# Project of “Software as a Service” Course

School of Electrical and Computer Engineering  
National Technical University of Athens

Spring Semester 2022-2023

Professors: V. Vescoukis, P. Tsanakas

# Description

The project involves the complete development of "myCharts", a small web-based software application that will be provided as a service (SaaS), with the specifications provided by the instructors. The development cycle does not include requirements investigation, analysis of customer business processes, etc., but aims to simulate conditions where, during the development of an application, each team specifies the specifications that are sufficiently known at the start. The focus is on the architecture, technologies, and application development tools provided as services, as well as the application of flexible (agile) development methodologies, also with the corresponding modern tools.

"MyCharts" is a web-based application/service that is aimed at users lacking the requisite technical expertise, thereby enabling them to generate charts and subsequently download them in various formats onto their computer systems. The service supports a range of 3 to 9 chart types, contingent upon the number of students present within each group. For each type, the user has the option to download a template file in csv format, illustrating how to populate chart data as well as providing supported formatting guidelines. To create a chart, the user must upload the requisite completed csv file. Upon successful generation of the chart, the user can then proceed to download the chart in one of the supported formats.

The minimum supported features are listed below:

* User management and login with a Google account
* Downloading a template file of data and formatting parameters for each supported chart type
* Uploading data and formatting parameter files
* Creating a chart and saving it to the server in supported formats such as pdf, png, svg, and html (for adding to a website)
* Displaying the charts created by each user and allowing them to download the selected format
* Managing quotas that allow the creation of charts.

The aforementioned functionalities are offered via three distinct use cases, which are adequately documented in the Software Requirements Specification (SRS) document provided within the Helios software suite. Moreover, the VPP file that comprehensively outlines the software requirements and use case model is readily accessible within Helios.

In order to generate the charts, one of the highcharts, matplotlib, and chartjs libraries will be utilized. The required chart types that must be supported are as follows:

| **Diagram** | **URL** |
| --- | --- |
| **Highcharts** |  |
| Basic line | www.highcharts.com/demo/line-basic |
| Line with annotations | www.highcharts.com/demo/annotations |
| Basic column | www.highcharts.com/demo/column-basic |
| Pie chart | www.highcharts.com/demo/pie-basic |
| Dependency wheel | www.highcharts.com/demo/dependency-wheel |
| Network graph | www.highcharts.com/demo/network-graph |
| Word cloud | www.highcharts.com/demo/wordcloud |
| Organization chart | www.highcharts.com/demo/organization-chart |
| Polar (radar) chart | www.highcharts.com/demo/polar |
| **Matplotlib** |  |
| Bar Label | matplotlib.org/stable/gallery/lines\_bars\_and\_markers/bar\_label\_demo.html#sphx-glr-gallery-lines-bars-and-markers-bar-label-demo-py |
| Simple plot | matplotlib.org/stable/gallery/lines\_bars\_and\_markers/simple\_plot.html# |
| Scatter plot with legend | matplotlib.org/stable/gallery/lines\_bars\_and\_markers/scatter\_with\_legend.html |
| Bar chart on polar axis | matplotlib.org/stable/gallery/pie\_and\_polar\_charts/polar\_bar.html |
| Stack plot | matplotlib.org/stable/gallery/lines\_bars\_and\_markers/stackplot\_demo.html#sphx-glr-gallery-lines-bars-and-markers-stackplot-demo-py |
| Stem plot | matplotlib.org/stable/gallery/lines\_bars\_and\_markers/stem\_plot.html#sphx-glr-gallery-lines-bars-and-markers-stem-plot-py |
| Discrete distribution as horizontal bar chart | matplotlib.org/stable/gallery/lines\_bars\_and\_markers/horizontal\_barchart\_distribution.html |
| Filled polygon | matplotlib.org/stable/gallery/lines\_bars\_and\_markers/fill.html |
| Nested pie charts | matplotlib.org/stable/gallery/pie\_and\_polar\_charts/nested\_pie.html#sphx-glr-gallery-pie-and-polar-charts-nested-pie-py |
| **Chartjs** |  |
| Line chart | www.chartjs.org/docs/latest/samples/line/line.html |
| Multi Axis Line Chart | www.chartjs.org/docs/latest/samples/line/multi-axis.html |
| Radar | www.chartjs.org/docs/latest/samples/other-charts/radar.html |
| Scatter | www.chartjs.org/docs/latest/samples/other-charts/scatter.html |
| Bubble | www.chartjs.org/docs/latest/samples/other-charts/bubble.html |
| Polar area | www.chartjs.org/docs/latest/samples/other-charts/polar-area.html |
| Doughnut | www.chartjs.org/docs/latest/samples/other-charts/doughnut.html |
| Stacked bar chart | www.chartjs.org/docs/latest/samples/bar/stacked.html |
| Multi series pie | www.chartjs.org/docs/latest/samples/other-charts/multi-series-pie.html |

