# "BikeSmart" Project Requirements Draft 2

This document describes the architecture, requirements, and design 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

| Version Number | Primary Author(s) | Version Description | Date Completed |
|---|---|---|---|
| 1.0 | Saili Raje, Joel Dick, Duncan Sommer, Oliver Townsend, Chris Karcher | Initial Version with user stories | 1/23/15 |
| 1.1 | Saili Raje, Joel Dick, Duncan Sommer, Oliver Townsend, Chris Karcher | Added system architecture and glossary | 1/29/15 |
| 1.2 | Saili Raje, Joel Dick, Duncan Sommer, Oliver Townsend, Chris Karcher | Added user stories and updated system architecture overview | 2/22/15 |
| 1.3 | Saili Raje, Joel Dick, Duncan Sommer, Oliver Townsend, Chris Karcher | Updated glossary. Added "bike circle" feature | 2/22/15 |
| 1.4 | Saili Raje, Joel Dick, Duncan Sommer, Oliver Townsend, Chris Karcher | Updated list of user stories and added wireframe mockups | 2/22/15 |
| 2.0 | Saili Raje, Joel Dick, Duncan Sommer, Oliver Townsend, Chris Karcher | Current Revision | 2/22/15 |

## Table of Contents

## 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, a database for mobile bike information 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 Internet of Things.

## 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. For this project, for prototyping purposes, we will be using a Motorola Moto G smartphone.

**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.

**Bike Circle** - a group that contains users and bikes. If you join a bike circle, or approve /someone to join a circle, you/that person automatically gets permission to request to use your bike and gets access to your bikes information (location, past rides, etc.)

## Requirements and user stories

| As a user... | Description | Time to Implement | Priority (0→8) |
|---|---|---|---|
| Login | login to online service to view data generated by bike | 2 hours | 8 |
| Identity | associate my identity with a bike | 1 hour | 8 |
| Add friends | add other users to my bike circles | 4 hours | 4 |
| Request bike | request bike from another user | 3 hours | 4 |
| Store bike's data | store my bike's data (location, tire pressure, etc.) in cloud | 4 hours | 8 |
| Share data | share my data with others | 4 hours | 4 |
| Control privacy | control the privacy of my data and who gets to see/use it | 4 hours | 4 |
| Graphic interface | access an intuitive graphic interface that presents collected data in easy to understand manner | 8 hours | 2 |
| Download more apps | download more applications that can be used with my bike's data | 4-12 hours (depending how many apps we choose to make) | 4 |
| Cross-platform | interact with BikeSmart system from multiple platforms (Android, iOS, Web, desktop) | 10-40 hours (Depending how many platforms we choose to develop on) | 2 |

| As a developer... | Description | Time to Implement | Priority (0→8) |
|---|---|---|---|
| Access bike data | access multiple sources of data on the bike, make sure can receive info and readings from various sensors | 4 hours | 8 |
| Build additional apps | build applications on top of BikeSmart platform using Parse database | 10-40 hours depending | 4 |
| Design/integrate sensors | be able to design and add new sensors to be used with the BikeSmart platform | 5-15 hours | 4-8 |

| As a group admin... | Description | Time to Implement | Priority (0→8) |
|---|---|---|---|
| Approve/add users | be able to approve and add people to bike circles | 4 hours | 4 |
| Remove users | be able to remove users from bike circle | 2 hours | 4 |
| Change circle privacy settings | modify data permissions for users in/outside the bike circle | 10 hours | 1 |
| Add administrators | promote other members of bike circle to administrator status | 5 hours | 2 |

| As a Bike Owner... | Description | Time to Implement | Priority (0→8) |
|---|---|---|---|
| Add a bike to my account | functionality that allows bike owner to add a bike to their bike circle | 4 hours | 8 |
| Set a bike in your bike list as the default bike | set a bike as default, should show up as the first bike in your list | 2 hours | 2 |
| Change your bike's status as public or private | if user owns the bike, they can change its status as public or private, if it's public, people that the bike has been shared with can view its data/location if it's private, they can't | 2 hours | 5 |
| Add/approve other users to my bike | be able to approve and add people to bike circles | 4 hours | 4 |
| View data generated by my bike | users should see bike data displayed on the "dashboard" | 4 hours | 8 |
| Control the privacy of data associated with bikes I own | Change what people you've shared your bike with can see (only location, no location, etc.) | 8 hours | 4 |
| View data from bikes shared with me | if a bike has been shared with me (and is public) the data generated by it should be displayed in my dashboard | 4 hours | 4 |
| Be notified if my bike is being moved by someone who is not me | if a bike's location changes and it's a certain distance away from me (meaning that I'm not the one moving | 8 hours | 8 |

