



ParkingIQ™



**CSCI 5673: DISTRIBUTED SYSTEMS
SPRING 2011**

**ADITYA SAWHNEY
AKASH AGRAWAL
MURALIKRISHNA NALLAMOTHU
SRINIVAS PANCHAPAKESAN**

[HTTP://CODE.GOOGLE.COM/P/PARKING-IQ/](http://code.google.com/p/parking-iq/)

Background



**WHAT IS PARKINGIQ?
MOTIVATION
CU PARKING SERVICE
PILOT PROJECT**

Motivation



What is ParkingIQ™?

An intelligent Parking Management System which serves the state as well as the public and automates various aspects of public parking thereby enhancing efficiency and transparency.

Motivation

- Develop a distributed system with real world application.
- Efficient parking solution – Customers and Enforcement agencies
- Avoid parking violations !!
- Comprehensive automation and integration
- First prototype – plan to implement at PTS @CU Boulder

Slide 3

AA2

Akash Agrawal, 5/1/2011

CU parking services



Present parking system – PTS @CU

- No available application or service for clients to locate parking spots
- Available parking lot spaces are determined by manual counting for maintaining the database
- Lot of multi-vendor systems which are not integrated as a result trending/planning entails manually aggregating reports from multiple systems

How can ParkingIQ™ help?

- Customers
 - Locate available parking spots easily using smart-phone device
 - Pay online – safe as no credit card required
 - Remotely add time
 - Refund for unused time
- Enforcement agencies [PTS]
 - Increased efficiency (cost and time) as manual steps are removed
 - Automated detection of violators and issuing of citations
 - Report generation - used for planning/event management
 - Dynamically vary cost based on peak hours/demand and supply
 - Analysis for future logistical upgrades

Pilot Project



Goal

Provide following services using “real” data from the Parking Services of CU:

Locate parking lots with available space (customers)

Using the current GPS location of an Android phone, display

- all the parking lots on a map
- corresponding lot id and available spaces

Also, provide ability to specify a particular location.

Generate Quarterly Report (PTS)

For given quarter and parking lot, generate a graphical report which displays the average space availability from Monday-Friday for 3 sessions a day – Morning, Afternoon & Evening.



Architecture



DESIGN GOALS
HIGH LEVEL ARCHITECTURE
BUSINESS LAYER DESIGN
DATA LAYER DESIGN

Design Goals



- **Elastic Scalability**
 - Infrastructure – Amazon EC2
 - Data store – Dynamo/Big Table
 - Computation – Hadoop/MapReduce
- **High Availability**
 - Failover resistant - Replication
 - Always writable
- **Eventual Consistency**
 - Its OK if sometimes we report incorrect space availability
- **Open Source components**
 - Students are broke !!

