ECE 1778: Creative Applications for Mobile Devices

Lecture 4

January 29, 2020

Today

1. Logistics – Plan, Assignments
   - Description of Assignment S3
   - Proposal
2. Block Diagrams + Living in Apps
3. Notes on Group Work
   - Working in Groups
4. A Note on the Android Life cycle
5. External Sensors/Gadgets you may want to use
6. TeamChooser
   - Lessons from an App development experience
7. Informal Topic Discussions/Coordination

Logistics

Assignments

- S2 was due yesterday
  - Any comments/questions?

- S3 is posted on course website
- P2 is due next Tuesday

- S1 and P1 have been graded
  - Apologies to iOS; we will have our own device soon.

Assignment S3

Due Tuesday February 4

All assignments are on both Course website and on Quercus

Assignment S3 – for Specialists

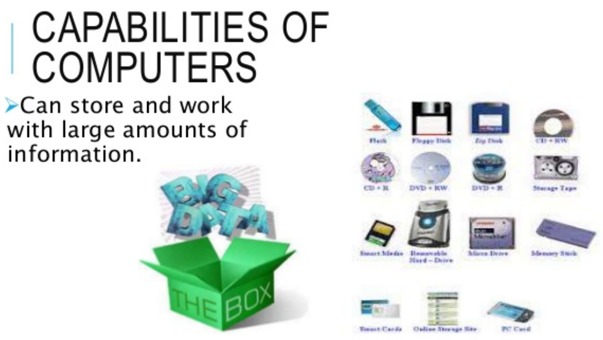
- Assignment S3 is new this year.
- The goal is to stimulate ideas for different features of the App in your project, and to aid discussion within team

- First:

1. Ask you to reprise your field description from S1.
2. Give best current version of your **project** description at the same level as the Approval-in-Principle

Part 3 – Consider what is possible

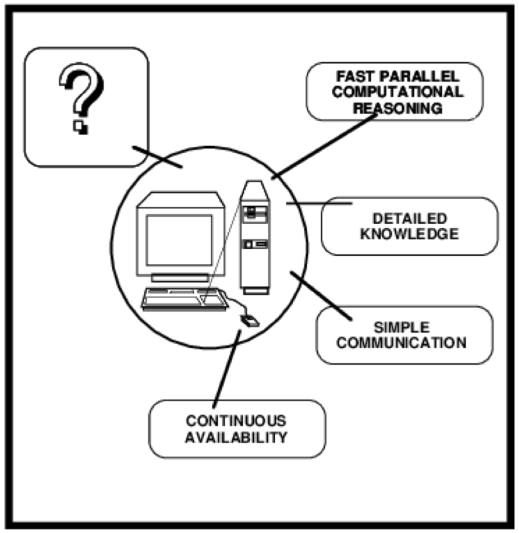
- Consider all the kinds of capabilities that exist in the phone, the internet and computers in general
  - I did this in Lectures 1, 2 and 3
  - e.g. Phone sensors
  - Give a short description of various computer abilities
  - Draw from your own knowledge of what you've seen apps/computers can do



CAPABILITIES OF COMPUTERS

➢Can store and work with large amounts of information.

BIG DATA
THE BOX
Flash
Floppy Disk
Zip Disk
CD+RW
CD+R
DVD+RW
DVD+R
Storage Tape
Smart Media
Removable Hard-Drive
Micro Drive
Memory Stick
Smart Cards
Online Storage Site
PC Card



Part 3 – Describe/Invent Features!

- Describe <u>separable</u> **Features** that you want in your app
  - **without** worrying about the resource, time or complexity
  - (want to free your mind from the constraints to begin)

- But what is a feature?
  - And what does separable mean?

Payments
Geo-Services
Loyalty
Good Burger
Social
Ordering

What is a Feature?

- It is a separate piece of functionality in software
  - it might be essential to the basic operation
  - It might be a ‘nice-to-have’ but not essential part.

- Example:
  - Essential: in app that gives list of ski-related products, the **list** itself is essential
  - Having the ability to sort the list based on price is a feature
    - Or sort on other aspects of the product

- How to do this?   “Live In” your app
  - Concrete example later

Parts 3 and 4

3. Rank features in order of importance and describe why they are high or low.

4. Determine with programming partners what order to build
   - Based on your joint understanding of what is necessary and how difficult features are
   - This list can be bigger than what you do in the actual project.

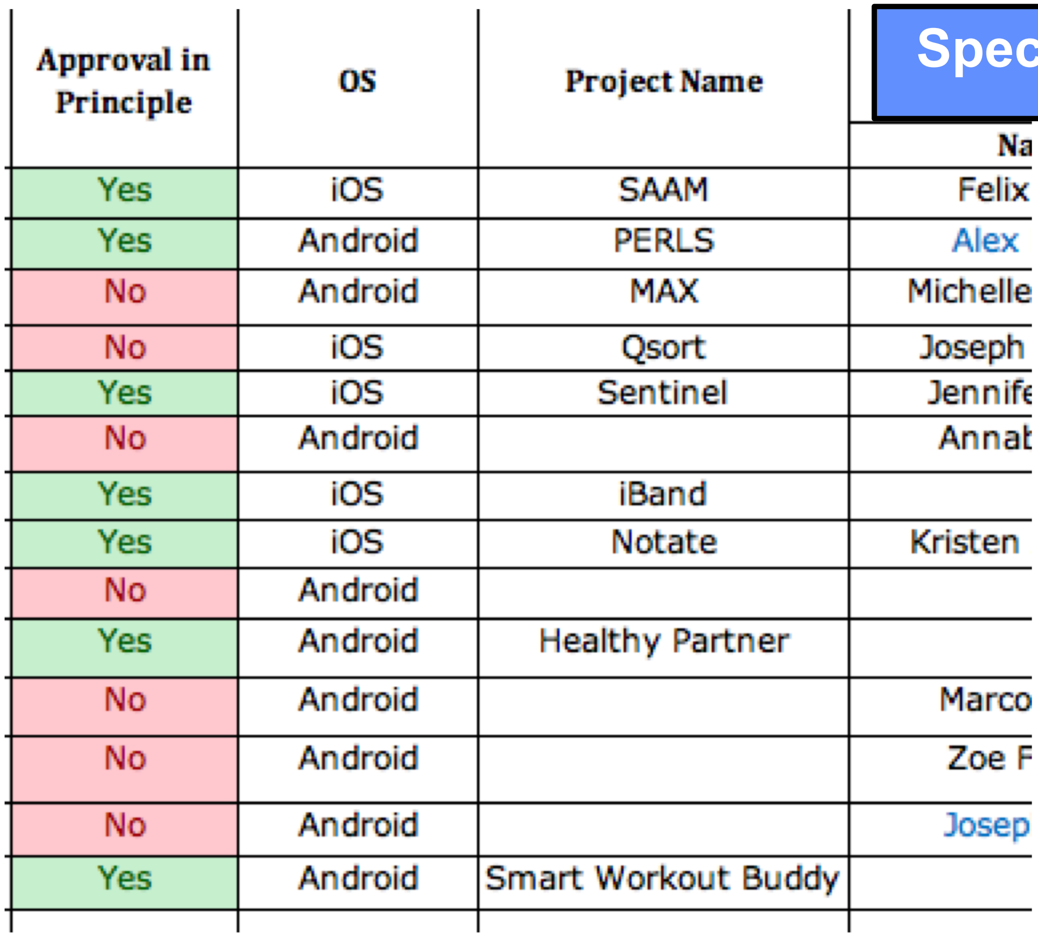
- Purpose – to direct your creativity, and to start the process of figuring out what is practical

Project

Project Stages

1. **Forming Groups**
   - Must be formed now
2. **Project Approval-in-Principle**
   - Should be wrapping up
3. **Project Proposal/Plan**
   - Document Due February 10th
4. **Proposal & Plan Presentations**
   - February 12th – see lecture 3 for details
   - **NOTE EXTRA LECTURE Wed February 12th, 6-8pm, RS 208**
5. **Spiral 2 & Spiral 4 Presentations**
   - 2: March 4/11 4: March 18/25
6. **Final Presentations**
   - Weeks of April 1/8
7. **Final Report Due April 15th**

Approval-in-Principle Somewhat Behind



Specialist

Approval in Principle	OS	Project Name	Na
Yes	iOS	SAAM	Felix
Yes	Android	PERLS	Alex
No	Android	MAX	Michelle
No	iOS	Qsort	Joseph
Yes	iOS	Sentinel	Jennife
No	Android		Annat
Yes	iOS	iBand	
Yes	iOS	Notate	Kristen
No	Android		
Yes	Android	Healthy Partner	
No	Android		Marco
No	Android		Zoe F
No	Android		Josep
Yes	Android	Smart Workout Buddy	

Proposal/Plan Due Monday Feb 10 @ 6pm

1. Describe Goal, make more precise
   - What & Why
2. Rough design of what the **user** of the App sees
   - Screen designs
   - Can use Marvel App - https://marvelapp.com
     - From Specialist Assignment 2
     - Any drawing package will do
3. **Block Diagram** overview of planned code
   - The large pieces (roughly 5) of the system
     - With short description of each (below picture)
   - Should be linked to the screens
   - I will discuss creation of block diagrams shortly

Plan, continued

4. Statement of Risks/Issues
   - What roadblocks/issues/challenges do you foresee?
   - Software, Hardware, Ethics, Data …
5. What do you need to learn that you don't know
   - all members
6. **Important:** Specialists
   - Submit a 500 word essay on
     1. How App relates to field of Specialist, and
     2. How the Specialist will contribute to project
        - Need to be an active participant; want clear thought here
          - **Issue Tracking** on Github?

- **Document must have these sections; will lose marks if missing**

Proposal/Plan Document

- length: 1500 words max
  - not including Specialist essay (#6)
  - include word count as part of document, penalty for overage
- Seeking clarity, not quantity of words
  - Omit needles words
- **One member of group should submit for group**
  - Under “Proposal Plan” for your group, under its name
- Worth 10% of grade
  - including in-class presentation done following week
- Document Due Monday February 10th at 6pm

