Syrian Arab Republic

Higher Institute for Applied Sciences and Technology

Course Specification Document

Title	Advanced Distributed Systems

Credits	2.5 ECTS

Aims	This course aims to introduce the student to advanced concepts in distributed systems, including design issues, event ordering techniques, synchronization, fault tolerance, distributed storage systems, microservices, and containers. This course is an extension of the fundamentals of distributed systems, addressing additional challenges in building distributed systems and providing solutions. It also covers modern models and techniques that are gaining popularity. By the end of the course, the student will have a deeper understanding of concepts and challenges in building distributed systems, and he will be qualified to develop distributed programs and systems that incorporate advanced technologies to tackle these challenges.

Intended learning outcomes

On successful completion of this course, the student will be able to:

- Acquire advanced concepts in distributed systems.
- Understand more design issues and challenges related to building and developing distributed systems.
- Understand the principles of building distributed systems and their applications at different levels, including macro-level systems and containers.
- Implement new techniques in building distributed systems, such as: concurrency, recurrency, microservices, micro services, containers, scalability, and distributed storage systems
- Explore concepts, challenges, and solutions in building advanced distributed systems through a comprehensive case study.

Syllabus

- **Distributed system design:** Review of distributed system concepts, Distributed Shared Memory (DSM) implementations, issues in Remote Procedure Call (RPC) design, challenges in code migration, scalability considerations, and edge computing.
- **Event ordering:** Kaneko method, vector clocks, and consistent global state.
- **Synchronization in distributed systems:** Consistency models, distributed transactions.
- **Fault tolerance:** CAP theorem theory, types of faults, redundancy types.
- **Distributed file systems:** GFS, HDFS, Amazon S3.
- **Microservices architecture:** Decentralized design, communication between microservices.
- **Containers and orchestration:** Docker containerization, Kubernetes orchestration, managing distributed containers, scalability, and load balancing.
- **Case study:** Advanced distributed application - Amazon DynamoDB.

Syrian Arab Republic

Higher Institute for Applied Sciences and Technology