Milestone

A part of the system has to be finished by class 8. This will be graded by a Code Review.

Component	Description	Percent
Must Haves	• Every component can be started independently • System can be build and run with no errors • Spring Boot used for REST API • JavaFX used for GUI • **GitHub repository link in submission**	**mandatory** If this is not in the project, the milestone will be graded with 0 points
REST API	The REST API will be implemented using Spring Boot and designed to provide example data for testing and demonstration purposes. It will include two endpoints: one to retrieve the data of the current hour, and another to filter historic data. The API will focus solely on returning structured example data to simulate real-world usage scenarios without relying on a persistent data store or handling complex business logic.	50%
GUI	The GUI will be developed using JavaFX, providing an intuitive and interactive interface for users to interact with the REST API. The application will include buttons and input fields to send requests to the API, such as fetching the hour and historic energy data. Retrieved data will be displayed dynamically within the application using visual components like tables, labels, or text areas. The design will focus on simplicity and clarity, ensuring a seamless user experience while demonstrating the integration of JavaFX with a REST API for real-time data interaction.	50%

Final Submission

The final submission must be presented in class and the project will be graded by Code Review.

Component	Description	Percent
Must Haves	• Every component can be started independently • System can be build and run with no errors • Spring Boot used for REST API • JavaFX used for GUI • RabbitMQ used for communication between services • **GitHub repository link in submission**	**mandatory** If this is not in the project, the final project will be graded with 0 points
REST API	The Spring Boot App reads the data from the database instead of using static sample data.	10%
Energy Producer	Sends production message in random 1-5 second intervals to the message queue. Production message must include a random but sensible kWh value.	10%
Energy User	Sends usage message in random 1-5 second intervals to the message queue. Usage message must include a random but sensible kWh value.	10%
Usage Service	Receives the production/usage messages and updates the usage table correctly. Afterwards sends an update message to the queue.	40%
Current Percentage Service	Receives the update message and updates the current percentage table correctly.	30%

Points can be omitted for badly written code, unsufficient presentation or the failure to explain the system.

Last modified: Monday, 20 January 2025, 1:18 PM

?