CO510 Software Engineering

David Barnes (convener), Olaf Chitil and Rogério de Lemos

Changes this term

- There will be fewer formal lectures.
  - Keep an eye on the Moodle page for details.
- Classes with exercises are replaced by workshops supporting the group assessment.

Reminder: some themes of the course

- The concept of ‘engineering’.
- Large scale development.
- The role and conduct of an individual within a multi-person, multi-organisational project.
- Dealing with uncertainty.
- Foundational concepts and skills for the final-year project.

The group assessment

- This is a long-term assessment.
- Issues from Stage 1.
  - Untestable user stories.
  - Most groups did not contact Miles for clarification.
  - Evidence of lack of a solid understanding of UML.
    - You will need UML in the exam as well as the project.
- Teams need to be able to rely on their members to function effectively and achieve their goals.
  - Make use of the carding system to prevent freeloading.

Remaining stages – proposed schedule

- Stage 2: Weeks 14-16; Planning & class diagrams.
- Stage 3: Weeks 16-18; Baseline implementation.
- Stage 4: Weeks 18-20; Review & implementation.
- Stage 5: Weeks 20-23; Review & implementation.

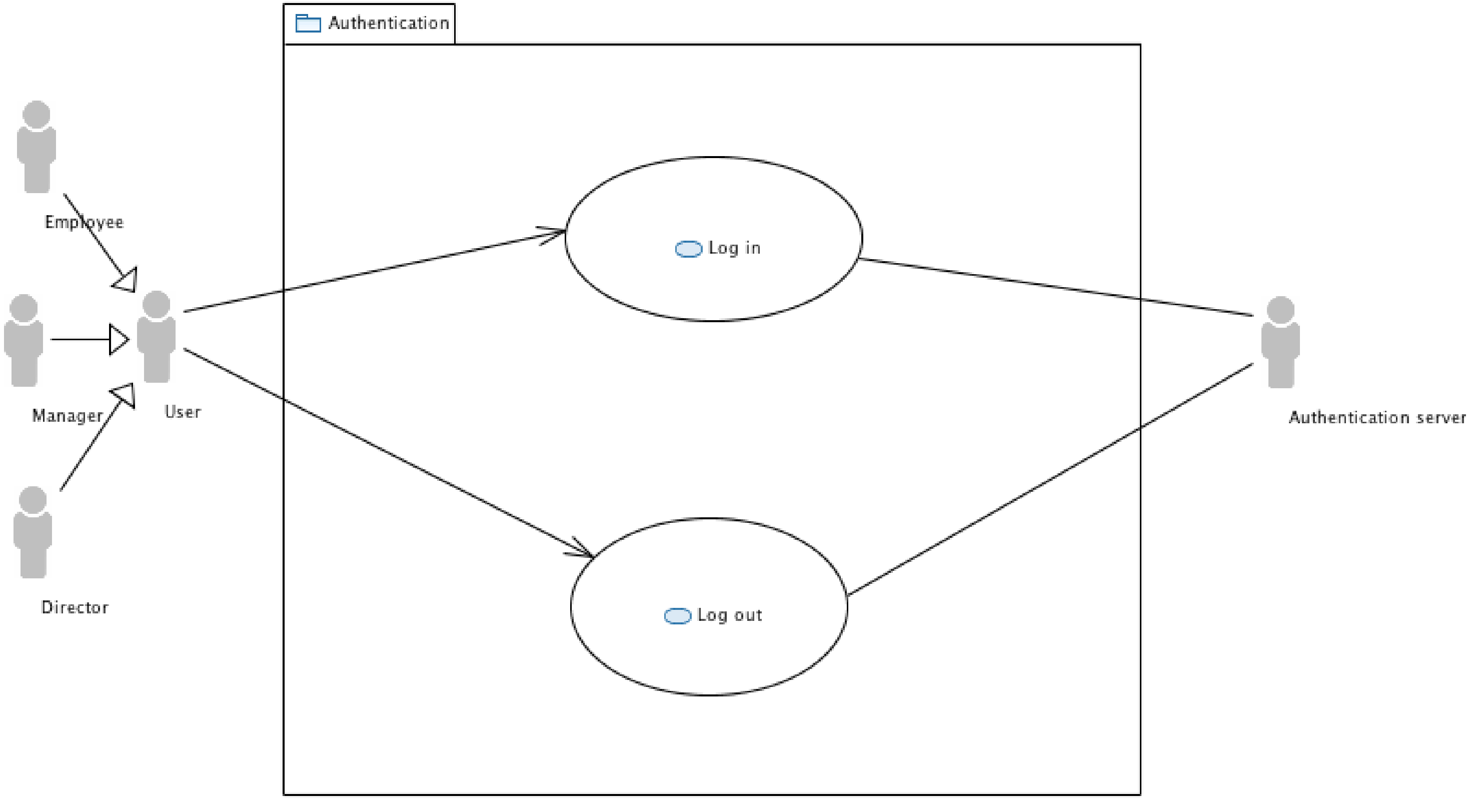
Stage 2

- Materials to be provided on the Moodle page:
  - The requirements for Stage 2.
  - A Papyrus project of UML use cases.
  - Scenario descriptions for the use cases.
  - Yuconz documents for personal details and performance review records.
- Scope: authentication, personal details and performance review.

