://learning.oreilly.com/library/view/head-first-software/9780596527358/ (Accessed 28 January 2024).

Koch, A.S. (2005). *"Agile Software Development"* [E-Book format]. Artech House Computing Library. Retrieved from https://us.artechhouse.com/Agile-Software-Development-P1197.aspx (Accessed 29 January 2024).

UK Government (2023). *"Small to medium sized enterprise (SME) action plan."* [online] GOV.UK. Retrieved from https://www.gov.uk/government/publications/fcdo-small-to-medium-sized-enterprise-sme-action-plan/small-to-medium-sized-enterprise-sme-action-plan (Accessed 28 January 2024).

Team, A.C. (2022). *"Waterfall Methodology: A Complete Guide."* [Blog] Adobe Business. Retrieved from https://business.adobe.com/blog/basics/waterfall (Accessed 28 January 2024).

Dennis, A., Wixom, B. H., & Tegarden, D. (2012), *"Systems Analysis and Design with UML, 4th Edition"* [E-book format]. Retrieved from https://learning.oreilly.com/library/view/systems-analysis-and/9781118037423/ (Accessed 29 January 2024).

Pal, M. (2020), *"Implementing Microsoft Dynamics 365 Customer Engagement"* [E-book format]. Retrieved from https://learning.oreilly.com/library/view/implementing-microsoft-dynamics/9781838556877/ (Accessed 2 February 2024).

Dennis, A., Wixom, B. H., & Tegarden, D. (2012), *"Systems Analysis and Design with UML, 4th Edition"* [E-book format]. Retrieved from https://learning.oreilly.com/library/view/systems-analysis-and/9781118037423/ (Accessed 4 February 2024).

Balti, H., Galloway, A. (2021), *"Job Ready Java"* [E-Book format]. Retrieved from https://learning.oreilly.com/library/view/job-ready-java/9781119775645/ (Accessed 6 February 2024).

Lucidchart. (n.d.). *"What is an ER Diagram?"* [online]. Retrieved from https://www.lucidchart.com/pages/er-diagrams (Accessed 6 February 2024).

Miro. (n.d.). *"Data Flow Diagrams vs Flowcharts"* [online]. Retrieved from https://miro.com/diagramming/data-flow-diagram-vs-flowchart/ (Accessed 6 February 2024).

Gillenson, M.L. (2011). *"Fundamentals of database management systems"* [E-Book format]. Retrieved from https://www.oreilly.com/library/view/fundamentals-of-database/9781118213575/ (Accessed 18 February 2024).

PlanetTogether (2020). *"Advantages and Disadvantages of Flowcharts"* [online] www.planettogether.com. Retrieved from https://www.planettogether.com/blog/advantages-and-disadvantages-of-flowcharts (Accessed 11 Feb. 2024).

Jaiswal, S. (n.d.). *"Advantages and Disadvantages of ER Model".* javatpoint [online] www.javatpoint.com. Retrieved from https://www.javatpoint.com/advantages-and-disadvantages-of-er-model#:~:text=Limited%20Expressivity%3A%20The%20ER%20model (Accessed 11 Feb. 2024).

Quick, L. (2023), *"Acceptance Criteria for User Stories: Examples and Best Practices"*, knowledgehut blog, Retrieved from https://www.knowledgehut.com/blog/agile/what-are-acceptance-criteria (Accessed 12 February 2024).

Productfolio. (n.d.). *"Acceptance Criteria"* [online]. Retrieved from https://productfolio.com/what-are-acceptance-criteria/ (Accessed 12 February 2024).

Rehkopf, M. (n.d). *"User stories with examples and a template"*, Altassian blog, Retrieved from https://www.atlassian.com/agile/project-management/user-stories#:~:text=software%20user's%20perspective.-,A%20user%20story%20is%20an%20informal%2C%20general%20explanation%20of%20a,value%20back%20to%20the%20customer (Accessed 16 February 2024).

Altexsoft. (2023). *"Acceptance Criteria for User Stories in Agile: Purposes, Formats, Examples, and Best Practices",* Altexsoft blog, Retrieved from https://www.altexsoft.com/blog/acceptance-criteria-purposes-formats-and-best-practices/ (Accessed 13 February 2024).

Womack, J.P. and Jones, D.T. (2003). *"Lean thinking: Banish waste and create wealth in your corporation"*. New York Etc.: Free Press.

Ashtari, H. (2022). *"Black Box vs. White Box Testing: Understanding 3 Key Differences"*. Spiceworks article. Retrieved from https://www.spiceworks.com/tech/devops/articles/black-box-vs-white-box-testing/#:~:text=Black%20box%20testing%20is%20a,and%20analyzes%20it%20during%20testing (Accessed 19 February 2024).