System design document for PlankaGBG

Lucas Karlsson, Joakim Ohlsson, Seif Bourogaa,
Joakim Tubring, Filip Hanberg

2020-09-25
version 1.1

1 Introduction

This document contains technical information about an application that document ticket controllers position in a tram network. The application is in this moment built as an Android application. Which displays the information and will allow for search queries on user reports of ticket controllers activity and allows the user to create new reports on their activity.

1.1 Definitions, acronyms, and abbreviations

Definitions etc. probably same as in RAD

2 System architecture

2.1 Overall description

At the moment all application components are run locally. There are plans on implementing a Server-Client model using a web-API to connect the server side with the client. The Server side will contain an server application and a database relaying information to the client-side.

2.2 Classes

2.2.1 AbstractReport

Responsible for creating Report objects. Contains the number of controllers, timestamp of report and and possibly an image.

2.2.2 Graph

The graph representation, contains an adjacencylist containing nodes and their connections to other nodes. Also responsible for creating, maintaining, traversing and updating the graph structure. The Graph class is dependent on the Node class for creating nodes. The class is also dependent on Station and Route classes for creating routes and stations.

2.2.3 Route

A representation of a route. Contains a line number and a list of stops on the route. Dependent on Node class for creating nodes.

2.2.4 Station

A representation of a station in the tram network contains the name of a station and a list of all individual stop positions. Dependent on Node class for creating nodes.

2.2.5 Incident

A representation of an incident. Contains a list of reports connected to the incident and the collective trust factor of the incident. Dependent on AbstractReport for the list of reports.

2.2.6 ReportRoute

A representation of an report bound to a station. Contains an AbstractReport and a Route.

2.2.7 ReportStation

A representation of an report bound to a route. Contains an AbstractReport and a Station.

2.2.8 Model

The model class will be the connection between the model and the GUI. At the moment it has not yet been implemented, more information on this later.

2.3 Communication protocols

At the moment there are no communication protocols in use as the application is only run locally.

2.4 Description of program flow

2.4.1 Startup

*CSV-files are processed by the Graph class.

*Graph is created using HashMap created from Node-objects.

*Nodes are mapped such that source(Key) node is mapped to destination(Value). The destination is all nodes we could travel to from sourceNode.

*The routes are created reading the Nodes of one line in travel order and binding them to a route object with an int containing the linenumber.

*The stations i created by creating a station object containing the station itself and a list with the stations individual positions A,B,C and so on.

*The data is stored in the Graph class.

2.4.2 RunTime

2.4.3 Closing

There are no methods that write data to offline storage at closing the application the data is simply deleted.

3 System design

We have decided to use Model-View-Controller. This is what our Domain Model looks like:

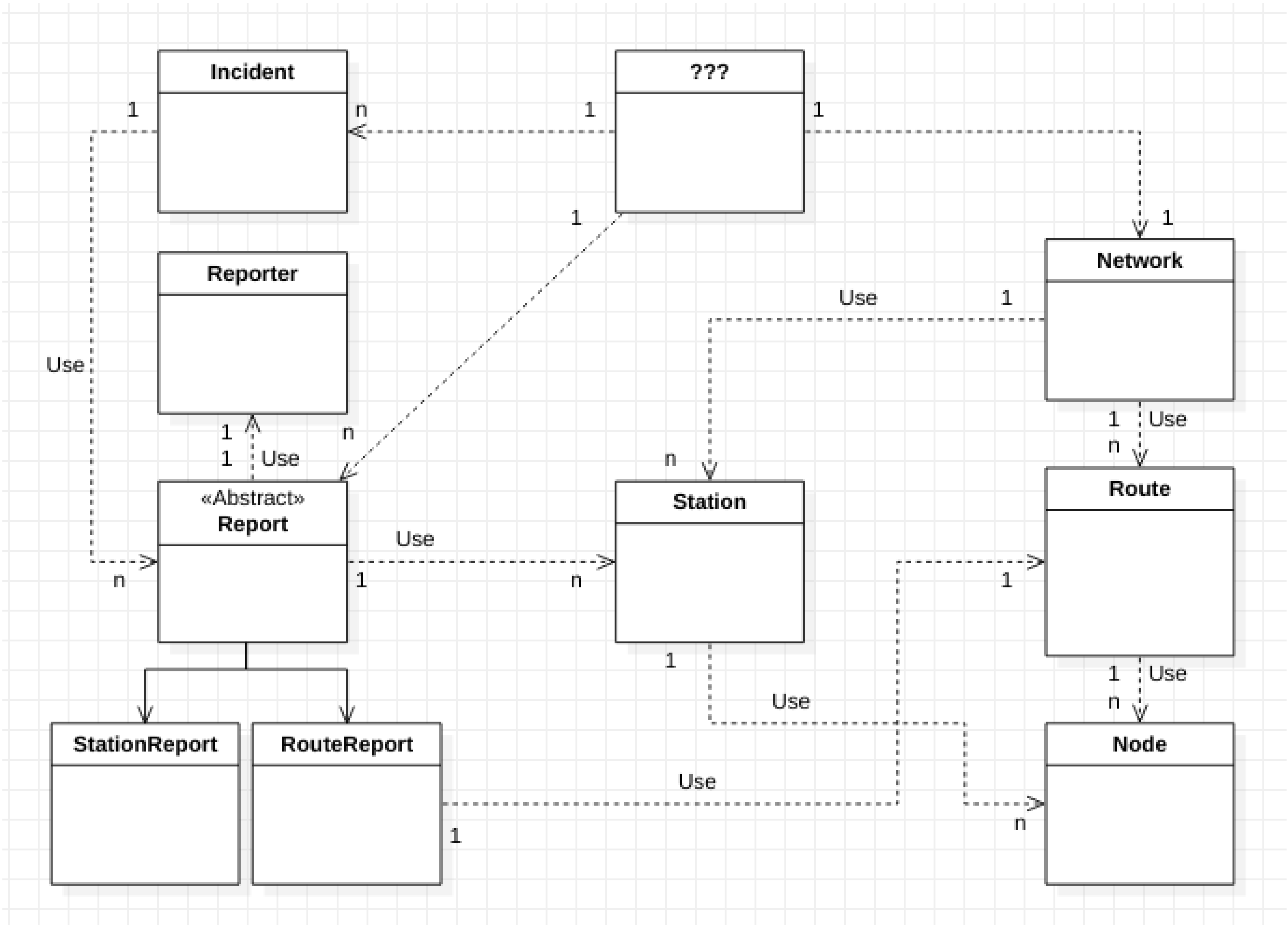

Figure 1: Domain Model

In the current iteration the focus has been on the Model Part of our MVC as shown above. Most of the classes has been implemented apart from the Model class.

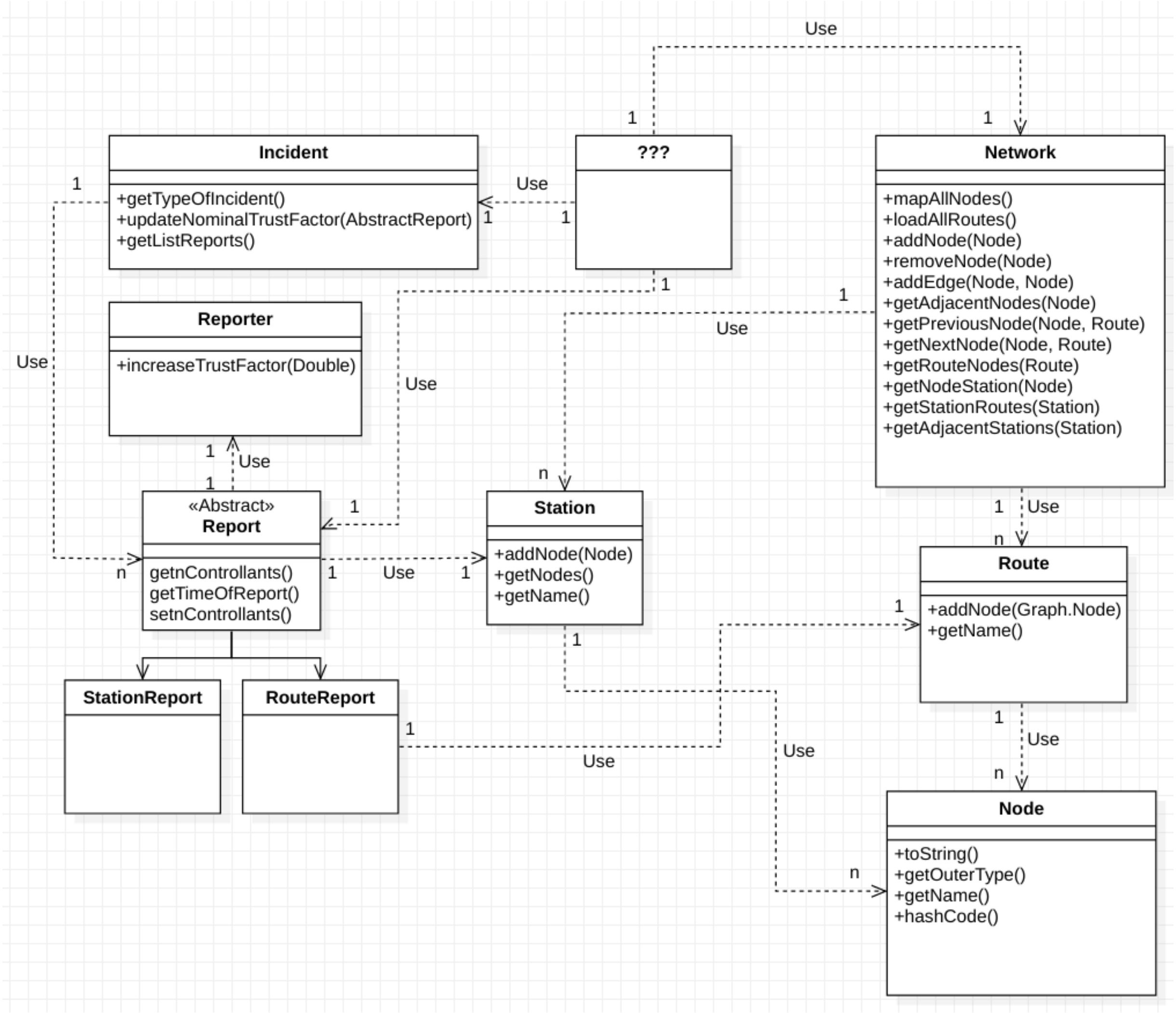

Figure 2: Design Model

We intend connect View to Model using Controllers for input and Observer Pattern for output.

4 Persistent data management

There is no persistent data apart from the CSV-files for creating the Graph and it's components.

5 Quality

Quality assurance will consist of unit testing and code reviews. Testing is done using JUnit-tests. Tests will be provided in test-directory and the code will be tested continuously throughout project.

5.1 Known issues

5.2 Access control and security

No access control is implemented at the moment.

6 References