UNIVERSITY OF ENGINEERING AND TECHNOLOGY,
PESHAWAR PAKISTAN

Main Campus

Software Engineering

Assignment 06

Submitted By

Name:	**Muhammad Mohsin**
Registration No.	**23PWBCS0973**
Semester:	**BS CS 5th**
Section:	**A**

Submitted To : Miss Kanwal

DEPARTMENT OF COMPUTER SCIENCE & IT

Accessibility Simulator Web Application

A. Problem Statement

Modern websites often overlook how users with disabilities experience digital content. This Accessibility Simulator helps developers and designers preview their website as it would appear to users with visual, motor, or cognitive impairments.

B. Project Scope

In Scope (System WILL Do):

- Simulate different types of colour-blindness.
- Apply low-vision effects (blur, zoom, contrast loss).
- Simulate motor impairment using jitter cursor.
- Allow users to load a URL for testing the web app.
- Apply multiple simulations simultaneously.
- Provide real-time preview updates.

- Perform automated WCAG accessibility checks.

Out of Scope (System WILL NOT Do):

- Modify or fix accessibility issues in websites.
- Provide backend services(storage), logins, or user accounts.
- Store personal data or usage history.
- Evaluate website performance or analytics.

C. Stakeholders

- Frontend Developer – Tests UI accessibility during development.
- UI/UX Designer – Evaluates design accessibility.
- QA Tester – Tests user flows under impairment simulations.
- Student / Educator – Learns about accessibility challenges.

D. Functional Requirements (8–10)

1. Allow users to enable and disable colour-blindness filters.
2. Allow users to select specific colour-blind types.
3. Provide low-vision filters including blur, zoom, and contrast reduction.
4. Enable jitter cursor mode to simulate motor impairments.
5. Allow users to enter URL for testing a websit/app.
6. Combine multiple accessibility filters at once.
7. Adjust intensity of each filter.

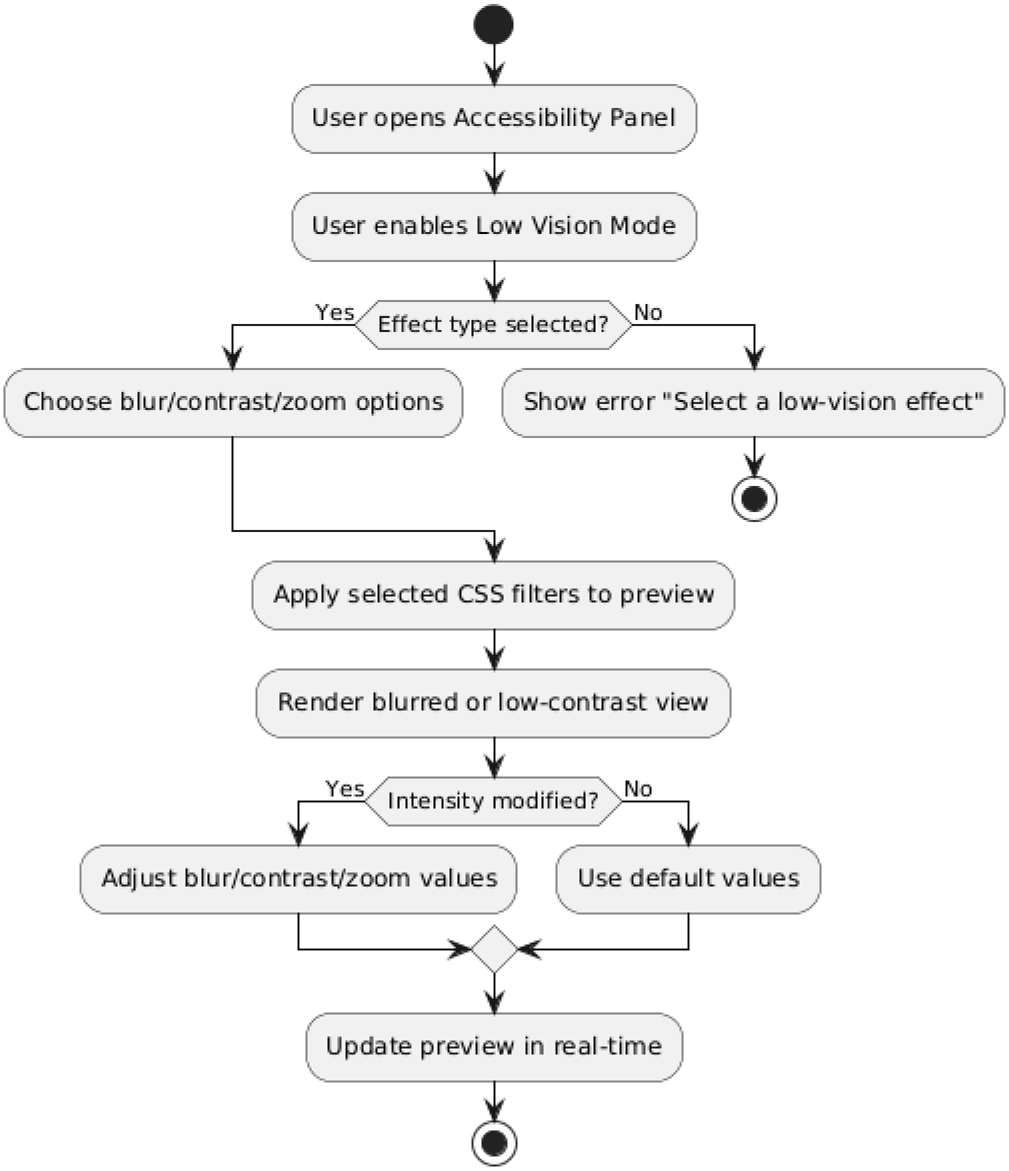
8. Display real-time preview updates.
9. Reset all filters and return the preview to default mode.

