KIT Campus Routing System

Functional Specifications Document

Vincent Diener, Kenan Ibrahimov, Sebastian Friebe,
Sarah Lutteropp, Melanie Magin, Kris Reiling

6th May 2013

Contents

1 Introduction 3

2 Objectives 3
- 2.1 Mandatory Criteria – Map Component . . . 3
- 2.2 Mandatory Criteria – Routing . . . 4
- 2.3 Mandatory Criteria – Administration Tool . . . 5
- 2.4 Facultative Criteria – Map Component . . . 5
- 2.5 Facultative Criteria – Routing . . . 5
- 2.6 Facultative Criteria – Administration Tool . . . 6
- 2.7 Excluded Criteria – Routing . . . 6
- 2.8 Excluded Criteria – Administration Tool . . . 6

3 Product Usage 7
- 3.1 Routing . . . 7
- 3.2 Administration Tool . . . 7

4 Product Environment 8

5 Functional Requirements 8
- 5.1 Main Functions . . . 8
- 5.2 Facultative Functions . . . 9

6 Nonfunctional Requirements 11
- 6.1 Routing . . . 11
- 6.2 Administration tool . . . 11

7 Product Data 12
- 7.1 Essential Data . . . 12
- 7.2 Facultative Data . . . 13

8 Global Test Cases 13
- 8.1 Routing - Test cases . . . 13
- 8.2 Administration tool - test cases . . . 14

9 System Model 15
- 9.1 Use Case Diagrams . . . 15
- 9.2 Activitiy Diagrams . . . 17
- 9.3 Graphical User Interface . . . 19
- 9.4 Graphical User Interface . . . 19

10 Glossary 21

Merge this to the routing and re-mention shortly in the Administration tool.

Difficult to read.

1 Introduction

— It is often difficult for KIT visitors and new ~~students~~ students to find the buildings, rooms or ways between them on the campus. Our product will simplify this task, leading to better time efficiency and less stress for everyone.

The general goal of this project is the design and implementation of a routing system for the KIT campus. The system will be able to compute and to display the shortest path between a starting point and a destination. It will be possible to route from and to buildings and to rooms inside buildings. For the latter, the system will display a floor plan of the building. It will also be possible to search just for the destination, without computing a route from some other point.

— On the administration side, the system will provide a graphical user interface (GUI) to modify the underlying map data if needed (e.g. because of construction works or more details are needed) or create it from scratch. It will be possible to easily add/delete buildings with or without routing information of the interior. Moreover, it will be possible to add vertices and edges with information that are necessary to do the routing.

The project will by shipped with the map data of at least two buildings of "KIT Campus Süd" and a map of the outside campus.

— Both routing and administration will share a component for map visualization.

Is this necessary in the intro?

2 Objectives

In this section, we describe in detail the functionality of the routing tool and the admin tool. This includes the functionality and the different ways for the user to input data into the tools, as well as how the tools format that information and show it to the user.

2.1 Mandatory Criteria – Map Component

2.1.1 Criteria used in Routing and Administration Tool

Displaying the map

- Display a position on the map
- Display a path between two places on the map
- Display ways passing under/through buildings differently
- Display a map of the campus without the underlying graph

  This probably means that by default you will show to the user the vertices and edges?

  It is also possible to just look at the map of the campus, without searching for any points or routes. Possibility to enable/disable the view of the graph on top of the map. (Probably only for the administration tool).
- Display a floor plan of a building

  Like the campus map, the floor plan of a building can also be viewed without specifically searching for a point within that building.

User Interaction with the map / Navigation ~~of~~ on the map

- Provide a zooming and panning function for the map

  The user can change his view-point by zooming in/out and panning

  It is vital for any application dealing with maps to provide those basic navigation features.
- Switch between floors of a building

When selecting a building, the main map view will be replaced by a floor plan of the selected building. Once this happens, it is possible for the user to switch between the different floors of the selected building.

- Find nearest vertex in the underlying graph of the map by clicking on the map – Functionality

2.1.2 Criteria only used in Administration Tool

- ~~Draw the~~ Underlying graph is displayed on top of the map.
- Highlight a set of vertices and edges

  Vertices and Edges of the underlying graph will be drawn in another color when selected by the user.
- Highlight an outline of a building Why do you need the outline? Or you mean the road around?

  An outline of a building will be drawn in another color when selected by the user.

2.2 Mandatory Criteria – Routing

2.2.1 Input

- Provide text fields for searching locations – The main point here is that user is able to search for desired location.

  Locations can be searched for by typing identifying information into a text field. This information includes building names/numbers, room name/numbers and addresses. Using these text fields, you can provide a start and end point when searching for a route.
- Provide auto-completion/ search suggestions in the search fields
- Select start and destination by clicking on the map

2.2.2 Computing the route

- Find a location
- Routing for pedestrians
- Use a fast algorithm for routing

  Calculation speed is essential for routing applications. Our goal is to minimize this time as much as possible. For further explanation, see "Nonfunctional Requirements".
- Compute the shortest path between two locations (in- and outdoor)

  It is possible to search for routes on the whole campus map. Those routes may or may not start or end in buildings.
- Provide shortcuts through other buildings, when computing shortest path, The system will take into account routs that pass through other buildings.
- When routing to a building (or lecture hall) with more than one entrance, the path to the nearest entrance is calculated and not the path to the main entrance.

