Lab – 1

Scenario: Online Food Delivery Platform

You are tasked with developing an online food delivery platform similar to popular services like Uber Eats or DoorDash. The platform will allow users to browse restaurants, place orders, and have food delivered to their location. Your team consists of a project manager, developers, and potential users representing different demographics and preferences.

Lab Activities:

1. **Introduction**: Explain the project goals and the importance of understanding user requirements for a successful online food delivery platform.
2. **Identify Stakeholders**: Discuss the various stakeholders involved, including customers, restaurants, delivery drivers, and administrators.
3. **Requirement Elicitation Techniques**: Introduce techniques such as interviews, surveys, and observations for gathering user requirements. Discuss how each technique can be applied to understand the needs of different stakeholders.
4. **Group Exercise**: Divide students into groups, with each group representing a different stakeholder (e.g., customers, restaurants, delivery drivers). Have them use various elicitation techniques to gather requirements specific to their stakeholder group.
5. **Documentation**: Instruct each group to document the user requirements they gathered, including features, functionalities, and any specific preferences or pain points identified.
6. **Validation and Verification**: Discuss methods for validating and verifying the gathered requirements to ensure they align with the project goals and are feasible within the given constraints.
7. **Presentation and Feedback**: Have each group present their findings, including the user requirements they gathered and how they prioritized them based on importance and feasibility. Encourage peer feedback and discussion.
8. **Reflection and Discussion**: Facilitate a discussion on the challenges encountered during the lab and strategies for addressing them in real-world projects, such as dealing with conflicting requirements or unclear user preferences.
9. **Wrap-Up**: Summarize the key takeaways from the lab, emphasizing the importance of considering user needs throughout the software development lifecycle for building successful and user-centric applications.

User Requirements for Online Food Delivery Platform:

1. **Customer Requirements:**
    - Ability to browse restaurants based on cuisine, location, and ratings.
    - Detailed restaurant menus with prices, descriptions, and images.
    - Option to customize orders (e.g., special instructions, dietary restrictions).
    - Secure and easy payment options (e.g., credit/debit cards, digital wallets).
    - Real-time order tracking and delivery updates.
    - Ability to rate and review restaurants and delivery experiences.
    - Loyalty programs or discounts for frequent orders.
2. **Restaurant Requirements:**
    - User-friendly interface for managing menu items, prices, and availability.
    - Integration with restaurant POS systems for real-time order processing.
    - Ability to set delivery zones, minimum order amounts, and delivery fees.
    - Notification system for new orders and order status updates.
    - Access to customer reviews and ratings for feedback and improvement.
    - Analytics dashboard for tracking sales, popular items, and customer demographics.
    - Seamless communication with delivery drivers for order coordination.
3. **Delivery Driver Requirements:**
    - Mobile app for receiving and managing delivery assignments.
    - GPS navigation for efficient route planning and navigation.
    - Real-time updates on order assignments, pick-up, and delivery status.
    - Ability to contact customers or restaurants for order-related queries.
    - Integration with payment systems for tracking earnings and tips.
    - Performance metrics and incentives for timely and accurate deliveries.
    - Support for multiple delivery platforms to maximize earnings.
4. **Administrative Requirements:**
    - Centralized dashboard for managing users, restaurants, and delivery drivers.
    - Access controls and permissions for different user roles (e.g., admins, moderators).
    - Tools for monitoring and resolving disputes, complaints, and quality issues.
    - Reporting and analytics features for tracking platform performance and trends.
    - Compliance with regulatory requirements and data protection laws.
    - Scalability and flexibility to accommodate growth and evolving business needs.

- Regular updates and maintenance to ensure system reliability and security.

By addressing these user requirements, the online food delivery platform can enhance user satisfaction, streamline operations, and drive business success. Additionally, thorough documentation and validation of these requirements will help ensure that the final product meets user expectations and fulfills its intended purpose effectively.

Now the work is yours:

Lab Example 1:

Scenario: Building a Task Management Application

You are part of a software development team tasked with building a task management application similar to Trello or Asana. The application should allow users to create, organize, and track tasks, assign tasks to team members, set deadlines, and monitor progress.

Lab Activities:

1. **Introduction:** Introduce the task management application project and the importance of understanding user requirements for its success.
2. **Identify Stakeholders:** Discuss stakeholders such as project managers, team members, administrators, and end-users, and their roles in defining user requirements.
3. **Requirement Elicitation Techniques:** Present various elicitation techniques such as interviews, surveys, and prototyping sessions for gathering user requirements.
4. **Group Exercise:** Divide students into groups representing different stakeholder roles (e.g., project managers, team members, end-users) and assign each group to use elicitation techniques to gather requirements.
5. **Documentation:** Instruct each group to document the user requirements they gathered in a structured format, such as user stories or a requirements document.
6. **Validation and Verification:** Discuss methods for validating and verifying user requirements to ensure their accuracy and feasibility.
7. **Presentation and Feedback:** Have each group present their findings and share the user requirements they gathered. Encourage peer feedback and discussion on the presented requirements.
8. **Reflection and Discussion:** Lead a reflection session where students discuss their experiences and insights gained from the lab exercise, emphasizing the importance of considering user needs in software development.

9. **Wrap-Up:** Summarize the key learnings from the lab and encourage students to apply the knowledge and skills gained in future software development projects.

Lab Example 2:

Scenario: Developing a Fitness Tracking Mobile App

You are part of a software development team tasked with creating a fitness tracking mobile application. The application should allow users to set fitness goals, track their workouts, monitor progress, and receive personalized recommendations based on their fitness level and preferences.

Lab Activities:

1. **Introduction:** Introduce the fitness tracking app project and the significance of gathering user requirements to create a user-friendly and effective application.
2. **Identify Stakeholders:** Discuss stakeholders such as fitness enthusiasts, personal trainers, nutritionists, and app developers, and their roles in defining user requirements.
3. **Requirement Elicitation Techniques:** Present various elicitation techniques such as surveys, focus groups, and usability testing for gathering user requirements.
4. **Group Exercise:** Divide students into groups representing different stakeholder roles (e.g., fitness enthusiasts, personal trainers, app developers) and assign each group to use elicitation techniques to gather requirements.
5. **Documentation:** Instruct each group to document the user requirements they gathered in a structured format, such as user stories or a requirements document.
6. **Validation and Verification:** Discuss methods for validating and verifying user requirements to ensure their accuracy and feasibility.
7. **Presentation and Feedback:** Have each group present their findings and share the user requirements they gathered. Encourage peer feedback and discussion on the presented requirements.
8. **Reflection and Discussion:** Lead a reflection session where students discuss their experiences and insights gained from the lab exercise, emphasizing the importance of considering user needs in software development.
9. **Wrap-Up:** Summarize the key learnings from the lab and encourage students to apply the knowledge and skills gained in future software development projects.