Comp 322/422 - Software Development for Wireless and Mobile Devices

Fall Semester 2017 - Week 1

Dr Nick Hayward

Course Details

Lecturer

- Name: Dr Nick Hayward
- Office: Doyle 307 (LSC)
- Office hours
  - *Friday afternoon by appointment (LSC)*
- Faculty Page

Course Schedule

Important dates for this semester

- DEV week: 9th to 16th October 2017
  - *n.b.* *no formal class: 13th October 2017*
  - *presentation & demo: 20th October 2017 @ 2.45pm*

- Thanksgiving break: 22nd to 25th November 2017
  - *n.b.* *no formal class: 24th November 2017*

- Final class: 8th December 2017
  - *presentation & demo: 8th December 2017 @ 2.45pm*

- Exam week: 11th December to 16th December 2017
  - *Final assessment due on 15th December 2017 by 2.45pm*

Initial Course Plan - Part 1

Up to ~ DEV Week

- begin development of a mobile application
  - *group project up to and including DEV week*

- Cross-platform options
  - *Apache Cordova & Onsen UI*
  - *React Native*

- mobile design considerations

- prototypes and tests

- data store options and usage

- API usage and integration

- lots of examples...

Initial Course Plan - Part 2

Up to the end of the semester

- continue to develop your app concept and prototypes
  - *group project augmented and improved from DEV week*

- continue React Native
  - *more examples and advanced options*

- comparison of mobile development options
  - *hybrid vs React Native vs native SDK*

- choose an option
  - *Swift and iOS*
  - *C# and Xamarin*

- many options for mobile development...

Assignments and Coursework

Course will include

- weekly bibliography and reading (where applicable)
- weekly notes, examples, extras...

Coursework will include

- quizzes or group exercises at the end of each section (Total = 30%)
  - *based on course notes, reading, and examples*

- mid-semester assessment (Total = 30%)
  - *DEV week: 9th to 16th October 2017*
  - *demo due 20th October 2017 @ 2.45pm*

- end of semester assessment (Total = 40%)
  - *demo due 8th December 2017 @ 2.45pm*
  - *report due 15th December 2017 @ 2.45pm*

Quizzes, group exercises...

Course total = 30%

- at least one week notice before quiz
  - *average time ~30 minutes (can be extended...)*
  - *taken towards the end of class*

- group exercises
  - *help develop course project*
  - *test course knowledge at each stage*
  - *get feedback on project work*

Development and Project Assessment

Course total = 70% (Parts 1 and 2 combined)

Initial overview

- combination project work
  - *part 1 = mid-semester **DEV Week** work (30%)*
  - *part 2 = final demo and report (40%)*

- group project (max. 4 persons per group)
- development of a mobile application
  - ***NOT** a responsive website viewed on a mobile device*
  - *Apache Cordova, React Native...*

- purpose, scope, and target audience is group's choice
  - ***no** to-do lists, note-taking, flashlights &c.*
  - *chosen project topic needs approval*
  - *data store, APIs, underlying structure &c. is group's choice...*

DEV Week Assessment

Course total = 30%

- cross-platform mobile app from scratch
  - *can be basic demo of intended final app*

- build using either
  - *Apache Cordova and UI (jQuery Mobile, OnsenUI &c.)*
  - *React Native*

- can be platform agnostic (cross-platform) or specific targeted OS, e.g.
  - *cross-platform app that builds for Android and iOS*
  - *targeted build for Android or iOS*
  - *consider choice, and explain why?*

- outline concept, research conducted to date

- consider applicable design patterns

- are you using any sensors &c. ?
  - *how, why?*

- prototyping
  - *demo current prototypes*
  - *any working tests or models &c.*

- anything else to help explain your mobile app...

