**Table of contents: Patunak**

**c. Section numbering. – patune pitu anka**

**d. Pagination. – page number**

**e. Citations and bibliography section.**

|  |  |
| --- | --- |
| **[01]. Introduction** |  |
| **[02]. Problem Identification** |  |
| **[03]. Objectives** |  |
| **[04]. Functional and Non-functional requirements** |  |
| **[05]. Diagram explanation** |  |
| **[06]. Design of each form in the application** |  |
| **[07]. Implementation** |  |
| **[08]. Details about the tools used in the design and development of the application** |  |
| **[09]. Initial plan vs. Actual plan** |  |
| **[10]. Work contribution** |  |
| **[11]. Future enhancements** |  |
| **[12]. Details of group members** |  |

**Introduction:**

A project management system is a digital tool designed to help individuals and teams efficiently plan, execute, and track the progress of projects. It provides a centralized platform where users can define project goals, allocate tasks, set deadlines, assign responsibilities, and monitor the overall project timeline. This system facilitates collaboration by allowing team members to communicate, share files, and update their progress in real time. Additionally, it offers features such as milestone tracking, resource management, and reporting to ensure projects are completed on time and within scope.

In this introduction, we provide an overview of such a system, its key components, and how the .NET Framework can be leveraged to build a visual project management tool.

.NET Framework and C# language have been used to create a project management system for building and construction. A user-friendly visual interface is used to make the project management system efficient. We use Visual Studio software for that.

The project consists of three main interfaces, each tailored to the unique needs of customers, employees, and management. These interfaces act as portals to the system, giving users access to specific functionalities that align with their roles and responsibilities. In addition, nine additional models are accessible through these primary interfaces, providing additional functionality and enhanced data management capabilities as needed.

**User Roles and Interfaces :** Add karanna one

**Problem identification.**

Complexity and Scale: Construction projects involve numerous stakeholders, intricate processes, and extensive data. Managing all these aspects manually can lead to errors, delays, and cost overruns.

Communication Challenges: Effective communication between contractors, subcontractors, architects, engineers, and clients is critical. Inefficient communication can lead to misunderstandings and project delays.

Resource Allocation: Efficiently allocating and managing resources, including labor, materials, and equipment, is vital. Poor resource management can lead to inefficiencies and increased costs.

Schedule Management: Construction projects are time-sensitive, and delays can be costly. Ensuring that tasks are completed on time and managing project schedules effectively is a critical challenge.

Budget Control: Staying within budget is essential. Overspending can harm profitability and project viability.

Quality Assurance: Ensuring that construction meets quality standards and regulatory requirements is crucial to avoid rework and legal issues.

**Objectives:**

Efficient Project Planning:

Enable project managers to efficiently plan construction projects, including defining tasks, allocating resources, setting timelines, and establishing dependencies.

Resource Optimization:

Develop optimization algorithms to make efficient use of resources, minimizing waste and costs.

Real-Time Communication:

Enhance communication between project stakeholders, ensuring that contractors, subcontractors, architects, and clients can share information and updates in real time.

Cost Estimation:

Build applications for project managers to track progress, allocate resources, and manage budgets effectively.

Document and Data Management:

Streamline document management, including contracts, blueprints, permits, and other project-related files.

Data Analytics and Reporting:

Provide insights into project performance through data analytics and reporting tools.

Security and Data Protection

Guarantee the security and confidentiality of sensitive project data and documents.

Scalability and Flexibility:

Design the system to be scalable, accommodating projects of various sizes and complexities.

User-friendly interface design

• Create intuitive and user-friendly interfaces for all modules to ensure ease of use for customers, employees, and management.

• Prioritize clear navigation, consistent layouts, and easy access to relevant information on each interface.

• Conduct user testing to gather feedback and improve the interface.

Authentication and Access Control

Implement secure login and authentication mechanisms for customers, employees, and management to ensure that only authorized users can access the system.

Define and enforce access controls based on user roles to restrict access to specific interfaces and functionalities.

Include password encryption and protection against common security vulnerabilities.

**Functional and Non-functional requirements**

**Functional Requirements:**

Project Planning and Scheduling:

Allow users to create and manage project plans.

Support task allocation, dependencies, and timelines.

Enable the creation of Gantt charts for visual project scheduling.

Resource Management:

Provide tools for resource allocation, including labor, equipment, and materials.

Allow users to assign resources to specific tasks.

Generate resource utilization reports.

Communication and Collaboration:

Offer real-time messaging and discussion forums for project teams.

