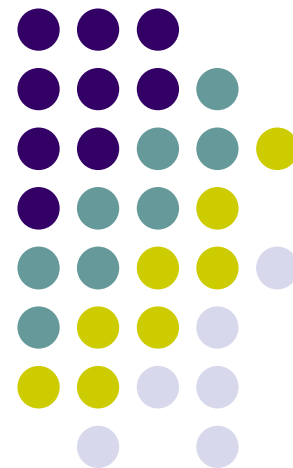


# Ubiquitous and Mobile Computing

## CS 528: *Parking Assist*

---

*Yijie Yan, Qinlun Luan,  
Wei Xiong, Zinan Yue  
Computer Science Dept.  
Worcester Polytechnic Institute (WPI)*





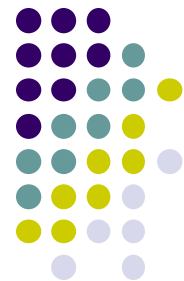
# Agenda

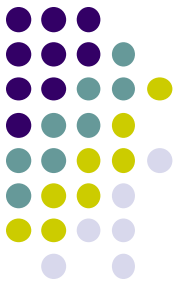
- Problem Motivation
- Related Work
- Methodology
- Implementation Plan
- Evaluation Plan
- Timeline
- Difficulty Point



# Problem Motivation

- Motivation for such problem
  - Hard to park in cities like Worcester, Boston, etc
  - Worse during rush hours
  - Easy to forget parking location



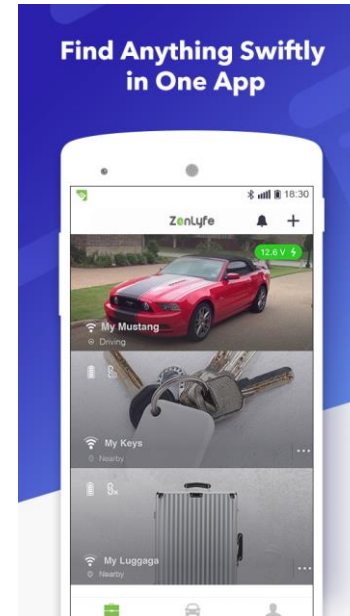
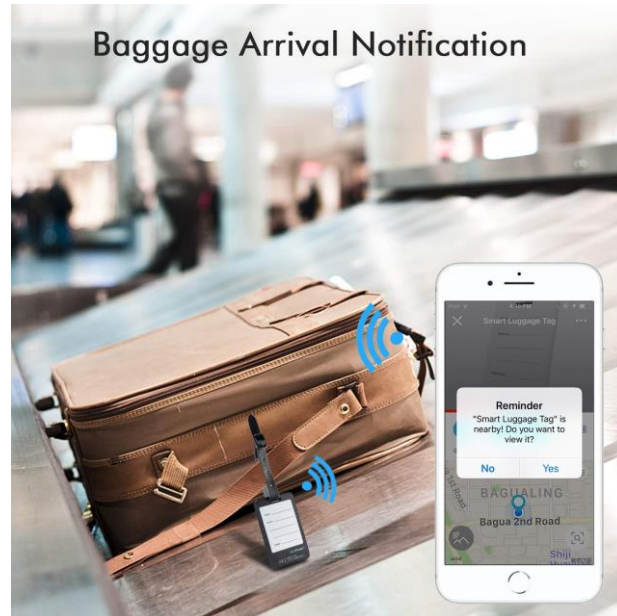


# Problem Motivation

- Why important?
  - 56 % of drivers forgot parking location
  - 1/7 drivers admit this fact
- Why using mobile solution?
  - Powerful components like camera, Bluetooth and GPS
  - Popular and easy to carry



# Related Works



## **Zenlife – SwiftFinder**

Track the check-in luggage Information  
By Receiving Bluetooth signal

[Amazon Reference](#)

[Google Play Reference](#)