A Note on Block Diagrams

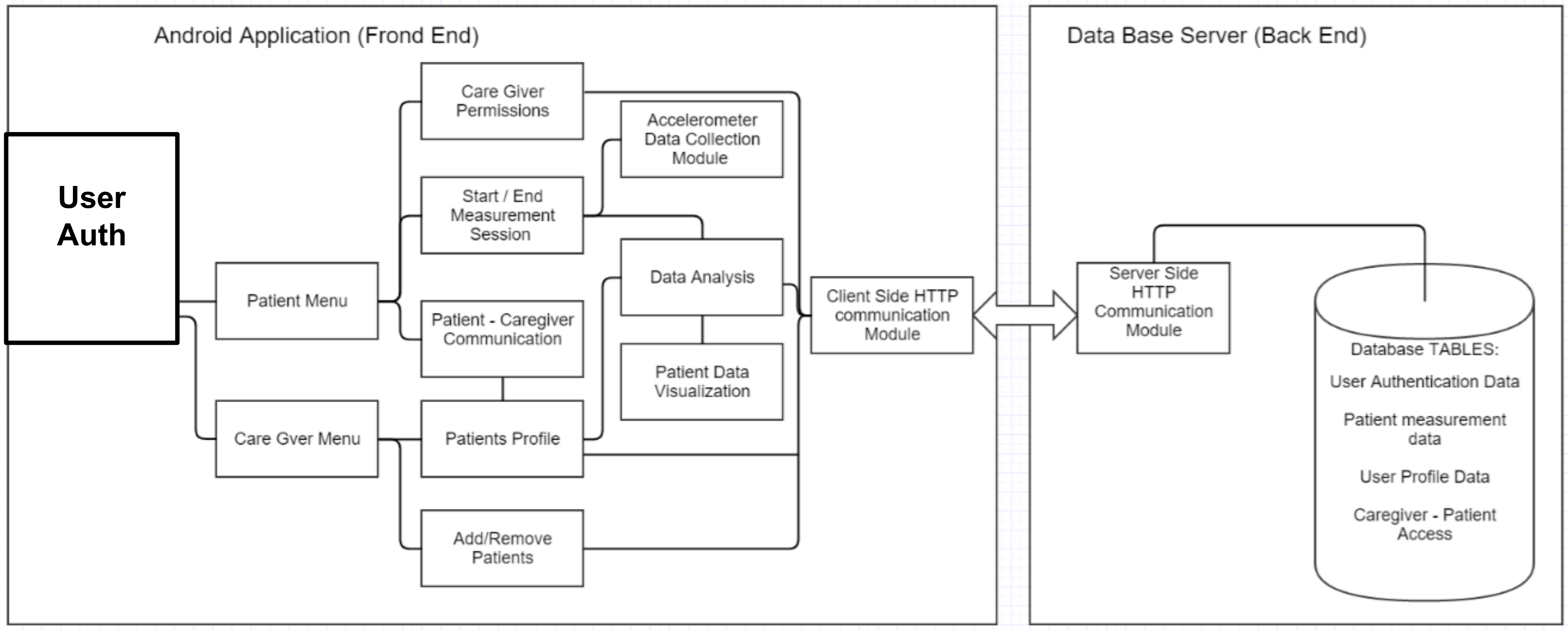
And the creative process

Block Diagrams

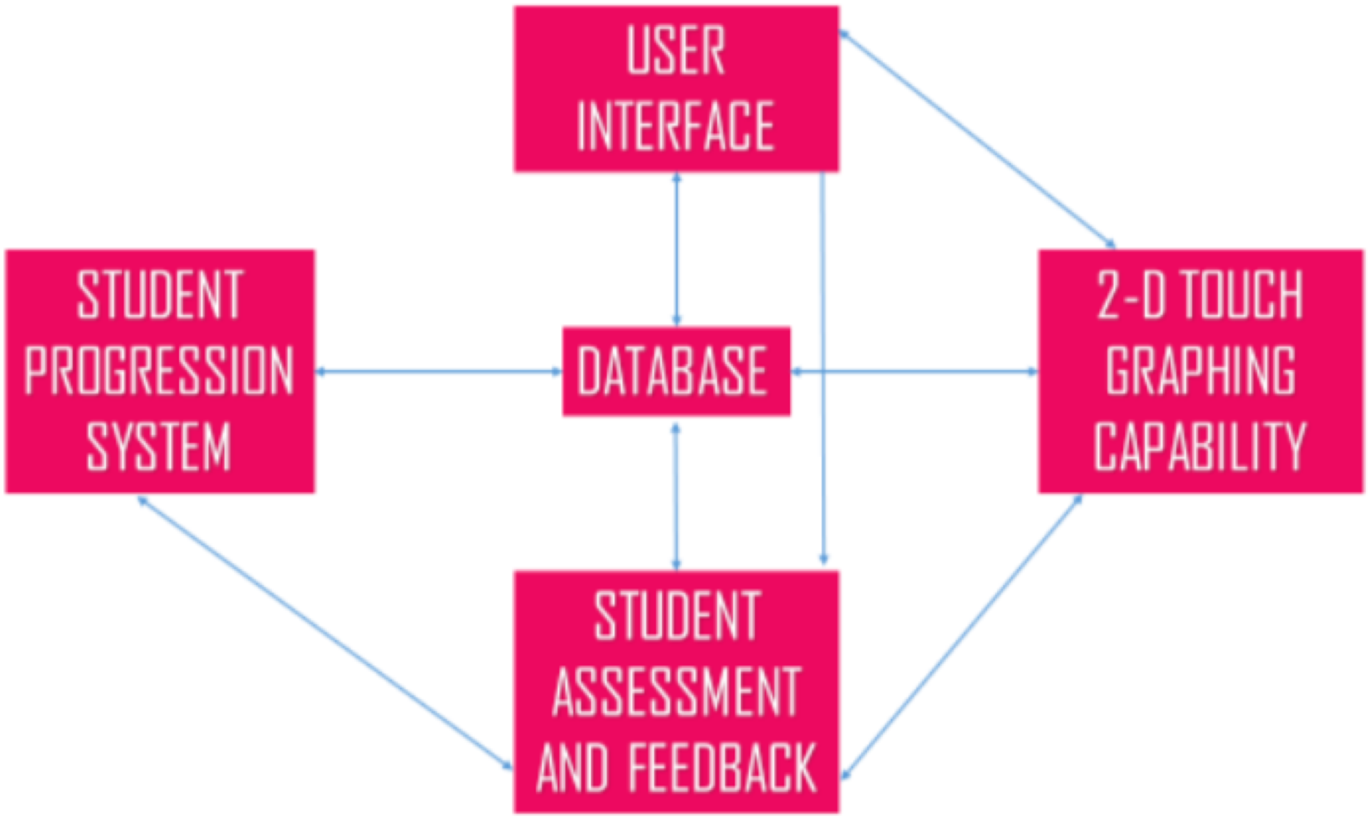
- In Proposal you're asked to give a block diagram of your planned software
  - In past, many have not understood this
- A Block diagram describes the major pieces of the project's functions (roughly 5)
- Is the first step in the standard **divide & conquer** approach
- Draw blocks, give each block a name
  - Name gives a sense of what it does
  - Provide several sentences that give more detail of function of each block

Block Diagrams

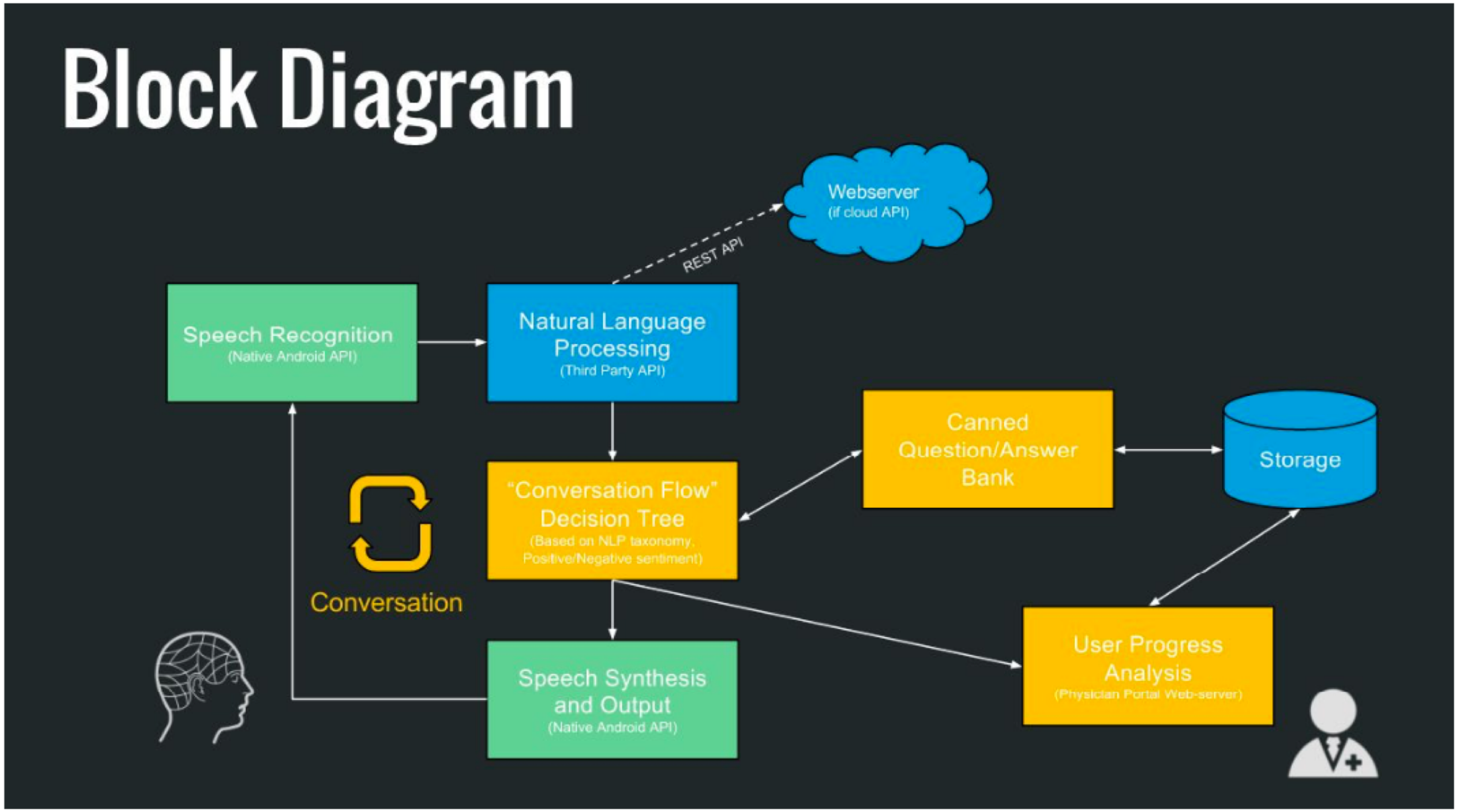
- Lines between blocks show communication
- May need to break blocks themselves down
  - In a hierarchy, in a subsequent part of your plan document
  - This example has **too many** blocks, too small font


Android Application (Frond End)
Data Base Server (Back End)
User Auth
Patient Menu
Care Gver Menu
Care Giver Permissions
Start / End Measurement Session
Patient - Caregiver Communication
Patients Profile
Add/Remove Patients
Accelerometer Data Collection Module
Data Analysis
Patient Data Visualization
Client Side HTTP communication Module
Server Side HTTP Communication Module
Database TABLES:
User Authentication Data
Patient measurement data
User Profile Data
Caregiver - Patient Access

Better, but maybe too simple


USER INTERFACE
STUDENT PROGRESSION SYSTEM
DATABASE
2-D TOUCH GRAPHING CAPABILITY
STUDENT ASSESSMENT AND FEEDBACK

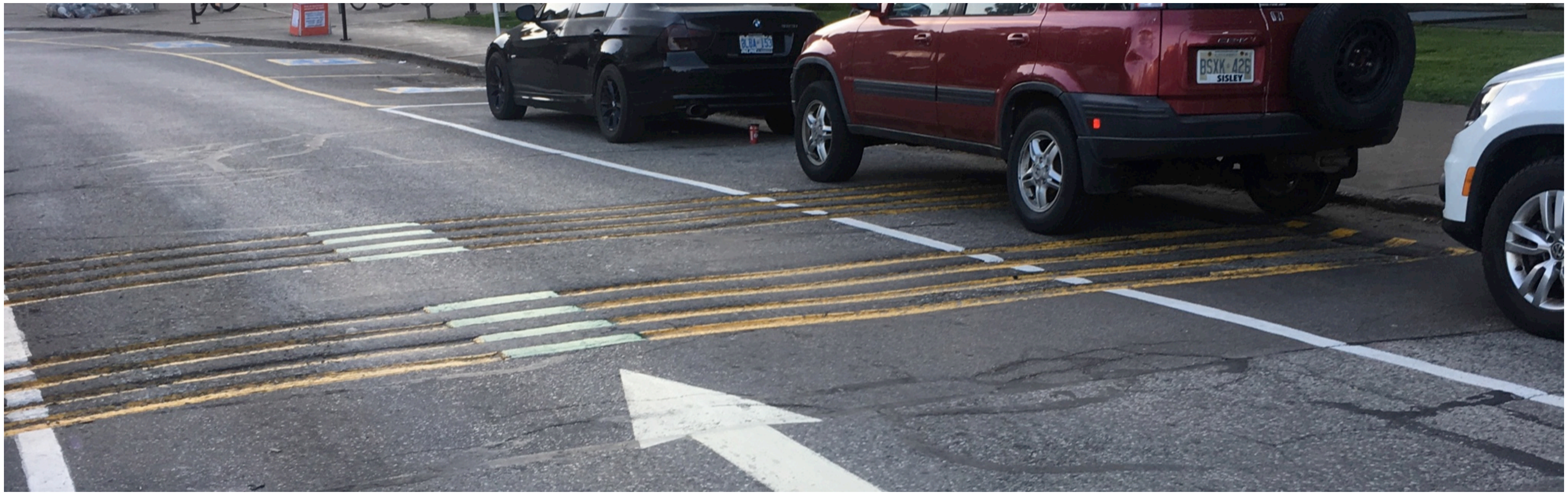
Good



Block Diagram
Webserver
(if cloud API)
REST API
Speech Recognition
(Native Android API)
Natural Language Processing
(Third Party API)
Canned Question/Answer Bank
Storage
"Conversation Flow" Decision Tree
(Based on NLP taxonomy, Positive/Negative sentiment)
Conversation
User Progress Analysis
(Physician Portal Web-server)
Speech Synthesis and Output
(Native Android API)

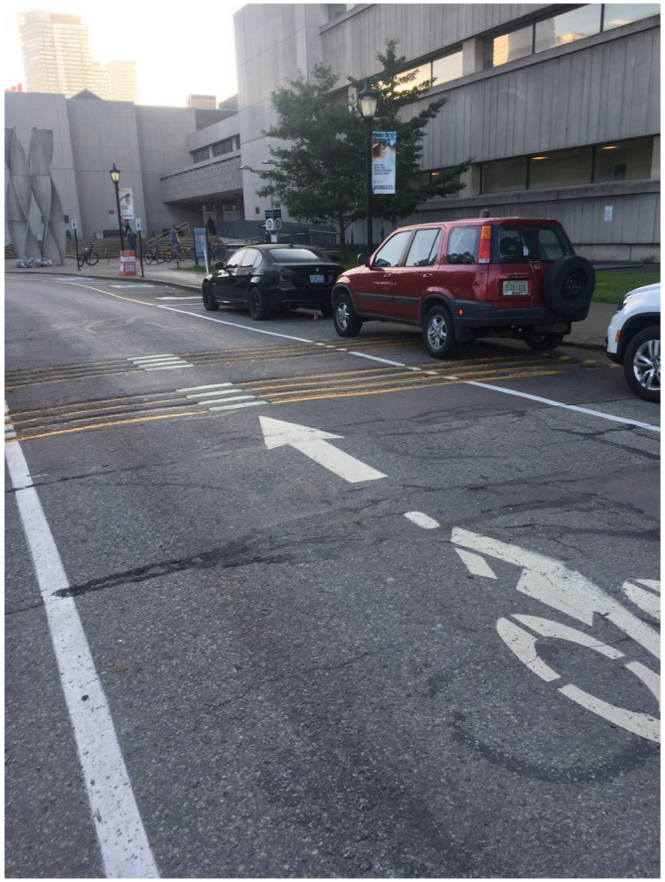
Let’s Work Through An Example

- Problem: Terrible Speed Bumps on King’s College Circle
  - very poorly designed – reverse bumps!
  - hitting the yellow part almost throws bike rider off bike!



