Rohit Kumar Naicker

Why Engineering

1. developing, designing or reverse engineering.
2. adaptable and a good conversationalist

3. Professionally 9 years of diverse experience in Full Stack Development.
4. Ever evolving Technology - reading feeds, blogs, HackerNews etc..

5. TimePass
    - Coding Competition
    - Topcoder Development Challenges
    - And to reenergize my self some console Games.

Acheivments

Toyota Automations

Role

- Identified and Volunteered
- Understood Current Model and Defined Scope

Why

- Issue Resolving Time Reduced from Days to seconds
- Man power reduced
- Effective Management
- Happy Client

Sena.com

Role

- 2 Member Team
- Managed everything
- Development and Deployment

Why

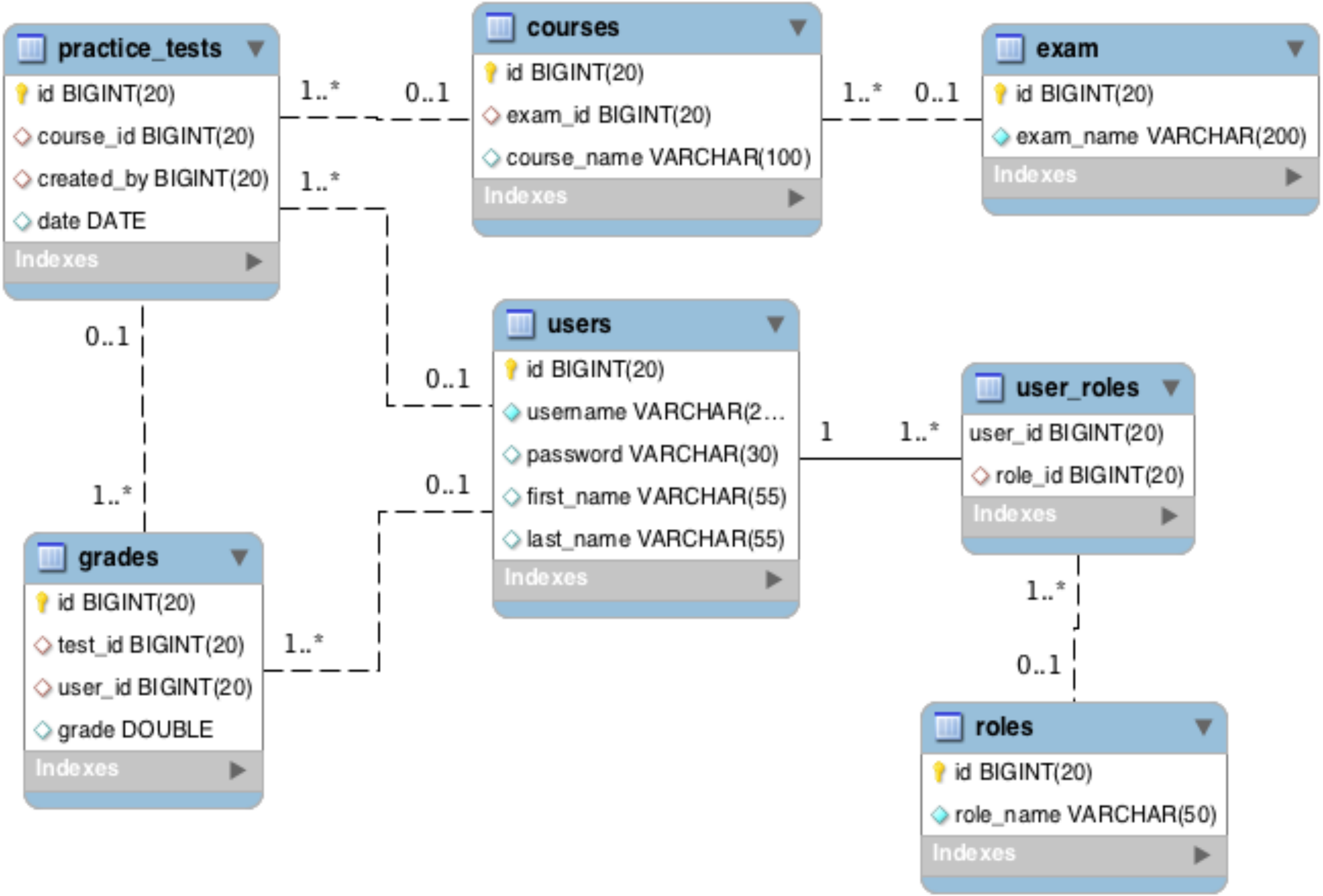
- PHP
- Wordpress, Themes
- Plugins Development
- Client Interaction

