# course syllabus and course requirements academic year 2024/2025 I. semester

|  |  |
| --- | --- |
| Course title | **Programming 5.** |
| **Course Code** | **IVB338ANMI** |
| ***Hours/Week: le/pr/lab*** | **1/2/0** |
| **Credits** | **4** |
| ***Degree Programme*** | **Computer Science Engineering BSc** |
| ***Study Mode*** | ***Full-time*** |
| ***Requirements*** | **Mid term exam** |
| ***Teaching Period*** | **2025/26/1** |
| ***Prerequisites*** |  |
| ***Department(s)*** | **Systems and Software Technologies** |
| ***Course Director*** | **Zidarics Zoltán** |
| ***Teaching Staff*** | ***Laborci Gergely*** |
|  |  |

# course description

This course provides students with practical, hands-on experience in designing and implementing modern distributed systems. The curriculum focuses on the principles of **microservices architecture**, guiding students through the development of a back-end system composed of multiple distinct services communicating via **HTTP**.

A key learning objective is to master real-time, two-way communication by connecting the server-side components to three different client applications—a **command-line interface**, a **web application**, and a **mobile application**—using **WebSocket technology**.

Beyond core technologies, the course emphasizes professional development practices, including version control in a GitHub monorepo, the creation of thorough API documentation, and the effective presentation of their work through a demonstration video and a defense of their architectural choices. Student performance is evaluated on the successful implementation and seamless integration of these components, with the highest marks awarded for a complete, multi-platform system that demonstrates a robust understanding of the covered technologies.

# SYLLABUS

## **1.** **goals and objectives**

The primary goal is to master the design and implementation of back-end systems using a microservices architecture with HTTP communication. Key objectives include using WebSockets for real-time communication across command-line, web, and mobile clients, while adopting professional practices like version control and API documentation. Upon completion, students will be able to build, document, and defend a complete, multi-platform distributed system.

## **2.** **course content**

|  |  |
| --- | --- |
|  | TOPICS |
| LECTURE | 1. **Introduction to Distributed Systems and Microservice Architecture** 2. **Inter-Service Communication: REST APIs and HTTP Protocols** 3. **Real-Time Communication on the Web: Fundamentals of the WebSocket Protocol** 4. **Client-Side Development I: Building and Integrating Command-Line (CLI) Applications** 5. **Client-Side Development II: Connecting Modern Web Clients to the Back-End** 6. **Client-Side Development III: Integrating Native and Hybrid Mobile Applications** 7. **Professional Development Practices: Monorepos, API Documentation, and Version Control** |

### **DETAILED SYLLABUS AND COURSE SCHEDULE**

|  |  |
| --- | --- |
| LECTURE | |
| week | **Topic** |
| 1. | Introduction to Distributed Systems and Microservice Architecture |
| 2. | Inter-Service Communication: REST APIs and HTTP Protocols |
| 3. | Real-Time Communication on the Web: Fundamentals of the WebSocket Protocol |
| 4. | Client-Side Development I: Building and Integrating Command-Line (CLI) Applications |
| 5. | Client-Side Development II: Connecting Modern Web Clients to the Back-End |
| 6. | Client-Side Development III: Integrating Native and Hybrid Mobile Applications |
| 7. | Professional Development Practices: Monorepos, API Documentation, and Version Control |
| 8. |  |
| 9. | *Project consultation* |
| 10. | *Project consultation* |
| 11. | *Project consultation* |
| 12. | *Project consultation* |
| 13. | Project Presentations and Evaluations |

## **3.** **assessment and evaluation**

##### **Attendance**

*In accordance with the Code of Studies and Examinations of the University of Pécs, Article 45 (2) and Annex 9. (Article 3) a student may be refused a grade or qualification in the given full-time course if the number of class absences exceeds 30% of the contact hours stipulated in the course description.*

***Method for monitoring attendance*** *(e.g.: attendance sheet / online test/ register, etc.)*

*attendance sheet*

##### **assessment**

***Course resulting in mid-term grade*** *(PTE TVSz 40§(3))*

The mid-term grade is determined by the final evaluation of the project. To be eligible for a passing grade, the project must be of acceptable quality, successfully defended by the student, and confirmed as an original, independent work.

The grade is awarded based on the scope of the completed components as follows:

* **Grade 2 (Sufficient) or 3 (Satisfactory):** Awarded for a functional back-end and a command-line (CLI) client. The specific grade within this range depends on the quality of the implementation.
* **Grade 4 (Good):** Awarded for a project that includes the back-end, CLI client, and a functional web client.
* **Grade 5 (Excellent):** Awarded for a complete project that includes the back-end and all three clients: CLI, web, and mobile.