**CSE432**

**Software Project Management**

**Practical File**

Submitted to

**AMITY UNIVERSITY, UTTAR PRADESH**

A blue and yellow logo

Description automatically generated with low confidence

**In partial fulfilment of the requirements for the award of the degree of**

**Bachelor of Technology**

**In**

**Computer Science & Engineering**

Submitted By:

Sumit Kumar

A2305220005

7CSE-2 X-Batch

Submitted to:

Dr. Abhishek Singhal

**DEPARTMENT OF AMITY SCHOOL OF ENGINEERING AND TECHNOLOGY**

**AMITY UNIVERSITY NOIDA UTTAR PRADESH**

**TABLE OF CONTENTS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No** | **PROGRAM** | **Date Assigned** | **Date Submitted** | **Signature** |
| 1 | Lab 1  To prepare software requirement specification(SRS) document for Payroll Management System. | 13-07-23 |  |  |
| 2 | Lab 2  To describe the activities that are to be conducted for development of the project Payroll Management System. | 27-07-23 |  |  |
| 3 | Lab 3  To draw the Program Evaluation and Review Technique (PERT) chart for the project Payroll Management System. | 27-07-23 |  |  |
| 4 | Lab 4  To draw the GANTT chart for the project Payroll Management System. | 10-08-23 |  |  |
| 5 | Lab 5  To prepare project plan for Payroll Management System, | 24-08-23 |  |  |
| 6 | Lab 6  To implement Critical path method for Payroll Management System. | 14-09-23 |  |  |
| 7 | Lab 7  To draw Workflow diagram for the Payroll Management System. | 28-09-23 |  |  |
| 8 | Lab 8  To design Slip chart, Timeline chart and Ball chart for Payroll Management System. | 05-10-23 |  |  |
| 9 |  |  |  |  |
| 10 |  |  |  |  |

**Lab 1**

**Software Requirements Specification Document**

**Payroll Management System**



Prepared by: Project Guide:

**Ranesh Krishna Dr. Abhishek Singhal**

**Nilesh Chopra Associate Professor**

**Vishal Singh Department of CSE**

**Sumit Kumar**

**Table of Contents**

**Table of Contents ii**

**Revision History ii**

**1. Introduction 1**

1.1 Purpose 4

1.2 Scope 4

1.3 Benefits 4

1.4 Definition, Acronyms and Abbreviations 5

1.5 Overview 5

**2. Overall Description 6**

2.1 Product Perspective 6

2.2 Product Functions 6

2.3 User Classes and Characteristics ………………….…………………….6

2.4 Operating Environment ………………………………………………….7

2.5 Design and Implementation Constraints ………………………………...7

2.6 User Documentation ...…………………………………………………..7

2.7 Assumptions and Dependencies………………………………………….8

**3. External Interface Requirements 9**

3.1 User Interfaces 9

3.2 Hardware Interfaces 9

3.3 Software Interfaces 9

3.4 Communications Interfaces 10

**4. System Features 10**

4.1 Employee Database 10

4.2 Payroll Processing 10

4.3 Period Management ……………………………………………………..10

4.4 Time Tracking Integration ………………………………………………10

4.5 Deductions and Taxes …………………………………………………...11

4.6 Payslip Generation ………………………………………………………11

4.7 Reports and Analytics …………………………………………………...11

4.8 Compilance and Legal reports …………………………………………..11

**5. Other Nonfunctional Requirements 11**

5.1 Performance 11

5.2 Scalability 11

5.3 Reliability 11

5.4 Availability 11

5.5 Security 11

5.6 Accessibility …………………………………………………………….12

5.7 Data Integrity ……………………………………………………………12

5.8 Auditability ……………………………………………………………...12

**6. Other Requirements 12**

6.1 Integration with HR Information System ……………………………….12

6.2 Integration with Attendance Management System ………………………12

6.3 Multicurrency Support …………………………………………………..12

6.4 Localization for Tax and Legal Requirements …………………………..12

6.5 Employee Self-Service Portal …………………………………………...12

6.6 Automatic Updates and Regulatory Compliance ……………………….13

6.7 GDPR Compiliance ……………………………………………………..13

6.8 Training and Documentation ……………………………………………13

**Appendix A: Analysis Models 13**

1 ER Diagram ……………………………………………………………….13

2 Class Diagram ……………………………………………………………..14

3 Activity Diagram ..………….……………………………………………..15

4 Sequence Diagram ………………………………………..……………….16

**Appendix B: References 16**

**1. Introduction**

In the fast-paced corporate landscape of today, ensuring smooth and precise management of employee payroll holds paramount importance for any company. Payroll involves intricate responsibilities, such as computing salaries, taxes, deductions, and bonuses, necessitating scrupulous attention to avoid mistakes and ensure compliance with local laws. To simplify and automate these intricate tasks, a Payroll Management System (PMS) becomes an invaluable asset for contemporary businesses, streamlining operations and boosting efficiency.

* 1. **Purpose**

The Payroll Management System will encompass a comprehensive set of features to handle various payroll tasks, such as employee information management, salary calculation, tax deductions, benefits administration, leave tracking, attendance management, and generating pay slips. The system will be designed to cater to the needs of both small and large organizations, enabling them to effectively manage their workforce and payroll processes.

* 1. **Scope**

This application facilitates seamless collaboration among users from different departments and locations by sharing a common database. It allows multiple PCs installed on various computers to access and interact with the same database concurrently. With the database hosted online, employees from different locations can conveniently use the application to manage their respective departments. The online domain of the application ensures accessibility and connectivity regardless of the physical location, enabling efficient and real-time collaboration among users.

* 1. **Benefits**
* **Efficiency:** Automates complex payroll tasks, reducing manual efforts and processing time, leading to faster and more efficient payroll processing.
* **Accuracy:** Minimizes human errors in salary calculations, tax deductions, and benefit administration, ensuring precise and reliable payroll outcomes.
* **Compliance:** Helps the organization adhere to local labor laws, tax regulations, and statutory requirements, reducing the risk of legal penalties and ensuring compliance with changing regulations.
* **Data Centralization:** Centralizes all payroll-related data in one secure database, providing easy access and better data management for HR and finance teams.
* **Cost Savings:** Streamlines payroll processes, reducing administrative costs, and preventing over or underpayments, leading to cost savings for the organization.
  1. **Definitions, Acronyms, Abbreviations**
* SRS - Software Requirements Specification.
* EPM - Employee Payroll Management
* HR: Human Resources, the department responsible for managing personnel, employee benefits, and workforce-related functions within an organization.
* API: Application Programming Interface. A set of rules and protocols that allow different software applications to communicate and interact with each other.
  1. **Overview**

The following subsections provide the complete overview of the software specifications requirements documentation for the product Employee Payroll Management. The entire SRS is documented in view of User and the following sub sections are arranged to give a complete outlook of the software, its perspective, features, system requirements and users know how it is.

**2. Overall Description**

**2.1 Product Perspective**

This purpose-built payroll software offers an intuitive and self-contained solution to efficiently manage all aspects of payroll processing within the company. Its user-friendly interface ensures easy navigation for both newcomers and experienced users, while its robustness guarantees the security and accuracy of sensitive payroll data. With streamlined operations, compliance with regulations, and real-time accessibility, the software proves to be an indispensable tool for enhancing overall efficiency and accuracy in payroll management.

**2.2 Product Functions**

**2.2.1 Employee module**

* Designation
* Department
* Employee details

**2.2.2 User module**

* Users
* User details

**2.2.4 Attendance module**

* Leave
* Attendance
* Managing Leave

**2.2.5 Salary module**

* Allowances
* Deductions
* Pay Slip

**2.3 User Classes and Characteristics**

**2.3.1 End Users**

* The end users do not need any specialized knowledge or skills.
* Basic familiarity with computer operations and databases is sufficient for the end users.

**2.3.2 Administrator**

* The administrator should have technical expertise to manage the system effectively.
* Effective data management skills are crucial for handling sensitive payroll information.
* Security consciousness is essential to protect payroll data from unauthorized access.
* The administrator must manage user access and permissions within the system.
* Knowledge of payroll laws and regulations is necessary for ensuring compliance.
* Problem-solving abilities are required to address complexities and discrepancies in payroll.
* Strong communication skills are important for interacting with stakeholders.
* Time management is vital for meeting payroll processing deadlines.
* The administrator should be adaptable to evolving payroll processes and updates.
* Ethical conduct is fundamental in handling confidential employee information.

**2.4 Operating Environment**

* Memory: Minimum of 512MB RAM (1GB Recommended)
* Hard Disk: At least 40 GB of storage space
* Printer: Required for generating physical copies of payroll-related documents
* Operating System: Compatible with Windows, macOS, Linux, or cloud-based platforms.

