CS 4518 Mobile and Ubiquitous Computing

Lecture 15: Final Project Slides/Paper, Other Ubicomp Android APIs

Emmanuel Agu
The Rest of the Class
The Rest of this class

- **Part 1: Course and Android Introduction**
  - Introduce mobile computing, ubiquitous Computing, Android,
  - Basics of Android programming, UI, Android Lifecycle

- **Part 2: Mobile and ubicomp Android programming**
  - mobile Android components (location, Google Places, maps, geofencing)
  - Ubicomp Android components (camera, face detection, activity recognition, etc)

- **Part 3: Mobile Computing/Ubicomp Research**
  - Machine learning (classification) in ubicomp
  - Ubicomp research (smartphone sensing examples, human mood detection, etc) using machine learning
  - Mobile computing research (app usage studies, energy consumption, etc)
Final Project Submissions
Final Project Submissions

- Still need to:
  - Give final **10-minute presentation** on your final app
  - Submit **final paper** describing your app
- Fully described on the final project website:
- Today: Just give a bit more detail, answer questions
Final Talk & Final Paper: Same Content

- Introduction
- Related Work
- Approach/methodology
- Implementation
- **Project timeline**
- Evaluation/Results
- Discussion
- Conclusion
- Future Work

---

Note: No timeline in final paper
Final Talk & Final Paper

- **Final talk:** Mar 2 or 3 (7/8 groups each day)
- **Submit:** talk slides + final paper (Written 5 pages max Word + PDF file): due Mar 3, 11.59PM!!

- Details
  - **Introduction**
    - List team members
    - State problem app you solved + social benefit, target community
    - Why is problem important?
    - E.g. Find statistics: How much time, money, resources is being wasted on this problem today? How many people problem affects
    - Potential gain: how will your solution save time, money, etc?
  - **Related work**
    - What other research has been done to solve this problem (academic + commercial apps)
    - How is your app/approach different? And how is it similar?
Final Talk & Final Paper

- **Methodology/App Design:**
  - Summarize how your app works
  - Illustrate using final app screens/flow:
Separate Vision and Prototype

1. Big picture if funds/time not an issue (e.g. company of 200 employees over 6 years)

2. Which reasonable Subset of the big vision can you do in 2.5 weeks?

Can make simplifying assumptions
Final Talk & Final Paper

- Implementation details of your prototype:
  - Emphasize mobile/ubicomp components used
Final Talk & Final Paper: Evaluation

- Depends on what your project is.
- **Basic question:** How well did your solution work?
  - **App user study:** post-survey after using your app
    - Get users to use/rate your app, ask users about likes dislikes
    - Will they use your app if available?
  - **Stretch?** Measure performance. E.g. energy consumption, bandwidth consumption, etc
Recruiting Subjects For User Studies

- **3Fs:** Friends, Family and Fools
- Easy: Classmates (Do a trade with another group)
  - You guys evaluate our app, we’ll evaluate yours
- On campus: post flyers, set up table at campus center
  - Probably overkill
Final Talk & Final Paper: Discussion, Conclusion, Future Work

- **Discussion:**
  - How was your app generally received? Rationalize your findings in user studies,
  - What aspects did users generally like/dislike?
  - Why you think certain features work? not work? etc

- **Future work**
  - Talk about features that would extend prototype in feature
  - Revisit big vision.
Your Team
Some Team Tips

- Everyone (team members) doesn’t have to do everything equally
- Team members can work on project aspects they are good at
- Example: Who is good at:
  - Android UI design (Android Studio design view, XML file, widgets, nice look)
  - Android programming (database, sensors, maps, backend)
  - Experimental evaluation/user studies
  - Machine learning
  - Writing, making presentations
Some Team Tips

- Team should have an honest conversation
- Decide who is good at what aspects, do it!
- Consider team online management tools (trello.com, gantt charts, etc)
- Assign tasks, mini-deadlines (every few days)
- Integrate features every few days => new version
- **Mantra:** Always have a working prototype, improve
Project Resources
Resources

