Syllabus: CSE 5236: Mobile Application Development

Instructor: Adam C. Champion, Ph.D.

Updated: January 5, 2019

Course Information

Instructor:	Adam C. Champion, Ph.D.
Office:	409 Caldwell Labs (primary)
Contact:	`champion@cse.ohio-state.edu`
Web:	`http://web.cse.ohio-state.edu/~champion.17/5236`
Office Hours:	Specified on course website
Class Times:	Specified on course website
Prerequisites:	Project course or graduate standing
Students:	Seniors and graduate students
Credit:	3 UG/G credit hours
Teaching Assistants:	TBD
Graders' Office Hours:	TBD, and by appointment

Course Description

- *Catalog description:* Mobile application development frameworks; Architecture, design and engineering issues, techniques, methodologies for mobile application development.
- *Additional Notes:* Course material includes readings from textbooks, reference books, and research papers and articles.

Course Objectives

- Course introduction. Lecture hours: 0.5 hour.
- Characteristics of mobile applications. Lecture hours: 1.5.
- History of mobile application frameworks. Lecture hours: 1.
- Overview of mobile application development languages: Objective-C and Java. Lecture hours: 3.
- Application models of mobile application frameworks. Lecture hours: 3.
- User-interface design for mobile applications. Lecture hours: 3.
- Managing application data. Lecture hours: 3.
- Integrating with cloud services. Lecture hours: 3.
- Integrating networking, the OS and hardware into mobile-applications. Lecture hours: 3
- Addressing enterprise requirements in mobile applications: performance, scalability, modifiability, availability, and security. Lecture hours: 3.
- Testing methodologies for mobile applications.Lecture hours: 3.
- Publishing, deployment, maintenance and management.
- Final exam. Lecture hours: 2.

Learning Outcomes

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

Means for Achieving Learning Outcomes

Active Learning: Teaching will be through in-class activities and discussion, and a project.

Class Discussions: You are expected to participate in discussions about the lecture material as well as the research presentations.

Project: There will be an integrative application development project. Students will need to reserve time outside of class to work on the project.

Readings

Extracts from the following books will be used in class:

1. Bill Phillips, Chris Stewart, Brian Hardy, and Kristin Marsicano, *Android Programming: The Big Nerd Ranch Guide*, Big Nerd Ranch LLC, 3rd edition, 2017;
2. Rajiv Ramnath, Roger Crawfis, and Paolo Sivilotti, *Android SDK 3 for Dummies*, Wiley.

The following books (available on Safari Books Online) may be used as references:

1. B. Phillips et al., *Android Programming: Big Nerd Ranch Guide* (as mentioned above);
2. Christian Keur and Aaron Hillegass, *iOS Programming: The Big Nerd Ranch Guide*, 6th edition, 2016;
3. Valentino Lee, Heather Schneider, and Robbie Schell, *Mobile Applications: Architecture, Design and Development*, Prentice Hall, 2004;
4. Tomasz Nurkiewicz and Ben Christensen, *Reactive Programming with RxJava*, O'Reilly Media, 2016;
5. Raoul-Gabriel Urma, Mario Fusco, and Alan Mycroft, *Java 8 in Action: Lambdas, Streams, and Functional-Style Programming*, Manning Publications, 2015;
6. Benjamin J. Evans and Martijn Verburg, *The Well-Grounded Java Developer: Vital Techniques of Java 7 and Polyglot Programming*, Manning Publications, 2013;
7. Brian Fling, *Mobile Design and Development*, O'Reilly Media, Inc., 2009;
8. Maximiliano Firtman, *Programming the Mobile Web*, O'Reilly Media, Inc., 2nd ed., 2013;
9. Cristian Crumlish and Erin Malone, *Designing Social Interfaces*, 2nd ed., O'Reilly Media, Inc., 2014;
10. Suzanne Ginsburg, *Designing the iPhone User Experience: A User-Centered Approach to Sketching and Prototyping iPhone Apps*, Addison-Wesley Professional, 2010;
11. Benjamin Muschko, *Gradle in Action*, Manning Publications, 2014;

12. Craig Larman, *Applying UML and Patterns: A Guide to Object-Oriented Analysis and Design and Iterative Development*, 3rd ed., Prentice Hall, 2004.

