

Project Proposal

Web Information Systems [MCS-202]

MakStat-Revised

SEEU University - 3rd of April 2021

Kaltrina Tahiri, Elira Karpuzi, Blend Hamiti

Proposal Questions and Answers

1. Which use case/problem are you addressing with your project?

This application will address the problem that most people face nowadays, the problem of employment on labour market, which in contemporary conditions is estimated as the most socially and economically vulnerable.

One of the essential problems that this project addresses is providing an application that will have reliable, valid, comparable and representative labour information. There is only one website which provides these information for North Republic of Macedonia and its generated charts are usually difficult to comprehend due to their scaling or labeling when a lot of data is selected to be shown and the general population has hard time to search through information. Also the current API doesn't work properly and does not have HAL format links between entities.

Despite production of labour information that is an essential issue and points out a challenge to overcome in many contexts, the true value lies in the analysis and interpretation made of them and furthermore their ability to support evidence-based decision. This will be crucial for informing policy-makers, decision makers and the public in general. (to provide information for the formulation, implementation and evaluation of policies and programmes for jobs in our country, for the research of labour market, setting goals and monitoring.) It is also addressed to people that wish to gain a basic understanding of the labour market in the North Republic of Macedonia in order to apply this in their work or to better grasp labour market developments.

This project also seeks to show the significance of labour statistics in terms of their analytical potential such as:

- Wages trends before and after covid-19 - The problem is that the pandemic has transformed into an unprecedented economic and labour market crisis, damaging workers and enterprises and we will show what has been the pandemic's impact on workers wages.
- Employment distribution by gender and education attainment- this will demonstrate a more comprehensive and accurate concept of inequalities based on gender
- Reasons why people are not looking for a job, by dividing data for urban or rural population, divided by gender over years.
- Activity rate, employment and unemployment rate of the population divided by gender and age groups.

2. What is your planned solution and how is it different from existing solutions?

In short, our solution comprises collecting data from the MakStat database, restructuring and storing collected data into our own database, publishing a RESTful API to browse

the data and a ReactJS web application to visualize the data and create an interactive interface for users.

Due to the size of the database, we decided to only select data related to the labour market and employee structures. These data categories seemed interesting to us in themselves, and also because we could not find any other applications that exposed this data in a more comprehensive form, i.e. as a RESTful API, or a web application. In addition, we plan to dedicate a section of the web page to the comparison of employment statistics in the periods before and after the pandemic, and that could be useful to media outlets as well.

To elaborate on the points of the first paragraph, we plan to download the data from the MakStat database and store it in our own database tables. This is necessary in order for us to publish our own API to browse data, the API that is offered in the MakStat database website, as mentioned in the first section, is largely inconsistent to work with. Thus, we intend to publish a better structured RESTful API and use the HAL convention to allow interactive browsing of data through the API.

Finally, we plan to show well-organized and interactive charts via a web application. The user should be able to switch or select subsets of data through the user interface and easily see different trends over time. This would also allow users to visually make links between different sets of data, which cannot be done if data were shown in table formats.

3. Which datasets do you take as input? How do you transform them into a usable structure?

We have chosen the MakStat database to select some data, the data are related to employee structure. The data are structured into three sessions, the first is about Employees, the second is about Unemployment and the third is about Average wages. All these parts will be separated by fields.

The first part is related to the percentage of employees and will be separated by year, sector, and education attainment. The second part of the project is about unemployment and the data will be separated by percentage of population employed, age, gender and education attainment, also we will include the reasons why the unemployment occurs. The third section is about Average wages and the separation will be done by year, sector, wage distribution, by gender there we will analyze the gender gap pay, and in the last we will have the differences before and after COVID, in this section will see how the wages are changes before and after pandemic.

The data will be taken as a JSON file. These files are composed into objects and by using java code we will restructure them into nested objects. The new structure will be plain old java objects. We have different tables for each page or section of the project.

The first table contains the data about sectors, how many employees have for each year from 2010 to 2019, the years are separated by gender (female and male, also has the option total for two genders), how many females and how many males the sector has. The second table is about age, same as the first one but the difference is to separate employees by age, on each sector how many employees are from up to 19 age to 65 and more age. The third table also is about sectors, but there the data are separated on each year from 2011 to 2019 how many employees with different levels of education there are in each sector.

In the second part the first table is about unemployment, the first table contains data about how many employees and unemployed each year, also are divided by age from 15 to 65 ages and over, also these data also have information about gender how many women and males are employed and unemployed at each age and year. The second is the same but instead of year has education levels. In the third table some reasons are listed and from these reasons we are able to see why many people (mens or females) are unemployed in each year from 2017 to 2019, in urban or rural places.

In the third section we have data about wages. First table is about wages for each sector each year. Second table contains data about average paid net wage per employee, by sector of activity, by NKD, by months. In frontend will appear the gender gap.

4. What is the estimated effort for this project (what tasks are involved, how much time do you expect to work on them) and how do you distribute work in your group?

The discussion of the database schema and the presentation for the project will be done collaboratively, while other tasks are distributed according to the paragraphs below.

Elira - Add desired structure to data and save them into our database

The data that we will have are JSON files, these files aren't in the desired structure, so we have to bring them over the Java in order to create the POJOs, we need to implement them into nested java objects and to put them in our database. I will try to use `"org.json.*"` library in order to construct my desired structure for the data, after the data are in the desired structure we will put them in the database in order to implement the solution that we have chosen.

For this task Elira expects to spend about 6 hours. This will include the time how to find the right solution and if the process will go slower I will increase my working time to 8 hours. I will restructure the JSON files using javascript code, and I will try to work hard to finish the files as soon as possible in order to put them in the database sooner.

Blend - Provide API to query structured data

Blend's tasks involve creating a development setup with Docker, and implementing the API that is served from our server. The Docker setup will allow all teammates to easily replicate and work on the same development setup on their computers. The setup will include a PostgreSQL database to store the data, a Spring Boot application as the backend, and a ReactJS application as the frontend. In addition, Adminer, which is a tool that allows us to manage the database from a web client is included. Regarding the API implementation, the writing of controllers (part of the MVC design pattern) is required, in order to decide how data from the database is mapped into the API.

To set up the required components with Docker, Blend expects to spend about 4 to 8 hours. This would include the time to look up the Docker repository for the desired components, write the build instructions for each component, and test the whole setup. This would be the first task in the project, such that we can all begin our work. Then, Blend expects to spend the rest of his time in the project creating the controllers that implement the API, which would be around 8 hours per week for 8 weeks. The API implementation entails processing of data and creating links according to the HAL convention to allow the user to navigate through the API.

KALTRINA - Design frontend application to visualize data and make it interactive

Kaltrina's task is to create the frontend of the web application with ReactJs. This includes to fetch the data from the server API created by other teammates and process them. In order to demonstrate the data in an interactive and responsive form, there are going to be added charts and graphs in the project by using React Chart Component. Depending on what is more compatible for the data-series the visualization can be in different chart types with various functions such as charts that support animation, zooming, panning, events etc. And despite the volume of data these components should not cause performance lag. The goal is to create a very effective, user-friendly and comprehensible user interface. This will be done by making it easy to navigate through the website and provide consistency, the user can find their way to a new page or a previous one very easily. The application has to be designed in a simple yet very efficient way for the users. To complete these tasks it may take approximately 8 hours per week per 8 weeks.

References

State Statistical Office of the Republic of Macedonia,

www.stat.gov.mk/OtvoreniPodatociApi_en.aspx.