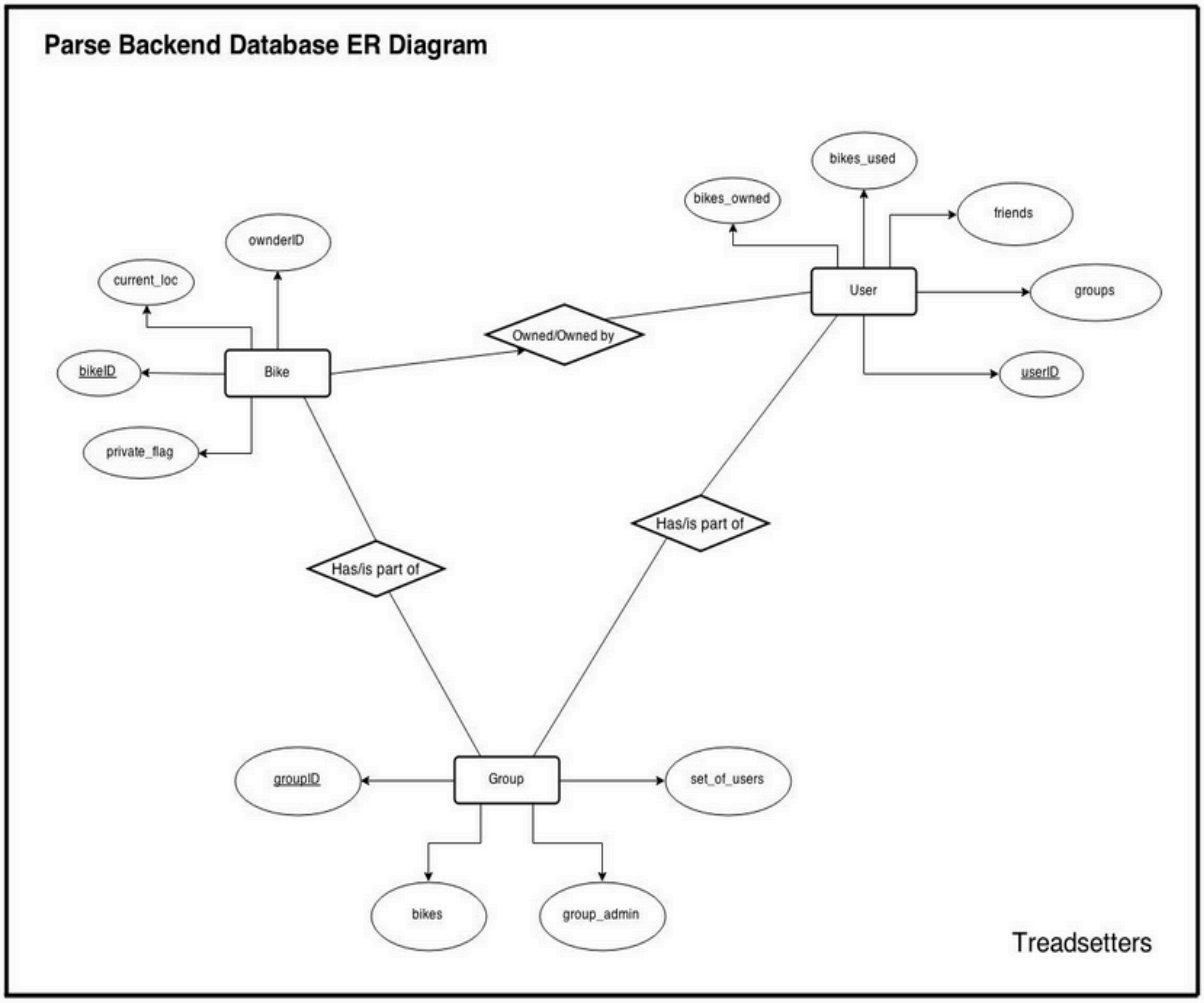
| | | | |
|---|---|---|---|
| | it) I should be notified | | |
| Enable/disable security systems for my bike | Turn on/off bike theft feature | 2 hours | 6 |
| Be notified if my bike's tire pressure falls below a threshold | Utilize bluetooth sensor data to trigger certain events/notifications. | 2 hours | 3 |

