Documentation: CA 5 FINAL

28.11.2014

Names:

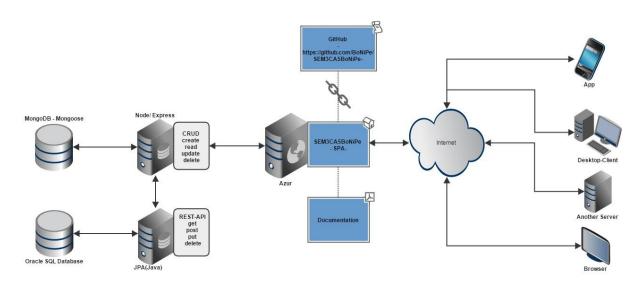
Boyko Surlev

Nikolaj Desting

Peter Tomascik



Architecture diagram

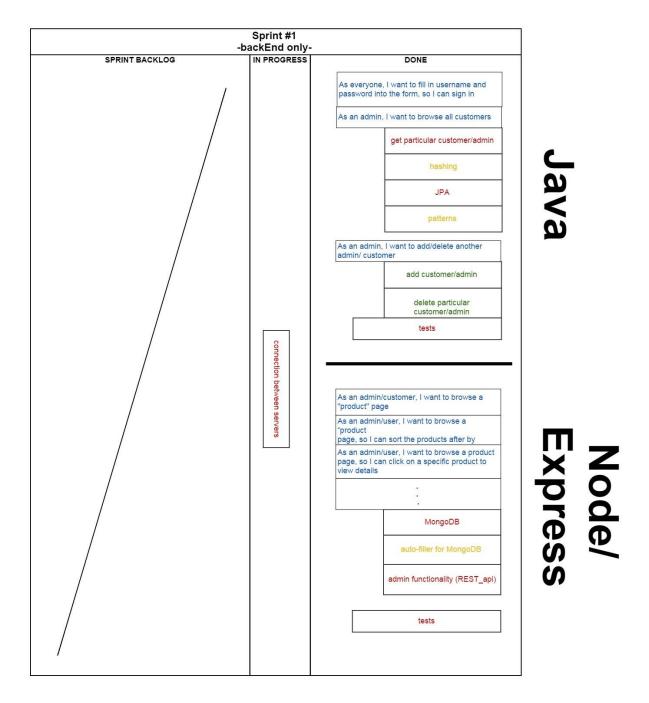


Server-side

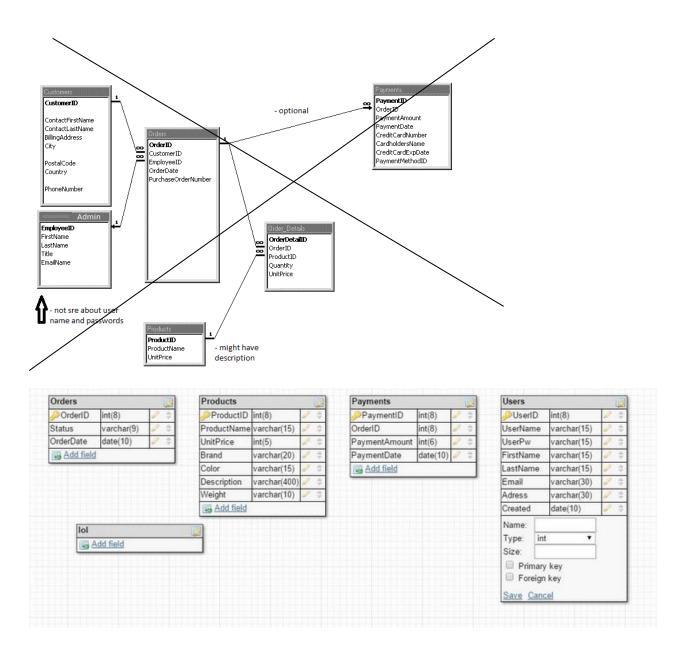
Client-side

 $\underline{http://tomascikpeter.wix.com/sem3ca5final} \quad \longleftarrow ---- \ \textbf{Ul mock-up}$

Sprint #1

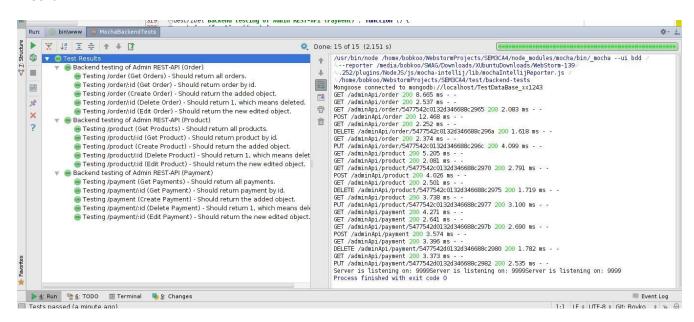


Domain Model



Back-end

Mocha



Rest_API

```
➡ SEM3CA4 ) ➡ server ) ➡ routes ) 🗃 REST_Admin_API.js )

    MochaBackendTests → ▶ 🕸 💖 🙄 👺 5 Q
                   ▼ 😌 🛊 🌣 ۴ 📵 REST_Admin_API.js × 🔯 index.js × 📵 REST_Users_API.js ×
   SEM3CA4 [SeedForSemeste
                                                 var router = express.Router();
         bín 🗀
                                                 var OrderDataLayerModel = require('../model/OrderDataLayer');
var ProductDataLayerModel = require('../model/ProductDataLayer');
var PaymentDataLayerModel = require('../model/PaymentDataLayer');
     ▼ □ database
            import_database.js
            model.js
            e orders.json
                                               ⇒//see all users, see all user details, create new admin user etc. customer <<

≙//***additional : change see user details (also admin should see all users details + change see delete
            payments.json
           products ison
                                               router.get('/order', function (reg. res) {...}):
     ▶ □ node_modules (library hom
      ▼ 🗖 public
                                               *router.get('/order/:id', function (req. res) {...});
           app
                                               @router.post('/order', function(reg.res) {...});
           stylesheets
         ► □ vendor
                                               router.delete("/order/:id", function(req.res) {...});
            index.html
                                         81
82
     ▼ 🗀 server
                                               @router.put("/order/:id", function(req.res) {...});
         ▼ □ model
                                               @router.get('/product', function (req, res) {...});
               db.js
                                        116
117
               @ OrderDataLayer.js
                                               @router.get('/product/:id', function (req. res) {...});
               PaymentDataLayer.is
                                        135
152
153
170
171
188
189
                                               router.post('/product', function(req.res) {...});
               ProductDataLayer.js
         ▼ 🗀 routes
                                               @router.delete("/product/:id", function(reg,res) {...});
               index.js
                                               router.put("/product/:id", function(req.res) {...}):
               REST_Admin_API.js
               REST_Users_API.js
                                               erouter.get('/payment', function (req. res) {...});
         ▼ 🗀 security
               authentication.js
secrets.js
                                               @router.get('/payment/:id', function (reg. res) {...}):
                                               router.post('/payment', function(req.res) {...}):
           views
           app.js
                                               router.delete("/payment/:id", function(req.res) {...});
      ▼ 🗀 test
                                               @router.put("/payment/:id", function(req.res) {...});
         backend-tests
                                                module.exports = router;
```

J-Unit



Who did what?

What?	Who?
Java + JPA	Peter (help of Boyko)
Node/Express + MongoDb	Boyko
Tests(all)	Boyko
documentation	Peter
backlog	Nick
sprint backlog	Peter
architecture diagram	Peter
deploying	Nick
Domain model	Nick+Peter
UI mock-ups	Nick+Peter