Module Description, available in: EN

Mobile Computing

General Information

Number of ECTS Credits

3

Module code

TSM_MobCom

Valid for academic year

2026-27

Last modification

2025-10-17

Coordinator of the module

Thomas Amberg (FHNW, thomas.amberg@fhnw.ch)

Explanations regarding the language definitions for each location:

- Instruction is given in the language specified for each location and module execution.
- Documentation is available in the language(s) listed for each location and module execution. If the documentation is in multiple languages, the percentage distributed is indicated (100% = all documentation provided).
- The examination, including both questions and answers, is provided entirely (100%) in the language(s) specified for each location and module execution. The exams are on-site.

	Lausanne			Lugano	Zurich		
Instruction					**X** E 100%		
Documentation					**X** E 100%		
Examination					**X** E 100%		

Module Category

TSM Technical scientific module

Profile Recommendations

This module is recommended for the following profiles:

	Zurich
Profile	CS

Lessons

2 lecture periods and 1 tutorial period per week

Entry level competences

Prerequisites, previous knowledge

Students have working knowledge in:
• Programming in Java or Kotlin
• Software development and frameworks
• Desktop or mobile user interface development
• Internet protocols, HTTP and sending Web requests

• Students bring a laptop to class.
• Students bring an Android phone to class (if available).

Brief course description of module objectives and content

This module enables students to develop advanced, native applications for the Android mobile operating system and get a solid understanding of mobile computing concepts. Building on the basics of mobile application development, this course covers a selection of application programming interfaces for on-board sensors and connectivity options for the integration with backend services, IoT platforms and peripheral devices. Lecturers share their experience and best-practices from recent projects involving mobile computing. Students work with both emulators and real devices.

Aims, content, methods

Learning objectives and competencies to be acquired

Application Development (50%)
• Students know how to design and implement native applications for mobile devices running Android, the most widely used mobile platform.
• Students have basic knowledge of user-interface design guidelines and techniques relevant for mobile application design.
• Students can describe the integration of their application with a cloud backend.

Sensors and Connectivity (50%)
• Students know how to use on-device sensor APIs for motion, position and environment.
• Students have basic knowledge of connectivity options like Near Field Communication (NFC), Bluetooth Low Energy (BLE) and Wi-Fi.
• Students can describe the integration of their application with a peripheral device or IoT platform using request/response or messaging protocols.
• Students have basic knowledge of prototyping a peripheral IoT device with an easy-to-use embedded hardware platform, sensors and actuators.

Module content with weighting of different components

Application Development
• Development of native mobile applications for Android, including user interfaces
• Specific aspects in mobile application programming such as application lifecycle, data storage, data synchronization with a cloud backend, and security of mobile applications.

Sensors and Connectivity
• Development with on-device sensor APIs for motion, position and environment.
• Specific aspects of connecting to peripheral devices with connectivity options like NFC, BLE and Wi-Fi, prototyping an IoT device and integration with IoT platforms.

Teaching and learning methods

• Ex-cathedra teaching
• Team project
• Exercises
• Self-Study

Literature

Android
• IDE https://developer.android.com/studio
• Docs https://developer.android.com/docs
• Source Code https://source.android.com/

Assessment

Additional performance assessment during the semester

The module contains additional performance assessment(s) during the semester. The achieved mark of the additional performance assessment(s) applies to both the regular and the resit exam.

Description of additional performance assessment during the semester

• A graded team project
• Counting 30% (project) and 70% (exam result).
.

Basic principle for exams

As a rule, all standard final exams are conducted in written form. For resit exams, lecturers will communicate the exam format (written/oral) together with the exam schedule.

Standard final exam for a module and written resit exam

Kind of exam

Written exam

Duration of exam

120 minutes

Permissible aids

Aids permitted as specified below:

Permissible electronic aids

• Computer with internet, Web search.
• Course repositories, slides and code.
• No communication (phone, chat, AI, …).

Other permissible aids

• Open book examination.

Exception: In case of an electronic Moodle exam, adjustments to the permissible aids may occur. Lecturers will announce the final permissible aids prior to the exam session.

Special case: Resit exam as oral exam

Kind of exam

Oral exam

Duration of exam

30 minutes

Permissible aids

No aids permitted