DEV Week Demo

DEV week assessment will include the following:

- brief presentation or demonstration of current project work
  - *due on Friday 20th October 2017*
  - *~ 10 minutes per group*

- presentation and demonstration...
  - *outline mobile app*
  - *show prototypes and designs*
  - *explain what does & does not work*
  - *...*

Final Assessment

Course total = 40%

- continue to develop your app concept and prototypes
  - *develop application using any of the technologies taught during the course*
  - *again, combine technologies to best fit your mobile app*

- if the app uses Apache Cordova
  - *implement a custom Cordova plugin for a native mobile OS*
  - *e.g. Android or iOS*

- produce a working app
  - *as far as possible try to create a fully working app*
  - *explain any parts of the app not working...*

- explain choice of technologies for mobile app development
  - *e.g. data stores, APIs, modules, &c.*

- explain design decisions
  - *outline what you chose and why?*
  - *what else did you consider, and then omit? (again, why?)*

- which concepts could you abstract for easy porting to other platform/OS?

- describe patterns used in design of UI and interaction

Goals of the course

An overview and demonstration of building cross-platform applications for mobile and wireless devices.

Course will provide

- guide to developing and implementing mobile applications from scratch
- cross-platform design and development
  - *using Apache Cordova & UI options, React Native...*
- best practices and guidelines for cross-platform development
- outline of example mobile design patterns
- comparisons with native SDKs and development
- guide to deploying and publishing final mobile app
- ...

Course resources - part 1

Website

Course website is available at https://csteach422.github.io

- timetable

- course overview

- course blog

- weekly assignments & coursework

- bibliography

- links & resources

- notes & material

NO Sakai

Course resources - part 2

GitHub

Course repositories available at
https://github.com/csteach422

- weekly notes
- examples
- source code (where applicable)

Trello group

Group for weekly assignments, DEV week
posts, &c.

- Trello group - COMP 422
  - *https://trello.com/csteach422*

Slack group

Group for class communication, weekly
discussions, questions, &c.

- Slack group - COMP 422
  - *https://csteach422.slack.com/*

A few questions...

What is mobile?

- what exactly do we mean by **mobile**?

- may seem like a simple question to answer
  - *do we categorise mobile based on the OS*
  - *is it Android, iOS, Windows Phone...*

- where do we draw the line for software development?

- 2010 Wired magazine interview with Mark Zuckerberg
  - *iPad is not a mobile device, it is a computer*

Video - iPad not mobile

funnylog.kr - "iPad isn't mobile~ It's a computer" by MarkZuckerb...

Source - YouTube - iPad isn't mobile...

Merging technologies

- merging of technology and traditional environments and interactions
  - *definition of mobile will alter and update as well*

- will we perceive in-car devices as mobile?
  - *e.g. touchscreen panels and consoles*
  - *same as phones, tablets?*

- these differences are important
  - *they help us consider designs, UIs, interactions*
  - *different motivations for development*

- currently best to consider *mobile* relative to OS
 - *e.g. associated with phones and tablets*

Mobile considerations

- surge in popularity for mobile devices, apps
  - *associated interactions and usage patterns*

- concept of **mobile first** entered broader lexicon
  - *developers and designers think in terms of **mobile first***

- encouraged to think in terms of mobile use cases, scenarios...

- think beyond standard desktop app or website

Usage stats

- usage stats are also v.interesting for developers
- e.g. many users now use smartphones for less frivolous activities, including
  - *62% have used their smartphones to query information about their health or a medical condition*
  - *57% have used their smartphones to complete online banking*
  - *44% have used their smartphones to search real estate listings or other housing information*
  - *43% searched for job listings and availability*
  - *40% to view and check government listings and information*
  - *30% to take an online course or class*
  - *18% to actually submit a job application*