SAT Demo App

Framework

- Dropwizard - Rest API
- AngularJS – Web
- Docker – Deployment and Managing
- MySQL
- Hibernate - ORM
- Python smtpd process – Dummy SMTP

DB Model



practice_tests
- id BIGINT(20)
- course_id BIGINT(20)
- created_by BIGINT(20)
- date DATE
- Indexes

courses
- id BIGINT(20)
- exam_id BIGINT(20)
- course_name VARCHAR(100)
- Indexes

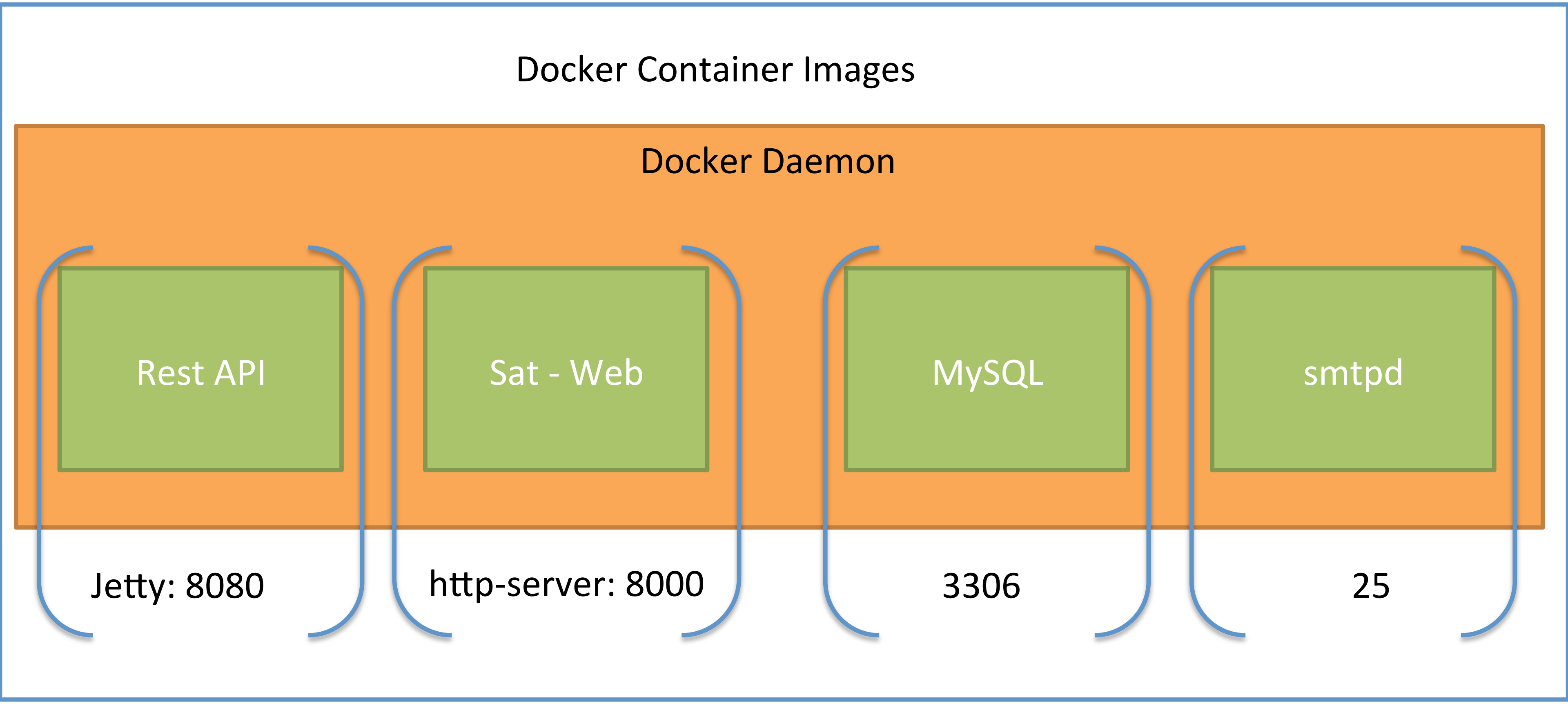
exam
- id BIGINT(20)
- exam_name VARCHAR(200)
- Indexes

users
- id BIGINT(20)
- username VARCHAR(2…
- password VARCHAR(30)
- first_name VARCHAR(55)
- last_name VARCHAR(55)
- Indexes

user_roles
- user_id BIGINT(20)
- role_id BIGINT(20)
- Indexes

grades
- id BIGINT(20)
- test_id BIGINT(20)
- user_id BIGINT(20)
- grade DOUBLE
- Indexes

