ومن يؤت الحكمة
فقد أوتي خيرا كثيرا

QUAID-I-AZAM UNIVERSITY

ISLAMABAD

# Project Plan Document

# Project Plan

According to ISO/IEC/IEEE 16326 Guidelines

**Course**: Software Construction

**Submitted by**: Group 1 (Software Construction Class)

**Team Lead**: Abdur Rahman

**Team Members**: Hussain Ali, Noman Khan

**Submission Date**: 10/07/2024

**Audience:** Development Team, Client (Prof. Onaiza Maqbool) and stakeholders.

**Author:** Abdur Rahman (Team Lead)

**Co-Authors**: Hussain Ali and Noman Khan

**Task Table**

| Author | Tasks Performed |
|---|---|
| Abdur Rahman | 1,2,3 |
| Hussain Ali | 4,8 |
| Noman | 5 |

**Table 2:** Task Table

## Signature Page

This project plan has been reviewed and approved by:

Name: Abdur Rahman Date: 10/07/2024

Title: Team Lead Signature AbdurRahman

Name: Hussain Ali Date: 10/07/2024

Title: Dev Team Member Signature Hussain Ali

Name: Noman Khan Date: 10/07/2024

Title: Dev Team Member Signature Noman Khan

Name: Dr. Onaiza Maqbool Date: 10/07/2024

Title: Client Signature ____________

## Change History

The change history of document is provided in table 1.

| Version | Date | Author | Change Description |
|---|---|---|---|
| 1.0 | [10/072024] | Abdur Rahman | Initial project plan |
| …… | ……. | ……. | ……. |

**Table.1:** Change History of Project Plane Document

# Preface

This document presents the project plan for GiftCart, developed as per ISO/IEC/IEEE 16326 guidelines. It outlines the project management process to ensure successful execution and delivery of the project.

## Table of Contents

Title Page ........................................ 01

Signature Page ........................................ 02

Change History ........................................ 03

Preface ........................................04

Table of Contents ........................................05

List of Figures ........................................06

List of Tables ........................................07

1. Project Overview ........................................08

1.1 Project Summary ........................................08

1.2 Purpose, Scope and Objectives ........................................08

1.3 Assumptions and Constraints ........................................10

1.4 Project Deliverables ........................................10

1.5 Schedule Summary ........................................10

2. References ........................................11

3. Definitions ........................................12

4. Project Context ........................................12

4.1 Process Model ........................................12

4.2 Methods, Tools and Techniques ........................................13

4.3 Product Acceptance Plan ........................................14

5. Project Planning ........................................15

5.1 Project Work Plans ........................................15

5.2 Work Activities ........................................15

5.3 Schedule Allocation ........................................ 16

5.4 Resource Allocation ........................................ 16

5.5 Gantt Chart ........................................17

8. Supporting Process Plans ........................................17

8.1 Risk Management ........................................ 18

## List of Figures

Figure 1: Schedule Chart ……………………………………………...............................................16

Figure 2: Resource Tabel …………………………………………………………………………..….16

Figure 3: Gantt Chart ……………………………………………………………………………………17

# List of Tables

Table 1: Change History of Project Plane Document.................................................03

Table 2: Task Table ………………………………………………………………………..01

# 1. Project Overview

## 1.1. Project Summary

The project aims to develop a comprehensive software solution for gift registries within retail shops, enabling a seamless and user-friendly experience for both hosts and guests. The system will facilitate the creation of personalized gift lists for various events, such as weddings, birthdays, and other celebrations.

Key features of the software include:

- **Gift List Management**: Shops can post a catalog of items with prices, allowing users to create event-specific gift lists based on their preferences and needs.
- **Privacy Protection**: Users can select gifts without revealing their choices to others, ensuring privacy and dignity throughout the gift selection process.
- **Dynamic Inventory Control**: Once an item is selected in the required quantity, it becomes unavailable for re-selection, streamlining the coordination of gifts.
- **Payment Processing**: Guests can securely make payments via credit card, with the system enforcing a short deadline for transactions before the event.
- **Event Coordination**: The software manages deadlines and provides notifications to ensure timely planning and execution of events.