So, Fill in Middle for Smooth Bike Cross

- This forces the cyclist into the middle of the road
- Into the path of any trailing car!



That’s Dumb, But Even Worse

- The speed bumps don’t really seem to affect cars at all!
- Watch:

App Goal:

- Measure Speed Bumps Effectiveness in Car
  - i.e. how noticeable is the bump?
  - Does the bump get worse with higher speed of a car?
- How could you do this with a phone?

Let’s Make the Block Diagram

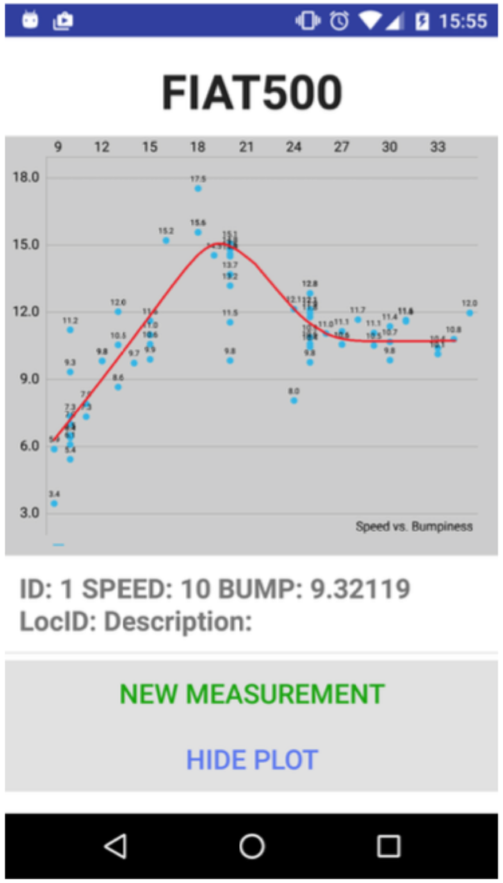
- What are the main components?

Let’s Figure out How to Enhance it

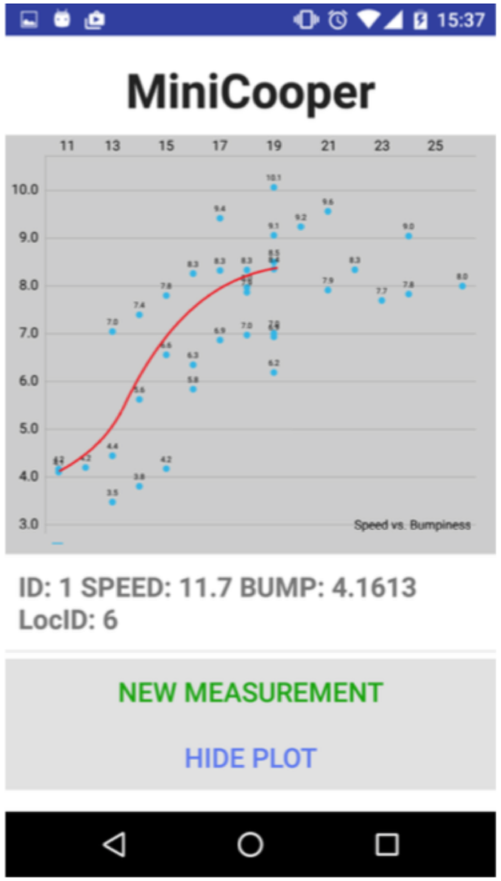
- Think about the context, what could we do to make this more useful/functional/helpful?
- That is – what other **features** might this app have?

Project: “How Bumpy” from 2015

- Measurement of Two Different Cars over bumps
  - Did it prove anything?



ID: 1 SPEED: 10 BUMP: 9.32119
LocID: Description:



ID: 1 SPEED: 11.7 BUMP: 4.1613
LocID: 6

A Note on Group Interaction

Group Interaction

- Now that groups are formed, it is very important for you to meet regularly and coordinate your activities
- Each of you will need to be assigned tasks, and make commitments to do those tasks at a particular deadline
- The expectations that each group member has of the others must be made clear
  - I suggest that these be written down
  - **Each** meeting in which tasks are decided:
  - Tasks should be recorded at end of each meeting
    - What
    - Who
    - When

“When” is a guess

- **Just** because you thought you could get something done, doesn’t mean you will be able to
- Tasks deadlines may have to ‘slip’ because of under-estimation
- is reality of engineering and indeed all work
  - A key opportunity to learn how to estimate work!

- Group must be forgiving of missed tasks
  - Must communicate and re-organize goals

Difficulties Can and Do Happen

- Most of the difficulties that have occurred in this course have been because expectations were not made clear
- Sometimes, group members consistently failed to live up to commitments.
- If this happens, please report it to me as soon as it is something that you cannot handle with internal discussion & resolution
  - We will help you resolve it
- Disagreements are part of all human relationships
  - One distinguishes oneself by how well you deal with them

What is a good way to communicate?

- Your favourite messaging app
  - Whatsapp, FB Messenger, WeChat….
- **Slack -** https://slack.com/intl/en-ca/
- Skype/Google Hangouts/Zoom
- Github Issue tracking

That’s About What Could Go Wrong

- What about doing things to make it go right?
  - See above about basic organization

- What is a good way to create a successful group?
  - Get to know each other – find out goals and interests
  - have some meals together – breakfast lunch or dinner.
  - Practice the notion of listening and giving everyone’s ideas a fair hearing.

GitHub Requirement

- We require you to use GitHub for storing software and tracking issues
  - Aside: what is source code control?
  - Demo issue tracker on github.com
- I will create the repository for you, and make **all** group members collaborators
  - Including specialists
  - Repository will be private
- Your immediate task, by this Friday:
  - Send me your **GitHub ID** so I can connect you to the repo
  - If you don’t have **ID**, go to **github.com** and sign up
    - Then send your GitHub ID to me with group name
    - I will link you to the repository

Class Participation & Peer Review

Class Presentations

- A key part of what happens in this course is the contribution you make to other's projects

- You will do many presentations in this class
  - Indeed, one side-effect of this project course is some real practice in giving high-quality, concise & clear communication
  - Most presentations will be 5 minutes in length
  - Must be geared so that most people in the class will understand

Peer Review

- Want everyone to come, listen & provide useful feedback
- Expectation that you'll listen and provide thoughtful feedback and suggestions to other's presentations

In Addition

- For each of Proposal, Spiral 2, and Spiral 4 you'll be asked to provide a written peer review for one other group; these will be graded
- Means you'll need to be here for every lecture, not just when you're presenting.

Peer Review for Proposal

Will be asked to answer questions such as:

1. In your own words, what was the goal of the project?
2. What parts of the proposal did you understand, and what parts could be more clear?
3. What was the best thing about the proposal?
4. What one thing could be improved the most?

External Hardware for your Project

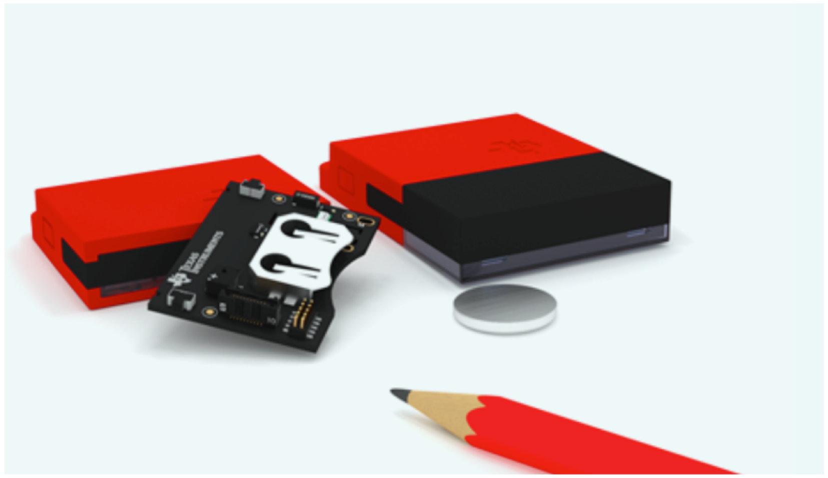
External Hardware/Gadgets

- If you know of some external hardware that will enable your project, you can request to have it purchased
  - I try to collect these for this course and research
  - No guarantee, depends on cost & function, but ask if you see something interesting!

- Some Examples ….

Texas Instruments Sensor Tag 2

- **Bluetooth Connection**
- **Sensors:**
  - 9 axis
  - Magnet sensor
  - Light
  - Ambient temperature
  - IR temperature
  - Humidity
  - Air pressure
  - Two Buttons, two lights, quiet buzzer!
- I can order these as needed
- Demo



Muse Head Band

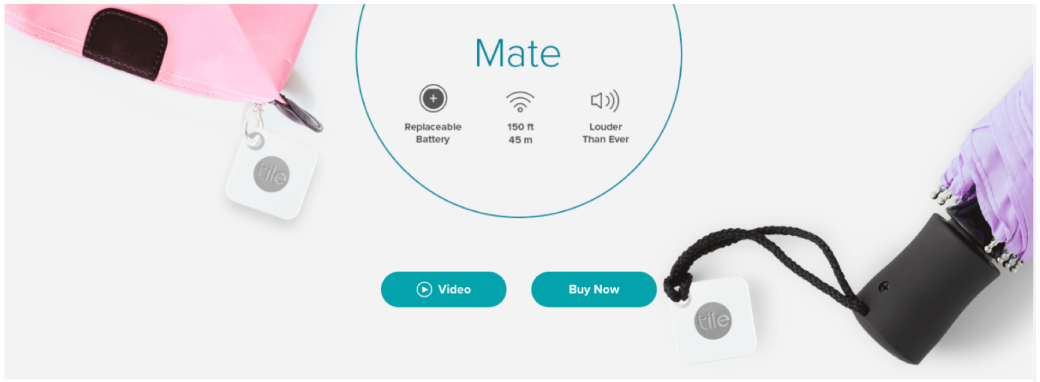
- Measures ‘brain activity’ through measurement of electro-magnetic waves brain produces
- Have 1 of these

TILE Tracker

- Note where things are, where they were lost!
  - https://www.thetileapp.com
  - https://youtu.be/WG7BdW7iFzo

  - Don’t have any, but could order
  - There is an unofficial API documented/coded here:

  - https://github.com/bachya/pytile



Other External Devices I have

- Wahoo Tickr (1) – heart rate monitor strap
  - bluetooth connected
  - Direct live heart rate

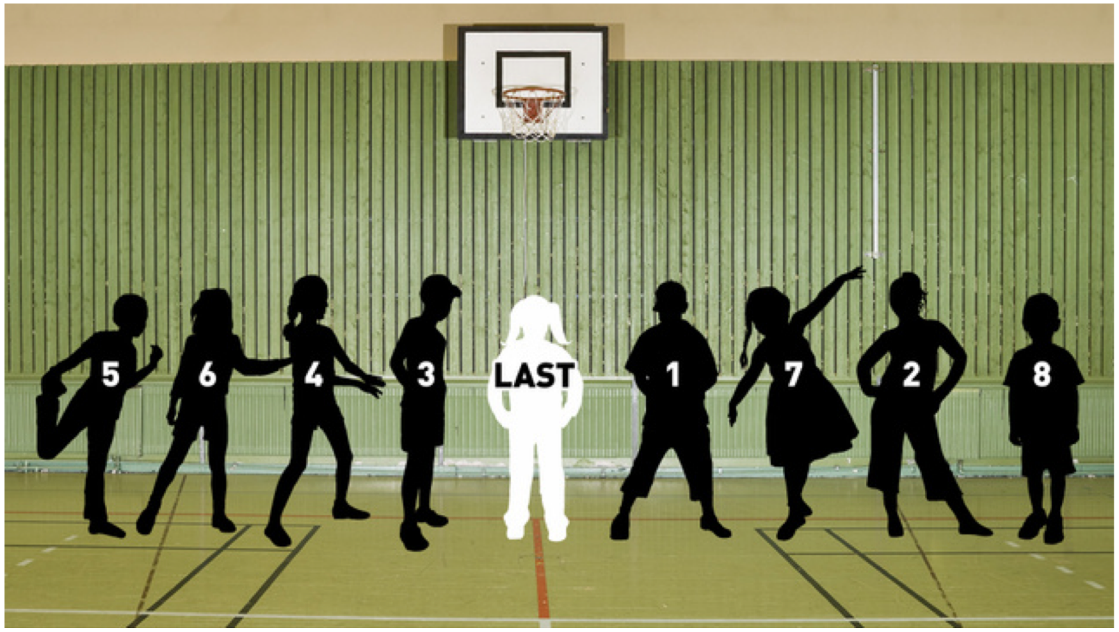
My App: TeamChooser

Solving a Problem in Pick-up Team Sports

Kids Who Play Want Games to be Fair!

