**School of Computer Science**

**UNIVERSITY OF PETROLEUM AND ENERGY STUDIES**

**DEHRADUN, UTTARAKHAND**



**Software Engineering**

**&**

**Project Management Lab**

**Lab File**

**(2023-2024)**

**for**

**4rd Semester**

**Submitted To: Submitted By:**

Dr Ram Kumar Akshat Negi

Assistant Professor-SS Btech CSF [4rd Semester]

School of Computer Science 500106533

Batch 2

**INDEX**

|  |  |  |
| --- | --- | --- |
| 1 | Exp1 | Functional and Non-functional |
| 2 | Exp2 | Data Flow Diagram (DFD) and Structured chart. |
| 3 | Exp3 | Use case diagram. |
| 4 | Exp4 | Class diagram, object diagram. |
| 5 | Exp5 | Sequence diagram, Collaboration diagram |
| 6 | Exp6 | SRS of online ride booking service |
| 7 | Exp7 | Perform Estimation of effort using FP estimation for chosen system. |
| 8 | Exp8 | Perform Estimation of effort using FP estimation for chosen system |
| 9 | Exp9 | Create a project plan, dashboard and reports using Microsoft project form any selected projects. |
| 10 | Exp 10 | Gantt Chart |

**Experiment 1**

Perform requirement analysis and to find the requirement specification (both functional and nonfunctional) of a given Problem.

**FUNCTIONAL REQUIREMENTS:**

1. Patient Management:

- Registration: The system should allow for the registration of new patients, including capturing their personal details, contact information, and medical history.

- Appointment Scheduling: Patients should be able to schedule appointments with doctors, and the system should manage these appointments efficiently.

- Admission and Discharge: Facilitate the admission and discharge process for patients, including bed allocation and discharge summaries.

2. Doctor Management:

- Profile Management: Maintain profiles for doctors, including their specialties, contact details, and availability.

- Appointment Management: Allow doctors to view their appointments, reschedule if necessary, and manage their availability.

3. Medical Records Management:

- Electronic Health Records (EHR): Store and manage patient medical records securely, including diagnosis, treatment plans, medications, lab results, and imaging reports.

- Record Retrieval: Provide quick access to patient records for healthcare providers as needed during consultations or treatments.

4. Billing and Invoicing:

- Billing and Invoicing: Generate bills for services provided, including consultations, procedures, medications, and other healthcare services.

- Insurance Management: Handle insurance information for patients, including verification, claims processing, and coordination of benefits.

5. Inventory Management:

- Medication and Supply Management: Keep track of medications, medical supplies, and equipment stock levels, and automate reordering processes when inventory levels are low.

6. Laboratory and Imaging Management:

- Test Orders and Results: Facilitate the ordering of laboratory tests and imaging studies, as well as the recording and retrieval of results.

- Integration: Integrate with laboratory and imaging systems for seamless transfer of data and results.

7. Reporting and Analytics:

- Reporting: Generate reports on various aspects of hospital operations, including patient demographics, financial performance, resource utilization, and quality metrics.

- Analytics: Provide analytical tools to identify trends, patterns, and areas for improvement in patient care and operational efficiency.

8. Security and Compliance:

- Data Security: Ensure the confidentiality, integrity, and availability of patient information through robust security measures, including user authentication and access controls.

- Compliance: Adhere to regulatory requirements such as HIPAA (Health Insurance Portability and Accountability Act) for patient data protection and privacy.

**NON-FUNCTIONAL REQUIREMENT:**

1. Performance:

- Response Time: The system should respond to user requests within an acceptable timeframe, ensuring smooth and efficient user interaction.

- Throughput: It should be able to handle a large number of concurrent users and transactions without significant degradation in performance.

- Scalability: The system should be scalable to accommodate growth in data volume, user base, and system load over time.

2. Reliability:

- Availability: The system should be available for use whenever required, with minimal downtime for maintenance or upgrades.

- Fault Tolerance: It should be resilient to failures, with mechanisms in place to detect and recover from errors gracefully to prevent data loss or service disruptions.

3. Security:

- Data Security: Ensure the confidentiality, integrity, and privacy of patient information through encryption, access controls, and secure transmission protocols.

- Authentication and Authorization: Implement robust mechanisms for user authentication and authorization to prevent unauthorized access to sensitive data and system functionalities.

- Audit Trails: Maintain detailed logs of user activities and system events for auditing and forensic purposes.

4. Usability:

- User Interface: The user interface should be intuitive, easy to navigate, and visually appealing, catering to users with varying levels of technical expertise.

- Accessibility: Ensure that the system is accessible to users with disabilities, complying with accessibility standards and guidelines.

