

Handout for reference: [https://docs.google.com/document/d/1pvi9kZDjsAFRXITPNf\\_j\\_knwYQfk5Q-yV7p-ggMuGig/pub](https://docs.google.com/document/d/1pvi9kZDjsAFRXITPNf_j_knwYQfk5Q-yV7p-ggMuGig/pub)

## Final Demo

### Demo (5min)

- Introduction (show 1min intro video)
  - Mention target audience - use bob as an example, taking kids to the EX, santa claus parade, couple going to see tiff, etc.
  - Explain how we solve this problem - easily find parking and its availability, no wasting time
- How to demo
  - Markers display vacancy, colour gives quick visual cue
  - I can search somewhere to change location (dundas square)
  - Change the filter to modify results
  - Click on marker to get detailed information
  - Mention how markers (that are displayed) are being updated every 15sec
  - Open the list view to get a detailed breakdown
  - Get directions for a marker
  - Show off heroku app where we can manage lots

### Technical Discussion (3min)

- We need to discuss 1-3 technical items (interesting hurdles we faced) and provide screen shots, code snippets, etc.
- Item 1 - MVC design (model view controller)
  - How we used it and why - Our solution is very modular (backend, hardware, front end application) **we can work on this point**
  - Cons and pros - seemed tedious at times but android naturally follows a MVC design pattern **we can work on this point too**
  - What we learned
  - Async Issues
- Item 2 - Interfacing our backend with hardware
  - Had to come up with a reasonable solution for tracking cars in parking lots - we observed that some parking lots already automate this
  - Realised our software must handle different implementations
  - Simple API that responds to POST requests - use the raspberry pi to demonstrate this (doesn't matter what the actual hardware is)
- Item 3 - UI improvements
  - We went over many iterations for the UI
  - We knew it wasn't amazing but it was hard to get right
  - Had a wake up call when the UI wasn't appreciated by the TA and he critiqued it fairly heavily

- We had to conduct our own usability testing and get outside critique (between ourselves, friends and TA) to get it right
- Similar to real world

#### Process Discussion (3min)

- We used slack to communicate and github issues for task distribution
- People's tasks were usually assigned modularly, so people can work on their own files and merging was simple
- Had weekly meetings and would add meetings as needed for phases
- Significant decisions we made (good or bad)
  - Peer review sessions (good, allowed people to make sure code was working and functioned with other people's work)
  - Assigning more than one person to a task when there is bottleneck (good, used for listview and controller)
  - Tasks were due day before phase, no follow up in between (bad, changed after first phase. We ran into the issue that some tasks were more complicated or time consuming than others)

#### Individual contribution (1min per person)

- You will talk about your main contributions - make sure to remember what you worked on and lessons you learned (technical or group related)