**2.5 Design and Implementation constraints**

* Security and Privacy: Robust measures to protect sensitive data.
* Accuracy and Precision: Accurate payroll calculations and processing.
* Scalability: Ability to handle growing data and employee numbers.
* Compliance: Adherence to legal and regulatory requirements.
* Usability: User-friendly and intuitive interface.
* Performance: Optimized for efficiency and responsiveness.
* Integration: Seamless data exchange with other systems.
* Reliability: Minimal downtime and backup provisions.
* Data Validation: Rigorous checks to prevent errors.
* Cost Constraints: Balancing functionality with budget limitations.
* Legacy System Integration: Coexistence with existing systems.
* Reporting and Analytics: Comprehensive insights into payroll data.
* Multilingual and Multicurrency Support: Handling diverse regions and currencies.

**2.6 User Documentation**

**2.6.1 Adding New Employees**

* Navigate to "Employee Management" from the menu.
* Click "Add New Employee" and fill in the required details, such as name, contact information, and job position.
* Save the information, and the new employee will be added to the system.

**2.6.2 Editing Employee Details**

* Go to "Employee Management" and search for the employee you wish to edit.
* Click on the employee's name to access their details.
* Make the necessary changes and save the updated information.

**2.6.3 Terminating Employees**

* In "Employee Management," find the employee to terminate.
* Select the "Terminate" option and provide the termination date and reason.
* Confirm the termination, and the employee's status will be updated accordingly.

**2.7 Assumptions and Dependencies**

* The product must have an interface which is simple enough to understand.
* All necessary hardware and software are available for implementing and use of the tool.
* The proposed system would be designed, developed and implemented based on the software requirements specifications document.
* End users should have basic knowledge of computer and we also assure that the users will be given software training documentation and reference material.
* The system is not required to save generated reports.

**3. External Interface Requirements**

**3.1 User Interfaces**

* Intuitive and user-friendly design for easy navigation and efficient usage by payroll administrators.
* Responsive across various devices, allowing access to the system from desktops, laptops, and mobile devices.
* Dashboard overview presenting key metrics, enabling quick insights into payroll-related data.
* Employee management with search and sorting capabilities, facilitating easy access to employee information.
* Step-by-step payroll processing, guiding administrators through each stage to ensure accuracy and completeness.
* Easy pay slip generation and distribution, streamlining the process of providing employees with their payment details.
* Pre-defined and customizable reports for generating various payroll-related analyses and summaries.
* Real-time data validation and error handling to identify and rectify issues promptly.
* Secure login and access controls, ensuring that only authorized personnel can access sensitive payroll data.
* Seamless integration with other systems, enabling smooth data exchange with HR and accounting platforms.
* Clear communication of data privacy and compliance measures, adhering to data protection regulations.
* Comprehensive help and support features to assist administrators in using the system effectively.
* Consideration for accessibility and localization, accommodating users with diverse needs and locations.
* Backup and recovery capabilities, safeguarding data in case of system failures or disasters.
* Optimized performance for efficient use, enhancing overall system speed and responsiveness.

**3.2 Hardware Interfaces**

* A server to host the software and database, ensuring centralized data storage and access.
* Workstations (desktops/laptops) for user access, allowing payroll administrators to interact with the system.
* A Local Area Network (LAN) to establish seamless connectivity between workstations and the server.
* Internet connectivity for cloud-based systems and external services, facilitating real-time data exchange and updates.
* Printers to generate pay slips and reports for physical documentation.
* Biometric devices or time clocks to track work hours accurately and efficiently.
* Barcode/QR code scanners for quick and reliable employee identification.
* Mobile devices, if supported, to enable remote access to the system for on-the-go management.
* External storage devices for backup and data transfer to ensure data safety.
* Time-tracking devices to import work hours and integrate them into the payroll system.
* Biometric access control systems to enhance security and restrict unauthorized access.
* Barcode/QR code printers for producing identification cards for employees.

**3.3 Software Interfaces**

* Operating System: Windows, macOS, Linux
* Front End: React, Java with JavaFX or Spring Boot
* Backend: MS SQL Server
* Web Browsers: Google Chrome, Microsoft Edge, Mozilla Firefox.
* Cloud Services: AWS, Azure, Google Cloud or any other.
* Barcode Scanning Software: Softscan, Zebra, WMS or any other.

**3.4 Communications Interfaces**

* Web-Based Communication: HTTP/HTTPS protocols.
* RESTful APIs for integration with other systems.
* Database Connectivity using SQL or specific database protocols.
* Email Notifications for system events.
* Messaging Services for alerts and communications.
* Cloud-Based Communication with APIs and secure connections.
* Integration with Time: Tracking Devices for work hours data.
* Integration with Accounting Software for financial data exchange.
* Security Protocols (SSL/TLS) for secure data transmission.
* Bi-Directional Data Exchange for accurate information sharing.
* Mobile App Communication for remote access.

**4. System Features**

**4.1 Employee Database:**

Maintain an all-inclusive database containing employee information, encompassing personal details, job positions, and employment status, ensuring a complete and accurate record of the workforce.

**4.2 Payroll Processing:**

Enable automated and precise calculation of employee salaries, benefits, taxes, and deductions based on work hours and company policies, streamlining the payroll process for efficiency.

**4.3 Period Management:**

Efficiently define and manage payroll periods, ensuring a well-organized and timely payroll processing system that aligns with company requirements.

**4.4 Time Tracking Integration:**

Seamlessly integrate with time-tracking systems, allowing the importation of precise work hours for accurate and error-free payroll calculations.

**4.5 Deductions and Taxes:**

Handle diverse deductions, including benefits, insurance, and taxes, with utmost accuracy and adherence to regulatory guidelines to ensure compliance.

**4.6 Pay slip Generation:**

Generate comprehensive and detailed pay slips for each employee, providing a transparent breakdown of earnings, deductions, and net pay, enhancing transparency and communication.

**4.7 Reports and Analytics:**

Facilitate the generation of a variety of insightful reports, such as payroll summaries, tax reports, and individual employee payroll details, empowering financial analysis and informed decision-making.

**4.8 Compliance and Legal Support:**

Stay abreast of the latest tax regulations and labor laws to ensure the system remains compliant with legal requirements, minimizing the risk of non-compliance issues.

**5. Other Non-Functional Requirements**

**5.1 Performance:**

The system should respond promptly to user actions, providing quick access to data and generating reports without significant delays.

**5.2 Scalability:**

The system should be scalable to handle an increasing number of employees and a growing volume of payroll data as the organization expands.

**5.3 Reliability:**

The system must be reliable, with minimal downtime or disruptions, to ensure uninterrupted payroll processing.

**5.4 Availability:**

The system should be available for access during regular payroll processing hours and must have provisions for backup and disaster recovery to ensure data availability.

**5.5 Security:**

The system should employ robust encryption, access controls, and authentication mechanisms to protect sensitive payroll information from unauthorized access or data breaches.

**5.6 Accessibility:**

The system should be accessible to users with disabilities, adhering to accessibility standards (e.g., Web Content Accessibility Guidelines - WCAG).

**5.7 Data Integrity:**

The system should maintain data integrity, ensuring that employee records, salary calculations, and financial data are accurate and consistent.

**5.8 Auditability:**

The system should maintain an audit trail of user activities and changes to employee records for accountability and auditing purposes.

**6. Other Requirements**

**6.1 Integration with HR Information System:**

The system should integrate with the organization's HR information system to access employee data, job positions, and other relevant information.

**6.2 Integration with Attendance Management System:**

If the organization uses an attendance management system, the payroll system should seamlessly integrate to import employee attendance data for accurate payroll processing.

**6.3 Multicurrency Support:**

For international organizations, the system should handle different currencies for payroll calculations and currency conversions.

**6.4 Localization for Tax and Legal Requirements:**

The system should be customizable to adhere to specific tax laws, labor regulations, and statutory requirements of different regions.

**6.5 Employee Self-Service Portal:**

Provide an employee self-service portal to allow employees to access their pay slips, tax documents, and update personal information.

**6.6 Automatic Updates and Regulatory Compliance:**

The system should receive automatic updates to accommodate changes in tax rates, labor laws, and other regulatory requirements.

**6.7 GDPR Compliance:**

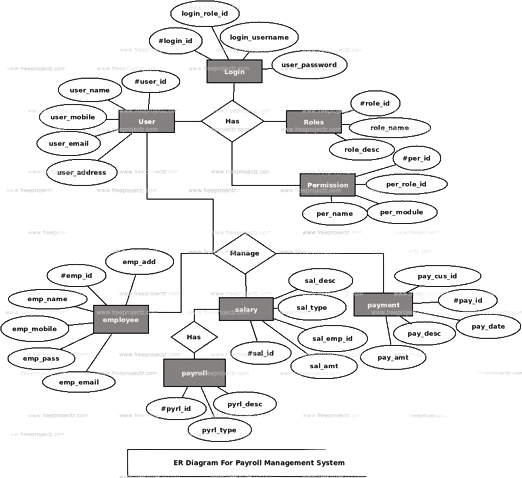
If the system handles data of European Union citizens, it should comply with the General Data Protection Regulation (GDPR) requirements.

