

Imbue Reality

Eli Spiegel, Christopher Livingston, Billy
Rathje, Patrick Green



im·bue

inspire or permeate with a feeling or quality

What is Imbue?

- A mobile application that allows users to learn about the places around them and to connect with others

What does the app do?

- Using Augmented Reality, we display nearby buildings and events.
- A compass-bound map view.
- Building/Event information screens
- Share your location with your Facebook friends

Video

What is Augmented Reality?

- Is a direct or indirect view of the real world environment enhanced with the help of sensors.
- Imbue uses GPS, Compass and Camera sensors.

Original Motivation

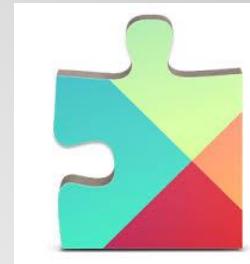
- To provide campus tours for Puget Sound and other campuses using augmented reality
- We found this view to be limited
- We've since expanded the app to provide augmented and location features for the entire world

Motivation Part II

- No campus tours apps currently use AR or interactive maps
- Maps like Google Maps do not incorporate AR, but could easily do so

APIs Used

- Parse
- GoogleMaps
- GooglePlaces
- Google Play
- Facebook



Parse



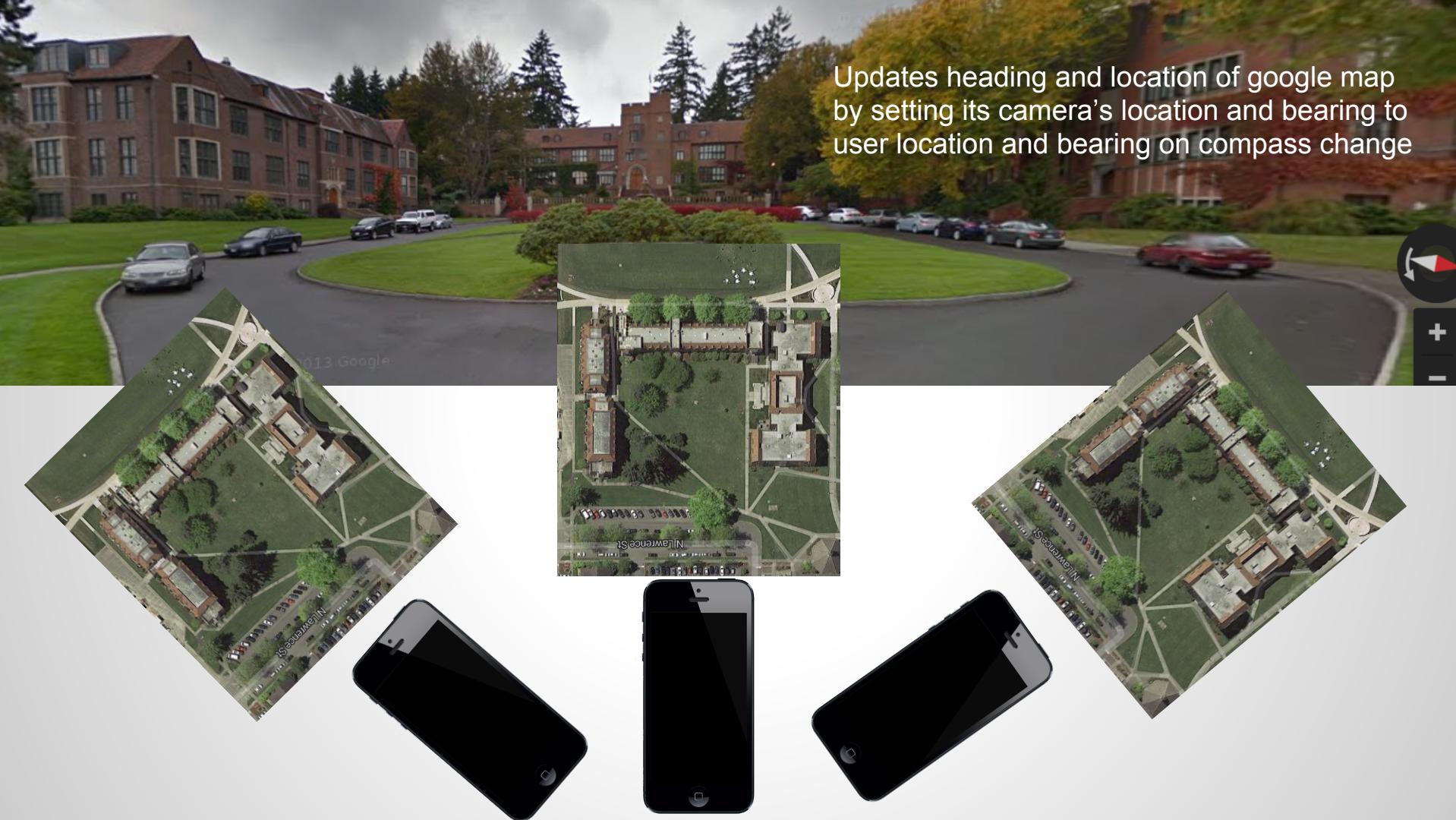
Building AR Algorithm

Angle of view: 100 degrees



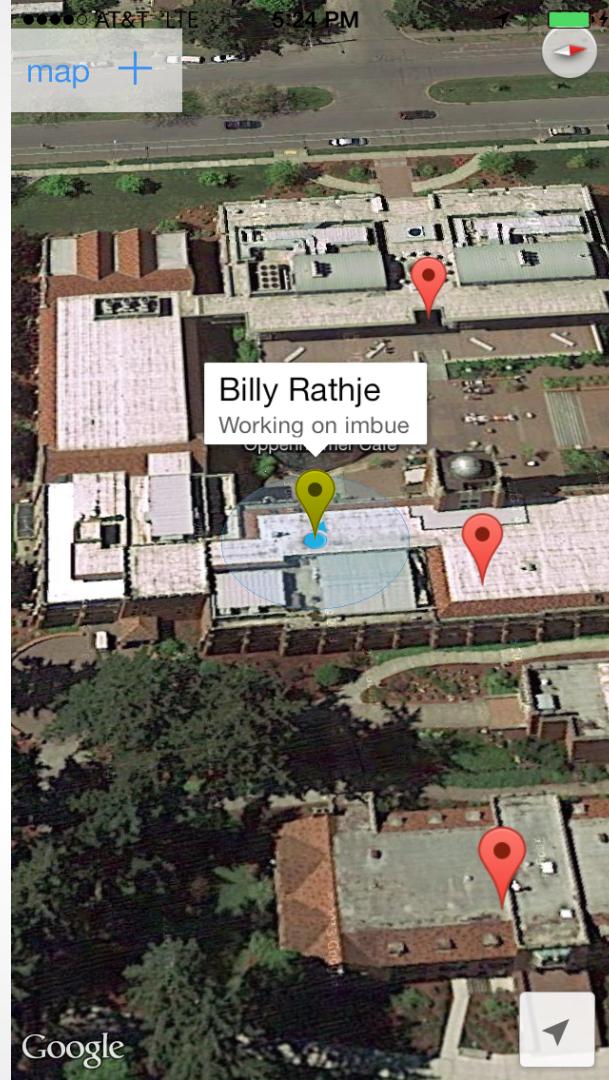
GPS - ~100 m

Updates heading and location of google map by setting its camera's location and bearing to user location and bearing on compass change



Nearby Friends

- Location is stored server-side. This is checked against a list of your facebook friends, and their locations are placed on your map.
- If you move far enough from your last shared location, your location will be set to “do not share.”



Share your current location

Please enter a message

[remove last location](#)

[ok](#)

[Cancel](#)