The development process follows the **Scrum framework**, an agile methodology that emphasizes iterative progress, collaboration, and flexibility. The project will be divided into sprints, allowing for continuous feedback and improvements as new features are developed. A dedicated team of three members will collaborate in an organized, transparent manner, using Scrum ceremonies like sprint planning, daily standups, and retrospectives to ensure smooth progress.

The project is scheduled to be delivered within **14 weeks**, with regular feedback sessions from stakeholders each week. This approach will ensure that the final product meets the needs of the client and provides a reliable, secure platform for managing gift registries.

## 1.2. Purpose, Scope and Objectives

### 1.2.1. Purpose

The purpose of this project is to develop a software solution that streamlines gift registry management for both event organizers and shops, ensuring a seamless, secure, and efficient gift selection and payment process.

### 1.2.2. Scope

#### 1.2.2.1. Name

The name of the project to be developed is **GiftCart.**

#### 1.2.2.2. Context

The software system to be created is standalone system. Usual dependencies of third-party payment handling are considered.

#### 1.2.2.3. Main Function

a. **Registry Management**: Allows users to create, view, and manage event-specific gift registries, including item selection and updates.
b. **User Interaction**: Enables friends to browse registries, select gifts, and make purchases without revealing others' selections.
c. **Inventory Control**: Automatically updates item availability based on gift selections, ensuring accurate stock management.
d. **Payment Integration**: Facilitates secure credit card payments for selected gifts within the system.
e. **Deadline Management**: Manages event deadlines for gift selection and payment, deactivating or removing registries as needed.

### 1.2.3. Objectives

a. **Privacy and Dignity**: Ensure that friends selecting gifts cannot see what others have chosen, maintaining privacy and dignity during the gift selection process.
b. **Unique Gift Selection**: Ensure that the host receives only the specific gifts they requested in the required quantities, preventing duplicate selections and ensuring the host gets exactly what they need.
c. **Accessibility**: Provide an intuitive and accessible platform for both hosts (event organizers) and guests (friends), allowing easy access to registries and gift selection from any device even from remote areas.
d. **Deadline and Event Coordination**: Manage event deadlines effectively, including coordination of gift selection and payment deadlines, to ensure timely gift purchases and event preparation.
e. **Seamless Payment Integration**: Implement secure, seamless credit card payment processing, ensuring guests can complete transactions efficiently and without difficulty.
f. **Receipt Generation**: Automatically generate and distribute receipts for both guests and the shop, confirming payment completion and allowing for easy tracking of transactions as a proof of gift.
g. **Inventory Management**: Keep real-time track of available and selected items, updating the shop's inventory and the gift registry to reflect changes as gifts are purchased.
h. **Registry Deactivation**: Automatically deactivate or delete the registry once the event deadline has passed or all items have been selected and paid for.

### 1.3. Assumptions and Constraints

a. **Time Constraint**: The project must be completed and delivered within a 14-week timeframe.
b. **Team Size**: The development team is limited to 3 members, which impacts workload distribution and project management.
c. **Weekly Feedback**: Regular feedback must be incorporated, as weekly progress reports and reviews are required.
d. **Technology Constraint**: Java must be used as the primary programming language for the development of the software.

