

Semester Project for the Course "Software Engineering"

NTUA - School of Electrical and Computer Engineering

Fall semester 2026-2027

Instructor: Prof. V. Vescoukis

Thematic areas

The thematic area of the project will be determined by each team, which is required to choose the subject, purpose and functionality of a software application to be implemented to the extent and according to the instructions that follow. For this purpose, 39 data sources grouped into 9 categories are proposed:

- | | |
|--|--|
| 1. Weather and meteorological data (9 sources) | 6. Statistical data (4 sources) |
| 2. Energy (6 sources) | 7. Geospatial data (GIS) (6 sources) |
| 3. Aviation (4 sources) | 8. Culture and media (5 sources) |
| 4. Natural disasters (3 sources) | 9. Tourism and short-term rentals (1 source) |
| 5. Environment and air quality (1 source) | |

Each of the above sources provides data through an application programming interface (API), a web service, or in csv format, etc. Each team will select 1 - 3 sources from one or more categories, in a way that the subject of a software application can be defined. The data sources are given in the Appendix; note that other sources providing data that fall into the same or into new categories may also be used, provided that they have feasible and lawful access.

For the selected application, each team will specify the corresponding requirements, define the architecture, carry out the detailed design, the authoring of the source code, the testing, as well as the documentation of all the above, according to the instructions in this document, the course lectures and the material posted on helios. All teams will build the application they select using both architectures: layered and Model-View-Controller (MVC). The following are to be included in each architecture:

A. Layered architecture

1. A back-end subsystem for data management and services.
2. A set of REST API programming interfaces for providing services and data, whose endpoints you will define and include in the documentation. These interfaces will be used a) for communication between the individual components of the application and b) for making selected services available to third parties over the internet.
3. A CLI (Command Line Interface) for implementing a way to control selected functions of the application through a shell. It will make use of the REST API interfaces mentioned above.
4. A web user interface (front-end) that will offer the user the functions to be implemented, with responsiveness a) for desktop browser and b) for smartphone.

B. Model-View-Controller (MVC) architecture

1. A back-end subsystem for data management and computation functions (Model & Controller).
2. A set of REST API programming interfaces for providing services and data, whose endpoints you will define and include in the documentation. These interfaces will be used a) for communication between the individual parts of the application and b) for making selected services available to third parties over the internet.
3. Two web user interfaces (front-end), each implemented as a separate View of the MVC architecture. Each interface will offer the user all the functions to be implemented. The first will be intended for a desktop browser and the second for a smartphone (a native smartphone application is not required; a separate page implementing the corresponding View is sufficient).

It is recommended that the project be carried out in **teams of 4 people**. Teams with fewer than 4 and with up to 6 people are accepted, with the deliverables adjusted as shown below.

Technical requirements - tools

For the **backend** any development stack with python or js with node/express may be used. For **data management** one of PostgreSQL and Mongo. For **frontend** the choice of frameworks is free.

In addition, the following are mandatory:

- **Github** for code management, **in a way that is integrated with the AI tool(s)** (Claude, Chat, Gemini etc.) that you will use, as will be indicated in the slides and the lectures. Each team member must use their own GitHub account.
- **PlantUML & markdown** for producing the individual diagrams and models as code (diagram-as-code), as well as the specification document (SRS), with support from AI systems. Documents in word format or diagrams in image format will not be accepted.
- **Visual Paradigm CE or Enterprise trial** (desktop version) for creating the complete model and documentation of the software using UML. The use of other UML diagramming tools is not permitted.
- **Github projects** for project management.

Deliverables

DELIVERABLE	Teams (people)		
	1-2	3-4	5-6
Architecture and documentation (Applies to both architectures: layered & MVC)			
SRS document	3 use cases	5 use cases	7 use cases
Data design diagrams	YES		
UML Activity / State diagrams	YES		
UML class & API diagrams	class & API		
UML component & deployment diagrams	YES		
Unified UML model in Visual Paradigm	YES		
Implementation (Applies to both architectures: layered & MVC)			
Data management and services (backend)	YES		
Data dump	YES		
RESTful API	YES		
API documentation	YES		
Command Line Interface (layered)	YES		
Frontend implementation	2 use cases	3 use cases	4 use cases
Testing (Applies to both architectures: layered & MVC)			
API functional tests	YES		
CLI testing	YES		
Performance testing	YES		
Tools			
Code management	YES		
Project management	YES		

Deliverable formats

The table below shows the formats of the deliverables, the required file names, the storage folder within the GitHub repo of the project, as well as what is included in the intermediate and final submission.