roles
- id BIGINT(20)
- role_name VARCHAR(50)
- Indexes



Docker Container Images

Docker Daemon

Rest API

Sat - Web

MySQL

smtpd

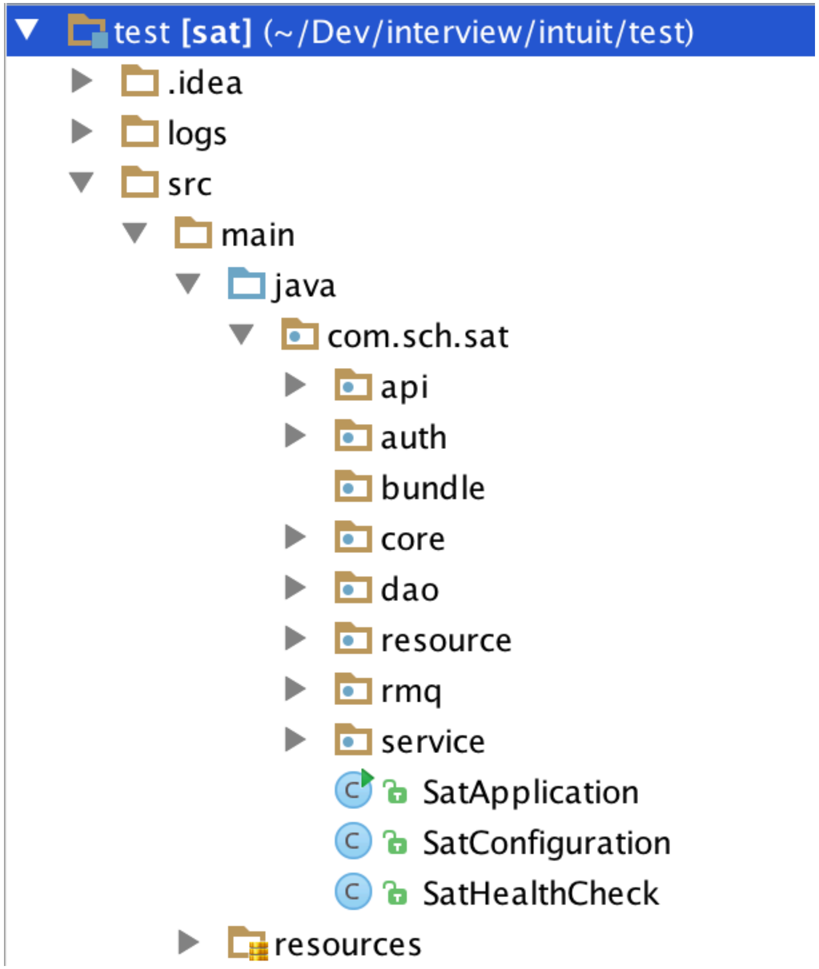
Jetty: 8080

http-server: 8000

3306

25

API Structure



```
▼ test [sat] (~/Dev/interview/intuit/test)
    ▶ .idea
    ▶ logs
    ▼ src
        ▼ main
            ▼ java
                ▼ com.sch.sat
                    ▶ api
                    ▶ auth
                      bundle
                    ▶ core
                    ▶ dao
                    ▶ resource
                    ▶ rmq
                    ▶ service
                      SatApplication
                      SatConfiguration
                      SatHealthCheck
            ▶ resources
```

API Structure

POST /api/register (com.sch.sat.resource.RegistrationResource)
For new registration

POST /api/authenticate (com.sch.sat.resource.LoginResource)
To login and validate

