

ISAT 480: Mobile App Development

Spring 2014 Syllabus

Professors: Dr. Morgan C. Benton & Dr. Emil H. Salib

Course Description

This is an introductory course in the development of applications for mobile devices such as smartphones and tablets. Participants will be given an overview of the current status and social context of the mobile app development industry, and have the opportunity to participate in and build skills covering the entire lifespan of mobile app development up to and including deployment.

Prerequisites and Expectations

Since this is primarily a programming course, participants are expected to come with a solid foundation in basic programming concepts. More specifically, the following would be extremely useful, knowledge about:

- Web languages: HTML, CSS, Javascript, jQuery
- Web servers: Apache, MySQL
- Server-side languages: PHP and/or Python
- Version control systems: Git and GitHub
- Virtualization

The class is open to juniors and seniors who have had at least one course in programming or by permission of the instructor.

Participants are expected to form teams of 3 members that will work independently and consistently throughout the semester to bring a mobile app of their own conception to the deployment phase.

Also, since mobile app development is still a very young discipline that is changing rapidly, participants will be expected to do research into the current trends in the industry and to share that research in the form of mini-lectures, tutorials, or presentations for the benefit of other members of the course in the context of frequent demonstrations of the concrete progress that has been made on their apps, to date.

Suggested Learning Objectives

Participants are expected to come into the course with a strong desire to build and deploy a mobile application. In the realization of that goal, it is expected that participants will develop competence in many of the areas below:

- Articulate an idea for a mobile app in the form of a project proposal with an appropriate scope and justification for building the app grounded in the social context of the target users
- Demonstrate understanding of the trade-offs involved with mobile app development including single- versus multi-platform solutions, standalone apps versus apps that depend upon (RESTful) web services, local versus remote storage, and privacy
- Develop collaborative skills including the use of a well-established version control system, e.g. GitHub, to manage project code, issues, and milestones
- Establish an effective development environment including a local web server, code editor, command-line tools, optionally bundled into a virtual machine sharable with team members
- Create design artifacts including user scenarios, architecture diagrams, screen mockups, and database schema
- Employ test-driven development (TDD) practices in the coding of a mobile app
- Develop competence in the use of command-line tools relevant to mobile app development, such as npm and cordova
- Develop RESTful web services using an established web development platform such as Laravel or Django
- Demonstrate the ability to leverage the capabilities of mobile devices, such as Wi-Fi, geolocation, local storage, cameras, accelerometers, and Bluetooth
- Understand the different types of revenue streams associated with mobile apps, and the ramifications of deploying apps to online stores such as the Apple App Store or Google Play
- Develop competence at oral communication of technological topics through frequent status reports

Participants are encouraged to add to this list in accordance with their personal goals.

Schedule

Class will be held weekly on Tuesday and Thursday from 5-6:15 in HHS 3022. Tuesday classes will generally consist of instructor-led tutorials, walkthroughs, and/or mini-lectures and discussions. Thursday classes will be student-led and consist of status reports, presentations, and/or time for team-based development work. In addition to class time, teams are expected to meet weekly with either or both instructors to address issues specific to each of the team projects.

Rough Schedule of Weekly Topics (Subject to Change)

Week	Topic	Instructors (Tue)	Teams (Thu)	Phase
1	Intro to Mobile Apps	Idea, Scenarios, Screens	Project Idea Gen.	0
2	Dev Environments	Dev Environment	Idea & Scenarios	1
3	Version/Issue Control	GitHub	Dev Environment	1
4	Architecture	Architecture (incl. DB)	Version/Issue Control	1
5	Server-side Elements	Test-driven Development	Architecture	1
6	API Development	REST API (server-side)	Tests	1
7	Client-side Interface	jQuery Mobile	Project Status	2
8	Storage	Storage	Project Status	2
Spring Break				
9	App Stores	Catch-Up	Project Status	2
10	Deployment	Deployment	Project Status	2
11	Devices/Sensors	Devices/Sensors	Project Status	2
12	Devices/Sensors	Devices/Sensors	Project Status	2
13	Devices/Sensors	Devices/Sensors	Project Status	2
14	Devices/Sensors	Devices/Sensors	Dry Run #1 Pres/Rept	
15	Wrap-Up	Wrap-Up	Dry Run #2 Pres/Rept	
Final		Final Presentation	Final Pres. & Report	

Instructor-led class meetings (Tuesdays) will consist of the introduction of core concepts coupled with a demonstration of those concepts in practice via a walkthrough. The instructors will present an application that they are developing in parallel with the students in the course. Student-led class meetings (Thursdays) will consist of presentations by the project teams of the current status of their projects. All teams are expected to present weekly, as well as participating in feedback/criticism of the other teams' presentations. In general, project team milestones will occur one week after the instructors have modeled a presentation on the same topic. This schedule is subject to change as necessary.

