



Development Plan.md 10.8 KB

Development Plan

- [Objectives](#)
- [Approach](#)
 - [Requirements](#)
 - [Constraints](#)
 - [Results expected](#)
- [Features](#)
 - [MOSCOW method of features](#)
 - [Resources](#)
 - [Priorities](#)
- [Design](#)
 - [Design Objectives](#)
 - [Design Strategy](#)
 - [Critical Features](#)
 - [Risk factors](#)
 - [To-avoids](#)
 - [Design validation and evaluation](#)
 - [Test approach](#)
 - [Test planning](#)
 - [Test validation](#)
 - [Data Validation](#)
- [Supporting Information](#)
 - [Code base and documentation](#)
- [Timeline](#)

Objectives

Investigate if there is a relationship between the weather type, and where cyclists prefer to ride.

Context

Strava is widely used by athletes around the world to keep track of their sports activities. With more than 42 million users and growing, it is one of the largest real-time sources of open data on recreational activities. The gps tracks of each activity are uploaded to the platform, and can be shared/viewed/compared with those of other users. One of Strava's key features is so-called "segments" which are portions of road or trail that are frequently used. This allows for users to directly compare times on a leaderboard, because all segments are automatically extracted from each uploaded activity.

We will leverage the segments' data as discrete points in any given area to compare with the corresponding local weather conditions. It is trivial that there will be more user activity on days with nice weather, however this tool will look for a correlation of *where* cyclists prefer to ride depending on the weather.

Approach

First, using the Strava API, segment ranking data is used to calculate daily activity frequency per segment for a given timeframe. Next, weather data from these days and locations are obtained from the KNMI, as well as the KNMI weather warnings. This is used to calculate weather scores based on rain, wind, and the weather warnings. For a given weather type, filter the days with that specific weather type. Then, filter the Strava data for those days and generate a heat map displaying the activities per segment. A reference output is generated, so that further weather dependent outputs can be interpreted.

In order to differentiate between locations that are inherently popular and locations popular due to specific weather conditions, the reference output will be made as a baseline. The exact parameters of this still needs to be determined, but will likely include variables such as day of week, season, and weather.

To prepare the data from Strava, some outliers will have to be removed. For example, some entries can have an entry date in the year 2030, which is obviously incorrect. These data points will have to be removed. Also, there are a couple of segments which were specifically built for cycling events. These segments have been used for only one day, but on that day, often more than 300 cyclist use that segment. This really messes up the average of the segments, so it was decided to remove these segments as well, since they do not provide information on where people cycle depending on the weather conditions.

Different modules will be created:

- A module to obtain and filter data from Strava
- A module to obtain and filter weather data
- A module to analyse the data obtained from Strava and the weather website
- A module to generate heat maps
- A controller module

Requirements

- Obtain weather data and KNMI warnings
- Use Strava API to obtain segment activity data
- Create function to calculate weather scores based on wind, rain, and KNMI warnings
- Date range as input parameter
- Strava activity is specified to cycling
- Pipeline output should be verifiable
- The intensity (total number) of activities must be normalized, to allow for comparison between high and low usage days.

Constraints

- Scope of this tool will be limited to the province Zuid-Holland. This is because a bigger scale (the whole of the Netherlands for example) would result in having to analyze too many segments, which the Strava API does not allow. Also it would take ages to run the code to obtain all the data.
- The results are representative of behavior of Strava users only, thus cannot be generalized for the general population.
- Weather changes throughout the day, so some form of averaging must be made to calculate daily weather score.
- Not all people who do sports use Strava, and not all Strava users publish their activities, so not all activities on the segments are registered
- Strava activity is specified to cycling.
- Reference output to compare and interpret weather dependent outputs.
- Need to take the growth/decline in Strava users over the time frame into account, or recognise it as a possible influencing factor.
- weather data is obtained for the time frame 7 am to 7 pm. This means that a little daylight in which people could go cycling is skipped in the summer.
- the weather stations used to obtain weather data are all located in Zuid-Holland. It is possible that there is a weather station in a different state that is closer to several segments.
- The coordinate grid used to obtain segment information from Strava is specified to 5x5 km. This implies that not all segments will be used, since the Strava API only returns the top 10 segments within a given coordinate square. This is not a very big problem, since most cyclists only use the popular segments, but it does have to be taken into account. It is possible to use a grid that is less coarse, but that also means having to do even more requests.

