**Ghazi University Dara Ghazi Khan**

Name:

Mehtab Hafeez.

Class:

BS\_IT\_ 7th Eve ( B ) .

Subject:

Software Requirement Engineering .

Formal Methods .

Reg No: 2021 GU 0809.

Session: 2021 \_ 2025.

Instructor: Sir Mr.Waseem Shab.

Page 1

**Traffic Light Controller Project Documentation.**

1. **Problem Statement and Requirements.**

* 1. **Problem Statement**.

Traffic congestion and safety at intersections are significant issues in urban areas. An effective solution is to implement a traffic light control system that effectively manages the flow of vehicles and pedestrians. The goal of this project is to develop a simulated traffic light control system that operates based on predefined rules, displaying the correct light states (Red, Yellow, Green) at appropriate intervals.

* 1. **Requirements.**

The primary requirements for the Traffic Light Controller system include:

**1. Functional Requirements:**

* The system must simulate a traffic light with three states: Red, Yellow, and Green.
* The red light should be displayed for a longer duration (e.g., 5 seconds).
* The green light should be displayed for a specified duration (e.g., 5 seconds).
* The yellow light should be displayed for a brief duration (e.g., 3 seconds).
* The system must visually update the light states to indicate the current state.
* Display the current state of the traffic light on the user interface.

**2. Non-Functional Requirements:**

* The UI should be user-friendly and intuitive.
* The system should be responsive and work well across different devices with varying screen sizes.
* Ensure that the system is efficient and does not consume excessive resources.

Page 2

1. Formal Specification and Modeling Process

**2.1 Formal Specification.**

The Traffic Light Controller can be formally specified using state transition systems, where the states represent the light colors and conditions under which the transitions between states occur:

* States:
* Red: When the red light is on.
* Green: When the green light is on.
* Yellow: When the yellow light is on.

* Transitions
* From Red to Green after 5 seconds.
* From Green to Yellow after 5 seconds.
* From Yellow to Red after 3 seconds.

# 2.2 Modeling Process

To model the traffic light controller, we utilized a combination of HTML, CSS, and JavaScript to create a web-based user interface:

* HTML,PHP : Defines the structure of the user interface, including the traffic light and status display.
* CSS: Adds styling to improve the overall appearance of the traffic light.
* JavaScript: Contains the logic to handle state changes, timings, and visual updates.
* Alloy Analyzer: Alloy Analyzer into the Traffic Light Controller project can enhance the project by providing formal verification capabilities. Alloy is a modeling language and tool that allows developers to specify a system's structure and behavior, enabling analysis of the model for correctness and consistency.

The system was modeled through a series of function calls (in JavaScript) that coordinate the transitions between states, ensuring the lights change at the appropriate intervals.

Page 3

# Verification Results and Issues Found

## Verification

The traffic light controller was tested using a combination of manual testing and informal verification methods, confirming that the light states transitioned correctly according to the defined specification.

Test Cases:

* Test Case 1: Verify the light turns green after 5 seconds of red.
* Test Case 2: Verify the light turns yellow after 5 seconds of green.
* Test Case 3: Verify the light turns red after 3 seconds of yellow.

## Issues Found

Syntax error and misbehave model in alloy checking cost becouse its payed During testing, a couple of issues were identified:

* Incorrect Status Messaging: The initial implementation displayed incorrect status updates. For example, the message indicated "Current State: Green" when the red light was actually displayed. This was corrected by adjusting the order of operations in the JavaScript statechanging logic.
* Visual Lag: On slower devices, there was a minor lag in visual updates as the light transitioned.

# Key Challenges and Lessons Learned

## Key Challenges

1. State Management : Effectively managing state transitions was a challenge, particularly ensuring that the status displayed matched the current light state.
2. **UI Responsiveness :** Making the UI responsive was also challenging, particularly to ensure consistent behavior across different platforms and devices.
3. Timing Consistency : Ensuring accurate timing across different browsers and devices was critical and required thorough testing.

Page 4

## Lessons Learned

1. Importance of Clear Requirements : Clearly defined functional and non-functional requirements helped streamline the development process and avoid scope creep.
2. Iterative Testing : Regular testing during development helped catch bugs early in the process, particularly related to state management.
3. User Interface Design : Designing a user-friendly interface is vital; the learning gained here will influence future projects.
4. Documentation : Keeping good documentation throughout the project is crucial for future maintenance and scalability.

# Conclusion

The Traffic Light Controller project provided valuable insights into software development, particularly in the areas of state management, user interface design, and testing. This experience underscores the importance of a clear problem statement, iterative verification processes, and responsiveness in software systems. The final product successfully simulated a traffic light controller, achieving the primary goals of the project while adhering to functional requirements. Future iterations could enhance the system with additional features, such as pedestrian signals or traffic flow analysis







