Project 1

The first application I was working on was called Purge. It deleted old entries in database and elastic search and made log reports. When I joined the project this app was ready and I added a couple of new features like deleting old log files and purging of another databases. The second application I worked on called Pimu. It has been done completely by myself. Pimu solved the following task: it connected to ftp folder, searched a corresponding \*.bcp file, downloaded it, parsed it and made changes such as capitalizing info in particular column or changing one symbolic sequence to another or something like this. After that Pimu put bcp data into the oracle database. This task should be accomplished every hour or every day or smth like that, so in installation script(bat file) I added entry for Windows Scheduler. I wrote tests for it using JUnit and Mockito. The third task was called ReportGenerator or smth like that. Application queried a lot of oracle databases(It seems there were 8 of them) with various metadata and after changing it generated a reports. It generated reports for 4 quarters of the year. I put data in local database(sqlLite) by portions and only after that generate reports. I used executor framework(Executor, Future interfaces) for querying all databases simultaneously.

Project 2;

I took part in development of portal for widget development. Customers can: 1) buy widgets uploaded on portal; 2) develop their own widgets and upload them on portal using instructions given on documentation pages. I accomplished the following tasks:

1) I updated design of admin part of this application. Previously there were used gsp(kind of jsp for grails). My goal was to rewrite pages using AngularJS and apply new styles. These changes resulted in grails controller changes. I changed them to return json in case of success and error. 2) Specific widget need specific data for working. Widgets take that data from data sources(services). I worked on component which should be a proxy for these data sources - DataSource Proxy(DSP). What I've done: a) Added database for DSP. Previously It hadn't got its own database but we decided to make it independent from other components. b) Previously DSP was using only Vert.x without Spring. I added Spring(MVC, Data). c) We planned that in production there will be several instances of DSP. I added RabbitMQ by using Spring AMQP abstraction. d) I did perfomance testing. I used jmeter for making requests. Created images of machine instances on Amazone and had an opportunity to measure DSP throughput. 3) We had two backend components. One of them was responsible for creation of authorization tokens and the other managed entiies of our application such as widgets, data sources, orders, subscriptions, etc. My task was to merge these two modules, remove useless code, merge similar classes, update build config. 4) I wrote documentation for backend services using https://apiary.io/. 5) Bug fixing.

Project 3,4

Included a lot of research of various backend components. Resolved performance issue by decreasing number of http and database requests. Code refactoring with accordance to SOLID principles. Writing backend code and front end code, writing unit tests for backend(Junit, Mockito) and unit tests for front end(Qunit).

Project 5

Groovy language development, Spring(MVC, Boot, Data JPA), Spock Framework for unit testing. Mostly I was working on:

1) Creation of new endpoints(included writing controllers, service layer and dao layer).

2) Code refactoring: a) breaking huge classes on small logically finished components for increasing readability, making maintenence and unit testing easier.

3) Moving code from one microservice to another.

4) Covering code with unit and integration tests.

5) Setting up code quality metrics(jacoco reports) and SonarQube configuration.