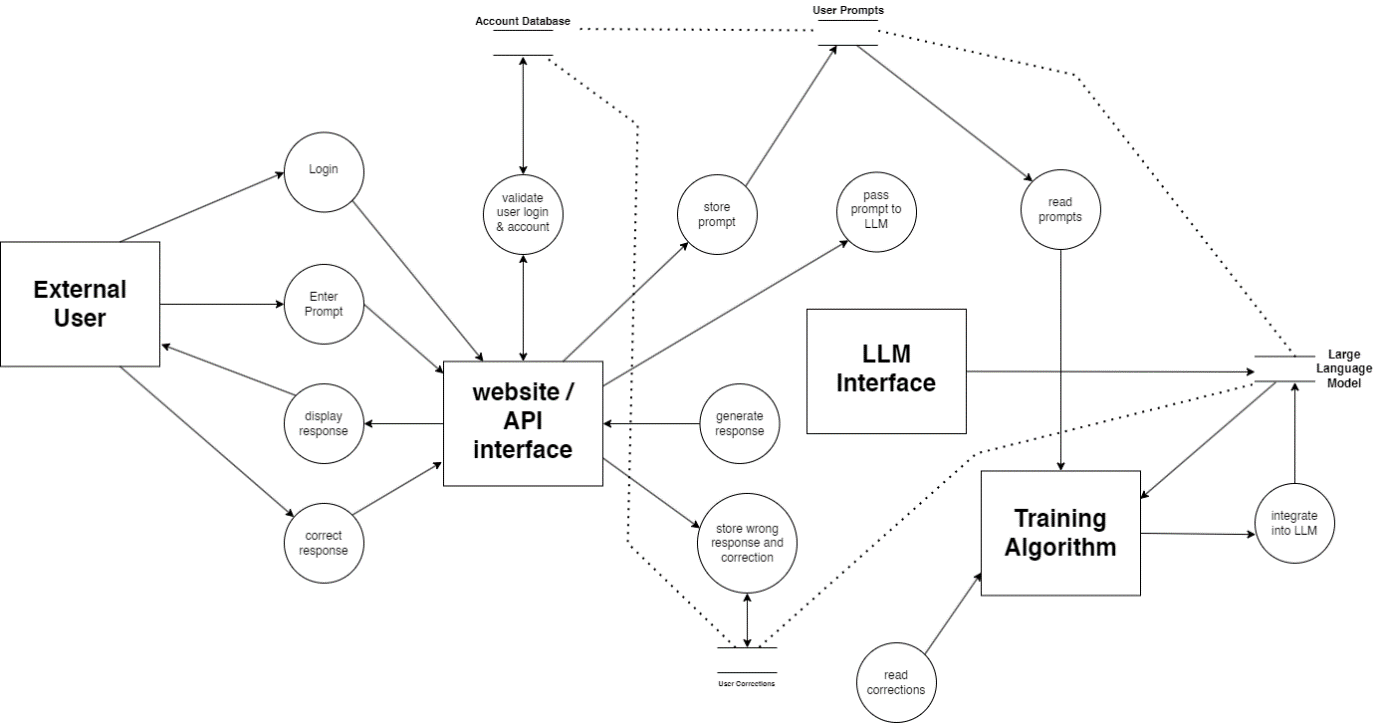
5. Maintainability:

- Modularity: Design the system with modular components that can be easily maintained, updated, or replaced without affecting the overall functionality.

- Documentation: Provide comprehensive documentation, including system architecture, configuration instructions, and troubleshooting guides, to support system maintenance and administration.

**Experiment 2**

To perform the function-oriented diagram: Data Flow Diagram (DFD) and Structured chart.

****

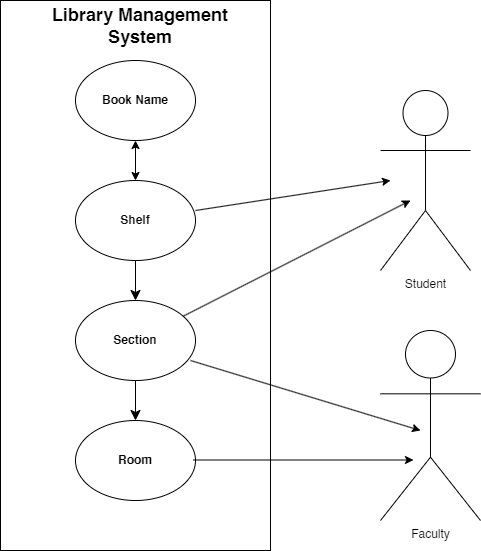
**Experiment 3**

To perform the user ‘s view analysis for the suggested system: Use case diagram.