Q W E R T Y U I O P

A S D F G H J K L

↑ Z X C V B N M ➔

123



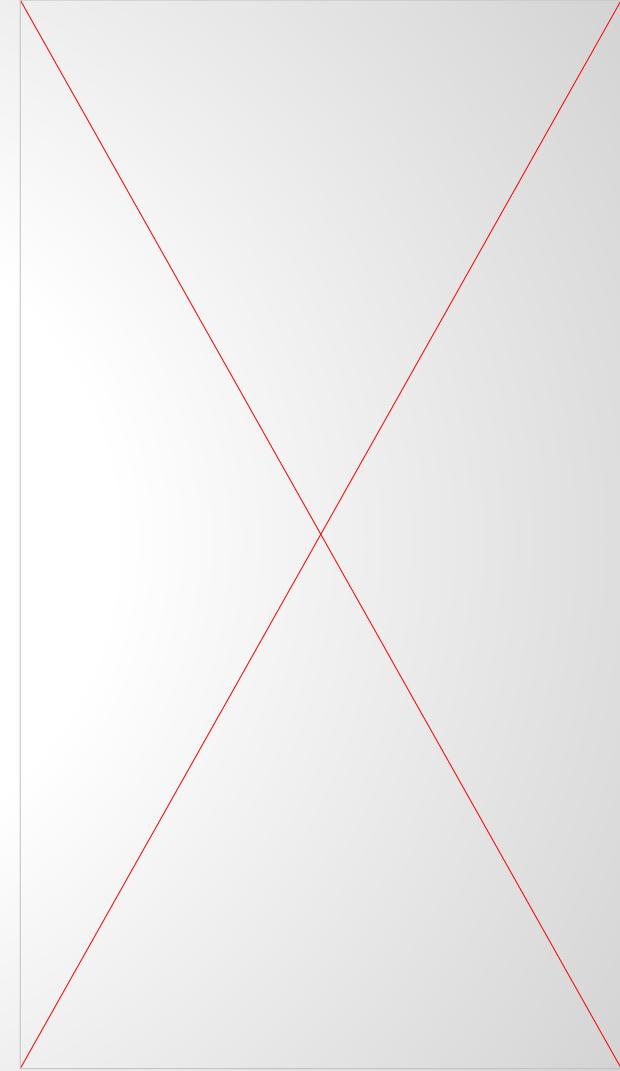
space

return

MapView Example

- Markers help display info
 - Red = Buildings
 - Yellow = Inside POI
 - Blue = Facebook Events
 - Purple = Friend's Location*

*yellow on iOS



iOS views

●●●○ AT&T LTE 9:59 PM

Back

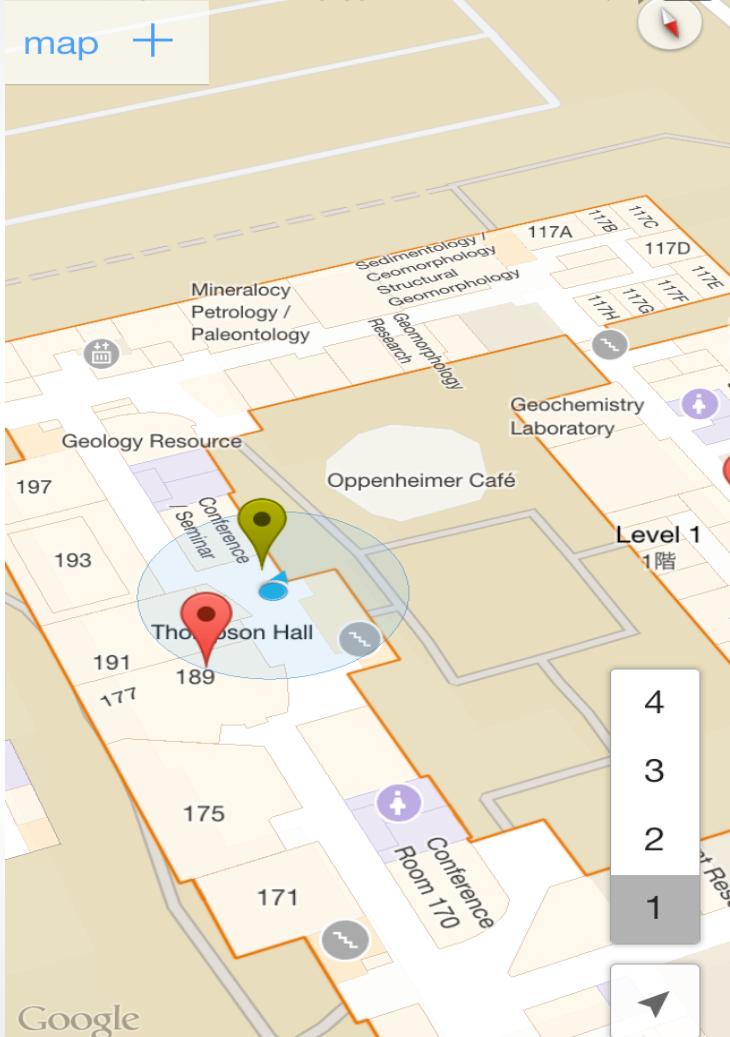
McIntyre Hall



McIntyre Hall is home to the departments of Comparative Sociology, Economics, the School of Business and Leadership as well as two general access computer labs. It also houses the auditorium where students watch Campus Films, the Honors Film Series or the latest offerings of the Japanese Animation Club.