### 1.4. Project Deliverables

1. **Project Plan**: A detailed document outlining the objectives, scope, timeline, resources, and risks, adhering to the ISO/IEC/IEEE 16326 guidelines.
2. **System Design Document**: A comprehensive design document describing the system architecture, database design, class diagrams, and UI mockups.
3. **Gift Registry Software**: The fully functional software that includes all the required features, such as registry management, payment integration, and inventory control, built using Java.
4. **User Documentation**: A user manual explaining how to use the system for both event hosts and guests, including steps for creating a registry, selecting gifts, and making payments.
5. **Test Plan and Test Cases**: A plan outlining the testing approach for the software, along with a suite of test cases to verify functionality, security, and performance.
6. **Weekly Progress Reports**: Regular updates submitted weekly, detailing the team's progress, any challenges encountered, and feedback integration.
7. **Final Report**: A summary of the entire project process, including lessons learned, challenges faced, and how they were resolved, delivered at the conclusion of the project.
8. **Deployment Package**: A packaged version of the software ready for deployment, including installation instructions and configuration details for the shop and server environment.
9. **Project Libre Schedule**: A detailed project schedule created using Project Libre, outlining the project phases, tasks, milestones, and deadlines.
10. **Source Code**: The complete source code of the software, well-documented and following coding standards, to be uploaded to a version control system (e.g., GitHub).

### 1.5. Schedule Summary

1. **Pre-Planning** (October 2, 2024)

In this initial phase, we will focus on understanding the project objectives and establishing early communication with the client to ensure alignment before formal planning begins.

2. **Planning** (October 3, 2024)

During the planning phase, a detailed project plan will be created, outlining all tasks, timelines, and resources required for the successful execution of the project. This phase will also include a review to confirm that all stakeholders are on the same page.

3. **Analysis** (October 9 – October 24, 2024)

The analysis phase will involve defining the use cases, developing analysis models, and creating the Software Requirements Specification (SRS). This will provide a comprehensive understanding of the system's functional and non-functional requirements.

4. **Design** (October 31 – November 21, 2024)

In this phase, the focus will be on creating detailed system designs, including data structures, interfaces, and the overall system architecture. The design phase ensures that all technical aspects of the project are well defined before development begins.

5. **Development** (November 27 – December 19, 2024)

During the development phase, the system will be built according to the design specifications. This includes developing the front-end, back-end, database connectivity, and other system components.

6. **Testing** (December 25 – December 26, 2024)

The system will undergo thorough testing to ensure it meets all functional requirements, performs well, and is free of defects before final delivery.

7. **Presentation** (January 1, 2025)

The project will conclude with a presentation to the client and stakeholders, where the final product will be demonstrated, marking the official handover of the system.

## 2. References

a. **ISO/IEC/IEEE 16326:2019**
Systems and Software Engineering — Life Cycle Processes — Project Management
Description: Provides the guidelines for structuring and content of the project plan.

*b.* **Java Documentation**
Oracle's official documentation on Java, used as the programming language for developing the system.
*Source: https://docs.oracle.com/javase/*

c. **Project Management Guidelines**
Course materials provided by the professor, covering project management and planning principles, feedback processes, and the use of Project Libre.

# 3. Definitions

This section provides definitions for technical terms and keywords used throughout the project plan.

1. **Gift Registry:** A list of items that a person compiles, typically for an event such as a wedding or birthday, allowing others to purchase gifts from the list.
2. **Host:** The person who creates the gift registry for their event.
3. **Guest**: A person who accesses the gift registry to select and purchase gifts for the host's event.
4. **Event Deadline**: A date set by the host and the shop, after which the gift registry is disabled, and purchases are no longer allowed.
5. **Java**: A high-level, object-oriented programming language used for developing the gift registry software. *Reference: Oracle Java Documentation.*
6. **PCI DSS (Payment Card Industry Data Security Standard)**: A security standard ensuring secure handling of credit card information. It is relevant to the payment integration in the system. *Reference: PCI Security Standards Council.*
7. **Secure Payment Gateway**: A service that processes credit card payments and ensures security during online transactions. In this project, it handles payments made by guests on behalf of the host.
8. **Project Libre**: A free, open-source project management software used to create the schedule and manage tasks for this project.
9. **ISO/IEC/IEEE 16326:** An international standard providing guidelines for project management, especially in software systems engineering. It serves as a framework for structuring the project plan.
10. **Frontend**: The part of the software system that interacts directly with users, including the graphical user interface (GUI) where hosts and guests can interact with the registry.
11. **Backend**: The server-side part of the software that handles data processing, business logic, database management, and communication with the frontend.
12. **Database**: A structured collection of data that stores information related to the gift registry, such as item lists, guest selections, and payment details.
13. **Gantt/Timeline Chart**: A visual representation of timespan of tasks carried out in the development process.