So Do Adults!

- When playing friendly games there is a need to choose who is on which team
- A common method is to have team captains, and they alternate choosing people, in a very public way
- How many people have been picked **first?** ☺
- How many people have been picked **last?** ☹



I’ve Been Playing Hockey for Many Years

- A friendly game, but still have problem choosing teams:
- Classic Canadian method:
  - Put players’ sticks into middle
  - One person randomly throws sticks to either side!
  - Random outcome!
- I once chose teams for a few years in friendly game
  - People complained a lot!

Play in Two Different Friendly Games

- **Wednesday** Game: terrible chooser (**Agar**)
  - People always complaining
  - Games often lopsided, much distaste
  - no-one else took over, though (didn't want the hassle?)

- **Sunday** Game: excellent chooser (**Paul**)
  - Paul had a natural ability to pick great teams!
  - Even when teams didn't look right, many more times than not, the game was fair
  - Became known as the '**algorithm**'

The Solution: TeamChooser

- Wouldn't it be great if an App made the teams?
  - No one to yell at
  - Possibly give better teams

- Who needs this?

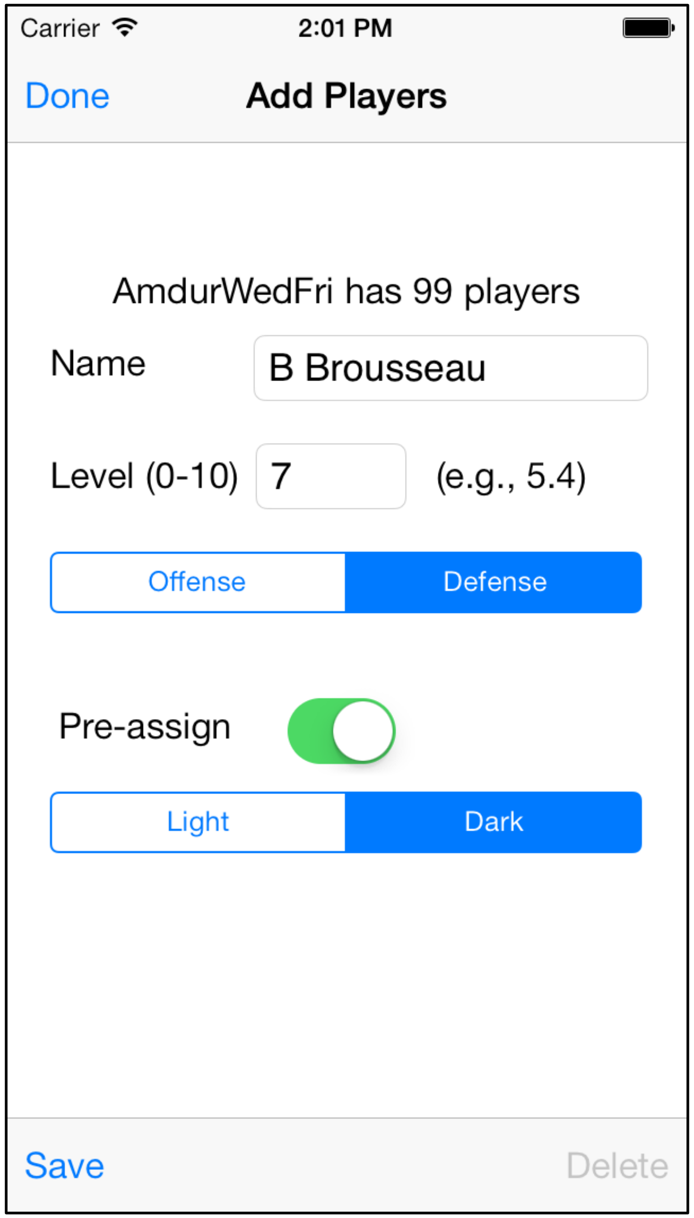
- Every pick-up hockey, soccer, basketball game around!

TeamChooser: How It Works

- Enter every player’s information in advance of game day
  - Player’s name
  - Preferred position (offence or defense)
  - **A rating, from 1-10, as to how effective player is**
    - Rating is the trickiest part

- On game day – select all players present
- Push ‘**Make Teams**’
  - And voila, two evenly matched teams

Do Demo! Entering Players


Carrier 2:01 PM

Done **Add Players**

AmdurWedFri has 99 players

Name: B Brousseau

Level (0-10): 7 (e.g., 5.4)

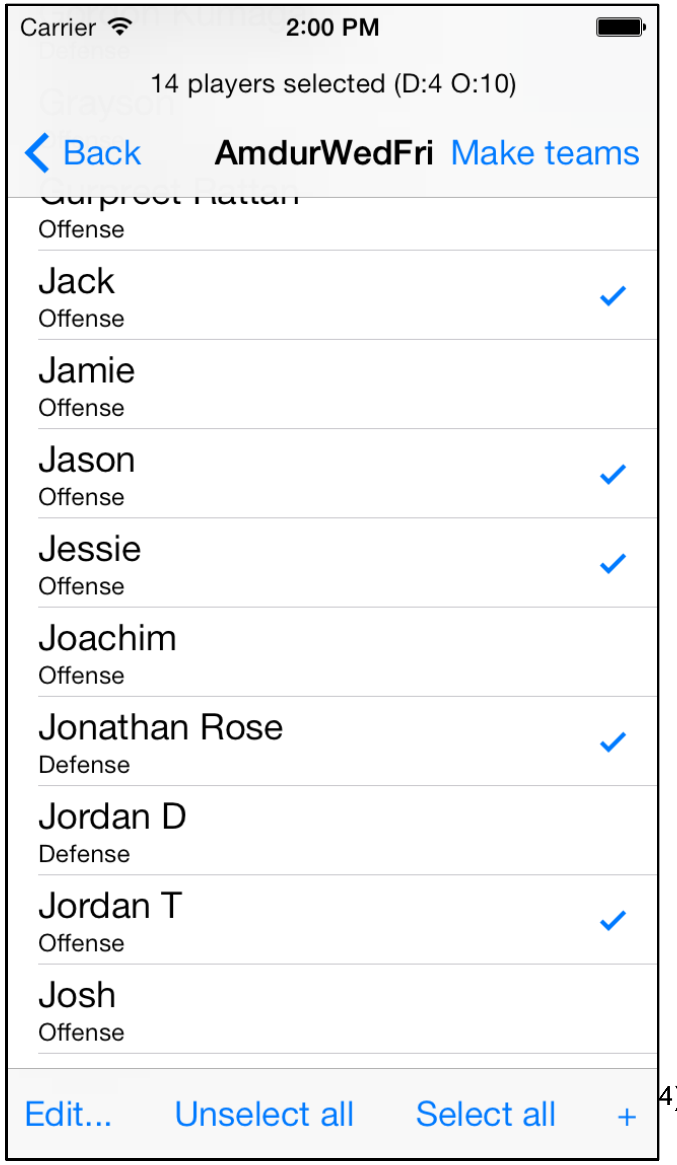
Offense [✓ Defense]

Pre-assign: on

Light [✓ Dark]

Save Delete

Selecting Present & Making Teams



Carrier 2:00 PM

14 players selected (D:4 O:10)

Back AmdurWedFri Make teams

Gurpreet Rattan
Offense

Jack ✓
Offense

Jamie
Offense

Jason ✓
Offense

Jessie ✓
Offense

Joachim
Offense

Jonathan Rose ✓
Defense

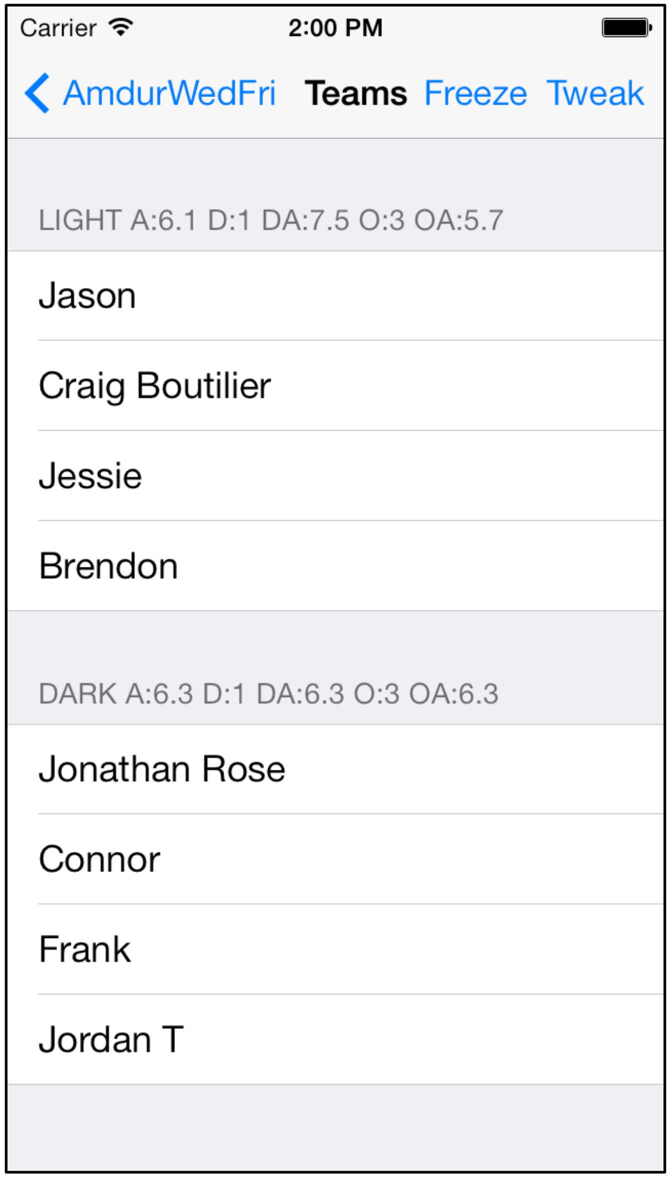
Jordan D
Defense

Jordan T ✓
Offense

Josh
Offense

Edit... Unselect all Select all +

4)