Architecture

<http://piq.com/parkinglots/?latitude=40.1056&longitude=-105.435>

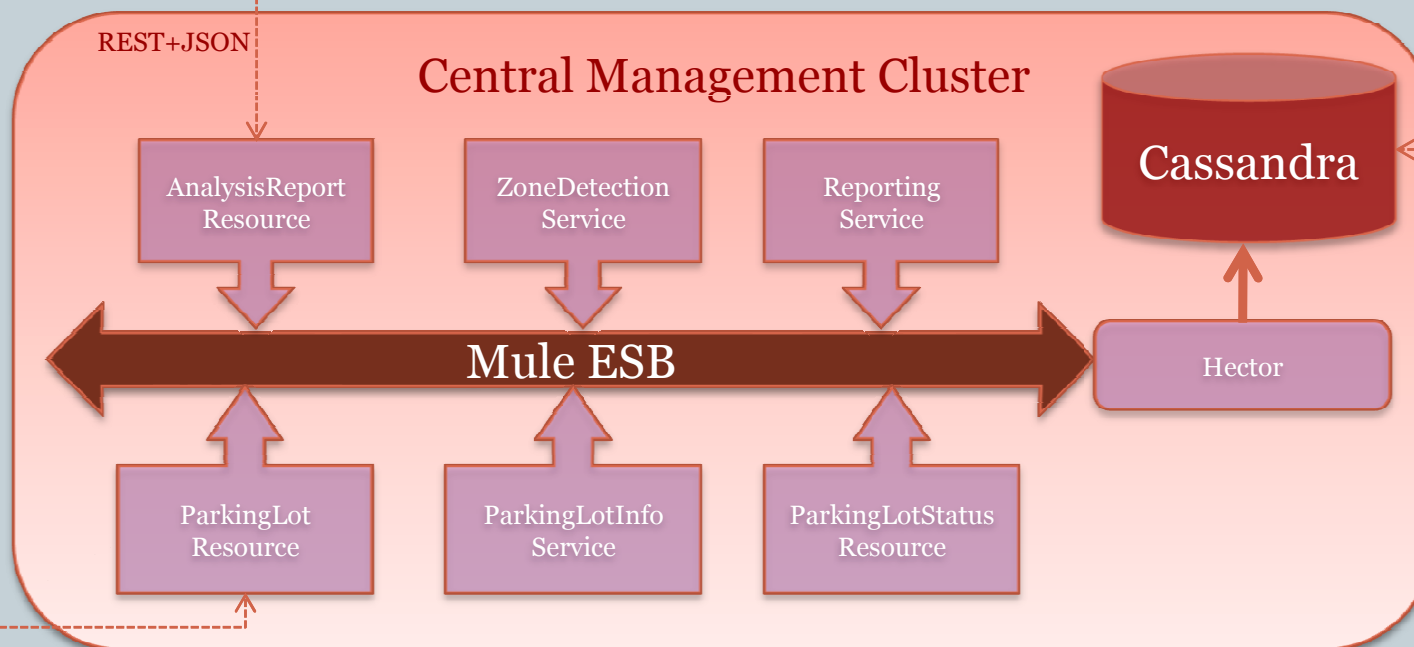


http://piq.com/reports/quarterly/201/?quarter=fall_2010

WebApp

ParkingLots

Periodically update the available space count



Business Layer Design



- **Mule ESB**
 - Used as application server (HTTP) and integration platform
 - Pluggable components
 - ✦ Jersey – JAX-RS implementation for REST
 - ✦ Jackson – JSON processor
 - Scalable and decoupled
- **Hector – client libraries for Cassandra**
- **REST resource components**
 - ParkingLotResource – GET parking lot details
 - AnalysisReportResource – GET quarterly analysis report

Location Parking Lots Flow



```
/**
 * Represents the REST resource for parking lots.
 */
@Path("/parkinglots")
public class ParkingLotResource {

    /**
     * Get parking lot information for given location.
     *
     * @param latitude the latitude of location
     * @param longitude the longitude of location
     * @return the list of parking lots which are close to the given location
     */
    @GET
    @Produces("application/json")
    public List<ParkingLotInfo> getParkingLots(
        @DefaultValue("0.0") @QueryParam(Const.Param.LATITUDE) float latitude,
        @DefaultValue("0.0") @QueryParam(Const.Param.LONGITUDE) float longitude) {
        GeoPoint location = new GeoPoint(latitude, longitude);

        // Figure out the zone to which the given belongs
        Zone zone = this.zoneDetectionService.identifyZone(location);

        // Get all the parking lots which lie in the zone
        List<ParkingLotInfo> parkingLots = this.parkingLotInfoService.getParkingLotInfo(zone);

        // Get the current status (available spaces) for each of those lots
        this.parkingLotStatusService.updateParkingLotStatus(parkingLots);

        return parkingLots;
    }
}
```

Data Layer Design



Data store – Apache Cassandra

- Open source and developed by Facebook
- Amalgamation of Dynamo (implementation) and Big Table (data model)
- Satisfies all the design goals of availability, consistency etc.
- Works on Amazon EC2 and integrates with Hadoop Map Reduce.