Facilitate document sharing and collaboration on blueprints, plans, and project-related files.

Provide comment and annotation features for document collaboration.

Budget Control and Cost Tracking:

Allow users to set project budgets and track expenses.

Generate cost reports and alerts for budget overruns.

Enable users to manage change orders and their impact on costs.

Quality Assurance and Compliance:

Support the creation of quality control checklists.

Enable inspection workflows and issue tracking.

Provide audit trails and documentation for compliance purposes.

Risk Management:

Allow users to identify, assess, and manage project risks.

Provide risk mitigation strategies and risk response planning tools.

Generate risk assessment reports and risk dashboards.

Document and Data Management

Create a centralized repository for project-related documents.

Implement version control for documents.

Allow users to set access permissions and manage document revisions.

Reporting and Analytics:

Generate customizable project performance reports.

Provide data visualization tools, including charts and graphs.

Offer dashboards for real-time project monitoring.

**Non-functional Requirements**

Non-functional requirements (NFRs), also known as quality attributes or system qualities, specify how a system should perform rather than what it should do. These requirements define the characteristics and constraints that shape the overall behavior of a software system. NFRs are essential for ensuring that the system meets user expectations and complies with organizational and industry standards. Here's a breakdown of the key aspects of non-functional requirements.

1. Security and Privacy:

- Data Security: Ensure strong encryption mechanisms to protect sensitive customer and financial data in transit and at rest.

- Authentication Security: Implement strict security measures for facial recognition to prevent unauthorized access.

- Compliance: Adhere to data protection regulations and industry standards to protect customer privacy.

2. Performance:

- Response time: Ensure that the system responds quickly to user interactions, including facial recognition, data retrieval, and transaction processing.

3. Reliability:

- Availability: ensuring high system availability, minimizing downtime and disruption to management operations.

- Fault Tolerance: Implement mechanisms to properly recover from system failures or errors.

4. Usage:

- User-friendly interfaces: Design intuitive and user-friendly interfaces for all types of users, including clear navigation and helpful error messages.

- Accessibility: Ensure that the system is accessible to users with disabilities, in accordance with applicable accessibility standards.

5. Maintainability:

- Code maintainability: Write well-documented, modular, and maintainable code to facilitate future updates and enhancements.

- Database maintenance: Develop routines for regular database maintenance, including backups and optimization.

6. Compatibility:

- Platform compatibility: Ensure that the application works seamlessly on different operating systems and browsers, if applicable.

- Database Compatibility: Ensure compatibility with the selected database management system.

User-friendly interface design

• Create intuitive and user-friendly interfaces for all modules to ensure ease of use for customers, employees, and management.

• Prioritize clear navigation, consistent layouts, and easy access to relevant information on each interface.

• Conduct user testing to gather feedback and improve the interface.

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Challenges**

* Scheduling Conflicts: Coordinating the schedules of four group members is challenging, making it difficult to find suitable meeting times.
* Task Management: Handling multiple tasks requires clear responsibilities and deadlines to prevent overlaps or omissions.
* Online Collaboration: Adapting to and effectively using online collaboration tools can be a learning curve for some group members.
* Communication Gaps: Misunderstandings and lack of clarity can occur due to differing schedules, leading to communication gaps.
* Motivation and Accountability: Keeping all group members motivated and accountable, especially when working remotely, poses a challenge.
* Conflict Resolution: Addressing disagreements or conflicts within the group requires effective resolution strategies.
* Workload Management: Balancing project work with individual commitments and responsibilities can lead to stress and time management challenges.
* Progress Tracking: Tracking project progress, especially when group members work at different times, can be complex.
* Peer Coordination: Ensuring alignment between different members' contributions can be tricky.
* Flexibility: Adapting to changing schedules and unforeseen circumstances requires flexibility and adaptability.

Time Management: Balancing individual tasks with team responsibilities was challenging. Meeting deadlines and managing time efficiently is critical.

Online Collaboration: Adapting to and effectively using online collaboration tools can be a learning curve for some group members.

Communication Gaps: Misunderstandings and lack of clarity can occur due to differing schedules, leading to communication gaps.

Motivation and Accountability: Keeping all group members motivated and accountable, especially when working remotely, poses a challenge.

Conflict Resolution: Addressing disagreements or conflicts within the group requires effective resolution strategies.

Workload Management: Balancing project work with individual commitments and responsibilities can lead to stress and time management challenges.

Flexibility: Adapting to changing schedules and unforeseen circumstances requires flexibility and adaptability.

Progress Tracking: Tracking project progress, especially when group members work at different times, can be complex.

