**CMP 403 TEST QUESTION AND ANSWERS**

**Q1. Compare and contrast between plan-driven /agile methods**

Comparison between Plan-Driven and Agile Methods:

Plan-Driven Methods:

1. **Sequential Approach:** Plan-driven methodologies, such as Waterfall, follow a sequential and structured approach to software development. Each phase is completed before moving on to the next.
2. **Upfront Requirements:** Requirements are typically gathered and documented comprehensively at the beginning of the project, with the aim of defining the project scope in detail.
3. **Rigidity:** The entire project scope, timeline, and budget are determined upfront and are less flexible to changes. Changes often require a formal change request process.
4. **Emphasis on Documentation:** Extensive documentation, including project plans, design specifications, and test plans, is emphasized to provide a clear roadmap for the project.
5. **Late Testing:** Testing usually occurs in a separate phase after development is complete, often during the testing phase.
6. **Challenges with Changes:** Changes to requirements during the project may be difficult to accommodate, leading to potential delays and increased costs.

Agile Methods:

1. **Iterative and Incremental:** Agile methodologies, like Scrum or Kanban, prioritize adaptability and collaboration throughout the software development process. Work is done in short, time-boxed iterations.
2. **Flexible Requirements:** Requirements are often more flexible and can evolve as the project progresses. They are usually captured as user stories or features, and detailed requirements may emerge over time.
3. **Adaptability:** Agile projects work in iterations, delivering increments of working software in each iteration. This allows for regular reassessment and adaptation to changing needs.
4. **Lightweight Documentation:** Documentation is typically more lightweight, with an emphasis on delivering working software over comprehensive documentation.
5. **Continuous Testing:** Testing is integrated throughout the development process, with a focus on continuous testing and quality assurance.
6. **Embracing Changes:** Collaboration, customer feedback, and changes to requirements are encouraged and expected throughout the project, allowing for responsiveness to changing requirements and market conditions.

Contrast:

* Plan-driven methods follow a more rigid and linear approach, while Agile methods are adaptive and iterative.
* Plan-driven methods aim to define all requirements upfront, whereas Agile methods embrace changing requirements throughout the project.
* Plan-driven methods prioritize comprehensive documentation, while Agile methods focus on delivering working software.
* Testing in plan-driven methods occurs late in the project, while Agile methods promote continuous testing.
* Plan-driven methods may struggle to accommodate changes, potentially leading to delays and increased costs, whereas Agile methods are designed to handle change effectively.

**Q2. Write the requirements for the four software projects**

**Employee Management System (EMS):**

**User Requirements:**

1. Users (Admins and Employees) should be able to access the system through secure authentication.
2. Admins should have the capability to add and edit employee profiles.
3. Employees should be able to check their leave status, qualifications, promotion history, and yearly holiday list.
4. Employees should be able to submit grievances and resignation requests through the system.

**System Requirements:**

1. The system must have secure user authentication mechanisms.
2. It should support role-based access control for Admins and Employees.
3. The system should capture and store employee attendance data, including images and GPS location.
4. It should provide features for leave management.
5. The system should have a database to store employee information and payroll data.

**Functional Requirements:**

1. Admins should be able to add and modify employee details, including name, job position, and qualification.
2. Admins should be able to access and modify salary details and breakdown for each employee.
3. Employees should have a dashboard to check their leave status, qualifications, and promotion history.
4. The system should support the submission and tracking of grievances and resignations.

**Non-Functional Requirements:**

1. The system should ensure data security and privacy.
2. It should be responsive and user-friendly.
3. It must have high availability to support 24/7 access.
4. The system should generate reports for payroll and attendance data.

**Payroll Management System:**

**User Requirements:**

1. Admins should be able to log in and track employee attendance and payroll.
2. Employees should have access to their own attendance and payroll details.

**System Requirements:**

1. The system should capture and store employee attendance data, images, and GPS locations.
2. It should support secure user authentication for Admins and Employees.
3. The system should provide real-time attendance tracking.
4. It should enable Admins to manage employee profiles and salary details.

**Functional Requirements:**

1. Admins should be able to track attendance data in real-time.
2. Admins should be able to add and modify employee profiles.
3. Employees should be able to access their attendance and payroll information.

**Non-Functional Requirements:**

1. The system should ensure data security and privacy.
2. It should have real-time processing capabilities for attendance data.
3. The system should be highly available.
4. It should be responsive and user-friendly.

**Fingerprint-Based ATM Card:**

**User Requirements:**

1. Users should be able to log in using their fingerprint for authentication.
2. Users should be able to perform various banking transactions, including cash withdrawal, money transfer, and balance inquiries.

**System Requirements:**

1. The system should capture and verify user fingerprints for authentication.
2. It should support secure user PINs for additional security.
3. The system should allow various banking transactions and account balance checks.
4. It should store and retrieve the last 5 transactions for each user.

**Functional Requirements:**

1. Users should be able to log in using their fingerprint and PIN.
2. Users should be able to conduct various banking transactions.
3. The system should display the last 5 transactions for users.

**Non-Functional Requirements:**

1. The system should ensure high security and data privacy.
2. It should provide fast and reliable fingerprint recognition.
3. It should be available 24/7.

**Android Local Ticketing System:**

**User Requirements:**

1. Users should be able to log in using Admin or User accounts.
2. Users should have access to a ticket booking form.
3. Admins should be able to recharge user account balances and view processed tickets.

**System Requirements:**

1. The system should support secure user authentication for Admin and User accounts.
2. It should provide a user-friendly ticket booking form.
3. The system should store user account balances and ticket data.

**Functional Requirements:**

1. Users should be able to select source, destination, travel class, and journey type in the booking form.
2. Users should be able to book and receive online tickets.
3. Admins should be able to recharge user account balances and view processed tickets.

**Non-Functional Requirements:**

1. The system should ensure data security and privacy.
2. It should be user-friendly and responsive.
3. The system should be available 24/7 for ticket booking and account management.