PORTFOLIO

Assignments for the course: Project: Data Engineering (DLBDSEDE02)

TABLE OF CONTENTS

1. TOPICS AND TASKS

Imagine that you work for a municipality that has installed various sensors throughout the city to measure environmental metrics. The project's overall goal is to provide planners with better information to improve the city's environmental conditions in the long term. Also, the data will be used for developing an application that quickly warns citizens if measures exceed recommended values. The project started a couple of months ago, and so far, some of the sensors have been installed. They have already collected the first data comprising roughly half a million measurements. Your goal is to design and implement a data processing system that reliably stores the data making it accessible and usable by the front-end applications planned for this project.

Within the framework of this course, one of the following topics must be selected.

1.1. Task 1: Choose a suitable database and store the data in batches

In this task, you design and implement a data system capable of storing the sensor data. Thoughts that you should keep in mind when choosing a storing option are the following:

- The first data have been collected, including values for temperature, humidity, smoke, etc. and you know the structure of these data to be stored. But the municipality plans to also install advanced sensors in the future capable of measuring additional metrics, e.g., carbon dioxide, noise, fine dust, etc. As of yet, the structure of the data collected by these additional sensors is not clear.
- The database to be chosen should provide the most straightforward data model possible to make it maintainable in a straightforward way.
- In your first prototype, you will store the data on your local machine, but the data should be stored in a distributed system across many nodes in the long term. You should choose an appropriate database solution that will allow you to seamlessly migrate the data system in larger setups, e.g., in the cloud, for increased reliability and horizontal scalability.
- The data system should be portable to other systems. This includes the system having no dependencies that bound it to your local machine. You can achieve this, for example, by containerization.

Task: Choose a suitable database and store the data in batches

Implement the data system in the following 3 phases:

1.1.1. Conception phase

This phase represents the most important part of the process. Anything that is overlooked or forgotten in this phase has a negative effect on the implementation later and will lead, in the worst case, to useless results.

The first step is to create a written concept to describe everything that belongs to the data system. **This step is perhaps the most important of the entire process**. It is crucial to take enough time for this phase **BEFORE** the next steps can be taken. **It is therefore essential to follow the sequence of the respective steps carefully.**

1. Choose an appropriate sample data source. Open datasets can be found, e.g., on www.kaggle.com or https://github.com/owid. Search for "sensor", "IoT", "environmental", "telemetry" and similar terms. Inspect the data and include its structure in your considerations for the following steps.
2. Briefly describe the overall goal of the data system.
3. Choose an appropriate database solution capable of fulfilling all requirements listed above. Describe how your intended data storing solution will be reliable, scalable, and maintainable.
4. Justify your decision and discuss alternatives and why they might not fit this use case.
5. Create a plan for implementing your data system prototype.

A conceptual text (1/2 DIN A4 page) has to be prepared for the submission, explaining these thoughts and considerations. The Concept will be submitted as **PDF file**. The text field inside the PebblePad template can be left empty.

Throughout the process, online meetings provide an opportunity to talk, share ideas and/or drafts, and obtain feedback. In the online meetings, exemplary work that has been previously submitted can be discussed with the tutor. Here, everyone has the opportunity to get involved and learn from each other's feedback. There are also other channels available for you to address questions to the tutor and/or to your fellow students, such as the CourseFeed. Through the latter, you will obtain feedback, tips, and advice before submitting your results of this phase. **It is recommended to make use of these channels to avoid errors and to make improvements.** You submit work after making use of all available informative media. This will be followed by feedback from the tutor, and the work in the second phase can begin.

1.1.2. Development phase/reflection phase

In this phase, **the data system is implemented** based on the concept from the conception phase with the help of the selected frameworks and tools. You will probably run into some issues at the beginning of this phase. This is intended and part of the learning cycle. To tackle these issues, consult the "Get Started" pages of Docker and the database you chose in the conception phase.
Here, in the development phase, the actual work of implementation begins:

- Install the database solution you chose in the conception phase. To minimize the dependencies of your local system, you should install Docker, load an open-source container containing an instance of the selected database from Docker Hub and run this container.
- Write a script to set up the database within that container as needed to fulfill the project requirements.
- Write a Python script to connect to the database and load the sample data into it.
- Create or modify a Dockerfile to include all required steps to automate the entire process. This will allow your data system to be portable to other environments, like another machine, possibly with another operating system or the cloud.
- Set up an open GitHub repository to store the code in a version-controlled way.