Image - Mobile's Share of Web Traffic
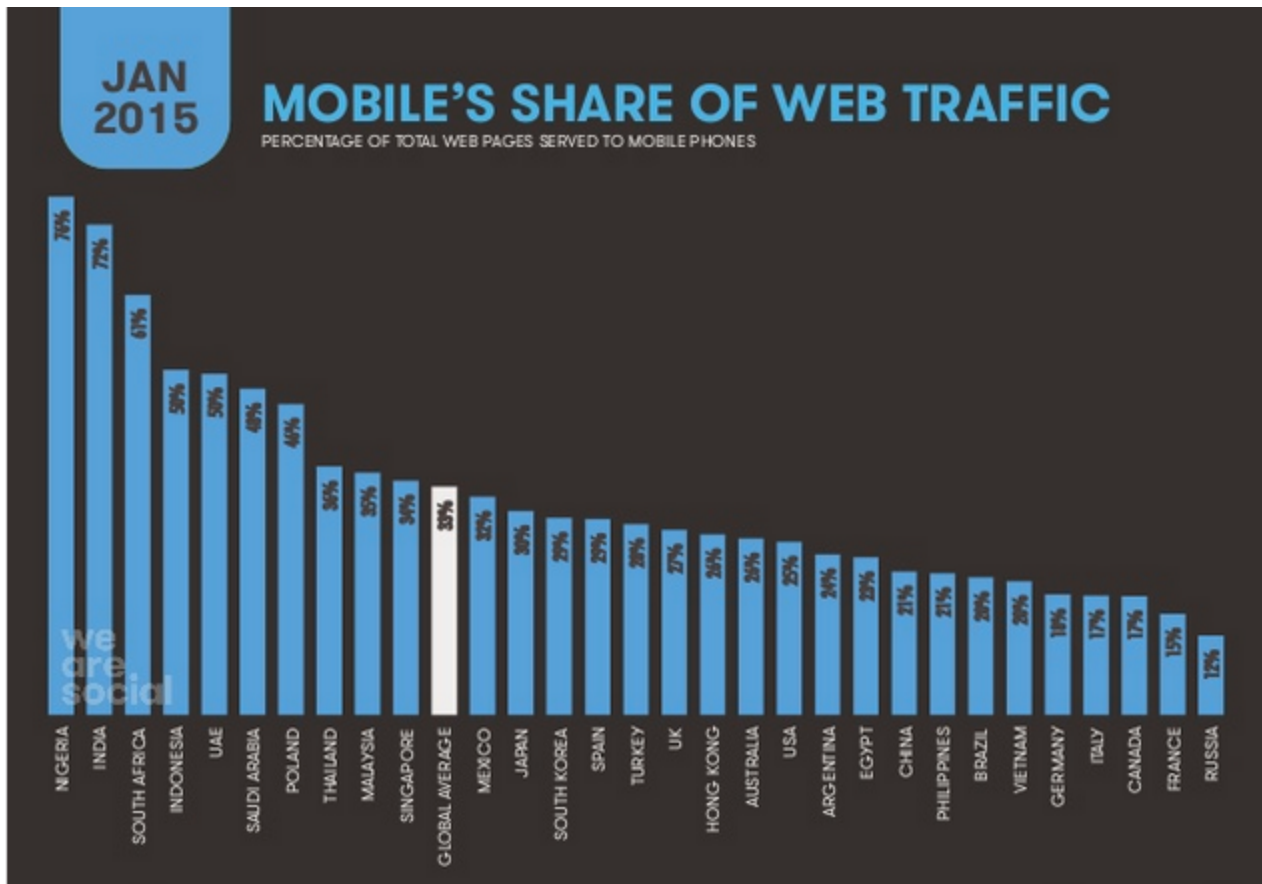
Source - We Are Social - Singapore

Video - Android One

Introducing Android One

Source - YouTube - Android One

Different types of mobile

- we need to be clear about the differences between mobile types
  - *mobile web*
  - *native mobile*
  - *hybrid mobile*

- each has its place in mobile development
- each has its own particular advantages and disadvantages

Mobile web

- apps viewed and run using a web browser
  - *usually, but not exclusively, a mobile device web browser*

- designed as *responsive* web apps or sites
  - *new generation of progressive apps becoming available*

- in this context *responsive* understood as adaptive views
  - *enables correct rendering on different resolutions of mobile and tablet devices*

- apps normally require user to be online with active data connection

- not true mobile apps
  - *may reflect same look and feel as native mobile OS app*

- apps not uploaded to mobile app stores

- unable to interact at the native, low-level of the mobile OS

Native mobile

- native mobile app development often perceived as *real deal*
  - *rightly or wrongly dependent upon your perspective*