Carrier 2:00 PM

AmdurWedFri Teams Freeze Tweak

LIGHT A:6.1 D:1 DA:7.5 O:3 OA:5.7

- Jason
- Craig Boutilier
- Jessie
- Brendon

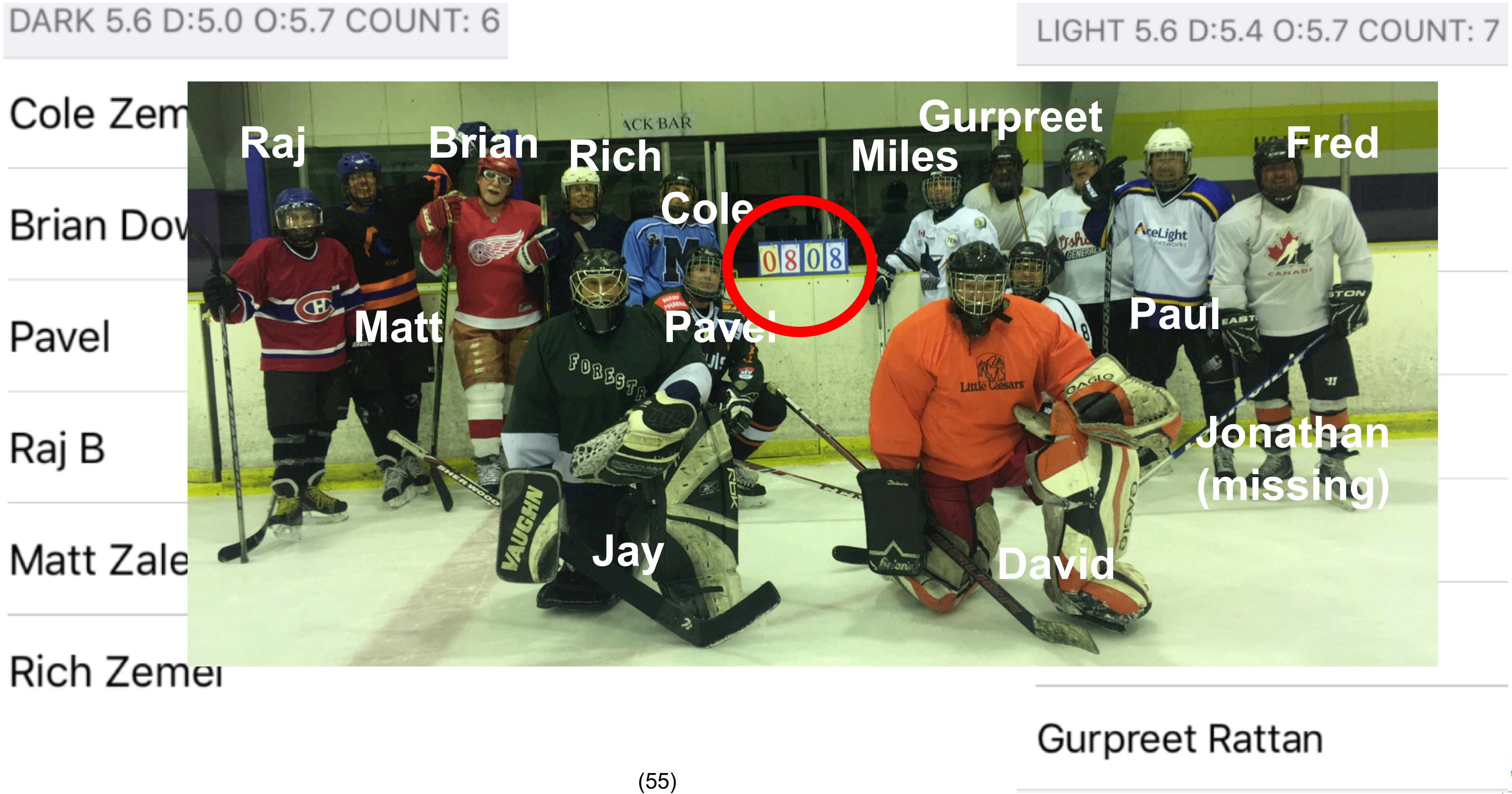
DARK A:6.3 D:1 DA:6.3 O:3 OA:6.3

- Jonathan Rose
- Connor
- Frank
- Jordan T

Example Game

Dark Team

Light Team


DARK 5.6 D:5.0 O:5.7 COUNT: 6

Cole Zem

Brian Dov

Pavel

Raj B

Matt Zale

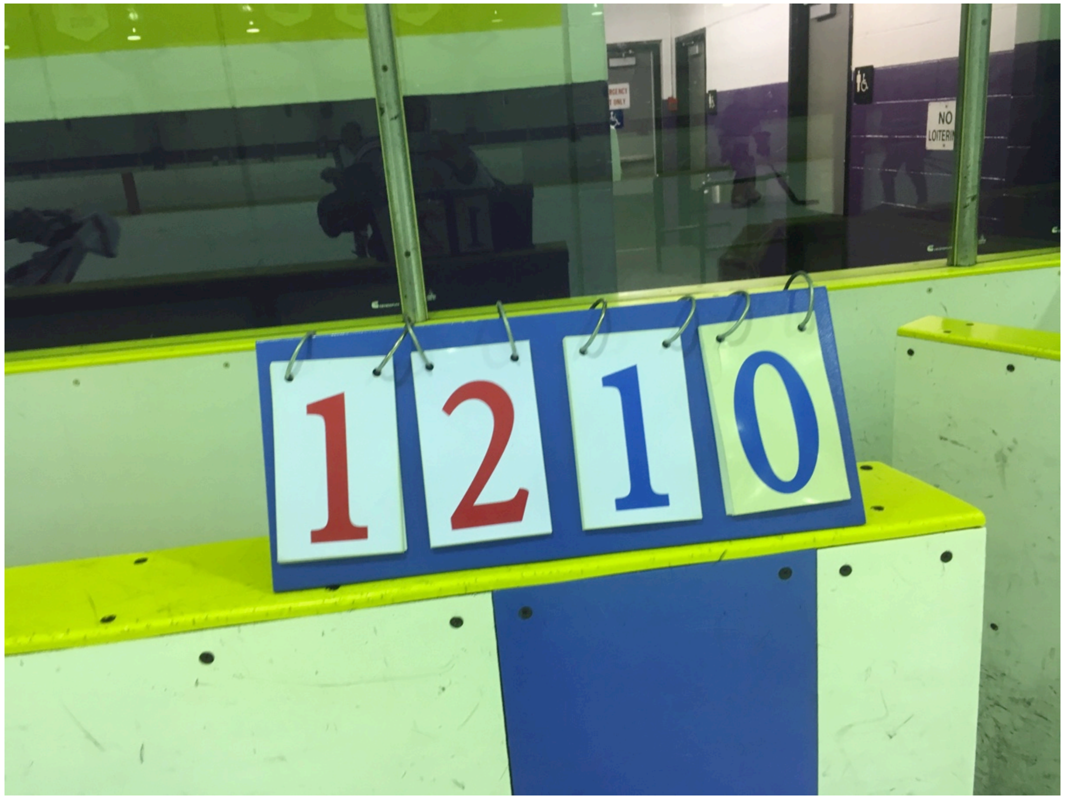
Rich Zemel

LIGHT 5.6 D:5.4 O:5.7 COUNT: 7

Gurpreet Rattan

Final Score in that Game

- A Victory for Team Chooser!



Team Selection Method ('Algorithm')

Step 1: Sort Players in Order of Rating, Highest to Lowest

Step 2: Alternate Team Assigned Going Down List

Team A		Team B
	6	
	5	
	4	
	3	
	2	
	1	

Team Selection Method

- Gets more complicated when add features!
  - Balance offense and defensive player count
  - Pre-assigns
  - Balance quality of offense and defense
- Many discussions from CS and ECE Professors over algorithms in hockey game!

Relating to Assignment S3

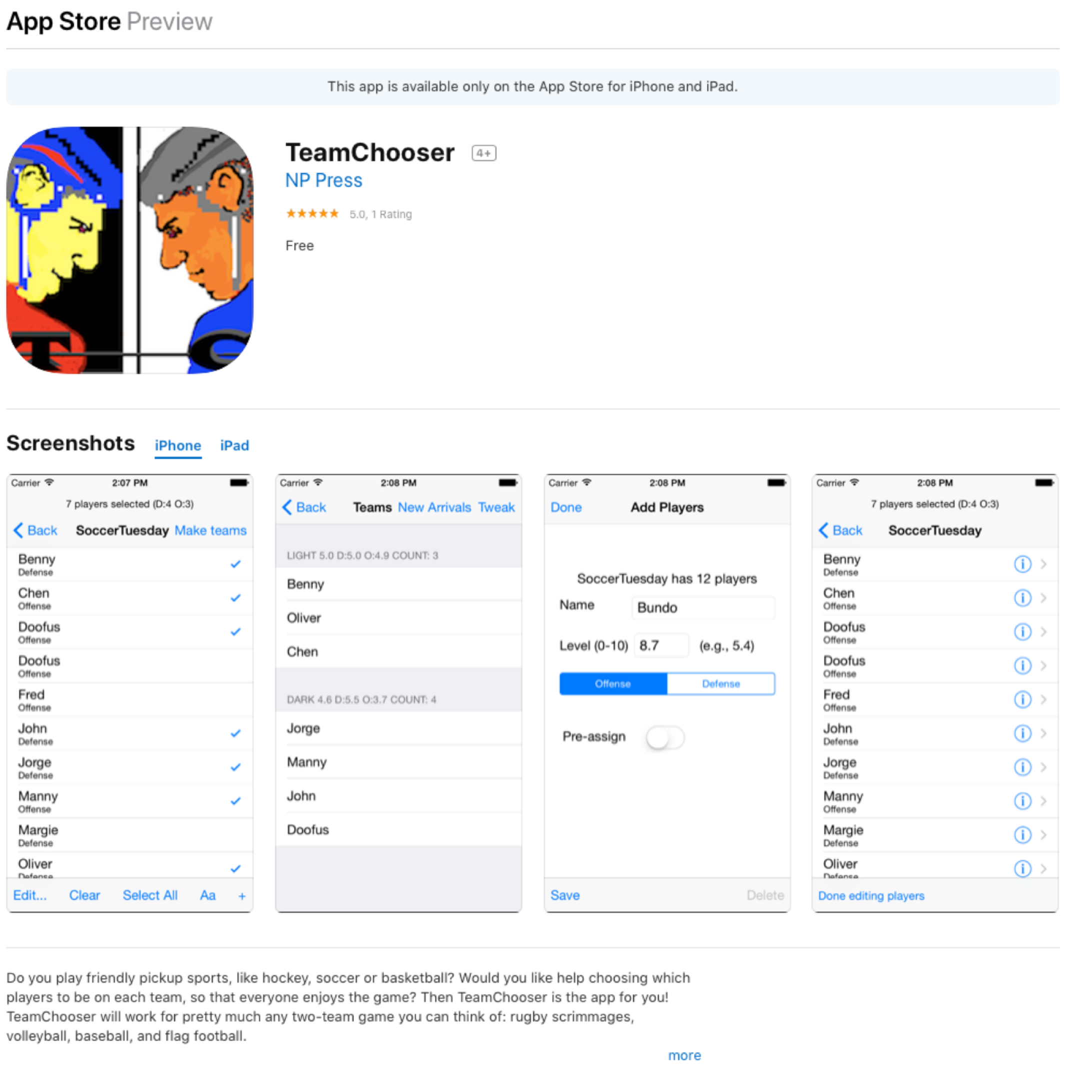
Features in App – Current and *Desired*

1. Input System – entering names and ranks
2. Choosing process and display
   - Very neat choosing + rating feature!
3. Randomized Order of Team Lists
   - So no-one is picked last
4. Large Font Button for far-sighted old folks!
5. Constraint: put selected people on fixed side
6. Constraint: equalize # of offence and defence-minded on each side
7. Late Arrival feature
8. *Infer people's scores based on result*
9. *Email/Share team files amongst multiple people*
10. *Print Out Chosen Teams for Posting*

Does it Work?

- Yes!
- I’ve been using it with friends in roughly 600 hockey games and it has often done a good job.
  - We’ve tweaked it’s algorithms here and there
  - Added some features
  - Occasionally very unbalanced games, bad luck?

On iPhone App Store Since May 2010



App Store Preview

This app is available only on the App Store for iPhone and iPad.

TeamChooser 4+
NP Press
★★★★★ 5.0, 1 Rating
Free

Screenshots iPhone iPad

Carrier 2:07 PM
7 players selected (D:4 O:3)
Back SoccerTuesday Make teams
Benny Defense ✓
Chen Offense ✓
Doofus Offense ✓
Doofus Offense
Fred Offense
John Defense ✓
Jorge Defense ✓
Manny Offense ✓
Margie Defense
Oliver Defense ✓
Edit... Clear Select All Aa +

Carrier 2:08 PM
Back Teams New Arrivals Tweak
LIGHT 5.0 D:5.0 O:4.9 COUNT: 3
Benny
Oliver
Chen
DARK 4.6 D:5.5 O:3.7 COUNT: 4
Jorge
Manny
John
Doofus

Carrier 2:08 PM
Done Add Players
SoccerTuesday has 12 players
Name Bundo
Level (0-10) 8.7 (e.g., 5.4)
Offense Defense
Pre-assign
Save Delete