**What is a use case diagram?**

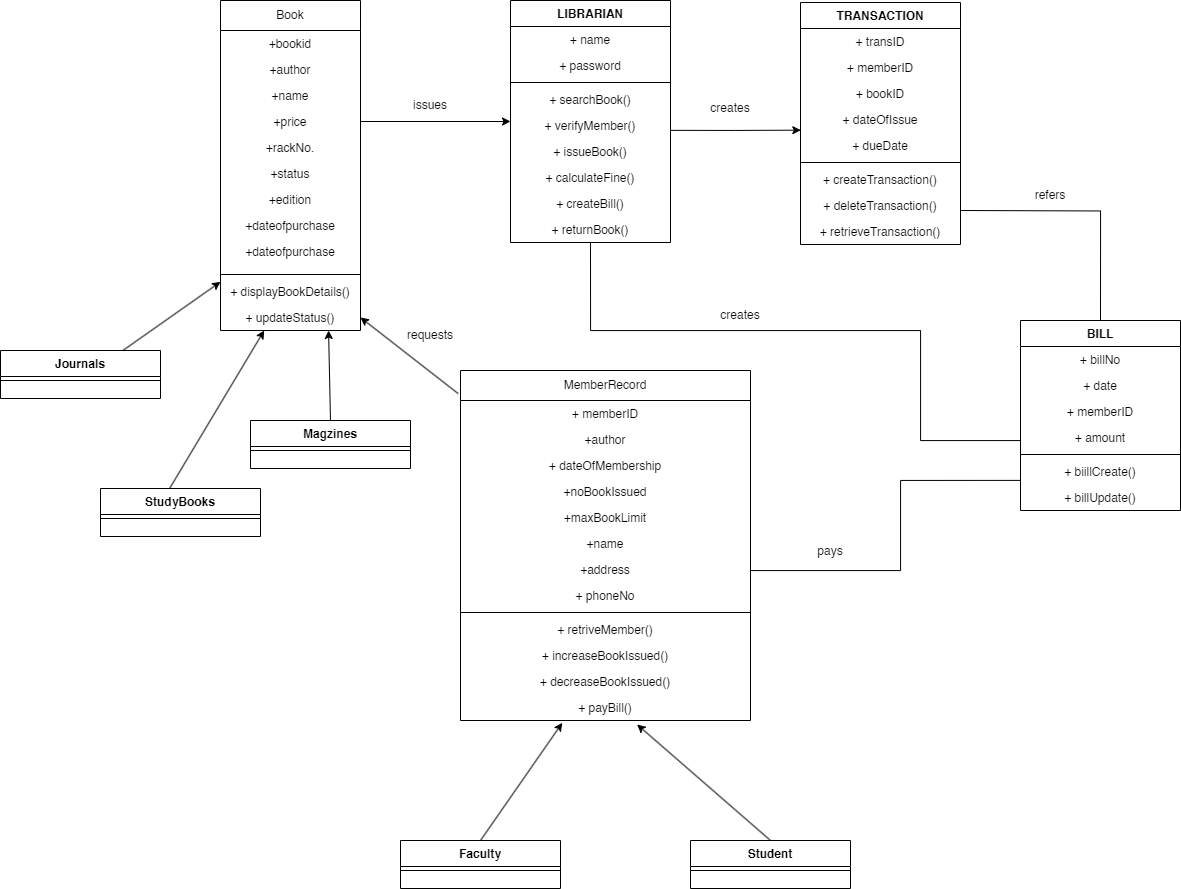
• In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system.

• Use case diagram depicts a high-level overview of the relationship between use cases, actors, and systems.

•

**Experiment 4**

To draw the structural view diagram for the system: Class diagram, object diagram.



**Experiment 5**

To perform the behavioral view diagram for the suggested system: Sequence diagram, Collaboration diagram

**What is an interaction diagram?**

Interaction diagrams are used in UML to establish communication between objects. They do not manipulate the data associated with the particular communication path. Interaction diagrams mostly focus on message passing and how these messages make up one functionality of a system. Interaction diagrams are designed to display how the objects will realize the particular requirements of a system. The critical component in an interaction diagram is lifeline and messages. Various UML elements have their own interaction diagrams. The details of the interaction can be shown using several annotations, such as the sequence diagram, timing diagram, and communication/collaboration diagram. Interaction diagrams capture the dynamic behavior of any system.

