**Retail SAAS High Level Design**

## Introduction

This document describes the high-level architecture for the retail saas.

Technology Stack:

Backend: Java, spring rest api’s deployed in tomcat server, Mongo-dB, Elastic search, AWS cloud.

Security: Spring security with JWT implementation.

Front-End: Angular JS website, Mobile Apps

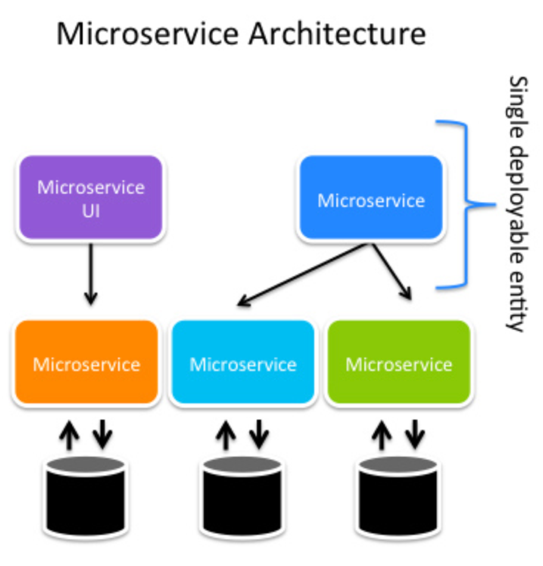
Cloud Deployment:

* Deployed In aws elasticbeanstalk tomcat linux env.
* M5 large (8 GB, 2 vCPU) General Purpose instance type.
* Auto scaling is configured minimum 2 and maximum 10 instances.
* If cpu usage increase to 40% than scale instances up.
* If traffic is increased than scale instances up.
* Configured cloud watch alarms if env. got degraded for any reson.
* Application load balancer is configured.

Additional Features to be added in Future:

* Reward/ loyalty program
* Offer management for supplier/retailer
* Dashboards inventory of products for supplier/retailer

High Level Architecture:

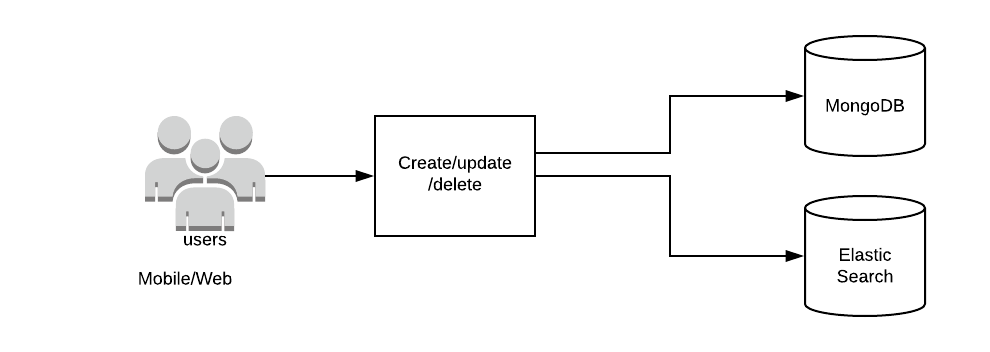


This solution is divided into different module based on the feature set.

* **Onboarding** 
  + Retailer & Suppliers should be onboard on the platform.
  + Retailer should be able to select the supplier he/she wants to work
  + Supplier should be able to select retailers they want to work

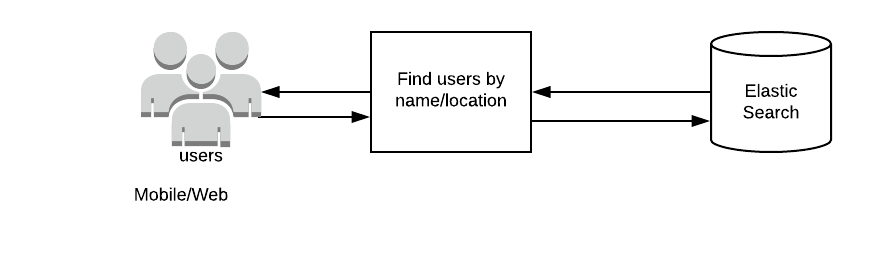
Block diagrams

Create/Update/Delete User:

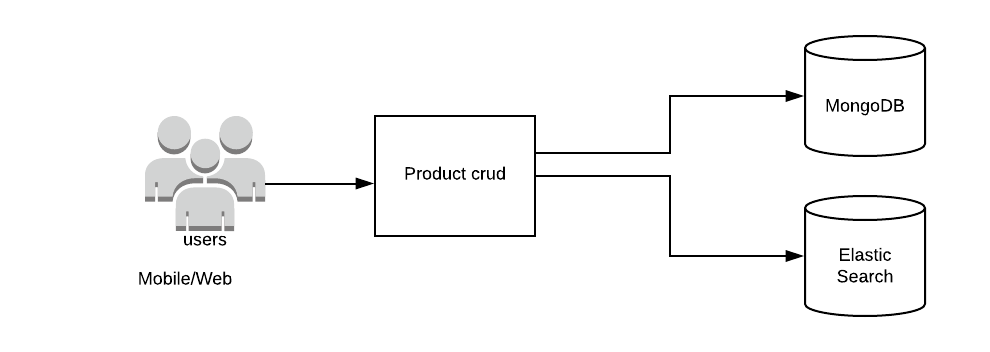


Rest API’s for create/update/delete users.

Find Users (By name /location):

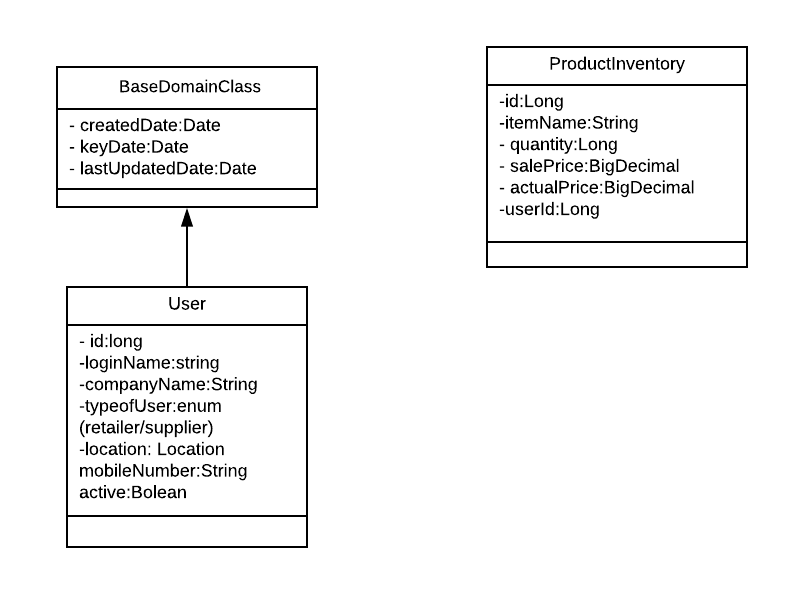
Rest API’s for find users.

Add/Modify/Delete products inventory for user



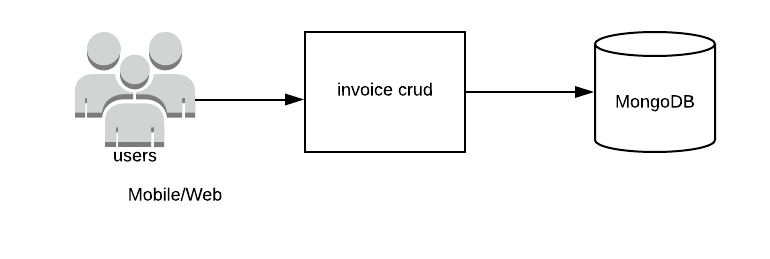
Rest api’s for add/modify/delete.

Data Models

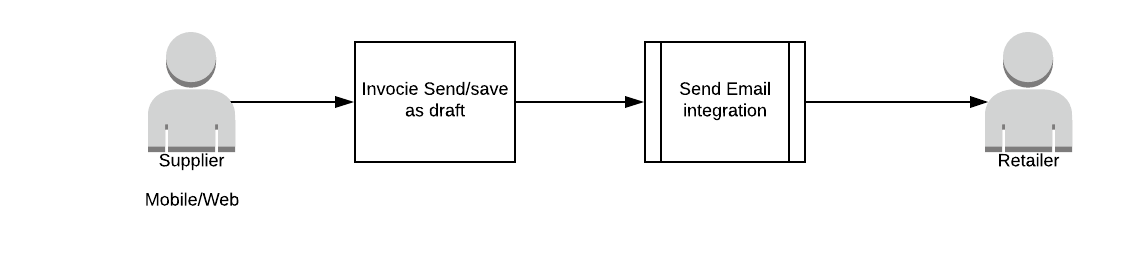


* **Generating Invoice** 
  + Suppliers should be able to upload the invoices in digital form onto the system and against a retailer

Block Diagram:



Send Invoice to Retailers:



Send email integration: We need to integrate a send email rest api. Which will responsible for creating the email template and send the email to retailer user.