GET /api/courses (com.sch.sat.resource.CourseResource)
Get Available Courses

GET /api/exams (com.sch.sat.resource.ExamResource)
Get Available Exams

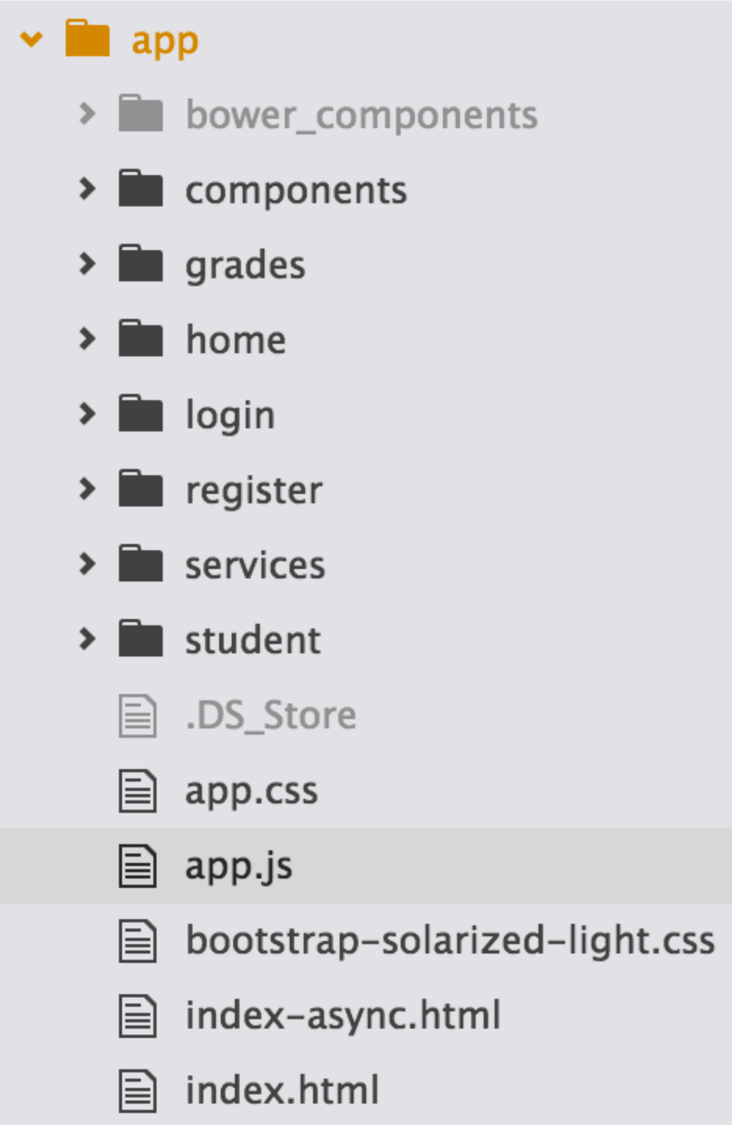
GET /api/grades (com.sch.sat.resource.GradesResource)
POST /api/grades (com.sch.sat.resource.GradesResource)
GET /api/grades/{id} (com.sch.sat.resource.GradesResource)
Grades get and update. TEACHER authorization required

API Structure

GET /api/schedule (com.sch.sat.resource.ScheduleTestResource)
POST /api/schedule (com.sch.sat.resource.ScheduleTestResource)
GET /api/schedule/{id} (com.sch.sat.resource.ScheduleTestResource)
Get available Practice tests

GET /api/users (com.sch.sat.resource.UserResource)
GET /api/users/{username} (com.sch.sat.resource.UserResource)
User Information

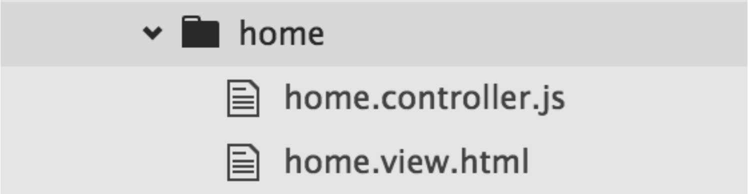
Web Structure



- app
  - bower_components
  - components
  - grades
  - home
  - login
  - register
  - services
  - student
  - .DS_Store
  - app.css
  - app.js
  - bootstrap-solarized-light.css
  - index-async.html
  - index.html

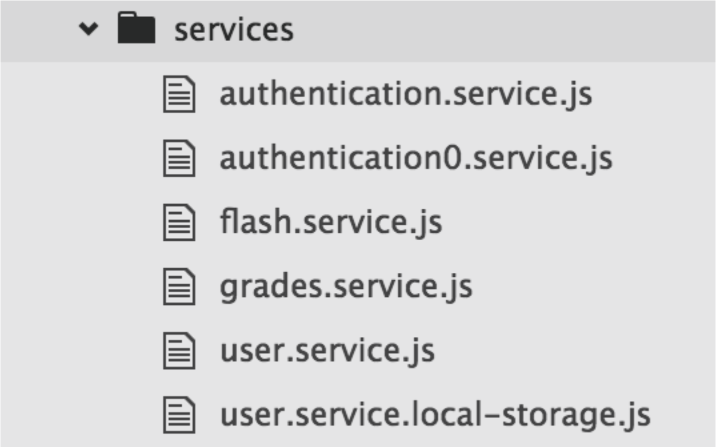
Controllers and Views

Grades, home, login, register, student



- home
  - home.controller.js
  - home.view.html

Services
API services



- services
  - authentication.service.js
  - authentication0.service.js
  - flash.service.js
  - grades.service.js
  - user.service.js
  - user.service.local-storage.js

How it Works

- Basic Authentication
- Rest API CORS enabled
- Redirected based on the role
  - Teacher schedule test and enter scores
  - Students receive mail
- AngularJS seperated base on MVVM patern

I wish I could have done

- Make this presentation with Deck.js and D3.js
- Validations

Thank You