- I have set up a links page for mobile and ubicomp projects

http://web.cs.wpi.edu/~emmanuel/courses/ubicomp_projects_links.html
What other Android APIs may be useful for ubicomp?
Google Now

- Intelligent assistant, gives
  - Recommendations (travel time, traffic, etc)
  - Information (e.g. scores from favorite spots teams)
- Works by recognizing repeated user actions on device (common locations, repeated calendar appointments, search queries, etc)
- Displays info as Information “Cards”
- **In future:** Can retrieve, use information on Google Now cards
Information on Google Now Cards
https://en.wikipedia.org/wiki/Google_Now

- Activity summary
  (walking/cycling)
- Birthday
- Boarding pass
- Concerts
- Currency
- Developing story and
  breaking news
- Events
- Event reminders
- Flights
- Friends' birthdays
- Hotels
- Location reminders
- Movies
- Nearby attractions
- Nearby events
- Nearby photo spots
- New
  albums/books/video
  games/TV episodes
- News topic
- Next appointment
- Packages
- Parking location
- Places
- Product listing
- Public alerts
- Public transit
- Research topic
- Restaurant reservations
- Sports
- Stocks
- Time to home
- Time reminders
- Traffic and transit
- Translation
- Weather
- Website update
- What to watch
Speaking to Android

https://developers.google.com/voice-actions/

- **Speech recognition:**
  - Accept inputs as speech (instead of typing) e.g. dragon dictate app?
  - Note: Requires internet access

- **Speech-to-text**
  - Convert user’s speech to text. E.g. display voicemails in text

- **Voice Actions:** Voice commands to smartphone (e.g. search for, order pizza)
**Gestures**

https://developer.android.com/training/gestures/index.html

- **Gesture:** Hand-drawn shape on the screen
- **Example uses:**
  - Search your phone, contacts, etc by handwriting onto screen
  - Speed dial by handwriting first letters of contact’s name
  - Multi-touch, pinching
More MediaPlayer & RenderScript


- Media Player can also **record** audio and video
  - Manipulate raw audio from microphone/audio hardware, PCM buffers
    - E.g. if you want to do audio signal processing, speaker recognition, etc
    - **Example:** process user’s speech, detect emotion, nervousness?

- **RenderScript**
  - High level language for GPGPU
  - Use Phone’s GPU for computational tasks
  - Very few lines of code = run GPU code
  - Useful for heavy duty tasks. E.g. image, video processing
Wireless Communication


- **Bluetooth**
  - Discover nearby bluetooth devices
  - Communicating over bluetooth

- **WiFi**
  - Scan for WiFi hotspots
  - Monitor WiFi connectivity, Signal Strength (RSSI)
  - Do peer-to-peer (mobile device to mobile device) data transfers
Wireless Communication

- **NFC:**
  - Contactless technology
  - Transfer small amounts of data over short distances
  - **Applications:** Share spotify playlists, Google wallet
  - **Google wallet?**
    - Store debit, credit card on phone
    - Pay by tapping terminal
Telephony and SMS


- **Telephony:**
  - Initiate phone calls from within app
  - Access dialer, etc

- **SMS:**
  - Send/Receive SMS/MMS from app
  - Handle incoming SMS/MMS in app
Other 3rd Party Stuff

http://web.cs.wpi.edu/~emmanuel/courses/ubicomp_projects_links.html

- **MPAndroid**: Add charts to your app

- **Trepn**: Profile energy usage of your app
Other 3rd Party Stuff
http://web.cs.wpi.edu/~emmanuel/courses/ubicomp_projects_links.html

- **Programmable Web APIs:** 3rd party web content (e.g. RESTful APIs) you can pull into your app with few lines of code
  - **Weather:** Weather channel, yahoo weather
  - **Shared interests:** Pinterest
  - **Events:** Evently, Eventful, Events.com
  - **Photos:** flickr, Tumblr
  - **Videos:** Youtube
  - **Traffic info:** Mapquest traffic, Yahoo traffic

- **E.g. National Geographic:** picture of the day