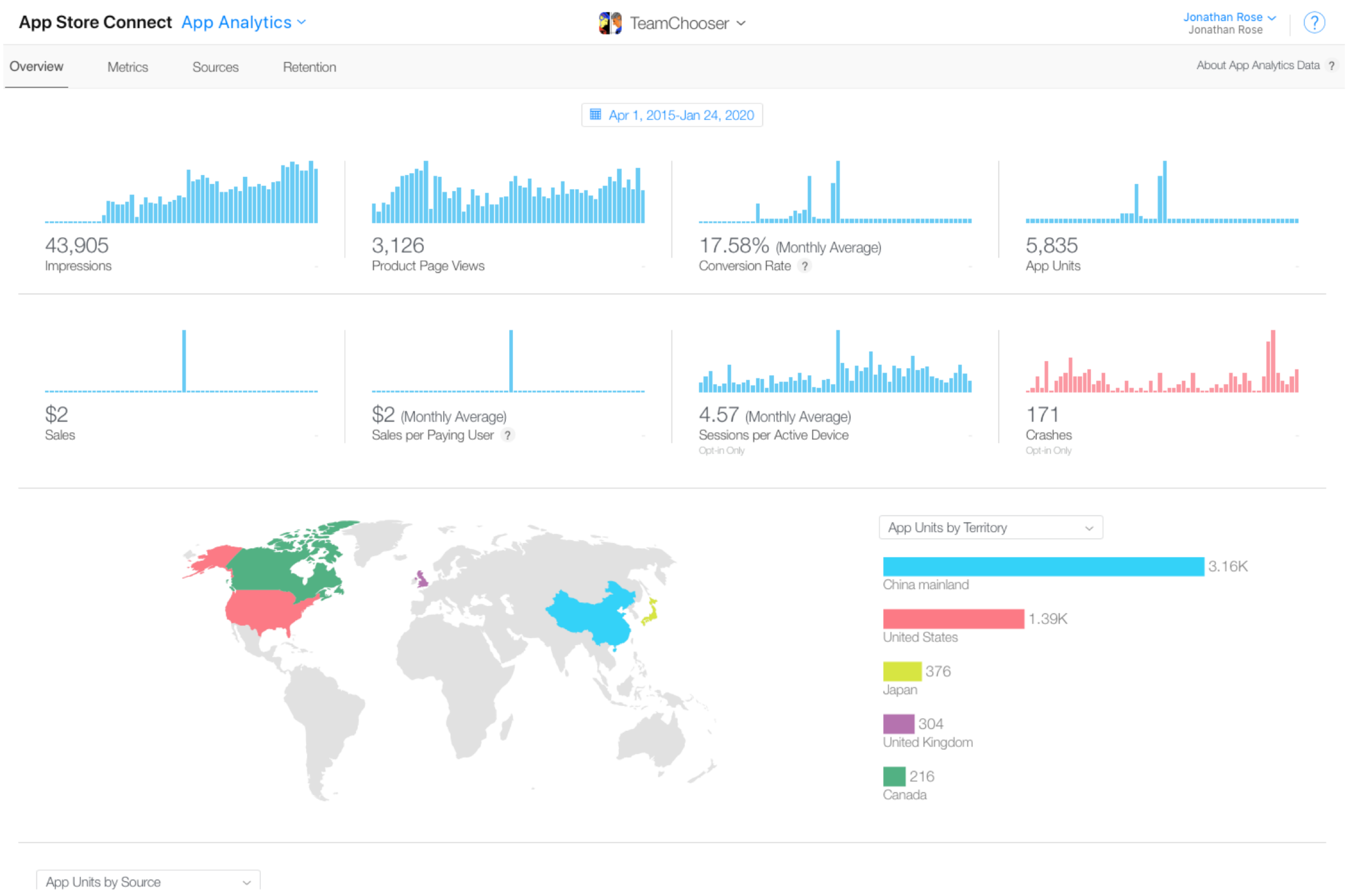
Carrier 2:08 PM
7 players selected (D:4 O:3)
Back SoccerTuesday
Benny Defense
Chen Offense
Doofus Offense
Doofus Offense
Fred Offense
John Defense
Jorge Defense
Manny Offense
Margie Defense
Oliver Defense
Done editing players

Do you play friendly pickup sports, like hockey, soccer or basketball? Would you like help choosing which players to be on each team, so that everyone enjoys the game? Then TeamChooser is the app for you! TeamChooser will work for pretty much any two-team game you can think of: rugby scrimmages, volleyball, baseball, and flag football.

more

- Free!
- 6000 total Downloads
- Mostly in US/Canada, but a few in UK, Ireland, Japan, Norway, Romania, Portugal, Australia, Denmark, Finland

Units



App Store Connect App Analytics

TeamChooser

Jonathan Rose
Jonathan Rose

Overview Metrics Sources Retention

About App Analytics Data ?

Apr 1, 2015-Jan 24, 2020

43,905
Impressions

3,126
Product Page Views

17.58% (Monthly Average)
Conversion Rate ?

5,835
App Units

$2
Sales

$2 (Monthly Average)
Sales per Paying User ?

4.57 (Monthly Average)
Sessions per Active Device
Opt-in Only

171
Crashes
Opt-in Only

App Units by Territory

China mainland	3.16K
United States	1.39K
Japan	376
United Kingdom	304
Canada	216

App Units by Source

Lesson Learned

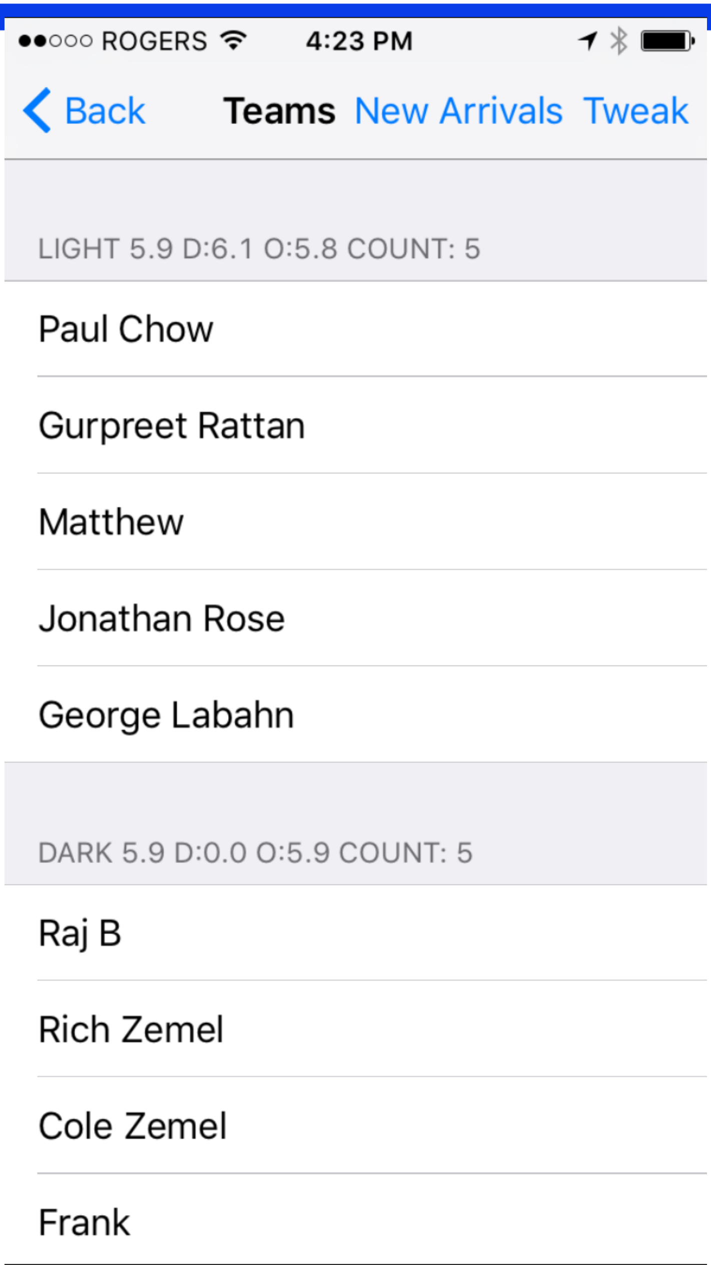
- From last time: Ratings of players, key part of engine, **must be kept secret!**
- Can add password to App so no-one can grab phone and look at their ratings or anyone else’s

Sad Outcome: Agar

- The day after the first use of TeamChooser on Wednesday
  - Recall very poor chooser – Agar
  - Folks joked and called it the iGar.
- Agar **never played again on Wednesday!**
  - Still plays hockey, but not with team he played with for decades

Feature: No-One Chosen Last!

- The order that people are listed (and spoken out loud by user) is randomized
- There is no way to infer rating


●●○○○ ROGERS 4:23 PM

Back Teams New Arrivals Tweak

LIGHT 5.9 D:6.1 O:5.8 COUNT: 5

Paul Chow

Gurpreet Rattan

Matthew

Jonathan Rose

George Labahn

DARK 5.9 D:0.0 O:5.9 COUNT: 5

Raj B

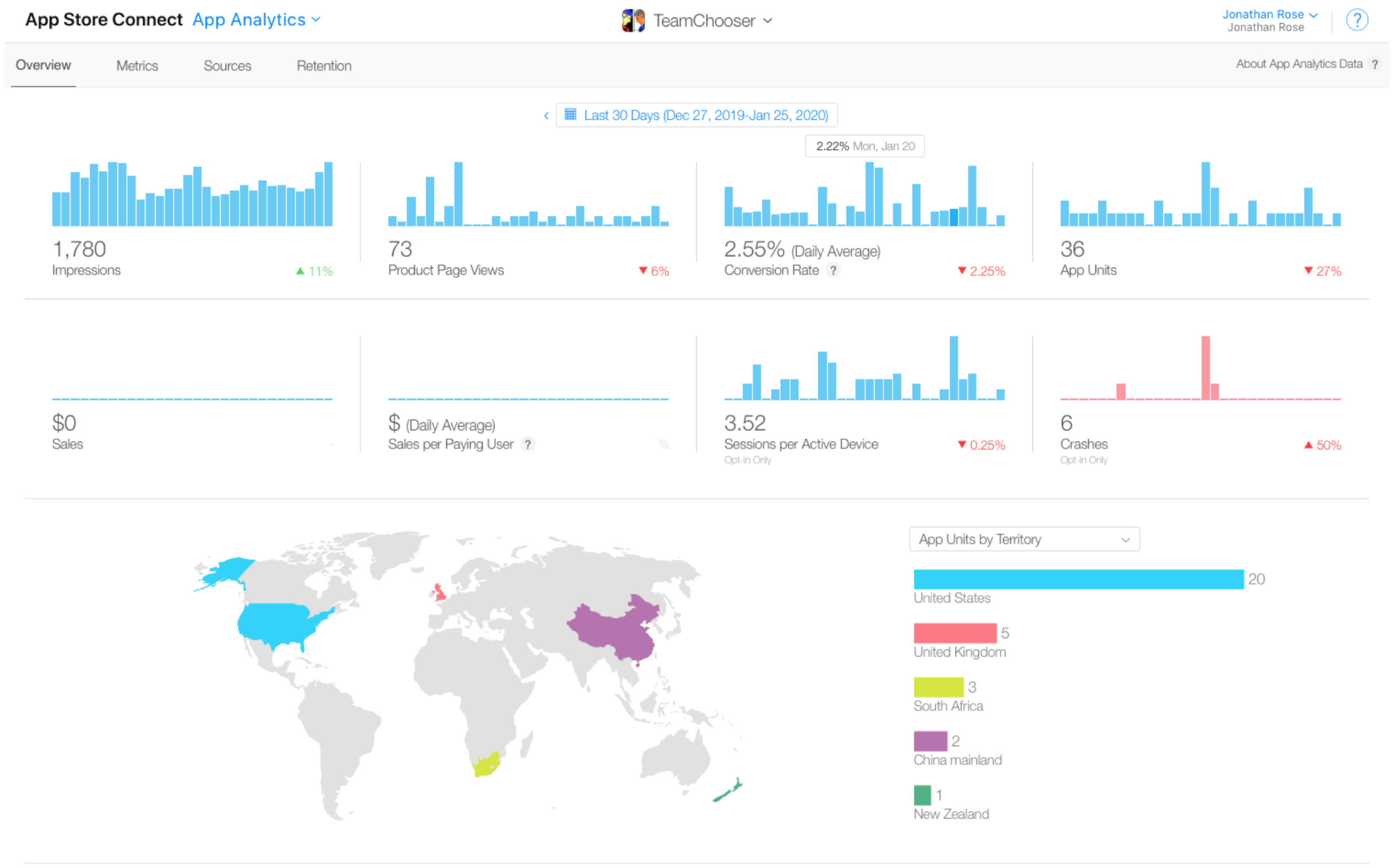
Rich Zemel

Cole Zemel

Frank

Downloads Over the Last 30 Days

- Is delightful to see people using!



App Store Connect App Analytics
TeamChooser
Jonathan Rose
Jonathan Rose

Overview Metrics Sources Retention
About App Analytics Data ?

