SYLLABUS

Mobile Application Development

University year 2025-2026

1. **Information regarding the programme**

1.1. Higher education institution	Babeş-Bolyai University
1.2. Faculty	Faculty of Mathematics and Computer Science
1.3. Department	Department of Computer Science
1.4. Field of study	Computer Science
1.5. Study cycle	Bachelor
1.6. Study programme/Qualification	Computer Science
1.7. Form of education	Full time

2. **Information regarding the discipline**

<table>
<tr><td colspan="3">2.1. Name of the discipline</td><td colspan="3"><b>Mobile Application Development</b></td><td colspan="3">Discipline code</td><td><b>MLE5078</b></td></tr>
<tr><td colspan="4">2.2. Course coordinator</td><td colspan="6">Dan Cojocar, PhD</td></tr>
<tr><td colspan="4">2.2. Seminar coordinator</td><td colspan="6">Dan Cojocar, PhD</td></tr>
<tr><td>2.4. Year of study</td><td><b>3</b></td><td>2.5. Semester</td><td><b>5</b></td><td>2.6. Type of evaluation</td><td><b>E</b></td><td colspan="3">2.7. Discipline regime</td><td><b>Compulsory</b></td></tr>
</table>

3. **Total estimated time** (hours/semester of didactic activities)

<table>
<tr><td>3.1. Hours per week</td><td><b>3</b></td><td>Of which: 3.2 course</td><td><b>2</b></td><td>3.3. Seminar/laboratory/project</td><td><b>1 lab</b></td></tr>
<tr><td>3.4. Total hours in the curriculum</td><td><b>42</b></td><td>Of which: 3.5 course</td><td><b>28</b></td><td>3.6. Seminar/laboratory/project</td><td><b>14</b></td></tr>
<tr><td colspan="5"><b>Time allotment for individual study (ID) and self-study activities(SA)</b></td><td><b>hours</b></td></tr>
<tr><td colspan="5">Learning using manual, course support, bibliography, course notes (SA)</td><td>10</td></tr>
<tr><td colspan="5">Additional documentation (in libraries, on electronic platforms, field documentation)</td><td>20</td></tr>
<tr><td colspan="5">Preparation for seminars/labs, homework, papers, portfolios, and essays</td><td>15</td></tr>
<tr><td colspan="5">Tutorship</td><td>8</td></tr>
<tr><td colspan="5">Evaluations</td><td>5</td></tr>
<tr><td colspan="5">Other activities</td><td>0</td></tr>
<tr><td colspan="4"><b>3.7. Total individual study hours</b></td><td colspan="2"><b>58</b></td></tr>
<tr><td colspan="4"><b>3.8. Total hours per semester</b></td><td colspan="2"><b>100</b></td></tr>
<tr><td colspan="4"><b>3.9. Number of ECTS credits</b></td><td colspan="2"><b>4</b></td></tr>
</table>

4. **Prerequisites** (if necessary)

4.1. Curriculum	Object Oriented Programming, Advanced Programming Methods
4.2. Competencies	Average Java/Kotlin programming skills

5. **Conditions** (if necessary)

4.1. For the course	Projector
4.2. For the seminar/lab activities	Internet access and ability to use personal laptops

6.1. **Specific competencies acquired**[1]

Professional/ essential competencies	• C1.3 Elaboration of adequate source codes and unitary testing of some components in a known programming language, based on given design specifications. • C1.5 Development of program units and elaboration of the corresponding documentation. • C6.3 Techniques for installation, configuration, and administration of systems and computer networks.
Transversal Competencies	• CT1 Application of efficient and organized work rules, of responsible attitudes towards the didactic-scientific domain, to creatively value one's own potential, with respect towards the principles and norms of professional etic. • CT3 Use of efficient methods and techniques to learn, inform, research and develop the abilities to value the knowledge, to adapt to requirements of a dynamic society and to communicate in Romanian language and in a language of international circulation.

[1] One can choose either competences or learning outcomes, or both. If only one option is chosen, the row related to the other option will be deleted, and the kept one will be numbered 6.

6.2. **Learning outcomes**

Knowledge	The graduate can design/develop/debug basic mobile applications.
Skills	The graduate can apply architectural styles, design patterns, and best practices in the field to design mobile applications.
Responsibility and autonomy	• The graduate is familiar with the tools used for testing, debugging, and validating mobile applications. • The graduate is familiar with project management tools, version control systems, and continuous integration/continuous delivery (CI/CD) concepts, methods, tools.