**6.8 Training and Documentation:**

Provide comprehensive user documentation and training materials to onboard new users and enable efficient system usage.

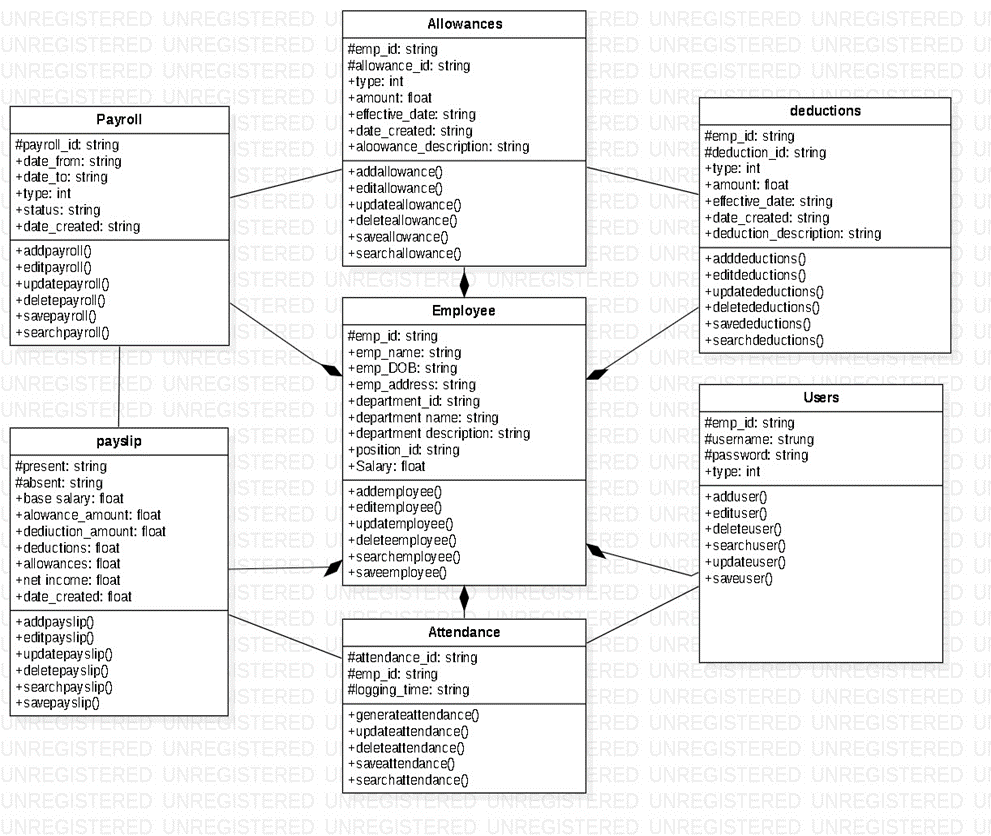
**Appendix A: Analysis Models**

**1. ER Model**



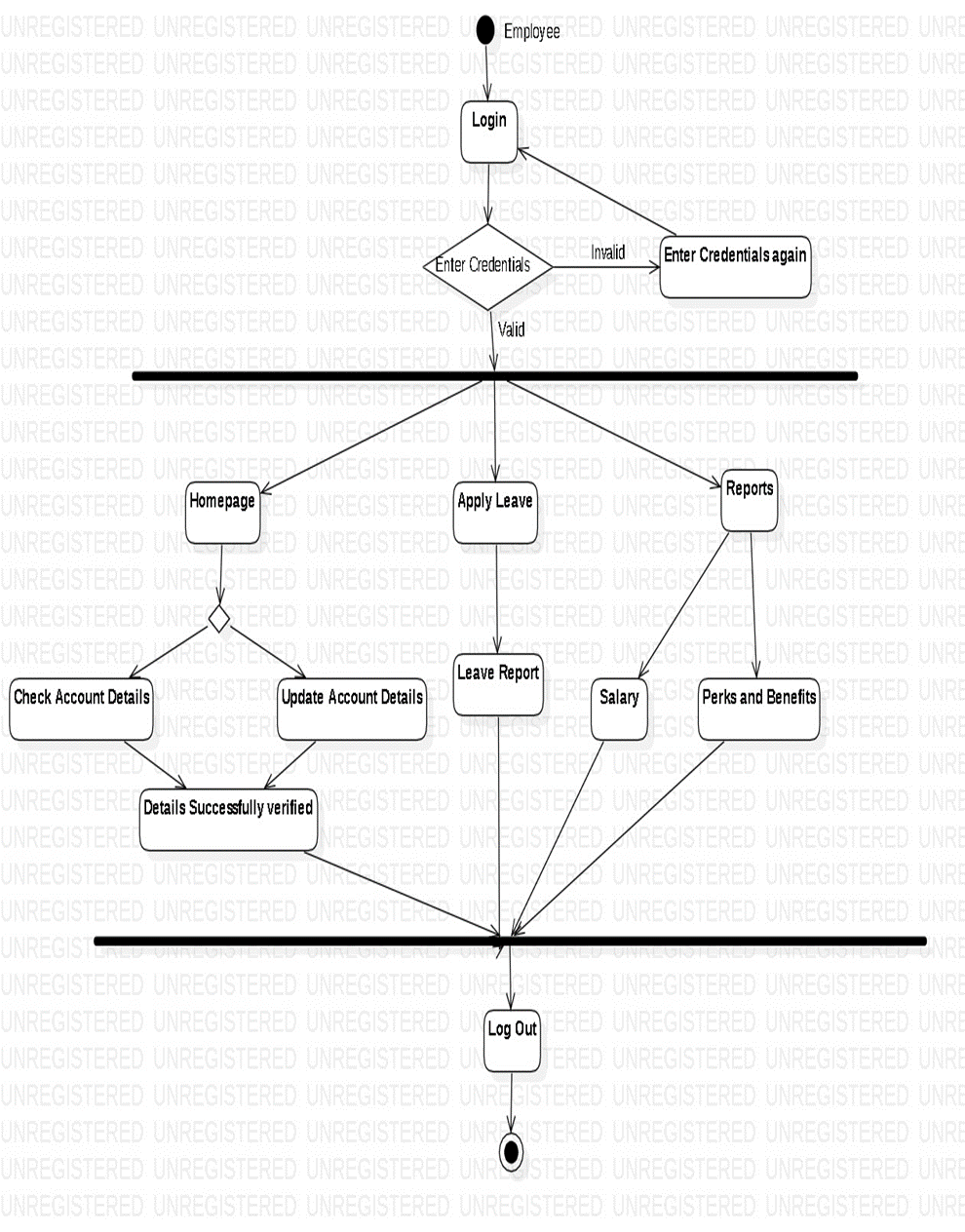
**Fig 1.1: ER model for Payroll Management System**

**2. Class Diagram**

****

**Fig 1.2: Class Diagram for Payroll Management System**

**3. Activity Diagram**

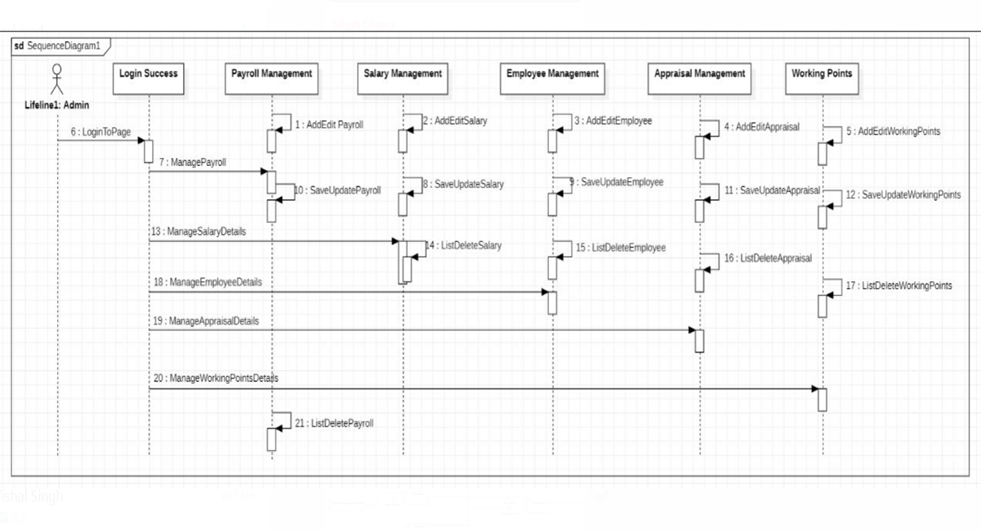
****

Diagram

Description automatically generated

**Fig 1.3: Activity Diagram for Payroll Management System**

**4. Sequence Diagram**

****

**Fig 1.4: Sequence Diagram for Payroll Management System**

**Appendix B: References**

* Wikipedia.org
* Jones and Bartlett Publishers - Essential SQL on SQL Server 2008, 2011
* Software Engineering by “Ian Sommerville”.
* “Payroll Management: A Comprehensive Guide to Payroll Compliance” by Steven M. Bragg.
* “The Payroll Source” by the American Payroll Association (APA).
* “Payroll Management: 2019 Edition” by Jean-Marc Baills and Steve Briggs.

