Week 6/7 — Warehouse management system — requirement specifications

---

Requirements:

- Must have products
  - Products must be able to be added and removed
  - ~~Dynamic ItemId~~
- Must have orders
  - Orders reserve items in system
  - SalesOrder; ReturnOrder; TransferOrder; PurchaseOrder
- Must be able to handle transactions; execution of above-mentioned order types
- Must be able to interact with RESTful API to communicate data with external systems
- Must have at least a minimal front-end
  - Login page
  - Warehouse page
    - Order view; order creation, order status
    - Warehouse view; products in warehouse
  - Products page
    - Product view; product creation, product status (?)
  - Admin page
    - User view; create, delete (?), edit, whatever
    - Warehouse**s** view, but different this time; create and manage warehouses
    - Role view; create and manage roles
- Must be scalable / flexible, open to future additions
- Must connect to a database
  - Appropriate CRUD-functions
- Installation guide must be included, I guess as a README

Non-specifics / agreements:

Language & frameworks:

- C#
- SQL

How to handle branching; when to push to GitHub:

- *Find out how this works*

Pull requests?:

- Very small group; perhaps checking one’s code through thoroughly should be enough. No handing off testing workload to other group member.

I would like for all code and comments to be written in English, and for all methods to have descriptive names and parametres.

Roles:

John:

- GitHub-setup
- ER diagrams + database-setup
- Kanban

Thor:

- UML diagram
- Development: Orders

Jakob:

- SCRUM-crap
- Requirement specs, presentation?
- Development: Warehouse, SKU, Product; interfaces

Shared:

- Cooperation document

Presentation:

Required to go over: choice of architecture (language, diagrams), how it was to work in a team (SCRUM, kanban) and a live(?) demonstration of our programme. We will have fifteen to twenty minutes to present.

Notes:

* Notes here as we progress *

1. Into-slide
2. Forklaring af opgaven, punktorden
3. Sarmarbejde, SCRUM, kanban
4. Demo — skærmoptagelse
5. ER-diagram — guf til data management-holdet
6. UML-diagram
7. Hvor udvidelig er vores kode; idéer til udvidelser, som vores projektstruktur kan understøtte? Interfaces hvorfra nye implementationer kan forekomme. Seperation af ‘production’- og ‘development’-environments.
8. Eventuelle tests, bevis for at (om) vores program er stabilt.
9. Outro-slide — *King Ghidora meme*