Data store requirements

- Parking lot 'static' information (location, id, zone, type)
 - **ParkingLot** CF
- Parking lot 'dynamic' information (available space count)
 - **LotStatusArchive** CF
- Quarterly report data → **QuarterlyAnalysis** SCF
 - ✦ generated periodically by running MapReduce on LotStatusArchive

Table 1: Parking Lot Information

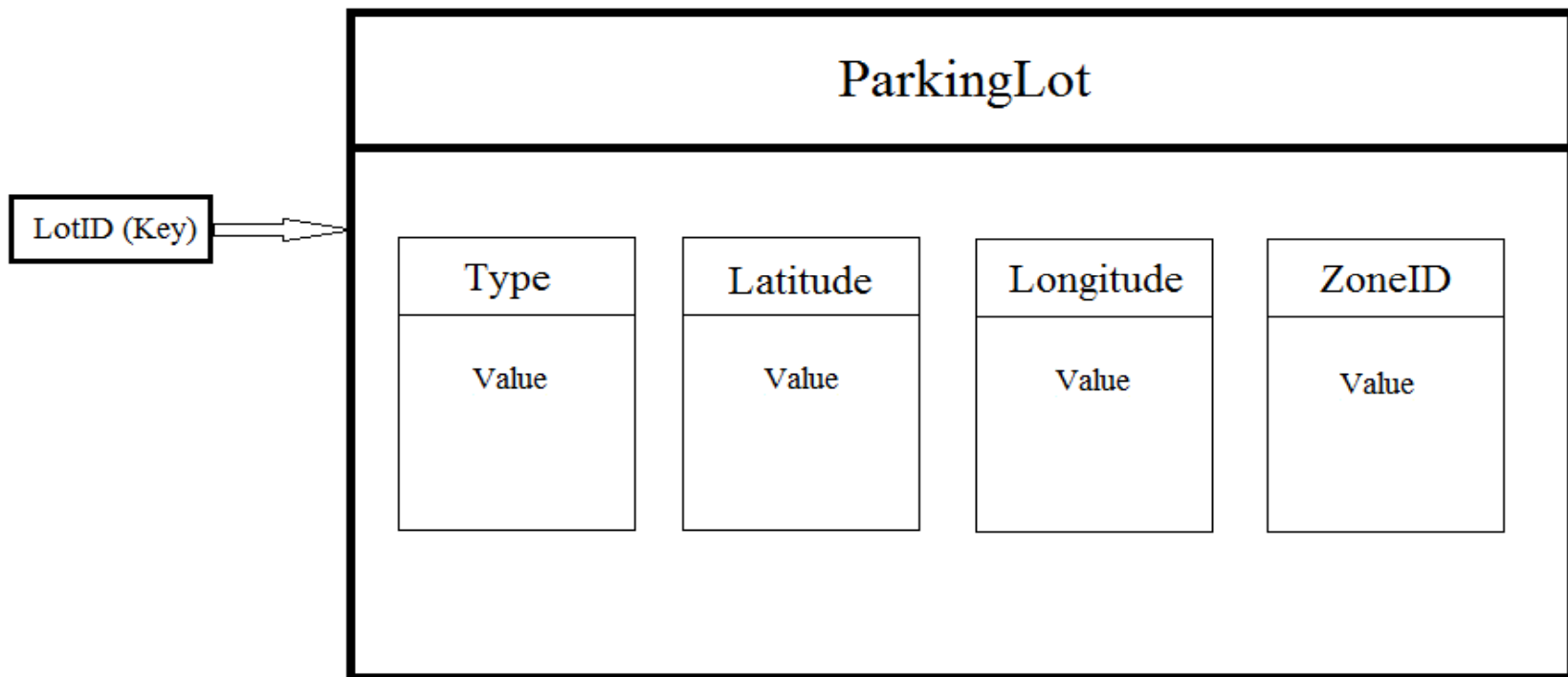


Table 2: Lot Status

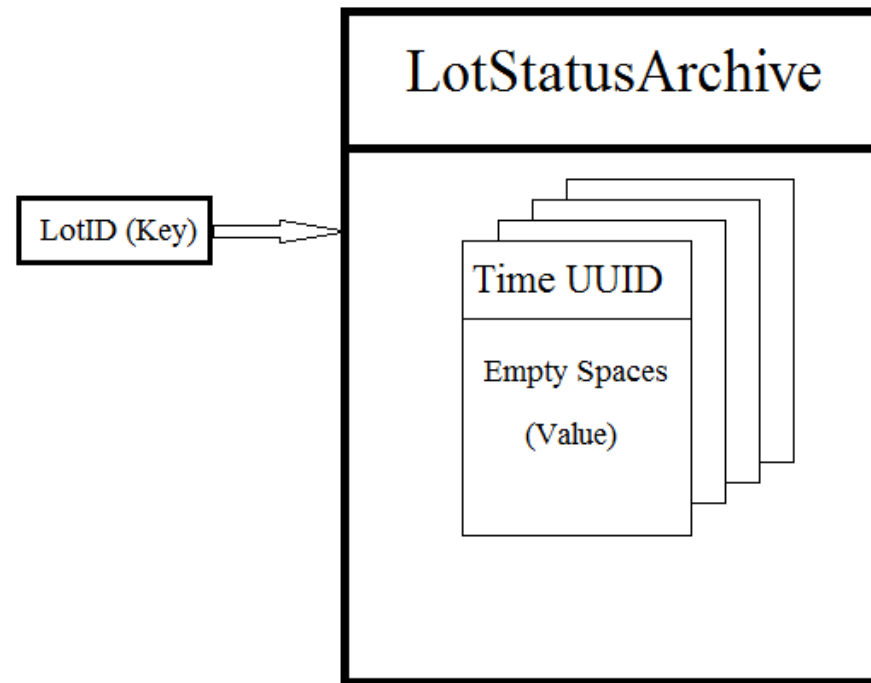
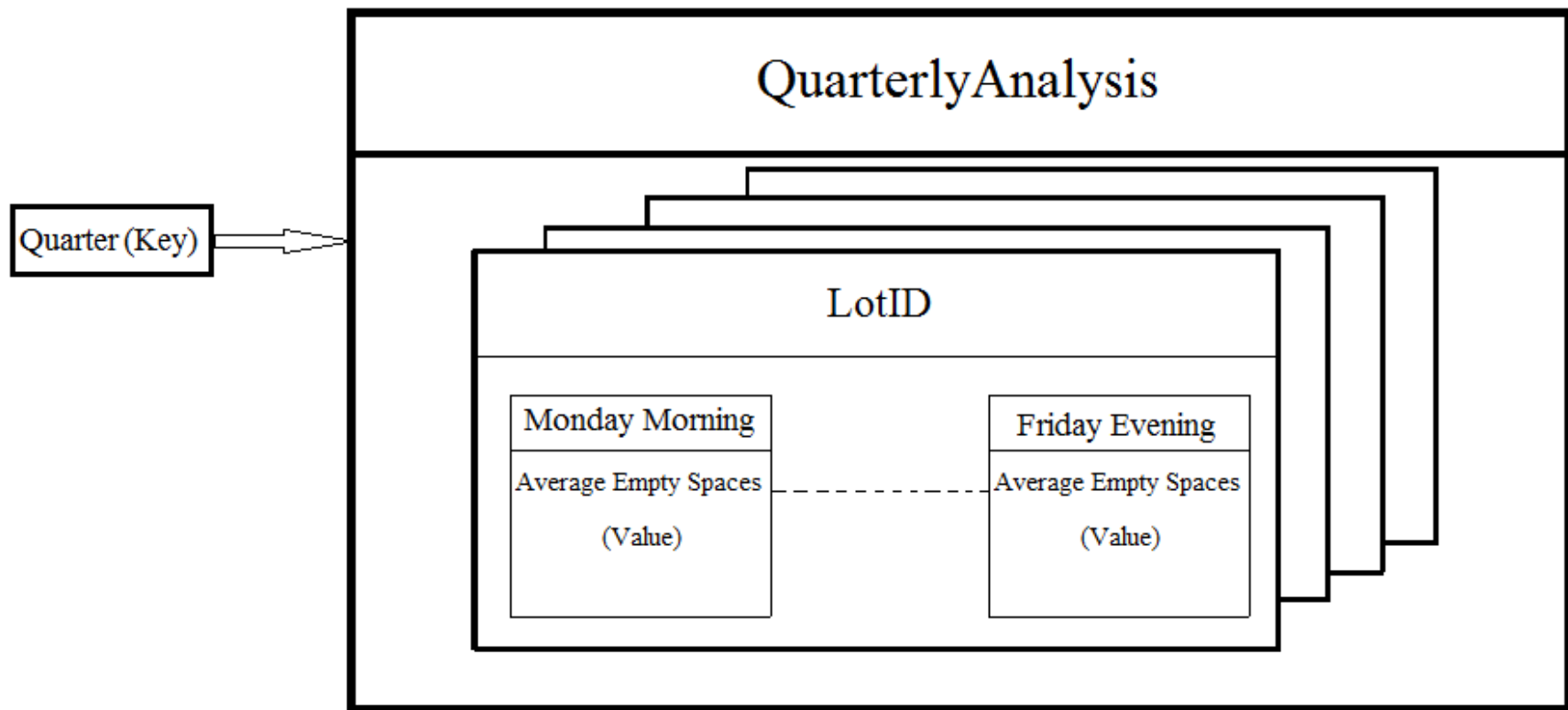