| **As a bike...** | **Description** | **Time to Implement** | **Priority (0→8)** |
|---|---|---|---|
| Collect data | collect location data from built in GPS | 2 hours | 8 |
| Connect to Bluetooth sensors | connect to Bluetooth sensors | 2-4 hours | 4 |
| Collect data from Bluetooth sensors | collect data from Bluetooth sensors, process if necessary | 4 hours | 4 |
| Push to cloud | push data from sensors to cloud | 2 hours | 4 |
| Pair with lock | pair with an electronic lock to control physical access to bike | 5 hours | 1 |

## System architecture overview

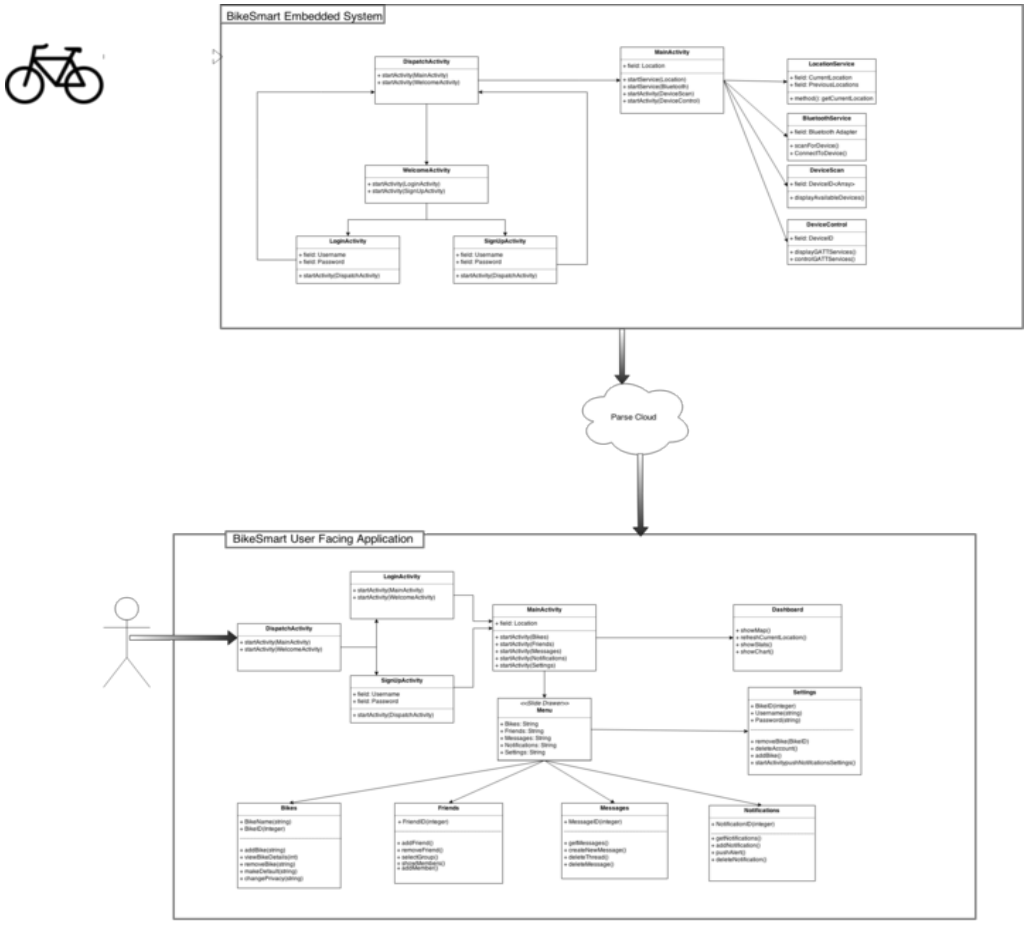
This system will be comprised of:

- 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.

