VDS India Hackathon 2015

Team : Code Avengers

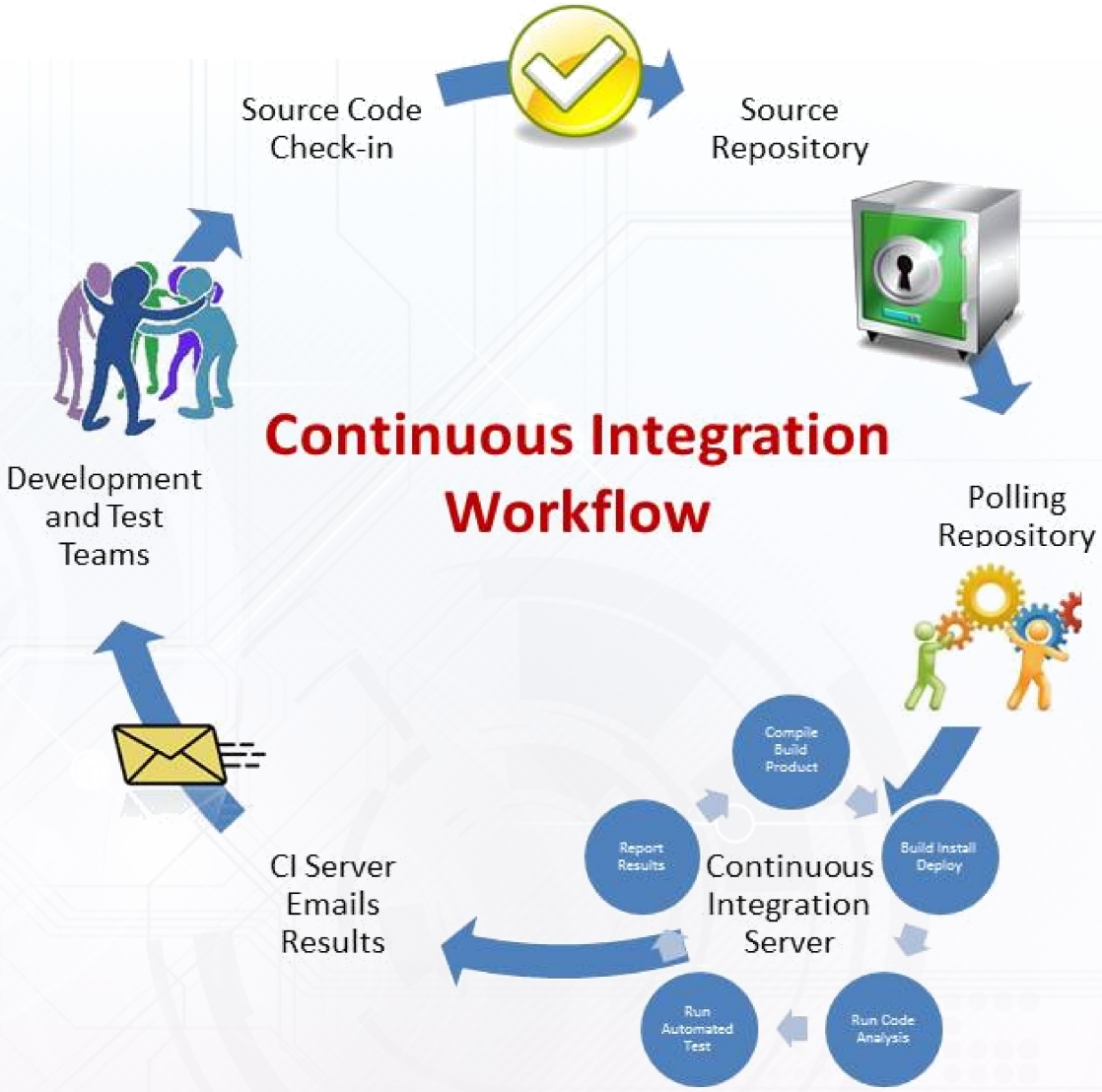
Continuous Integration (Vision)