2.2.3 Output

- Display a path between two places using the Map Component

  After the user has specified a start and end point, the calculated route is shown on the map. The route will be clearly visualized and easy to understand.

- Textbased route output

  Show a written form of the route. E.g., "Go 100m. Turn right. Go 25 m. You reached your destination. "

2.3 Mandatory Criteria – Administration Tool

- Create, open or save a map

  It is possible to open and save a map of the campus or of a building and its underlying graph.

- Exchange the background image of the map
- Create and remove buildings, vertices and edges
- Edit information tags attached to buildings, vertices and edges

  These information tags include: Links to other vertices, link to building, name, address and opening hours

  ? ?

- Add floor plans to a building – Clarify this
- Move vertices and edges on a map. (modify)
- Add entrances of a building – Maybe this is only a tag of a vertex?

What about stairs... don't you need to know that some edges represent stairs? How are you going to add them?

2.4 Facultative Criteria – Map Component

- Change route graphically by moving points/ add waypoints (see Google Maps)

Not clear what do you mean...

2.5 Facultative Criteria – Routing

2.5.1 Filtering

By filtering nodes and edges, we can restrict our graph to a subgraph of it (e.g. use no edges tagged with "contains stairs", use only vertices with valid opening hours). This will allow the following additional modes: ↳ Until now I do not see how you store that an edge represent stairs...

- Search for multiple places (i.e. restrooms)
- Search for the nearest restroom/ cycle stand/ coffee machine
- Routing for wheelchair users
- Routing for cyclists

The same?

- Find the nearest exit inside a building
- Search only inside a building (Why this is different from global search, What will happen if you do not implement this feature and a user enters two points in the same building?)

  It is possible to search for routes within certain buildings only.

- Search for points of interest

  Which?

It is possible to search for certain places within buildings, (e.g. locate restrooms inside a building) as well as search for points of interest outside of buildings (e.g. locate bycicle stands on the map).

- Create an own filter for routing

2.5.2 Searching

- Do spellchecking in the search fields ???
- Bookmark favourite search requests

2.5.3 Computing the routes

- Show alternative routes (Do you mean just display several routs or routs for bykes, wheelchairs?) Probably out.
- Calculate average time needed to walk the route, enter own average speed

2.5.4 Different Platforms

Why this is in "Routing"?

Android App

- Provide an Android App
- Use GPS position to find the nearest vertex in the underlying graph of the map
- Live routing on a smartphone or tablet with GPS

Java Applet

- Provide a Java Applet

2.6 Facultative Criteria – Administration Tool

- Import OpenStreetMap data
- Add new categories of information tags (i.e. "contains stairs") So routs for bykes, wheelchairs are facultative.
- Temporarily close some roads (e.g. which are under construction)

2.7 Excluded Criteria – Routing

- No routing for cars
- No map data outside "KIT Campus Süd" will be provided by the project
- No server based application
- No speech input
- No public transport for parts of the route will be used
- No photo tags of locations/places

2.8 Excluded Criteria – Administration Tool

- No server based application
- No auto generation of the map graph from images
- No freehand drawing on the map ???

3 Product Usage

This section describes how and by whom the product will be used and what they need to use it. As they share a lot of their functionality, the routing tool and administration tool will be implemented within the same application. This allows the user of the administration tool to switch to the routing tool at any time to test the map data he just created. However, the target audience for the tools is not the same so a user of the routing tool will most likely never use the administration tool.

3.1 Routing

3.1.1 Scopes

Nowadays, it is not unusual for universities to be situated within extensive campuses. While this is of course convenient, it can also be quite confusing for students, especially for freshmen. They have to look up lecture halls on big, confusing maps that are often not even up to date. As a result, a lot of time is wasted. We seek to change this with our campus routing tool. The tool provides a well-arranged, simple yet powerful map application that can be intuitively used by students or people working on the campus to easily find any building or calculate the shortest route between any two points. The location or route is then visualized in a way that enables users to fully understand where they are or where they are supposed to go with just one quick look.

3.1.2 Target Audiences

As mentioned above, the main target audience for the routing application are university students, with the focus being on the students who are new to the campus. No previous knowledge is needed to use it, so everyone who knows how to use a computer and has access to one will also be able to navigate the routing tool. Another target audience is the people working on the campus, such as professors, security personnel and cleaners. They too might have problems finding their way on the campus when they start to work there, and they are also able to easily use the routing tool.

3.2 Administration Tool

3.2.1 Scopes

The administration tool is used to create the map data that the routing tool uses. It allows to create from scratch or modify all the data that is shown by the routing tool. This includes loading a picture, creating the corresponding map graph and adding information to it's vertices and edges. The main goal of the administration tool is to make this difficult, tedious task simple and quick by providing an intuitive interface and automating many of the time-comsuming steps.