As class time may not be sufficient to cover all weekly topics in depth, participants are expected to cover elements not covered in class independently. Weekly team status reports are to be presented and uploaded to Canvas. They should reflect the outside preparation that teams have devoted to understanding and gaining hands-on experience in the topics for the week. In other words, although the instructors will attempt to cover as much of the proposed material as possible each week, it should be understood that the scope of this field is greater than what can be covered in a 75 minute class, and as such, participants will need to be diligent about filling the gaps in their understanding on their own outside of class. While instructors will point out useful resources, participants are encouraged to consult with the instructors if they have questions, and/or to let their classmates' know about any particularly useful resources they have discovered.

Evaluation & Grades

The course instructors will determine the semester grade for the course in consultation with participants. The base grade for the course will be determined by the degree of completion of the team project and the quality of the final presentation. Only projects that have been successfully deployed will be eligible for an “A”. Instructors reserve the right to adjust the semester grade downward from the base grade (determined by how close to completion the project is) to account for failure to contribute to the class and team effort.

Grade breakdown

- Project Status Phase 1 (5%)
- Project Status Phase 2 (15%)
- Dry Run 1 & 2 (Presentation, Demo, Report) (20%)
- Final (Presentation, Demo, Report) (60%)

Participants are required to keep a written log of their activities from week to week. This should be kept in a publicly available location that is shared among the team, such as a shared Google Docs folder or using the issue tracker on GitHub. If participants are unhappy with their final grade, their primary recourse will be the weekly logs. Log entries should consist of the following

- Description of the task
- Specific name(s) of who is responsible
- When it is due
- Current status of the task (done, in progress, late, deleted)
- Date of completion

Instructor Information

	Dr. Emil H. Salib	Dr. Morgan C. Benton
Office Location	ISAT/CS 312	ISAT/CS 124
Phone	540-568-6876	973-495-7736
Email	salibeh@jmu.edu	bentonmc@jmu.edu
Office Hours	TBD	ISAT/CS 337 8-noon M-F

Texts and Resources

There will be no textbook for this course, but the following resources should come in very handy:

- **Vmware** (<http://www.vmware.com/>)
It is recommended that one of the teams' first priorities is to create an image of a virtual machine that contains all of the tools and settings necessary for successful collaboration among the team, the class, and the instructors.
- **Ubuntu** (<http://www.ubuntu.com/desktop>) (12.04 LTS)
Ubuntu is a Linux distribution recommended for the development environment.

You will also want to make sure that you set up a LAMP server environment. There are many tutorials for this (e.g. <http://www.unixmen.com/install-lamp-server-apache-mysql-php-ubuntu-13-10-server/>). I recommend using MariaDB instead of MySQL, although either would be fine. NOTE: there may be better tutorials than this one. It was just the first one I found.

- **Sublime Text** (<http://www.sublimetext.com/>)
Sublime Text is a text editor. It can be used for free (although it's cheap and supporting the developers is highly recommended). You don't have to use this editor if you are already more comfortable with another one, but it is nice, lightweight, but with a lot of useful features and customizations.
- **GitHub** (<http://github.com>)
GitHub is a website designed to support collaboration. All teams are **strongly** encouraged to use GitHub to coordinate their activities.
- **Node.js and npm** (<http://nodejs.org>)
You may or may not need node.js, but npm (node.js package manager) will be necessary to install the cordova command line tools.
- **Cordova** (<https://npmjs.org/package/cordova>) aka Phonegap
Cordova is a set of tools that allows you to build multi-platform mobile apps using HTML, CSS, and Javascript. It is the platform the instructors will be using in their walkthrough.
- **Kivy** (<http://kivy.org/#home>)
Kivy is an open source python library for rapid development of applications that make use of innovative user interfaces, such as multi-touch apps.
- **Titanium Appcelerator** (<http://www.appcelerator.com/>)
Titanium includes both an IDE (integrated development environment) and a full framework for building multi-platform mobile apps. It is an alternative to cordova.
- **jQuery Mobile** (<http://jquerymobile.com/>)
jQuery and jQuery mobile are Javascript frameworks that will greatly increase the speed and ease with which
- **Laravel** (<http://www.laravel.com>)
Laravel is a web application development framework based on PHP. It is useful for rapid API development for mobile apps that have any sort of networked component.
- **Django** (<http://djangoproject.com>)
Similar to Laravel, Django is a web application framework based on Python. It is also useful for rapid API development, especially using Tastypie (<http://django-tastypie.readthedocs.org/en/latest/>). This is offered as an alternative to Laravel in case you are more comfortable with Python than PHP.
- **Pusher** (<http://pusher.com/>)
Pusher is a handy tool for adding pub/sub (aka realtime notifications, aka Comet) functionality to your mobile app.

As you can tell from this long list of resources, there may be a sharp learning curve for many people this semester. It is highly recommended that you take a divide-and-conquer approach within your team.