# 4. Project Context

## 4.1. Process Model

The project will employ an iterative and incremental process model using **Scrum** for the development of the Gift Cart project. Scrum promotes effective communication, collaboration, and the swift delivery of value to users. Each sprint will concentrate on delivering functional increments, including registry creation, gift selection, and payment processing, facilitating ongoing improvement and refinement based on stakeholder input.

- **Sprint Planning**: Defining the goals for the upcoming sprint.
- **Daily Stand-ups**: A brief meeting to review progress and address any blockers.
- **Sprint Review**: Demonstrating the completed features to stakeholders for feedback.
- **Sprint Retrospective**: Reflecting on the sprint to identify areas for improvement.

## 4.2. Methods, Tools, Techniques

### 4.2.1. Methods:

- **OOP (**Object oriented programming)

  The ‘’ GiftCart’’ system will be developed using object-oriented programming principles making it modular and reuseable. This will allow us to create objects like ‘User’, ‘Item’, ‘Payment’ etc

### 4.2.2. Tools:

To support the management and development of the gift cart project following tools will be employed:

1. **JAVA, JAVA SWING**
   The system will be developed using JAVA as programming language and JAVA SWING for building the user interface
2. **Visual Studio Code** (VS Code)
   VS Code will be used as the code editor for writing, managing, and debugging the system’s source code.
3. **MS Word**
   For all project related documentation, including requirement documents, user manuals and reports MS Word will be used.
4. **Project Libre**
   Project libre will be used for project management basically for creating project timeline, scheduling tasks, tracking milestones and ensuring that project is always within deadlines.
5. **Microsoft SQL Server**
   Microsoft SQL Server will serve as database for storing project`s data including user information, gift cart registry details, and payments records.
6. **MS PowerPoint**
   Microsoft PowerPoint for final presentation of product

7. **Argo UML**
   A software for standard modelling of design and analysis models.

## 4.3. Product Acceptance Plan

The Product Acceptance Plan outlines how the "GiftCart" system will be reviewed, tested, and approved to ensure it meets the agreed-upon requirements.

### 4.3.1. Acceptance Criteria:

Each feature (like registry creation, gift selection, and payment processing) will be evaluated based on simple criteria such as:

- **Functionality**
- **Usability**
- **Performance**
- **Security**

### 4.3.2. User Acceptance Testing (UAT):

At the end of each development phase, users will test the system to make sure it meets their needs. During UAT:

- Users will try out the system to ensure it works as expected.
- Any problems found will be noted and fixed before moving forward.

### 4.3.3. Testing and Review Strategy:

- After each task or feature is completed, it will be reviewed to make sure it meets the requirements.
- We will submit deliverables to the client on a weekly basis for feedback.
- Based on the client’s feedback, necessary changes will be made.
- We will discuss the next steps with the client regularly to ensure the project stays on track and meets expectations.

# 5. Project Planning

## 5.1. Project Work Plan

### 5.1.2. Work Activities:

Work activities are tasks according to which the project work is performed.

Work activities in this project are:

1. **Planning:**

   In planning stage, we develop the following:

   a. Develop Project Work Plan:  We Create a detailed project schedule, timeline.
   b. Define Project Scope: Identify project goals, deliverables.
   c. Review: We review our work performed during planning.

2. **Analysis:**

   a. Define Use Cases: We write use cases in text form and also draw use case diagram.
   b. Develop Analysis Model: Analysis model like domain model, system sequence diagram etc are created.
   c. SRS Development, Review, and Refinement: We develop, review, and refine, if necessary, software requirements.

