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

Diagram:

- Dialog Flow Service
- Image Recognition Service
- GeoFence Service
- Bluetooth Service

- Convert Voice to Commands to Interact with Activities
- Recognize image through trained model
- Check GeoFence Dwell
- Check Distance between Driver and Vehicle

- Instructions
- Vehicle Information
- Notifications about Vehicle Location
- Notifications on whether Driver Leave
Implementation Plan

- User Interface
## Evaluation Plan

<table>
<thead>
<tr>
<th>Use Cases Test</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>JUnit</td>
<td>Expresso</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Alpha Test</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>UI Performance</td>
<td>APP Component Integrations</td>
</tr>
<tr>
<td>Code Refactoring</td>
<td>Stable Development Velocity</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Public Beta Test</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribute the Beta version of the application to the selected WPI students, faculties and staffs</td>
<td>Get feedbacks</td>
</tr>
</tbody>
</table>
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

- 1/7 drivers admit this fact: [https://www.cars.com/articles/parking-in-these-cities-is-the-worst-1420697595754/](https://www.cars.com/articles/parking-in-these-cities-is-the-worst-1420697595754/)
- Android GeoFence Documentation: [https://developer.android.com/training/location/geofencing](https://developer.android.com/training/location/geofencing)
- Dialog flow: [https://dialogflow.com/](https://dialogflow.com/)
Thank You

Q & A