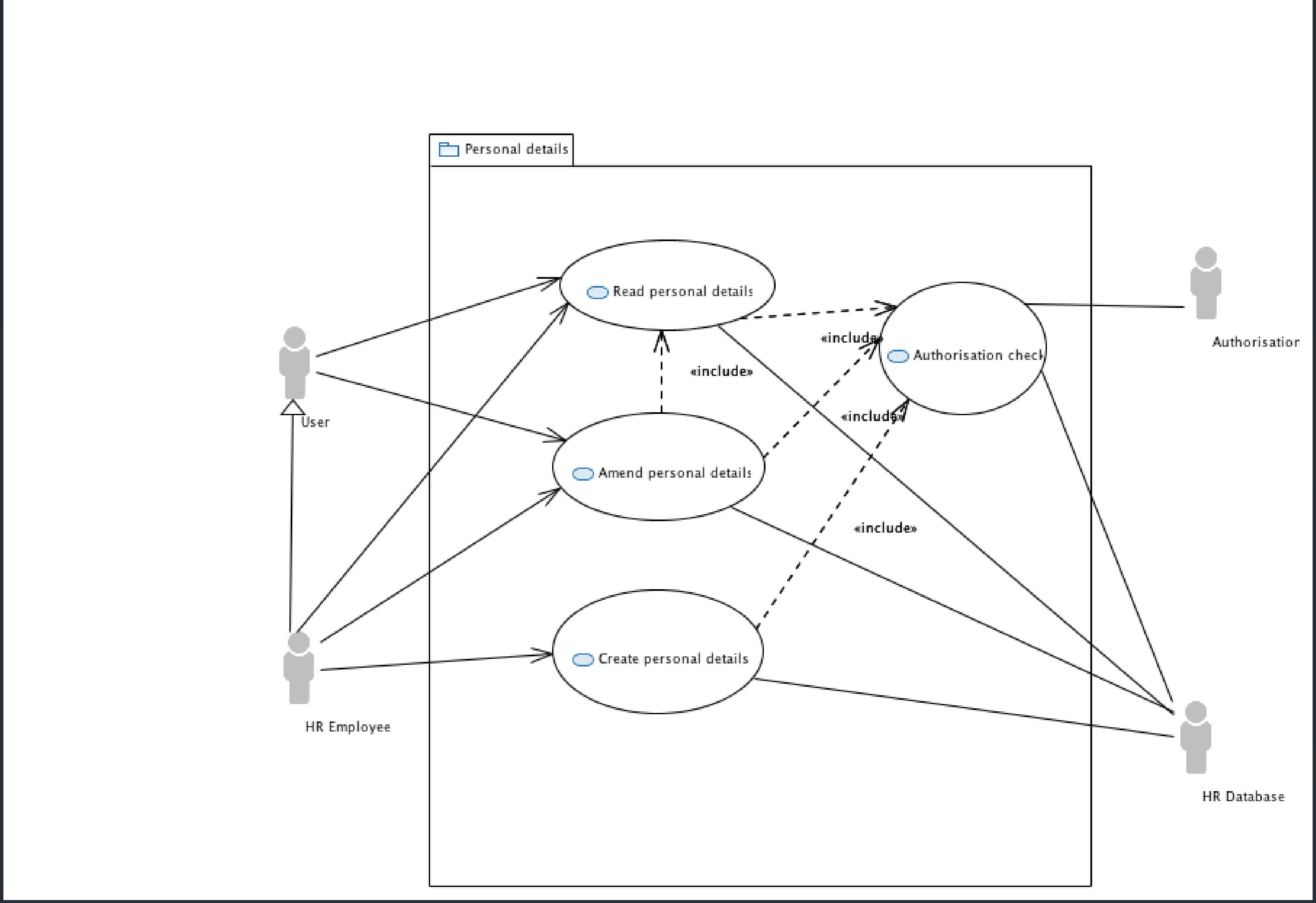
Use case diagrams

- Use case diagrams will be provided.
- All groups will start afresh from these diagrams.
- You should assume that these are potentially inaccurate, incomplete and/or inconsistent!
- It is your responsibility to check them, seek clarification, and make necessary adjustments.

