# SECD/SCSD 2613: System Analysis And Design

UTMSpaceKL

Semester 02, 2024/2025

---

## Project Phase 1

## Legacy Healthcare Management System

## KPJ Healthcare Berhad

**Team Name:** Group 5

**Team Members:**


1. SITI NOOR HIDAYAH BINTI MAD AKHIR (SX230266ECJHS04)
2. NOR LAILY BINTI ZAKARIA (SX230265ECJHS04)
3. LEE RU XIU (SX230523ECJHS04)
4. SIONG WEI LOONG (SX230277ECJHS04)


**Project Phase 1 GitHub Link:**

https://github.com/Siong010/Group5_Project1_SAD_20242025

# Table of Contents

# 1.0 Introduction

This proposal presents a strategic plan to modernize KPJ Healthcare's Legacy Healthcare Management System (HMS), which has been the backbone of its administrative and financial operations for over two decades. As the demands of healthcare delivery evolve in an increasingly digital and interconnected environment, the need for a scalable, secure, and user-friendly management system has become critical.

The existing HMS, while foundational to operations such as billing, inventory management, accounting, and financial reporting, no longer meets modern performance, usability, and security expectations. The system's outdated technologies create operational bottlenecks, limit integration with modern tools, and expose the organization to cybersecurity vulnerabilities.

This modernization initiative aims to transition the Legacy HMS to a robust, modular platform built with current technologies. By doing so, KPJ will not only preserve its core business workflows but also enable enhanced system performance, improved accessibility across devices, better integration capabilities, and long-term sustainability. The following sections provide a detailed overview of the current system, identify key issues, and outline the rationale for change.

# 2.0 Background Study

KPJ's current Healthcare Management System was originally developed in the late 1990s and has undergone minimal updates over the years. The system was designed to support a wide range of hospital operations, including patient billing, procurement and materials tracking, financial accounting, and general ledger management. Despite its historical reliability, the system was built on a monolithic architecture using proprietary and now largely obsolete development tools.

The technology stack consists of an aging backend system that lacks modularity, a rigid user interface optimized for outdated desktop browsers, and Legacy tools for reporting and access control. Over time, some components have been patched or partially upgraded, but these improvements have not been sufficient to keep pace with modern technology standards or user expectations.

Key functions are still dependent on desktop-based access, static reporting tools, and manual data workflows, which hinder operational agility. The system also lacks the flexibility to integrate with third-party services such as insurance platforms, patient mobile apps, or cloud-based analytics tools—features that are increasingly standard in modern healthcare environments.

## 3.0 Problem Statement

KPJ's current HMS is no longer capable of supporting the growing operational, security, and digital transformation needs of the organization. The reliance on outdated technologies has led to a range of technical, functional, and strategic challenges that must be addressed.

**Key Problems Identified:**

1. Obsolete Technology Stack

   The backend was developed in Progress 10C, and the Silverlight-based frontend are no longer supported by modern development tools or browsers. Silverlight has been deprecated and is incompatible with most current web environments.

2. Limited Browser Compatibility

   The user interface was designed for Internet Explorer, which is now phased out. As a result, users face significant issues when attempting to access the system on modern browsers like Chrome, Firefox, Edge, or Safari.

3. Security and Compliance Risks

   The lack of modern authentication, encryption, and access control mechanisms exposes the system to potential data breaches and non-compliance with industry standards for data privacy and security.

4. High Maintenance Burden

   Legacy code, hardcoded logic, and scarce developer resources for outdated platforms make it difficult to apply updates, fix bugs, or scale the system as operational needs evolve.

5. Poor User Experience (UX)

   The current UI design does not meet modern usability standards. It is non-responsive, form-heavy, and lacks intuitive navigation, particularly for mobile users and new staff.

6. Integration Barriers

   The system's monolithic structure and lack of API support hinder integration with newer platforms such as digital payment systems, mobile health apps, or insurance claim portals.

**Organizational Impact**

Without significant intervention, KPJ faces increasing operational costs, slower service delivery, and potential non-compliance with healthcare data standards. These challenges threaten both short-term efficiency and long-term competitiveness in a healthcare landscape that demands agility, security, and digital readiness.

