UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

COS 301 Software Engineering

Mini-Project Demo 1 Instructions
2025

---

1. Introduction

This document provides the Demo 1 instructions for the COS 301 Mini-Project 2025. In general, demos are assessment opportunities that we use to track progress and milestones. Though demos are mostly technically focused, a certain soft skill is required to convey your demo well. This document provides an outline of all the technical requirements as well as some suggestions for what makes a good demo.

The demo is on **Friday, 14 March 2025. The demo will be online on Blackboard Collaborate.** Project Managers, please make sure to book one demo slot for your team when the demo slots open.

The general theme of the demo is for us to see a **Beta version of the MPDB project** that you will be **engineering**. This Beta must be "sold" to us, as a client of the Mini-Project, during an **"elevator pitch"-style** demo done by your team's project manager. Each Project Manager will get **10 minutes** to demo their Mini-Project.

2. Prerequisites

You have received the Mini-Project 2025 Playbook. This document serves as the Mini-Project proposal received from the client (the lecturers of this module). It is not a formal requirement specification but rather a proposal for a project, along with client wishes. This document derives the components and deliverables of the Mini-Project.

You also have to be registered on Hyperperform. This is a requirement.

3. Plagiarism

The Mini-Project shares a lot of overlap with existing tutorials and courses that are publicly available. We would like to remind the students here that they are allowed to use references as inspiration but cannot clone or use those references as if it is their own. Students have to do their own novel work and cannot claim ownership of work that is not theirs. Please be warned, that if your project resembles one of these existing courses or tutorials, we will treat it as a plagiarism case.

4. Demo Components

During the **10 minutes** that **Project Managers** demo their team's project, we want you to present to us:

1. Project Introduction
2. Documentation
3. Project Plan
4. Implementation (Progress)
5. Testing
6. CI/CD

These aspects will be explained in more detail in the sections below.

Project Introduction

Introduce us to your team's project by giving us:

- Your team name/number
- Background: Provide a brief background on why your team is undertaking this project and its relevance.
- Scope: Clearly define the scope of your project by specifying which features you will be implementing and any limitations or exclusions.

Documentation

The deliverables for Documentation are:

- **A README.md** file, which is a home page for your project. Please put links here to all relevant resources like:
    - Project Information
        - Short description of your project
        - This description should also appear in the About field in your GitHub repository
        - The description should follow the following format: "TeamName - ProjectName - Project Description."
    - Demo Links (To videos, documentation, etc)
        - A link to your recorded demo video is named "TeamName-Demo1."
        - A link to the Requirement Specification (SRS)
        - A link to your design specification.

        - A link to your GitHub Project Board - Project management tool
    - Team members and roles
        - Short profile description of each member of your team.
        - The individual professional profile of each team member, including a link to LinkedIn, is visible on your GitHub.
    - Your GitHub repository should contain relevant information such as the following:
        - Git structure (mono repo)
        - Git organisation and management
        - Branching strategy
        - Code Quality badges (only the applicable ones):
            - Code Coverage — Coveralls, Codecov, SonarCloud
            - Build status - GitHub Actions
            - Requirements - requires.io
            - Issue tracking - GitHub Issues
- **A requirement specification** (Markdown). This requirement specification is a formal document capturing the requirements of the client. Usually, it also contains some background information about the project. The wishes from the client were given in the Mini-Project 2025 Playbook (see link above). You need to formally list the **technical and non-technical requirements** in a requirement specification. This should contain not only the direct wishes of the client but also any other requirements that are missing in the details. The requirement specification also captures use cases (sometimes in the form of a use-case diagram).
- **A design specification** (Markdown) is a formal document that captures your visual and architectural design elements. These include wireframes, mockups, and **screenshots of views/features/pages** that clearly illustrate what is to be implemented. Further, the design specification highlights your **architectural specifications** through an **architecture diagram**.

Project Plan

Share your plan for the project with us. The deliverables here are:

- A **project board on GitHub** with tickets/cards clearly shows a project plan/roadmap.
- Tell us what **Software Engineering Methodology** you are following and why.
- What is your team composition, i.e. roles allocated to members?