**The different types of interaction diagrams defined in UML:**

**• Sequence diagram**

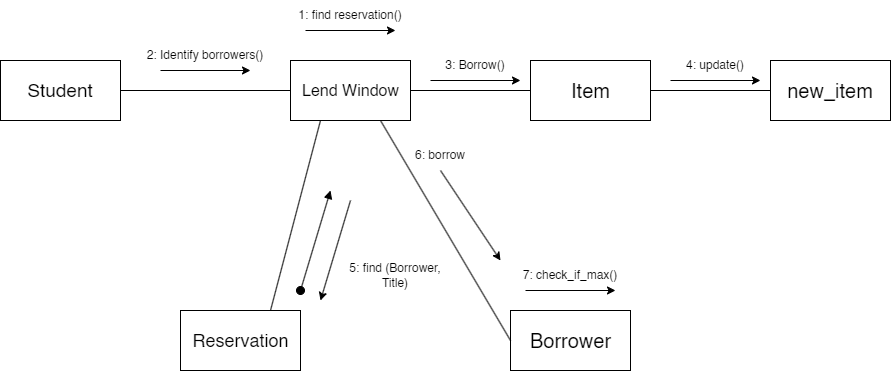
**• Collaboration diagram**

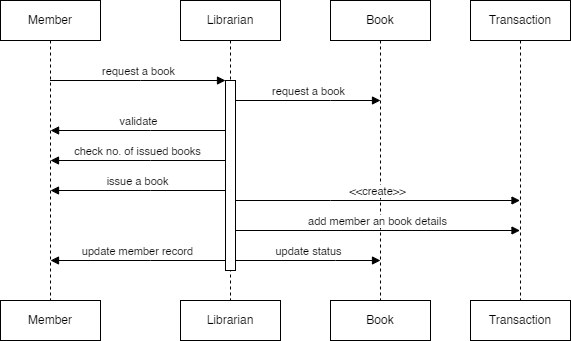
**• Timing diagram**

• The purpose of a sequence diagram in UML diagrams to visualize the sequence of a message flow in the system. The Sequence Diagram in Software Engineering shows the interaction between two lifelines as a time-ordered sequence of events.

• The Collaboration Diagram in UML is also called a communication diagram. The purpose of a collaboration diagram is to emphasize structural aspects of a system, i.e., how various lifelines in the system connects.

• Collaboration Diagram depicts the relationships and interactions among software objects. They are used to understand the object architecture within a system rather than the flow of a message as in a sequence diagram. They are also known as “Communication Diagrams.”





**Experiment 6**

**SRS FOR LIBRARY MANAGEMENT SYSTEM**

**1. Introduction**

The LMS is a software application designed to automate and streamline the management of library resources and services. It will cater to the needs of librarians, patrons, and administrators.

**2. Overall Description**

The LMS will provide a centralized platform for managing the following functionalities:

**Resource Management:**

* *Add, edit, and delete information about books, periodicals, and other library materials.*
* *Track inventory levels and availability of resources.*
* *Categorize resources by subject, genre, or other relevant criteria.*
* *Patron Management:*
* *Register new patrons and manage their accounts.*
* *Track borrowing history and fines.*
* *Set borrowing limits based on patron type (student, faculty, etc.).*

**Circulation Management:**

* *Issue and return library materials.*
* *Renew loans.*
* *Generate automated overdue notices and fines.*
* *Manage holds and waiting lists for popular resources.*

**Reporting and Analytics:**

* *Generate reports on resource usage, borrowing trends, and patron activity.*
* *Provide insights for collection development and service improvement.*

**3. Specific Requirements**

**3.1 Functional Requirements**

**User Roles and Permissions:**

* *The system shall differentiate between users with varying roles (librarian, patron, administrator).*
* *Each role shall have specific permissions to access and manage functionalities.*

**Resource Management:**

* *The system shall allow adding resources with details like title, author, ISBN, publication date, etc.*
* *The system shall support uploading cover images or descriptions for resources.*
* *The system shall allow searching and filtering resources based on various criteria.*

**Patron Management:**

* *The system shall allow registering patrons with information like name, contact details, and membership type.*
* *The system shall allow searching and editing patron information.*

**Circulation Management:**

* *The system shall allow librarians to check out resources to patrons by scanning barcodes or searching the database.*
* *The system shall track due dates and automatically calculate fines for overdue returns.*
* *The system shall allow patrons to renew loans online or at the library.*
* *The system shall notify patrons of upcoming due dates and overdue fines.*

**Reporting and Analytics:**

* *The system shall generate reports on resource usage statistics (borrows, renewals, etc.).*
* *The system shall generate reports on patron activity and identify popular resources.*
* *The system shall allow exporting reports in various formats (PDF, CSV).*

**3.2 Non-Functional Requirements**

**Performance:**

* *The system shall respond to user queries promptly.*
* *The system shall be able to handle concurrent access by multiple users.*

**Security:**

* *The system shall protect sensitive user information (patron data, borrowing history) with secure access controls.*
* *The system shall encrypt data transmission to prevent unauthorized access.*

**Availability:**

* *The system shall be highly available with minimal downtime.*
* *The system shall have mechanisms for data backup and disaster recovery.*

