"BikeSmart" Project Requirements

This document describes the requirements, use cases, technologies employed, and system architecture of the "Bikesmart" Project - designed and implemented by the Treadsetters team.

Treadsetters
Saili Raje, Lead
Joel Dick
Chris Karcher
Duncan Sommer
Oliver Townsend

Revision History

<table>
<thead>
<tr>
<th>Version Number</th>
<th>Primary Author(s)</th>
<th>Version Description</th>
<th>Date Completed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>Saili Raje, Joel Dick, Duncan Sommer, Oliver Townsend, Chris Karcher</td>
<td>Initial Version with user stories</td>
<td>1/23/15</td>
</tr>
<tr>
<td>1.1</td>
<td>Saili Raje, Joel Dick, Duncan Sommer, Oliver Townsend, Chris Karcher</td>
<td>Added System Architecture and Glossary</td>
<td>1/29/15</td>
</tr>
</tbody>
</table>

Table of Contents

- Introduction
- Glossary of Terms
- Revision History
- System Architecture Overview
- Requirements (functional and non-functional)
  - At least 10 use cases or user stories
- Prototyping code and test cases (Github URL)
- System models – N/A
- Appendices – Technologies employed
Introduction

There is no existing platform that allows a bicycle to communicate with other devices via the Internet. Such a system will need to solve power and connectivity issues related to an embedded system on a bike. Our team is designing BikeSmart to enable developers to create profitable applications for users ranging from the casual commuter to the professional cyclist. This project will also act as a proof of concept for further integration of bikes into the IoT.

Glossary of Terms

**Embedded System** - an embedded system is a computer system with a dedicated function within a larger mechanical system, such as a bicycle

**Internet of Things (IoT)** - the interconnection of uniquely identifiable embedded computing devices within the existing Internet infrastructure.

**Parse** - a cloud based application engine that allows developers to receive and distribute information and messages to devices on the internet.

System architecture overview

This system will be comprised of:

- An Android service that runs on an embedded system simulated by a mobile phone that captures data from the bike, stores it locally, and sends it to a cloud database intermittently while minimizing power consumption.

- A backend database running on a remote server that will receive and process data, then distribute the data to remote clients.

- At least one specialized mobile application that will utilize the data provided by the database to deliver content to a Bikesmart user.

- A frontend web interface which gathers content from the remote server and presents it in a clear and organized manner to users.

Requirements (functional and non-functional)

As a user, I should be able to:

- login to an online service to view data generated by my bike
- associate my identity with a bike
- add other users to my bikes
- store my bike’s data (location, tire pressure, etc) in the cloud
- share my data with others
- control the privacy of my data
- access an intuitive graphic interface that presents collected data
- download applications that utilize collected data
- interact with the BikeSmart system from multiple platforms

As a developer, I should be able to:
- access multiple sources of data on the bike
- build applications on top of the BikeSmart platform
- design sensors and integrate them into the BikeSmart system

Prototyping code and test cases (Github URL)
https://github.com/sraje/CAPSTONE

System Models

Appendices/Technologies Used

- Parse Application Engine and API
- Google Location API
- Android SDK
- Pivotal Tracker
- Github