Finished Project of “YourRoute“:

1 Task Description (Mila)

2 Production Steps (Liu, Fabian)

- 2.1 OSRM – Occurred Issues (Liu)
- 2.2 BRouter using Linux (Fabian)
  - 2.2.1 General Description
  - 2.2.2 Used Data / Installation
  - 2.2.3 System Architecture
  - 2.2.4 License
- 2.3 OpenRouteService (ORS) (Liu)
  - 2.3.1 General Description
  - 2.3.2 API Key and Request
  - 2.3.3 License

3 Backend (Alberto, Benedikt)

- 3.1 Code-based Solutions
- 3.2 Technical Background

4 Frontend (Alberto, Benedikt)

5 Using Code by Others – Requirements (Alberto)

6 Possible Improvement of App (Mila)

Finished Project of “YourRoute“:

1 Task Description

In slide only maybe:

WebGIS Application for crisis routing (blocked areas, ...) using OSRM, pgRouting or Brouter

What to say:

- Aim of this work: Devlope a web-application for crisis-routing
- This task does not only include all those “normal” functionalities, but also the definition of so-called no-go areas, which are areas that should be avoided during the routing
- So if a computed route is passing through one of these blocked-areas, the route has to be recomputed around this area
- Group members who had to do with this: Alberto Rodrigo Martínez, Benedikt Futterer, Chenfeng Liu, Fabian Finkbeiner and Lyudmila Gorokhova

2 Production Steps

2.1 OSRM – Occurred Issues

- Regarding Milestone 1 we have developed a project plan and we have designed some first mockups
- But the main task of this milestone was to do some researches about the pros and cons of the 3 services OSRM, pgRouting and BRouter, also regarding the system architecture and the license
- Another task was to create a website for implementing the app and to document the project work
- Beside the decision what vehicles to use and which country resp. which area to show within OpenStreetMap, we also had to decide which service to use and therefore we first of all have chosen OSRM
- Installation of frontend and backend of OSRM with Windows

- Functionality for blocked areas doesn´t exist (because no on-the-fly calculation → it´s precalculated)

(see table including all possible routing services together with their appropriate functionalities)
- Functionality for blocked areas only possible for BRouter in form of circles and for OpenRouteService (ORS) in form of polygons

2.2 BRouter using Linux

2.2.1 General Description

- Routing engine designed to calculate primarily optimal cycling routes using OpenStreetMap and elevation data
- Programming language: Java

2.2.2 Used Data / Installation

Linux
- Getting Linux on PC first (e.g. in form of Virtual Box)
- Because also for installing OSRM everything was described in the internet for Linux and not for Windows and so it was very difficult and time-consuming to install that with Windows

Used Data
- BRouter uses user-generated, collaboratively collected free geodata (thanks to OpenStreetMap.org and contributors), along with SRTM for terrain data (SRTM 90m Digital Elevation Data)
- Uses highly configurable routing profiles which may be selected or customized
- Just calculates tracks as GPX-Files (it does not display any map or give any navigation instructions)
  → Therefore need of a map-tool in order for BRouter to be useful
- Currently, BRouter cooperates with 3 different maptools (OsmAnd, Locus Map and Oruxmaps), so you need to install, and get familiar with, at least one of them

Installation
- Possibility to install the BRouter-App either from Google's Play Store, F-Droid or directly from the website (APK-File contained within the BRouter-distribution zip-file)
- After accepting a base-directory proposal, BRouter creates subfolders relative to this base directory
- Download BRouter-Web as subdirectory “brouter-web“ of the “brouter“-directory
- Steps for downloading described in the BRouter-web-readme-file

- 2 possibilities of downloading BRouter:
  - Downnload complete code from GitHub and compile it (but therefore Java 6 is needed)
  - Download of precompiled one (if you don´t have a version of Java 6 on your machine)
- In addition download of routing data (in our case Andorra, because fewest storage space regarding all countries worldwide)
- In the BRouter documentation it is described how changes in that service can be done

- Download and unzip latest BRouter revision
- Download one or more data file(s) into „segments4“-directory

- Configure URL to “profiles2“-directory
- Set “BR.conf.profilesUrl“ in “config.js“
- Add your API keys for e.g. Bing (optional)
- Add your API keys for e.g. Digital Globe (optional)

Run App
- Run BRouter like it is described in the Brouter-web documentation
    - Start a BRouter in the standalone dictionary
      (to run BRouter on Linux use “./server.sh“, but it´s also possible to run it on Windows and then use “server.cmd“)
    - Serve the Brouter-directory in the brouter directory
- Open one of our webpages (e.g. “home.html“)

2.2.3 System Architecture
- Works fully offline on any Android phone and is interfaced to some of the most popular Android map tools
- Offers alternative route calculations
- Supports via- and nogo-points resp. nogo-areas in form of circles
- Can consider long distance cycle routes
- Routing data is available worldwide with automatic weekly updates
- Computation time of a route increases quadratic to the distance

2.2.4 License
- The “GNU General Public License” is a free, copyleft license for software and other kinds of works (in this case we use the license “GPL v3”)
- Everyone is permitted to copy and distribute verbatim copies of this license document, but changing is not allowed
- This license is intended to guarantee your freedom to share and change all versions of a program – to make sure it remains free software for all its users
- The “General Public Licenses” are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things