**Usability:**

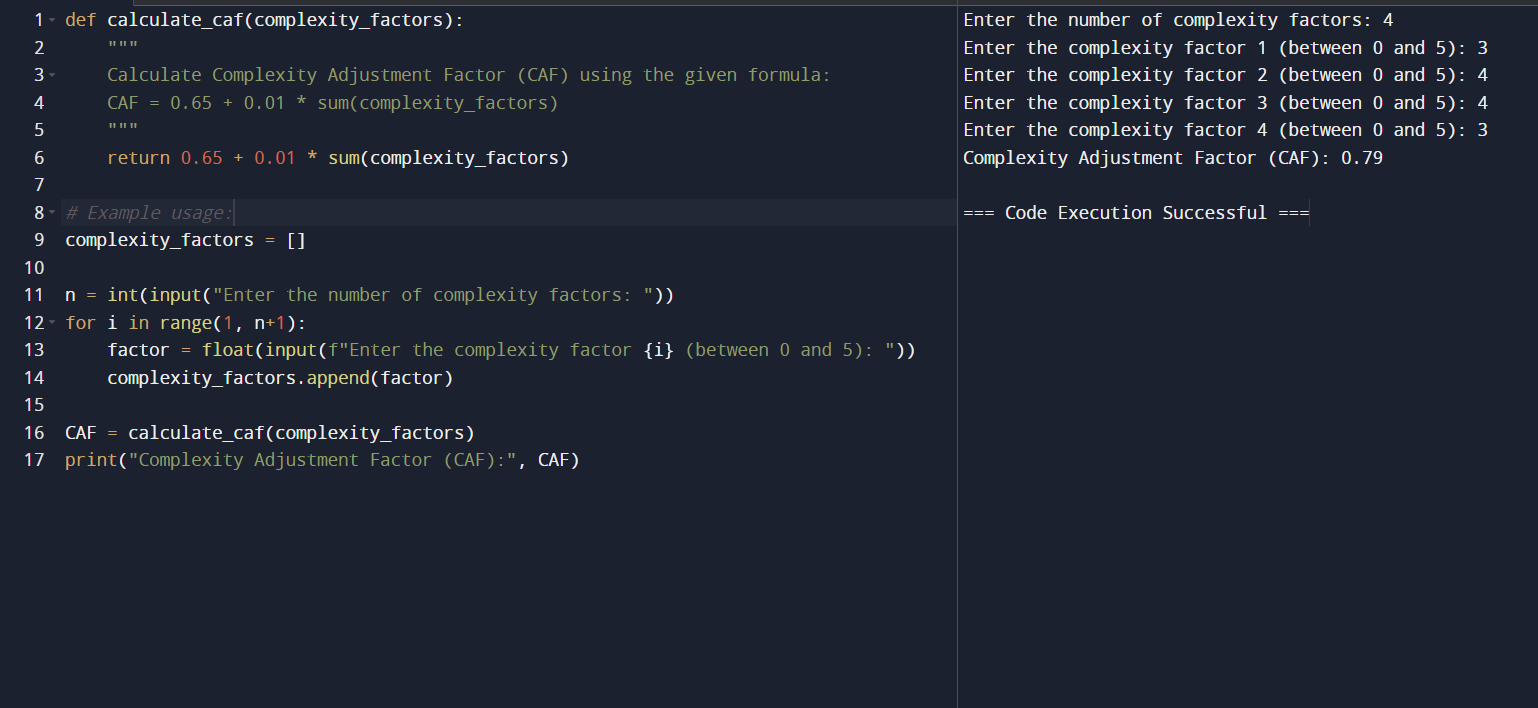
* *The system interface shall be user-friendly and intuitive for librarians and patrons with varying technical skills.*
* *The system shall provide online help and tutorials for user guidance.*

