CHAPTER 3

METHODOLOGY

3.1 Overview

This chapter describes the chosen methodology for the development of the proposed system. The development of the system involves several processes which will be carried out in several stages. Stages involved were used to achieve the objectives by reviewing the literature, conducting system analysis, designing the systems and performance testing. The tools being used in developing this simulation system will also be described within this chapter.

3.2 Research Workflow

The research workflow is the overview of a distinct set of activities, actions, tasks and work products of the overall proposed systems. Each stage consists of a number of processes in achieving the objectives of this proposed system. Figure 3.1 illustrates the research workflow for the whole development of this proposed system following some of the various operations during pre-processing from a study. Table 3.1 shows the research objectives with process, output and method.

Table 3. 1 Research Objectives

Problem Statement	**Objectives**	**Processes**	**Output**	**Method**
What are the technical challenges and solutions for developing a patient simulator for ophthalmology training that uses real-world patient data and generates realistic eye disability scenarios?	To identify and analyse the technical challenges and solutions for developing a patient simulator for ophthalmology training that uses real-world patient data and generates realistic eye disability scenarios.	Define research problems, objectives and the scopes. Study the type of games framework and its outcome towards education	Literature review and analysis of related works	To analyse existing simulation

How can algorithm techniques be applied to enhance the realism and effectiveness of a patient simulator for ophthalmology training that focuses on the diagnosis of eye disabilities?	To apply and evaluate algorithm techniques to enhance the realism and effectiveness of a patient simulator for ophthalmology training that focuses on the diagnosis of eye disabilities.	Design and develop the system and user interface. Implementation	Medical simulation learning games desktop console	Game development agile

How can user interface design and user experience evaluation methods be used to optimize the usability and accessibility of a patient simulator for ophthalmology training that incorporates different diagnostic tools and techniques?	To design and test user interface and user experience evaluation methods to optimize the usability and accessibility of a patient simulator for ophthalmology training that incorporates different diagnostic tools and techniques.	Test the result given by comparing it with the other games	Evaluation of the system	Experimental on user

3.2.1 Define Research Problems, Objectives and Scopes.

Before the development of the proposed system, problem identification of the existing method is carried out by interviewing the lecturer and checking the existing simulation game system. Based on the defined problem, the objectives of this research along with its scope are determined.

3.2.2 Study The Type Of Games Gameplay And Interface

The type of simulation games and interface are identified and studied. The related papers or journals are included in the literature review. The main information that is being searched is how a interface in desktop console work and the effectiveness of the games towards education. The games interface and gameplay has been develop based on existing simulator education features, strengths and weaknesses. The final outcome decided by the best approaches and algorithms to be implemented in developing the system.

3.2.3 Design and Develop the System And User Interface

In design and develop the user interface and its functionality, there are few tools have been used as shows in table 3.2.

Table 3. 2 Software Development Tools

Tools/ Software	Description
Unity	It is the world's first game-development device. Unity provides multi-platform applications designed to build 2D and 3D sports. Developers use this to create lots of easy and AAA-class games. Unity already has a large community with an engaging range of guides and tutorials.
Microsoft Visual Studio	An integrated development environment (IDE) from Microsoft. It is used to build computer systems, websites, desktop software, web utilities and smartphone applications. Visual Studio uses Microsoft applications creation platforms including Windows API, Microsoft Silverlight and Windows Presentation Foundation. The code can be both native and controlled.

3.3 Design

The design of the simulation game has undergone a thorough development process, consisting of six distinct phases. These phases have been crafted in accordance with the principles of the agile methodology, ensuring a well-structured system and minimizing the occurrence of any errors during the development stages.