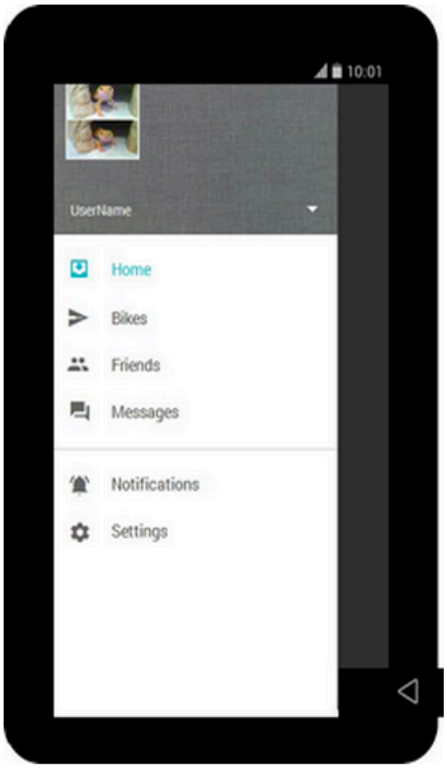

Parse Backend Database ER Diagram
bikes_used
bikes_owned
friends
ownderID
current_loc
User
groups
Owned/Owned by
bikeID
Bike
userID
private_flag
Has/is part of
Has/is part of
groupID
Group
set_of_users
bikes
group_admin
Treadsetters

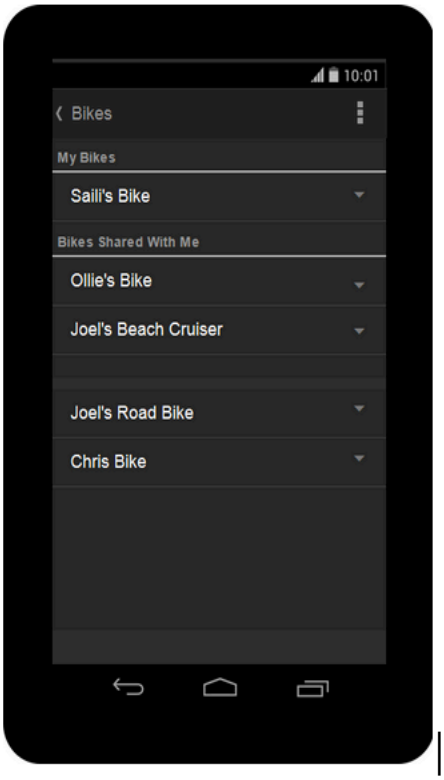
## UML Diagram

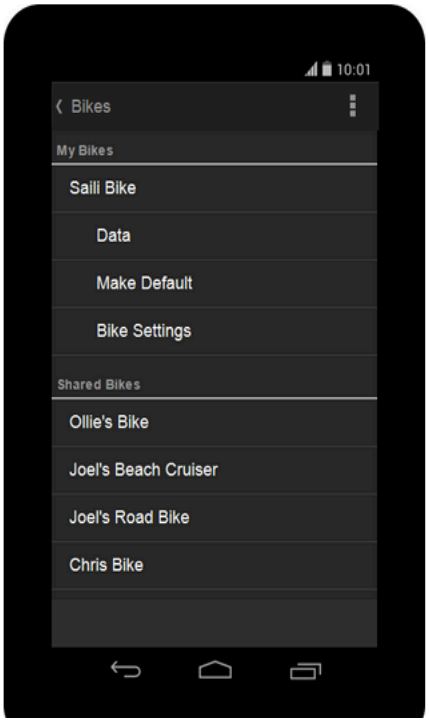
This diagram is a visual representation of the bikesmart system. It includes activities, methods, components, and intra-system relationships.

