Date: 6/18/2015 at 9.30 AM EST

|  |  |  |
| --- | --- | --- |
| Bangalore Shivacharan – CEO | David Hong – product manager | Nayanjeet Medhi |
| Jaikumar Madhava | Satya Govindu | Jaychand |
| Pradeep KTR | Pradeep DV | Akshay |
| Sridhar | Jagan | Sajil |

1.       Git hub organization, private repository is created

Link: <https://github.com/TPRockville/jDerive> Organization name: TPRockville

Everyone has been invited to join. Go ahead and contribute

2.       AWS URL has been created and mongoDB instance installed over there

3.       Team loaded a set of event data in MongoDB instance on 192.168.20.50 server

4.       Rockville team discussed and came up with a use case that analyzes the spike in a custom-graph (JC will provide more details via a wireframe) and presents the data in a user-readable format on the screen

1. UI design – We’ll have a call discussing the use case scenario and come up with a  mockup
   1. My suggestion, as we thought yesterday, is to use a front-end MVC and make REST connection to the real-time processing engine to generate reports.
2. Back-end work to begin
   * + We need to start writing RESTful API (entry point) into the application
     + There are two processes that need to be handled –
     + 1. Real-time data processing

Real-time data processer uses the processed data (mongoDB?) and also the user input to derive the report that needs to be presented to the user.

* + - 2. The batch data processing

The Batch processing code regularly looks for the event and enforcement reports and processes the feed and integrates into the local database.

* + - We need to quickly create a data model that we shall be using to save the data. (This will be based on the structure of the feed + categorized for our convenience). I thought a little about it today, but could not conclude on how the data could be organized. So, we can derive this after we go over the use case with JC.

1. We need not worry about login/user management etc. for this project at this point.
2. Deployment plan –
   1. Let’s list out the technical stack at the earliest so that we can come up with a deployment plan for this project. We need your inputs for this to happen.
3. QA/Automation
   * + As Pradeep TR mentioned, we can either choose Junit or Cucumber to test our back end code
     + Front-end testing, I am thinking if it needs any automation to start with, we’ll talk about it sometime today/tomorrow
4. Everyone, please start using the private GitHub, temporarily named jDerive (we can rename it based on some consensus).