# *Documentation of Sprint #3: Final Project*

***12.12.2014***

**Names:**

Boyko Surlev

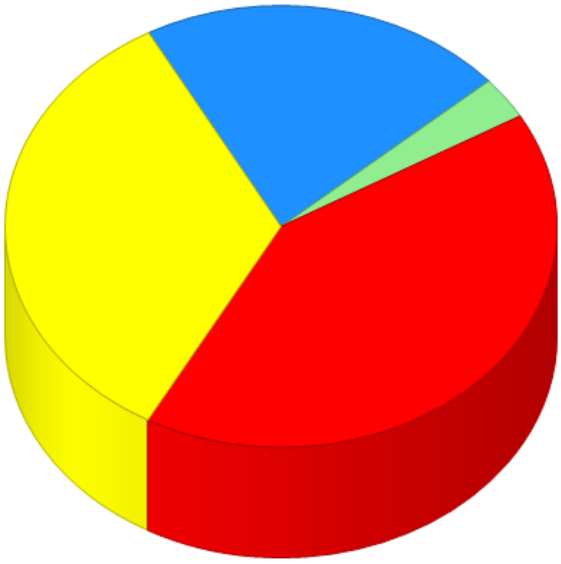
Nikolaj Desting

Peter Tomascik

## GIT overview

**5079 Lines of code**

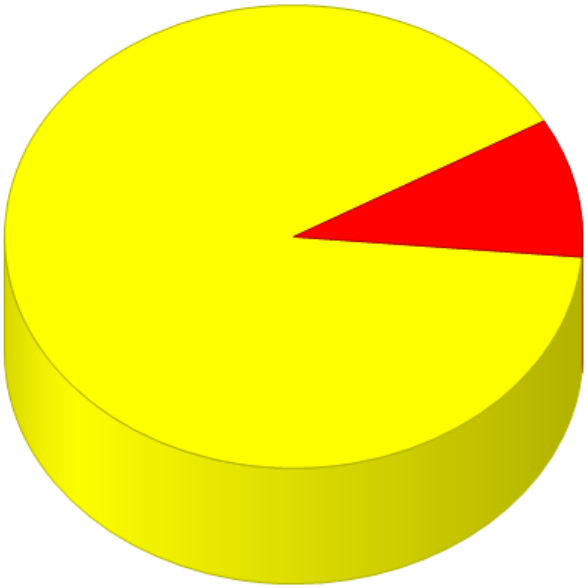
2100 Lines of code in .js files (41,3%)
1728 Lines of code in .html files (34,0%)
1093 Lines of code in .xml files (21,5%)
158 Lines of code in .css files (3,1%)