Last 30 Days (Dec 27, 2019-Jan 25, 2020)

2.22% Mon, Jan 20

1,780	Impressions	▲ 11%
73	Product Page Views	▼ 6%
2.55% (Daily Average)	Conversion Rate ?	▼ 2.25%
36	App Units	▼ 27%
$0	Sales	-
$ (Daily Average)	Sales per Paying User ?	%
3.52	Sessions per Active Device (Opt-in Only)	▼ 0.25%
6	Crashes (Opt-in Only)	▲ 50%

App Units by Territory

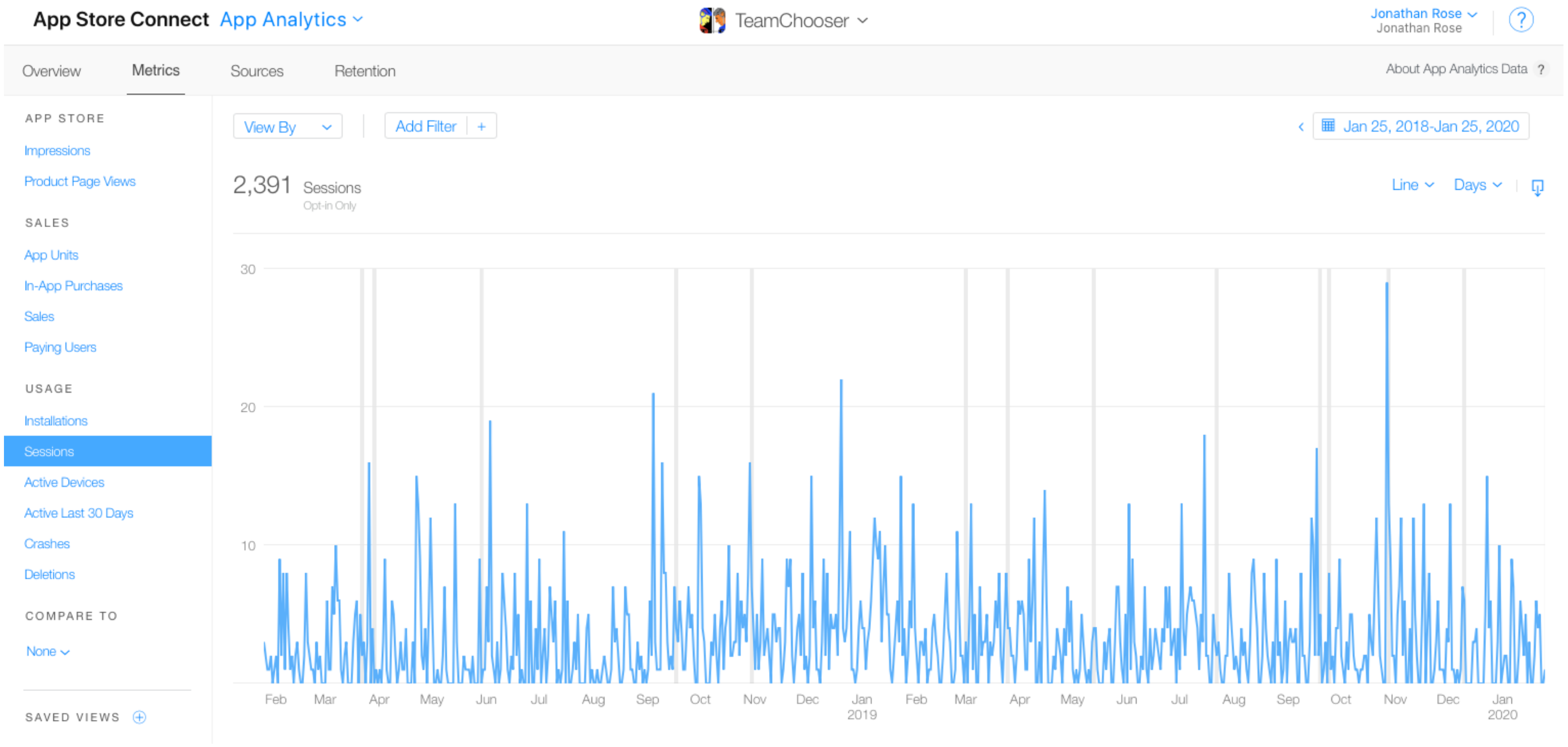
United States	20
United Kingdom	5
South Africa	3
China mainland	2
New Zealand	1

Is Anyone Using it Who Bought It?

- Instrumented both with Apple Analytics & Firebase lytics
  - Very easy to insert into any app

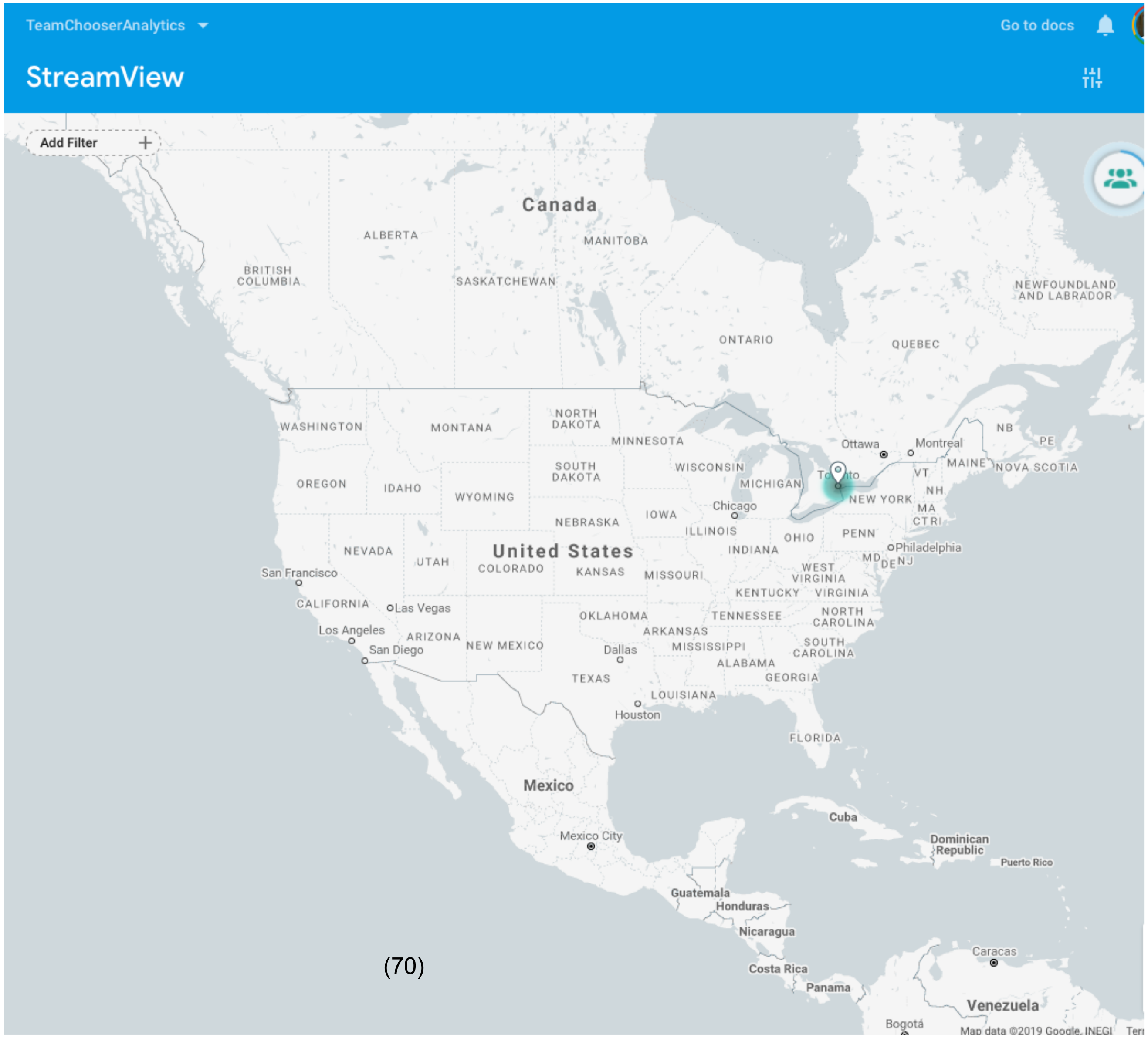
- Reports:
  - # of users sessions, amount of time spent on app
  - Specific pages/events, as you wish from each user
  - Location of user, if already use GPS (no other ID).
  - **Anything I wish to report!**

Session Report from Apple Analytics

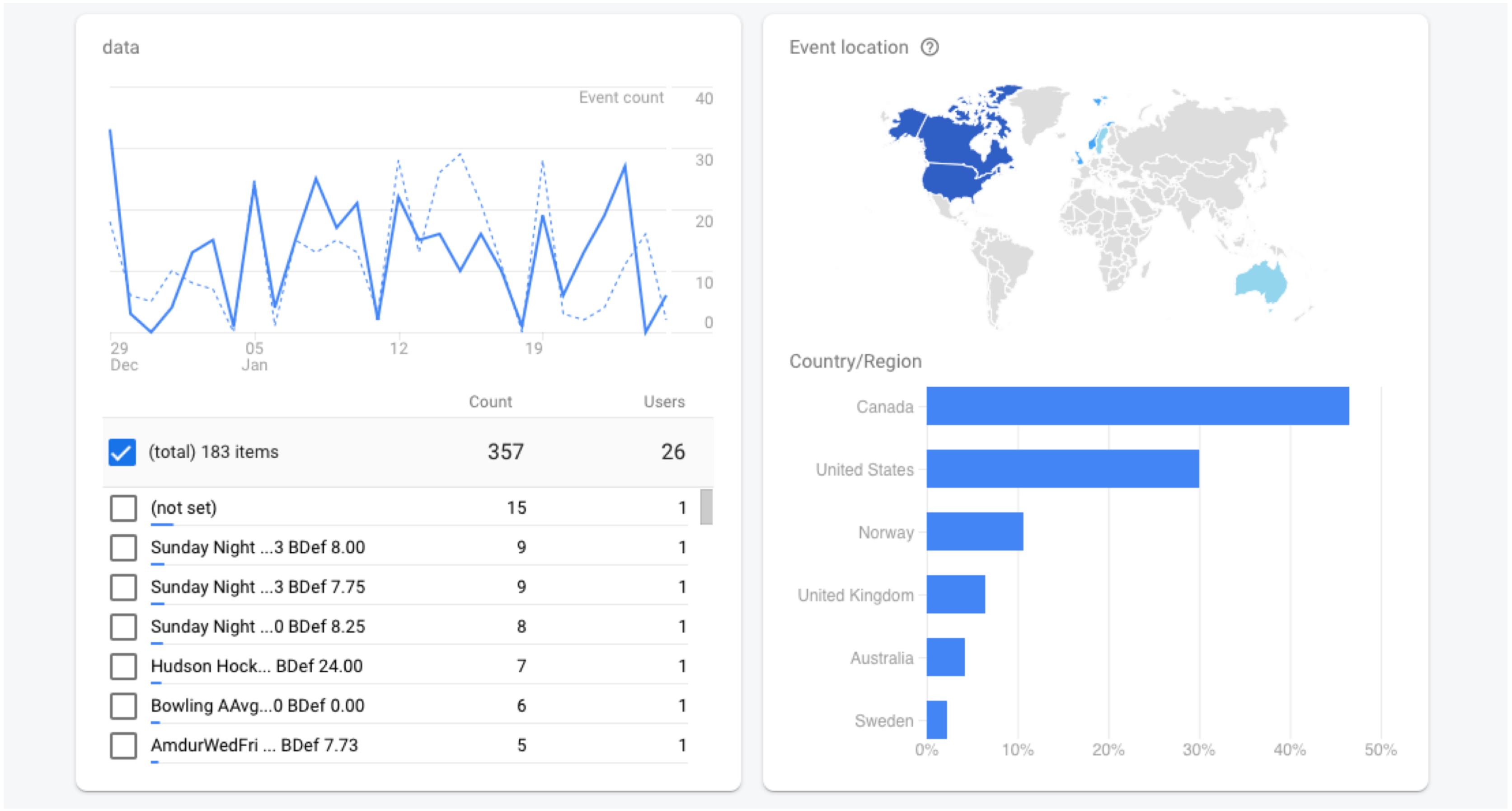


2,391 Sessions
Opt-in Only

Stream View – Live!



Event Logs



data

	Count	Users
✓ (total) 183 items	357	26
(not set)	15	1
Sunday Night ...3 BDef 8.00	9	1
Sunday Night ...3 BDef 7.75	9	1
Sunday Night ...0 BDef 8.25	8	1
Hudson Hock... BDef 24.00	7	1
Bowling AAvg...0 BDef 0.00	6	1
AmdurWedFri ... BDef 7.73	5	1

Event location

Country/Region

Small Programming Note: The Android Life Cycle

Android Application Life Cycle

- Recall: Activities are screens that the user sees, and associated process
- Android manages these Activities as a **stack**.
- When a new activity is started, it is placed on the top of the stack and becomes the running activity
- The previous activity always remains below it in the stack,
  - and will not come to the foreground again until the new activity exits.

Important to Pay Attention to ‘LifeCycle’

- To ensure app behaves well in several ways, including:

1. Does not crash if the user receives a phone call or switches to another app
2. Does not consume valuable system resources when the user is not actively using your app
3. Does not lose the user's progress if they leave your app and return to it at a later time
4. Does not crash or lose the user's progress when the screen rotates between landscape and portrait orientation.

An Activity Can Be in 1 of 4 ‘States’

State 1: Active/Running

- Activity in the foreground of the screen (at the top of the stack)
- Has ‘focus’, meaning user interactions go to it.

State 2: Paused

- activity has lost focus but is still visible
- a new smaller or transparent activity has focus on top of the activity)
- A paused activity is completely alive (it maintains all state and member information and remains attached to the window manager), but can be killed by the system in extreme low memory situations.

Activity States 3 and 4

State 3: Stopped

- activity is completely obscured by another activity
- retains all state and member information
- no longer visible to the user so its window is hidden
- it will often be killed by the system when memory is needed elsewhere.