## 4.0 Proposed Solutions

To address the limitations of the current Legacy Healthcare Management System, we propose a complete modernization using a contemporary and scalable technology stack. The new system will be built with a modern frontend framework, such as React or Angular, which will provide a responsive and intuitive user interface accessible across all major web browsers and devices. This will significantly enhance the user experience and ensure compatibility with modern web standards.

On the backend, we recommend using ASP.NET Core Web API, a robust and high-performance framework that supports modular development and seamless integration with other systems. For data storage, the system will utilize SQL Server or PostgreSQL, both of which are reliable, secure, and capable of handling large volumes of healthcare data.

User authentication and access control will be managed using IdentityServer, OAuth2, or Azure Active Directory (Azure AD). These identity management solutions ensure secure and role-based access to sensitive patient and financial information, complying with industry standards and privacy regulations.

For reporting and analytics, the new system will support tools such as Power BI, SQL Server Reporting Services (SSRS), Telerik, or DevExpress. These tools offer flexible, interactive, and visually rich reporting capabilities, enabling hospital management to make data-driven decisions efficiently.

Lastly, the solution will be deployed either on Microsoft Azure Cloud for scalability and resilience, or through on-premises Docker containers for organizations that require local hosting. This hybrid approach provides flexibility to align with the organization's infrastructure preferences and compliance requirements.

In summary, the proposed solution is designed to be secure, scalable, user-friendly, and future-ready, ensuring the healthcare organization is well-equipped to meet evolving operational and regulatory demands.

## Feasibility Study

### 1. Technical Feasibility

The modernization of the KPJ Healthcare Management System (HMS) is technically feasible and well-supported by today's software development ecosystems. The proposed technology stack includes mature frameworks and libraries such as ASP.NET Core for backend development and React or Angular for the frontend. These technologies are widely adopted, actively maintained, and backed by strong developer communities, ensuring long-term stability and support.

The system will be built using RESTful APIs, which allow different components of the application to communicate in a standardized and modular way. This modular architecture supports scalability and simplifies integration with external systems such as national health registries, insurance platforms, or mobile applications.

To enhance deployment efficiency and reduce human error, Continuous Integration and Continuous Deployment (CI/CD) pipelines will be established. These automated pipelines allow code changes to be tested and deployed quickly and reliably, enabling faster feature delivery and bug fixes.

### 2. Operational Feasibility

From an operational perspective, the modernized HMS offers clear improvements in usability and manageability. One of the main advantages is enhanced user experience (UX), especially with support for all modern web browsers. Unlike the Legacy system which relied on outdated platforms like Internet Explorer and Silverlight, the new system will be fully compatible with Chrome, Firefox, Edge, Safari, and others—ensuring consistent accessibility across devices and platforms.

The user interface will be designed to be intuitive and user-friendly, significantly reducing the learning curve for new staff. This ease of use will facilitate faster onboarding and minimize training costs, allowing hospital personnel to adapt quickly and efficiently to the new system.

Additionally, the system will include Role-Based Access Control (RBAC), ensuring that users only have access to the information and functions relevant to their role. This not only enhances operational efficiency by streamlining workflows but also strengthens security and data governance, which is critical in a healthcare environment where patient data must be protected.

## 3. Economic Feasibility

The economic feasibility of modernizing the KPJ Healthcare Management System (HMS) is strongly supported by both cost savings and long-term benefits. The total estimated cost of the project over a five-year period is RM726,000, which includes investments in new hardware, modern software licenses, external consultants for development and migration, and staff training.

Despite the upfront investment, the project is expected to generate substantial annual savings and productivity gains, amounting to approximately and average of RM787,897 per year. These gains come from improved operational efficiency, reduced manual processing, faster reporting, and fewer system downtimes. The weekly cost savings alone are estimated at nearly RM20,000, which translates to over RM1 million annually just from streamlined workflows and reduced errors.