Implementation (Progress)

This is the technical part of the demo. For this component, we need to see how you are progressing on the implementation part of your project. Show us the **“in-progress” workings/skeleton code for the requirements that you have identified**. Importantly, these do not have to be finalised features, but there must be some remnants of skeleton, structure, scaffolding and basic implementations. For demo 1, this **does not** have to be **deployed or fully implemented yet**. Again, a **Beta-version, Proof of Concept** that is demoable.

Testing

Furthering on from the technical aspects. We require you to show us the following:

- The tests that you have identified that need to be run against your set of identified requirements
- There must be:
    - Mock **unit tests.**
    - Mock **integration tests/end-to-end tests.**
- These tests **do not have to be implemented yet**, but their skeletons should be there and **runnable**.

CI/CD

Finally, you must present your CI/CD plan and progress toward its implementation. This includes:

- A Git repository on GitHub where you version-control your code.
- Tell us which Git branching strategy you are using
- What is your review process
- Show us your CI/CD pipeline in action. This means:
    - It should run actions for:
        - **Linting**
        - **Building**
        - **Testing**
        - **Deploying**
    - These do not have to be finalised yet, but executing these stages against “mock scripts” is good enough.
    - **IMPORTANT: Ensure only your main/master branch is only performing CI/CD due to limited build credits on GitHub. NO feature branches should be using CI/CD. Teams who do not obey this rule will be penalized.**
- Tell us about your deployment plans:
    - Where are you deploying to?
    - How are you deploying?
    - When are you deploying?

5. Rubric

The scoring weight for each component is given in the rubric below:

Project Introduction (1 min)	5
Documentation (2 min)	15
Project Plan (2 min)	10
Implementation (Progress) (2 min)	20
Testing (2 min)	20
CI/CD (1 min)	20
Overall Demo Presentation	10
TOTAL	**100**

6. Recording (In case of Emergency)

Please pre-record your demo in case of an emergency. You will demo live, but we use this as a backup. For example, sometimes, load shedding causes disappointments.

7. Further Tips

- Present the demo using PowerPoint, Google Slides or Canva (https://www.canva.com/).
- Speak clearly and avoid distracting things like audio in the background.
- Be prepared and organise your demo according to the instructions and rubric.
- Make sure that your README includes links to any artefact or deliverable that we might ask for. These include your repository, documentation, project board, and proof of testing.
- Stick to your allowed time frame. No extra time is allowed.
- Ensure your diagrams correlate with how you learned to design them from COS 301 and COS 214 (consider draw.io for diagrams, as a visual paradigm can be a headache).
- Groups from 2022 for reference:
    - https://github.com/COS301-SE-2022/Pure-LoRa-Tracking
    - https://github.com/COS301-SE-2022/The-Au-Pair
    - https://github.com/COS301-SE-2022/Japanese-Writing-Evaluator
    - https://github.com/COS301-SE-2022/CryptoHub
    - https://github.com/COS301-SE-2022/ReverseHand

    - https://github.com/COS301-SE-2022/Map-out-game-reserves-using-aerial-photographs
- Groups from 2023 for reference:
    - https://github.com/COS301-SE-2023/AI-Photo-Editor
    - https://github.com/COS301-SE-2023/Blue-Skies
    - https://github.com/COS301-SE-2023/Domain-Pulse-A-Sentiment-Analysis-Platform
    - https://github.com/COS301-SE-2023/Avalanche
    - https://github.com/COS301-SE-2023/WriteToPdf
- Groups from 2024 for reference:
    - https://github.com/COS301-SE-2024/GameOnConnect
    - https://github.com/COS301-SE-2024/Web-Exploration-Engine
    - https://github.com/COS301-SE-2024/Dispute-Resolution-Engine
    - https://github.com/COS301-SE-2024/MoodMix
    - https://github.com/COS301-SE-2024/Crop-Prediction-System