3. **Design:**

   a. Data Design: We design database schema and structure.
   b. Interface Design: We design the user interface as well.
   c. Detail Design: We develop detailed system architecture.
   d. Review and Refinement of Design: We review and refine design, if necessary ,to ensure quality.

4. **Development:**

   Here the implementation is done**.**

   a. Database Connectivity: We establish database connections of our system.
   b. Front-end Development: We develop user interface components which are front-end.
   c. Class Development: Software classes are developed carefully.
   d. Refinement of Classes and Traceability Design: Classes are refined, and traceability is ensured.

5. **Testing:**

   a. Various tests are performed on our software system to ensure that it is meeting the standards and requirements.

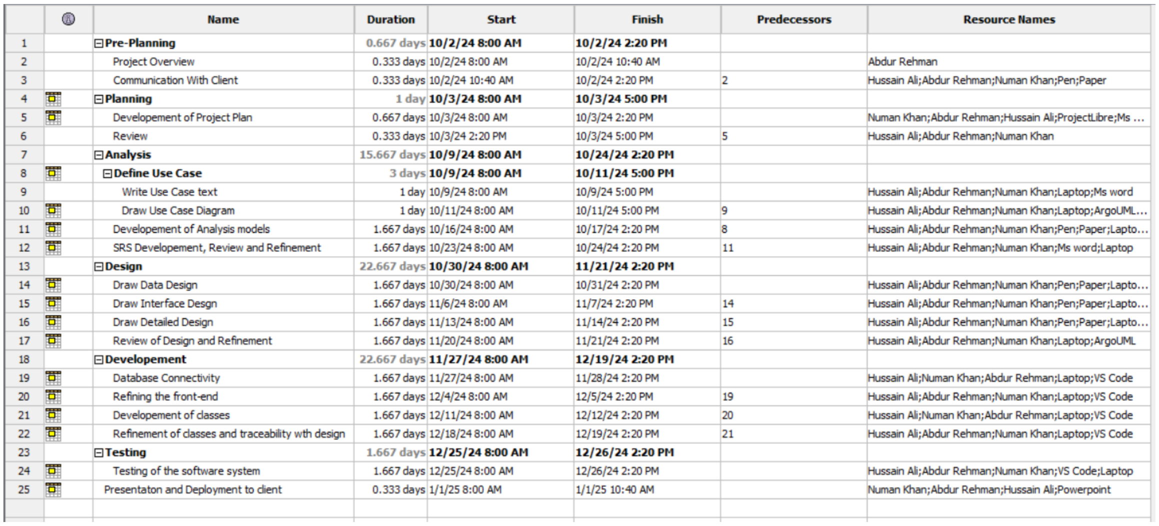
6. **Resources Required:**

   Resources are assets that help in completion of work activities.