When analysed using standard cost-benefit techniques, the project yields a Net Present Value (NPV) of approximately RM1.05 million, and it's 5 years project duration, the profitability index showing 1.45 which the index is more than 1, indicating a highly favourable return. The payback period is estimated to be around 2 years, after which the system continues to deliver financial and operational benefits without additional major investment. Moreover, a 7% discount rate was used to reflect the time value of money, and the analysis included sensitivity testing with a ±10% variation in costs and a ±15% variation in projected benefits to ensure robustness.

In conclusion, the modernization initiative is economically viable and sustainable. The long-term cost savings, increased service efficiency, and better user experience justify the investment and make it a sound financial decision for the organization.

Below is the detailed calculation for the Cost-Benefit Analysis (CBA) for KPJ to modernize their HMS:

# Cost-Benefit Analysis (CBA)

| Estimated Cost | |
|---|---|
| Hardware | RM300,000 |
| Software | RM200,000 |
| Consultant | RM100,000 |
| Training | RM80,000 |
| Supplies | RM20,000/Year |
| IS Support | RM300,000/Year |
| Maintenance | RM100,000/Year |

| Assumptions | |
|---|---|
| Project duration | 5 Years |
| Discount rate | 7% |
| Sensitivity factor(cost) | 1.1 |
| Sensitivity factor(benefits) | 0.85 |
| Annual change in benefits | 5% |
| Annual change in production costs | 7% |

| Estimated benefits | |
|---|---|
| Weekly cost savings | RM19,800/Week |

| Costs | Year 0 RM | Year 1 RM | Year 2 RM | Year 3 RM | Year 4 RM | Year 5 RM |
|---|---|---|---|---|---|---|
| Development Costs | | | | | | |
| Hardware | 330,000 | | | | | |
| Software | 220,000 | | | | | |
| Consultant | 88,000 | | | | | |
| Training | 88,000 | | | | | |
| **Total** | 726,000 | | | | | |
| Production Costs | | | | | | |
| Supplies | | 22,000 | 23,540 | 25,188 | 26,951 | 28,838 |
| IS Support | | 330,000 | 353,100 | 377,817 | 404,264 | 432,563 |
| Maintenance | | 110,000 | 117,700 | 125,939 | 134,755 | 144,188 |
| Annual Prod. Costs | | 462,000 | 494,340 | 528,944 | 565,970 | 605,588 |
| (Present Value) | | 431,776 | 431,776 | 431,776 | 431,776 | 431,776 |
| **Accumulated Costs** | | 1,157,776 | 1,589,551 | 2,021,327 | 2,453,103 | 2,884,879 |

| Benefits | Year 0 RM | Year 1 RM | Year 2 RM | Year 3 RM | Year 4 RM | Year 5 RM |
|---|---|---|---|---|---|---|
| Reduced weekly costs | | 875,160 | 918,918 | 964,864 | 1,013,107 | 1,063,762 |
| (Present Value) | | 817,907 | 802,619 | 787,616 | 772,895 | 758,448 |
| Accumulated Benefits | | 817,907 | 1,620,525 | 2,408,141 | 3,181,036 | 3,939,484 |
| (Present Value) | | | | | | |
| **Gain or Loss** | | (339,869) | 30,974 | 386,814 | 727,933 | 1,054,605 |
| **Profitability Index** | 1.45 | | | | | |

## 5.0 Objectives

The primary objective of this project is to modernize an existing Legacy Healthcare Management System that currently suffers from poor user experience, limited accessibility, outdated technology, and maintenance difficulties. The system is widely used for administrative processes such as patient registration, billing, inventory/materials management, and accounting, but has become inefficient due to its structure, security risks, and lack of scalability.

This modernization effort is not a redesign of business logic but rather a technical transformation of the interface, backend services, and overall user experience. The project focuses on ensuring the system remains functionally familiar to users while significantly improving accessibility, maintainability, and performance across modules.

The specific objectives of the project include:

1. **Modernize the User Interface (UI/UX):**

    - Replace outdated, form-heavy desktop interfaces with a clean, responsive web-based interface that is intuitive and mobile-friendly.

2. **Enable Cross-Platform Accessibility:**

    - Develop the system to be compatible with modern web browsers, allowing users to access the application from both desktop and mobile devices without needing specialized hardware or software.

