**Software Engineering Lab – 1**

**Dinesh Kumar M**

**22BRS1369**

**Traffic Monitoring System**

Objective:

Monitor the traffic in different areas of the city and guide the citizens to move to a place consuming less amount of time while avoiding traffic and providing them with the most efficient route.

**WATERFALL MODEL**

1. Requirement Gathering and Analysis:

* Understand the needs of citizens, traffic patterns, and existing infrastructure.
* Collect data on peak hours, congestion points, and user preferences.
* Define system requirements, including accuracy, response time, and scalability.

1. System Design:

* Create a high-level design (HLD) that outlines the system’s architecture.
* Specify components like traffic sensors, data processing servers, and user interfaces.
* Detail how data will flow from sensors to the guidance system.

1. Implementation:

* Develop the system based on the design.
* Code traffic data collection modules, route optimization algorithms, and user interfaces.
* Integrate with existing infrastructure (e.g., GPS services).

1. Testing:

* Conduct unit tests for individual components.
* Perform integration testing to ensure seamless communication.
* Validate accuracy, responsiveness, and reliability.

1. Deployment of System:

* Deploy the system in targeted areas.
* Train citizens on using the guidance app.
* Monitor real-world performance and address any issues.

1. Maintenance:

* Regularly update traffic data and algorithms.
* Address bugs, optimize routes, and adapt to changing conditions.
* Provide ongoing support to citizens.

Requirement gathering & analysis

Maintenance

Deployment of System

Testing

Implementation

System Design

**INCREMENTAL MODEL**

1. Requirements Gathering and Initial Design:

* Identify high-priority features and gather initial requirements.
* Create a basic design for the core functionality (e.g., data collection, route optimization).

1. First Increment:

* Develop and deploy the initial version with essential features.
* Citizens can access basic traffic information and receive simple route suggestions.

1. Feedback and Enhancements:

* Gather user feedback and monitor system performance.
* Based on feedback, enhance the system incrementally (e.g., add real-time data, improve accuracy).

1. Subsequent Increments:

* Iteratively add features: dynamic rerouting, alternative routes, personalized recommendations.
* Each increment builds upon the previous version, improving efficiency and user experience.

1. Testing and Deployment:

* Test each increment thoroughly.
* Deploy updated versions to targeted areas, ensuring backward compatibility.

1. Ongoing Maintenance:

* Continuously refine the system based on real-world usage.
* Address bugs, optimize algorithms, and adapt to changing traffic patterns.

Build n

Build 2

Build 1

Requirements

Implementation

Testing

Design & Development

Implementation

Testing

Design & Development

Implementation

Testing

Design & Development