# Methodology

- GPS Positioning
  - Take a photo when parking
  - Store and retrieve location information
  - Create Geofence
- Bluetooth Connection
  - Detect whether drivers leave
- Geofence
  - Send notification to user when close to car



# Methodology(con't)

- Image Recognition API
  - Recognize car photo and retrieve useful information
  - By Sighthound
- Speech Recognition API
  - Dialogflow API for voice command assist





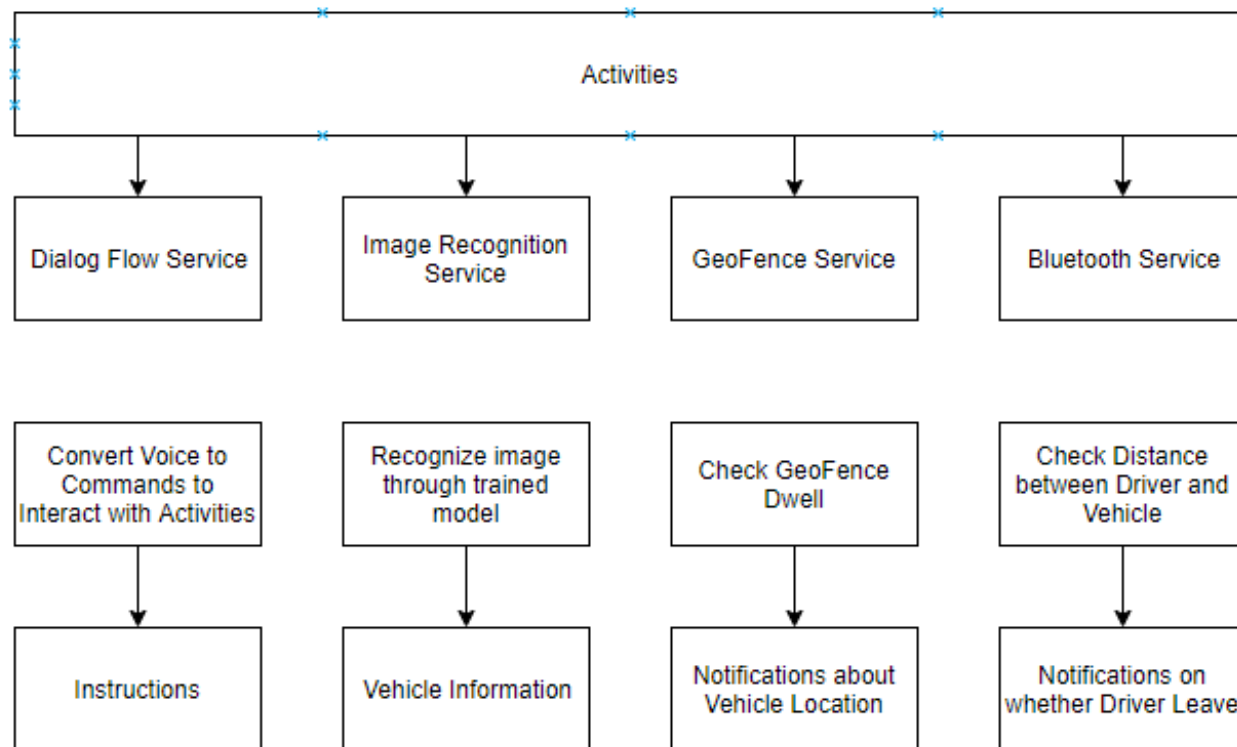
# Implementation Plan

- Modules
  - Bluetooth
  - Geofence & GPS
  - Sighthound (Image Recognition API)
  - Dialog Flow (Speech recognition API)