E. Non-Functional Requirements

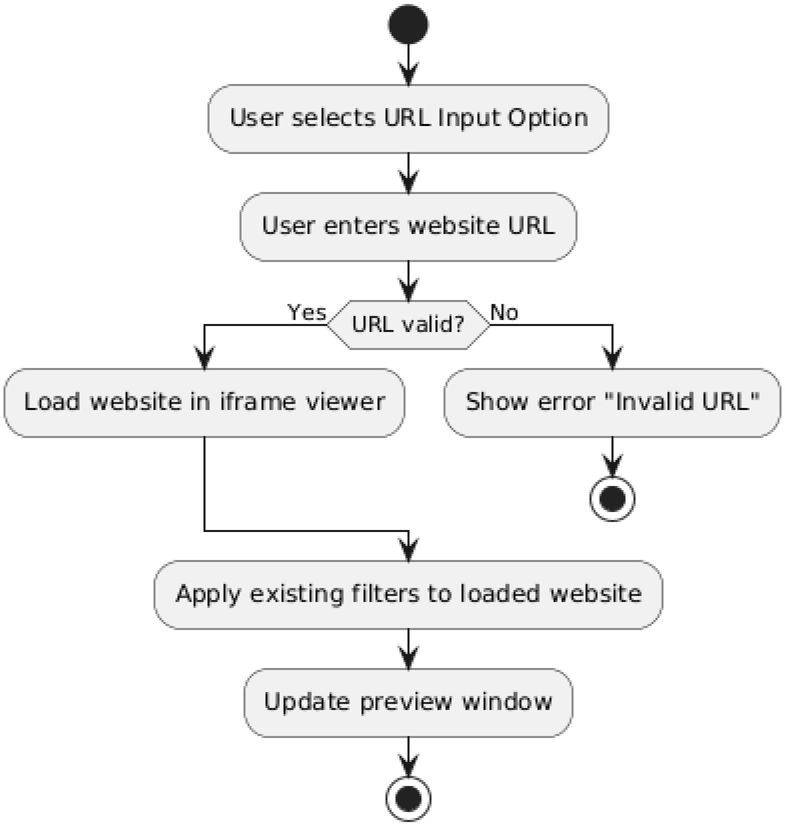
1. Provide UI updates with a response time under 100 ms.
2. Maintain compatibility with major browsers.
3. Ensure a user-friendly, clean, and intuitive interface.
4. Operate fully on the client side with no data storage.
5. Use modular component architecture for easy maintenance.

