CA 3 MEB – (Mongo Express Booststrap)

There is a concept called the MEAN stack (Mongo-Express-Angular-Node) that refers to applications programed with these tools/frameworks. This is the goal for the final semester project, but for now, we do not have the "A" jet, so therefore the MEB stack. Thus, in this CA you need to create an application using all the stuff we have learned in this block: Node, Express, Mongo and Bootstrap – the later so famous MEB stack!

When this CA is completed, you are supposed to know how to:

- Use express to create a web server using Jade
- Use express to create web services
- Use mongoose and mongoDb from within Node.js
- Make a nice user interface based on concepts from Bootstrap.js

Studypoints for this period/CA are given as sketched below:

For your participation in the class (one for each day)	11 points
Each member in a team can earn additional 19 points for the CA as sketched below:	
For your contribution to the code and documentation (verified via Git + your documentation must include a note **in a separate section**, with who did what)	Up to 5 points
The quality of your design, test strategy/coverage and documentation	Up to 5 points
How far you came with step 1 and 2	Up to 3 points
For the bonus steps and part 3	Up to 6 points

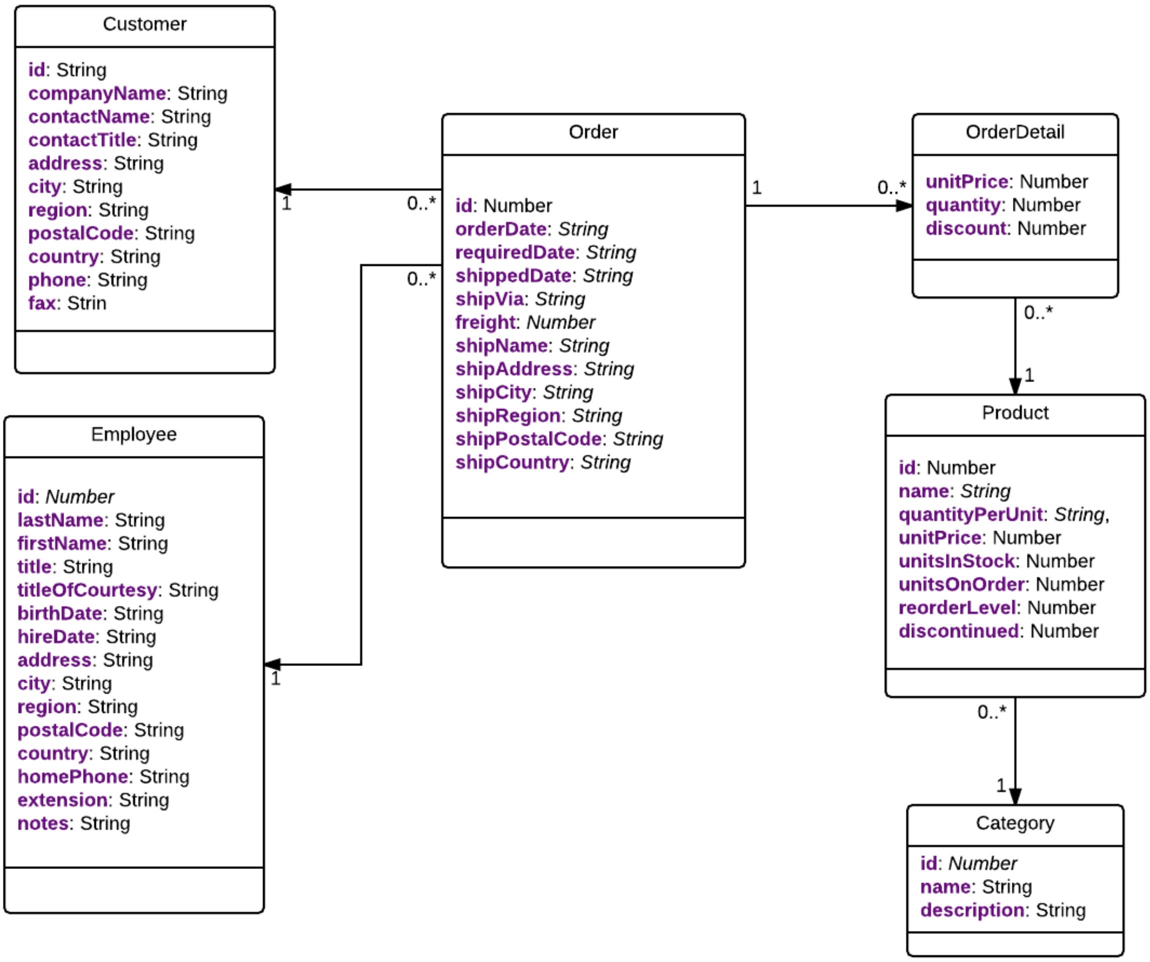
Demonstration and what to hand in for this CA

- The code must be made available via Github.
- A short description including (each with a separate heading):
  - A design description of the chosen design (both on front and backend).
  - A section stating who did what
  - A description of what you have added (bonus tasks and/or part 3)
- All documentation and the projects html-pages must be made available via a small web-site published on your web-server.
- The project must be made available to everyone via your group account on Azure

Hand-in: Friday (31.10.2014) before 24.00.