- Load your Dockerfile and all associated files needed to run the container into the GitHub repository. Cloning this repository, it should be possible to run the container that automatically loads the sample data into the chosen database.

An explanation of the procedure is submitted as **text (1/2 DIN A4 page)**. The file should contain the link to the GitHub repository. Furthermore, the steps of this phase should be described briefly. The Explanation will be submitted as **PDF file**. The text field inside the PebblePad template can be left empty.

Again, it is recommended to use the provided channels to avoid errors and improve your work. Once this is done, you can hand in your second phase results for evaluation. Following feedback from the tutor, your work on the final draft will continue in the third phase.

1.1.3. Finalization phase

In the finalization phase, the goal is to **optimize the data system** after receiving feedback from the tutor and completing the task. Certain elements may have to be improved or changed again.

You create a **full abstract (2 DIN A4 page)** describing the solution of the task in terms of content and concept, presenting a short break-down of the technical approach in a clear and informative way. This abstract should also include a personal reflection about the project: What did you learn on a personal level? What were the problems you ran into, and how did you solve these issues? What are your problem-solving strategies for similar upcoming projects?

In addition, you upload the **finished product** as a PDF file with a link to the GitHub repository containing a functional data system as code, together with a short technical description how to use the code. A zip folder is not necessary in this course.

Also, in the finalization phase, **it is recommended to use these channels to avoid errors and improve your work.** After submitting the third portfolio phase, the tutor submits the final feedback, which includes evaluation and scoring within six weeks.

1.2. Task 2: Design and implement a stream processing pipeline

All sensors the municipality has installed throughout the city are capable of sending near real-time measurements as a continuous data stream. In this task, you design and implement a data system capable of storing and processing a continuous sensor data stream. You will use Kafka or Spark Streaming to stream the data to a database of your choice.

Task: Design and implement a stream processing pipeline

Implement the data system in the following 3 phases:

1.2.1. Conception phase

This phase represents the most important part of the process. Anything that is overlooked or forgotten in this phase has a negative effect on the implementation later and will lead, in the worst case, to useless results.

The first step is to create a written concept to describe everything that belongs to the data system. **This step is perhaps the most important of the entire process**. It is crucial to take enough time for this phase **BEFORE** the next steps can be taken. **It is therefore essential to follow the sequence of the respective steps carefully.**

1. Choose an appropriate sample data source. This task is about streaming data. For your project, you have four options to use sample data:
    a. Load an open static dataset from the internet. Open datasets can be found, e.g., on www.kaggle.com or https://github.com/owid. Search for "sensor", "IoT", "environmental", "telemetry" and similar terms. Next, write a short script to feed the data entry by entry to your system with defined time intervals between them, e.g., one entry per 10 seconds.
    b. Use an openly available data API to connect directly to a stream of data. You might find openly available streaming data APIs on the internet.
    c. Simulate a data stream using a tool of which many are available on the internet or write a short program on your own. This program will run on your local machine, simulating sensor data.
    d. Use the sensors in your smartphone as sample data. There are openly available tools like "PhonePi" or "Sensor Node" that allow you to stream your phone's sensor data to a Python environment on your machine.
2. Briefly describe the overall goal of the data system.
3. Choose either Kafka or Spark Streaming for your data processing pipeline. Describe how your intended solution will be reliable, scalable, and maintainable.
4. Create a plan for implementing your data system prototype.

A conceptual text (1/2 DIN A4 page) has to be prepared for the submission, explaining these thoughts and considerations. The Concept will be submitted as **PDF file**. The text field inside the PebblePad template can be left empty.