F. Activity Diagrams

Activity Diagram - Apply Low Vision Filter

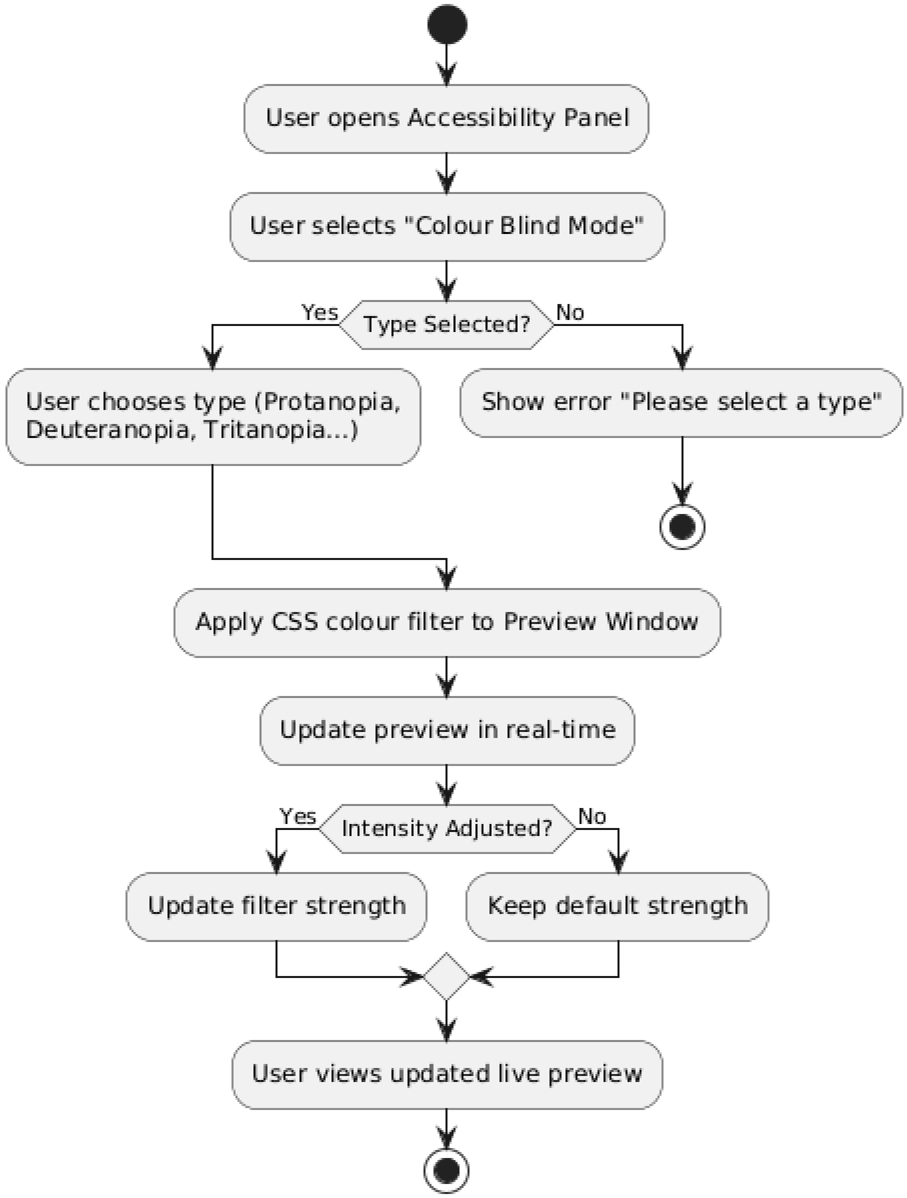

User opens Accessibility Panel
User enables Low Vision Mode
Effect type selected?
Yes
No
Choose blur/contrast/zoom options
Show error "Select a low-vision effect"
Apply selected CSS filters to preview
Render blurred or low-contrast view
Intensity modified?
Yes
No
Adjust blur/contrast/zoom values
Use default values
Update preview in real-time

Activity Diagram - Load Website URL

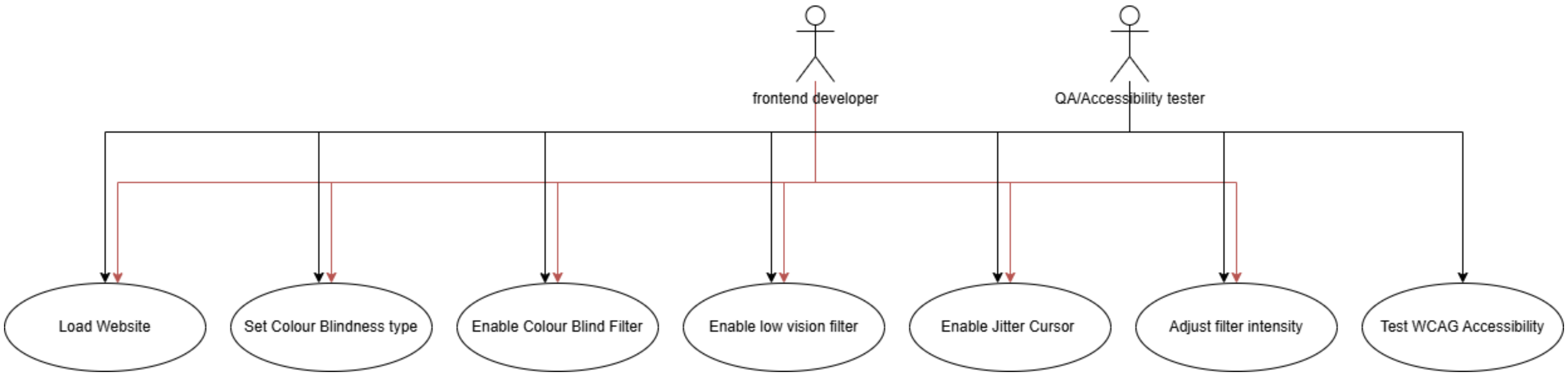


User selects URL Input Option
User enters website URL
URL valid?
Yes
No
Load website in iframe viewer
Show error "Invalid URL"
Apply existing filters to loaded website
Update preview window

Activity Diagram - Apply Colour Blind Filter


User opens Accessibility Panel
User selects "Colour Blind Mode"
Type Selected?
Yes
No
User chooses type (Protanopia, Deuteranopia, Tritanopia...)
Show error "Please select a type"
Apply CSS colour filter to Preview Window
Update preview in real-time
Intensity Adjusted?
Yes
No
Update filter strength
Keep default strength
User views updated live preview

G: Use Case Diagram


frontend developer
QA/Accessibility tester
Load Website
Set Colour Blindness type
Enable Colour Blind Filter
Enable low vision filter
Enable Jitter Cursor
Adjust filter intensity
Test WCAG Accessibility