1. **Deliverables:**
    a. Source code
    b. Unit Test Case

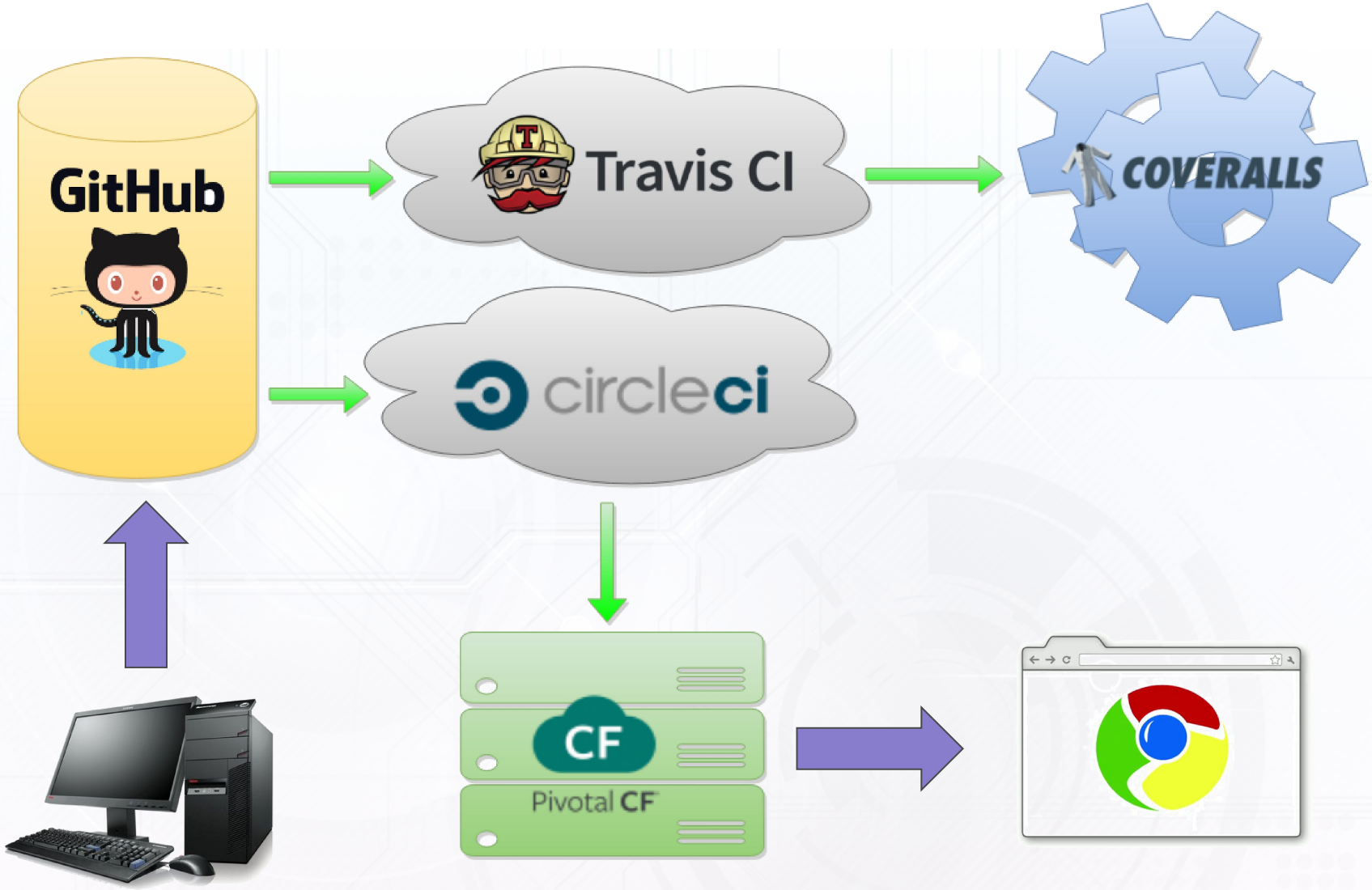
2. **Repository:**
    a. Github

3. **Constinuous Integration:**
    a. Build
    b. Test Execution
    c. Code Metrics
    d. Deployment

4. **Monitoring:**
    a. Email
    b. Webhooks



Continuous Integration（Architecture）



Continuous Integration (Links)

Repository:

- Github - https://github.com/aroychoudhury/VerizonHackathon2015

Continuous Integration:

- Travis CI - https://travis-ci.org/aroychoudhury/VerizonHackathon2015
- Circle CI - https://circleci.com/gh/aroychoudhury/VerizonHackathon2015

Deployment:

- Pivotal CF - https://console.run.pivotal.io

Metrics:

- Coveralls - https://coveralls.io/github/aroychoudhury/VerizonHackathon2015

Travis CI

circleci

COVERALLS

Tools

Image Manipulation:

- Pixlr - https://pixlr.com/editor/
- Image Optimizer - http://www.imageoptimizer.net/Pages/Home.aspx

HTML Editing:

- Best Online HTML Editor - http://bestonlinehtmleditor.com/
- Circle CI - https://circleci.com/gh/aroychoudhury/VerizonHackathon2015

DB Diagram:

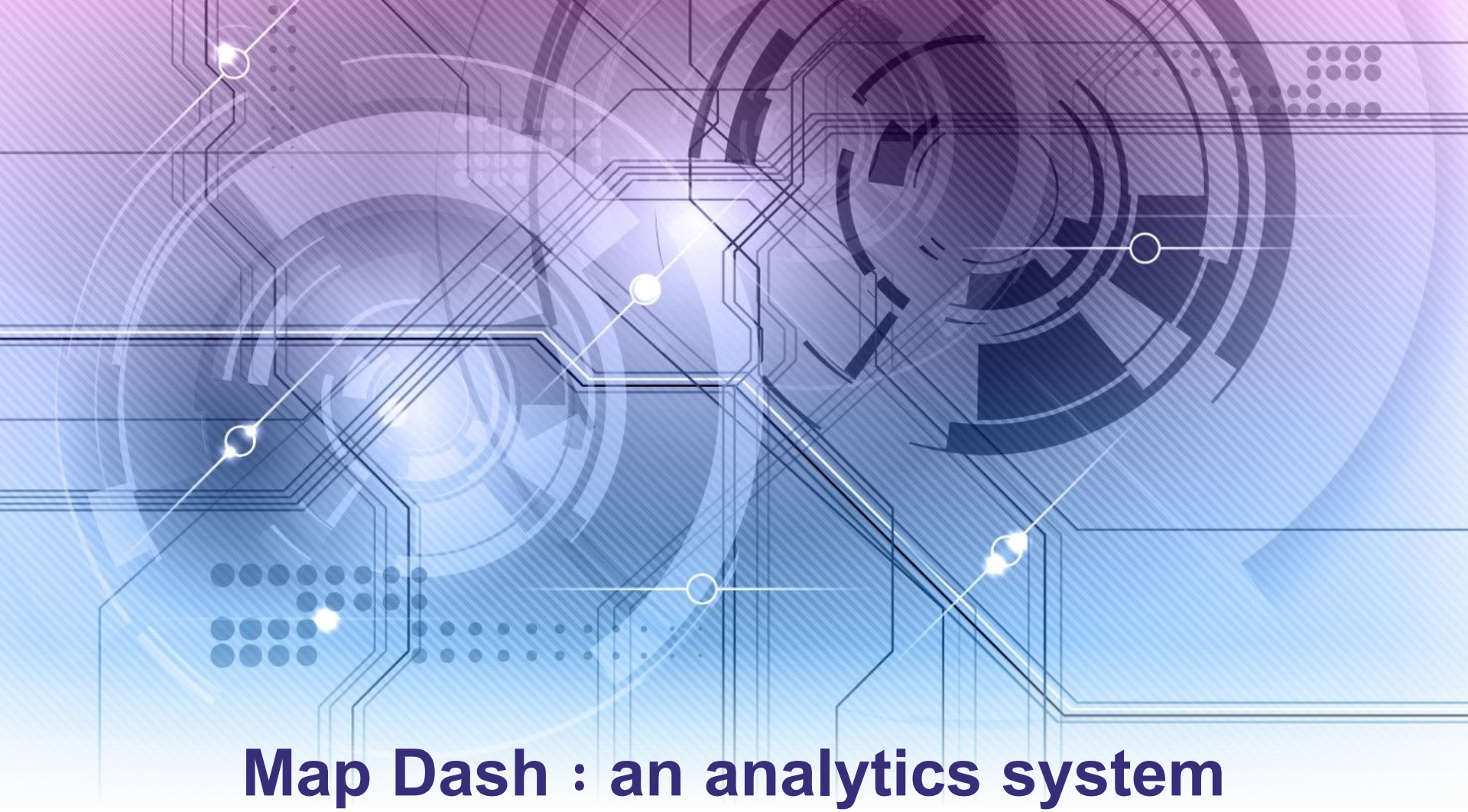
- Vertabelo - https://www.vertabelo.com/

Diagrams:

- Draw IO - http://draw.io

Code Metrics:

- Google Code Pro - Eclispe Plugin



Map Dash : an analytics system

A Code Avengers Initiative

Map Dash

Map Dash (*Map Analytics Dashboard*) is an attempt to add Geographic Context to data analytics. This is an attempt to provide more insight to data such as - *Population base across Cities*, *Accidents within Urban & Rural Areas*, *Customer Spread across Regions* etc.