3. **Decouple Monolithic Backend into API Services:**

    - Rebuild the backend using modular API-driven architecture (e.g. RESTful services), allowing for more scalable and maintainable code.

4. **Strengthen System Security:**

    - Implement updated authentication and authorization mechanisms with proper role-based access control, reducing vulnerability to unauthorized access and data breaches.

5. **Improve Maintainability and Scalability:**

    - Use modern programming frameworks and best practices to reduce technical debt, simplify updates, and support future feature expansion.

6. **Retain Core Business Logic and Functionality:**

    - Ensure continuity by preserving the system's core workflows and logic, allowing minimal disruption to users during and after the migration.

7. **Enhance System Reporting and Data Presentation:**

    - Improve the quality, speed, and visual clarity of the reporting module, allowing for better data-driven decision-making.

8. **Ensure Seamless Data Migration and Minimal Downtime:**

    - Prepare strategies for migrating data from the Legacy system to the new structure while minimizing disruption to ongoing operations.

## 6.0 Scope of the Project

This project focuses on the technical modernization of a Legacy Healthcare Management System that was previously used for administrative and financial operations in a medical environment. The modernization effort targets the system's interface, backend architecture, and user accessibility, while preserving the core business logic and workflows that supported essential modules such as billing, materials management, and financial reporting.

The scope is carefully defined to enhance maintainability, scalability, and usability, while postponing any significant changes to business workflows or integration with third-party systems to future phases. This approach aims to ensure a smooth transition to newer technologies with minimal user retraining or operational disruption.

**In-Scope Features:**

1. **User Interface (UI/UX) Modernization**
   - Redesign the system interface with a responsive web-based layout.
   - Implement standard design patterns for better user experience.
   - Ensure compatibility with desktop and mobile browsers.

2. **Modular API-Based Backend Development**
   - Convert existing backend logic into RESTful APIs.
   - Separate modules for billing, materials management, accounts, and reporting.
   - Use secure and maintainable backend frameworks.

3. **Authentication and Authorization**
   - Implement role-based access control (RBAC) and session management.
   - Improve login workflows with enhanced security protocols.

4. **Core Functional Modules (Front-End & Back-End Upgrade Only)**
   - **Billing & Payment Module:** Maintain functionality while improving user flow and backend stability.
   - **Materials/Inventory Management:** Redesign front-end for usability and rewire backend services for better performance.
   - **Accounts & General Ledger:** Present complex accounting data through simplified, intuitive interfaces.
   - **Reports Module:** Enhance existing reports visually, introduce filters, and export functions (e.g. PDF, Excel).

5. **Training & Documentation**
   - Provide internal documentation for system administrators and end-users.
   - Conduct training sessions post-deployment to ensure smooth transition.

6. **Data Migration Strategy**
   - Design data extraction and migration scripts from the Legacy database.
   - Validate migrated data to ensure accuracy and completeness.

**Out-of-Scope Features:**

To ensure the project remains focused and manageable within time and resource constraints, the following elements are explicitly excluded from the current development phase:

1. **Changes to Business Logic or Workflow**
   - The way forms and processes work (e.g., billing flow, report calculations) will not be redefined or reengineered.

2. **Integration with External Third-Party Systems**
   - APIs to insurance portals, payment gateways, or medical platforms will be handled in future phases.

3. **Mobile App Development**
   - The system will be mobile-browser friendly, but no native mobile app will be created in this phase.

4. **Advanced Analytics or AI Features**
   - Dashboards will be improved, but no predictive analytics or machine learning will be included.

5. **Redesign of Data Schema**
   - The database structure will only be refactored as needed to support API endpoints; a full schema redesign is not planned.

# 7.0 Project Planning

The modernization of the Legacy Healthcare Management System will follow a modular, structured development plan to ensure smooth transition, minimal disruption, and maintainable outputs. This section outlines the project's key roles and responsibilities, as well as a high-level Work Breakdown Structure (WBS) that defines the tasks to be completed in each phase.

A collaborative team with complementary technical and managerial skills will be assigned to execute this project. The system will be developed in iterative stages, with testing and feedback integrated at each milestone.

