Code Quality

— Ganesh Arshavin

Task 1

Rough estimates on when the product can be launched together with milestones that the product team could measure progress with

Assumptions :

- Each sprint is 2 weeks long .
- POC of Service 1 is done , but no documentation .
- RFC phase of a service involves the flattening of functional requirements .
- Each sprint to be preceded by backlog grooming and sprint planning and succeeded by demo of the features to product team + feedback loop for next sprint if needed
    - Backlog grooming : An overview of what is going to be implemented that sprint , in terms of tasks
    - Sprint Planning : Developers estimate the tasks and we figure how we can schedule the sprint .
- Each service must have the following steps to consider it complete
    - RFC - For the service
    - Tasks/Features Including UI intergration
    - Testing
    - Bug fixes
    - Demo
    - Acting on feedback or improvements
        - Subsequent demos and approvals

    - Deployment
    - Monitoring
- Time after phase 3 (Refer PDF) is allocated for Scaling changes if necessary for indudivual services

Please find the PDF attached for the Timeline , Data is represented via Gnatt Chart

A diagram showing how the core services should interact with each other

Assumptions :

- We are using Micro service architecture for our services .
- Services will communicate to other services either in HTTP or GRPC
- The component diagram will only try to tackle the interaction of services .
- Metrics , Scaling will not be handled here (Will be handled in Infrastructure diagram)
- Will be using consul for service registration , discovery , Key value (config management .
- We will be using HAProxy loadbalancer with Consul for load balancing
- Each micro service will have the following structure in a docker
    - Main service (Healthcheck route)
    - Process to periodically fetch config from consul and reload service
    - Workers if any .
- Each node will have a consul agent to register itself on the consul service registry , the service.json file for the node will be populated in cloud init .

Please refer the PDF in the folder `core-services` component diagram.

A diagram showing how all services in a production environment interact with each other to fulfil a user's request.

Assumptions :

- We do not have a mobile app .
- We use external services for push/email notification .
- We use BQ for archiving data