●●●○ AT&T LTE 9:59 PM

map +



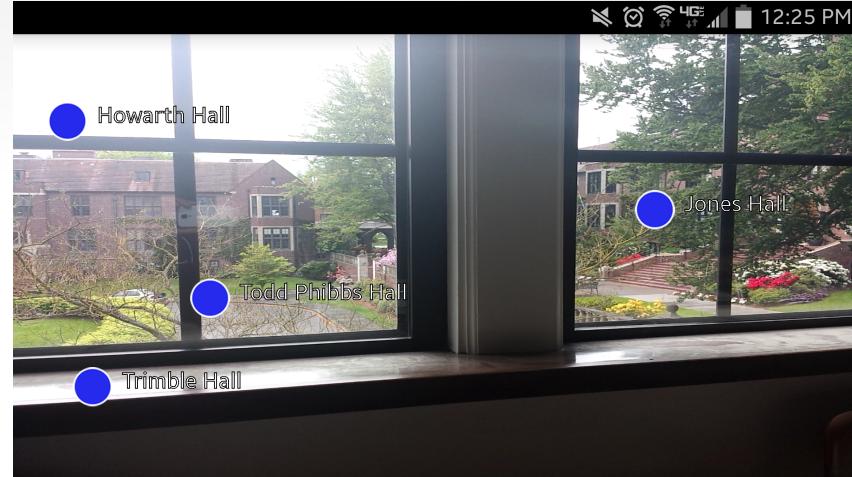
Camera View

Two View Components:

- The camera
- The bitmaps and text drawn on top
- Open GL

AR calculations happening in real time

Google Places



- Query Buildings/Businesses
 - 1000 meter radius around current location
 - Based on prominence
 - Events, too
- Displayed on map and in camera

Android

- Activities for different functions
 - MainActivity
 - CameraActivity
 - SearchActivity
 - Building/Event Info Activity
- Each has an View (XML)
- API 14(Ice Cream Sandwich) and Up

iOS

- ViewController

- Main functionality. Implements location algorithm
 - Creates map with various markers
 - Displays camera view on rotation

- ContentViewController

- Displays content/detail view:
 - Makes a server query for building information and photo
 - Downloads Facebook event photos

- Bearing

- Calculates bearing to a coordinate

Demo Time

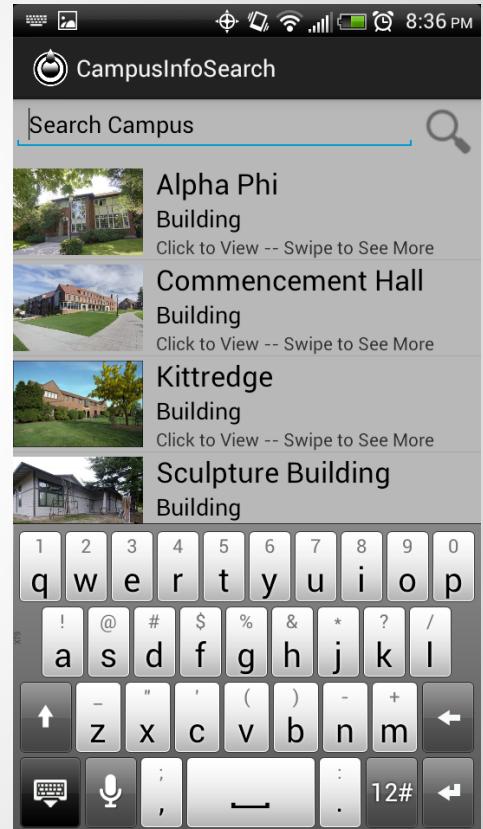
ios: [https://www.youtube.com/watch?
v=MQ8Pa7IK84k](https://www.youtube.com/watch?v=MQ8Pa7IK84k)

