# Web Engineering Project Description

V. Andrikopoulos

March 20, 2019
Version: 1.3

## 1 Introduction

The Web Engineering capstone project allows you to put into practice the material discussed during the lectures and tutorials in groups. The following sections discuss the example application to be used across all group projects for the course, the milestones for the project, and the foreseen deliverables and assessment points.

*Note*: This document is a "living" one, meant to be updated in the following weeks. Use the version identifier for an indication of changes to it.

## 2 Case Study Description

The Web app to be developed helps users to decide which carriers (airlines) and which airports to use in the USA for their air travel needs. It builds on the Airline delays dataset made available by the CORGIS dataset project, itself a curated version of the data published by the Bureau of Transporation Statistics of the US Government. Both a JSON and a CSV version of the dataset are available. **Update Feb 26:** *only the JSON version of the dataset is considered authoritative; use that in order to offer all the endpoints discussed below.* It contains statistics for all reported delays per carrier per airport per month in the USA from 2003 to 2016.

The goal of this project is to deliver to users both basic and advanced features based on this dataset as a Web app. It comes with a minimum set of requirements on what features are to be delivered, and fosters creativity on the app developer side by allowing for identification of further features proposed and implemented by each group.

## 3 Milestones

There are three milestones for this project. The first milestone is common for all groups, while the latter two are specific to the design decisions taken by each group.

### 3.1 M1: API Design

Design a RESTful API that allows for accessing the following data:

1. all airports available in the USA,
2. all carriers operating in US airports,
3. all carriers operating at a specific US airport,
4. all statistics about flights of a carrier *from/to* a US airport for a given month/all months available (‡),
5. number of on-time, delayed, and cancelled flights of a carrier *from/to* a US airport for a given month/all months available,
6. number of minutes of delay per carrier attributed to carrier-specific reasons (i.e. attributes carrier and late aircraft in the dataset)/all reasons, for a given month/all months available and for a specific airport/across all US airports,
7. descriptive statistics (mean, median, standard deviation) for carrier-specific delays (as above) for a flight between any two airports in the USA for a specific carrier/all carriers serving this route.

Entries entries marked with (‡) require support for both retrieval and manipulation (addition, update, deletion) of data through the API; otherwise only retrieval is to be supported. Each API endpoint should support both JSON and CSV representations of the resources (i.e. Content-Type is application/json and text/csv) available at least by an appropriate query parameter. JSON is the default option if none is specified.

### 3.2 M2: Architecture, technology selection & API implementation

Provide a high-level architecture of the Web app, including the technologies to be used and their justification. Implement the API as the back-end of the app using the technology of your choice, and document the implementation code appropriately.

Identify one or more value-added features to be offered through the Web app (*optional*). Each feature has to have sufficient complexity and deliver added value to the users of the Web app. Examples of such features are:

- Rankings of carriers for different combinations of data (e.g. number of delays, duration of delay, ratio of cancellations etc.)
- Visualization of the carrier/airport statistics
- Crowdsourcing of the experiences of users for carrier delays and cancellations

- Option for users to comment about their own experience about delays with a specific carrier (user login etc not necessary), including user ranking of carriers

The use of a database for persistence is optional but recommended.

### 3.3 M3: Web app implementation & demonstration

Implement a UI front-end for the Web app which builds on the implemented API. The goal of this task is to demonstrate the developed features as a complete Web app; graphic and UI design skills will not be evaluated. The use of Web technologies discussed during lectures and tutorials is expected and desired, respectively.

Demonstrate your implementation successfully. The demonstration should walk a hypothetical user through the UI, showing the delivered features (basic and/or advanced) while interacting with the endpoints developed in the previous milestone. Submitting plain HTTP requests to the back-end endpoints should also be possible on request, either by use of a CLI tool like curl or by using an administrative interface to the API as provided by your chosen back-end framework.

Deployment of the Web app for demonstration purposes is at your own responsibility and discretion. If you bring a laptop for this purpose, please make sure you also bring an adapter to HDMI or RGB.

## 4 Deliverables & Assessment

Table 1 contains the deliverables expected per milestone and their associated deadlines.

Table 1: Deliverables and deadlines per milestone

| Milestone | Date | Requirements |
|---|---|---|
| M1 | March 4, 12:00 | API documentation |
| M2 | March 18, 12:00 (soft)<br>March 22, 12:00 (hard) | Back-end software<br>Project report (interim), back-end software (with documentation) |
| M3 | March 29, 09:00 | Project report (final), front- and back-end software (with documentation), demonstration of the Web app |

All reports and software are to be delivered by doing a pull request on a repository that you set up as a group, and to which the TA assigned to your group is added as your collaborator. TA assignment will commence as soon as the group registration finishes.

*Note:* Failure to do a pull request with the foreseen deliverable by the defined deadline results in 0 points for this milestone. The final grade for the assessment is weighted by $(0.2, 0.3, 0.5)$ per milestone M1 to M3, respectively.

### 4.1 Minimum requirements

In order to receive a "passing" grade (5.0)[1], a project has to provide an efficient design and implementation for M1 and M2, respectively. This means that all endpoints identified in Section 3.1 are expected to be available for testing — including during the demo day — and their documentation is appropriate and complete. Furthermore, the delivered Web app in M3 provides a UI front-end for users to interact with all these endpoints. The technologies to be used for the implementation of both the front- and the back-end are on the discretion of each group, but their selection is rationalized as part of the final project report. Finally, all delivered code is clearly structured, and sufficiently commented.

### 4.2 Requirements for higher grade

Assuming that the project fulfills the minimum requirements, additional points will be given for projects providing the users with added value by:

1. implementing additional endpoints to the API and offering them as features in the UI front-end (2 points),
2. adopting one or more of the technologies presented during the tutorials for the lecture (1 point),
3. using at least one 3rd party API to deliver advanced features to the Web app users (1 point),
4. the design and implementation follow appropriate principles and patterns discussed during the course, and the report rationalizes the developers' decisions effectively (1 point).

Bonus points will also be given for delivering advanced features of sufficient complexity not foreseen in Section 3.2.

## Change Log

**Version 1.3** Refinement of Section 3.3.

**Version 1.2** Refinement of Section 3.2.

**Version 1.1** Clarifications in Section 3.1.

**Version 1.0** Initial version of the project description.

[1] *Note:* Please do notice that as per the Ocasys entry for the course, there is no formal minimum for the course assignment, only the one imposed by the grading formula for the course.