

E-MAPIS Documentation

Table of Contents

1. REQUIREMENT'S SPECIFICATION.	2
1.1 PURPOSE OF THE SYSTEM.	2
1.2 A HIGH-LEVEL OVERVIEW OF THE SYSTEM.	2
1.4 THE MOST IMPORTANT QUALITY ATTRIBUTES (NON-FUNCTIONAL REQUIREMENTS).....	3

1. Requirement's specification.

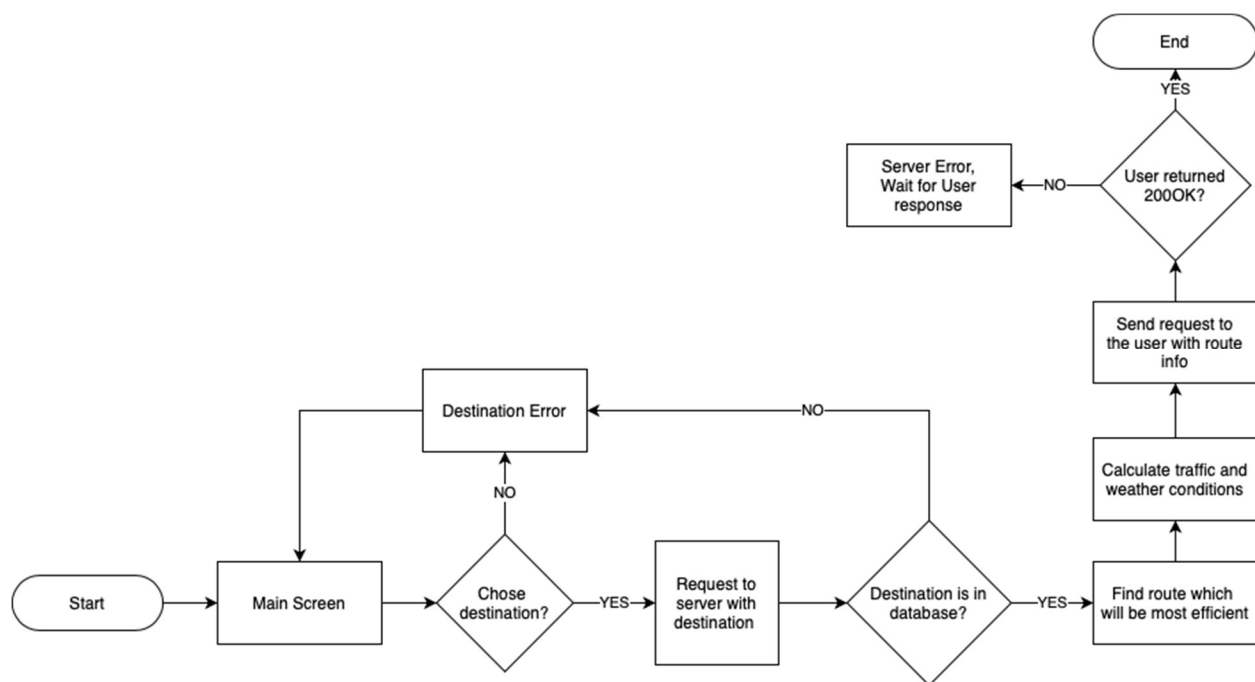
1.1 Purpose of the system.

E-MAPIS will be a data collection application for vehicles, that will give its users useful information to plan a stress-free road trip and make their daily commute a lot better, as well as, collect various anonymous data. The application will be able to collect location data and provide the user with functionality to input various trip and vehicle parameters (such as energy or gas consumption for the trip, vehicle type and details, etc.) In addition, the application will take traffic and weather conditions into consideration to calculate the most efficient route to a destination.

1.2 A high-level overview of the system.

E-MAPIS will consist of 2 main parts. First, a server with GIS data, data about the vehicle, various algorithms for fastest route searching, weather, elevation, and traffic condition APIs. The second part will consist of the user application, which will be responsible for sending/receiving requests, processing data, collecting data from the user's device, and a GUI.

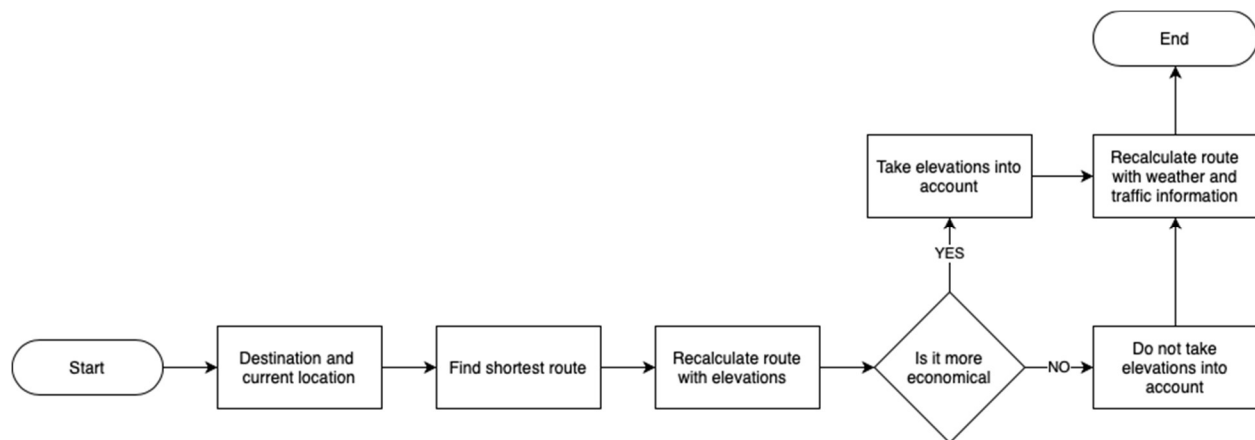
The table below (see *Flowchart 1*) explains the architecture that would be used to develop the system. The architecture flowchart provides an overview of the system, simply identifying the main components that would be developed for the application.



Flowchart 1. An example of a high-level

1.3 Complete functional requirements (reasoning/value behind the requirement).

The main tool for this app will be a relational database, in particular PostgreSQL. After a fair amount of research, we found a suitable PostgreSQL geographic information system extension PostGIS. It adds support for geographic objects allowing location queries to be run in SQL. To test if the data was correctly inserted, we will use the QGIS tool. For weather, traffic, and elevation information we will use public APIs.



Flowchart 2. Effective route-finding algorithm

1.4 The most important quality attributes (non-functional requirements).

SECURITY

The app will be using HTTPS requests to ensure data encryption.

CAPACITY

For the time being, E-MAPIS will not store a big amount of data. Implementing Vilnius map in the database will take up from 3-10 GB of memory. On the other hand, the application will be collecting GIS data, fuel consumption, and other details from the user, so storage may be in shortage.

COMPATIBILITY

The application will be usable only on Android devices, with the option to also implement it to iOS in the future. The minimal and recommended requirements will be determined but are considered to be low.

USABILITY

Every person that is capable of making calls with their smartphone, will immediately understand how to interact with E-MAPIS. The user interface will be simple and uncomplicated.

SCALABILITY

The application is not considered to be run under high workloads.

PERFORMANCE

The application itself is considered to be a lightweight application which means it will function briskly.