

Mobile Solutions

Syllabus: BIS3064 – Mobile Solutions

Prof. Dr. Thomas Schuster - Hochschule Pforzheim / Pforzheim University

Course	BIS3064 – Mobile Solutions
Workload	3 ECTS credits (2 SWS)
Level	Advanced level
Prerequisites	<ul style="list-style-type: none">• BIS2050 - Programming and Database Systems• BIS2070 - Business Process Management and Transactional Processing Systems.
Validity	Winter 2020/2021
Teaching	Prof. Dr. Thomas Schuster
Place and time	see LSF
Information and additional documents	https://lms.hs-pforzheim.de/course/view.php?id=3702

Objective

Mobile computing devices have become ubiquitous in our communities. In this course, we focus on the creation of mobile solutions for various modern platforms, including major mobile operating systems (especially Android and iOS). Topics include mobile device architecture, programming languages, software engineering, user interface design, and app distribution.

Topics

- Client Hardware (Desktop vs. Mobile)
- Hybrid Application Development based on Apache Cordova and JavaScript (Ionic)
- Android Development (Java)
- iOS Development (Swift)
- RESTful and Non-RESTful apps
- Creating and Incorporating Web/Cloud Services
- Mobile Sensors
- Security and Trust Management
- Privacy and Ethics
- Usability and Accessibility

Event Outline (Preliminary)

Lecture	Content
1	Welcome, overview, organization of the module, dates and groups of exercises
2-4	Platform architectures
5-8	Hybrid application development
9-12	Web & cloud services
13-15	Data processing

Teaching and learning concept

This event focusses students in the sixth semester of Business Administration / Digital Enterprise Management (bachelor's degree programme at the faculty of economics and law of the Pforzheim University). The lecture is accompanied by exercises (practical cases and a group project), which empower the students to utilize and develop mobile application software for company purposes. It is expected that students spend time in self-studying in addition to the lectures and exercises (about 30 hours in presence), roughly: about 15 hours of exam preparation, 30 hours of project work, pre-and post-processing the lectures (including literature study), and finally, 15 hours of presentation of problem solving.

The lecture will be held in a seminar style to convey the necessary programming knowledge. This means that leading questions, overviews, definitions and examples, and short exercises will be used to present the contents in an application-oriented style. It is expected that each student actively participates in the lecture.

My intention as a lecturer

I am always happy to serve as conversation partner and support you with advice. Please contact me during the lecture or via yammer (or e-mail).

I am willing to contribute to your successful learning and provide an understanding of the practical impacts of the learning content.

Comprehension questions should, if possible be equated during the lessons. Your comments, which serve the progress of all others are also welcome. Also, suggestions for improvements, and constructive criticism are very welcome. My goal is that you can successfully pass the exam, however the major part of the work is up to you.

General

- I have an open door policy, in that if my door is open, by all means stop on in and say hi or ask a question. If my door is closed, then I'm heads down on some task, on the phone, in a meeting... It's always a good idea to email me or send a note on yammer before coming to make sure I'm here.
- I can't stress enough that yammer is the best way to get in touch with me.
- Please don't hesitate to contact me if you have any problems, concerns, questions, or issues regarding the course, material, or anything else in the class.

Project Groups

The projects to be undertaken are group projects. Details of the groups are:

- Normal group size is three. Individuals may be able to work by themselves with good reason. Expectations will not be adjusted due to smaller group size. Groups of five and more are strongly discouraged, but may be allowed under extreme circumstances. Expectations for groups of four and more students are higher than those for groups of three.
- You may have different groups for the different projects.
- In general, all group members will receive the same grade for graded assignments. However, group members will evaluate their peers and any student who appears to not be contributing may be penalized.
- Each group will be responsible for assigning tasks to its group members (according Disciplined Agile Delivery).

You are expected to work as a member of your group in this course and cooperate with your colleagues. Cooperation means attending group meetings, completing your assignments properly and on time, letting your group know if you will be out of town, responding to e-mail from your group, and so on. If there is a lack of cooperation by any group member, it must be brought to the attention of the instructor as soon as it happens. If the lack of cooperation is serious, the offending group member's semester grade will be lowered.

