PROJECT REPORT

BOB
(A docker powered build and testing system)

UE20CS352 – Project

Submitted by:

Mohamed Ayaan	**PES2UG20CS200**
Navin Shrinivas	**PES2UG20CS237**
Mukund Deepak	**PES2UG20CS206**

Under the guidance of

Prof. Saranya
Designation
PES University

January - May 2023

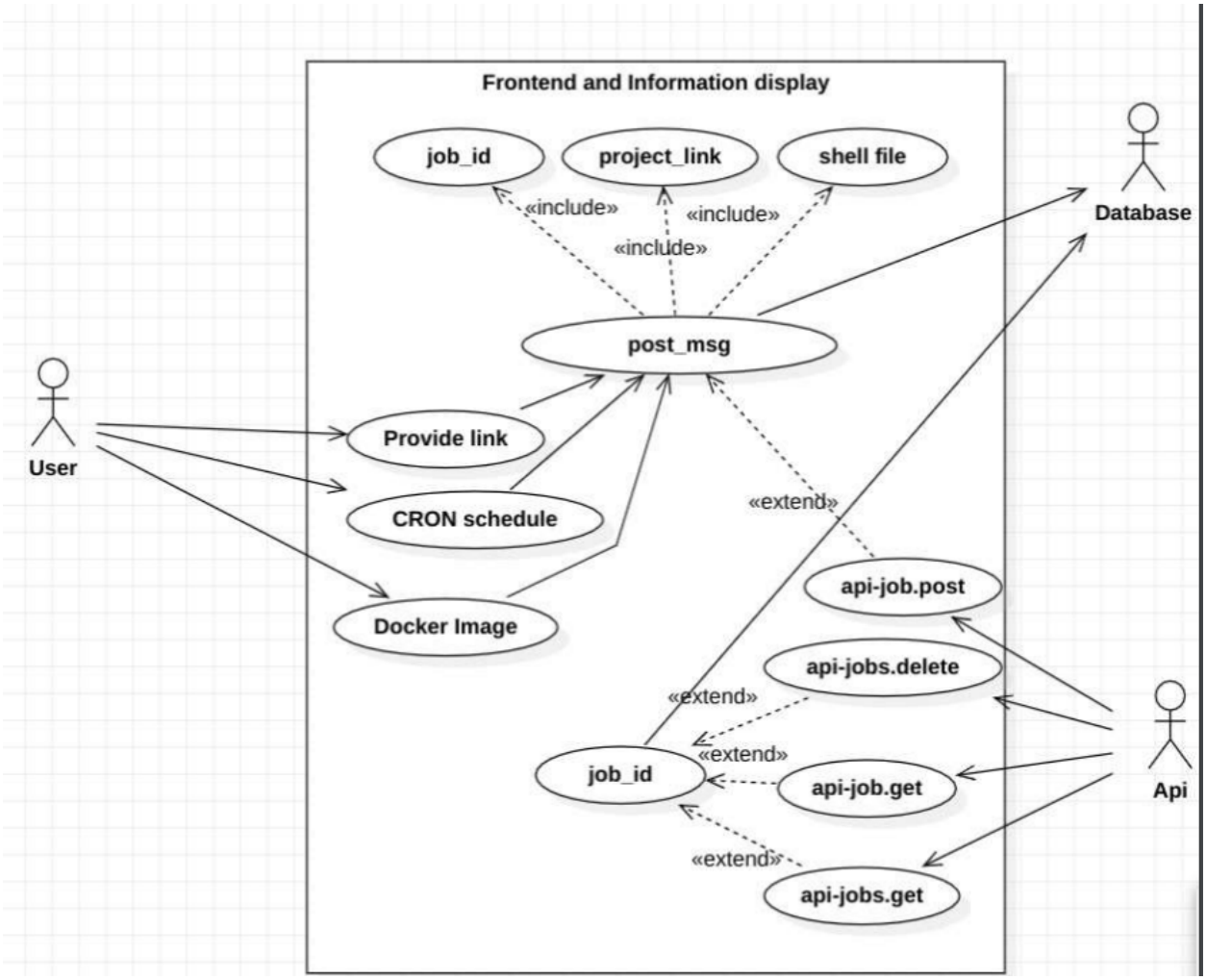
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

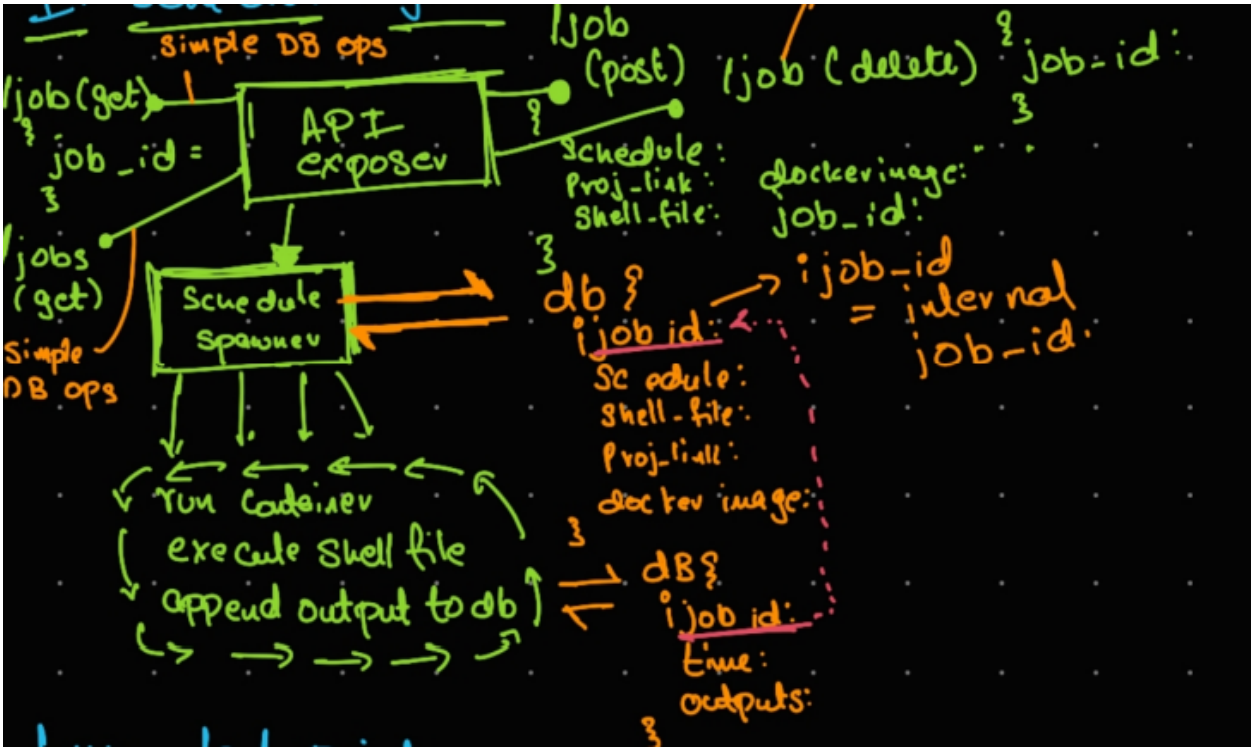
1. Problem Statement

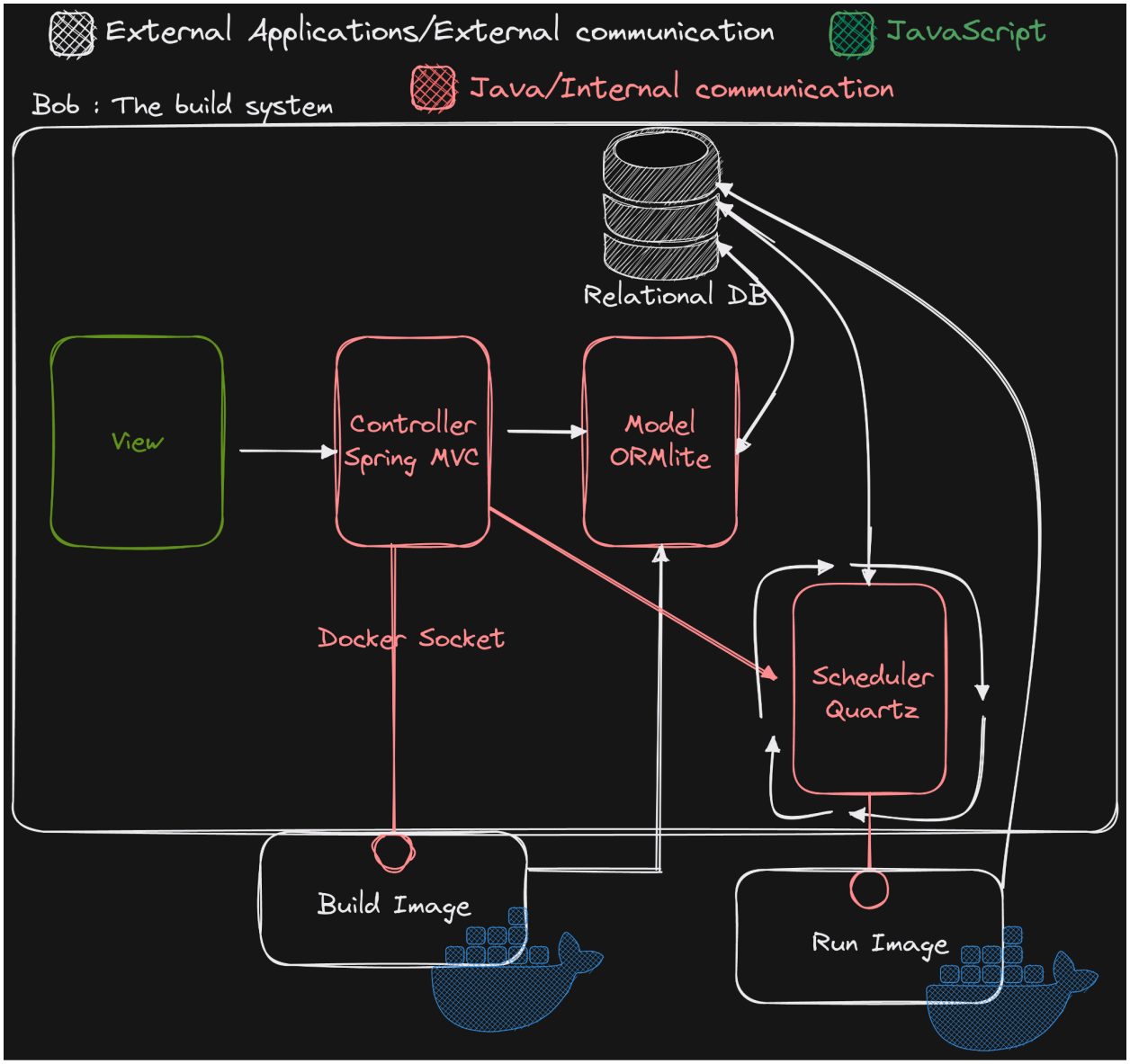
Jenkins, an industry standard tool for deployment,building and more for GitHub projects however it can be very complicated and difficult to handle at times.

Bob, our alternative is an automated build system. Uses containers to spawn dedicated runtimes for every job. Jobs are handled using POSIX-compliant cron syntax. Bob will build and test projects using the docker container the user provides that is published to DockerHub. The shell script that is run on the docker container is also provided by the users.

2. Models




simple DB ops
/job (post)
/job (delete)
job-id:
/job (get)
job_id =
API exposer
schedule:
proj_link:
shell_file:
dockerimage:
job_id:
/jobs (get)
simple DB ops
Schedule Spawner
db{
ijob id:
schedule:
shell_file:
proj_link:
docker image:
ijob-id = interval job-id.
Run Container
execute shell file
append output to db
dB{
ijob id:
time:
outputs:



3. Architecture Patterns, Design Principles and Design Patterns

3.1. Design Patterns

- API gateway: Aggregate calls to microservices in a single location, the API Gateway. The user makes a single call to the API Gateway, and the API Gateway then calls each relevant microservice.