# Implementation Plan

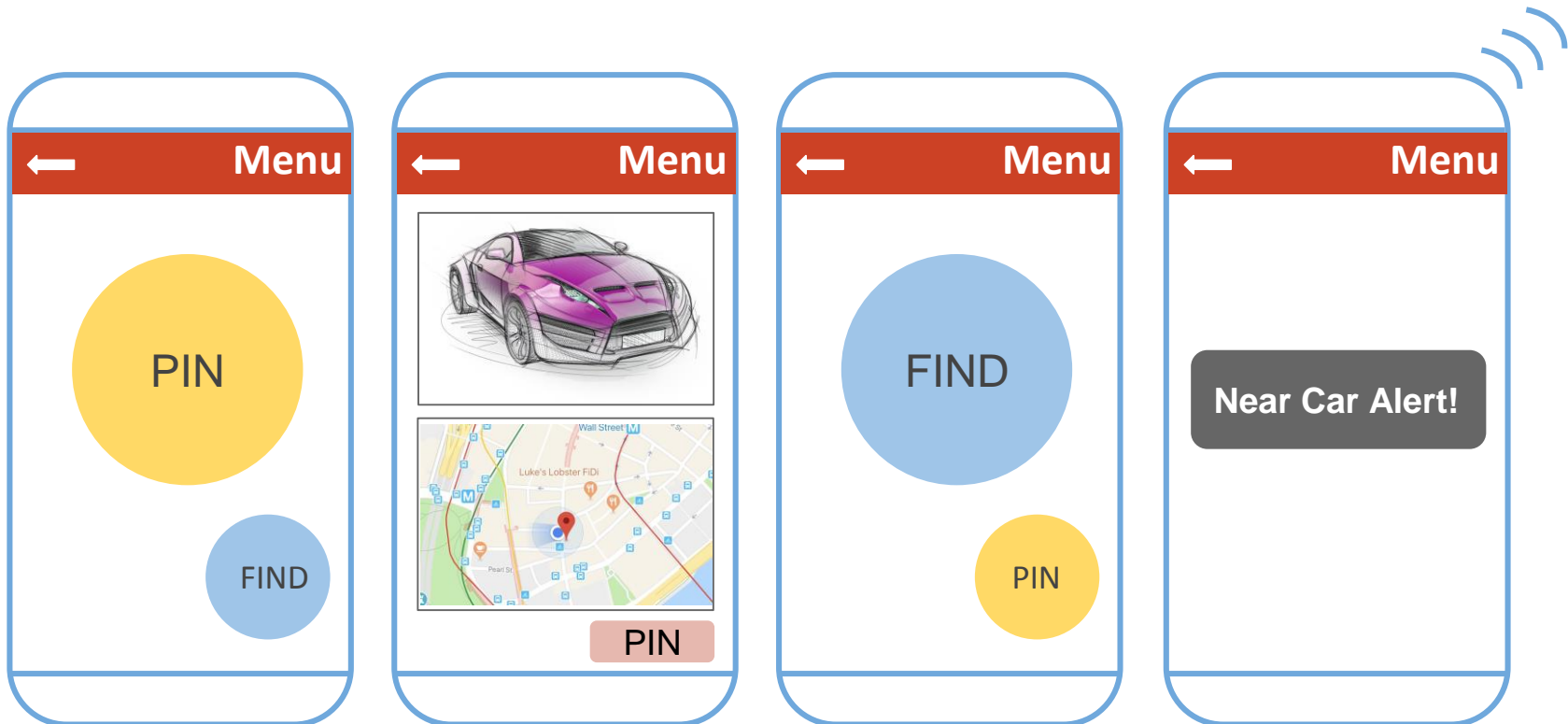
- Software Architecture



# Implementation Plan



- User Interface





# Evaluation Plan

## Use Cases Test

JUnit

Expresso

Travis CI

## Alpha Test

UI Performance

APP Component Integrations

Code Refactoring

Stable Development Velocity

## Public Beta Test

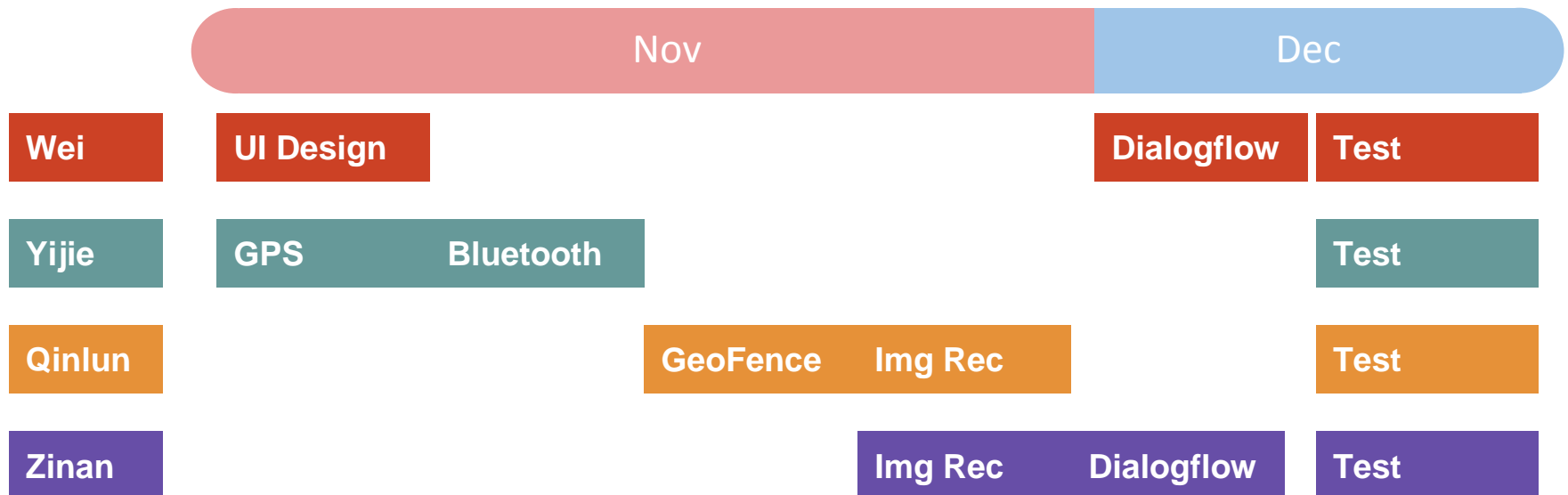
Distribute the Beta version of the application to the selected WPI students, faculties and staffs

Get feedbacks



# Timeline

- The five major parts would be completed each within one week, and one week for application test





# Difficulty Points

- UI Design: 5+ Screens
- Maps
- Location Sensing
- Camera: taking pictures
- Communicate via Bluetooth

- GeoFencing
- Speech Recognition

- Machine Learning: detect plate number, maker, model

- Total Points: 42



# References

- **56 % of drivers forgot parking location:** <https://www.prnewswire.com/news-releases/confusedcom-reveals-56-of-drivers-forget-where-they-park-123446154.html>
- **1/7 drivers admit this fact:** <https://www.cars.com/articles/parking-in-these-cities-is-the-worst-1420697595754/>
- **Parking is hard:** <https://www.dailymail.co.uk/news/article-2688266/The-200-years-spend-looking-lost-cars-One-seven-drivers-admit-forgotten-parked-vehicle.html>
- **Android GeoFence Documentation:** <https://developer.android.com/training/location/geofencing>
- **Dialog flow:** <https://dialogflow.com/>
- **ZenLyfe:** [https://play.google.com/store/apps/details?id=com.snappwish.swiftfinder&hl=en\\_US](https://play.google.com/store/apps/details?id=com.snappwish.swiftfinder&hl=en_US)
- **Kaggle Dataset:** <https://www.kaggle.com/jutrera/stanford-car-dataset-by-classes-folder>





**Thank You**  
**Q & A**