Academic Honor

- In general, we expect that you will be using code, examples, and ideas from many different websites and resources for your projects. This is allowed within reason. Wholesale copying of an entire project is definitely not allowed. Using code to round out a feature is allowed. If you ever have a question about what is or is not appropriate, ask first!
- In **all** cases, you need to cite all sources at the top of the file where the code or algorithm was used **and** you should note all sources in your documentation.
- Failure to properly attribute your sources will result in failing the complete project.

Code of Conduct for Students

In this course, there will be a focus on working well together and learning about the development process. A large portion of that process involves interpersonal skills and conflict management. Students and staff are all expected to treat each other with respect. This includes, but certainly is not limited to:

- Excessive web browsing during class
- Disrespectful language
- Promptness for all deadlines and class meetings
- Quality work
- Not working well with your partner
- Collaborating with other teams
- Be fair to the other students!
- Be punctual and don't leave early!

The number one problem with professionalism in class is the overuse of laptops and mobile devices. Taking notes on a laptop and following along with the slides is welcome and encouraged. Doing work for other classes, watching videos, chatting, or anything else that distracts

from your ability to learn and follow along (or anyone around you), will result in a professionalism penalty. Students can and will be penalized for unprofessional behavior.

Your class work might be used for research purposes. For example, we may use anonymized student assignments to design algorithms or build tools to help programmers. Any student who wishes to opt out can contact the instructor or do so after final grades have been issued. This has no impact on your grade in any manner.

Learning Outcomes

1. LO1. Be exposed to technology and business trends impacting mobile applications
2. LO2. Be competent with the characterization and architecture of mobile applications.
3. LO3. Be competent with understanding and description of enterprise scale requirements of mobile applications.
4. LO4. Be competent with designing and developing mobile applications using one application development framework.

Relationship Course Outcomes to general Competence Goal

General program goals of competences can be found here: [Competence Goals](#). The relationship between the learning outcomes of this course and the program outcomes is shown in the table below:

Goals	1				2		3	4	5		6	7		
Outcomes	1.1	1.2	1.3	1.4	2.1	2.2			5.1	5.2		7.1	7.2	7.3
LO1					XXX	XXX		XX						
LO2						XXX			X	XX			XXX	X
LO3	X			XX			XXX					XXX	XXX	
LO4							XXX				XXX	XXX		XXX

X - contribution; XX - extended contribution; XXX - focussed contribution

Recommended literature

- B. Phillips, K. Marsicano, and C. Stewart, Android Programming: The Big Nerd Ranch Guide. Atlanta, Georgia: Pearson Education, 2017.
- N. Smyth, Android Studio 3.0 Development Essentials - Android 8 Edition, 1st ed. CreateSpace Independent Publishing Platform, 2017.
- D. Griffiths and D. Griffiths, Head First Android Development: A Brain-Friendly Guide, 2nd edition. Beijing Boston Farnham Sebastopol Tokyo: O'Reilly UK Ltd., 2017.
- V. Nahavandipour, iOS 11 Swift Programming Cookbook: Solutions and Examples for iOS Apps, 1st ed. O'Reilly Media, 2017.
- A. Tucker, Xamarin Unleashed, 01 ed. Place of publication not identified: Pearson Professional, 2018.
- M. Neuburg, Programming iOS 11: Dive Deep into Views, View Controllers, and Frameworks, 1st ed. O'Reilly Media, 2017.
- Additional documents and information: <https://lms.hs-pforzheim.de/>

Examination and Grading Criteria

The course is graded as follows:

- Class Participation: 10%
- Project: 50%
- Iterations (Team Work Organisation): 15%
- Final Product Presentation: 10%
- Report (Homework): 15%

The project grade is calculated from its iterations and collaborative team work. All reviews serve as checkpoints; important: contents (product increment) and team collaboration (user stories and iteration planning) outmatch the presentation method (e.g. no powerpoint for reviews needed). The report homework is graded individually.

The following evaluation scheme is used: 'Very good' means exceptional performance that is far above the average. 'Good' means good performance, that is above the average. 'Satisfactory' means average performance, which certainly has shortcomings, but basically complies with the requirements. 'Adequate' means underperformance with conspicuous defects. 'Poor' means not acceptable performance, which no longer complies with the requirements.