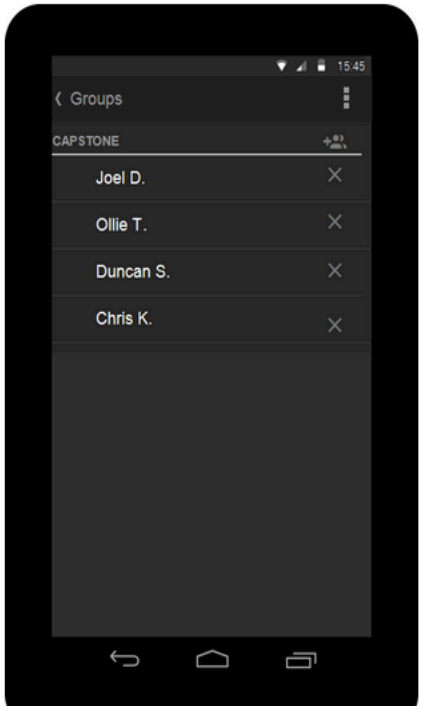


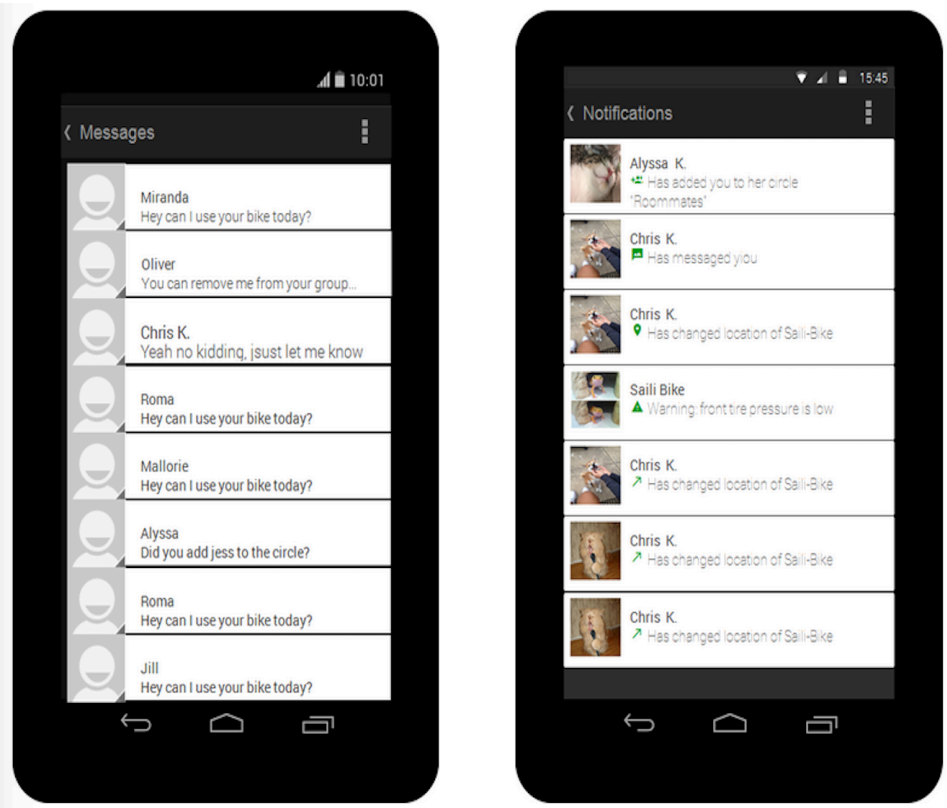
## Mockups/Wireframes






## Potential Obstacles

A foreseeable obstacle is how to standardize human and bike activity. Ideally, we'll be pushing almost any kind of data imaginable to users' cloud accounts and maybe even from bike to bike. Bike to bike communication is certainly going to be possible in the near future with Qualcomm's LTE direct technology on the horizon. Assuming other smart bike competitor companies will begin to emerge, it will be useful to have our bike be able to interact with a bike or bike database built by another company. How would we ensure stable and reliable communication between two or more bikes/bike networks without any form of standardization in data communication?

Another obstacle, of course, is privacy. At a high level, all connections made with the Parse database are made with HTTPS and SSL and it will reject all non-HTTPS connections. This completely absolves any need to worry about man in the middle attacks. At a lower level, we will be taking advantage of Parse's Access Control Lists (ACLs). ACLs allow developers to control who can access which sets of data through these lists. Lists contain objects and each object has a list of users and roles including what permissions that user or role has.

Finally, another hurdle we might not be able to tackle very elegantly is the inclusion of a physical lock. For the purposes of the project, we will most likely just have a boolean variable that controls whether or not one can unlock/use the phone screen that's attached to the bike. The bike will have to

be unlocked using the user app through a secure login of the owner of that particular bike.

## Prototyping code and test cases (Source Code)

https://github.com/sraje/CAPSTONE

## Appendices/Technologies Used

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