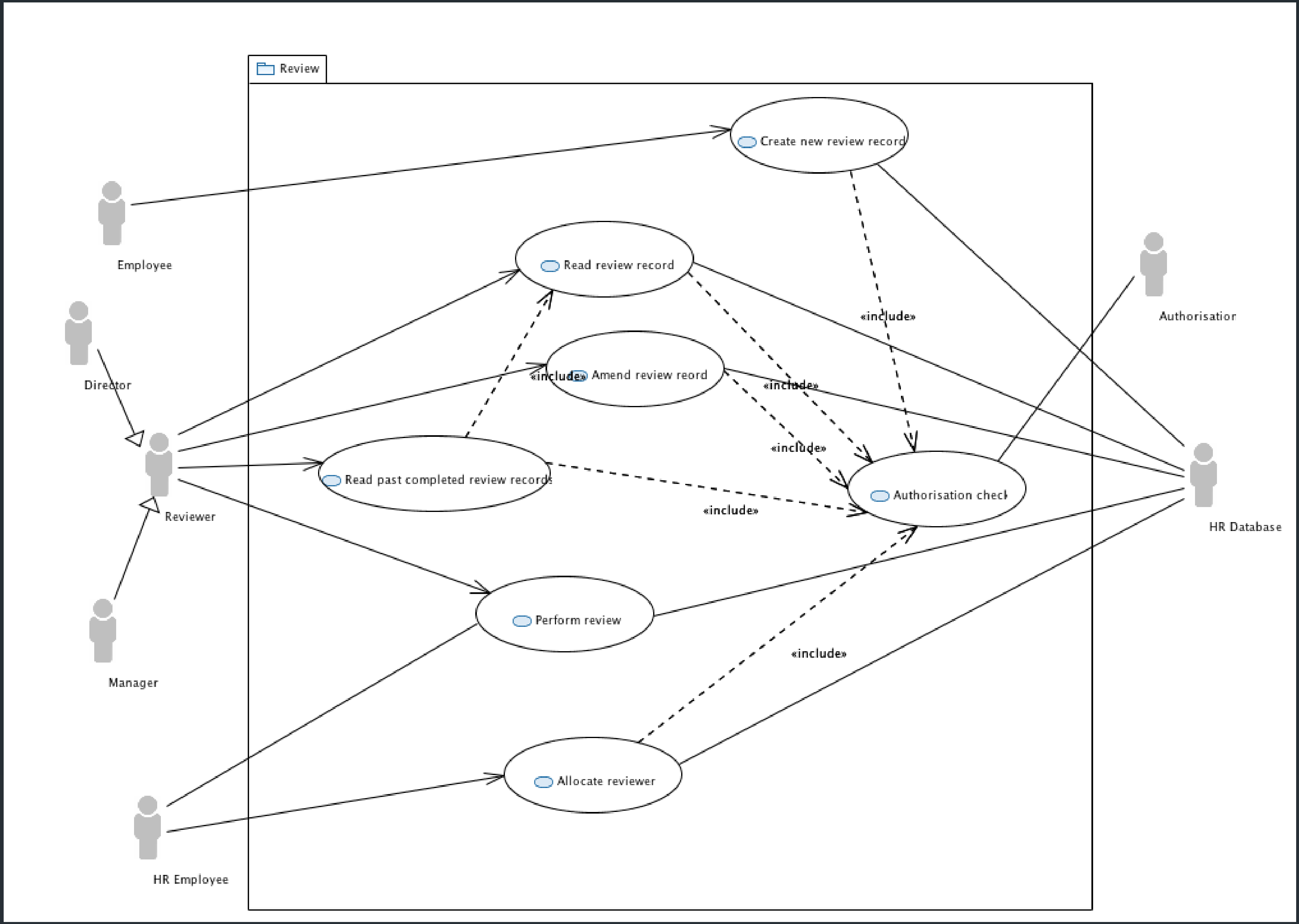
Authentication



Personal details



Performance review



Stage 2 Requirements

- Develop plans for how you will organise your group and schedule its effort over the coming term given the timetable of the assessment’s stages;
- Develop plans for how you will use your group’s gitlab space to coordinate your work and track issues as they arise;
- Develop plans for how you will ensure quality and consistency in everything the group does or produces;
- Analyse and refine the provided use cases;
- Start designing the system in UML from the revised use cases and other descriptions of the system.

Planning documents

- Project management: how you are going to plan and manage your work and monitor progress.
- QA: How you will achieve consistency and quality in project documents.
- Reviews: how you are going to review documents and software.
- Version control; e.g. use of gitlab. Resolving conflicts.
- Software tests, fault and bug reporting.
  - Derive tests, primarily, from the use cases.

Review and revision – the missing elements

- Did last week’s minuted actions happen?
- Did someone correct the typos in document X spotted during its review?
- Have regression tests been run since the fix of bug X?
- Have we revised the GANTT estimates in the light of experience?

Planning is the priority

- The priority of this stage is the planning documents.
- The class and sequence diagrams will likely not be completed.
  - But that doesn’t mean they should be ignored.

Group working

- The time available for each stage is short.
- Every group member must be proactive and responsive.
- You need to meet regularly – physically or virtually:
  - The workshops provide supervised common time.
  - Unused lecture slots.
  - Share timetables.
- Document every meeting – even virtual ones.

Group working - accountability

- Identify individual responsibilities.
- Review progress/completion at each meeting.
- Evidence progress.
- Revise planning document in the light of progress (or lack of it).
- Issue cards for repeated failure to deliver.

Group working

- Don’t partition the group into designers and coders or documenters and designers.
- The UML design and the code have to match.
  - Otherwise the design is pointless.
- Every individual needs to be able to use UML – and to be able to code!

Any questions?