Table 3: Quarterly Analysis



Demo



QUARTERLY REPORT GENERATOR

Quarterly Report



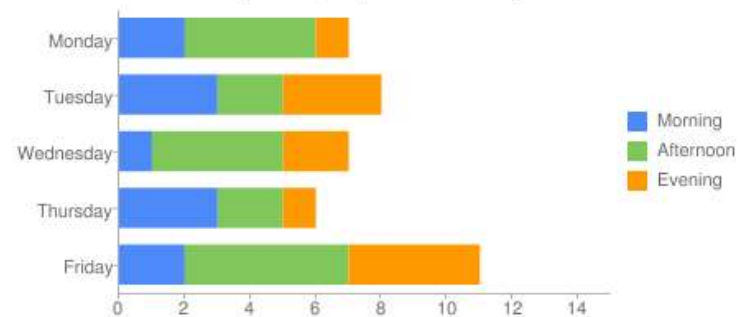
ParkingIQ™ Quarterly Report

Quarter: Fall 2010 ▾

Parking Lot: 201 ▾



Daily average space availability

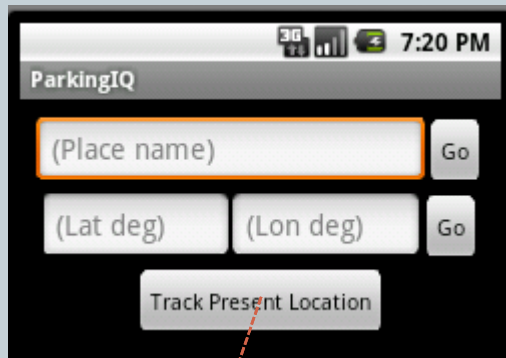


Android Application



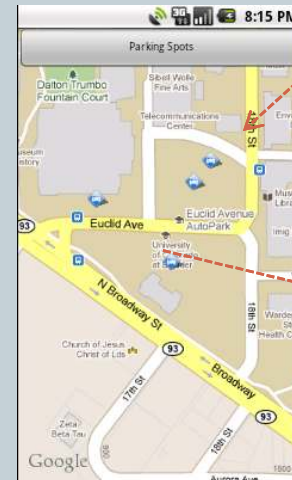
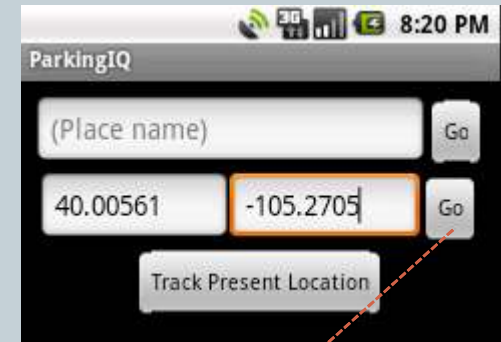
OVERVIEW
IMPLEMENTATION DETAILS

Overview



Permissions:
INTERNET
FINE_LOCATION
COARSE_LOCATION
GOOGLE MAPS

Modules:
GPS Services
MapActivity
Google Maps API
Android.location
AsyncTask
DDMS !



Implementation Details



- **MapActivity**
- **GPS Services**
 - To obtain current location of Customer requesting parking lot data
 - `Android.location`
- **AsyncTask**
 - GPS service time-out
 - Included Button
- **DDMS !**

Demo



ANDROID APPLICATION

Future work



- Integrate into PTS@CU – Real-time database
- Customers
 - Parking lot booking service – credit card
 - Android client - Better UI
- Enforcement agencies [PTS]
 - Parking violation detection tool
 - Multiple servers
 - Load balancers