Rest API endpoints details:

* crud for invoice
* send invoice to retailer
* Invoice saves as draft
* **Credit** 
  + Suppliers can provide credit-based payments. Which means that once an invoice is generated retailer will have X days before the payment has to be completed
  + System should be able to do the scheduling of the invoice presentation and have proper reminders to be sent to the retailer
  + Suppliers should have the capability of configuring the credit period based on historical transactions.

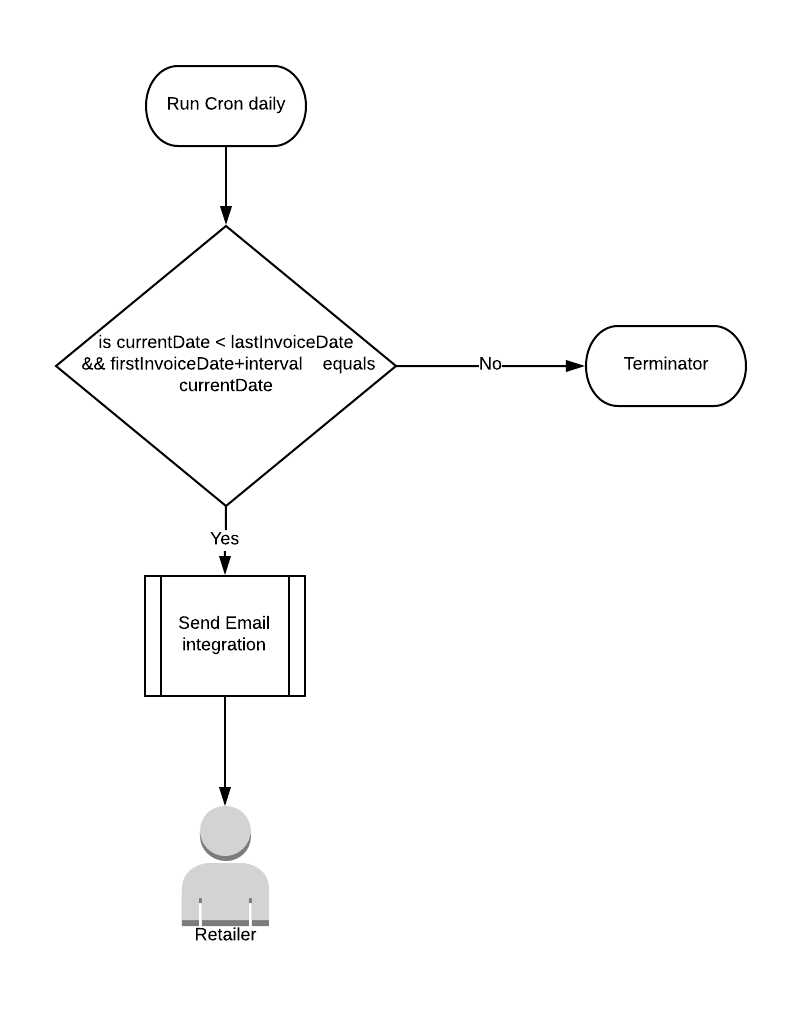
Note:

* While creating the invoice supplier should be able to fill the dueDate and creditDays as mention in the domain modal.