- Callback: Callback is a piece of executable code that is passed as an argument to other code, which is expected to call back (execute) the argument at some convenient time.

- Data Access Object: The object provides an abstract interface to some type of database or other persistence mechanisms.

- Pipeline: Allows processing of data in a series of stages by giving in an initial input and passing the processed output to be used by the next stages.

- Thread: Create a method via which a connection can be made to the frontend and data can be supplied to it.

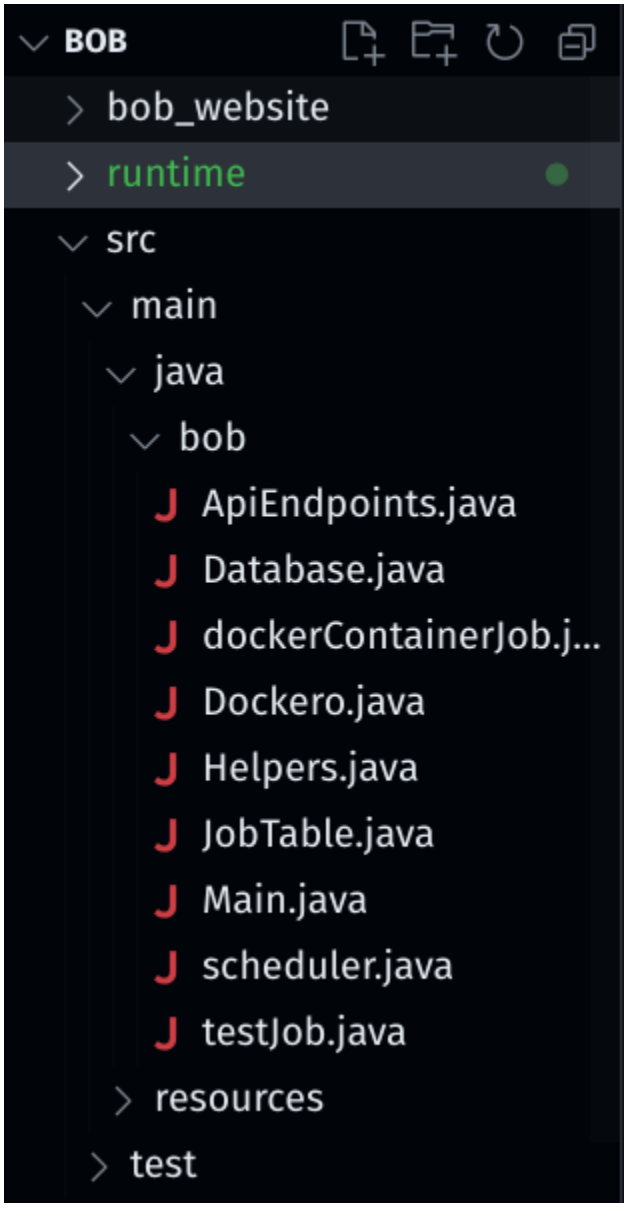
https://java-design-patterns.com/patterns/pipeline/

3.2. Architecture Patterns

We are following Model View Controller patterns wherein the APIEndpoints.Java and Database.java are responsible for creating a database, and connecting the endpoints between the view and the controller modules are the **Model modules**.

In our case, almost every other .java file is a **Controller module** which is tasked with dealing with various situations, ranging from storing the data within the docker image to creating the docker image itself.

We have written a javascript frontend in Next.JS so as to simplify the process and easily obtain data from the user in a succinct manner, this is our **View Model**.



3.3. Design principles

OCP:- We followed OCP when we had to allow various functions to be created for the class which allows us to extend the project even further but the modification of it is no longer required as each function has a succinct task.

ISP:- We ensured that the users would always have to deal with the smallest possible interface as we aimed to simplify the experience of the build system

4. **Github Link**

https://github.com/mukunddeepak/bob