Driver Onboarding Service

About Me

Ankita Gupta this side!

- Having 8 years of experience
- Working as Senior developer at Oracle OCI since 3+ years
- Prior to that worked for 5 years with SAP Labs
- MTech grad from BITS Pilani
- Tech stack: Java, Microservice, OCI, Springboot

Problem Statement

a) Allow a driver to sign-up and enter their profile information

b) Trigger onboarding processes like document collection, background verification, shipping of tracking device, etc.,

c) Allow a driver to mark when they are ready to take a ride

Building a ride management application like Uber. We are responsible for the driver onboarding module.

Define list of the API interfaces required to expose to support above requirements

Define data stores, tech stacks that can support the use case.

Provide design diagrams to describe clear design of the overall solution

Design should be resilient, scalable, performant, with test cases

Approach

Desiging a microservice which caters to a driver onboarding module for a ride sharing app

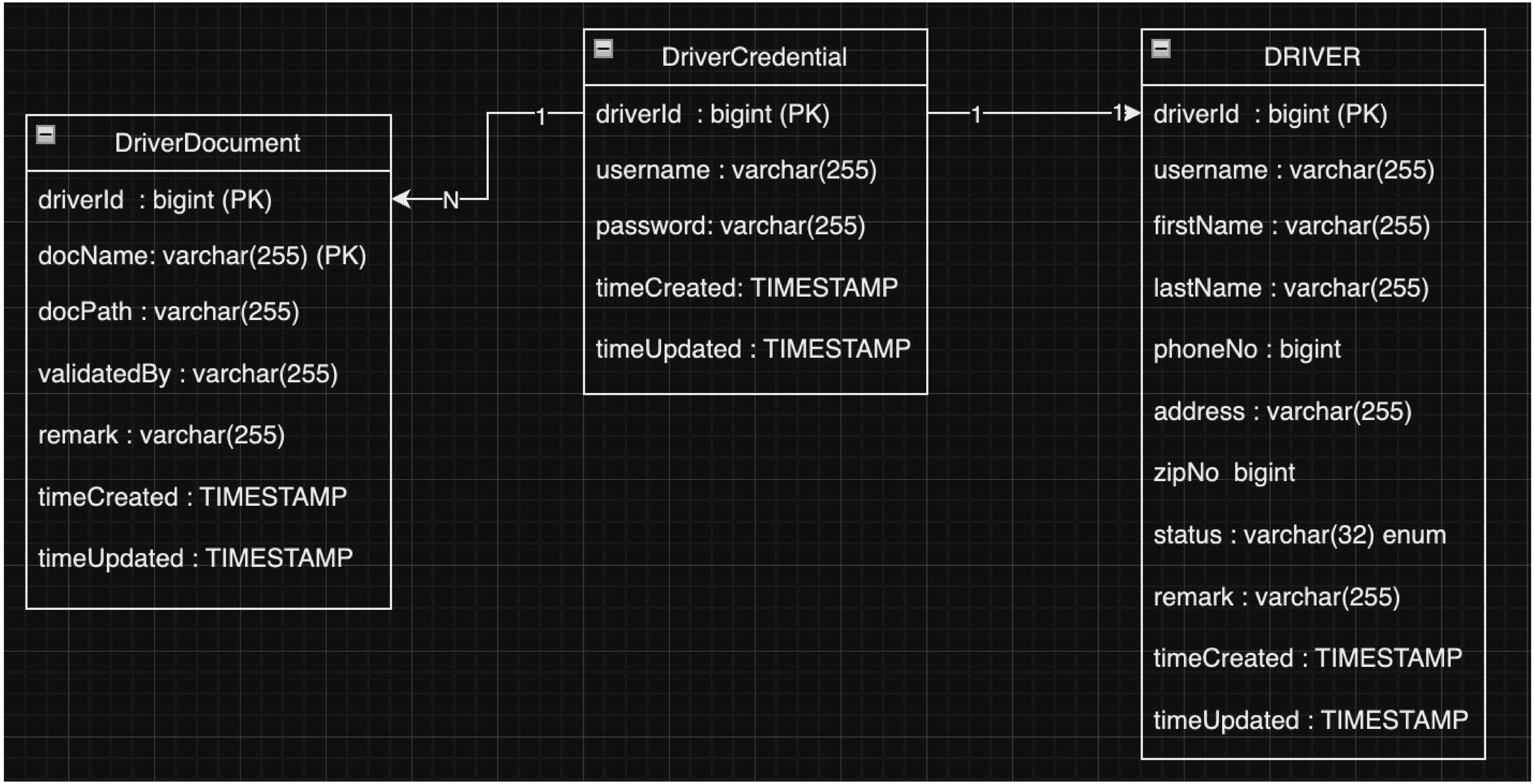
Tech Stack

- Backend - Java
- Framework - Springboot
- Build - Maven
- Deployment/Hosting: Docker, Kubernetes(Minikube)
- Database: PosrgreSQL

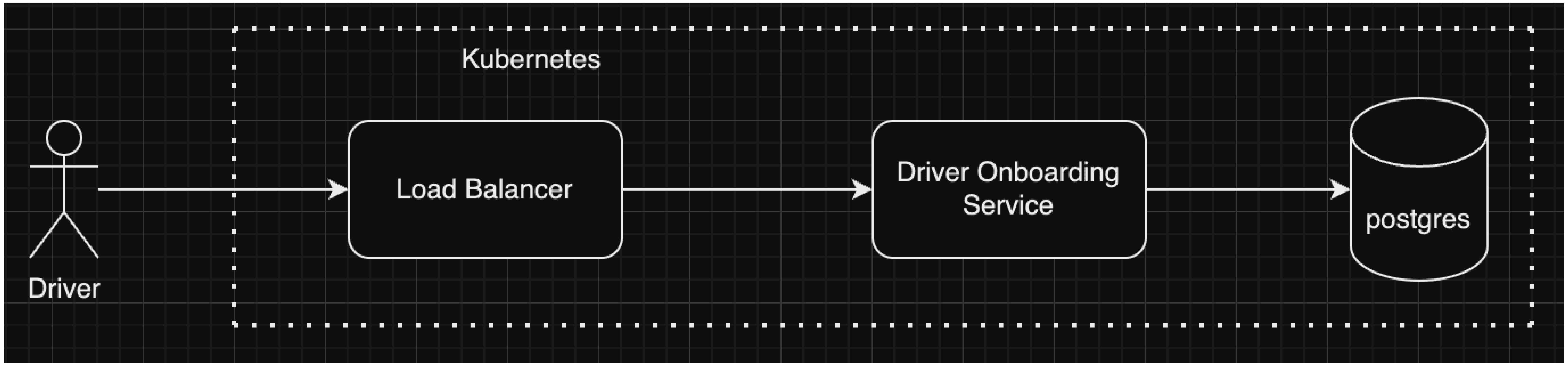
For this use case I have majorly focused on the backend side of the things like:

- APIs : To onboard driver, maintain profile, upload relevent documents
- Database : To persist driver details and docs
- Deployment : To deploy the application & end-to-end execution

Database Diagram



High Level Diagram



Flow:

* Driver interact with host port exposed by Load balancer service

* Loadbalancer service routes the traffic to app server

* App server serves the request and store the Driver details in postgres Db and document it is storing in server disk(This can be enhanced to use a object storage service for multiple replicas to serve)

base-url: v1/rider/

APIs

1. POST /driver/register : To register a new driver
2. POST /driver/login : To verify login of a driver
3. GET /driver : Get all drivers
4. GET /driver/{id} : Get Particular driver details
5. PUT /driver : Update driver details
6. PUT /driver/{id}/status : Update driver status
7. DELETE /driver : Delete the driver
8. POST /driver/document/{id} : Insert driver document
9. GET /driver/register/{id}/{filename} : Download the driver document

Thanks!

Do you have any questions?

https://github.com/Ankita173/DriverOnboardingService

gupta.nt28@email.com
+91 96631 55200