DELIVERABLE	Format	Deliverable file name	Submission 30.11.2026	Final submission	Deliverable storage location
Documentation					<i>github.com/ntua/softeng26-XX/ +</i>
SRS document	markdown with plantuml	srs-softeng26-XX.md	X	X	documentation/requirements
Data design diagrams	plantuml	softeng26-XX- [diagram_id].puml	X (conceptual only)	X	documentation/diagram-as-code/mvc and documentation/diagram-as-code/layered
UML Activity / State diagrams			X	X	documentation/diagram-as-code/mvc and documentation/diagram-as-code/layered
UML class & API diagrams			-	X	documentation/diagram-as-code/mvc and documentation/diagram-as-code/layered
UML component & deployment diagrams			-	X	documentation/diagram-as-code/mvc and documentation/diagram-as-code/layered
Unified UML model in Visual Paradigm	Visual Paradigm (vpp)	softeng26-XX-mvc.vpp and softeng26-XX-layered.vpp	-	X	documentation/visual-paradigm
Implementation					<i>github.com/ntua/softeng26-XX/ +</i>
Data management and services (backend)	Source code	(according to the development environment)	-	X	mvc/backend and layered/backend
Data dump	SQL or json		-	X	mvc/data and layered/data
RESTful API	Source code		-	X	mvc/backend and layered/backend
API documentation	Postman scripts, OpenAPI yaml		-	X	documentation/apis

DELIVERABLE	Format	Deliverable file name	Submission 30.11.2026	Final submission	Deliverable storage location
Command Line Interface (layered)	Source code		-	X	mvc/cli and layered/cli
Frontend implementation (desktop)			-	X	mvc/frontend and layered/frontend
Smartphone frontend version			-	X	
Testing					<i>github.com/ntua/softeng26-XX/ +</i>
API functional tests	test scripts	(according to the development environment)	-	X	mvc/testing and layered/testing
CLI testing			-	X	
Performance testing			-	X	
Tools					<i>github.com/ntua/softeng26-XX/ +</i>
Code management	-	-	X	X	layered/* and mvc/*
Project management	-	-	X	X	projects

Grading and weights

For Erasmus students, the grade is derived from the project and its oral presentation. The partial weights per deliverable are shown in the table below.

DELIVERABLE	Relative weights	
Documentation: 45%	layered	mvc
SRS document	10%	
Data design diagrams	5%	5%
UML Activity diagrams	5%	5%
UML State diagrams	5%	5%
UML class diagrams	5%	5%
UML API diagrams	5%	5%
UML component diagrams	5%	5%
UML deployment diagrams	5%	5%
Unified UML model in Visual Paradigm	10%	10%
Implementation: 25%		
Data management and services (backend)	20%	20%
Data dump	5%	5%
RESTful API	5%	5%
API documentation	5%	5%
Command Line Interface (layered)	10%	
Frontend implementation (desktop)	5%	5%
Smartphone frontend version	5%	5%
Testing: 10%		
API functional tests	15%	15%
CLI testing	15%	15%
Performance testing	20%	20%
Tools: 10%		
Code management	50%	
Project management	50%	
Overall impression: 10%		
TOTAL: 100%		

The grades of team members may differ according to their recorded contribution.

Deadlines

Team declaration: 31.10.2026. Timely declaration and participation in teams is necessary for recording each student's contribution to the project, which will begin on **1.11.2026**.

Submission 1: 30.11.2026. The complete SRS document and diagrams as shown in the table "Deliverable formats", on GitHub.

Final submission: 10.1.2027. All deliverables on GitHub.