Results expected

Users will bike different routes based on weather conditions. For example:

- Cyclists will less likely ride on the road in the rain, and will therefore look for routes in the forest or rural areas.
- On days with high wind, coastal areas will have less activity than other regions.

Features

MOSCOW method of features

This is a prioritized list of features that were considered during the initial design phase.

Must Have

- Strava module
- Weather module
- Analysis module
- Controller module

Should Have

- Visualisation Module

Could Have

- Advice where you should bike based on weather predictions
- Interactive heat map

Won't Have

- App to show data results
- Feedback to Strava
- Expand data analysis to outside The Netherlands
- Implementation of traffic signs and jam avoidance

Resources

- Strava api
- KNMI weather data
- Google Maps (results visualization)

Priorities

1. Collect and filter data both Strava and weather data
2. Data handling and preparation for verification
3. Data visualization
4. Data verification

Design

Design Objectives

- Design for maintainability
- Desgin for reusability
- Design for extensibility

Design Strategy

We will be using scrum to develop the software. This is to allow for test driven development, which is essential when developing a software in such a short timeframe. The software will most likely consist or three primary modules: Strava data, weather data, and data processing/visualization. A priority will be to clearly communicate within the team how these modules need to interact with each other. Furthermore, writing and saving tests throughout the development will help prevent updated code from breaking previously tested code.

Critical Features

- module that obtains and filters Strava data
- module that obtains and filters weather data
- function that generates a reference output
- module that displays the data

Risk factors

- reference output is not representative
- data cannot be validated
- Unclear expectations and/or results

To-avoids

- Waterfall method, not responding to change
- not following the plan
- losing view of what other group members are working on
- Not following priorities of what the software must be able to do

Design validation and evaluation

Every week, the team has a meeting to evaluate on the week before and reflect on itself on group dynamics. Possible action points are written down for the next Sprint.

Test approach

- create unit tests using pytest
- use CI pipeline on GitLab
- possible manual testing

Test planning

Because we chose Test Driven Development, testing is done alongside code writing. Also, team members code review each other's code.

Test validation

Test outputs will be compared with expected results. Also, using the points from the data validation section below, we will try to validate the outputs.

Data Validation

We would like to validate our results by comparing the output to logical scenarios. For example:

- Generating a heat map of segment activity for different weather types, and visually comparing
- Recognizing if a KNMI warning is given, there should be very little activities (ie dangerous road conditions)
- Exceptionally strong wind and activities along the coast.

Additionally, we recognize that filtering of obviously false data (users uploading data to the incorrect activity type, incomplete gps tracks, etc.) is already done by Strava.

Supporting Information

Code base and documentation

GitLab will be used to manage code versions, store documentation, and enable a Continuous Integration pipeline.

Timeline

Project group meetings are held every Tuesday morning. The timetable below is thus based on weeks starting on Tuesdays. Meetings with the TA is on Thursday.

Week	Date	Activities
1	12 nov - 19 nov	<ul style="list-style-type: none">Select project ideaMake draft Development plan
2	19 nov - 26 nov	<ul style="list-style-type: none">Discuss Development planFind resourcesFinalize Development planStart Design plan
3	26 nov - 3 dec	<ul style="list-style-type: none">Discuss Design planWork on Design plan
4	3 dec - 10 dec	<ul style="list-style-type: none">Validate Design planPeer Review 1Start implementation
5	10 dec - 17 dec	<ul style="list-style-type: none">Implementation
6	17 dec - 7 jan	<ul style="list-style-type: none">ImplementationMidterm PresentationPeer Review 2
7	7 jan - 14 jan	<ul style="list-style-type: none">Implementation
8	14 jan - 21 jan	<ul style="list-style-type: none">Implementation Wrap-UpValidation
9	21 jan - 24 jan	<ul style="list-style-type: none">Final Presentation

[Next Page \(Design Plan\) >>>](#)