Online research sites include:

- Safari Text Books Online, `http://library.ohio-state.edu/search/y?SEARCH=Safari`
- Business Source Complete, `http://library.ohio-state.edu/record=e1000557`
- ACM Digital Library, `http://library.ohio-state.edu/record=e1000050`
- IEEE Xplore, `http://library.ohio-state.edu/record=e1000005`

Grading Criteria

The course is graded as follows:

Class Participation:	10%
Project:	35%
Midterm:	15%
Final:	25%
Report:	15%

The *Project* grade is calculated from six project checkpoint grades:

Checkpoint 1. Students install the sample Tic-Tac-Toe application on an Android emulator on their personal laptop computers or on real-world mobile devices. Students submit one or more screenshots of the application running on the emulator or devices. **(5 points)**

Checkpoint 2. Students design mobile applications for the Android or iOS platforms that uniquely meet clear needs in today's markets. Student design documents include narratives, categorized use cases, screen flows, and database schemata (if applicable). Students elaborate their application designs to graders. **(5 points)**

Checkpoint 3. Students submit screenshots of their applications in progress showing debugging facilities (e.g., setting breakpoints) in their IDEs (Android Studio or Xcode), results of application method profilers, and logs of activity lifecycle methods in their IDEs. Students also submit `.zip` files or their code. Students demonstrate installation of their applications on real-world mobile devices and use of debugging facilities to graders. **(10 points)**

Checkpoint 4. Students submit screenshots of their persistent data storage schema, source code repository (with version control support), and (partial) code classes. Students also submit `.zip` files of their code. Students demonstrate their app's ability to perform creation, retrieval, update, and delete tasks with persistent data, their use of a shared code repository, and that their class implementations indicate adherence to "best practice" object-oriented design principles. **(10 points)**

Checkpoint 5. Students submit lists of use cases corresponding to their applications' *functional* requirements as well as `.zip` files of their code. Students demonstrate their applications' functionalities to graders on real-world mobile devices, who verify that the use cases corresponding to functional requirements are implemented. Students' applications may not function during screen rotation or in the absence of network connectivity and/or current device location. **(20 points)**

Checkpoint 6. Students submit lists of use cases corresponding to their applications' *non-functional* requirements as well as `.zip` files of their code. Students are expected to improve application performance (compared with that of Checkpoint 3) using the application method profiler, perform unit tests, and implement a non-functional requirement of their choice (such as design refinements or enhanced security). Students' applications are expected to function during device rotation and in the absence of network connectivity and/or current device location. Students demonstrate their applications' functionalities to graders on real-world mobile devices, who verify that the use cases corresponding to non-functional requirements are implemented. **(20 points)**

Checkpoint 1 is graded individually; Checkpoints 2–6 are graded as student groups. For Checkpoints 2–6, groups meet with graders and all group members receive the same grades, with one exception: if certain group member(s) miss their appointments with the graders for project checkpoint(s), these member(s) receive 90% of their groups' grades for the respective checkpoint(s). For example, if Alice and Bob form a group, Bob misses the group's meeting with the grader for Checkpoint 3, and the group receives 10/10 points, then Bob receives 9/10 points.

Course Policies

The following course policy applies to the classroom and team interactions—you are expected to show the same responsibility to your teammates as you do to me:

- *Attendance:* You are expected to attend all classroom sessions and team meetings, and do all the assigned work, self-study and readings.
- *Class Preparation:* You are expected to be prepared for class, participate in the discussion, answer questions, etc., on the topic for the day.
- *Missed Classes, Exams, Presentations, and Meetings:* You are responsible for all class lectures, including handouts and notes. There will be no make up exams, presentations, lectures, etc.
- *Assignments:* Assignments (if any) are due in hard copy at the beginning of class on the due date. Assignments must be typed and formatted appropriately.
- *Contact and Class Information:* The course web site (see above) and email communication will be extensively used and relied upon for this course. Please familiarize yourself with these resources, provide me with your email address in the questionnaire, and check your email at least once every day.
- *Sharing and Attribution of Intellectual Property and Information:* You are free to exchange and use any information from each others' projects. You may also freely research and use information legally available from the Web or other sources. However, you ***must*** properly attribute each piece of borrowed intellectual property.

Relationship to BS-CSE Course Outcomes

The relationship between the learning outcomes of this course and the CSE Outcomes (`http://web.cse.ohio-state.edu/~soundarajan.1/ugsc/programs/cseobjectives.shtml`) is shown in the table below:

Learning Outcome	CSE (a)	CSE (b)	CSE (c)	CSE (d)	CSE (e)	CSE (f)	CSE (g)	CSE (h)	CSE (i)	CSE (j)	CSE (k)	CSE (l)	CSE (m)	CSE (n)
LO1								X	X	X				
LO2			X		X						X	X		
LO3			XX		XX						XX	XX		XX
LO4			XX		XX						XX	XX		XX
Summary			XX		XX			X	X	X	XX	XX		XX

Representative Assignments

Students will be required to implement a mobile application on two separate platforms.