## 7.1 Human Resource

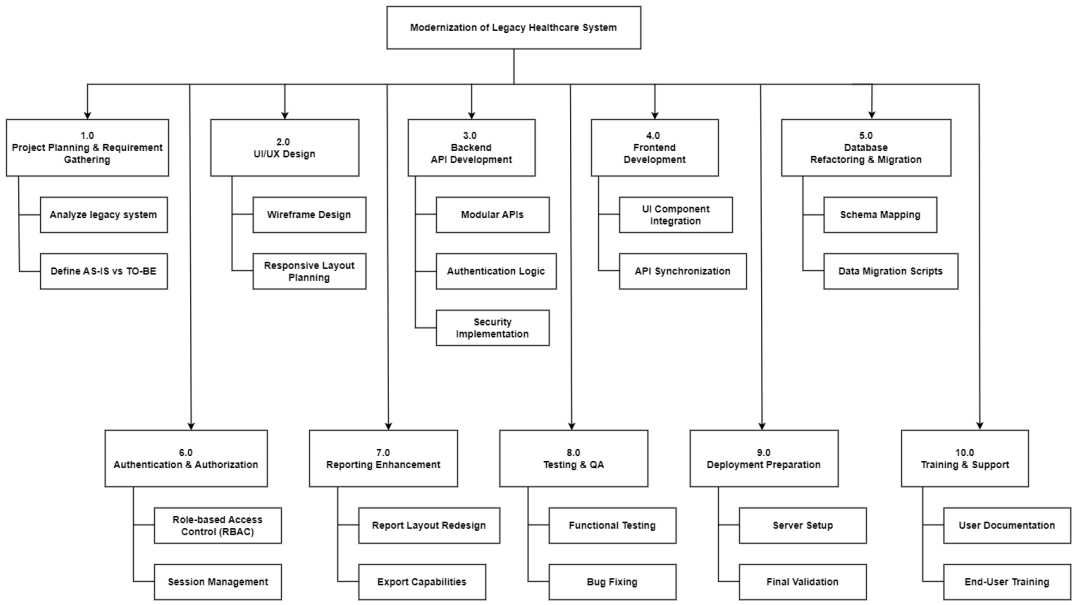
The project team will consist of both technical and functional roles.

| Role | No. of Personnel | Responsibilities |
|---|---|---|
| Project Manager (PM) | 1 | Oversees the project, coordinates tasks, ensures timelines and manages stakeholders. Also responsible for system analysis tasks including requirement documentation, AS-IS/TO-BE analysis, and process clarification. |
| UI/UX Designer | 1 | Designs wireframes, ensures usability and accessibility across platforms. |
| Frontend Developer | 1 | Implements the responsive interface and integrates with backend APIs. |
| Backend Developers | 2 | Develops modular APIs, implements authentication, and business logic mapping. |
| Database Engineer | 1 | Manages data migration, validates schema, and ensures data integrity. |
| QA Tester | 1 | Conducts test cases, verifies functionality, and reports bugs. |
| Training & Support | 1 | Creates documentation and supports users during transition. |

## 7.2 Work Breakdown Structure (WBS)

The Work Breakdown Structure (WBS) outlines all project activities grouped into logical work packages. This structure ensures that all project tasks are tracked and completed in sequence or in parallel as applicable.

| ID | Task / Phase | Description |
|---|---|---|
| 1.0 | Project Planning & Requirement Gathering | Analyze Legacy system, interview users, define AS-IS vs TO-BE. |
| 2.0 | UI/UX Design | Design wireframes, ensure accessibility and consistency. |
| 3.0 | Backend API Development | Build scalable APIs and map Legacy logic. |
| 4.0 | Frontend Development | Develop responsive UI and integrate with APIs. |
| 5.0 | Database Refactoring & Migration | Design migration strategy, validate data. |
| 6.0 | Authentication & Authorization | Implement secure role-based access control. |
| 7.0 | Reporting Enhancement | Improve visuals, filters, and export capabilities. |
| 8.0 | Testing & QA | Perform test cases and bug fixing cycles. |
| 9.0 | Deployment Preparation | Setup server environment and final testing. |
| 10.0 | Training & Support | Train users, distribute manuals, and provide support. |