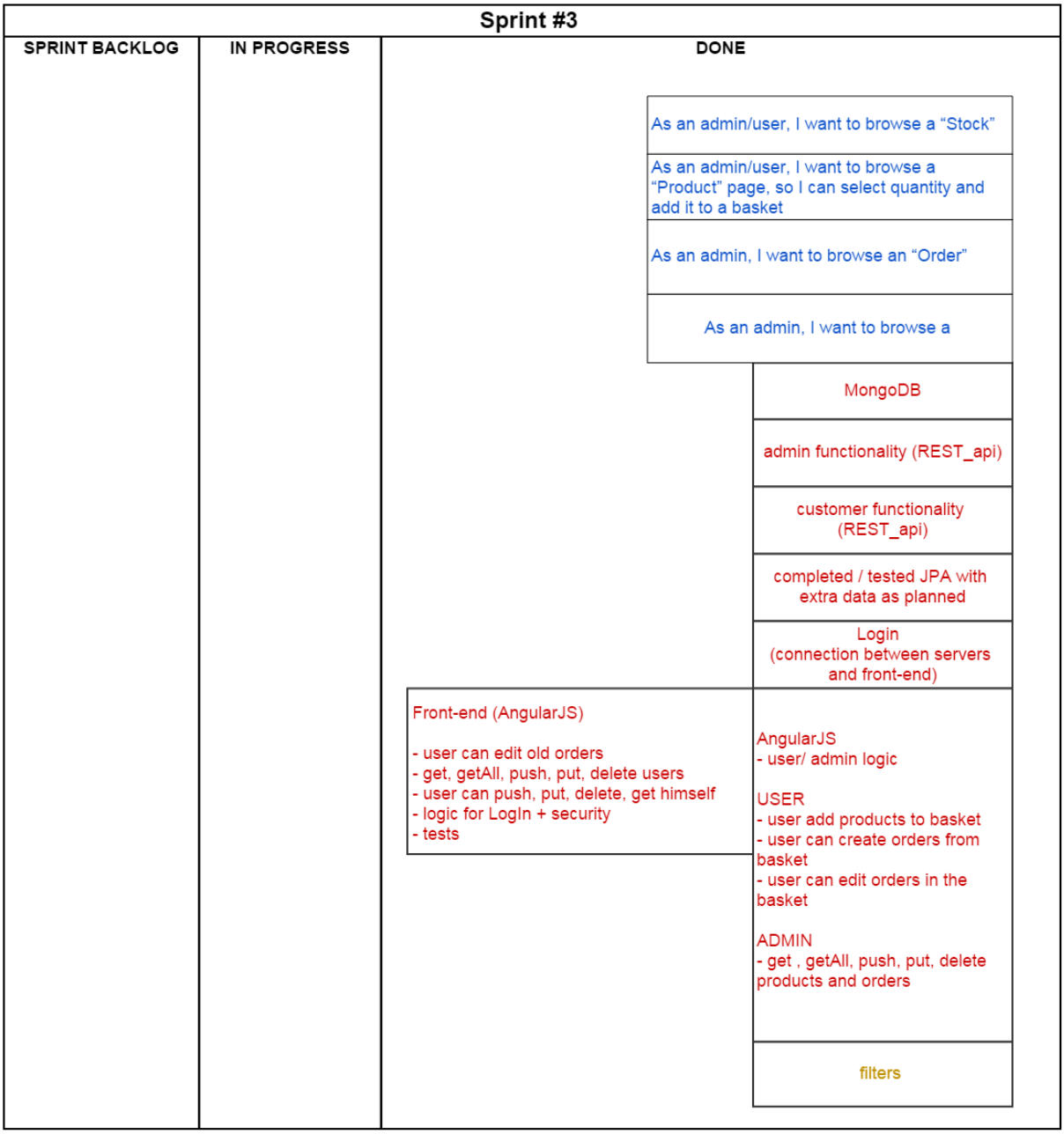


Excluding merges, **4 authors** have pushed **30 commits** to master and **49 commits** to all branches. On master, **63 files** have changed and there have been **3,752 additions** and **852 deletions**.

**494 Lines of test code**

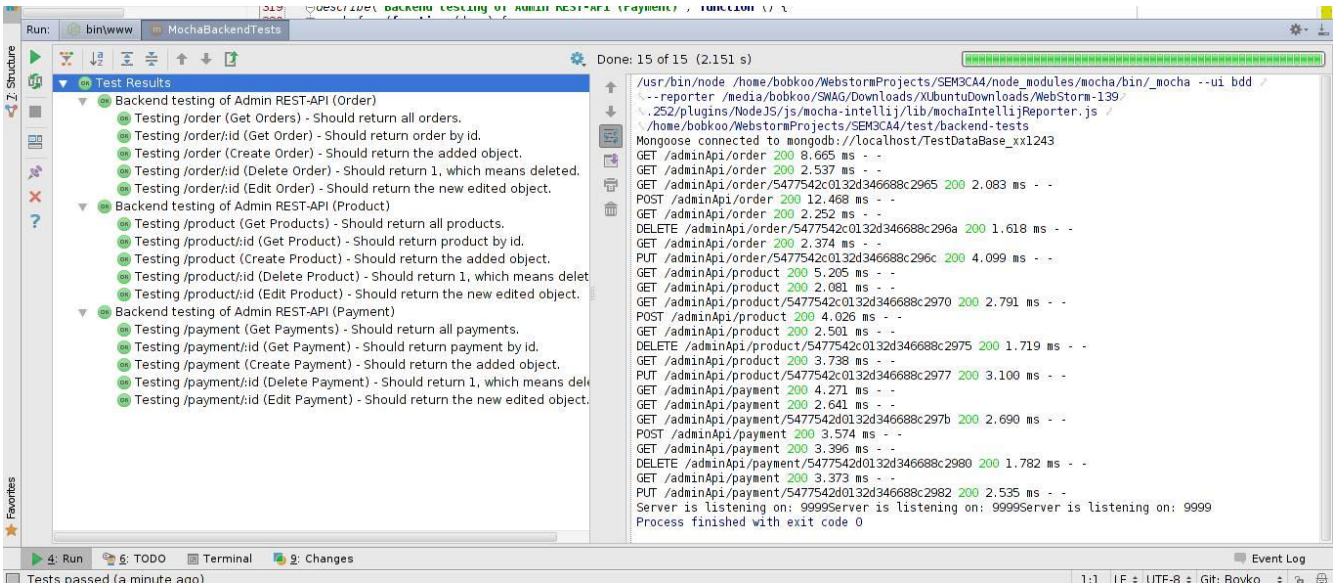
494 Lines of testcode (9,7%)
4585 Lines of production code (90,3%)