# Teams, Tools

The project will be carried out by teams of 2-6 individuals, who are required to implement all the **mandatory functionalities** of myCharts mentioned above, as a web application that will be deployed in Docker container environment. Upon finalization of the teams, the corresponding GitHub repositories will be made available to them by the instructors.

The architecture that you will implement will be the microservices architecture. The following table shows the diagrams that should be implemented according to the population of each team.

| **Groups of 1-2 students  (3 diagrams)** | **Groups of 3-4 students  (a total of 6 diagrams)** | **Groups of 5-6 students  (a total of 9 diagrams)** |
| --- | --- | --- |
| Highcharts Library | | |
| Basic line | Basic column | Pie chart |
| Line with annotations | Network graph | Organization chart |
| Dependency wheel | Polar (radar) chart | Work cloud |
| Matplotlib Library | | |
| Bar Label | Bar chart on polar axis | Discrete distribution as horizontal bar chart |
| Simple plot | Stack plot | Filled polygon |
| Scatter plot with legend | Stem plot | Nested pie charts |
| ChartJS Library | | |
| Line chart | Scatter | Doughnut |
| Multi Axis Line Chart | Bubble | Stacked bar chart |
| Radar | Polar area | Multi series pie |

The tools that will be utilized are as follows:

* Design, Architecture: Visual Paradigm
* Development: WebStorm, node, express, Bootstrap, μία από τις βιβλιοθήκες highcharts, matplotlib και chartjs, καθώς και Python. DBMS της επιλογής σας.
* Management and commenting source code: GitHub
* Agile Project Management: GitHub/YouTrack
* Deployment: Docker containers

It is important to note that the utilization of an alternate library, aside from Highcharts, is feasible only if it provides precisely corresponding types of charts.

# Deliverables, weights

The project’s deliverables are as follows:

1. **The file vpp should contain the architecture** of your implementation in the form of UML component, deployment, and sequence diagrams, as well as the design of the data infrastructure (ER for relational databases or class diagrams for NoSQL). Each component included in the diagrams should have a link to the corresponding file or folder of source code on GitHub. These functions are provided through three use cases, which are documented in the Software Requirements Specification (SRS) document available on Helios**.** (20%)
2. **The Github repository** that includes the deployment scripts and source code is available**.** (30%)
3. **The application** operates in Docker containers. Optionally and in addition to the mandatory Docker containers, you can upload the application to cloud infrastructure**.** (20%)
4. **Stress tests** will be conducted using the jMeter tool to demonstrate the behavior of the application under heavy loads of requests. The test results must be delivered for review and analysis. (10%)
5. **Updated GitHub repository** with the relevant **GitHub Project Board** (3 – 4 sprints), as well as proper **source code commenting**. (20%)

# Timetable

According to the principles of agile software development, there are no intermediate deliverables in the sense of a complete version of diagrams or code. Instead, snapshots of each team's repository and issues as they are automatically recorded on GitHub will be taken as such. In addition, submissions of the vpp file will be made through Helios. The intermediate deliverables will be made as follows:

* **Sunday 23.4.2023** (right after Easter holidays)
* **Within the final week of the semester**
* **The final deliverable** of the project must be submitted by the day before the course examination, accompanied by the submission of the final version of the vpp file.
* **Presentation** of your work right after the end of the examination period. The duration is 10 minutes, which includes the installation of the Docker containers on a computer of the Software Laboratory and a demonstration of the application’s operation.