### 7.3 PERT Chart (Based on WBS)

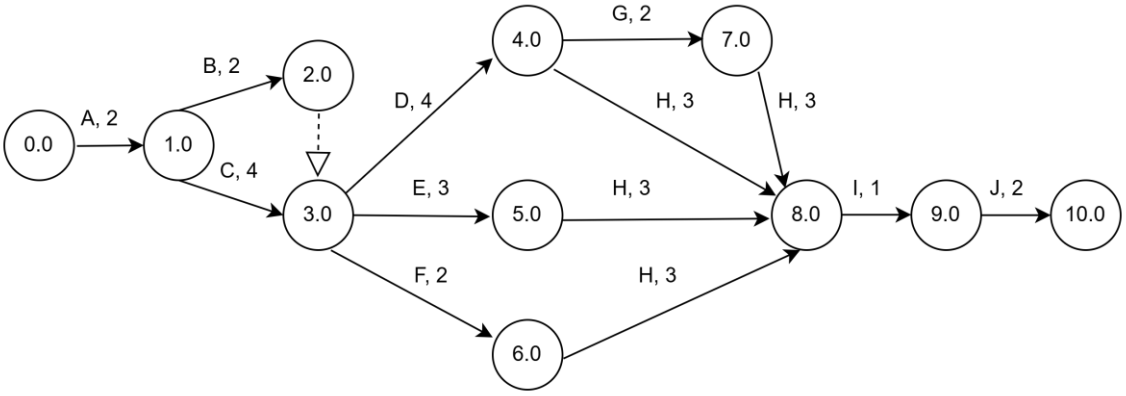
Key Elements of the PERT Chart:

**Task Dependencies**: Each activity in the Work Breakdown Structure (WBS) is represented with arrows indicating predecessor-successor relationships.

**Estimated Time for Completion**: Tasks are assigned estimated durations based on their complexity.

**Critical Path Identification**: Identifies the sequence of tasks that directly impact the total project duration.

| ID | Activity | Task | Duration (weeks) | Predecessor |
|---|---|---|---|---|
| 1.0 | A | Project Planning & Requirement Gathering | 2 | - |
| 2.0 | B | UI/UX Design | 2 | 1.0 |
| 3.0 | C | Backend API Development | 4 | 1.0 |
| 4.0 | D | Frontend Development | 4 | 2.0, 3.0 |
| 5.0 | E | Database Refactoring & Migration | 3 | 3.0 |
| 6.0 | F | Authentication & Authorization | 2 | 3.0 |
| 7.0 | G | Reporting Enhancement | 2 | 4.0 |
| 8.0 | H | Testing & QA | 3 | 4.0, 5.0, 6.0, 7.0 |
| 9.0 | I | Deployment Preparation | 1 | 8.0 |
| 10.0 | J | Training & Support | 2 | 9.0 |




Note: All durations are in weeks.

Path 1: A→ C→ E→ H→ I→ J Length= 2 + 4 + 3 + 3 + 1 + 2 = 15 Weeks

Path 2: A → B→ E → H→ I→ J Length= 2 + 2 + 3 + 3 + 1 + 2 = 13 Weeks

Path 3: A→ C→ F→ H→ I→ J Length = 2 + 4 + 2 + 3 + 1 + 2 = 14 Weeks

Path 4: A→ C→ D→ G→ H→ I→ J Length = 2 + 4 + 4 + 2 + 3 + 1 + 2 = 18 Weeks

Path 5: A → C → D → E → H → I → J Length = 2 + 4 + 4 + 3 + 3 + 1 + 2 = 19 Weeks

Since the critical path is the longest path through the network diagram, Path 5 are the critical path for this Project.