   **Work Resources :** Human resources (team members/people involve).

   **Material Resources**: Hardware, software, infrastructure.

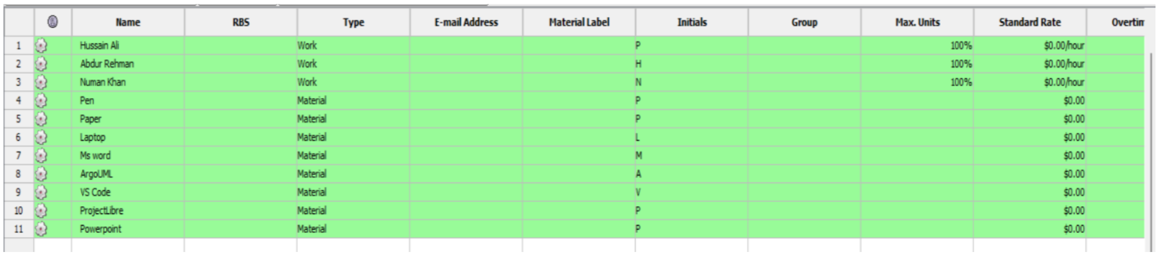
### 5.1.2. Schedule Allocation



|  | ⓘ | Name | Duration | Start | Finish | Predecessors | Resource Names |
|---|---|---|---|---|---|---|---|
| 1 |  | ⊟**Pre-Planning** | 0.667 days | **10/2/24 8:00 AM** | **10/2/24 2:20 PM** |  |  |
| 2 |  | Project Overview | 0.333 days | 10/2/24 8:00 AM | 10/2/24 10:40 AM |  | Abdur Rehman |
| 3 |  | Communication With Client | 0.333 days | 10/2/24 10:40 AM | 10/2/24 2:20 PM | 2 | Hussain Ali;Abdur Rehman;Numan Khan;Pen;Paper |
| 4 | ▦ | ⊟**Planning** | 1 day | **10/3/24 8:00 AM** | **10/3/24 5:00 PM** |  |  |
| 5 | ▦ | Developement of Project Plan | 0.667 days | 10/3/24 8:00 AM | 10/3/24 2:20 PM |  | Numan Khan;Abdur Rehman;Hussain Ali;ProjectLibre;Ms ... |
| 6 |  | Review | 0.333 days | 10/3/24 2:20 PM | 10/3/24 5:00 PM | 5 | Hussain Ali;Abdur Rehman;Numan Khan |
| 7 |  | ⊟**Analysis** | 15.667 days | **10/9/24 8:00 AM** | **10/24/24 2:20 PM** |  |  |
| 8 | ▦ | ⊟**Define Use Case** | 3 days | **10/9/24 8:00 AM** | **10/11/24 5:00 PM** |  |  |
| 9 |  | Write Use Case text | 1 day | 10/9/24 8:00 AM | 10/9/24 5:00 PM |  | Hussain Ali;Abdur Rehman;Numan Khan;Laptop;Ms word |
| 10 | ▦ | Draw Use Case Diagram | 1 day | 10/11/24 8:00 AM | 10/11/24 5:00 PM | 9 | Hussain Ali;Abdur Rehman;Numan Khan;Laptop;ArgoUML... |
| 11 | ▦ | Developement of Analysis models | 1.667 days | 10/16/24 8:00 AM | 10/17/24 2:20 PM | 8 | Hussain Ali;Abdur Rehman;Numan Khan;Pen;Paper;Lapto... |
| 12 | ▦ | SRS Developement, Review and Refinement | 1.667 days | 10/23/24 8:00 AM | 10/24/24 2:20 PM | 11 | Hussain Ali;Abdur Rehman;Numan Khan;Ms word;Laptop |
| 13 |  | ⊟**Design** | 22.667 days | **10/30/24 8:00 AM** | **11/21/24 2:20 PM** |  |  |
| 14 | ▦ | Draw Data Design | 1.667 days | 10/30/24 8:00 AM | 10/31/24 2:20 PM |  | Hussain Ali;Abdur Rehman;Numan Khan;Pen;Paper;Lapto... |
| 15 | ▦ | Draw Interface Desgn | 1.667 days | 11/6/24 8:00 AM | 11/7/24 2:20 PM | 14 | Hussain Ali;Abdur Rehman;Numan Khan;Pen;Paper;Lapto... |
| 16 | ▦ | Draw Detailed Design | 1.667 days | 11/13/24 8:00 AM | 11/14/24 2:20 PM | 15 | Hussain Ali;Abdur Rehman;Numan Khan;Pen;Paper;Lapto... |
| 17 | ▦ | Review of Design and Refinement | 1.667 days | 11/20/24 8:00 AM | 11/21/24 2:20 PM | 16 | Hussain Ali;Abdur Rehman;Numan Khan;Laptop;ArgoUML |
| 18 |  | ⊟**Developement** | 22.667 days | **11/27/24 8:00 AM** | **12/19/24 2:20 PM** |  |  |
| 19 | ▦ | Database Connectivity | 1.667 days | 11/27/24 8:00 AM | 11/28/24 2:20 PM |  | Hussain Ali;Numan Khan;Abdur Rehman;Laptop;VS Code |
| 20 | ▦ | Refining the front-end | 1.667 days | 12/4/24 8:00 AM | 12/5/24 2:20 PM | 19 | Hussain Ali;Abdur Rehman;Numan Khan;Laptop;VS Code |
| 21 | ▦ | Developement of classes | 1.667 days | 12/11/24 8:00 AM | 12/12/24 2:20 PM | 20 | Hussain Ali;Numan Khan;Abdur Rehman;Laptop;VS Code |
| 22 | ▦ | Refinement of classes and traceability wth design | 1.667 days | 12/18/24 8:00 AM | 12/19/24 2:20 PM | 21 | Hussain Ali;Abdur Rehman;Numan Khan;Laptop;VS Code |
| 23 |  | ⊟**Testing** | 1.667 days | **12/25/24 8:00 AM** | **12/26/24 2:20 PM** |  |  |
| 24 | ▦ | Testing of the software system | 1.667 days | 12/25/24 8:00 AM | 12/26/24 2:20 PM |  | Hussain Ali;Abdur Rehman;Numan Khan;VS Code;Laptop |
| 25 | ▦ | Presentaton and Deployment to client | 0.333 days | 1/1/25 8:00 AM | 1/1/25 10:40 AM |  | Numan Khan;Abdur Rehman;Hussain Ali;Powerpoint |