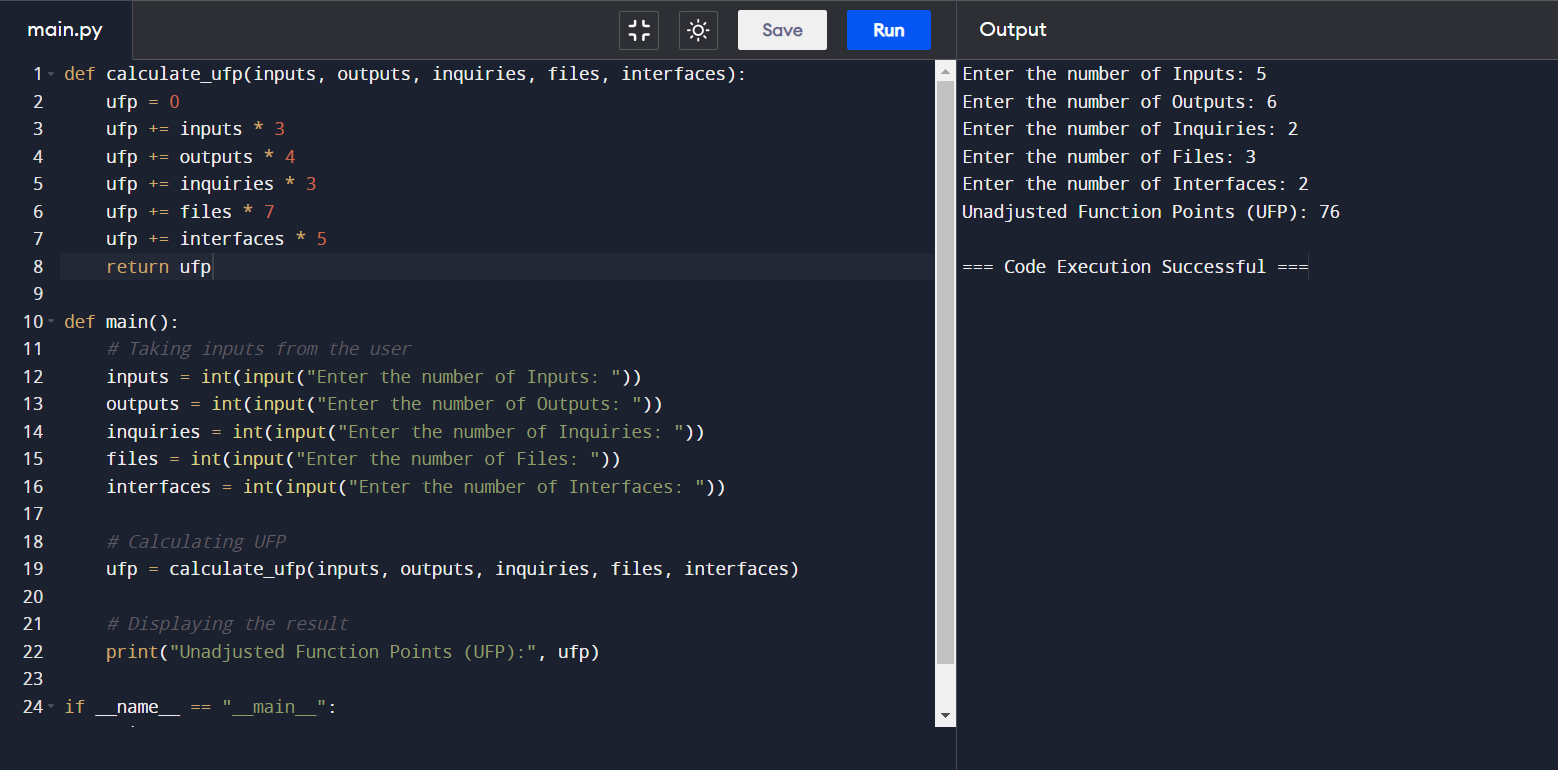
**4. System Interfaces**

* *The LMS will have a graphical user interface (GUI) accessible through a web browser.*
* *The system may also provide a mobile interface for patron convenience (optional).*
* *The system shall integrate with barcode scanners for efficient resource check-in/out.*

**Experiment 7**

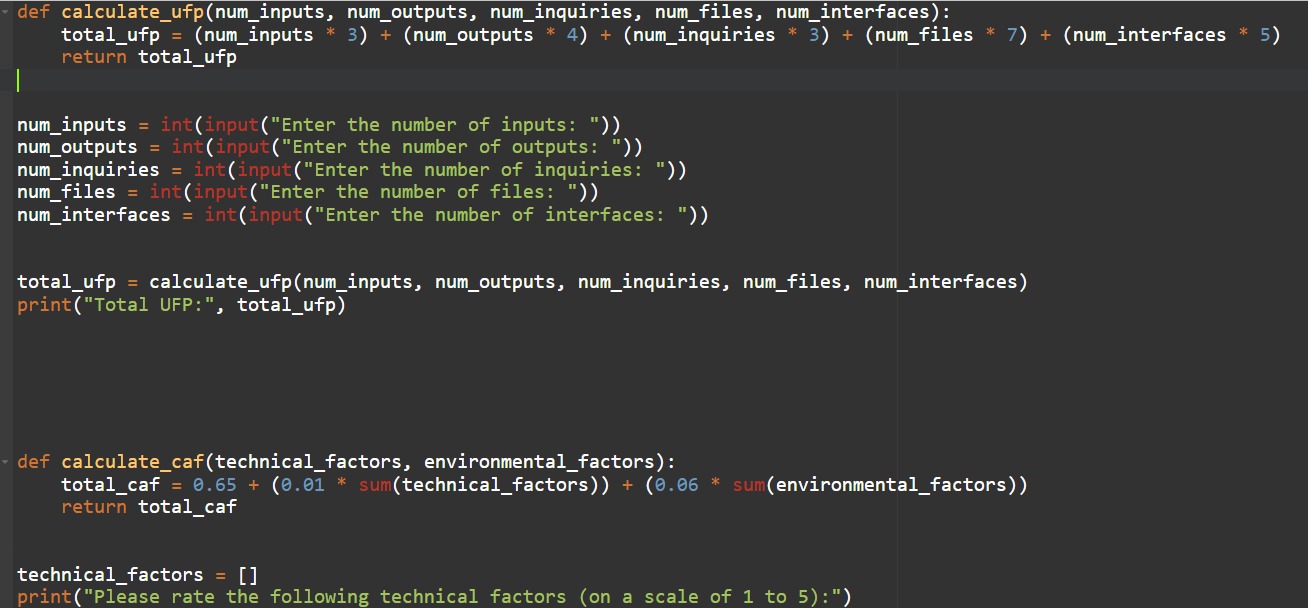
Perform Estimation of effort using FP estimation for chosen system.

