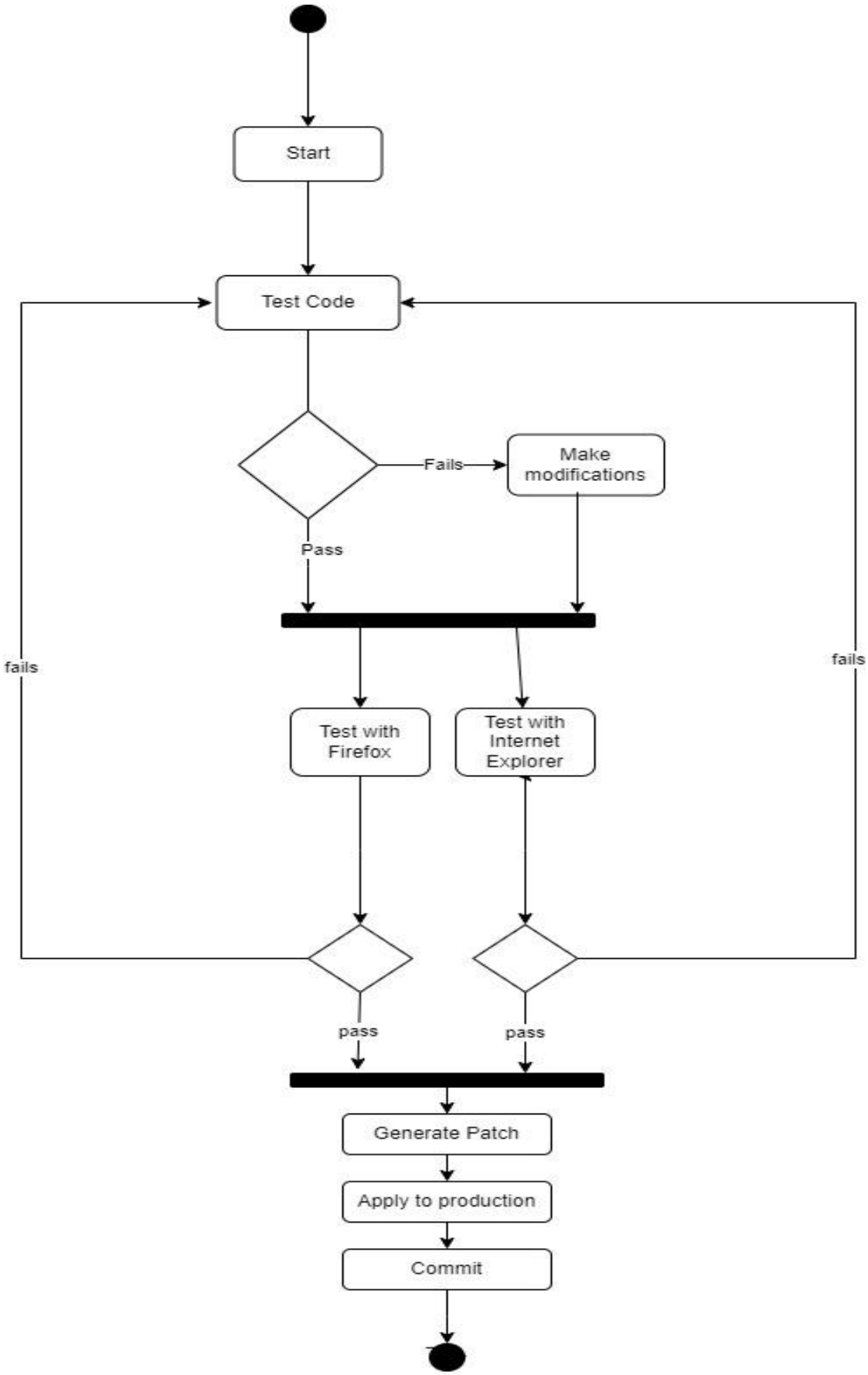
Name : Dhruvilsinh Thakor
ID : 202101462
LAB-6
IT314


Start
Test Code
Fails
Make modifications
Pass
fails
fails
Test with Firefox
Test with Internet Explorer
pass
pass
Generate Patch
Apply to production
Commit

Think over the following questions:

Testing with Multiple Browsers:

- To represent testing with multiple browsers, create a test matrix listing supported browsers and their versions. Set up separate testing environments for each browser, both automated and manual. Utilize tools like Selenium or TestCafe for automated testing. Use a centralized reporting system for results analysis.

Concurrent Patch File Generation and Subversion Repository Update:

- Yes, patch files can be generated concurrently by multiple developers. Code reviews, testing, and integration can also occur concurrently. Developers work on separate branches, create patch files, and submit them for review before integration.

Parallel Production Code Patching and Subversion Repository Update:

- Yes, patching production code and updating the Subversion repository can be done in parallel with careful coordination. The production patch should be isolated and tested in a staging environment before rollout. Implement a controlled rollout strategy and monitor the production environment for issues. Have a rollback plan in case of critical problems.

Q-Identify the basic units of work and visualize the workflow:

- **Basic Units of Work:**
  - Requirement gathering and analysis
  - Design and architecture planning
  - Development of code and features
  - Testing and Quality Assurance
  - Deployment and release management
  - Maintenance and support

- **Workflow Visualization:**

  The workflow can be represented as a linear process, with each stage flowing from one to the next. Here's a simplified representation

Identify activities that could be done in parallel:

- Development and Testing:
  - While one team is developing features, another can be testing previously developed ones. This helps identify and address issues early.

- Requirement Gathering and Design:
  - Preliminary requirement gathering can occur simultaneously with the design phase. Designers can start creating wireframes or prototypes based on initial requirements.

- Deployment and Maintenance:
  - Deployment of a new release can happen while the previous version is still being maintained. This ensures a seamless transition for users

    .

Identify stages from where progress could be made only after a list of criteria is satisfied:

- Testing Stage:
  - Progress from the testing stage to deployment should only happen after all identified bugs and issues are addressed and the acceptance criteria are met.

- Deployment Stage:
    - Before deployment, criteria such as successful testing, code reviews, and stakeholder approvals should be met to ensure a stable release.

- Maintenance Stage:
    - Progress to the next development cycle or major release should only occur after all critical issues in the current version are resolved and user feedback is incorporated.

- Release Planning Stage:
    - Progress to development should only happen after thorough analysis and prioritization of gathered requirements.

Class Diagram :