**Lab 2**

**Aim-** To describe the activities that are to be conducted for development of the project Payroll Management System.

The activity table serves as a visual representation of the project's timeline and the tasks required to achieve its successful completion. It plays a crucial role in project management by providing a structured framework that guides the project team, allocates resources efficiently, and enables effective tracking of progress.

The development of the project - Payroll Management System involves several activities that are essential for its successful implementation. These activities are as follows:

**1. Project Initiation:** The project's purpose and scope are defined. Clearly outline what the payroll management system is expected to achieve. Identify stakeholders, including HR personnel, finance professionals, and IT staff. Establish communication channels to keep them informed throughout the project. Create a project team with roles and responsibilities. This may include project managers, developers, database administrators, and testers.

**2. Requirement Gathering:** This activity involves identifying the needs of the organization and employees, and determining the features and functionalities that the system should have. The requirements are gathered through interviews, surveys, and meetings with stakeholders. Document these requirements in a clear and organized manner. Use techniques like interviews, surveys, and workshops to collect information. Ensure that the team is aware of legal and regulatory compliance requirements for payroll processing in the jurisdiction. This includes tax laws, labor regulations, and reporting standards.

**3. Designing:** Once the requirements are gathered, the next step is to design the system. This involves creating a blueprint of the system, including the database schema, user interface design, and system architecture. The design should be detailed enough to ensure that the system meets the requirements. Design the user interface (UI) for the system. Consider creating separate interfaces for payroll administrators and employees. Develop data flow diagrams to illustrate how data moves through the system, from employee data input to payroll calculation and reporting. Define security measures, access controls, and encryption methods to protect sensitive payroll data.

**4. Development:** After the design is finalized, the development of the system can begin. This involves writing the code for the system, integrating the database, and testing the system for bugs and errors. The development should be done in a structured and organized manner to ensure that the system is reliable and efficient. Develop employee self-service portals for functions like updating personal information, accessing pay stubs, and requesting time off. Create admin dashboards for payroll administrators to manage employee records, process payroll, and generate reports. Implement proper coding standards and practices and conduct regular code reviews to ensure code quality.

**5. Testing:** Once the system is developed, it needs to be tested thoroughly to ensure that it meets the requirements and is free of errors. This involves unit testing, integration testing, and system testing. The testing should be done in a controlled environment to ensure that the system is stable and secure. Perform unit testing to validate individual components or functions within the system. Conduct integration testing to ensure that different modules of the system work seamlessly together. Engage in user acceptance testing (UAT) with stakeholders to ensure that the system meets their requirements and expectations. Identify and fix any bugs or issues discovered during testing.

**6. Deployment:** After the testing is complete, the system can be deployed to the production environment. This involves installing the system on the server, configuring it, and making it available to the users. The deployment should be done in a way that minimizes downtime and ensures that the system is accessible to the users. Ensure that the system is scalable to accommodate your organization's growing workforce. Implement backup and disaster recovery procedures to protect against data loss or system failures.

**7. Documentation:** Create user manuals and documentation for system administrators and end-users. This documentation should explain how to use the system effectively. Document the system architecture, database schema, and codebase for reference and future maintenance. Develop training materials to help end-users understand how to use the system efficiently.

**8. Training:** Provide training sessions for payroll administrators and end-users. These sessions should cover various aspects of the system, such as data entry, reporting, and troubleshooting. Offer ongoing support and training resources to address any questions or issues that may arise.

**9. Maintenance:** Once the system is deployed, it needs to be maintained to ensure that it continues to function properly. This involves fixing bugs, updating the system, and providing support to the users. The maintenance should be done in a timely and efficient manner to ensure that the system is always up-to-date and secure. Provide ongoing technical support to assist users with any problems or questions. Monitor system performance and security, and proactively address any issues that arise.

**10. Post-Implementation Review:** After the system is live, evaluate its performance and gather feedback from users and stakeholders. Identify areas for improvement and plan for future enhancements to the system. Ensure that the project has successfully met its objectives and requirements.

**11.** **Documentation Update and Knowledge Transfer:** Regularly update documentation to reflect any changes or improvements made to the system. Ensure that team members are knowledgeable about the system, even if they weren't involved in its initial development. Knowledge transfer is crucial for ongoing support and maintenance.

**12. Project Closure:** Complete all project documentation and reports, including a final project report summarizing the project's achievements and challenges. Conduct a project closure meeting with stakeholders to review the project's outcomes and obtain formal acceptance and sign-off.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Activity** | **Description** | **Dependencies** | **Start Date** | **End Date** | **Resources** | **Status** |
| Project Initiation | Start of the Project | None | 03-07-23 | 03-07-23 | Project Manager | Completed |
| Requirements Gathering | Gather user and system requirements | Project Initiation | 03-07-23 | 16-07-23 | Business Analyst | Completed |
| Designing | Create a high-level design of the system | Requirement Gathering | 16-07-23 | 31-07-23 | System Architect | Completed |
| Development | Develop the Project | Design | 31-07-23 | 21-08-23 | DevOps Team | Completed |
| Testing | Test the system for functionality and bugs | Development | 21-08-23 | 04-09-23 | QA Team  Users | Started |
| Deployment | Deploy the system for public use | Testing | 04-09-23 | 11-09-23 | DevOps Team | Not Started |
| Documentation | Create user and technical documentation | Development Testing | 16-07-23 | 03-10-23 | Tech Writer | In Progress |
| Training | Train users and support teams on the system | Deployment | 11-09-23 | 25-09-23 | Trainers | Not Started |
| Maintenance | Provide ongoing support and maintenance | Documentation Update & Knowledge Transfer | 10-10-23 | 26-10-23 | DevOps Team | Not Started |
| Post- Implementation Review | Review the Project | Training | 25-09-23 | 03-10-23 | DevOps Team | Not Started |
| Documentation Update & Knowledge Transfer | Update documentation & transfer them to others. | Post- Implementation Review | 03-10-23 | 10-10-23 | DevTeam  Tech Writer | Not Started |
| Project Closure | Close the Project | Maintenance | 26-10-23 | 27-10-23 | Project Manager | Not Started |

**Lab 3**

**Aim-** To draw the Program Evaluation and Review Technique(PERT) chart for the project Payroll Management System.

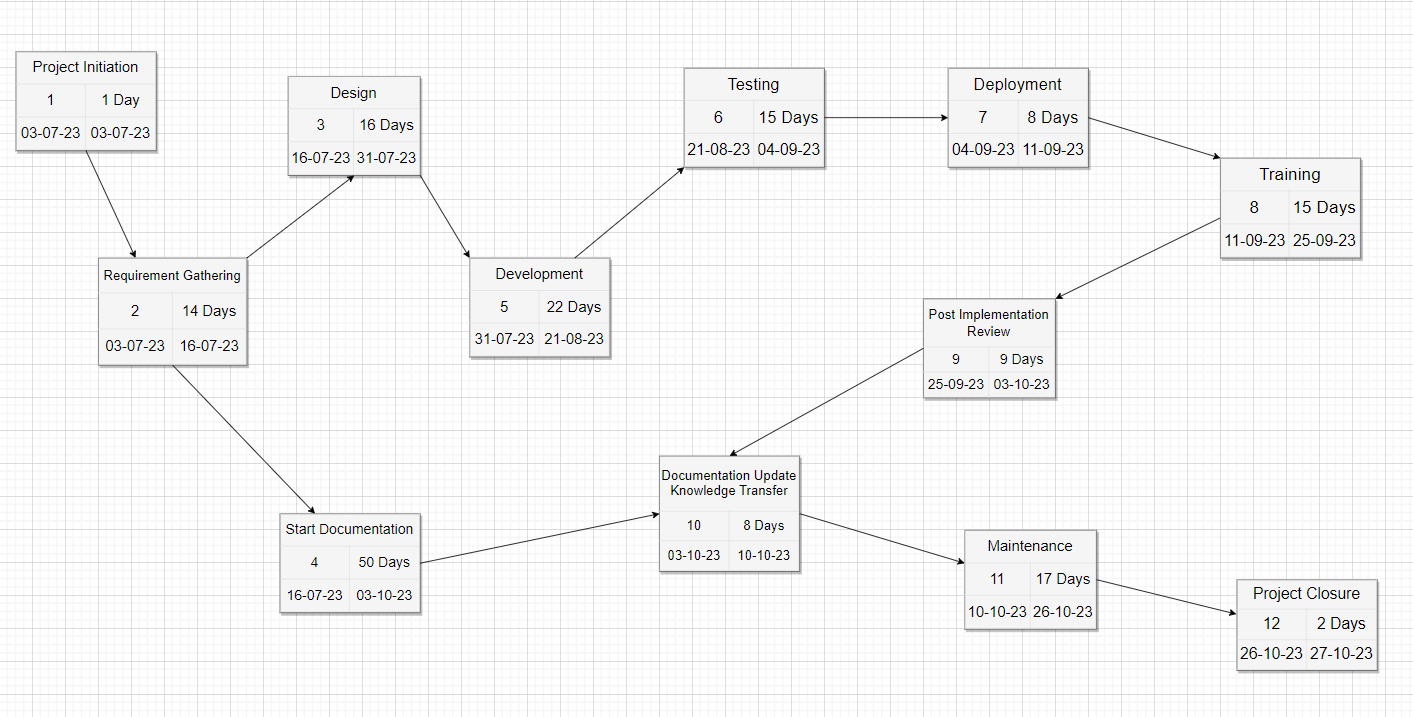


Fig 3.1: Pert Chart using draw.io

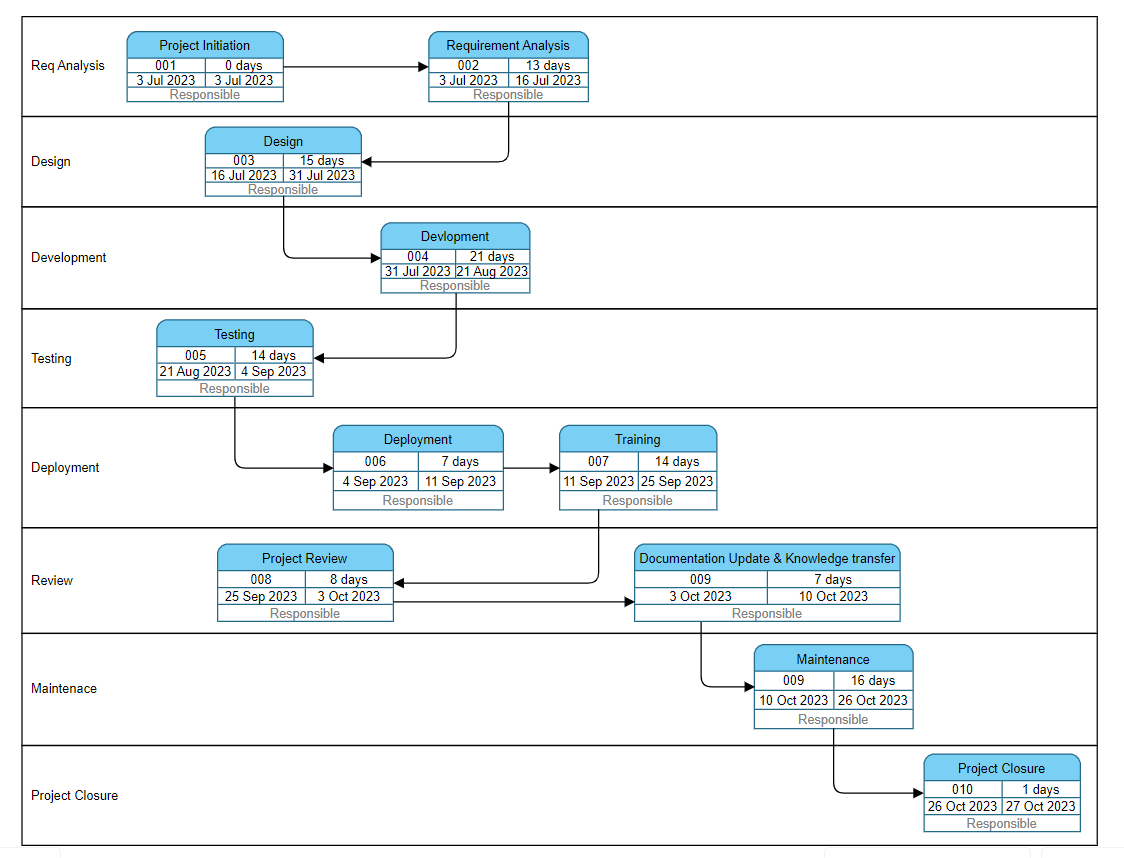
****

Fig 3.2: Pert Chart using Visual Paradigm

**1. Project Initiation (03-07-23 to 03-07-23) Total Days: 1**

* The project's purpose and scope are defined. Clearly outline what the payroll management system is expected to achieve.
* Identify stakeholders, including HR personnel, finance professionals, and IT staff.
* Establish communication channels to keep them informed throughout the project.
* Create a project team with roles and responsibilities.
* This may include project managers, developers, database administrators, and testers.

**2. Requirement Gathering (03-07-23 to 16-07-23) Total Days: 14**

* This activity involves identifying the needs of the organization and employees, and determining the features and functionalities that the system should have.
* The requirements are gathered through interviews, surveys, and meetings with stakeholders. Document these requirements in a clear and organized manner.
* Use techniques like interviews, surveys, and workshops to collect information.
* Ensure that the team is aware of legal and regulatory compliance requirements for payroll processing in the jurisdiction.

**3. Design (16-07-23 to 31-07-23) Total Days: 16**

* This involves creating a blueprint of the system, including the database schema, user interface design, and system architecture.
* The design should be detailed enough to ensure that the system meets the requirements. Design the user interface (UI) for the system.
* Consider creating separate interfaces for payroll administrators and employees.
* Develop data flow diagrams to illustrate how data moves through the system, from employee data input to payroll calculation and reporting. Define security measures, access controls, and encryption methods to protect sensitive payroll data.

.

**4. Documentation (16-07-23 to 03-10-23) Total Days: 50**

* Create user manuals and documentation for system administrators and end-users. This documentation should explain how to use the system effectively.
* Document the system architecture, database schema, and codebase for reference and future maintenance.
* Develop training materials to help end-users understand how to use the system efficiently.

**5. Development (31-07-23 to 21-08-23) Total Days: 22**

* This involves writing the code for the system, integrating the database, and testing the system for bugs and errors.
* The development should be done in a structured and organized manner to ensure that the system is reliable and efficient.
* Develop employee self-service portals for functions like updating personal information, accessing pay stubs, and requesting time off.
* Create admin dashboards for payroll administrators to manage employee records, process payroll, and generate reports.
* Implement proper coding standards and practices and conduct regular code reviews to ensure code quality.

**6. Testing (21-08-23 to 04-09-23) Total Days: 15**

* This involves unit testing, integration testing, and system testing.
* The testing should be done in a controlled environment to ensure that the system is stable and secure.
* Perform unit testing to validate individual components or functions within the system.
* Conduct integration testing to ensure that different modules of the system work seamlessly together.
* Engage in user acceptance testing (UAT) with stakeholders to ensure that the system meets their requirements and expectations.

**7. Deployment (04-09-23 to 11-09-23) Total Days: 8**

* This involves installing the system on the server, configuring it, and making it available to the users.
* The deployment should be done in a way that minimizes downtime and ensures that the system is accessible to the users.
* Ensure that the system is scalable to accommodate your organization's growing workforce.
* Implement backup and disaster recovery procedures to protect against data loss or system failures.

**8. Training (11-09-23 to 25-09-23) Total Days: 15**

* Provide training sessions for payroll administrators and end-users.
* These sessions should cover various aspects of the system, such as data entry, reporting, and troubleshooting.
* Offer ongoing support and training resources to address any questions or issues that may arise.

**9. Post-Implementation Review (25-09-23 to 03-10-23) Total Days: 9**

* After the system is live, evaluate its performance and gather feedback from users and stakeholders.
* Identify areas for improvement and plan for future enhancements to the system. Ensure that the project has successfully met its objectives and requirements.

**10.** **Documentation Update & Knowledge Transfer (03-10-23 to 10-10-23) Total Days: 8**

* Regularly update documentation to reflect any changes or improvements made to the system.
* Ensure that team members are knowledgeable about the system, even if they weren't involved in its initial development.
* Knowledge transfer is crucial for ongoing support and maintenance.

**11. Maintenance** **(10-10-23 to 26-10-23) Total Days: 17**

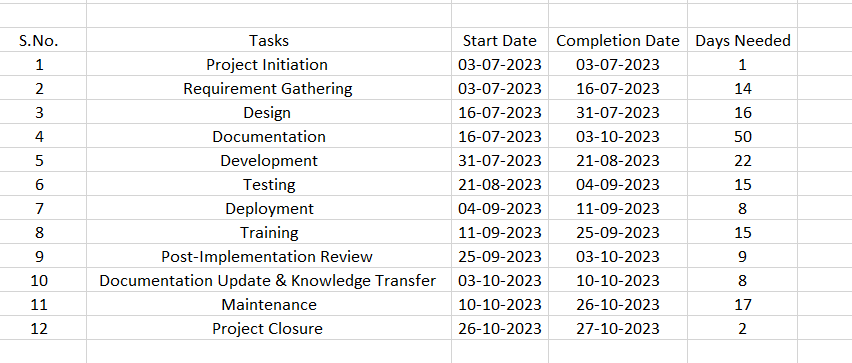
* This involves fixing bugs, updating the system, and providing support to the users.
* The maintenance should be done in a timely and efficient manner to ensure that the system is always up-to-date and secure.
* Provide ongoing technical support to assist users with any problems or questions. Monitor system performance and security, and proactively address any issues that arise.

**12. Project Closure (26-10-23 to 27-10-23) Total Days: 2**

* Complete all project documentation and reports, including a final project report summarizing the project's achievements and challenges.
* Conduct a project closure meeting with stakeholders to review the project's outcomes and obtain formal acceptance and sign-off.

**Lab 4**

**Aim-** To draw the GANTT chart for the project Payroll Management System.



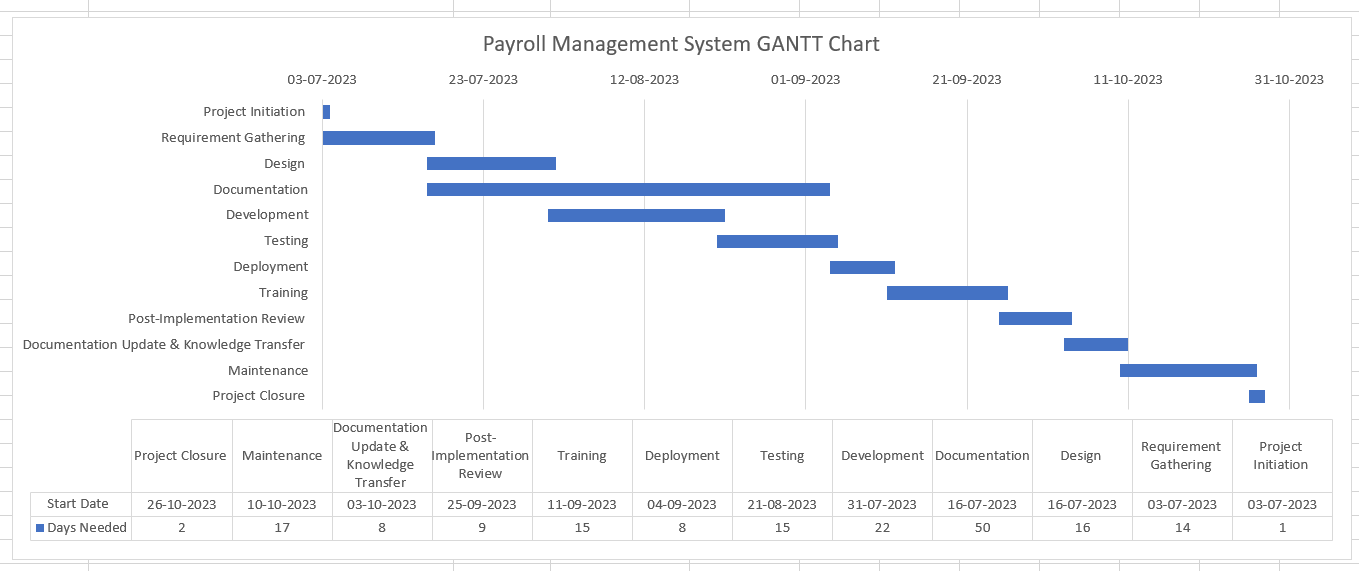


Fig 4.1: GANTT Chart

**Lab 6**

**Aim-** To implement Critical path method for Payroll Management System.

**Theory**

The Critical Path Method (CPM) is a project management technique used to plan, schedule, and manage complex projects. It is particularly useful for projects with a large number of activities that need to be coordinated to ensure the project is completed on time. CPM helps project managers identify the most critical tasks that can impact the overall project timeline and allows them to focus their efforts on managing those tasks.

1. Task Identification: The first step is to identify all the tasks or activities required to complete the project. Each task should have a clear description, a duration estimate, and a list of dependencies (tasks that must be completed before it can start).
2. Network Diagram: Once all the tasks are identified, they are arranged in a network diagram, also known as a CPM network or PERT (Program Evaluation and Review Technique) chart. This diagram shows the sequence of tasks and their dependencies.
3. Duration Estimation: Each task is assigned a duration estimate, which represents the amount of time required to complete it. These estimates can be based on historical data, expert judgment, or other relevant information.
4. Duration Estimation: Each task is assigned a duration estimate, which represents the amount of time required to complete it. These estimates can be based on historical data, expert judgment, or other relevant information.
5. Critical Path Identification: The critical path is the longest sequence of dependent tasks that determines the overall duration of the project. Tasks on the critical path have zero slack or float, meaning any delay in these tasks will directly impact the project's completion date. It's crucial to identify the critical path because it helps project managers focus on the most time-sensitive activities.
6. Slack Analysis: Tasks that are not on the critical path may have some slack or float, meaning they can be delayed without affecting the project's overall duration. Project managers can use slack analysis to identify tasks that can be delayed optimizing resource allocation and project scheduling.
7. Schedule Development: With the critical path and slack analysis completed, a project schedule is developed. This schedule outlines when each task should start and finish to ensure the project's timely completion.
8. Monitoring and Control: Throughout the project, project managers continuously monitor the progress of tasks and compare it to the planned schedule. If any tasks on the critical path are delayed, it may necessitate corrective actions to bring the project back on track.

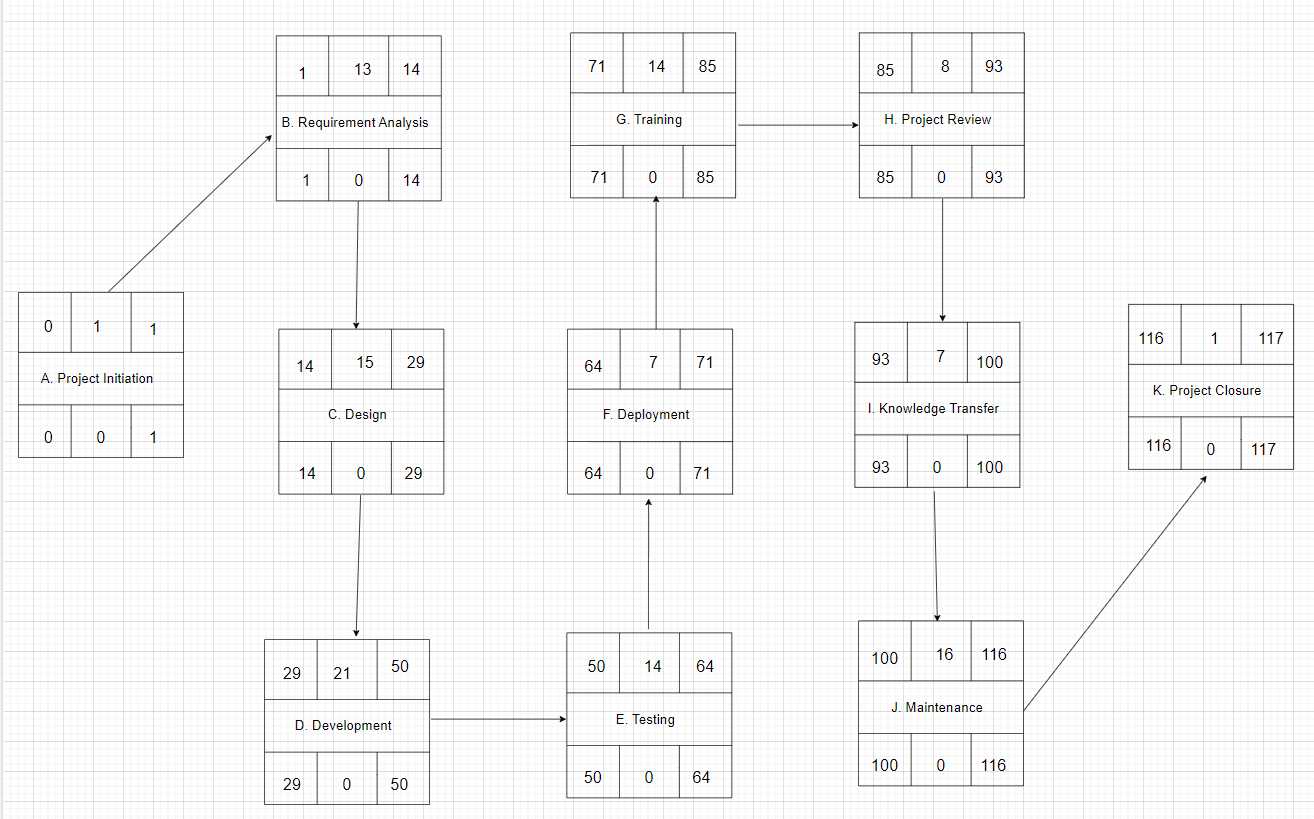


Fig 6.1 CPM Diagram using Activity on Node Method

Critical Path- A, B, C, D, E, F, G, H, I, J, K

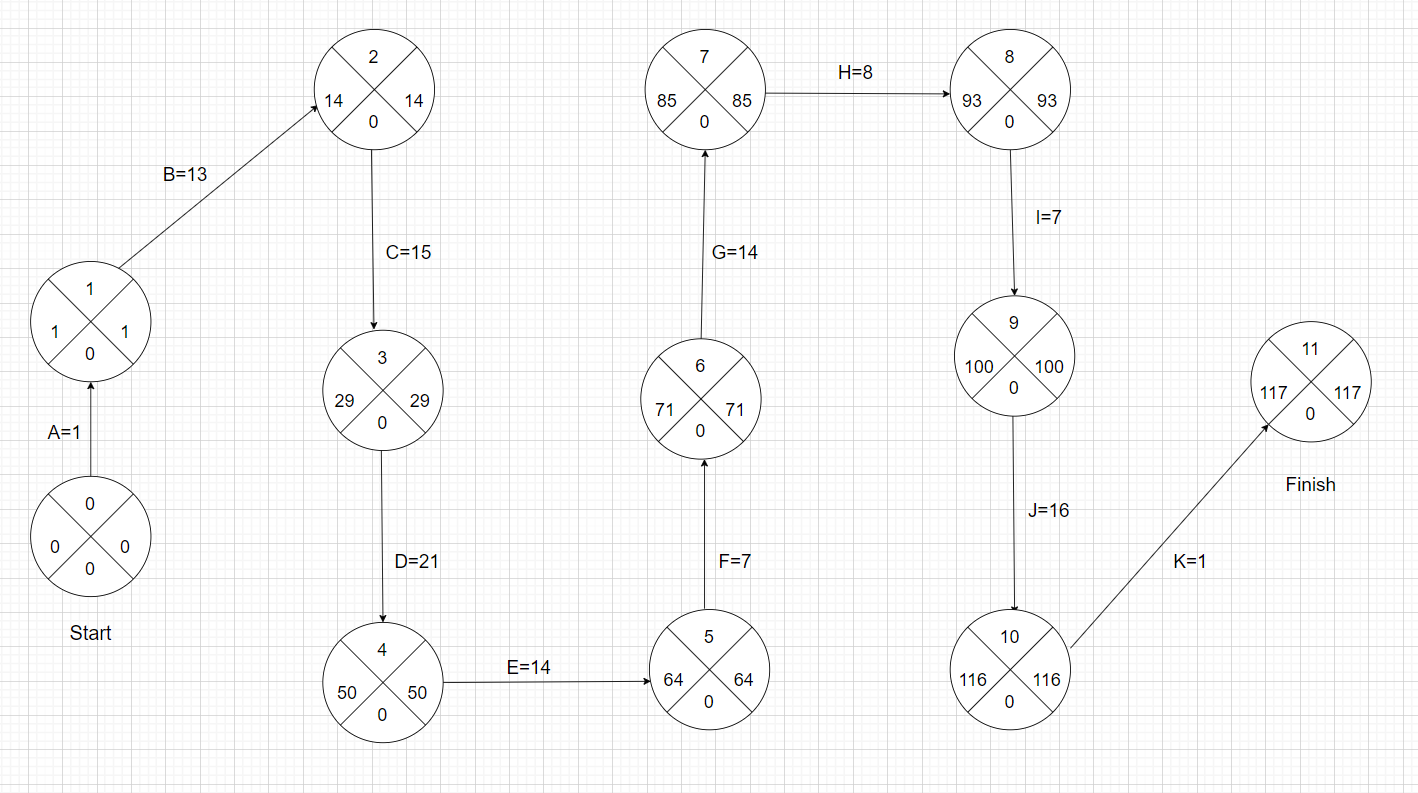


Fig 6.2 CPM Diagram using Activity on Arrow Method

Critical Path- A, B, C, D, E, F, G, H, I, J, K

**Lab 7**

**Aim-** To design the workflow diagram for the Payroll Management System.

A workflow diagram, also known as a workflow chart, is a visual representation of a business process, project, or job in the form of a flowchart. It provides a graphic overview of the process, showing step by step how work is completed from start to finish, and who is responsible for work at each point in the process.

Workflow diagrams use standardized symbols and shapes to depict the various tasks and steps needed to complete the process. They are commonly used for project planning, business process mapping, and business process modeling, but can have other applications in industries like manufacturing and engineering.

Workflow diagrams are beneficial to project management because they help team members better understand the task sequences in which they’re involved, create better communication between departments, and give teams a firm grasp on what they have to do. Workflow diagrams can also be used to identify and fix weak points within a long-standing process or workflow, and to isolate and repair inefficiencies and eliminate roadblocks.

A diagram of a diagram

Description automatically generated

Fig 7.1 Workflow Diagram for Payroll Management System

**Lab 8**

**Aim-** To design Slip chart, Timeline chart and Ball chart for Payroll Management System.

**Slip chart-** is a type of visual progress chart used in project management to show the overall progress of a given project over time. Slip charts are a line graph that plots out tasks, milestones, and events against time, allowing project managers to see which tasks have been completed, if any tasks are slipping behind, or if any tasks are ahead of schedule.

Slip charts are a visual indication of activities that are not progressing to schedule. They are an alternative view of a Gantt chart by providing a visual indication of those activities which are not on schedule. Slip charts are a simple but effective progress report where milestones are plotted on a grid to show when they are scheduled to occur. The more the slip line bends, the greater the variation from the plan.

Additional slip lines can be included at regular intervals, and as they build up, the project manager will gain an idea as to whether the project is improving or not. Slip charts are also used in statistical process control (SPM) to represent how often a process produces parts that fail to meet certain quality standards.

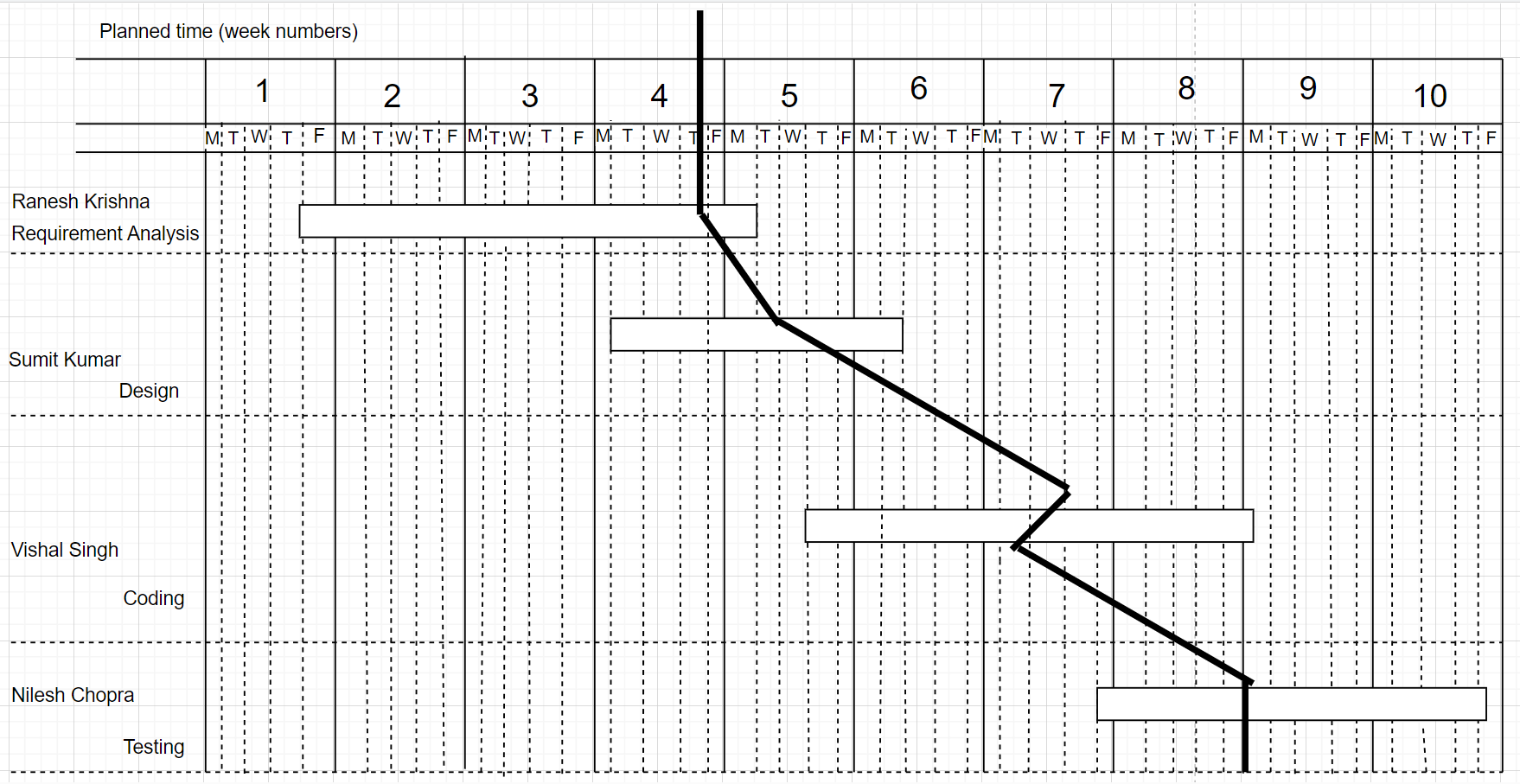


Fig 8.1 Slip Chart for Payroll Management System

**Timeline chart-** is a visual representation of a set of events or processes arranged chronologically. It is a simple yet powerful diagram that can be used to track projects to completion, illustrate historical events, or conceptualize event sequences or processes. Timeline charts typically include dates and descriptions, and some may also include images and headers.

There are several different types of timeline charts available, including event timelines, periodical history timelines, brand historical timelines, and AI growth timelines. Timeline charts are a valuable tool for organizations that need a concise way to visualize a process or event chronologically, and they can help manage complex tasks and ensure they're completed on time.

Some of the benefits of using timeline charts include providing a clear overview of what needs to be done and when, making it easier to allocate resources at the right time, and serving as a decision-making aid for current and future projects.

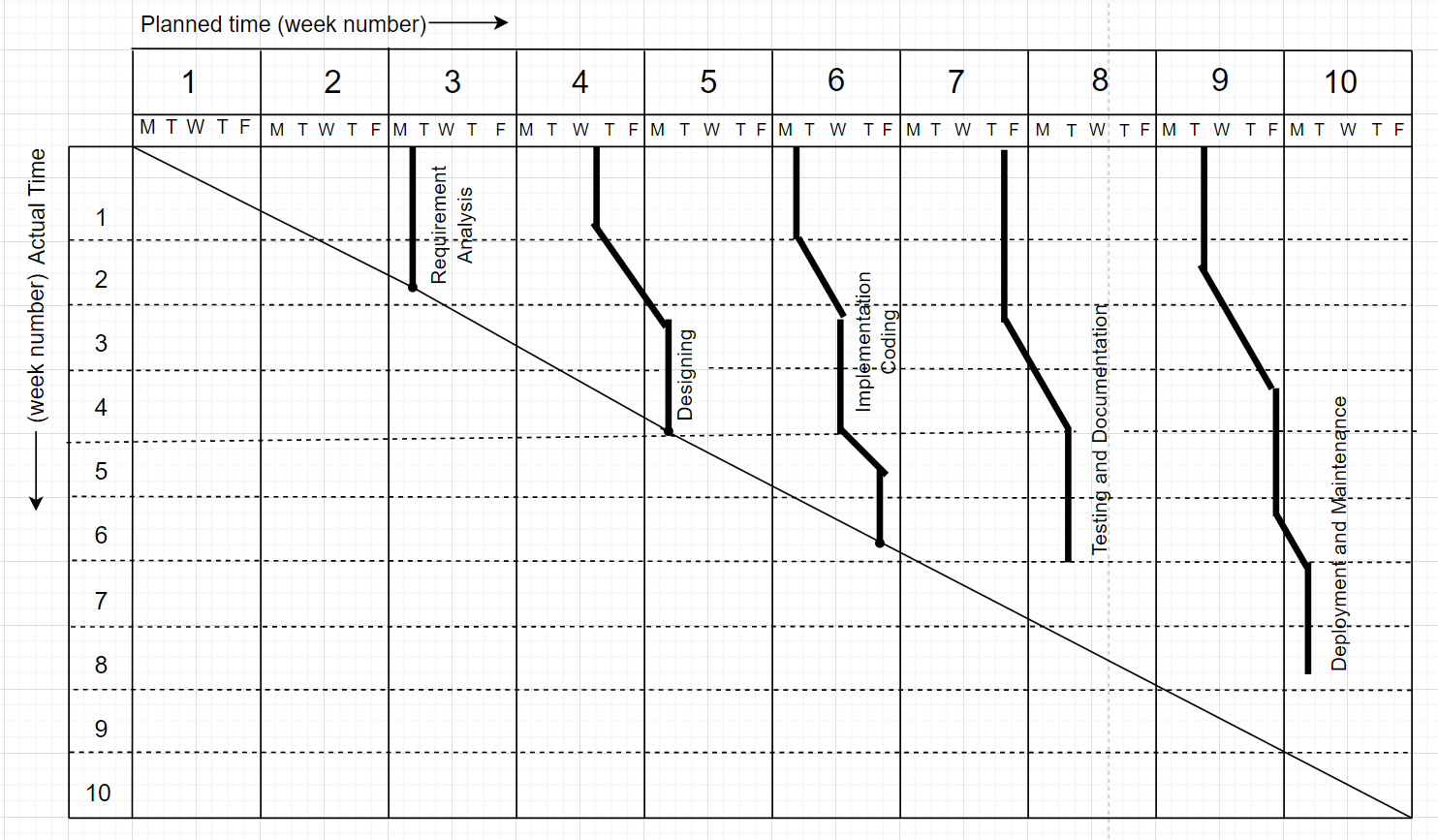


Fig 8.2 Timeline Chart for Payroll Management System

**Ball chart-** also known as a Harvey ball chart, is a type of chart that uses circles to represent data. It is a visual representation of qualitative information such as ease of use, efficiency, safety, taste, or quality. Ball charts circles contain original scheduled dates and the actual dates the activity/event took to complete.

They are also used in project management for project tracking, in lean manufacturing for value-stream mapping and continuous improvement tracking, and in business process modeling software for visualization. In a ball chart, the circles can indicate the start or end of a process, and they can be colored or filled to represent different values or categories. The chart consists of multiple circular shapes, with each colored quadrant representing a certain set of data.

The colored segment usually represents the data or criterion that has been achieved or completed, while the non-colored or blank segment represents the data or criterion to be achieved. Harvey balls are a powerful tool for visualizing different types of data, and they can be used to show different types of data in a single slide or page. However, they may not be the best choice for visualizing large amounts of data, as they can become confusing and difficult to distinguish between each set of data.

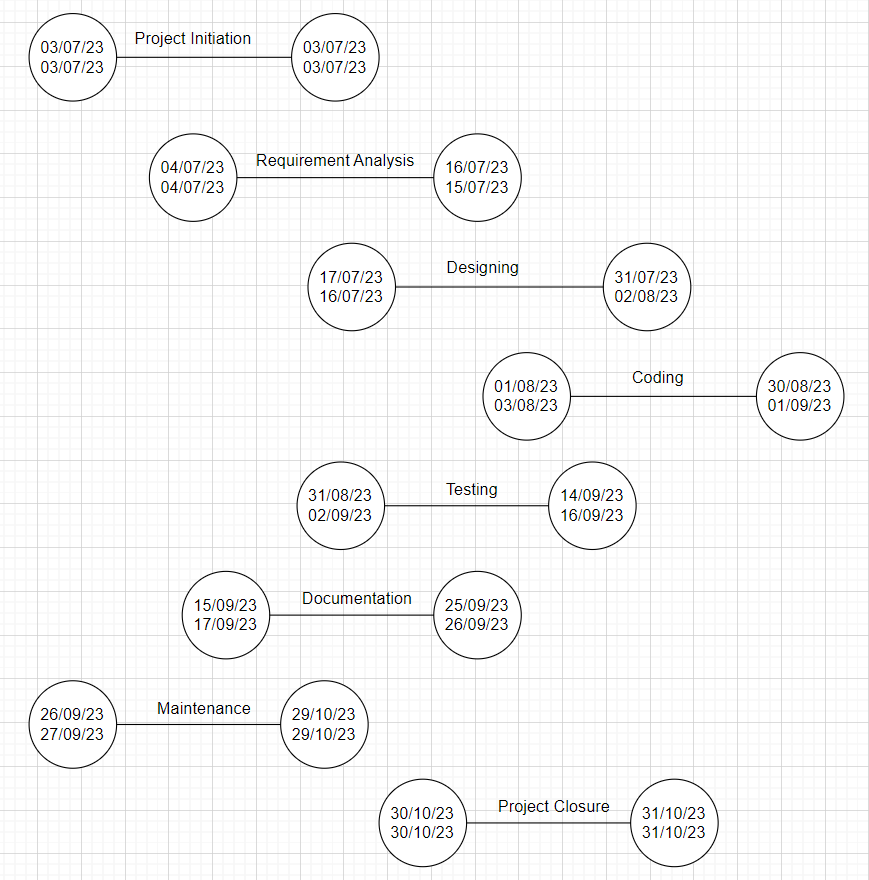


Fig 8.3 Ball Chart for Payroll Management System