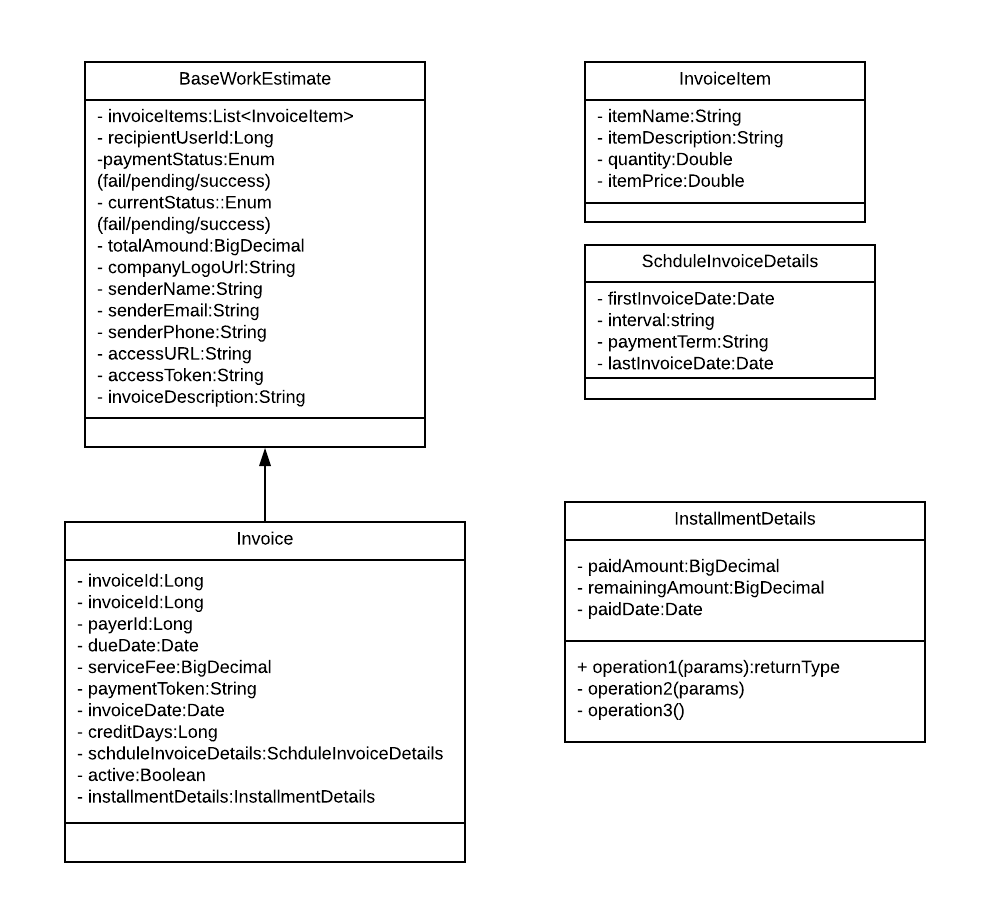
Scheduling Invoice:

While creating the invoice supplier should be able to fill the scheduling information.

Cron job which will always look for SchduleInvoiceDetails.



Data Model:



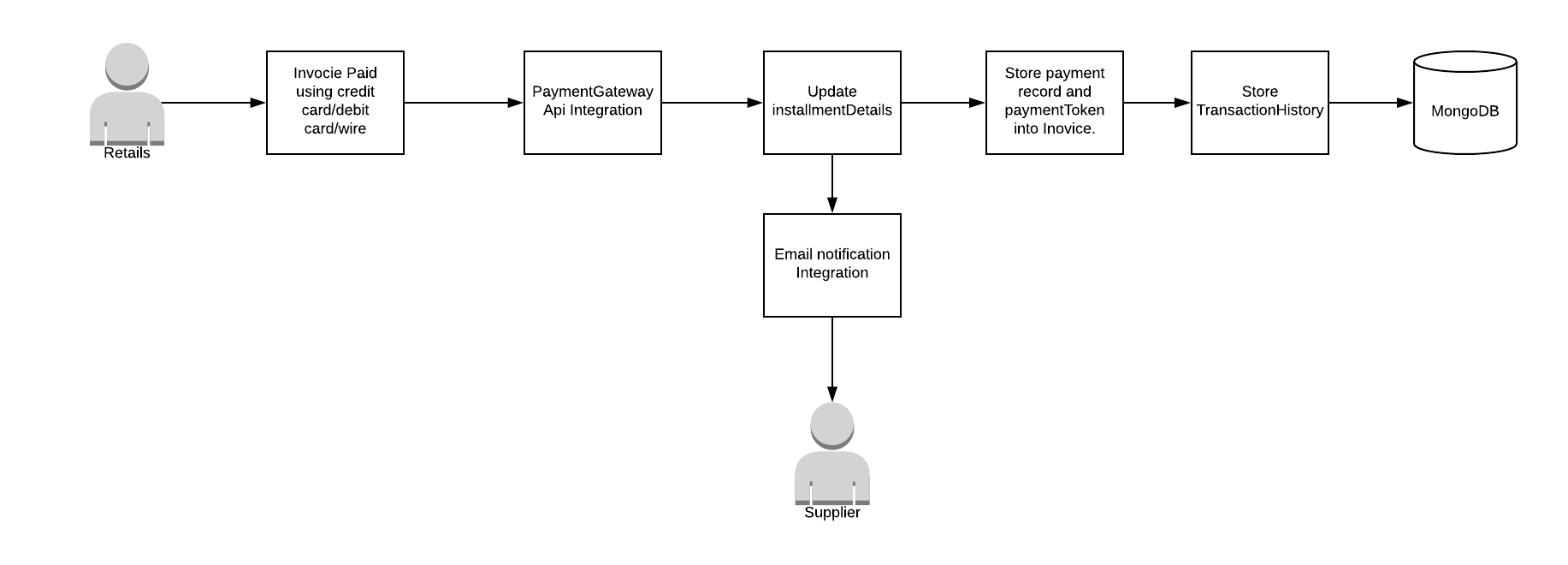
* **Payment**

## Retailer should be able to pay in Full or Partial against an invoice

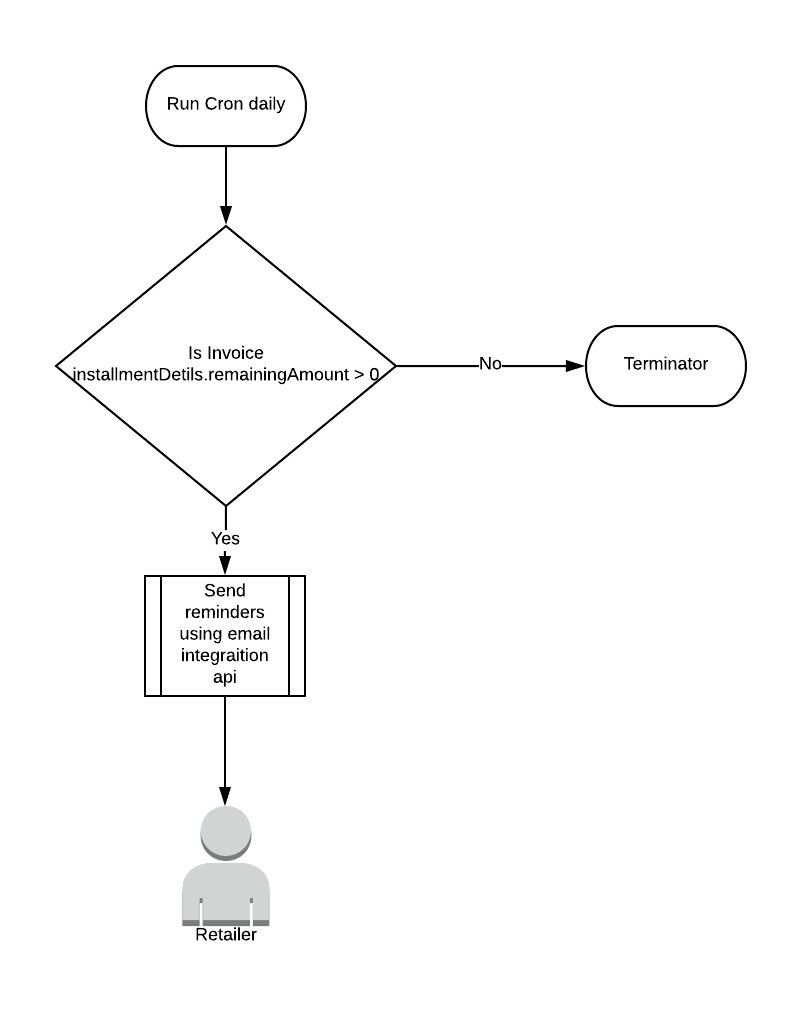
## System should be able to differentiate between partial and full payments and send reminder in case of partial payments

## Retailer should have multiple mechanism to pay the invoice.

Payment Gateway Integration: Responsible for create a payment using credit card/debit card/wire and store the paymentToken into invoice details. Change the payment status in invoice details.



Send Remainders for partial payments:



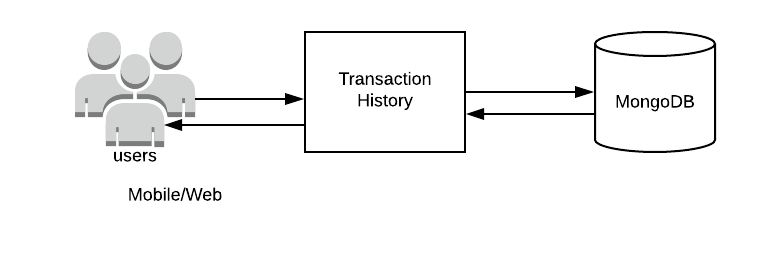
## 

* **Reporting & Analytics**

## Suppliers should be able to see data around the retailers and their payments

## Suppliers should be able to analyze retailers who are prompt in payment and should be able to select those top retailers wherein they can provide them credit facility

## Should have recon report to understand invoice uploaded and payment received and how much is outstanding per retailer.



Domain modal:

