

# 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.



## 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.

*We will give a  
brief introduction  
to the identified  
problems.*

02



# Our Goals and Objectives .....

# 03

- **Efficient Project Planning**

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

- **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.

---

# Our Goals and Objectives .....

# 03

- **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.

- **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.

# Our Goals and Objectives .....

## 03

- **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.

- **Resource Optimization**

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

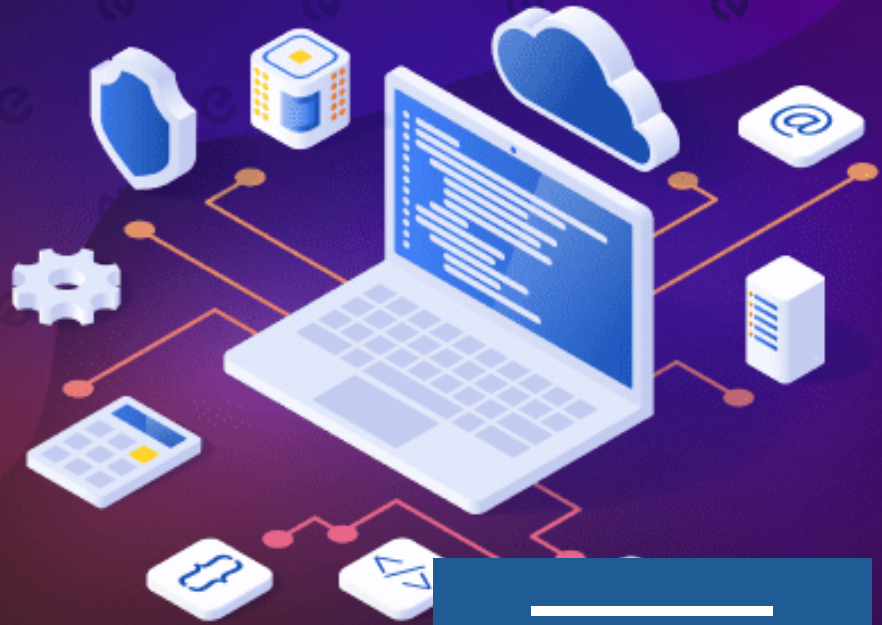
- **Cost Estimation**

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

- **Real-Time Communication**

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

# Functional vs Non-functional Requirements



## Functional Requirement .....

- **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.

*Functional requirements are explicit descriptions of what a software system or product must do to meet user needs and perform its intended tasks.*

04



# Functional Requirement .....

---



- **Reporting and Analytics**

Generate customizable project performance reports.  
Provide data visualization tools, including charts and graphs.

Offer dashboards for real-time project monitoring.

- **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.

- **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.

- **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.

- **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.

# Functional Requirement .....

---



- **Reporting and Analytics**

Generate customizable project performance reports.  
Provide data visualization tools, including charts and graphs.

Offer dashboards for real-time project monitoring.

- **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.

- **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.

- **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.

- **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.



# Non Functional Requirement

---

- **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.

---

*Non-functional requirements are specifications that define how a software system or product should perform, including attributes like speed, reliability, security, and usability, rather than specific functionalities.*

04

- **Performance:**

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

- **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.

# Non Functional Requirement

---



- **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.

- **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.

- **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**

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.