3.2.2 Target Audiences

The map data will most likely be provided by the universities for their students. Thus, the target audience for the administration tool are people working for the university. While the tool is mainly self-explanatory, it is also very versatile. The graph can be edited in a lot of ways, so it takes some time to completely grasp the whole functionality of the administration tool. However, this learning curve is not very steep so even the university students will not have a problem learning how to use the tool in a short amount of time. This makes them the second target audience: students at universities where no good, easily accessible map exists can use the administration tool of our application to create one and share it with their colleagues.

4 Product Environment

Both the routing tool and the administration tool will run on any decent Linux-PC with Java 1.7 installed on it.

5 Functional Requirements

5.1 Main Functions

5.1.1 Map Component

Necessary for the mandatory criteria. -?

Used in Routing and Administration Tool

- **/FM010/** Map is viewed on the screen.
  The user should be able to view a part of the underlying graph of the map, highlighted in a color (e.g. the calculated route) ??? Highlighting???
- **/FM020/** Change viewpoint: ~~Heading User can change his viewpoint by draging the mape~~
  The user should be able to view a section of the map and be able to upscale, downscale and move this section.
  - ~~the~~ Paning function
  - ~~User can~~ zoom in and zoom out function.
- **/FM030/**
  The user should be able to select a vertex or an edge, leading to show this vertex or this edge in another color. (Why this for the routing?)
- **/FM040/** separate functions
  If the user zooms in to a building, he can click to view the different floorplans of the building and change between them
  - User can change his view to a different map associated with a building by clicking on the building.
  - User can go back in his views?

Used only in Administration Tool

- **/FM050/**
  The user should have a view of the underlying graph of the map

5.1.2 Routing

Necessary for the mandatory criteria.

- **/FR010/**
  The user should have a view of the whole implemented campus-plan and the possibility to zoom and pan on this map (also without an active route) This is already mentioned. More looks like a map functionality.
- **/FR020/**
  If the user zooms in to a building, he can click to view the different floorplans of the building and change between them Very similar to FM040.
- **/FR030/**
  The user can type in a starting point and an end point into two different corresponding textfields where he wants to route to and a button to begin the calculation of the (shortest) route between these two points for pedestrians
- **/FR040/**
  The data in the textfields can be: a building number, a building number combined with a room number, an address of a building (road and housenumber) or the name of the building/room/location

- **/FR050/**
  After a route is calculated the user sees on the map the calculated route as a colored path, there is also displayed the distance from that route

- **/FR060/** Text fields are editable? Not very clear functionality?...
  If the user doesn't want the route any more, he can can calculate a new one by typing in the new positions in the textfields or he can abort it, to clear the map with the old displayed route

- **/FR070/**
  There is a location search function to look only for a location where this is on the plan without a concrete startpoint, this is made if only in the destination textfield is a content. Then the position is marked on the map ??
  Search for a point?

5.1.3 Administration Tool

Necessary for the mandatory criteria.

- **/FA010/**
  If a edge or a vertex is selected, information tags about the current selected vertex or edge are shown in a seperate part of the GUI where the user can edit this information tag and add new information tags. Separate information for vertices and edges.
  These information tags are: Name, address, link to building, links to other vertices, contains stairs (needed for wheelchair users), opening hours and some additional text

- **/FA020/**
  The user can open and save a map of the campus or of a building and its underlying graph.

- **/FA030/**
  The user can create a new map.

- **/FA040/**
  The user can exchange the background image of the map

- **/FA050/**
  The user can create, move and remove buildings, (vertices) and edges by clicking on the map.
  remove/modify the ✓ ?

- **/FA060/**
  Then user can create links between vertices of different maps
  using mouse

  In order to store maps seperately and connect them with each other, vertices on different maps will be identified with each other by linking them. I.e., it is possible to split the map of the whole campus into smaller maps and put them together. (TODO: Maybe it will be helpful to show two maps at the same time in the GUI in order to select the vertices identified with each other?) You probably need smth like this to add floorplans.
  But how to implement stairs then?

5.2 Facultative Functions

Necessary for the facultative criteria.

5.2.1 Routing

- **/FR080/** Why this is in routing? Does not sounds as a functionality
  The user should be able to get the application as an Android App or as a Java Applet???

- **/FR090/** Show current position.
  There is the possibility to show the actual position of the user on the map by using a GPS signal
  Not in criteria.

- **/FR100/**
  The user can navigate to a destination from his actual position by using GPS to get the position
- **/FR110/** GPS coordinates? Why?
  To search a location the user can type in also the concrete coordiantes of it into the texfield
- **/FR120/**
  The user can ~~change~~ the routing mode between pedestrian, bicycle and wheelchair ~~users~~ switch
- **/FR120/** Text-based output can be viewed
  If there is an active route, the user can activate also a textbased route output so that there is not only the map in with the route is displayed
- **/FR130/**
  If there is an active route, the user can enter his own average speed, so that the average time needed to go along this route can be calculated and displayed
- **/FR140/**
  The user can mark a route as a favorite, so the route will be saved an can be reused antoher time
- **/FR150/**
  There will be the possibility to calculate and show different alternative routes
- **/FR160/**
  The user can mark different points on the map and than make route from a start point to a destination over these points
- **/FR170/**
  There is a function to search from a given point for nearest points of interests (e.g. restrooms, etc.)
- **/FR180/** Similar to FR120.
  Offer for the user more efficient routing options like a filter (e.g. use no stairs, or make routing through buildings if this is shorter than outside(displayed otherwise), etc.)