State 4: Destroyed

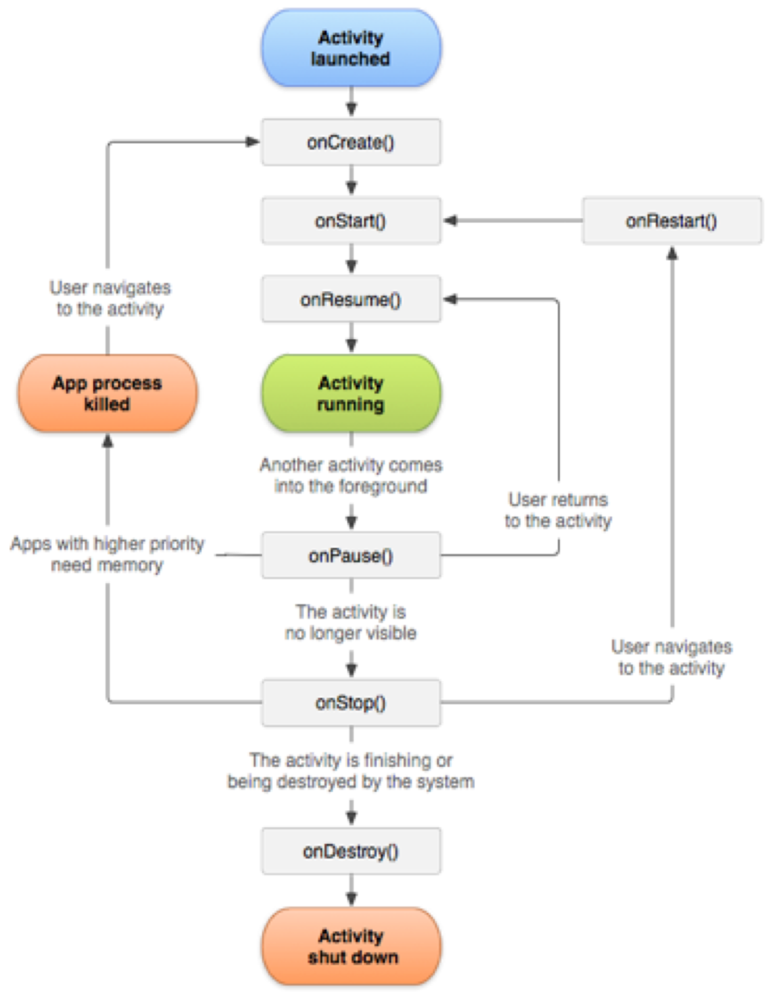
- If an activity is paused or stopped, the system can drop the activity from memory by either asking it to finish, **or simply killing its process.**
- When displayed again to the user, it must be completely restarted and restored to its previous state.

Android Talking to Your App

- The Android operating system asks (or tells) your app to go into those different states by invoking methods associated with your Activity

Methods Called By Android to Change States

- Diagram shows states and methods called to change state
  - Colours: the states
  - https://developer.android.com/guide/components/activities/activity-lifecycle


Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

Three Key States

- Activity can be in 1 of 3 states for long period of time:

1. Resumed
   - In this state, the activity is in the foreground and the user can interact with it. (Also sometimes referred to as the "running" state.)
2. Paused
   - In this state, the activity is partially obscured by another activity—the other activity that's in the foreground is semi-transparent or doesn't cover the entire screen. The paused activity does not receive user input and cannot execute any code.
3. Stopped
   - In this state, the activity is completely hidden and not visible to the user; it is considered to be in the background. While stopped, the activity instance and all its state information such as member variables is retained, but it cannot execute any code.

State Management

- The other states (Created and Started) are transient and the system quickly moves from them to the next state by calling the next lifecycle callback method. That is, after the system calls onCreate(), it quickly calls onStart(), which is quickly followed by onResume().

- Depending on the complexity of your activity, you probably don't need to implement all the lifecycle methods.
- However, it's important that you understand each one and implement those that ensure your app behaves the way users expect.

References

1. The Android Documentation:

https://developer.android.com/guide/components/activities/activity-lifecycle

2. Murphy, Busy Coder's Android, Chapter "Activities and their Lifecycles" (Page 316)

- Once your project gets going, it is really important to read through this and understand it
  - Previous years' students pointed out that this was the key thing they had not understood in Android, that caused the most problems

The Key ‘LifeCycle’ Methods

OnCreate()

- Familiar with already – brings the activity to life

OnPause()

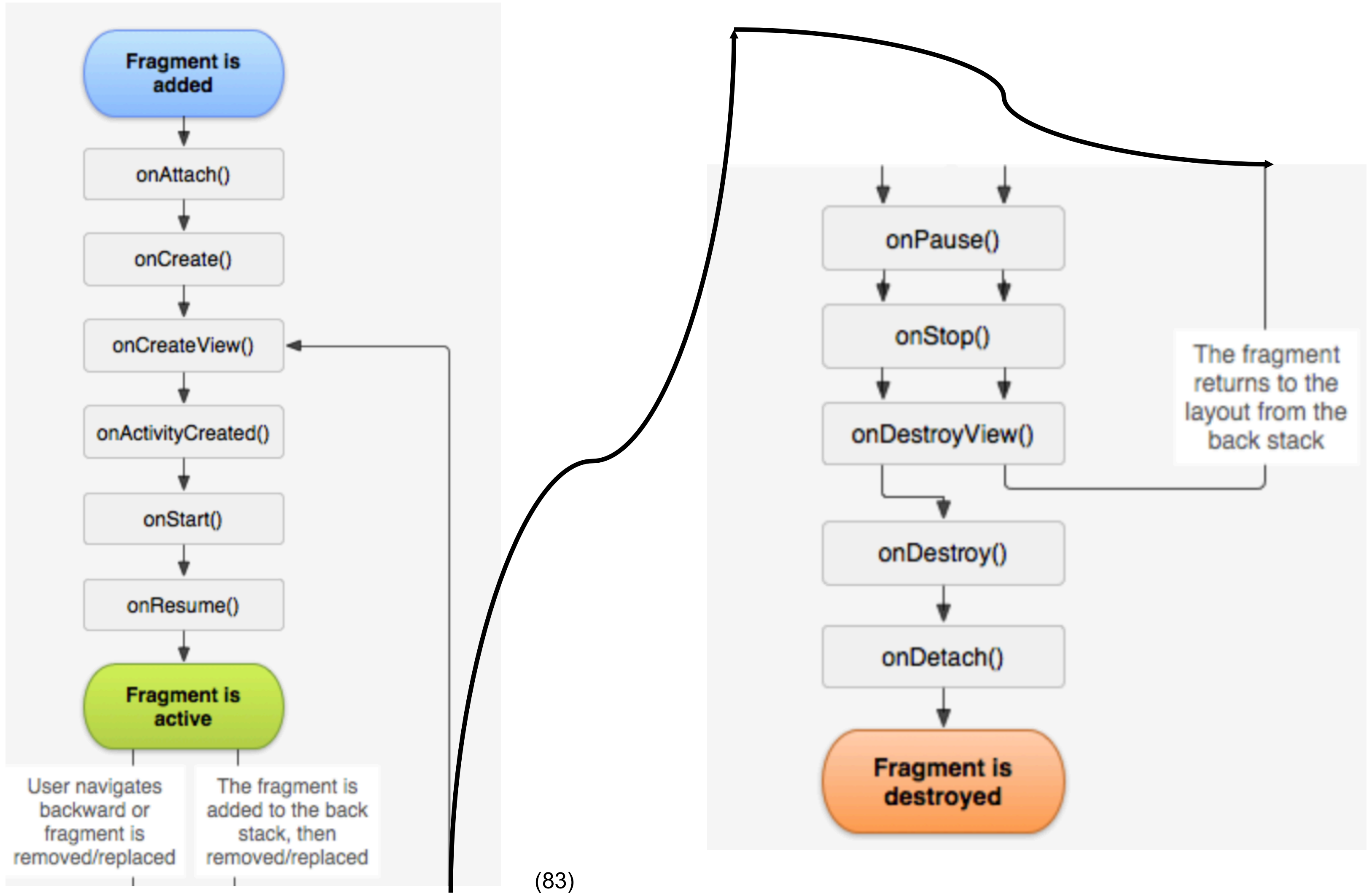
- Another Activity has gained the ‘focus’
- Should stop any background threads, release large resources (such as a camera)
- **No guarantee that OnDestroy() will be called**, so best to save **all** state here

OnResume()

- Called as activity starts, **or** is restarted from a pause
- Can recall state from file, refresh the User Interface – see example

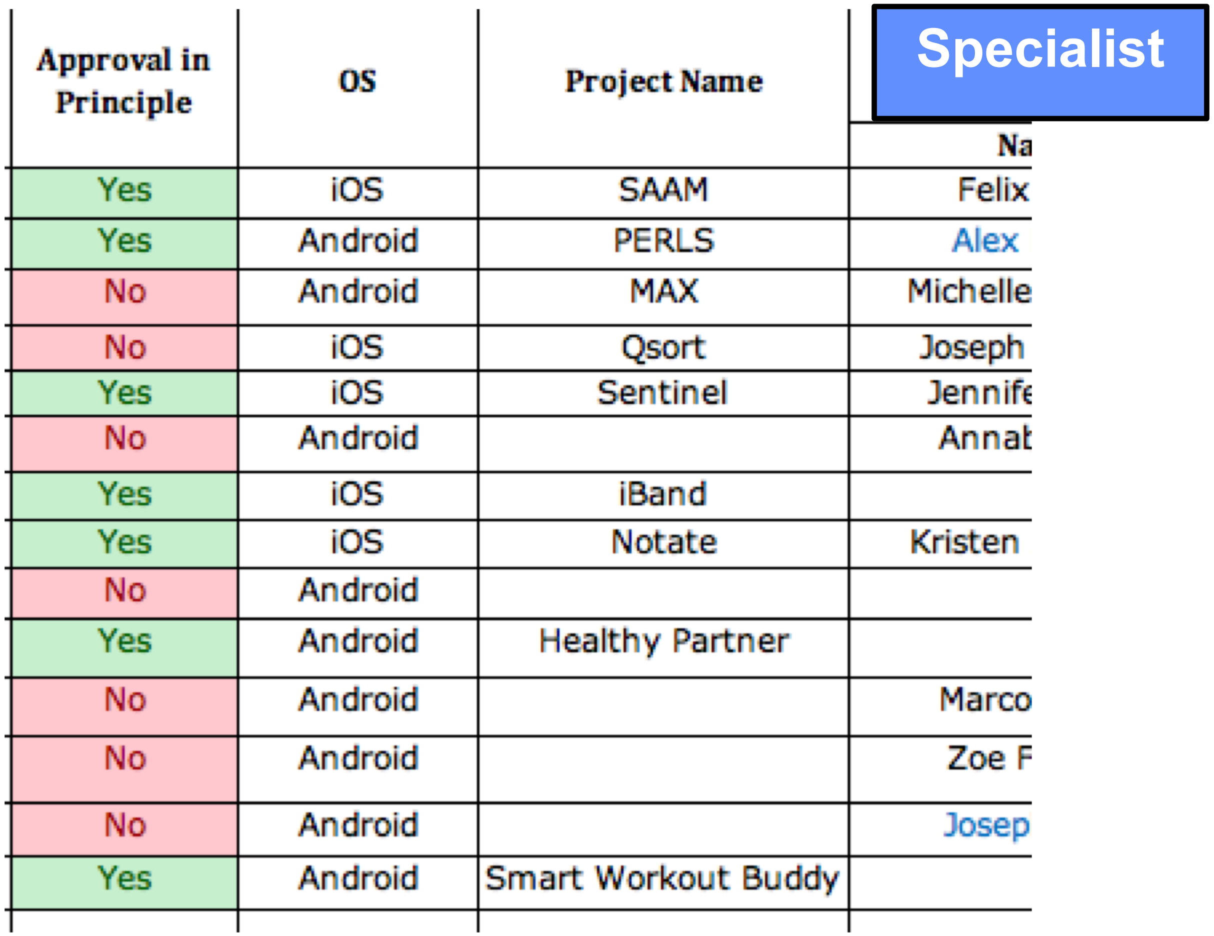
Fragments Behave Similarly

http://developer.android.com/guide/components/fragments.html



Topic Discussion Time

Groups in Progress



Specialist

Approval in Principle	OS	Project Name	Na
Yes	iOS	SAAM	Felix
Yes	Android	PERLS	Alex
No	Android	MAX	Michelle
No	iOS	Qsort	Joseph
Yes	iOS	Sentinel	Jennife
No	Android		Annat
Yes	iOS	iBand	
Yes	iOS	Notate	Kristen
No	Android		
Yes	Android	Healthy Partner	
No	Android		Marco
No	Android		Zoe F
No	Android		Josep
Yes	Android	Smart Workout Buddy	