TEAM : SSD20
PROJECT : VIRTUAL LABS: SOFTWARE ENGINEERING LAB - REQUIREMENTS ENGG - FUNCTIONAL USE CASES

BRIEF PROBLEM STATEMENT

To build a virtual lab that helps the user learn about creating use cases for the requirement analysis phase in software engineering. The goal is to provide the user with a problem statement and let him create(draw) use case diagram according to it and then ask the user to make changes in the diagram by interactively guiding her/him.
This will help the user to familiarize themselves with the basic concepts of use cases & creating them.
Along with the interface, the site would also consist of brief information, theory, and procedure regarding use case diagrams, their creation, and their uses in the requirement gathering phase.

SOLUTION APPROACH

The site will consist of theory of use case diagrams : about, purpose, notation, uses.
Also, users would be given 2 problem statements.
The first problem statement will consist of the problem statement along with the steps of creating the corresponding use case diagram for it in detail to help the user understand how such diagrams are created from problem statements.
The steps will consist of :

a. Learning about the 4 components(and their representation) of a use case diagram : actor, use case, relationships, system boundary.
b. Identify the potential actors (humans or external systems) of the system.
c. For each category of users, identify all roles played by the users relevant to the system.
d. Identify what users are required by the system to perform a given task.
e. Create use cases for every task and connect them with their actors.
f. Structure the use cases.
g. Prioritize, review, estimate and validate the users.
h. Verify the view.

The second problem statement will be an interactive platform to help the user learn how to make use case diagrams. A problem statement given will let the user create(draw) use case diagram according to it. To help the user learn more, the site will interact with the user and suggest her/him help her/him to draw the correct use case diagram.
This will consists of following tasks :

a. Identifying the various actors of the system.
b. Identify the main tasks/modules of the system and group the requirements accordingly.
c. For each task, identify the inputs required and the output to be produced.
d. Arrange the tasks and build relationships between its elements to generate a flowchart for the task.
e. Then combine the flowcharts to demonstrate the flow of the entire system.
f. Answer a few questions (MCQs) to get their context clear.
g. For drawing flow charts and use case diagrams, the elements would be already provided, among which the user must choose the correct ones and drag and drop them into the canvas and link them to create the diagram.

h. The simulation won't move forward unless the user has performed each task correctly.
i. For each task, if the user is stuck, hints would be made available according to what the user is finding difficult.
j. In the end, provide the correct diagram for the same for the user to check his/her solution.

SYSTEM REQUIREMENTS

Tech Stack to be used :

- Basic HTML,CSS and JS.

- JS Libraries to be used :
  1. ReactJS - For an interactive UI
  2. GoJS - For building UML diagrams

System should have any one of the following browsers :
IE10+ /any modern versions of Chrome/Firefox/Safari.

CONSTRAINTS

1. Functional Requirements -

The software must be able to provide the user with an interactive platform to learn thoroughly about the concepts stated in the problem statement.

Scope - Designing the virtual lab with above mentioned functionalities. Would be kept flexible to make sure that content can be changed, if required in future.
Schedule : It must not take more than 30 days to finish the project.
Budget : The project will be built for free of cost using open source/free tools.

2. Non-Functional Requirements -

Performance : The virtual lab must perform its function without causing any memory/ time load to the system.
Security : The software must not cause any security concerns to the user or their system.
Scalability : The software must be flexible enough to make sure that extra changes made to it in future could be done without causing issues in its current version.

TIMELINE

- **12-17 October**
  Preparation of roadmap and Identifying the different modules which need to be implemented and explore the technology stack.

- **18-24 October**
  Implementation of simulation functionality where users would be able to create use case diagrams using various components along with a suggestion pop up which would

guide the user while creating the diagram. The user can ask for hints and get help from a set of FAQs if he gets stuck anywhere.

- **25-31 October**
  Implementation of Introduction, theory, references, procedure to create uml diagrams and case study tabs(these would be static and will contain theoretical content) where the user would be able to study and learn about what are functional use cases, how are they gathered and how are they useful in software development.

- **1-7 November**

  Implementation of exercise and quizzes tabs where users would be able to attempt various quizzes and exercises and self evaluate. After submission the user would be able to see his marks and the correct answer against each question.

- **8-15 November (buffer time)**

  Pending modules(if any) would be completed and testing would be performed.