- development of apps using SDKs and APIs for specific mobile OS
  - *Java for Android*
  - *Swift (& Objective C) for iOS*
  - *.Net for Windows 10 Universal Platform*

- learn and develop different SDK &c. for each native OS

- developer will need to implement code and logic for each platform
  - *both mobile OS implementation and desktop development*

- issue with modified app design and logic
  - *need to meet requirements and restrictions*
  - *limits imposed by each mobile OS...*

Hybrid mobile - Part 1

- hybrid mobile apps share a lot with native mobile apps
  - *e.g. characteristics, design traits, functionality*

- however, they are developed using different tools, technologies, methods...

- Apache Cordova apps developed using common web technologies
  - *HTML5 (HyperText Markup Language)*
  - *CSS (Cascading Style Sheet)*
  - *JS (JavaScript)*
    - many supported libraries & frameworks
    - many options for UI design and development

Hybrid mobile - Part 2

- attempt to leverage ease and speed of development
  - *due to web technologies*
  - *larger developer base for web development*

- and power of native functionality and hardware
  - *using plugins*

- benefit compared to native mobile
  - *option to use same code base for single app*
  - *same code across multiple mobile OSs*

- inherent benefit and grace of web stack for mobile app development
  - *ability to code once, run across multiple mobile platforms*

- still need to make changes to port an app from platform to platform
  - *often minor and trivial changes*
  - *in particular when compared with native OS development*

- other benefit is use of same languages across multiple platforms
  - *until development of custom plugins...*

Considerations for mobile web

- many benefits to native app development

- obvious benefit is optimised nature of compiled code

- native apps will often be slightly faster than hybrid apps

- choice of development route will depend upon many factors
  - *time*
  - *cost*
  - *development expertise and experience*
  - *chosen platform(s)*
  - *scale of application*

- often a case of personal development preference

Summary of options

Here is a useful table summarising your options for mobile development.

Technology	App Store	Technologies	Cross-platform	Native support	Performance (best practices)
Mobile web	No	HTML, CSS, & JS	Yes	Partial at best	Very good (most of the time)
Hybrid	Yes	HTML, CSS, JS, & APIs	Yes (modifications)	Full (using plugins)	Very good to Excellent
Native	Yes	Native SDK & APIs	No (requires porting)	Full	Very good to Excellent (depends on developer)

Cross-platform - intro

- inexorable rise in popularity of mobile devices
  - *rise in number of mobile OSs*
  - *each competing for market space*
  - *in particular in the consumer space*

- each OS offers similar options and features

- many mobile OS options, including
  - *Android*
  - *iOS*
  - *Windows 10 Universal platform*
  - *LG webOS*
  - *BlackBerry 10*
  - *various OSs for wearables*
  - *...*

Cross-platform - issues and concerns

- mobile market largely dominated by big two
  - *Android and iOS*

- reduced field still introduces issues and concerns for developers

- each mobile OS implements their own
  - *SDK (software development kit)*
  - *API (application program/programming interface)*

- similarities exist but
  - *they use different programming languages*
  - *whilst achieving the same end goals*
  - *Java for Android & Swift (Objective-C) for iOS*

- each mobile OS has its own peculiarities
  - *differing design philosophies &c.*

Cross-platform - common issues and solutions

- common issues might include
  - *permissions*
  - *access to underlying services within an OS*
  - *e.g. SMS rights and logic for different mobile OSs*

- cross-platform alternatives allows us consider unified development environment
  - *access and harness native device*
  - *leverage native functionality, performance, features...*

- leverage common tools and technologies
  - *HTML5, CSS, JavaScript - Apache Cordova and React Native*
  - *C# and Xamarin*
  - *create easier cross-platform apps*

Image - Apache Cordova

Source - Apache Cordova

Apache Cordova - what can it do?