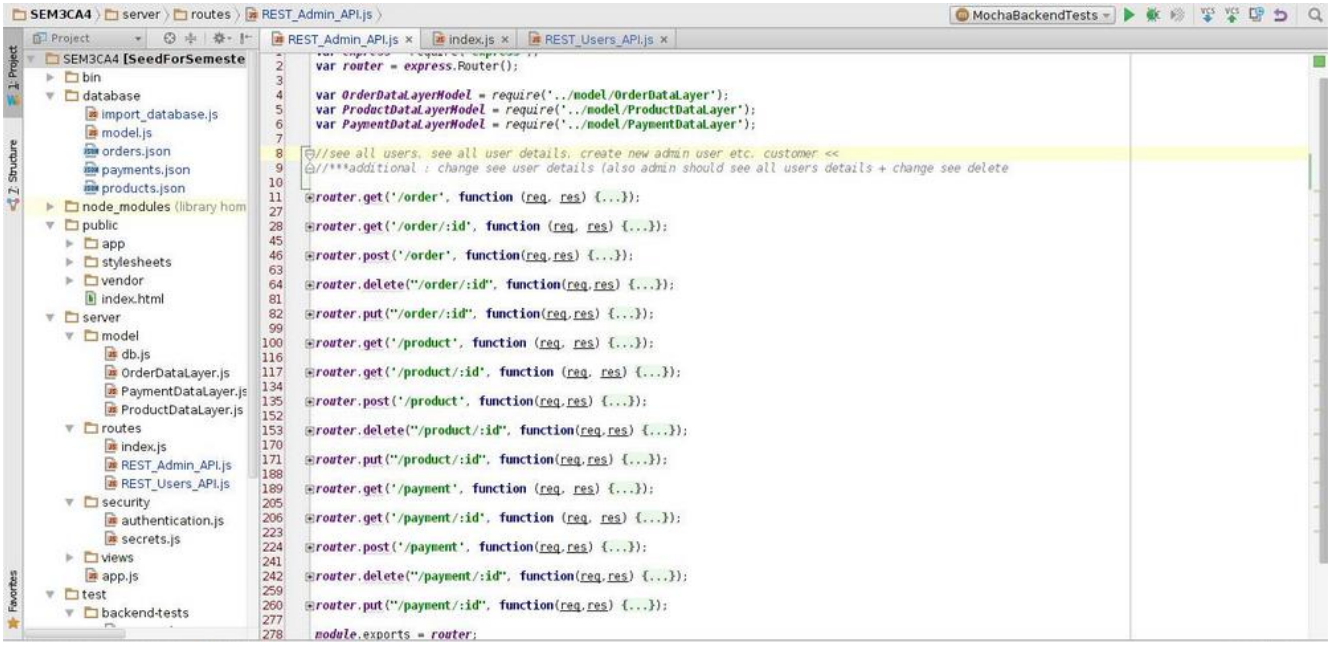



Sprint #3
SPRINT BACKLOG
IN PROGRESS
DONE
As an admin/user, I want to browse a "Stock"
As an admin/user, I want to browse a "Product" page, so I can select quantity and add it to a basket
As an admin, I want to browse an "Order"
As an admin, I want to browse a
MongoDB
admin functionality (REST_api)
customer functionality (REST_api)
completed / tested JPA with extra data as planned
Login (connection between servers and front-end)
Front-end (AngularJS)
- user can edit old orders
- get, getAll, push, put, delete users
- user can push, put, delete, get himself
- logic for LogIn + security
- tests
AngularJS
- user/ admin logic
USER
- user add products to basket
- user can create orders from basket
- user can edit orders in the basket
ADMIN
- get , getAll, push, put, delete products and orders
filters

## Back-end

## Rest_API (Node) + Mocha tests



**JAVA + J-Unit**

```
//Singleton
private static FacadeLogic instance = null;

protected FacadeLogic() {...3 lines }

public static FacadeLogic getInstance() {...6 lines }

private void initializeTransactions() {...3 lines }

@Override
public String getUsersAsJSON() {...18 lines }

@Override
public String authenticateUserViaJSON(String username, String password) {...19 lines }

@Override
public String addUserViaJSON(String json) {...20 lines }

@Override
public String editUserViaJson(String json) {...45 lines }

@Override
public Integer changePasswordViaJson(String json) {...22 lines }

@Override
public Integer deleteUserViaJSON(String username) {...16 lines }

@Override
public String hashMe(String hashed) throws NoSuchAlgorithmException {...11 lines }
```

**Who did what?**

| What? | Who? |
|---|---|
| Java server + JPA | Peter / Boyko |
| Node-Express + MongoDb | Boyko |
| AngularJS | Boyko |
| server connection | Boyko / Peter |
| majority of tests (karma / JUnit) | Boyko |
| sprint documentations | Peter |
| final documentation | Peter / Boyko |
| product Backlog | Nick |
| sprint backlogs | Peter |
| architecture diagram | Peter |
| sequence diagram | Boyko |
| deploying (heroku/ azur) | Nick / Peter |
| design | Boyko |
| UI mock-ups | Nick / Peter |

**NOTE:** Nik and Peter created 2 functions of Node and AngularJS, particularly Payment and Your Orders. Rest of the work was done by Boyko.

**How far we have got**

If we look at the backlog, we discovered that we had to be realistic about what we were able to implement and what we were not able to.

We have all the requirements fulfilled except for the front-end tests, which gave us problems. If we look at the product backlog there are only a few of the stories that we never implemented. We never divided the orders into different tables depending on their status, but the main functionality that we thought was most important is implemented and works.

The same thing goes for the company profiles that we wanted to connect the users with. Instead of spending time on trying to make the complete application, we developed the functions for an admin and a user, which was the core logic of the whole system.

In the end, we are pleased with our application and we think this could be worth developing further in the future.