5.2.2 Administration Tool

Necessary for the facultative criteria.

- **/FA070/**
  When creating a new map, the user can choose to import OpenStreetMap data
- **/FA080/**
  The user can ~~see~~ and edit the information tags of vertices and edges in a separate area of the GUI (? -> GUI)
  for vertices or edges?
- The user can define new categories of information tags (i.e. "contains stairs", road temporarily closed)
- **/FA090/**
  The user can define a new set of filters for edges and vertices and save it

  These filters can be of the form "has property" or "has not property"
  What for?

6 Nonfunctional Requirements

6.1 Routing

- /NF010/
  Performance: compute the shortest route in less than one second.
- /NF020/
  Performance: GUI reacts to the user-input in less than one second.
- /NF030/
  Reliability: when searched for the destination via user input, error rate is < 10 %.
- /NF040/
  Flexibility: the using of the Java-technology ensures the runnability of the product on all common operating systems.
- /NF050/
  Flexibility: The user interface is exchangeable.
- /NF060/
  Usability: a high user-friendliness is achieved through a simple and intuitive understandable, graphical user Interface.
- /NF070/
  Usability: All menus must have a consistent format.
- /NF080/
  Maintainability: maintainability and expandability of the software should be possible without a high costs.
- /NF090/
  Stability: Arbitrary keypress doesn't lead to the system crash.
- /NF100/
  Stability: There can be no unhandled exceptions from incorrect user input.

6.2 Administration tool

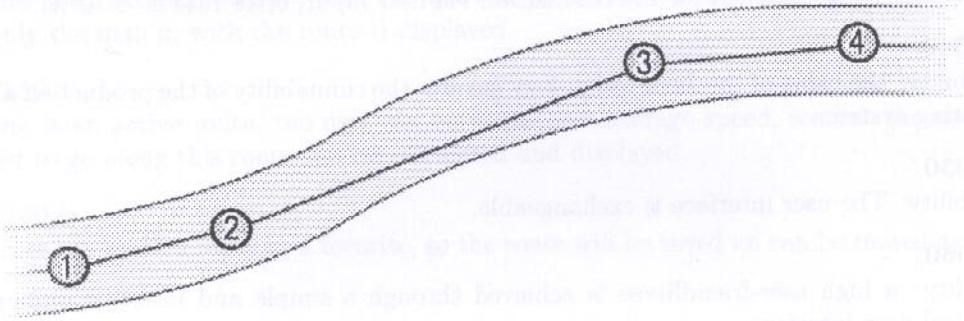
- /NF110/ PDF?
  Flexibility: PNG and JPEG image files can be submitted for use as a map. To product data.
- /NF120/
  Flexibility: The user interface is exchangeable. what do you mean?
- /NF130/
  Reliability: software crash doesn't cause losing the saved dates.
- /NF140/
  Maintainability: maintainability and expandability of the software should be possible without a high costs.
- /NF150/
  Usability: 90 % of novice users can learn to operate major use cases without outside assistance
- /NF160/
  Usability: The product shall use symbols and words that are naturally understandable by the user community.

7 Product Data

7.1 Essential Data

Deprecated! We are going to change the format. You can ignore /D010/.

- **/D010/**
  The graph data in the OSM XML format. We chose this format because it's human-readable, easy to understand and easy to modify. The basic structure is as follows:
  Consider the following excerpt from a map graph:



In OSM XML, the graph could be represented like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<osm version="0.6">
        ...
        <node id="4" lat="54.0901712" lon="12.2482612"/>
        <node id="1" lat="54.0901453" lon="12.2482255"/>
        <node id="2" lat="54.0901233" lon="12.2482561"/>
        <node id="3" lat="54.0901546" lon="12.2482782">
                <tag k="amenity" v="bicycle_parking"/>
        </node>
        ...
        <way id="743">
                ...
                <nd ref="1"/>
                <nd ref="2"/>
                <nd ref="3"/>
                <nd ref="4"/>
                ...
                <tag k="highway" v="footway"/>
        </way>
        ...
</osm>
```

As seen in the example, key/value pairs can be assigned to nodes or ways as node-tags and way-tags, e.g. node 3 that has been defined as a bicycle stand. A different OSM XML-tag, the <relation></relation>-tag, can be used to specify relations between ways and nodes that are unrelated to routes, e.g. building outlines.

For a complete documentation, see `http://wiki.openstreetmap.org/wiki/OSM_XML`. However, we might need to make minor changes to the format to make it suit our needs. The details of

our format will be presented in the next document.

- /D020/
  The picture of the map. Formats where mentioned before.
- /D030/
  The session data. This includes information about what data has to be automatically loaded when the program starts. For example, when the user loads a map for routing and ends the program, that map will be automatically loaded the next time they start the program.

7.2 Facultative Data

Filter, Bookmarks, (new) Tags

8 Global Test Cases

8.1 Routing - Test cases

8.1.1 Searching