APPENDIX: Catalogue of Data Sources

This catalogue brings together indicative data sources: open APIs, downloadable datasets and web services, grouped in 9 thematic categories. Each group can choose one or more sources and use them to define the thematic area of their own assignment. The catalogue is not binding: another source may be proposed, provided its availability and terms of use are documented.

1. Weather and meteorological data

Open-Meteo

Open API for weather forecasts and historical weather data, drawing on more than 30 national meteorological models and historical records going back to 1940 (ERA5). It also includes air quality, marine and climate projection data.

Access: REST API; no registration or key required for non-commercial use (up to roughly 10,000 calls/day); JSON, CSV, XLSX.

Link: <https://open-meteo.com/>

Meteostat

Open platform providing historical weather and climate data from stations worldwide: hourly, daily and monthly values, as well as climate normals.

Access: JSON REST API via RapidAPI (key required; free plan of 500 calls/month) or bulk file download without registration; JSON, CSV. A Python library is also available.

Link: <https://dev.meteostat.net/>

openSenseMap

Platform for publishing and visualising environmental measurements from citizen-operated sensors ("senseBoxes"): temperature, humidity, pressure, particulate matter, UV and other quantities, with both current and historical values.

Access: REST API; no key required for reading (a key is needed only to submit measurements); JSON, GeoJSON, CSV.

Link: <https://docs.opensensemap.org/>

Xweather - Weather API

Commercial weather data service (Vaisala) covering current conditions, forecasts, severe weather alerts, air quality, maritime and fire weather data, as well as lightning strike data.

Access: REST API with a key (client ID and secret); free tier of 15,000 calls/month with no credit card; JSON, GeoJSON, CSV, TSV.

Link: <https://www.xweather.com/products/weather-api>

Xweather - MapsGL

Weather mapping SDK with more than 70 preconfigured layers (radar, satellite, forecast fields, alerts) and a timeline spanning historical, real-time and forecast data.

Access: SDK for JavaScript, iOS and Android, layered on Mapbox GL, MapLibre, Google Maps or Leaflet; requires an Xweather key and an active subscription.

Link: <https://www.xweather.com/products/maps-gl>

Xweather - Phrases API

Turns weather data into natural language descriptions in more than 50 languages, through three endpoints: current conditions, alerts, and activity-specific impacts.

Access: REST API with a key; free 30-day trial, paid thereafter; JSON.

Link: <https://www.xweather.com/products/phrases-api>

meteo.gr - Weather station network (National Observatory of Athens)

Interactive map of the National Observatory of Athens automatic weather stations across Greece, showing each station's location and current observations, along with specialised maps (wind, rainfall, pressure, UV).

Access: Web page with no documented API or bulk download; HTML only. Programmatic use would require scraping, subject to the Observatory's terms of use.

Link: <https://meteo.gr/Gmap.cfm>

meteo.gr - City forecast page (example: Athens – Centre)

Current conditions from nearby stations, a four-day forecast, three-hourly detail, climate values and station records, plus pollen and dust data. The page is parameterised by a city identifier (city_id).

Access: Web page with no API; HTML only.

Link: https://www.meteo.gr/cf.cfm?city_id=12

meteo.gr - Live observations map

Colour-coded map of Greece rendering a selected quantity (for example daily maximum temperature) across the National Observatory of Athens station network.

Access: Web page with no API; HTML only. The page is protected by an automated-access challenge.

Link: https://meteo.gr/greece_heat_Map.cfm?metric=highTempC

2. Energy

ENTSO-E Transparency Platform - RESTful API

Pan-European electricity market and system data published by Transmission System Operators under Regulation (EU) 543/2013: load, generation by unit type, cross-border flows, balancing, outages and prices.

Access: REST API; free registration and a security token are required (approval takes about three working days); XML (IEC 62325-351 market documents), ZIP for large responses.

Link: <https://documenter.getpostman.com/view/7009892/2s93JtP3F6>

ENTSO-E - RESTful API integration guide

Index of the official documentation for the API above: endpoints and request methods, parameter lists (document types, PsrType, EIC area codes), query size limits and security topics.

Access: Documentation (HTML); not itself a data source.