OpenGL

- Would allow a z-axis for the camera view
- Very choppy
- Text was hard to display

Hosting Data Locally in SQL

- Store data hosted on Parse locally
- Allows for faster queries
- Requires less network data
- Unique data modification and display
- Easier to define relations



Cloud Vs. Local

- As the project became more social hosting data internally became more difficult and became less relevant
- Pulling information from the cloud was simply quicker and easier

Project Challenges

- Getting set up with multiple devices
- Integrating with Parse
- OpenGL
- SQLite DB
- Keeping track of queries
- Refactoring
- iOS code differences
- Time for review

Code Differences

Objective C:

```
GMSMarker *marker = [[GMSMarker alloc] init];  
[_arrayOfMarkers addObject: marker];
```

Java:

```
GMSMarker marker = new GMSMarker();  
arrayOfMarkers.add(marker);
```

What We Learned!

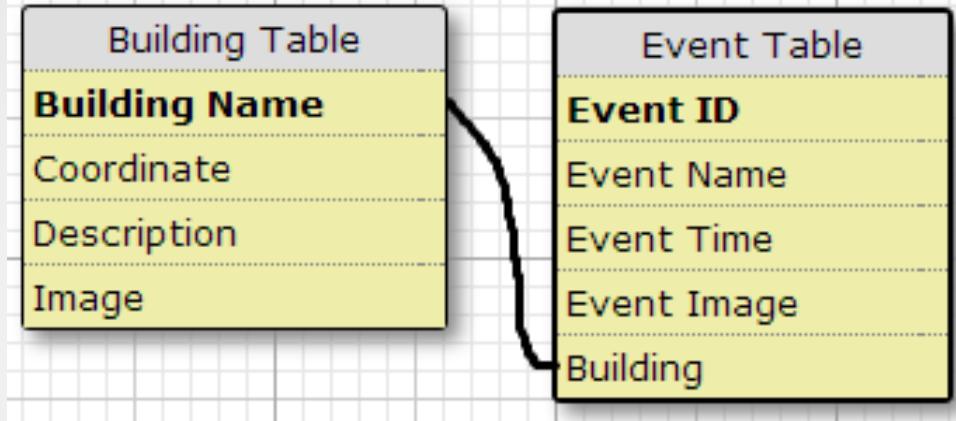
- Android and iOS development
- Multiple APIs
- JSON
- Cloud Code
- Database management
- Open GL
- SQLite

Android link



<https://play.google.com/store/apps/details?id=com.cs440.capstone>

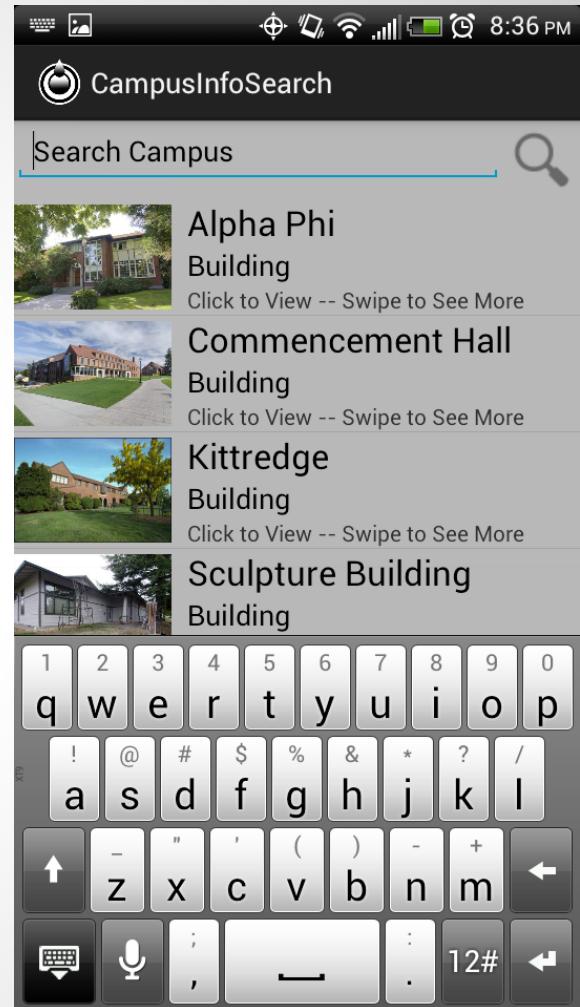
Logical Representation of Data



- Assumed that every building would have a unique name
- Building name was used as a Foreign Key within the Event Table