Throughout the process, online meetings provide an opportunity to talk, share ideas and/or drafts, and obtain feedback. In the online meetings, exemplary work that has been previously submitted can be discussed with the tutor. Here, everyone has the opportunity to get involved and learn from each other's feedback. There are also other channels available for you to address questions to the tutor and/or to your fellow students, such as the CourseFeed. Through the latter, you will obtain feedback, tips, and advice before submitting your results of this phase. **It is recommended to make use of these channels to avoid errors and to make improvements.** You submit work after making use of all available informative media. This will be followed by feedback from the tutor, and the work in the second phase can begin.

1.2.2. Development phase/reflection phase

In this phase, **the data system is implemented** based on the concept from the conception phase with the help of the selected frameworks and tools. You will probably run into some issues at the beginning of this phase. This is intended and part of the learning cycle. To tackle these issues, consult the "Get Started" pages of Docker, Kafka, and Spark Streaming.
Here, in the development phase, the actual work of implementation begins:

- Install either Kafka or Spark Streaming together with a database of your choice. To minimize the dependencies of your local system, you should install Docker and load an open-source container containing an instance of Kafka or Spark Streaming and a database. You can search Docker Hub, load, and run a suitable openly available container image.
- Write a script to set up Kafka or Spark Streaming within that container as needed to fulfill the project requirements.
- Write a script to stream the sample data into the chosen database using either Kafka or Spark Streaming.
- Create or modify a Dockerfile to include all required steps to automate the entire process. This will allow your data system to be portable to other environments, like another machine, possibly with another operating system or the cloud.
- Set up an open GitHub repository to store the code in a version-controlled way.
- Load your Dockerfile and all associated files needed to run the container into the GitHub repository. Cloning this repository, it should be possible to run the container that automatically streams the sample data into the chosen database.

An explanation of the procedure is submitted as **text (1/2 DIN A4 page)**. The file should contain the link to the GitHub repository. Furthermore, the steps of this phase should be described briefly. The Explanation will be submitted as **PDF file**. The text field inside the PebblePad template can be left empty.

Again, it is recommended to use the provided channels to avoid errors and improve your work. Once this is done, you can hand in your second phase results for evaluation. Following feedback from the tutor, your work on the final draft will continue in the third phase.

1.2.3. Finalization phase

In the finalization phase, the goal is to **optimize the data streaming system** after receiving feedback from the tutor and completing the task. Certain elements may have to be improved or changed again.

You create a **full abstract (2 DIN A4 page)** describing the solution of the task in terms of content and concept, presenting a short break-down of the technical approach in a clear and informative way. This abstract should also include a personal reflection about the project: What did you learn on a personal level? What were the problems you ran into, and how did you solve these issues? What are your problem-solving strategies for similar upcoming projects?

In addition, you upload the **finished product** as a PDF file with a link to the GitHub repository containing a functional data system as code, together with a short technical description how to use the code. A zip folder is not necessary in this course.

Also, in the finalization phase, **it is recommended to use these channels to avoid errors and improve your work**. After submitting the third portfolio phase, the tutor submits the final feedback, which includes evaluation and scoring within six weeks.

2. TUTORIAL SUPPORT

In principle, several channels are open to attain feedback for the portfolios. The respective use is the sole responsibility of the user. The independent development of a product and the work on the respective portfolio parts is part of the examination performance and is included in the overall assessment.

On the one hand, the tutorial support provides feedback loops on the portfolio parts to be submitted in the context of the conception phase as well as the development and reflection phase. The feedback takes place within the framework of a submission of the respective part of the portfolio. In addition, regular online tutorials are offered. These provide you with an opportunity to ask any questions regarding the processing of the portfolio and to discuss other issues with the tutor. The tutor is also available for technical consultations as well as for formal and general questions regarding the procedure for portfolio management.

Technical questions regarding the use of "PebblePad" should be directed to the exam office via mail.

3. EVALUATION

The following criteria are used to evaluate the portfolio with the percentage indicated in each case:

Evaluation criteria	Explanation	Weighting
Problem Solving Techniques	*Capturing the problem *Clear problem definition/objective *Understandable concept	10%
Methodology/Ideas/Procedure	*Appropriate transfer of theories/models *Clear information about the chosen Methodology/Idea/Procedure	20%
Quality of implementation	*Quality of implementation and documentation	40%
Creativity/Correctness	*Creativity of the solution approach *Solution implemented fulfils intended objective	20%
Formal requirements	* Compliance with formal requirements	10%

The design and construction of the portfolio should take into account the above evaluation criteria, including the following explanations:

Problem Solving Techniques: Correct reflection and implementation of data engineering concepts such as reliability, scalability, and maintainability. Demonstration of independently tackling technical issues. Following appropriate problem-solving strategies and independently coming up with solutions.

