Sprint 0 (24-Nov-2023 to 30-Nov-2023)

Pod Division

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Azure | |  | Azure | |  | GCP | |  | GCP | |  | GCP | |
| **Name** | **POD** |  | **Name** | **POD** |  | **Name** | **POD** |  | **Name** | **POD** |  | **Name** | **POD** |
| Kethavath Jagadeesh | 1 |  | Vaditya Moksha Vinayak | 2 |  | Mukul Verma | 3 |  | Noel Vincent Polakallu | 4 |  | Modalavalasa Jayanth | 5 |
| Nandela Sri Gowri Niharika | 1 |  | Boyapati Hritikesh | 2 |  | Sathiya Seelan Sivaraj | 3 |  | Kanakala Durga Prasanth | 4 |  | Kondampally Aakash Reddy | 5 |
| Rohith Kumar Reddy Tumati | 1 |  | Dhruv Rajesh Deshmukh | 2 |  | Mothe Rohith | 3 |  | Vemula Rithik Prasad | 4 |  | Rahul Manoj Pathak | 5 |
|  |  |  |  |  |  | Hitarth Arora | 3 |  | Sachin Naik | 4 |  | Vinay Chennamolla | 5 |

User Stories

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Sprint - 0** | |
| **Name** | **POD** | **User Story Number** | **Role** |
| Kethavath Jagadeesh | 1 | 11 | Backend |
| Nandela Sri Gowri Niharika | 1 | 2 | Devops |
| Rohith Kumar Reddy Tumati | 1 | 10 | Design patterns-UI |
| Vaditya Moksha Vinayak | 2 | 3 | Devops |
| Boyapati Hritikesh | 2 | 8 | UX |
| Dhruv Rajesh Deshmukh | 2 | 1 | Devops |
| Mukul Verma | 3 | 5 | Devops |
| Sathiya Seelan Sivaraj | 3 | 1 | Devops |
| Mothe Rohith | 3 | 2 | Devops |
| Hitarth Arora | 3 | 4 | DB |
| Noel Vincent Polakallu | 4 | 4 | DB |
| Kanakala Durga Prasanth | 4 | 3 | Devops |
| Vemula Rithik Prasad | 4 | 12 | UI |
| Sachin Naik | 4 | 10 | Design patterns-UI |
| Modalavalasa Jayanth | 5 | 7 | Backend |
| Kondampally Aakash Reddy | 5 | 8 | UX |
| Rahul Manoj Pathak | 5 | 11 | Backend |
| Vinay Chennamolla | 5 | 6 | UI |

1. Documentations & README.MD file in each repo (Both for FrontEnd & BackEnd)
2. Test Cases (TDD/BDD) with Coverage [Selenium, Cucumber, Mockito, Junit, JMeter] feature files, Test Runner & Sonar Coverage.
3. Swagger for API documentation
4. CI/CD Pipeline – Jenkins Config
5. Containerization & K8S [Docker-compose]
6. Front End (Only React & Redux)
7. Make sure that all the pods are using Java17 version only.
8. Create Mock screens with Figma
9. Configure Load Balancer, API Gateway, Cloud Discovery service
10. Logging using AOP
11. Configure ELK, Kafka, Prometheus & Grafana Monitoring
12. Search for Google Image Processing Tools, Scanning Tools to recognize the data from the uploaded documents like PAN, Aadhaar etc.,
13. Create Centralized setup document in Confluence which includes links to Azure, GCP, Bitbucket, Pipeline, Micro-service endpoint details and any other configuration related details (Ex: setting up ssh for bitbucket, network setup for docker images etc.,)
14. Ensure Security features (Role based Authentication and Authorization, OAUTH/JWT, Password Encryption, Centralized config server, Secure Secret storage service)
15. Zipkin
16. Api Building CRUD - Customer  
    2. Testing     
      2.1 Unit Testing   
      2.2 Integration Testing  
      2.3 NFR   
        2.3.1 Stress Testing   
        2.1.1 Regression Testing   
      2.3 Automation Testing   
        2.3.1 Selenium + BDD (Cucumber)    
    3. Reporting   
      3.1 Elastic   
      3.2 Logstash   
      3.3 Kibana   
    4. Containerization for all the services (docker)  
    5. Orchestration for services if needed (Kubernetes)  
    6. CI/CD - Jenkins   
    7. Bitbucket - for code repository   
    8. Quality gateway on sonar PS   
    9. ESB- with kafka, we should have min 1 service should have communication between    
      two services with kafka or we can have kafka used in ELK stack   
    10. Cloud - please get SME input for multi cloud (but it hybrid cloud as of now)  
    11. Swagger open api 3.0 - pls refer to swagger official site   
    12. Auth and Authorization- we shall do it with JWT (please talk to with SME)  
    13. Email services when user registers   
    14. Validation services on UI + services   
    15. Service DTO Layer

16. configuring error handling   
17. using loggers to DB