2.3 OpenRouteService (ORS)

2.3.1 General Description
- Offers spatial services, that have been created by several users
- In those services free geographic data from OpenStreetMap have been collected collaboratively
- This data is based on open standards by the so-called Open Geospatial Consortium (OGC)
  (OGC is an international organization and the task of it is to make quality open standards for the global geospatial community)
- Currently a service has been implemented within the frame, which allows the user to include maps, but there´s also one for geocoding and another one for routing in the appropriate web pages or applications
- By requesting, accessing and using ORS and the provided API, the user agrees to the appropriate terms of use

Routing

- According to diverse criteria the route service determines travel routes and navigation information

Geocoding
- This service is about transforming a location-description, like e.g. a street address, a postal code or a place name, into a normalized one with a point geometry;
  usually this normalized location-description is placed using Cartesian coordinates, very often latitude and longitude
- To transform a normalized location-description with a point geometry back into textual location-description again there is the possibility of a reverse geocoding;
  so in other words, this reverse geocoding service changes the coordinate of a point into the description of the next enclosing object which surrounds the given coordinate

Accessibility Analysis
- The isochrone service provides a maximum reachability area from any given point for a set duration;
  so the user has the possibility to specify intervals to return a list of isochrones

2.3.2 API Key and Request

API Key

Request
- Regarding ORS we have no code, so we can only send a request
- Actually “Direct Routing Request (via GET)“ is needed, but a possibility would be trying to do it in form of a POST-request

Parameters of “Direct Routing Request (via GET)“:
- start = latitude and longitude of the starting point (e.g. '7.0892567,50.7265543')
- via (optional) = intermediate points (seperated by blanks) (e.g. '7.0920891,50.7295968 7.1044487,50.7247613 7.1049637,50.7298142')
- end = latitude and longitude of the end point
- routepref = routing profile (e.g. 'Car')
- weighting = property of the routing: 'Fastest', 'Shortest', 'Recommended'
- distunit = distance unity of the routing calculation (standard in kilometer 'KM', meter 'M' and miles 'MI')

2.3.3 License
- Content is available under “Creative Commons Attribution-ShareAlike 2.0 license“

3 Backend

3.1 Code-based Solutions

3.2 Technical Background

4 Frontend

Creating Website

Data:
- 5 HTML-files

    - “home.html“ = Starting page
    - “mockup.html“ = Page including mockups, how the project should finally look like resp. which components should be included within the app
    - “result.html“ = Where the actual project work should be included
    - “workflow.html“ = Where the working steps are described more detailed
    - “about_us.html“ = Description of the group members, who have taken part on this project
- 3 CSS-files
    - “stylesheet_home.css“ = For the layout of the starting page
    - “stylesheet_mockup_workflow_about_us.css“ = For the layout of the pages showing the mockups, the workflow of our project and some information about the group members
    - “navigation_dropdown_menue.css“ = For the layout of the dropdown menue
- Additional folder with images (= 5 jpg-files)
    - 1 photo of a mountainous landscape
    - Our company logo within the footer
    - Our 3 mockups

Content of the webpages:
- All pages have
    - Same header photo
    - Same dropdown menue
        - Including the main points “HOME“, “RESULT“, “WORKFLOW“ and “ABOUT US“ to go to the top of the other webpages
        - But also including subpoints of these webpages, that are directly linked to the appropriate themes within these webpages
    - Same footer content
        - Including some links to additional information for the user, that might be helpful to get some more information about the development of the app (like e.g. about the services pgRouting and BRouter, the for the app chosen country Andorra, OpenStreetMap, the company GEOFABRIK, Hochschule Karlsruhe, GitHub, the programming language JavaScript, HTML and last but not least CSS)
        - Including the logo of our company “YourRoute“
- Starting page:
    - Description of the task in general, including a link to the appropriate 3 mockups
    - About our used service OSRM in general and in comparison to the other 2 services pgRouting and BRouter, about its used data and about its license, including also a link to the OSRM homepage
    - On the right side also the names of the group members are listed
- Page about the mockups:
    - Shows 3 mockups with a short description
        - First one showing the starting page
        - Second one an example for car routing with an intermediate target and a blocked area
        - Third one an example for pedestrian routing also with an intermediate target
- Page, where the actual app should be included
    - Also with additional information about the data in form of some descriptive text, but of course also in form of some links to the used sources
- Workflow-Page

    - Should include our project plan, with a detailed description of the milestones and their appropriate working steps
    - Should also include a workload of our project (which working step should be finished at what time and who will be responsible for each task)
  - Webpage about us
    - Should contain a short description of each group member (what and where has the person done his Bachelor studies and what are his qualifications)

5 Using Code by Others - Requirements

6 Possible Improvement of App

- Actually an integration of the ORS-functionality for blocked areas into the BRouter-GUI with all its additional functionalities would be the best solution;
  Our plan was to let the user decide whether he wants to use BRouter or ORS for routing by clicking on the corresponding button.
  If you click on the Brouter-button routing should be done by BRouter and the user should be able to draw those nogo-areas by circles. If you click on the ORS-Button a request should be send to ORS and the answer should be shown into the Brouter-frontend. ORS enables the possibility to draw blocked-areas in form of polygons then.
  (but this doesn´t work in our project because we weren´t able to finish it. If you want to include this functionality you can´t use an iframe.)
- Set the same start point for the BRouter- and ORS-maps didn´t work because permalink-method of ORS is broken