**Figure.1**: Schedule Chart

### 5.1.3. Resource allocation



|  | ⓘ | Name | RBS | Type | E-mail Address | Material Label | Initials | Group | Max. Units | Standard Rate | Overtin |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | ⚙ | Hussain Ali |  | Work |  |  | P |  | 100% | $0.00/hour |  |
| 2 | ⚙ | Abdur Rehman |  | Work |  |  | H |  | 100% | $0.00/hour |  |
| 3 | ⚙ | Numan Khan |  | Work |  |  | N |  | 100% | $0.00/hour |  |
| 4 | ⚙ | Pen |  | Material |  |  | P |  |  | $0.00 |  |
| 5 | ⚙ | Paper |  | Material |  |  | P |  |  | $0.00 |  |
| 6 | ⚙ | Laptop |  | Material |  |  | L |  |  | $0.00 |  |
| 7 | ⚙ | Ms word |  | Material |  |  | M |  |  | $0.00 |  |
| 8 | ⚙ | ArgoUML |  | Material |  |  | A |  |  | $0.00 |  |
| 9 | ⚙ | VS Code |  | Material |  |  | V |  |  | $0.00 |  |
| 10 | ⚙ | ProjectLibre |  | Material |  |  | P |  |  | $0.00 |  |
| 11 | ⚙ | Powerpoint |  | Material |  |  | P |  |  | $0.00 |  |