Link: <https://transparencyplatform.zendesk.com/hc/en-us/articles/15692855254548-Sitemap-for-Restful-API-Integration>

IPTO (ADMIE) - Operation & Market Files API

Service for locating and downloading the operation and market files published by the Greek transmission system operator (SCADA system load, load forecasts, market data), searchable by file type and by a date or date range.

Access: Web service with no key or registration; returns JSON containing the URLs of the files, which are spreadsheets (XLS/XLSX/CSV).

Link: <https://www.admie.gr/en/market/market-statistics/file-download-api>

RAAEY - Geospatial data on energy projects (WFS)

Download service for the Greek energy regulator's geospatial data: wind farms and individual wind turbines by licensing stage, geothermal and solar-thermal plants, small hydroelectric stations, anemometric masts, waste management facilities and administrative boundaries.

Access: OGC WFS 2.0.0 (GeoServer) with no registration; supports GetFeature with spatial and temporal filters and result paging; GML, GeoJSON, CSV, Shapefile, KML.

Link: <https://geo.rae.gr/geoserver/wfs?service=WFS&version=2.0.0&request=GetCapabilities>

energycost.gr - Energy tariff comparison (RAAEY)

Official tool for comparing all commercially available electricity and natural gas tariffs, consolidating every charge into a single final price per kWh based on the user's own consumption.

Access: Interactive web tool with no documented API or bulk download; HTML only.

Link: <https://energycost.gr/>

PVGIS - Photovoltaic Geographical Information System (JRC)

Computes solar radiation (hourly, daily, monthly) and the estimated performance of photovoltaic systems for any location, as well as a Typical Meteorological Year (TMY) and the terrain horizon profile.

Access: REST API (GET only) with no registration or key; limit of 30 calls/second per IP address; CSV, JSON, EPW.

Link: https://joint-research-centre.ec.europa.eu/photovoltaic-geographical-information-system-pvgis/using-pvgis-5/api-non-interactive-service_en

3. Aviation

OpenSky Network

Community-based receiver network that collects and publishes air traffic data (ADS-B, Mode-S, FLARM) for research purposes, with a live flight map and historical datasets.

Access: Web portal, REST API and a historical database; a free account is required; use is restricted to research and non-commercial purposes.

Link: <https://opensky-network.org/>

OpenSky Network API - Documentation

Retrieval of live aircraft “state vectors” (position, velocity, heading, identity), flights by aircraft or airport, and trajectories. It does not include commercial flight schedules or delays.

Access: REST API with OAuth2 (client credentials); daily quotas in credits: 400 anonymous, 4,000 with an account, more for those feeding the network; JSON. Python and Java libraries are available.

Link: <https://openskynetwork.github.io/opensky-api/>

AeroDataBox

Commercial aviation data API: flight schedules and real-time flight status, ADS-B position tracking, aircraft and airport reference data, and flight time estimation.

Access: REST API with a key, via RapidAPI or API.Market; free tier of 400 units/month, sufficient only for a prototype; JSON, with an OpenAPI v3 specification available.

Link: <https://aerodatabox.com/doc>

OurAirports - Open data

Free, community-edited global airport database, with files covering airports, runways, radio frequencies, nav aids, countries and regions.

Access: Bulk file download with no API, registration or key; the data is released to the public domain; CSV (also KML and RSS). A copy of airports.csv (about 85,900 records) is already included in the assignment folder.

Link: <https://ourairports.com/data/>

4. Natural disasters

GDACS - Global Disaster Alert and Coordination System

A cooperation framework between the United Nations and the European Commission providing near-real-time alerts and impact estimates for earthquakes, tropical cyclones, floods, volcanoes, droughts and forest fires, on a three-level colour-coded alert scale.

Access: Web portal and open feeds with no registration; RSS/GeoRSS, CAP, KML, JSON/GeoJSON. Optional free registration for email or SMS alerts.

Link: <https://www.gdacs.org/>

GDACS API (Swagger)

The programmatic interface behind GDACS: event lists and details filterable by hazard type, date, country and alert level, modelling products (shakemaps, cyclone storm surge, volcanic ash), Copernicus EMS activations and historical EM-DAT records.

