

1 Program Design

Goal of this program design is to model the architecture and detailed design based on the requirements you have written in the first program design in last quarter. As deliverable of this program design, you shall submit:

- A System Design Document (PDF) following the structure discussed in the lecture, describing the overall software architecture of the truck system. This document must contain all relevant architectural and design information and serves as the primary basis for assessment.
- The detailed design of the emergency stop system as part of the UML model below.
- An Enterprise Architect project file containing the UML diagrams you created for this program design. This file is submitted as supplementary material and will not be graded; all relevant content must be included in the System Design Document.

1.1 Schedule

Due date for group selection:

Send an e-mail to Stimac.Daniel@dozent.dhbw-ravensburg.de with all members of the group and a till **13.02.2026, 23:59**. Please define a group name and nominate one responsible member who will upload the final version on behalf of the group.

Q & A Meeting:

Q & A sessions is planned for February 16th and February 23rd

Please send an advanced notice to Stimac.Daniel@dozent.dhbw-ravensburg.de if you have questions.

Due date for Submission: 02.03.2025, 23:59

2 Tasks

2.1 Group Task [0 Points]

You shall form groups of max. four students. We will limit the program design on the software that is located in the truck and will exclude the PC application.

In this group discuss the requirements you have formulated in the last program design and make sure that all team members have a common understanding of the requirements.

Agree on an overall software architecture that you want to use for your truck. You will later need to document the architecture you have agreed on and document why you have chosen this architecture and what potential alternatives exist. Also decide whether you want to implement the software in C or in C++ and whether you want to take full advantage of object-oriented design or not (your decision will impact how you must model the architecture and detailed design as discussed in the lecture).

2.2 Document the Architecture [75 Points]

In this step you shall create the above-mentioned system design document, that documents the software architecture that you have agreed on in Step 2.1. Per group you are only allowed to submit one version of an Enterprise Architect model that shall contain all views / diagrams mentioned below.

The following guidelines shall help you to assess what is expected in detail for “Proposed Software Architecture” of the system design document and the Enterprise Architect Model:

Sub-Section	Expected Diagrams	Points
Overview	Context Diagram (Component or Package Diagram) modeling the overall truck system with system boundaries (e.g. connection to HW / PC application)	10
Subsystem decomposition: subsystem decomposition responsibilities of individual subsystems	Decompose the software architecture with a Package or Component Diagram into sub-system. Decompose each sub-system into a class-diagram. The class diagram shall contain attributes and public operations for the classes.	25

	Include here the discussion on potential architecture alternatives and the rational why your team has agreed on this specific architecture.	
Hardware/software mapping: how subsystems are assigned to hardware	Provide a deployment diagram how the software subsystems are mapped to hardware.	5
Global software control: initiation of requests subsystem synchronization concurrency	Create at least one sequence diagrams that model the interaction of the “Emergency Stop” with your subsystem, the sequence diagrams shall re-use the classes and interfaces you have created.	20
Boundary conditions: start-up, shutdown, and error behavior	Create at least one sequence diagram that models: start-up, shutdown and one error scenario.	15

2.3 Detailed Design [25 Points]

Select the class implementing most of the functionality “Emergency Stop” from your model and completely specify the detailed design. The detailed design needs to be specified in a detailed enough way so that all information that is required to implement the software is captured in the model.

Create a Package “Detailed Design” that shall contain all diagrams created for this step.

The detail design shall include the following:

- The class shall be refined with private methods

- There shall be a sequence diagram specifying how the class is used (how the external interfaces are called) and what dependencies the class has, as well as how the internal (private) methods are called.
- For each method of the class there shall be either a state-machine diagram or an activity diagram created that models the behavior of the class. Use the note / comment feature of the model elements to specify additional information required for implementation. If your state-machine is implemented on the level of the class and controlled by several methods, only one state machine diagram is sufficient.

3 General Considerations

In all the above-mentioned tasks during grading is considered how the methods and concepts discussed in the lecture are applied. You can add notes in the enterprise architect model to provide additional information or clarification. Furthermore, consistency of the architecture is considered during grading.