

# Project Specification

Software Engineering Design  
2DV603-19VT

Mauro Caporuscio, Mirko D'Angelo  
Department of Computer Science and Media Technology

# •••• Project Plan

---

- Plan:
  - w14 [05/04/2019]: Project Specification
  - w15 [12/04/2019]: Q&A Time (Project checkpoint)
  - w16 [17/04/2019]: Submission **D1: Requirement specification document**
  - w17 [26/04/2019]: Q&A Time (Project checkpoint)
  - w18 [03/05/2019]: Submission **D2: Design document**
  - w19 [10/05/2019]: Q&A Time (Project checkpoint)
  - w20 [17/05/2019]: Q&A Time (Project checkpoint)
  - w22 [31/05/2019]: Submission **D3: Final release** (code and (revised) documentation)
  - w23 [05/06/2019]: Final Workshop

# ••• Project Groups

---

- **MyMoodle** → Project group choice
  - Choose your group
  - 7 members for each group
- **TEAMMATES** → <http://teammatesv4.appspot.com/>
  - Every 1 week (compulsory) peer review
  - Compulsory!

# Thesis Management

- The goal of this Project is to create a system managing the **thesis activities of Linnaeus University** for the department of Computer Science
- **Scenario 1: Mirko's thesis**

From this year a new system is managing all the thesis at the department. Mirko is a master student who is following the master's thesis course. After logging in into the system, Mirko is recognized as a student and can submit the Project Description document for his thesis. The document is validated by the coordinator some weeks later. Mirko is now allowed to submit the final Project Plan for the thesis and suggest a supervisor from the list of available supervisors in the system. After the deadline for the Project Plan submission, the coordinator validates the Project Plan and, after the ok from the Supervisor, Mirko is now able to submit the Final Report. After receiving the comments to apply for the final submission, Mirko submits the Final Report. The final grade is later issued by the coordinator.

# ••• Thesis Management

---

- **Scenario 2: Coordinators in action!**

Diego is the project coordinator for the bachelor's thesis course. After logging in into the system, he is able to specify the deadlines for the documents that the students are required to submit. After the first deadline, he evaluates the project descriptions by issuing a pass or fail grade. After the second deadline, it is time to repeat the process for the project plans submitted by the students. At any time he is able to see who has already submitted a particular document and who is missing. Moreover, Diego can see through the system the list of students supervised by a supervisor. After the report submission of a student, the system enables the bidding process for the readers so that the system is able to match readers to a particular report. Diego now confirms or re-assign the readers for a report and assigns an opponent to it. Only after reading the final evaluations of the readers for the submitted report Diego can issues a grade for the thesis.

- **Write the remaining scenarios by looking at the Thesis Management Process**
  - Between 3 and 5 scenarios

# Thesis Management

- **Constraints:**

- Programming language: Java
- Type of application: Web Application

- **Concerns:**

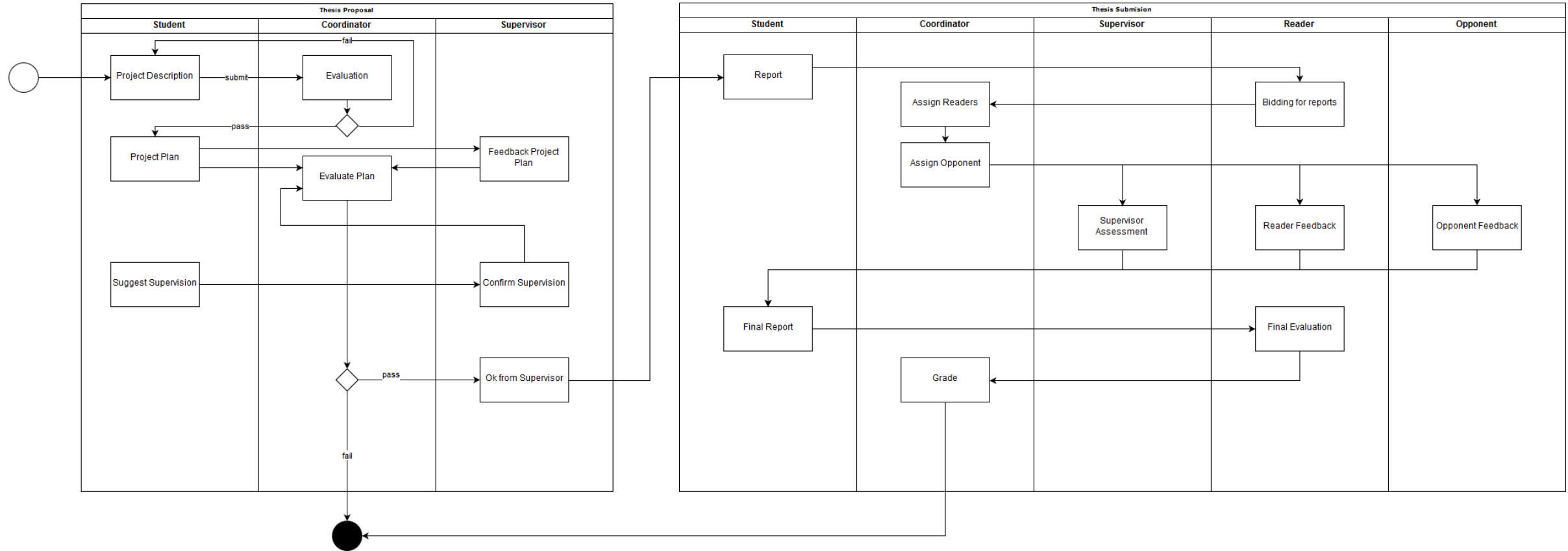
- Parametric deadlines
- Extensions on deadlines
- Conflicts
- Evaluation metrics
- Visibility

- **Non Functional Requirements:**

- Assuming that all the  $N$  students will try to submit the report during the last 10 minutes before the deadline (so you can think as the system is receiving a  $6*N$  requests per hour during that period), and that it is required a response time for the "Submit report" action below 10 seconds, calculate the maximum number of students ( $N$ ) that your system can afford.



# Thesis Management Process



# ••• D1: Requirement Specification Document

---

- Identify stakeholders
- Elicit and label the functional and non-functional requirements
- Perform a systematic ‘checklist-based’ requirements analysis
- Classify the identified requirements using the ‘faceted approach’
- Perform ‘traceability analysis’ in terms of dependency between the requirements and other artifacts
- Propose at least one test case for each of the requirements
- Model the requirements with UML



# D2: Design Document

---

- **Purpose:**
  - What system or part of the system this design document describes
  - Make reference to the requirements that are being implemented by this design (traceability)
- **General priorities:**
  - Describe the priorities used to guide the design process
- **Outline of the design:**
  - Give a high-level description of how you the design of the application
- **Major design issues:**
  - Discuss the important design issues that had to be resolved
  - Give the possible alternatives that were considered, the final decision and the rationale for the decision
- **Design details:**
  - Software Architecture
    - Reference Architectural Pattern (if any)
    - The set of components
      - For each component: discuss both requires and provides interface
    - The UML Component Diagram of the architecture
  - Component Implementation
    - For each component identified in Software Architecture
      - Design Principles addressed (if any)
      - The UML Class Diagram

# ••• D3: Final Release

---

- Refinement of D1
- Refinement of D2
- Code
- Instructions to run the code