Access: REST API with no key or registration, exposing more than 100 endpoints; JSON and GeoJSON, with some endpoints returning files (PDFs, images).

Link: <https://www.gdacs.org/gdacsapi/swagger/index.html>

NASA EONET - Earth Observatory Natural Event Tracker

A metadata repository of natural events (wildfires, storms, volcanoes, ice, floods) that links each event to its sources and to relevant satellite imagery layers.

Access: REST API (version 3) with no documented key requirement; filters by category, source, status (open/closed), date range and bounding box; JSON, GeoJSON, RSS, ATOM.

Link: <https://eonet.gsfc.nasa.gov/docs/v3>

5. Environment and air quality

OpenAQ

A non-profit organisation that aggregates ground-level air quality measurements from stations worldwide and harmonises them into a single format so that they are comparable: PM2.5, PM10, SO₂, NO₂, CO, O₃ and other pollutants, together with station metadata.

Access: REST API v3 with a free key (X-API-Key header) and a bulk historical archive on AWS Open Data (S3); JSON and CSV. Use is subject to the terms of the original data providers.

Link: <https://docs.openaq.org/>

6. Statistical data

Eurostat - Data access via API

Hub page for Eurostat's interfaces for retrieving statistical data and structural metadata: API Statistics, SDMX 3.0, SDMX 2.1, the Catalogue API and an asynchronous API for large extractions.

Access: REST APIs with no registration or key; JSON-stat, SDMX-ML, SDMX-CSV, TSV.

Link: <https://ec.europa.eu/eurostat/web/user-guides/data-browser/api-data-access/>

Eurostat - SDMX 3.0 API (getting started guide)

Retrieval of time series and structural metadata (data structure definitions, codelists, concept schemes, dataflow hierarchy) following the SDMX standard, with filtering by series key and time period.

Access: SDMX REST API (HTTP GET) with no key; SDMX-ML 3.0 and 2.1, SDMX-CSV, TSV, JSON-stat, with optional gzip compression.

Link: <https://ec.europa.eu/eurostat/web/user-guides/data-browser/api-data-access/api-getting-started/sdmx3.0>

Eurostat - City statistics (Urban Audit)

Comparable measurements of the different aspects of quality of urban life in European cities: demography, social and economic indicators, education, environment, transport and culture, with around 144 variables at city and functional urban area level.

Access: Metadata page (HTML, with Excel and PDF annexes); the data itself (the urb_* datasets) is retrieved through the Eurostat Data Browser and APIs.

Link: https://ec.europa.eu/eurostat/cache/metadata/EN/urb_esms_es.htm

World Bank - Indicators API

Access to around 16,000 indicator time series from more than 45 World Bank databases (World Development Indicators, International Debt Statistics, Human Capital Index and others), many spanning more than 50 years.

Access: REST API with no registration or key (authentication is no longer required); JSON and XML, with bulk CSV/Excel downloads also available.

Link: <https://datahelpdesk.worldbank.org/knowledgebase/articles/889392>

7. Geospatial data (GIS)

Overpass API (OpenStreetMap)

A read-only API that returns selected portions of the OpenStreetMap data; it acts “like a database over the web”, executing queries and returning the matching elements (nodes, ways, relations).

Access: HTTP API with no key on the public instances, under a fair-use policy (under 10,000 queries and 1 GB per day, and far lower for a continuously running application); OSM XML, JSON, CSV; Overpass QL query language. The data is licensed under ODbL.

Link: https://wiki.openstreetmap.org/wiki/Overpass_API

GISCO - NUTS (Eurostat)

Geometries of the NUTS statistical and administrative units at levels 0 to 3 for Europe (region polygons, boundaries and label points), for release years from 2003 to 2024 and at several generalisation scales (1:1M to 1:60M).

Access: Bulk download over HTTP with machine-readable JSON indexes, no registration or key; GeoJSON, TopoJSON, Shapefile, GeoPackage, GeoParquet, PBF, SVG, CSV, in EPSG 3035 / 4326 / 3857.