Scheduling Conflicts: Coordinating the schedules of four group members is challenging, making it difficult to find suitable meeting times.

Task Management: Handling multiple tasks requires clear responsibilities and deadlines to prevent overlaps or omissions.

Coordination issues: Ensuring alignment between the contributions of different members can be tricky.

**[11]. Future enhancements**

Artificial Intelligence (AI) and Machine Learning:

Integration of AI and machine learning algorithms for predictive analytics, task automation, and intelligent decision support. AI can assist in resource allocation, risk assessment, and project forecasting.

Advanced Reporting and Analytics:

More sophisticated reporting and analytics capabilities, including real-time dashboards, customizable KPIs, and data visualization tools for better project insights and decision-making.

Enhanced Collaboration Features:

Improved collaboration tools, including virtual whiteboards, integrated messaging, and video conferencing, to facilitate seamless communication among remote and global teams.

Blockchain Integration: Blockchain technology for enhanced security, transparency, and traceability in project management, particularly in industries like supply chain management and finance.

IoT Integration:

Integration with the Internet of Things (IoT) devices to collect real-time data from project equipment and assets, enabling proactive maintenance and monitoring.

Automation of Routine Tasks:

Automation of repetitive tasks such as status updates, document approvals, and task assignments to free up project managers for more strategic work.

Enhanced Mobile Experience:

Mobile apps with improved functionality and offline capabilities to allow team members to work on projects from anywhere.

AI-driven Risk Management:

AI-powered risk assessment and mitigation tools that can predict and proactively address potential project risks.

Security and Privacy:

Enhanced security measures to protect sensitive project data and compliance with evolving data privacy regulations (e.g., GDPR).

Resource Optimization:

Advanced resource management features for optimizing the allocation of human and material resources based on skill sets and availability.

* **Programming Language:** The primary programming language used for this project is C#, which is commonly used for Windows Forms applications and integrates with various libraries and frameworks.
* **Development Environment**: The project is likely developed using Visual Studio, a popular integrated development environment (IDE) for C# and .NET development. Visual Studio provides a user-friendly interface for building Windows Forms applications and offers debugging and design tools.
* **SQL Server Database:** The project connects to a SQL Server database using System.Data.SqlClient. This is evident in the code where SQL Server connections and commands are used for data storage and retrieval.
* **Windows Forms:** The user interface (UI) for the project is built using Windows Forms. Windows Forms is a graphical user interface framework provided by Microsoft for developing desktop applications in C#.
* **Database:** The project appears to interact with a database to store information related to employees, customers, items, and sales. The exact database management system (DBMS) is SQL Server.

**Canva**

**Draw.io**

Programming Language

The programming language used is C#, which is commonly used for Windows Forms applications and integrates with various libraries and frameworks.

Development Environment

The project is likely developed using Visual Studio, a popular integrated development environment (IDE) for C# and .NET development. Visual Studio provides a user-friendly interface for building Windows Forms applications and offers debugging and design tools.

SQL Server Database

The project connects to a SQL Server database using System.Data.SqlClient. This is evident in the code where SQL Server connections and commands are used for data storage and retrieval.

Windows Forms

The user interface (UI) for the project is built using Windows Forms. Windows Forms is a graphical user interface framework provided by Microsoft for developing desktop applications in C#.

Database

The project appears to interact with a database to store information related to employees, customers, items, and sales. The exact database management system (DBMS) is SQL Server.

Canva

Canva is a user-friendly online graphic design tool that helps people create a wide range of visual content easily, even without prior design experience. It is used to create project reports.

Draw.io

Draw.io is a web-based diagramming tool that allows users to create diagrams and visual representations of ideas and concepts.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Student Number** | **Contact Number** | **Email** |
| **Chameera Peshan** | PS/2019/265 | +94 76 007 8842 | wijerat-ps19265@stu.kln.ac.lk |
| **Harshana Eshan** | PS/2019 /137 | +94 71 604 8560 | eshanhg-ps19137@stu.kln.ac.lk |
| **Randila Shehari** | PS/2019/225 | +94 71 696 4666 |  |
|  | PS/2019/022 | +94 76 927 9527 |  |
|  | PS/2019/244 | +94 71 568 6673 |  |

[wijerat-ps19265@stu.kln.ac.lk](mailto:wijerat-ps19265@stu.kln.ac.lk) [wijerat-ps19265@stu.kln.ac.lk](mailto:wijerat-ps19265@stu.kln.ac.lk) wijerat-ps19265@stu.kln.ac.lk