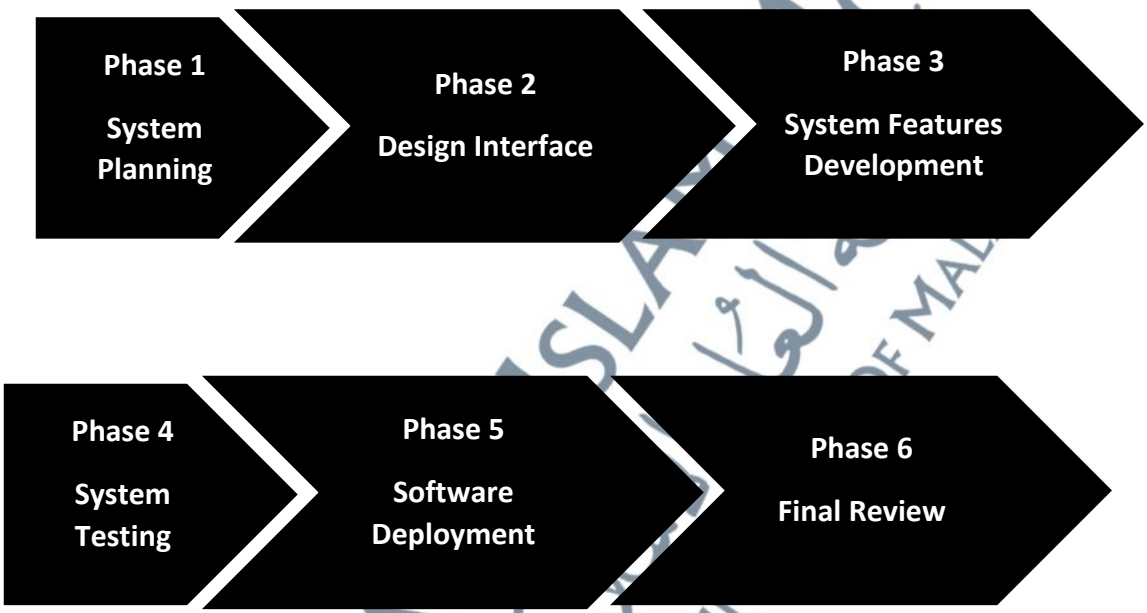

Figure 3. 1 Phase Diagram

3.3.1 System Planning

The planning phase for the patient simulator developed in Unity involves several key steps. First, the system setup phase focuses on installing the necessary program and software components, creating directories, extracting files, and configuring the operating system to align with the current version. It is important to ensure that the Unity version used is up to date to ensure compatibility with the target desktop version.

Next, the collection and prioritization of method and technical requirements for the patient simulator software are vital. These requirements form the basis of the game design documentation, providing a clear reference for future development. Tools like user stories, epics, themes, or features can be employed to describe the requirements in a concise and comprehensive manner.

During this phase, the release strategy is carefully planned, determining the frequency of delivering working software to users. Tools such as release plans or roadmaps are utilized to communicate the vision and expectations to the users effectively. The first sprint is then planned by selecting a subset of requirements from the product backlog specifically tailored to the patient simulator. This subset of requirements is chosen based on what can be feasibly completed within a fixed time frame, typically ranging from 1 to 3 weeks. Clear criteria, such as sprint goals or definitions of done, are established to define the successful completion of each sprint, ensuring the development progresses smoothly for the patient simulator in Unity.

3.3.2 Design Interface

The design phase within the context of agile software development is a collaborative and iterative approach that employs various methods and tools to develop a product focused on user satisfaction. This phase encompasses seven distinct stages, including understanding the problem, conducting research, sketching and designing. Throughout each stage, a range of techniques and tools are employed to define the problem statement, explore potential solutions, and assess the results. The design process continues in cycles until the final product aligns with the user's needs and fulfils their expectations.

3.3.2.1 Understand

The patient simulator product aims to address a specific problem, which involves providing medical students with a realistic and secure environment to enhance their skills. The needs and expectations of the users are carefully considered in the development process. In this thesis, user personas representing various types of medical students and instructors are created to gain a deeper understanding of the target audience. Additionally, user journeys are crafted to visually illustrate the users' interactions with the product, depicting the sequential steps, actions, emotions, and touchpoints they experience throughout their engagement with the simulator.

3.3.2.2 Sketch

A multitude of ideas and concepts are generated for the patient simulator product, exploring various potential scenarios and cases that can be simulated. The fundamental structure and progression of the product are also outlined, capturing the user's journey from beginning to end. This process involves creating storyboards, which consist of a series of drawings or visuals illustrating how users will interact with the product in specific contexts or scenarios. By visualizing these concepts through sketches and storyboards, creativity is stimulated, allowing for a more comprehensive understanding of the product's potential and fostering innovative thinking.

As an essential part of our product development process, we have also created a use case diagram to provide a clear overview of how the patient eye simulation diagnosis software will be utilized in different scenarios. This diagram showcases the various actors and their interactions with the system, highlighting the key functionalities and objectives of the software. The things worth to notable are:

- Patient: The primary user of the software, the patient, interacts with the system to simulate different eye conditions and undergo diagnosis.