- /**T010**/ Search location by a lecture hall name.
- /**T020**/ Search location by a building number.
- /**T030**/ Search location by a address.
- /**T040**/ Search location by a building number and a room number.
- /**T050**/ Search location by a invalid input.

8.1.2 Computing shortest path

- /**T080**/ Compute the shortest path between the starting point and the destination by covering all possible combination of them (see Searching).

8.1.3 Visualization

- /**T090**/ Display a map of the campus.
- /**T100**/ Zoom the map.
- /**T110**/ Pan the map
- /**T120**/ Display the entered location.
- /**T130**/ Display floor plan of a building, by entering any room number or lecture hall name inside of this building.
- /**T140**/ Display the shortest path between a starting point and destination by covering all possible combinations of them (e.g. display the shortest path between Audimax and SR118 in the building 50.34) This is similar to T050.

8.2 Administration tool - test cases

8.2.1 Map manipulation

- /**T150**/ Load a map.
- /**T160**/ Store loaded map.
- /**T170**/ Create a new map.
- /**T180**/ Zoom and pan the map.

8.2.2 Information manipulation on map

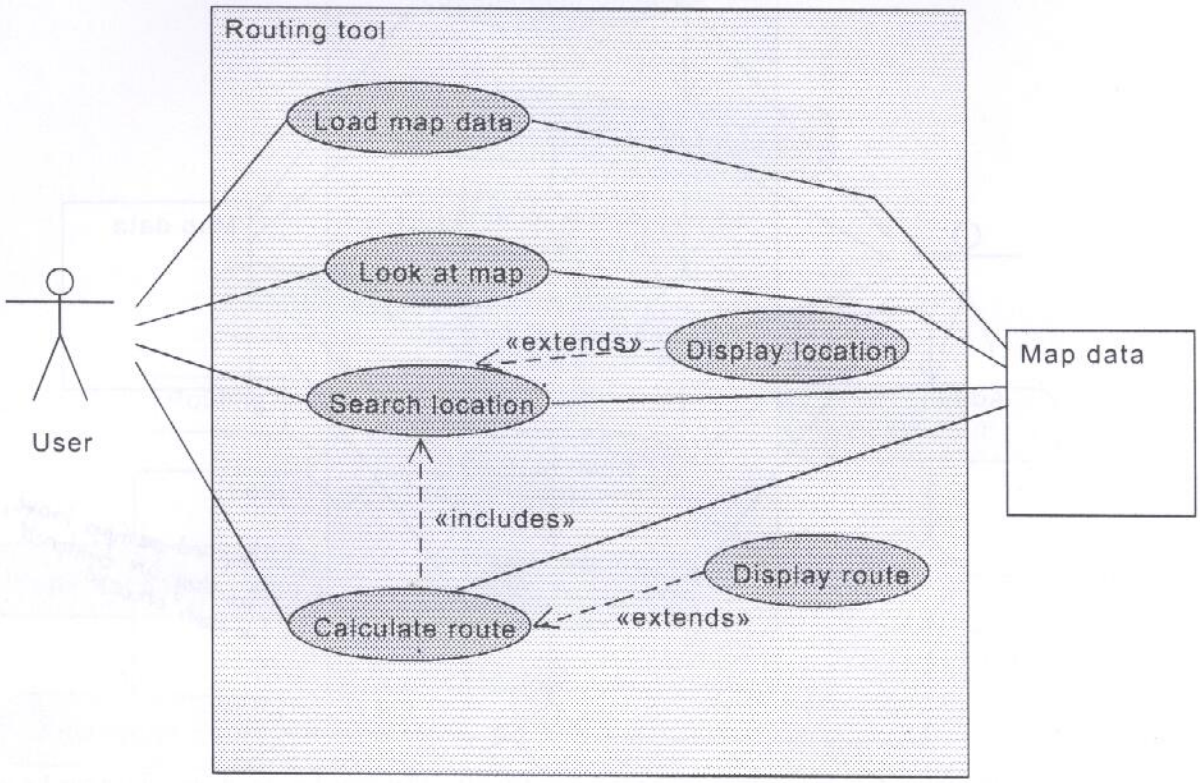
- /**T190**/ Delete/Add buildings.
- /**T200**/ Add floor plans to a building.
- /**T210**/ Delete/Add vertices/edges.
- /**T220**/ Move vertices/edges.
- /**T230**/ Edit Information tags.

9 System Model

9.1 Use Case Diagrams

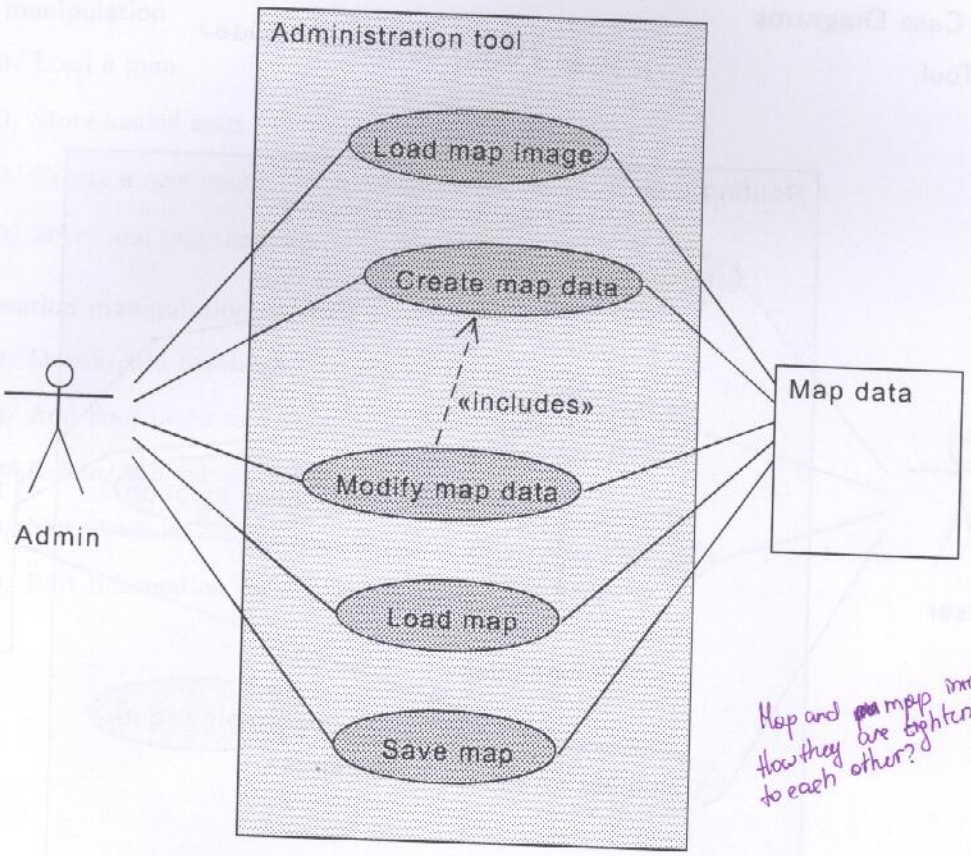
Routing Tool

Model / View / Controller overview



After starting the program and loading the map data, a zoomed out map of the area is displayed. Now the user can look at the map without entering positions. It is possible to zoom into the map, move the displayed section and click on buildings to show their floor plans, when they are provided by the map creator.
