

Requirements Workshop

Come up with a team name for your group.

Tech Titans.

Please list the names and PIDs of the team members who are present today (or knowingly absent)

Alexander Georgiev (alexandersg)

Prahaara Malli SH (prahaara08)

Akshath Majumder (akshath)

Daniel Lanigan (dylanigan)

Provide your preliminary project idea (or set of ideas). This is not a commitment to a project.

- TODO App: Desktop application to help developers keep track of tasks and prioritize items.

Using the approved idea for your group's course project, complete the following activities related to requirements analysis.

1. Provide an example of five hypothetical non-functional requirements for this system. Be sure to include the specific type of requirement discussed in class, with each requirement coming from a unique category.

- TODO App
 - Database to store the list of tasks.
 - Application to have low latency.
 - Make the GUI user friendly.
 - Cross Platform Compatibility.
 - Should work for multiple Users at a time.

2. Provide an example of five hypothetical functional requirements for this system.

- TODO App
 - Ability to Add Tasks.
 - Ability to Remove Tasks.
 - Ability to Edit Tasks.
 - Assign Prioritize of Task.
 - Set Type of Task (Gym Task, Home Task, Work Task).

3. Think of a specific task required to complete each of the functional requirements and non-functional requirements mentioned above. Estimate the amount of effort needed to complete this task using function points (i.e., using the values here). Briefly explain your answer.

- Non-functional requirements
 - Task: Adding Cross Platform Compatibility functionality.
 - Effort Estimate: Let's estimate this task will be medium effort. Ensuring cross-platform compatibility involves developing and testing the application to work seamlessly on various platforms. This includes adjusting design, testing on different devices, and addressing platform-specific issues.
- Functional requirements
 - Task: Implement the "Create New Task" functionality, including the UI, backend logic, and database integration.
 - Effort Estimate: Let's estimate this task will be medium effort. Creating a new task involves designing the user interface for task creation, handling user inputs, validating data, and storing it in the database. It requires moderate complexity.

4. Write three user stories from the perspective of at least two different actors. Provide the acceptance criteria for these stories.

User Story 1: Task Creation

Actor: Regular User

Story: The user wants to create tasks in the TODO app so that they can organize their work effectively.

Acceptance Criteria:

- When logging into the TODO app, they should see an option to create a new task.
- They should be able to enter a task title, description, and due date.
- They should be able to set the priority of the task (e.g., high, medium, low).
- After creating the task, it should appear in their task list.
- They should receive a confirmation message when the task is successfully created.

User Story 2: Task Deletion

Actor: Regular User

Story: The user wants to be able to delete a completed task they have finished.

Acceptance Criteria:

- The user should be able to select a task from their task list.

- There should be a "Delete" button or option available for the selected task.
- When they confirm the deletion action, the task should be permanently removed from their task list.
- They should receive a confirmation message when the task is successfully deleted.

User Story 3: Task Prioritization

Actor: Gym Personal Trainer

Story: This trainer wants to set the types of tasks that he needs to do for his work and for himself at the gym.

Acceptance Criteria:

- The trainer should be able to create a new task.
- During task creation, he should be able to specify the type of task (either "Work" or "Gym").
- He should have the option to set the priority of the task (e.g., high, medium, low).
- The trainer should be able to view a list of tasks categorized by type (Work or Gym).
- Within each type, tasks should be sorted based on their priority.
- He should be able to edit or delete tasks as needed.
- The trainer should be able to delete tasks when completed.

5. Provide two examples of risk that could potentially impact this project. Explain how you would mitigate these risks if you were implementing your project as a software system.

Technology Compatibility Issues:

Risk: The application may face compatibility issues with different operating systems or hardware configurations.

Mitigation: Conduct thorough cross-platform testing during development to identify and address compatibility issues early on. Use a cross-platform development framework or technology that allows you to build the app for multiple operating systems simultaneously.

Data Loss or Corruption:

Risk: Users may encounter data loss or corruption, potentially resulting in the loss of important tasks and information.

Mitigation: Implement robust data backup and recovery mechanisms within the application. Allow users to create and restore backups of their task lists.

6. Describe which process your team would use for requirements elicitation from clients or customers, and explain why.

Document and Analyze Requirements:

Proper documentation of requirements is crucial for clarity and analysis. Analyzing requirements helps in identifying inconsistencies and conflicts.

Review and Validate Requirements:

Reviewing and validating requirements with stakeholders ensures that they accurately represent the stakeholders' needs and expectations.

Manage Changes:

Requirements can change over time due to evolving business needs or new insights. It's essential to have a process in place to manage and track changes.

Baseline Requirements:

Once the requirements are stable and agreed upon, it's important to create a baseline to serve as the foundation for development.