- System: The system itself encompasses the software's core functionality, including simulating various eye conditions, providing diagnostic results, and maintaining patient records.

Use cases are depicted by the ellipses, connecting actors with specific interactions or functionalities. This diagram serves as a visual aid to demonstrate the various ways in which our software will be employed within the healthcare context, ensuring a comprehensive understanding of its utility and potential benefits.

By combining sketches, storyboards, and use case diagrams, we not only stimulate creativity but also create a robust foundation for developing our patient eye simulation diagnosis software, aligning it with the needs and expectations of both patients and medical professionals. This holistic approach to product design will undoubtedly contribute to the success of our innovative solution.

3.3.2.3 Design

Prototypes of the patient simulator product are developed, encompassing both low-fidelity and high-fidelity representations. Low-fidelity prototypes, such as wireframes, are constructed to depict the layout and navigation structure of the product's interface. Wireframes serve as schematic diagrams illustrating the arrangement and connections of various interface elements. In contrast, high-fidelity prototypes, known as mockups, are created to showcase the visual design and branding of the product. Mockups are static images that exemplify the intended appearance of the interface, including aspects such as colors, fonts, icons, and images. The design stage is pivotal in defining both the visual aesthetics and functional characteristics of the product, enabling a clearer understanding of its overall look and operation.

3.3.2.4 Prototype

Functional iterations of the patient simulator product are developed, allowing for real user testing. Tools like Figma or Adobe XD are utilized to construct interactive prototypes, which simulate the product's functionality across various scenarios and cases. These clickable prototypes provide dynamic models that enable users to interact with the product as if it were fully operational. Users can navigate through different screens, input text, click on buttons, and perform other actions representative of real-world usage. The prototype stage serves the purpose of showcasing the product's behaviour in response to user interactions, facilitating valuable feedback and insights for further refinement.

3.3.3 System Features Development

During this stage, the development of product features takes place in short time intervals known as sprints or iterations, typically lasting 1-3 weeks. These features are selected from the product backlog, which is a prioritized list of requirements and user stories. User stories provide concise descriptions of what users aim to accomplish or experience with the patient simulator product. For example, a user story could be: "As a medical student, I want to diagnose a patient with chest pain to practice my clinical reasoning skills."

To ensure manageable tasks within a sprint, user stories are further broken down into smaller components. These tasks are then assigned to different developers who work either independently or in pairs. Throughout the development process, a variety

of tools and techniques are employed for coding, testing, and integration. The Unity game engine serves as the primary tool for software development, complemented by the use of the C# programming language.

Effective communication and coordination among the developers are maintained through regular daily stand-up meetings. During these meetings, progress updates, challenges, and plans are shared to ensure a collaborative and synchronized development process. As an illustration, the creation of a login and sign-up system for the patient simulator product can be accomplished within a single sprint, aligned with the user story: "As a medical student, I want to create an account and log in to the product, so that I can access my profile and progress." Using tools like Unity and C#, developers can code, test, and integrate this feature. For storing and retrieving user data, Unity provides the persistent data path, which is a location on the user's device where data can be stored between sessions. XML can be used as the file format for storing user information, and Unity's built-in XML serialization and deserialization features can be utilized to read and write data to the XML file.

The implementation of the login and sign-up functionality would involve capturing user inputs, validating them, and managing the creation and authentication of user accounts. The persistent data path can be used to save the necessary user data, such as usernames and passwords, to the XML file. Unity's XML serialization and deserialization methods would handle the conversion between the XML file and the corresponding data structures in the application. By leveraging these Unity features, developers can create a robust login and sign-up system that allows users to register, log in, and access their profiles and progress.

In making of evaluation for eye disability diagnosis, the first aspect of the evaluation pertains to the technical proficiency of the students. A few data have been collected from the Universiti Sains Islam Malaysia's lecturer to make the evaluation for this simulation. In steps 1 through 9, students are required to move the pen to specific positions on the patient's side, corresponding to different points on an eye chart. These 9 steps include:

1. Students move the pen to the left of the patient's side.
2. Students move the pen to the top left of the patient's side.
3. Students move the pen to the bottom left position on the eye chart.
4. Students move the pen to the middle top of the patient's side.
5. Students move the pen to the middle bottom of the patient's side.
6. Students move the pen to the right of the patient's side.
7. Students move the pen to the top right of the patient's side.
8. Students move the pen to the bottom right of the patient's side.
9. Students lift up the patient's eyelid to examine the eye.

