

Visual transportation support system

VTSS

1. Introduction

This application shows how people are using the public transportation system in Gothenburg, Sweden.

2. Purpose

The purpose of this application is to highlight possibilities for improvement of the public transportation system in a city.

3. Intended Audience

The VTSS application is intended to be used by city planners and administrative personnel at the västtrafik office.

4. Intended Use

In essence the application will graphically visualize bottlenecks and blind spots regarding bus and tram stops, which are acquired by analysing the user-requests to and from various destinations in the greater gothenburg area.

5. Scope

The application will take user requests as data input and plot these points on a graphically visualized map. It will mark blind spots and bottlenecks by colored markers, in order for users of the app to see where performance issues may arise, for example during certain hours of the day, when more people than usual are using public transportation (e.g. rush hour).

6. Overall description

The software will be a new product created by students of the Gothenburg University. It is using third party components, such as the Västtrafik API, the Google maps API and the Mapbox API. The app is built for västra götaland regionen as an aid to improve the infrastructure of the public transportation system.

7. User needs

The Västtrafik personnel and city planners are the intended users of this system. They will receive the visualization of the potential bottlenecks or blind spots on a map as user-requests are sent.

8. Assumptions and Dependencies

Assumptions:

The amount of requests during a set period of time won't be larger than the population of the greater gothenburg area (1 040 000)

Dependencies :

Our dependencies are the Västtrafik API (resrobot) , Google maps API & mapbox API.

9. System features and requirements

Functional Requirements for the Creator:

- C1. The system shall generate timestamps for each route at the start of a route
- C2. The system shall send the generated JSON data via MQTT
- C3. The system has to be an external entity, connected via MQTT. It shall publish under the topic "external"
- C4. All components that are subscribing to MQTT have to be able to change the TOPIC, the MQTT BROKER IP and the PORT without code changes
- C5. The system shall provide two different scenarios, simulating different load on the system

Extra-functional Requirements (non-functional) for the Creator :

- EC1. The system should handle disconnection by reconnecting within 2 seconds.

Functional Requirements for the Generator:

- G1. The system shall generate random locations (latitude,longitude) in the Gothenburg area
- G2. The system shall send the generated JSON data via MQTT
- G3. The system shall save the generated points in the given format in JSON object
- G4. The system shall create timestamps for the generated requests, that are adjustable to suit the chosen scenario
- G5. The system shall offer a monotonically growing requestId, i.e. a counter starting at 1, for each device (in accordance with Lamport Timestamps).
- G6. The system shall be able to adjust the frequency and number of generated data

Extra-functional Requirements (non-functional) for the Generator :

EG1. The requests shall be sent every second with a maximum delay of 1 second.

Functional Requirements for the Visualizer:

V1. The system shall receive JSON data via MQTT

V2. The system shall plot each data point on a map

V3. The system shall provide a GUI

V4. The system shall represent the flow of user requests

Extra-functional Requirements (non-functional) for the Visualizer :

EV1. The data should be plotted with a frequency of at least one second

Functional Requirements for the System as a whole:

S1. The system shall offer documentation based on the assignment description

Functional Requirements for the Documentation:

D1. The documentation repository shall contain a document performanceAndFaultTolerance.pdf

D2. The document performanceAndFaultTolerance.pdf shall contain an explanation of your choices for QoS levels for each publication and subscription

D3. The document performanceAndFaultTolerance.pdf shall contain an explanation of the fault-tolerance mechanism(s) you have implemented in the system

D4. The document performanceAndFaultTolerance.pdf shall contain a discussion about the potential impact of your architecture as well as the choices of D2. and D3. above on scalability, throughput, latency, availability and consistency.

D5. The repository shall contain diagrams of the following types to present your understanding of the system: Sequence Diagram, Component Diagram (both implementation and logical/conceptual), Deployment Diagram, Add further diagrams to highlight noteworthy aspects of the system.