Methodology/Idea/Procedure: Correct usage of technical frameworks and reasonable argumentation for technical choices.

Quality of implementation: Fulfillment of the technical requirements. Reproducibility of the created product in the form of a functional Dockerfile provided in a GitHub repository. Concise and comprehensive documentation.

Creativity/Correctness: Creative approach to the problems and correct task fulfillment by the implemented system

Formal requirements: Compliance with the formal requirements for submission (see below)

4. FORMAL GUIDELINES AND SPECIFICATIONS FOR SUBMISSION

4.1. Components of the examination performance

The following is an overview of the examination performance portfolio with its individual phases, individual performances to be submitted, and feedback stages at one glance. A template in "PebblePad" is provided for the development of the portfolio parts within the scope of the examination performance. The presentation is part of this examination.

Stage	Intermediate result	Performance to be submitted
Conception phase	Portfolio part 1	• Concept presentation in written form, which concisely shows that you thought about the requirements listed in the task description (200 words; approx. 1/2 page as PDF-file)
		Feedback
Development phase/ reflection phase	Portfolio part 2	• Explanation of the implementation in written form, including a link to a Git Hub repository (200 words; approx. 1/2 page as PDF-file)
		Feedback
Finalization phase	Portfolio part 3	• The full abstract as an attached 2 page PDF file including the project documentation as well as a technical and personal project reflection. • The final product as a PDF file with a link to the GitHub repository and a short description about how to use the code • Result from phase 1 (can be revised in the meantime) • Result from phase 2 (can be revised in the meantime)
		Feedback + Grade

4.2. Format for Digital File Submission

Conception phase	
Recommended tools/software for processing	Git Hub, IDE of choice (VS Code), Docker for your local machine
Permitted file formats	PDF
File size	-
Further formalities and parameters	Files must always be named according to the following pattern: **For the performance-relevant submissions on "PebblePad":** Name-FirstName_MatrNo_Course_P(hase)-1_S(ubmission) Example: Mustermann-Max_12345678_DataEngineering_P1_S

Development/reflection phase	
Recommended tools/software for processing	Git Hub, IDE of choice (VS Code), Docker for your local machine
Permitted file formats	PDF
File size	-
Further formalities and parameters	Files must always be named according to the following pattern: **For the performance-relevant submissions on "PebblePad":** Name-FirstName_MatrNo_Course_P(hase)-2_S(ubmission) Example: Mustermann-Max_12345678_DataEngineering_P2_S

Finalization phase	
Recommended tools/software for processing	Git Hub, IDE of choice (VS Code), Docker for your local machine
Permitted file formats	full abstract as PDF + final product as PDF (with link to GitHub)
File size	as small as possible
Further formalities and parameters	Files must always be named according to the following pattern: **For the performance-relevant submissions on "PebblePad":** Name-FirstName_MatrNo_ Course _P(hase)-3_S(ubmission) Example: Mustermann-Max_12345678_DataEngineering_P3_S

4.3. Format of Abstract

Length	2 pages of text
Paper size	DIN A4
Margins	Top and bottom 2cm; left 2cm; right 2cm
Font	General Text - Arial 11 pt.; Headings - 12 pt., Justify
Line Spacing	1,5
Sentences	Justified; hyphenation
Footnotes	Arial 10 pt., Justify
Paragraphs	According to mental structure - 6 pt. after line break
Affidavit	The affidavit shall be made in electronic form via "myCampus". No submission of the examination performance is possible before it. Please follow the instructions for submitting a portfolio on "myCampus".

If you have any questions regarding the submission of the portfolio, please contact the exam office via mail.

Please also note the instructions for using PebblePad & Atlas!

Good luck creating your portfolio!