This task necessitates a high level of precision and dexterity, as even the slightest deviation from the prescribed positioning could impact the precision of the diagnosis. Proficiency in this aspect of the assessment reveals the student's capacity to execute delicate and precise maneuvers, a skill of paramount importance in real-world medical assessments.

Furthermore, the evaluation gauges the student's meticulousness. Each item on the checklist specifies an exact location to which the pen must be moved. Failure to adhere precisely to these instructions could lead to an erroneous diagnosis. This

underscores the utmost significance of attention to detail in the field of medicine, where even minor mistakes can result in grave consequences for patients. Students who exhibit scrupulousness in executing these steps manifest their dedication to patient well-being and the precision of their diagnostic procedures.

3.3.3.1 Algorithm for a Realistic Patient Eye Simulation

Simulations have become powerful tools for both education and professional training across various domains. In the field of medicine, simulations offer an invaluable means of honing diagnostic skills and familiarizing practitioners with complex procedures. One such application is a patient eye simulation, designed to provide an immersive experience in diagnosing eye conditions. In this context, we'll outline an algorithmic approach to create a realistic patient eye simulation.

This algorithm will encapsulate the essential elements required for an authentic simulation. It includes initializing simulation parameters, managing the simulation loop, updating the simulation's visual and diagnostic components, defining termination conditions, and concluding with a presentation of the diagnosis. While this algorithm offers a foundational framework, the development of a truly realistic simulation involves the integration of advanced medical knowledge, graphics rendering, and sophisticated diagnostic algorithms. Nonetheless, this algorithm serves as a starting point for conceptualizing and structuring such a simulation.

3.3.4 System Testing

System testing plays a crucial role in minimizing errors and ensuring software quality in the field of software engineering. User testing is typically conducted after the completion of each feature release. Towards the end of a sprint, the developed feature undergoes testing to validate its compliance with quality standards and user expectations. A variety of test types are employed, including functional tests, performance tests, security tests, and usability tests.

In this research, a total of 17 respondents, comprising lecturers, medical professionals, and medical students, were actively engaged. This diverse group of participants contributes significantly to enhancing the quality and reliability of the system. Ultimately, their insights and feedback serve to ensure the system's effectiveness in meeting the various user requirements and expectations.

Functional tests are conducted to verify if the login and sign-up system functions as intended and meets the acceptance criteria outlined in the user story. These tests verify if users can successfully register and sign in using their email and password. Performance tests assess the system's ability to handle high loads and respond promptly, ensuring it can manage concurrent requests without experiencing crashes or slowdowns. Security tests aim to ascertain if the login and sign-up system safeguards data and prevents unauthorized access. Encryption, validation, and verification techniques are examined to ensure user data is secure and protected against hacking or spoofing attacks.

Usability tests focus on assessing the ease of use and user-friendliness of the login and sign-up system. These tests evaluate whether the system's interface is clear

and intuitive, provides helpful feedback and error messages, and adheres to user expectations and conventions. The testing process involves different roles, such as developers, testers, and users. Various tools and techniques are employed, including test automation tools, test cases, test scenarios, and test reports.

Test automation tools are software applications that automate the execution and reporting of test results, eliminating the need for manual intervention. Test cases encompass specific sets of inputs and anticipated outputs used to verify the functionality of product features. Test scenarios depict realistic situations or stories that illustrate how users would interact with the product features in a given context. Test reports provide comprehensive summaries of test results, encompassing pass or fail statuses, identified defects or errors, and recommendations for improvement.

By employing these rigorous testing approaches and leveraging diverse tools and roles, the development team can enhance the quality and reliability of the login and sign-up system, ultimately ensuring its effectiveness in meeting user requirements and expectations.

3.3.5 Software Deployment

During this phase, the patient simulator software undergoes preparation and verification processes before it can be deployed. This includes comprehensive checks to ensure the quality, functionality, compatibility, and security of the software. Documentation, such as installation guides, user manuals, and release notes specific to the patient simulator software, is created or updated. Additionally, deployment activities are planned and scheduled, including selecting the deployment method, setting up the deployment environment, and defining the roles and responsibilities of the deployment team.