Link: <https://gisco-services.ec.europa.eu/distribution/v2/nuts/>

GISCO - Countries (Eurostat)

Country boundary geometries with worldwide coverage (polygons, boundaries and label points), for the release years 2001, 2006, 2010, 2013, 2016, 2020 and 2024.

Access: Bulk download over HTTP with JSON indexes, no registration or key; the same range of formats and projections as NUTS.

Link: <https://gisco-services.ec.europa.eu/distribution/v2/countries/>

Eurostat GISCO - Local Administrative Units (LAU)

The local administrative units (municipalities and communes) of the European Statistical System, which form the building blocks of NUTS; the annual release also includes resident population figures per unit where available.

Access: Download from the page and programmatically through the GISCO distribution API, with no key; GeoPackage, Shapefile, TopoJSON, GeoJSON, PBF, SVG, CSV, for years from roughly 2011 to 2024.

Link: <https://ec.europa.eu/eurostat/web/gisco/geodata/statistical-units/local-administrative-units>

Open-Meteo - Geocoding API

Place-name search returning coordinates, elevation, time zone and population from the GeoNames database, with name matching in many languages and administrative hierarchy information.

Access: REST API with no key for non-commercial use; JSON and Protocol Buffers. The data comes from GeoNames (CC-BY).

Link: <https://open-meteo.com/en/docs/geocoding-api>

Athens GIS Repository

A collection of open spatial layers for Athens: administrative boundaries, transport networks (metro, tram, buses, road network), points of interest (schools, universities, banks, fuel stations), environmental layers (Natura 2000, rivers), urban planning data, natural hazards and population statistics.

Access: Interactive map with per-layer download and no registration; there is no documented API; licensed CC BY 4.0. The repository is in BETA.

Link: <https://athensgis.gr/>

8. Culture and media

MusicBrainz API

Interface to the MusicBrainz database, an open encyclopaedia of music metadata covering artists, releases, recordings, labels, works and the relationships between them, aimed at media players, taggers and similar applications.

Access: REST API with no key, but a meaningful User-Agent is mandatory and the rate limit is 1 call/second; full database dumps are also published; XML and JSON. The core data is CC0.

Link: https://musicbrainz.org/doc/MusicBrainz_API

ListenBrainz

A service that publicly records users' listening history and exposes listen histories, user and site statistics, recommendations, playlists and feedback.

Access: REST API (public endpoints for reading, a token for submissions and personal data), rate limit of about 1 call/second; bulk data dumps are also published twice a month; JSON.

Link: <https://listenbrainz.readthedocs.io/en/latest/>

Wikidata - MediaWiki Action API

Machine-readable access to Wikidata's structured, multilingual knowledge base (Q items and P properties), covering people, works, places, organisations and identifiers linking to a wide range of other databases.

Access: HTTP Action API with no key for reading; a SPARQL endpoint and periodic data dumps are also available; JSON, XML, RDF. The content is CC0.

Link: <https://www.wikidata.org/w/api.php>

TMDB - The Movie Database API

A community-built database of films, television series and cast and crew, with images, translations and reference data (languages, countries, time zones).

Access: REST API (v3); an account and a key are required; free for non-commercial use, with an attribution requirement; JSON.

Link: <https://developer.themoviedb.org/docs/getting-started>

Open Library - APIs

The Internet Archive's open, editable book catalogue, with metadata for works, editions, authors and subjects, plus search, covers and lists.

Access: REST APIs with no key (rate limit of 1 call/second, 3 with an identifiable User-Agent); monthly bulk data dumps are also available; JSON, YAML, RDF/XML.

Link: <https://openlibrary.org/developers/api>

9. Tourism and short-term rentals

Inside Airbnb

A project that compiles publicly available Airbnb listing data to support research and public debate on the impact of short-term rentals on cities and neighbourhoods. For each city it publishes detailed listings, an availability calendar, reviews and neighbourhood boundaries.

Access: Bulk file download per city (no API or key) for more than 100 cities, usually updated quarterly; CSV (compressed) and GeoJSON; licensed CC BY 4.0. Download page: <https://insideairbnb.com/get-the-data/>

Link: <https://insideairbnb.com/explore/>