### 7.4 Gantt Chart

Below is a simplified Gantt chart outlining the timeline based on the durations from the PERT analysis:

| Task ID | Task Name | Start Week | End Week | WK01 | WK02 | WK03 | WK04 | WK05 | WK06 | WK07 | WK08 | WK09 | WK10 | WK11 | WK12 | WK13 | WK14 | WK15 | WK16 | WK17 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1.0 | Project Planning & Requirement Gathering | WK01 | WK03 | 1 | 1 | 1 | | | | | | | | | | | | | | |
| 2.0 | UI/UX Design | WK03 | WK05 | | | 1 | 1 | 1 | | | | | | | | | | | | |
| 3.0 | Backend API Development | WK03 | WK07 | | | 1 | 1 | 1 | 1 | 1 | | | | | | | | | | |
| 4.0 | Frontend Development | WK05 | WK09 | | | | | 1 | 1 | 1 | 1 | 1 | | | | | | | | |
| 5.0 | Database Refactoring & Migration | WK07 | WK10 | | | | | | | 1 | 1 | 1 | 1 | | | | | | | |
| 6.0 | Authentication & Authorization | WK07 | WK09 | | | | | | | 1 | 1 | 1 | | | | | | | | |
| 7.0 | Reporting Enhancement | WK09 | WK11 | | | | | | | | | 1 | 1 | 1 | | | | | | |
| 8.0 | Testing & QA | WK11 | WK14 | | | | | | | | | | | 1 | 1 | 1 | 1 | | | |
| 9.0 | Deployment Preparation | WK14 | WK15 | | | | | | | | | | | | | | 1 | 1 | | |
| 10.0 | Training & Support | WK15 | WK17 | | | | | | | | | | | | | | | 1 | 1 | 1 |

## 8.0 Benefit and Overall Summary of Proposed System

The proposed modernization of the Legacy Healthcare Management System provides substantial benefits to the organization, particularly in the areas of performance, security, and usability. By refactoring the system architecture into a modular API-based framework and introducing a modern web interface, the project ensures both immediate and long-term value.

**Key Benefits:**

1. **Improved Accessibility:**
    - Users can access the system through any modern browser across devices (desktop, tablet, smartphone).
    - No dependency on platform-specific installations.
2. **Enhanced User Experience:**
    - Intuitive, responsive, and user-friendly design increases staff efficiency.
    - Simplified interfaces reduce training needs and human error.
3. **Scalability and Maintainability:**
    - Modular backend architecture enables easier updates, debugging, and future expansion.
    - Separation of concerns allows independent scaling of system modules.
4. **Increased System Security:**
    - Implementation of role-based access control (RBAC) ensures data integrity and limits unauthorized access.
    - Updated security protocols reduce risk of breaches and compliance issues.
5. **Better Reporting and Decision Support:**
    - Enhanced reporting modules with export capabilities improve visibility and operational decision-making.
    - Filters and visual improvements allow for quicker data analysis.
6. **Minimized Disruption during Migration:**
    - Legacy business logic is retained to avoid retraining disruptions.
    - Comprehensive data migration planning ensures continuity of operations.

7. **Long-Term Cost Savings:**

- Reduced maintenance costs due to modernized codebase.
- Less reliance on Legacy developers and specialized hardware.

**Summary**

This project addresses critical pain points in the existing KPJ healthcare system infrastructure while preserving operational continuity. The strategic modernization plan ensures a smooth transition with minimal downtime, all while delivering a highly maintainable, secure, and user-centric platform. The system is well-positioned for future enhancements and integration in subsequent phases, including mobile app development and advanced analytics, further amplifying the return on investment.

## Appendix A: Disclaimer

This academic report is prepared solely for educational purposes as part of a university assignment. The content, analysis, and system design presented herein are based on a Legacy system observed from a real-world environment within a healthcare context, including references to operational flows inspired by KPJ Healthcare Berhad.

All information used in this document is either publicly accessible or generalized for academic use. This report is not intended for commercial use, publication, or internal distribution within KPJ Healthcare or its subsidiaries.

The authors do not claim ownership of any proprietary system, software, or data related to KPJ Healthcare. All rights and intellectual property belong to their respective owners.

## Appendix B: References


1. Sharina, N. I. (2020). 02 Topic II: Project Planning Process [PDF slides]. UTM, SECD2613 Systems Analysis and Design.
2. Kendall, K. E., & Kendall, J. E. (2019). Systems Analysis and Design (10th ed.). Pearson Education.

# Appendix C: GitHub Snapshot