7. **Objectives of the discipline** (outcome of the acquired competencies)

7.1. **General objective of the discipline**	• To acquire an insight into how to build mobile applications. • Knowledge of key base concepts for developing mobile applications.
7.2. **Specific objective of the discipline**	• To attain a basic level of the design principles of a mobile application. • To get a good grasp of basic mobile development components. • To be a solid base for preparing to become a Mobile Application programmer.

8. **Content**

8.1. **Course**	Teaching methods	Remarks
1. Base Android tooling • Android Studio. • Activity/Fragment lifecycle. • User interfaces.	Exposure: description, examples, discussion of case studies, live demo.	
2. Lists and rest resources • Views • Background processing • Networking		
3. Master-details and rest resources • More views: NavigationDrawer • OkHttp, JsonReader, JsonWriter • ContentProviders		
4. Local persistence • Preferences and Files • Databases: SQLite, Room, Realm.		
5. Securing mobile apps • Android security model • JSON Web Tokens • OAuth 2.0		
6. Synchronizing data • WebSockets • Local synchronization services • LoaderManagers		
7. Reactive programming • Realm - real-time database • Rx - reactive programming • Coroutines		
8. System services and sensors • Services • Processes • Sensors		
9. Animations • ValueAnimator. • ObjectAnimator. • Transitions framework		
10. Firebase Services		

• Authentication • Database • Remote Config		
11. Monetize • Ads • In-app billing • Firebase		
12. Awareness and nearby • Anticipate and react • Nearby • Physical Web		
13. Test your app • Junit • Mockito • UI Automator, Expresso • Firebase test lab • Performance testing		
14. Exam simulation and discussions • Sample exam requirement • Exam discussions	Discussion of case studies, exam discussions.	
Bibliography • Android Development. http://developer.android.com/index.html • React Native. https://facebook.github.io/react-native/ • Flutter. https://flutter.io/docs • Android codelabs. https://developer.android.com/get-started/codelabs • Vogella. Android Development Tutorials. http://www.vogella.com/android.html		

8.2. Seminar/Laboratory	Teaching methods	Remarks
1. Getting Started • Create Android and Flutter sample applications. • Discuss the L1 and L2 assignments.	Exposure: description, discussion. Evaluation.	The lab is structured as 2 hours classes every second week.
2. Specification evaluation.		
3. CRUD Specifications discussion. Specification reevaluation.		
4. Evaluate the UI module.		
5. Evaluate the local persistence logic.		
6. Evaluate the network/online communication logic.		
7. Bonus/Final project evaluation.		
Bibliography • Android Development. http://developer.android.com/index.html • React Native. https://facebook.github.io/react-native/ • Flutter. https://flutter.io/docs • Android codelabs. https://developer.android.com/get-started/codelabs • Vogella. Android Development Tutorials. http://www.vogella.com/android.html		

9. **Corroborating the content of the discipline with the expectations of the epistemic community, professional associations, and representative employers within the field of the program**

- The course aligns with the IEEE and ACM curriculum recommendations for Computer Science programs.
- It is included in the study programs of major universities both in Romania and internationally.
- The course content is considered essential by software companies for developing solid, industry-relevant programming skills.

10. **Evaluation**

Activity type	10.1. Evaluation criteria	10.2. Evaluation methods	10.3. Percentage of final grade
10.4 Course	• The basic principle of the domain. • Apply the course concepts. • Problem-solving.	Practical examination	40%

10.5 Seminar/ laboratory	• Be able to implement course concepts and algorithms. • Apply techniques for different classes of programming languages.	• Practical evaluation during the semester. • Portfolio.	60%
10.6. Minimum standard of performance			
• Attend 90% of lab activities during the semester. • At least grade 5 (1 to 10 scale) in all activities, seminar/lab, and written exam. • The final grade must be at least 5.			

11. Labels ODD (Sustainable Development Goals)[2]

Not applicable.

Date:

Signature of course coordinator,
Dan Cojocar, PhD

Signature of seminar coordinator,
Dan Cojocar, PhD

Date of approval:

Signature of the head of department,
Adrian Sterca, PhD

[2] Keep only the labels that, according to the Procedure for applying ODD labels in the academic process, suit the discipline and delete the others, including the general one for Sustainable Development – if not applicable. If no label describes the discipline, delete them all and write “Not applicable.”.