If he wants to, he can enter a position on the map. This position can be given as coordinates, address, the name of a place or by clicking on the wanted position. By entering only one position, the position will be displayed at the map but when he inputs a second position a route between the locations can be calculated and is displayed. The search criteria can be further specified by filtering for some tags, for example only stair-free routes can be allowed.
To provide this functionality the program needs to have access to the map data that has been created with the administration tool previously.

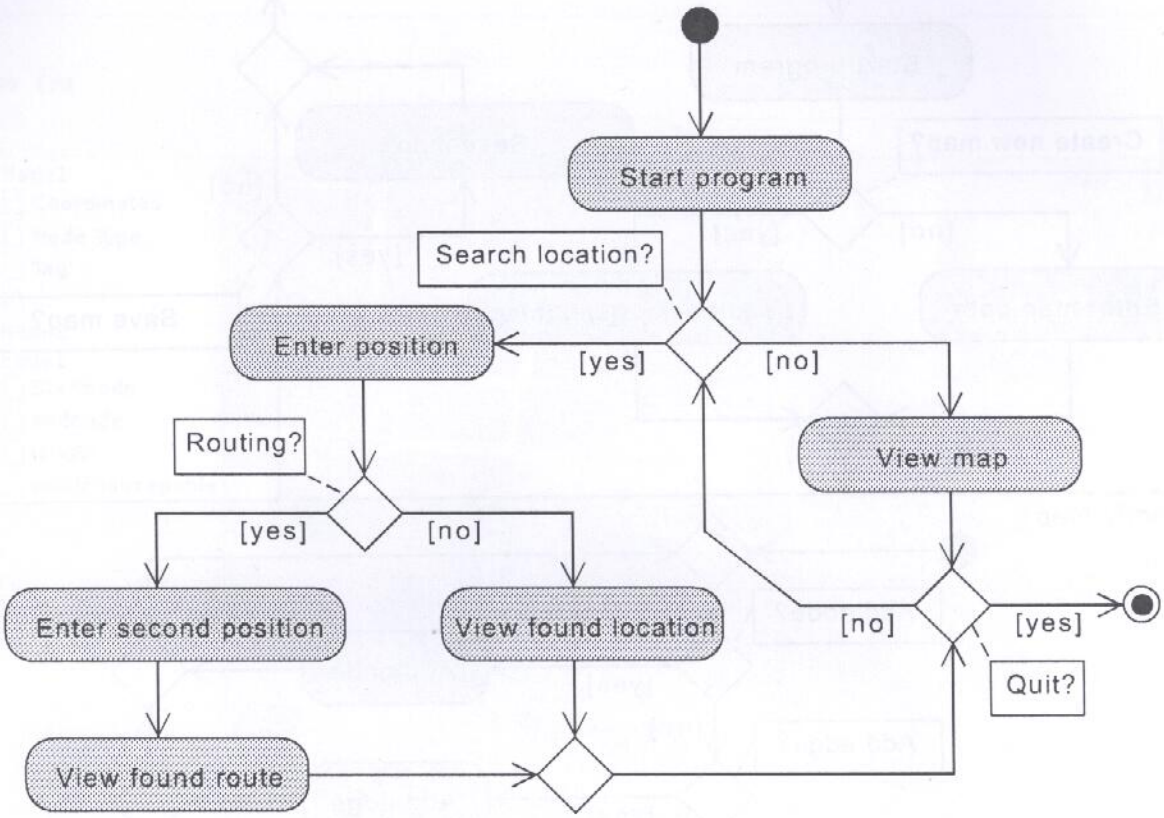
Administration Tool



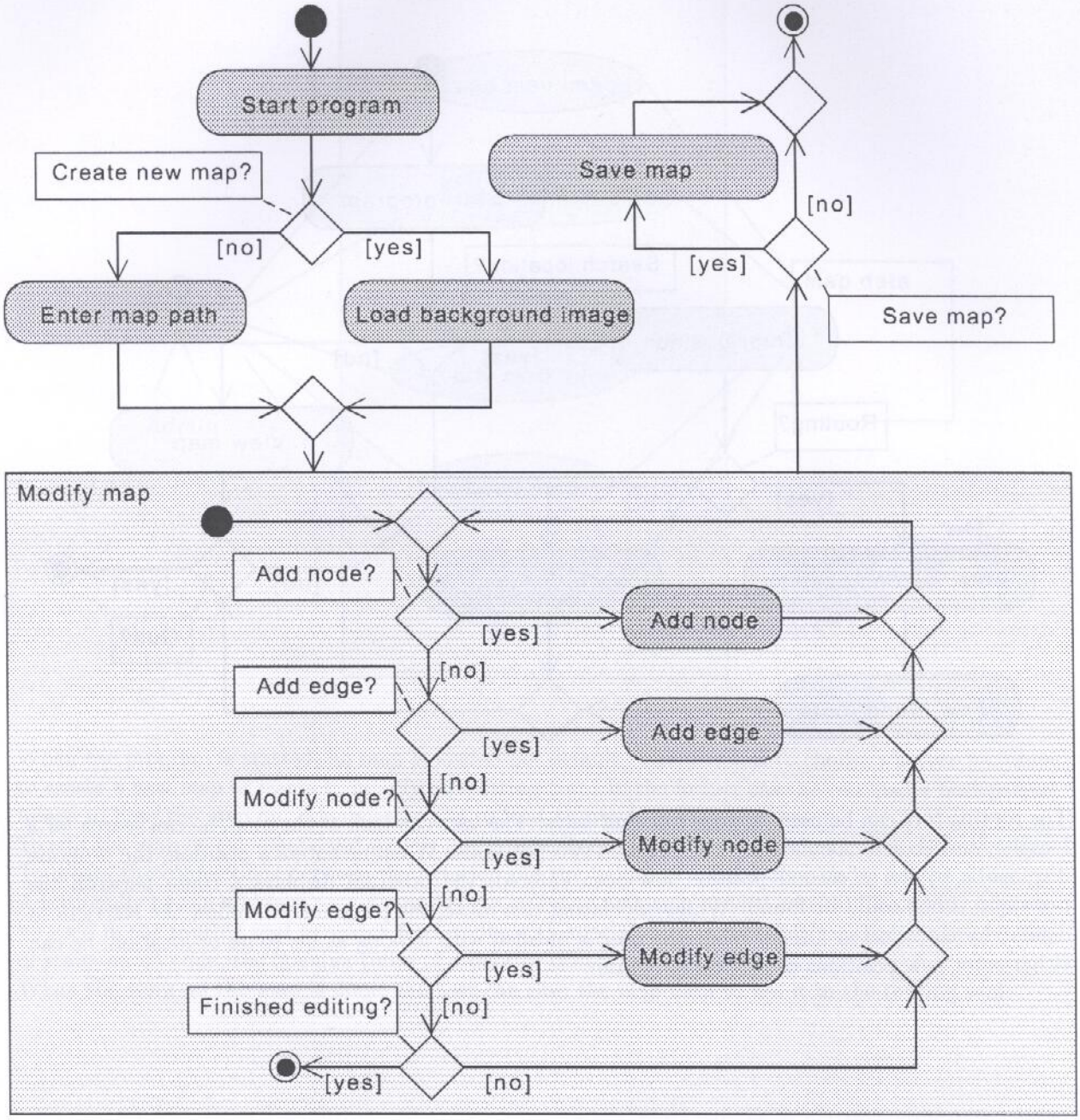
When the program is started, no map is loaded per default. The user has to choose whether he wants to create a new map or wants to modify an existing map. In the former case he can load a background image to base the map on, otherwise he has to load an previously saved map.
To edit the map he can create and modify paths, which consists of the modification of nodes and edges. To create the connections between the exterior and the buildings special door-nodes can be added. Additionally he can add tags to nodes and edges or delete no longer needed entries. The added tags can be used in the routing tool to search for a route with special criteria; possible tags are the availability of stairs on an edge, the opening times of a door or other, user defined, tags.
When the work on the map is done, the user can save the map data to use it in the routing tool.

9.2 Activitiy Diagrams

Routing Tool