The patient simulator software is then installed and launched on the designated devices or servers where it will be utilized. This can be done manually through human intervention or automatically using tools or scripts that facilitate the deployment process. In this simulation, we utilize an automated system to ensure a user-friendly experience. Two deployment methods are employed: direct installation, which involves copying and executing the patient simulator software files on the target devices or servers, and package installation, which entails creating and distributing a package file containing the necessary software files and installation instructions.

To ensure the patient simulator software meets user requirements and expectations, it undergoes testing by users. User Acceptance Testing (UAT) is performed to validate the software's intended functionality and compliance with user requirements. UAT involves the utilization of real data and scenarios that simulate real-world usage of the patient simulator software. Various types of tests are conducted during UAT, including functional tests to verify that the software features operate as

intended and align with the acceptance criteria defined in user stories, performance tests to assess the software's ability to handle high workloads and respond promptly, usability tests to evaluate the user-friendliness and ease of use of the software features, and security tests to ascertain data protection and prevent unauthorized access. UAT involves the participation of different roles, such as users, stakeholders, and testers. Various tools and techniques, such as test cases, test scenarios, and test reports, are employed to facilitate the testing process. Test cases involve specific inputs and expected outputs used to verify the functionality of the software features, while test scenarios outline realistic situations or stories describing user interactions within specific contexts. Test reports summarize the test results, including pass or fail statuses, identified defects or errors, and feedback or suggestions for improvement.

By meticulously following these processes, the patient simulator software undergoes thorough testing and validation, ensuring it meets the needs and expectations of users, stakeholders, and testers before it is ready for deployment.

3.3.6 Final Review

The ownership of the patient simulator software is transitioned from the development team to the maintenance team to ensure its ongoing support and maintenance. During this transfer, all necessary product files, documentation, and data related to the patient simulator software are handed over to the maintenance team. In addition, comprehensive training and support are provided to the maintenance team to ensure they possess the necessary knowledge and skills to effectively utilize, operate, and maintain the software. This transfer is formalized through obtaining approval and sign-off from the users signifying their acceptance and ownership of the patient simulator software.

Once the deployment phase is completed, a thorough review and evaluation of the entire process take place. This involves documenting and finalizing all the activities that were undertaken during the deployment, including installation, testing, and training. The outcomes of the deployment are assessed, with a focus on user satisfaction, product quality, and overall deployment performance. Any issues or risks that arose during the deployment phase are identified and addressed, ensuring a smooth transition and optimal functioning of the patient simulator software.

Furthermore, this review process aims to capture valuable lessons learned and best practices from the deployment phase. These insights are documented and shared within the development and maintenance teams, facilitating knowledge sharing and enhancing future deployment efforts. By analyzing the deployment process and its outcomes, the team can identify areas of improvement and implement necessary refinements for future deployments. This comprehensive review not only ensures the

successful deployment of the patient simulator software but also paves the way for continuous enhancement and optimization of the overall simulation experience.

3.4 Summary

The agile methodology, with its user-centered and iterative approach, is particularly beneficial for the development of patient simulator software. This software aims to provide medical students with a realistic and immersive training environment, allowing them to practice essential skills in a safe and controlled setting. By adopting the agile principles and values, the design phase of this software follows a collaborative and iterative process that ensures the product meets the unique needs and expectations of medical students.

During the design phase, a deep understanding of the challenges faced by medical students is gained. Extensive research is conducted to explore existing patient simulator solutions, best practices, and emerging technologies that can enhance the training experience. This knowledge serves as a foundation for generating innovative ideas and concepts that will form the basis of the patient simulator software.

The design process involves the creation of low-fidelity and high-fidelity prototypes, which allow for the visualization and refinement of the software's features and functionalities. Through interactive versions of the software, medical students can provide valuable feedback and insights, ensuring that the product aligns with their training requirements. Usability tests are conducted to evaluate the software's ease of use, intuitiveness, and effectiveness in simulating real-world medical scenarios.

Incorporating agile principles, the design phase is not a linear process but rather a cyclical and adaptive one. Feedback from medical students, instructors, and stakeholders plays a vital role in shaping the product's design and functionality. Continuous improvement is emphasized, allowing for enhancements and adjustments based on user experiences and evolving training needs.

By embracing the agile methodology in the design phase of patient simulator software, the development team can create a user-centric product that enhances the learning journey of medical students. The iterative nature of agile development enables the team to respond to changing requirements, incorporate emerging technologies, and deliver a high-quality software solution that meets the demands of modern medical education.