**Figure.2:** Resource table

### 5.1.4. Gannt/Timeline chart

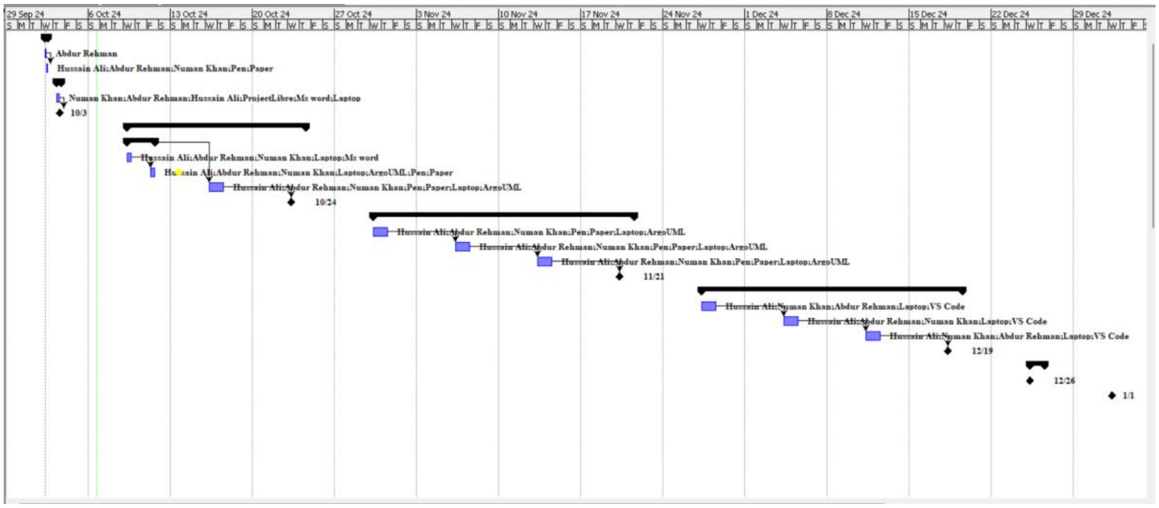


**Figure.3**: Gantt Chart/Timeline Chart

# 8. Supporting Process Plan

## 8.1: Risk Management:

In risk management plan as a development team we have to identify and manage potential risks that may occur throughout project lifecycle, focusing on identifying, analyzing, treating and monitoring these risk.

### 8.1.1. Identify Risks:

As a development team we Identify risks related to development, operation and management of Gift Cart Software. Some examples include

1. **Security Risks:** Data breaches.
2. **System Failure:** Risk of system downtown.
3. **Payment Gateway Issues:**
4. **Scalability:** Risk that system will not handle high traffic.
5. **Schedule Delays:** Falling behind the project timeline**.**

6. **Resource Issues:** Lack of team members, budget and technical resources.
7. **Client Misunderstanding:** Misinterpretation of client requirements leading to rework.

### 8.1.2. Risk Analysis:

We analyse the identified risk and determine their potential errors and likelihood.

1. **Security Risks:** High impact as data breach could lead to loss of customer trust.
2. **System Failure:** Medium to high depending on the length of downtown.
3. **Payment Gateway Issues:** High impact as failed payments could result in unprocessed order.
4. **Scalability:** High impact as system crashes during traffic.
5. **Schedule Delays:** Moderate delays can impact project milestones, but they can be adjusted.
6. **Resource Issues:** High impact as lack of resources can impact project delay and its quality.
7. **Client Misunderstanding:** Misalignment with client requirements can cause major rework.

### 8.1.3. Risk Treatment:

Here we try to either reduce the risk or will reduce the possibility of risk:

1. **Security Risks:** Implement Strong encryption.
2. **System Failure:** Setup backup servers and monitors to reduce downtown.
3. **Payment Gateway Issues:** Test the payment gateway thoroughly and provide multiple options.
4. **Scalability:** Use efficient cloud infrastructure to handle increased traffic.
5. **Schedule Delays:** Use project management tools, define better times, and monitor milestones.
6. **Resource Issues:** Ensure proper planning for recourses.
7. **Client Misunderstanding**: Conduct regular client meetings, use proper documentation and create prototypes to validate misunderstanding.

### 8.1.2. Risk Monitoring:

Here we monitor actions that we take to treat risks are working accordingly or not:

1. **Security Risks:** Regular security checks.
2. **System Failure:** Monitor the system to ensure that it is responsive.
3. **Payment Gateway Issues:** Track payment completion rates and solve related issues immediately.
4. **Scalability:** Monitor traffic patterns to avoid scalability issues.
5. **Schedule Delays:** Use Gantt charts or project management tools to track progress.
6. **Resource Issues:** Review resource usage and availability regularly to ensure the project stays on track.
7. **Client Misunderstanding:** Maintain regular client feedback loops to make sure that project is on right track