NOVA SCHOOL OF SCIENCE & TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE

MEI - Master in Computer Science and Engineering

Internet Applications Design and Implementation

Course overview

Course Staff

Pedro Alves (theoretical + practical)

pedro.h.alves@gmail.com (*)

Hugo Pereira (practical)

hg.pereira@campus.fct.unl.pt

(*) Will update as soon as I have an institutional email

Goals

- Understand modern Internet application architecture
  - Principles of modular and maintainable software design
  - Enterprise design patterns, IoC, DI
  - Loosely coupled components

- Design and implement full-stack Internet applications
  - Frontend
  - Backend
  - APIs
  - Database
  - Security

Goals

By the end of this course, students should be able to design, implement and critically evaluate Internet-scale software systems

Syllabus

- History of web applications
- Modularization, coupling, IoC/DI patterns
- Software architecture: two-tier, three-tier, services, microservices
- RESTful applications
- Web Frameworks
- Server-side rendering: View templates, Forms, Navigation
- Persistence
- Security, Authentication and Authorization
- Traditional web application architecture
- Microservices architecture
- Client-side rendering: MPAs and SPAs; Reactive component-based frameworks
- Mobile application development

Lectures

- 2 hours is a long time to spend in a purely lecture-based format. Therefore, lectures will be interspersed with short individual exercises, small-group discussions, and brief presentations of conclusions to the class.
- Important: Bring paper and pen/pencil to the class!
- More important: No smartphones!!

Pratical classes

- **Tech Stack (mandatory)**
  - Server side: Spring Boot 3.5 + Java 17 + Kotlin 1.9 + Open API + JUnit 5 + MySQL
  - Client side: React + Typescript
- **Tools Stack**
  - git + GitHub
  - Intellij Ultimate Edition (available for free using student email)

- Practical class will always focus on the content lectured in the previous theoretical class (tuesday classes will be about the previous week lecture)
- First half of the semester, there will be some weekly assignments to submit via Drop Project (graded). Assignments will be published on Thursdays, deadline is the Sunday after next at midnight (example: published on 26 Feb, deadline on 8 Mar)
- Afterwards, it will be a mix of project support and optional (non-graded) lab exercises

Evaluation

- **Theoretical component (*Tc*) (60%, min: 9.5)**
  - two written Tests (T1=40%*Tc*, T2=40%*Tc*) or one Exam
- **Practical component *(Pc)* (40%, min: 9.5)**
  - weekly practical assignments (20%*Pc*)
    - individual, submitted to Drop Project, using GitHub
    - no limit or penalty on the number of submissions (attempts)
    - grade: 1 if it passes at least half of the tests, 0.5 if it passes one or more tests
  - 1 project (*P*), 3 mandatory parts (80%*Pc*) - P1 not graded, P2 70%*P*, P3 30%*P*
    - teams of 2 or 3 members
    - submissions using GitHub Classroom
    - graded oral discussion (individual) - project grade may be confirmed or lowered, failure is possible

Evaluation Schedule

Test 1	9 May
Test 2	1 Jun
Exam	22 Jun
Project submission (part 1)	29 Mar
Project submission (part 2)	6 May
Project submission (part 3)	29 May
Project discussions	8, 9 Jun

Attendance

Although attendance is not mandatory, students are strongly encouraged to attend all class sessions, including both lectures and lab classes. Missing classes may result in gaps in understanding of the course material.

While the lecture slides are comprehensive, class time also includes important discussions and collaborative activities that are essential to the learning process.

If you miss a class for any reason, you are responsible for reviewing the material and catching up on any missed content before the next session.

Code of Conduct

- You are expected to complete the practical assignments **on your own**.
- For the project, you are expected to work in teams of 2 or 3
  - We assume fair treatment and a fair balance of work between team members
- You may discuss the assignments/project with your classmates, but you may not share code
- If you have questions, talk with your instructor. You can also use zulip but never paste code or links to code

Academic Fraud

Examples of Academic Misconduct

- Copying during written exams
- Accessing unauthorized materials during written exams
- Sharing lab exercises code between students
- Sharing project code between groups
- Submission of work largely generated by AI systems

Important: Both the student who copies and the student who allows copying or shares code will be subject to the same penalties.

Use of GenAI

- The use of GenAI tools (e.g., ChatGPT, Copilot, Claude) is allowed and even encouraged both during class and at home.
- **Use them to learn, not to replace you!**
- During written tests, exams, and project discussion, the use of any GenAI tool is strictly prohibited

Resources

- https://iadi-2026.github.io/ (schedule, slides and other documents)
- drop project: https://iadi.dropproject.org/ (practical exercises)
- https://fct-iadi-2026.zulipchat.com/ (questions, announcements)
  (invitation link will be sent soon)

Bibliography

- Martin Fowler. **Patterns of Enterprise Application Architecture.** USA: Addison-Wesley Longman Publishing Co., Inc. 2002. isbn: 0321127420.
- Len Bass, Paul Clements, and Rick Kazman. **Software architecture in practice**, 3rd Edition. Addison-Wesley Professional, 2015. isbn: 0321815734.
- Chris Richardson. **Microservices Patterns**. Manning Publications, 2018. isbn: 9781617294549.
- Robert C. Martin. 2017. **Clean Architecture: A Craftsman's Guide to Software Structure and Design** (1st. ed.). Prentice Hall Press, USA.
- Laurentiu Spilca. **Spring Start Here**. Manning Publications, 2021.

GitHub user <-> Student info

Fill out this form ASAP

https://forms.gle/TEc17ZE5RGHagbnf7

You will only be allowed to submit to Drop Project after entering this info