The rooting tool can be used in three different ways: The user can look at the map, he can search for a location and he can calculate a route between two positions. When he enters a position, the program displays it. After a second position has been entered, the program displays a route between the positions, if one could be found. While calculating this route, the program will follow the restrictions defined by the selected filters. When the user is finished with looking at his route or location, he can either enter new positions or exit the program.



The first decision after starting the editor is whether to load existing map data or to load a background image to create a new map. After doing one of these, the map can be modified. To do this the user can add, edit or remove nodes or edges between nodes. The tags of existing nodes and edges can be edited to attach further information to them, e.g. adding the name of a building to a node. When the user has finished editing the map, he can save his changes or drop them if they are not useful.

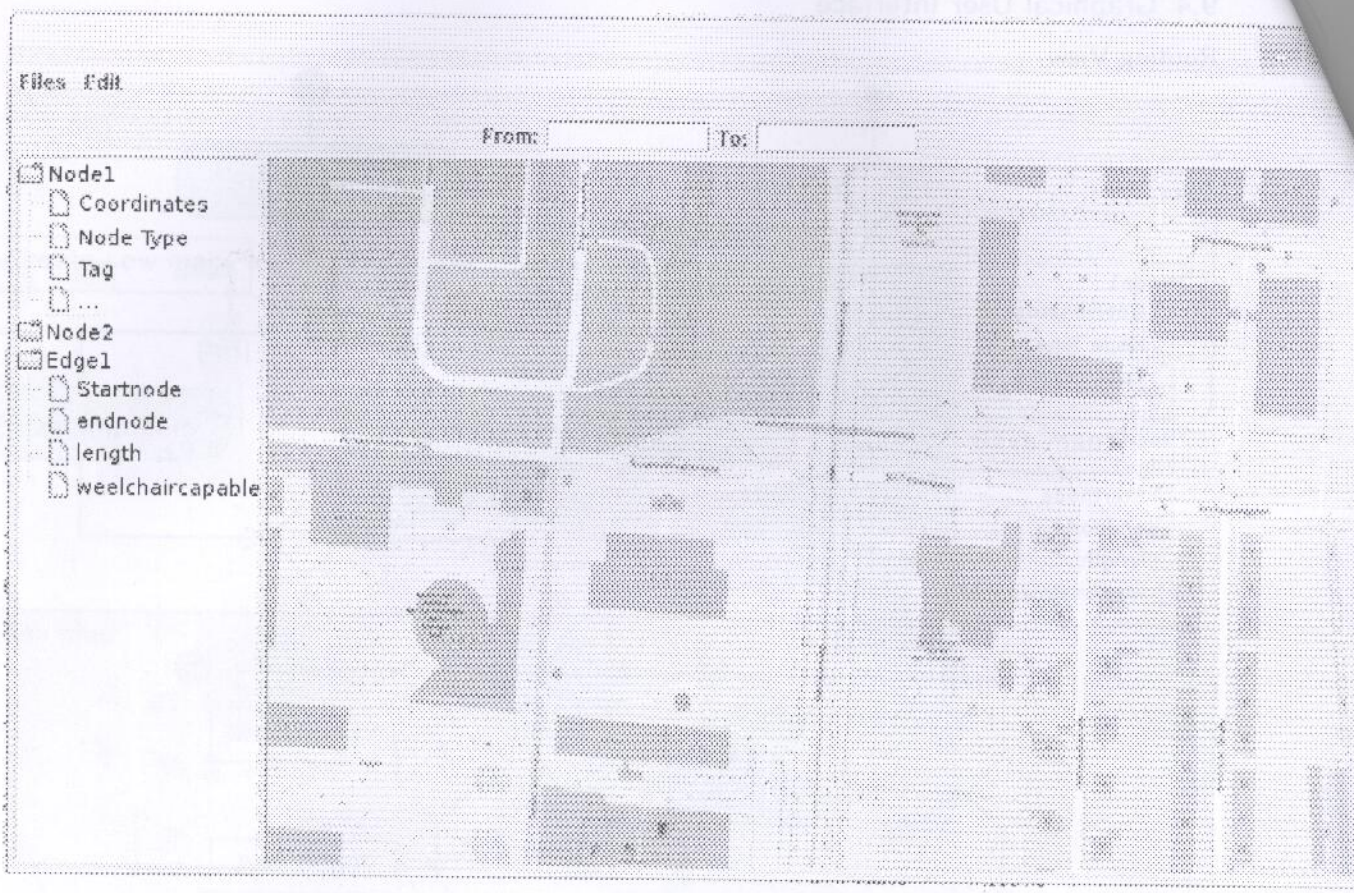
9.3 Graphical User Interface

9.4 Graphical User Interface

Routing View

The routing view has two text fields labeled "From: " and "To: " for inserting the start and endpoint of a route. The main area below shows the map and the calculated path of the route.

Administration View



The administration has the option to load and save map graphs and background pictures via the Files dialogue. In the toolbar below will be an item collection of the available graph modifying tools ans a search field for locations. On the right side will be a list of all nodes and edges with sub entries of their tags. The main area shows the editable graph of the map and the underlying background image.