The below sections provide more information of MapDash system.

1. **Features:**
    a. Data integration with Google Maps
    b. Charting capability on the Geographic data
    c. Mobile first UI design and implementation (Bootstrap/jQuery)
    d. Robust Spring Driven Backend Architecture
    e. jUnit Based automated Unit Testing capabilities [via Continuous Build]
    f. Code Metrics Generation capabilities [via Continuous Build]

2. **Best Practices:**
    a. Architecture and Database Schema design diagrams
    b. Java Docs for all Public and Protected methods
    c. Metrics Data generated as part of Development process

Architecture Design

Spring Components used:

- JDBC Template (Spring JDBC)
- Rest Controller (Spring MVC)
- Java Config (Spring Core)
- Bean Injection (Spring Core)
- Transaction Management (Spring AOP)
- Unit Testing (Spring Test)

Logging:

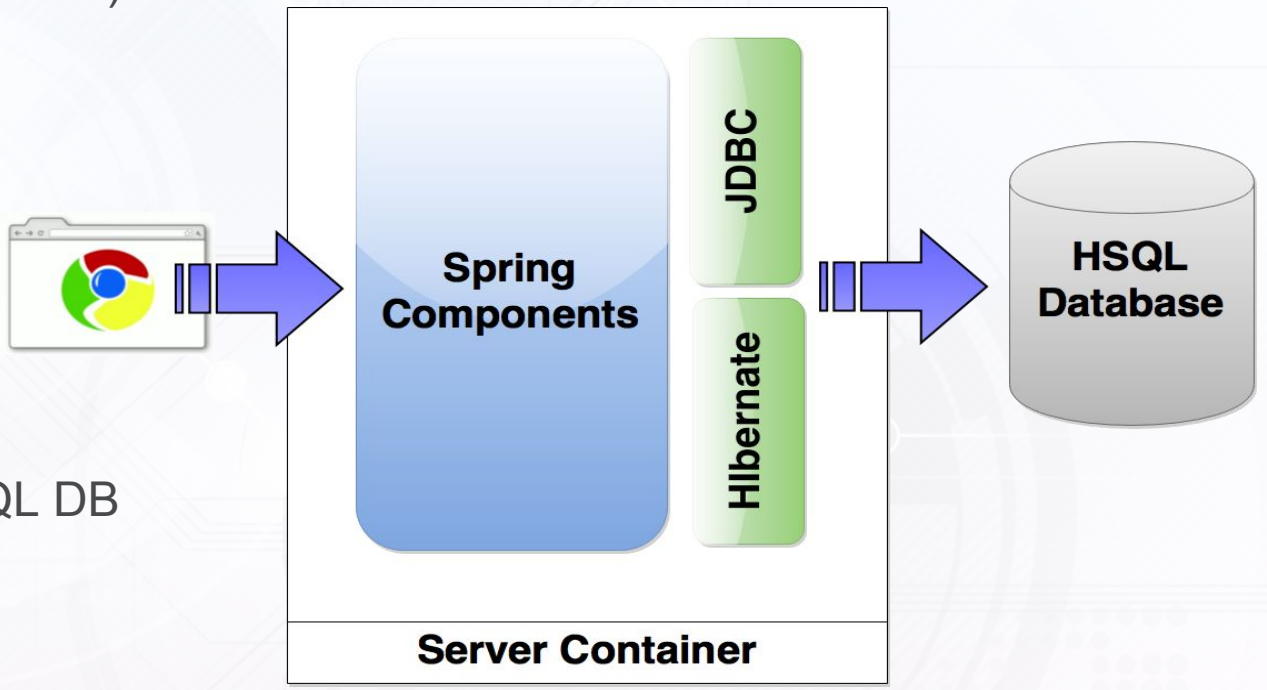
- SLF4j - Log4j

Hibernate:

- Hibernate 4

Database:

1. In memory HyperSQL DB



Database Design

LOCATION_MASTER:

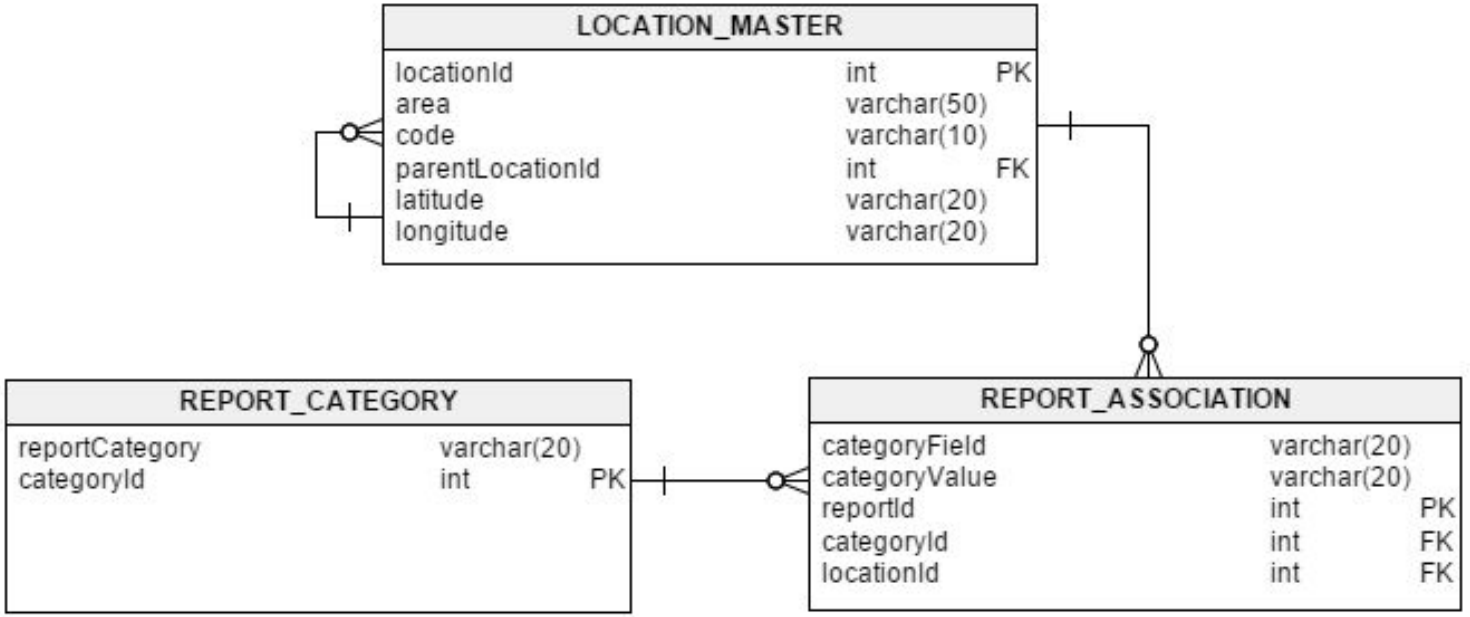
- Location data and details

REPORT_CATEGORY:

- Data Categories and other details

REPORT_ASSOCIATION:

- Associations between Locations & Categories



Looking Ahead

Looking ahead, we see a lot more that we can do with this application.

1. **More Features:**
    a. Integration with Enterprise Data Processing systems
    b. Creating plugins for common Databases/Big Data Systems
    c. Add dynamic data uploading functionality
    d. Allow greater interactivity to the user leveraging Google Maps plugin
    e. Create HTML widgets (similar to MakrMyTrip widgets we see on webpages)

2. **Achieve More Best Practices:**
    a. 85% Test Coverage
    b. Greater Code/Comment Ratio
    c. Introduce a MVC Framework for future UI work

Map Dash (Links)

Application:

- http://codeavengerz.cfapps.io/maps.html

Java Docs:

- http://codeavengerz.cfapps.io/docs/index.html

Metrics:

- http://codeavengerz.cfapps.io/metrics/metrics.html

Team Blog:

- http://codeavengerz.cfapps.io

Thank you !!!!!