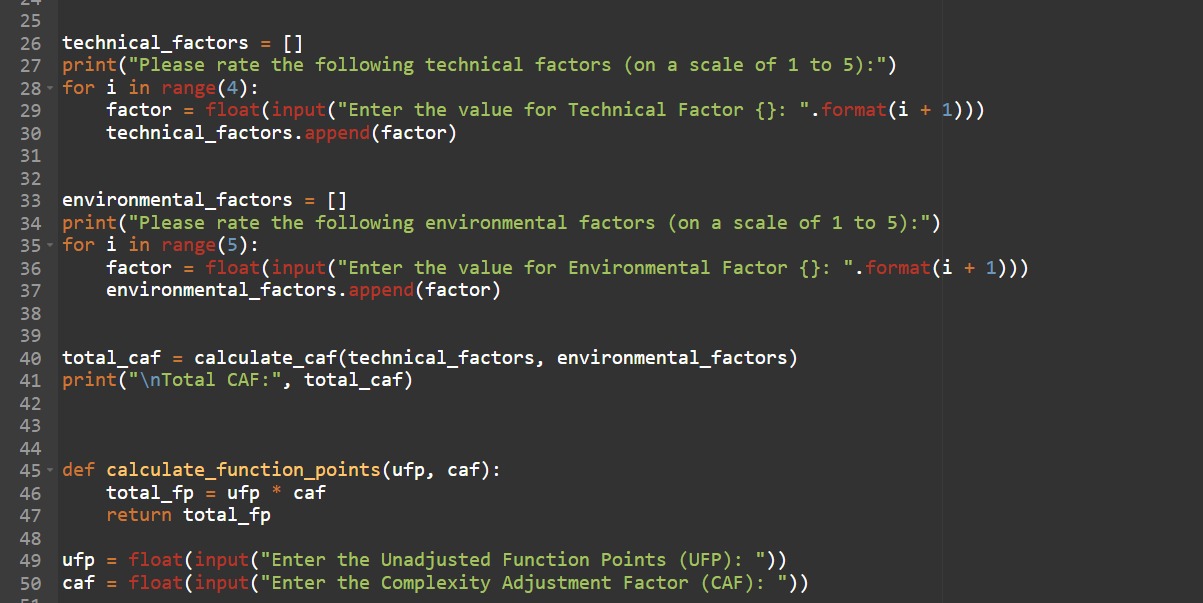


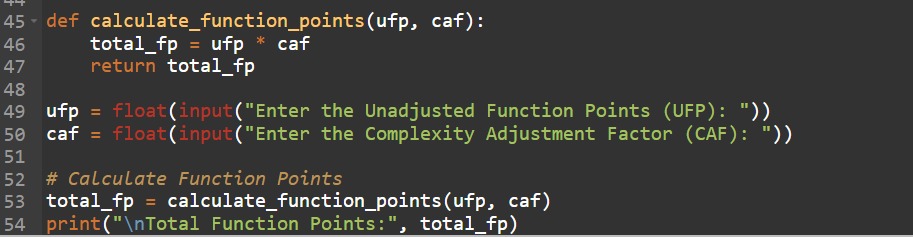


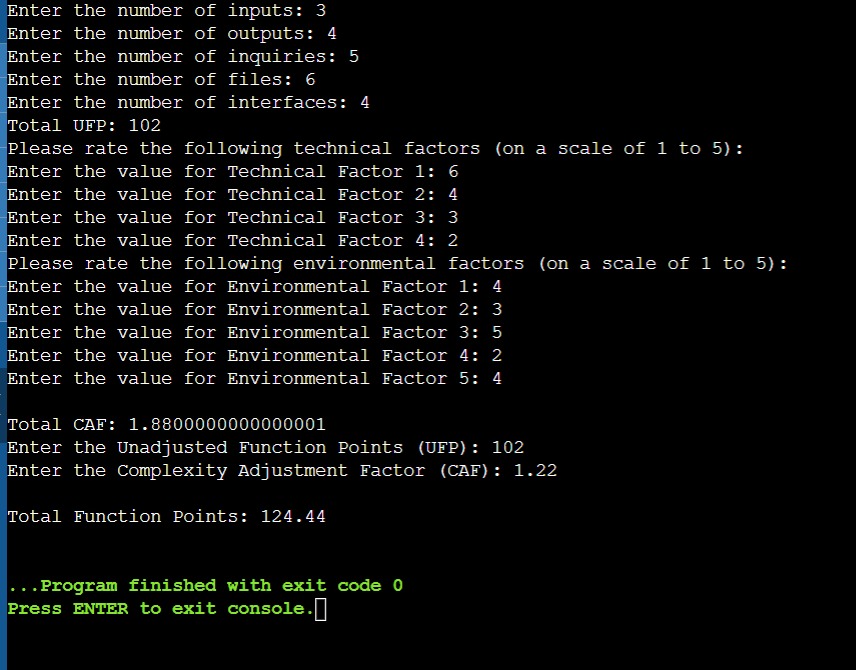
**Experiment 8**

Perform Estimation of effort using FP estimation for chosen system









**Experiment 9**

Create a project plan, dashboard and reports using Microsoft project form any selected projects.