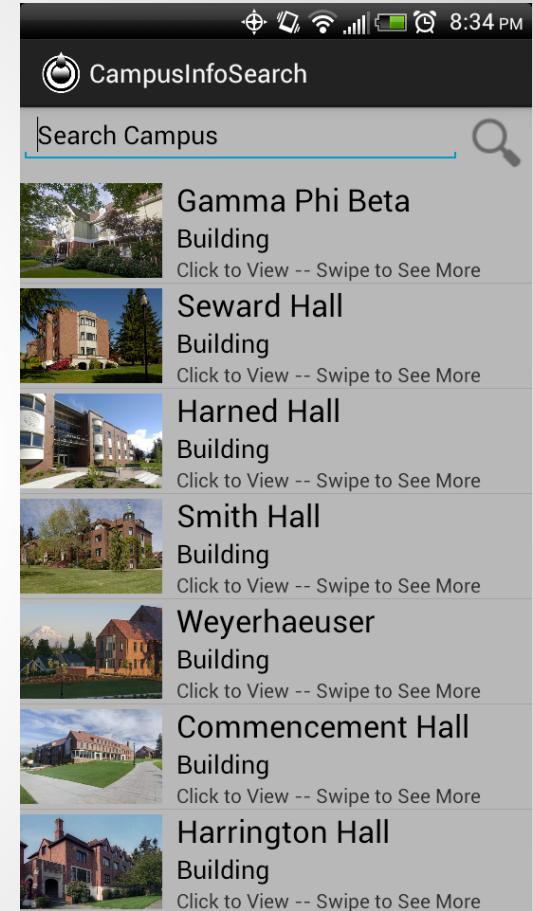
What I Used

- SQLite
- SQLiteOpenHelper
- ParseAnalytics
- Basic Database management techniques



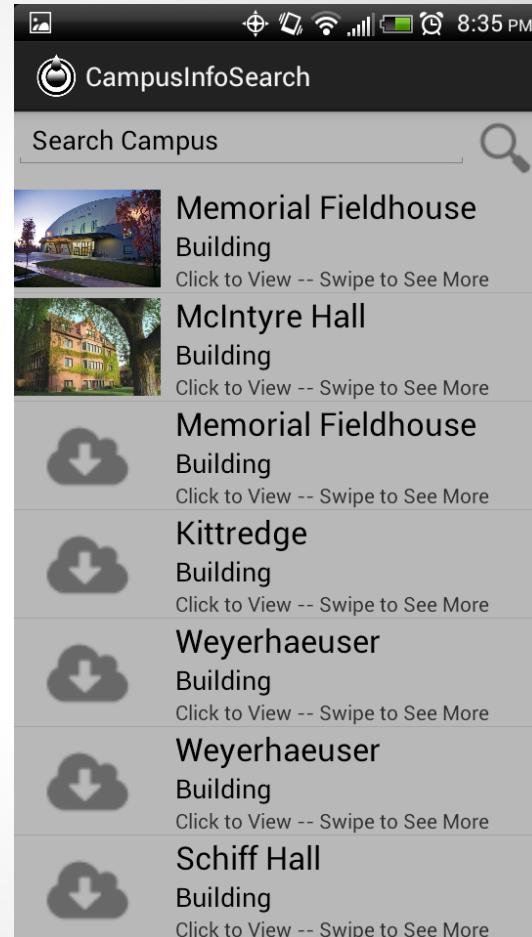
View Level of Data

- Provide the user with suggested search items
- Access to building information including images, and events
- Search for building by name or by a common substring



Challenges

- Building the bridge between Parse and SQL
- Storing and retrieving images
- Loading data within the UI thread
- Thread management issues



Successes

- Built a SQL database
- Data stored in the database changes dynamically with the data hosted on Parse
- Able to query locally and display relevant information
- Multi Threaded code to retrieve data
- User interface successfully implemented