- designed to offer a simple, powerful set of API calls
  - *calls to JavaScript functions*
  - *functions map native OS code to plugins and code in Cordova*
  - *enables access to core functionality for a device*

- allows us to transfer, manipulate, control
  - *data and resources from the native OS and device*
  - *moves it to the web view in our Cordova app*

- allows us to provide same user experience as native app
  - *minus a few base caveats*

- cross-platform support

Apache Cordova - platform support

support includes following mobile OSs

- Android
- iOS
- Windows 10 Universal platform
- Ubuntu
- LG webOS
- BlackBerry 10
- ...

Apache Cordova - functionality and plugins

- allows us to create native mobile applications using a set of common web technologies
  - *including HTML5, CSS, and JavaScript*

- a set of JavaScript APIs
  - *provides access to natively built core plugins*

- currently offers many core APIs

- includes some of the following native functionality,
  - *access the device's microphone for recording &c.*
  - *capture photos using the device's camera*
  - *photo retrieval from the OSs gallery/photo album*
  - *retrieve device information*
    - locale
    - various sensors such as motion, location, connection information, compass...
  - *retrieve device data, contact information...*
  - *process files from/to storage*
  - *...*

Apache Cordova - documentation and APIs

official *Cordova* API documentation is currently available at the following URL,

- Apache Cordova API
- Apache Cordova GitHub
- Android API
- iOS API
- ... & many others

Apache Cordova - why choose it?

- potential to develop once, re-use with ease

- Cordova helps us solve some of the following mobile development issues,

1. different programming languages for different mobile OSs
   - *different programming philosophies, conventions, best practices, guidelines...*

2. unique problems inherent to each given mobile OS
   - *e.g. handling and routing SMS requests, data storage, privacy features...*

3. developing, testing, and maintaining applications across multiple mobile platforms

4. ...

Apache Cordova - overview of APIs

Platform APIs (cordova-plugin)	Android	iOS	Windows Universal Platform
Accelerometer (device-motion)	Yes	Yes	Yes
BatteryStatus (battery-status)	Yes	Yes	Yes
Camera (camera)	Yes	Yes	Yes
Capture (media-capture)	Yes	Yes	Yes
Compass (device-orientation)	Yes	Yes	Yes
Connection (network-information)	Yes	Yes	Yes
Contacts (contacts)	Yes	Yes	Yes
Device (device)	Yes	Yes	Yes
Events	Yes	Yes	Yes
File (file)	Yes	Yes	Yes
File Transfer (file-transfer)	Yes	Yes	Yes (no support for `onprogress` or `abort`)
Geolocation (geolocation)	Yes	Yes	Yes
Globalisation (globalization)	Yes	Yes	Yes
InAppBrowser (inappbrowser)	Yes	Yes	Yes
Media (media)	Yes	Yes	Yes
Notification (dialogs)	Yes	Yes	Yes
Splashscreen (splashscreen)	Yes	Yes	Yes

Platform APIs (cordova-plugin)	Android	iOS	Windows Universal Platform
Storage	Yes	Yes	Yes (localStorage & indexedDB)
Vibration (vibration)	Yes	Yes	Yes

Apache Cordova Documentation - Platform Support

Apache Cordova - API details

- many of these mobile native function APIs self explanatory

- a few examples,

- **capture**
  - *record various media directly from the native device*

- **connection**
  - *provides information about the device's wifi and cellular connections*

- **device**
  - *provides useful information on a device's hardware and software*
  - *native device model, the current platform and its version...*

- **events**
  - *particularly important, and useful, API*
  - *e.g.* `deviceready`, `backbutton`, `batterystatus`, *volume events...*

- **file** api to help process device files

- receive the device's location using GPS or network signals

- many more...

- Apache Cordova Documentation - Platform Support

References

- Carmody, Tim., *Fighting Words: Defining "Mobile" and "Computer"* Wired. 11.08.2010. http://www.wired.com/2010/11/fighting-words-defining-mobile-and-computer/

- Smith, Aaron., *U.S. Smartphone Use in 2015* PewResearchCenter. 04.01.2015.http://www.pewinternet.org/2015/04/01/us-smartphone-use-in-2015/

- Various., *Digital, Social & Mobile in 2015* We Are Social Singapore. 01.20.2015. http://www.slideshare.net/wearesocialsg/digital-social-mobile-in-2015