Project Plan:

Initiation Phase:

Define project scope, objectives, and stakeholders.

Conduct a feasibility study.

Create a project charter.

Planning Phase:

Develop a Work Breakdown Structure (WBS).

Define activities, dependencies, and durations.

Allocate resources (human, financial, and material).

Create a project schedule.

Identify risks and develop a risk management plan.

Establish communication channels and reporting mechanisms.

Execution Phase:

Develop the online ride booking platform.

Implement features such as user registration, ride booking, payment processing, etc.

Conduct user testing and feedback sessions.

Train staff and users on how to use the app.

Monitoring and Controlling Phase:

Track project progress against the baseline schedule and budget.

Manage changes and address issues as they arise.

Monitor quality and ensure adherence to specifications.

Conduct regular meetings and status updates.

Closing Phase:

Perform project closure activities such as handing over deliverables to the operations team.

Conduct a post-implementation review.

Document lessons learned.

Dashboard:

Key Performance Indicators (KPIs):

Number of registered users.

Number of rides booked per day/week/month.

Revenue generated.

Customer satisfaction ratings.

Project Progress:

Gantt chart showing tasks completed vs. remaining.

Resource utilization.

Budget vs. actual expenses.

Risk Management:

Risks identified vs. risks mitigated.

Impact and probability matrix.

Quality Assurance:

Defect density.

Number of bugs resolved.

User acceptance testing status.

Reports:

Project Status Report:

Summary of project progress.

Accomplishments during the reporting period.

Issues and challenges encountered.

Action items for the next reporting period.

Resource Allocation Report:

Allocation of human and material resources.

Resource utilization rates.

Any resource constraints or bottlenecks.

Budget Report:

Comparison of budgeted vs. actual expenses.

Breakdown of costs by category (development, marketing, etc.).

Forecasted expenses for the remaining project duration.

Risk Management Report:

Overview of identified risks.

Mitigation strategies implemented.

Impact on project schedule and budget.

Quality Assurance Report:

Summary of testing activities.

Number of defects found and resolved.

User feedback and satisfaction ratings.

Using Microsoft Project Online:

Create a new project in Microsoft Project Online.

Input tasks, durations, dependencies, and resources.

Generate reports and dashboards using built-in templates or customize as needed.

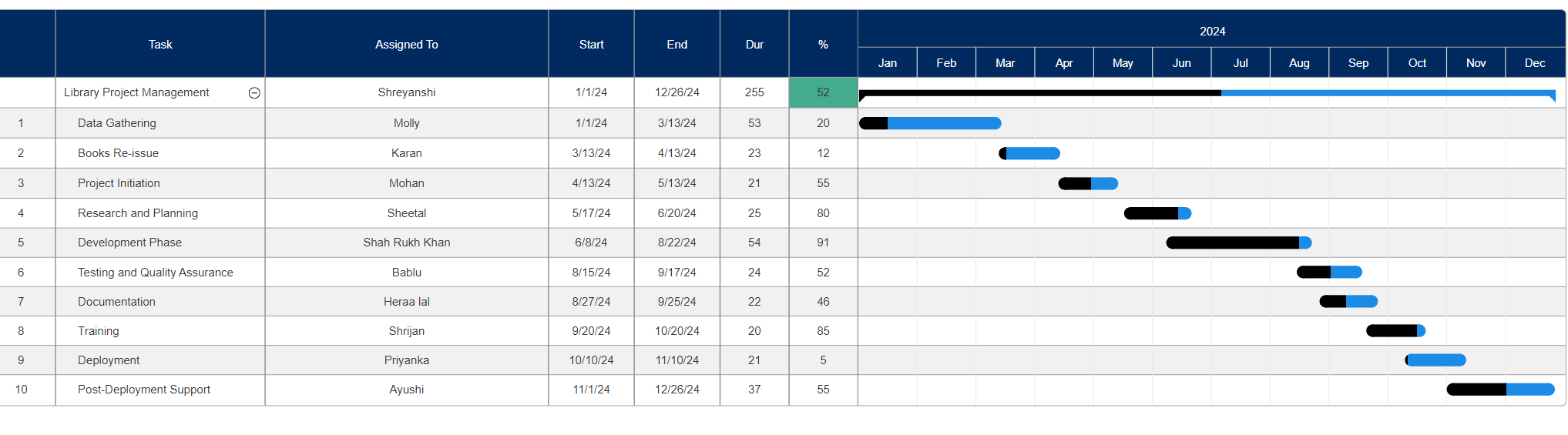
Share project updates with stakeholders and team members.

Collaborate in real-time using the online platform

**Experiment 10**

What is a Gantt chart?

A Gantt chart is a project management tool that illustrates work completed over a period of time in relation to the time planned for the work. It typically includes two sections: the left side outlines a list of tasks, while the right side has a timeline with schedule bars that visualize work. The Gantt chart can also include the start and end dates of tasks, milestones, dependencies between tasks, and assignees.

****