The Business Domain

In this CA we will work with a classic order database. The schema for the Order object is:



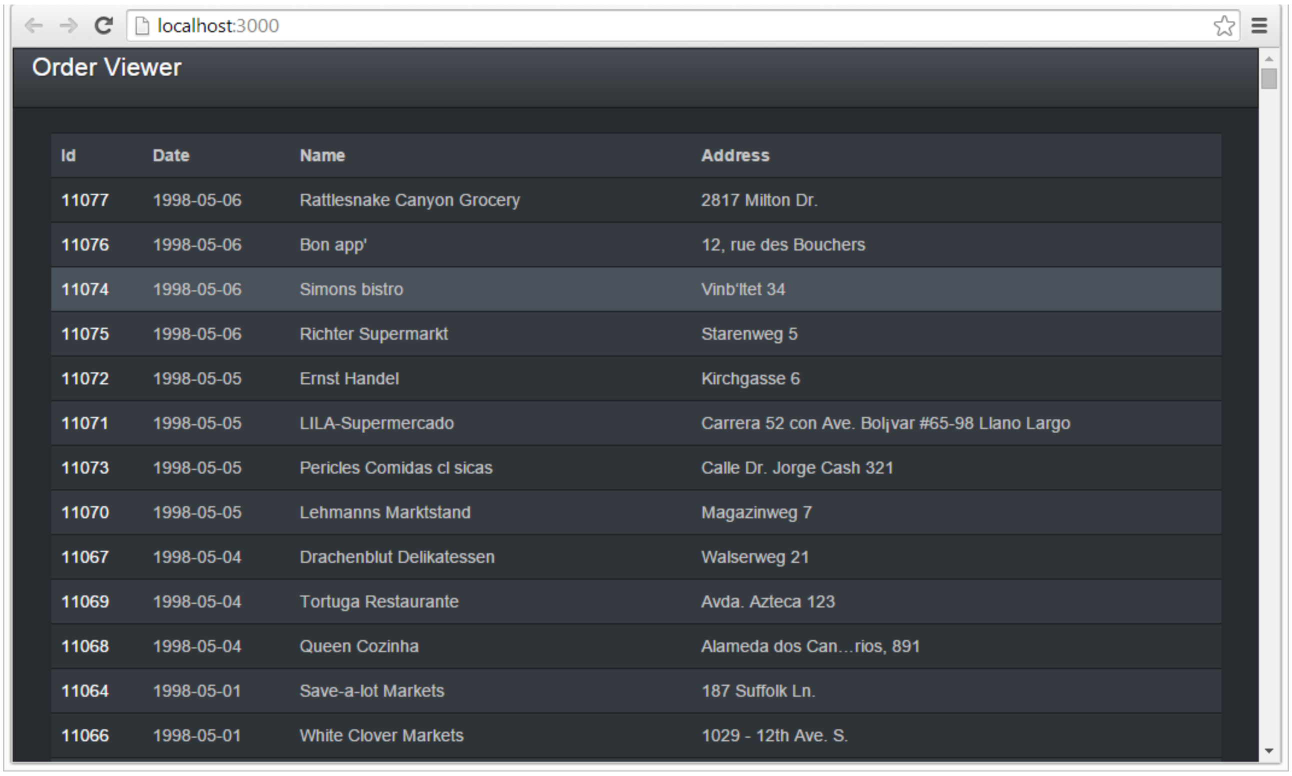
The data is a fraction of a well-known relational database previously used as the main sample database for getting started with Microsoft SQLServer. The relational model is converted into an object-oriented model in JSON and should be saved in your Mongo database. (see next paragraph)

The startup project

A startup project can be found in the `ordersViewStart.zip` file. The startup project is an express.js startup project with stylesheets and javascripts files for `bootstrap.js` and `JQuery` added. Furthermore, you will find a directory named `database` with a file `import_database.js` that will convert the relational model into an object-oriented model and importing that model into MongoDb (remember to start MongoDb before running this file). Line 5 in this file defines the database URL, `mongodb://localhost/northwind`, change this if you want to install the database into another location. In the `database` directory, you will also find the `Mongoose` schema for the database that you can use to communicate with the database. Compare the mongoose schema to the above object model to get a better understanding!

Part I

Create a web page that can show a list of all orders in the database (just significant information that can fit on one single line). Use bootstrap to design the page. On the backend (your express server) you must use Jade to create the HTML page. Use the layout structure given in the startup project. Here is an example creating a bootstrap table (using the table, table-striped and table-hover bootstrap css classes). The data must be sorted with the newest order first. The id must be a link such that a click will bring you the order details page (next part). Natural, you do not need to use the design theme used here. Go to getbootstrap.com, wrapbootstrap.com, startbootstrap.com or the like to get inspired.

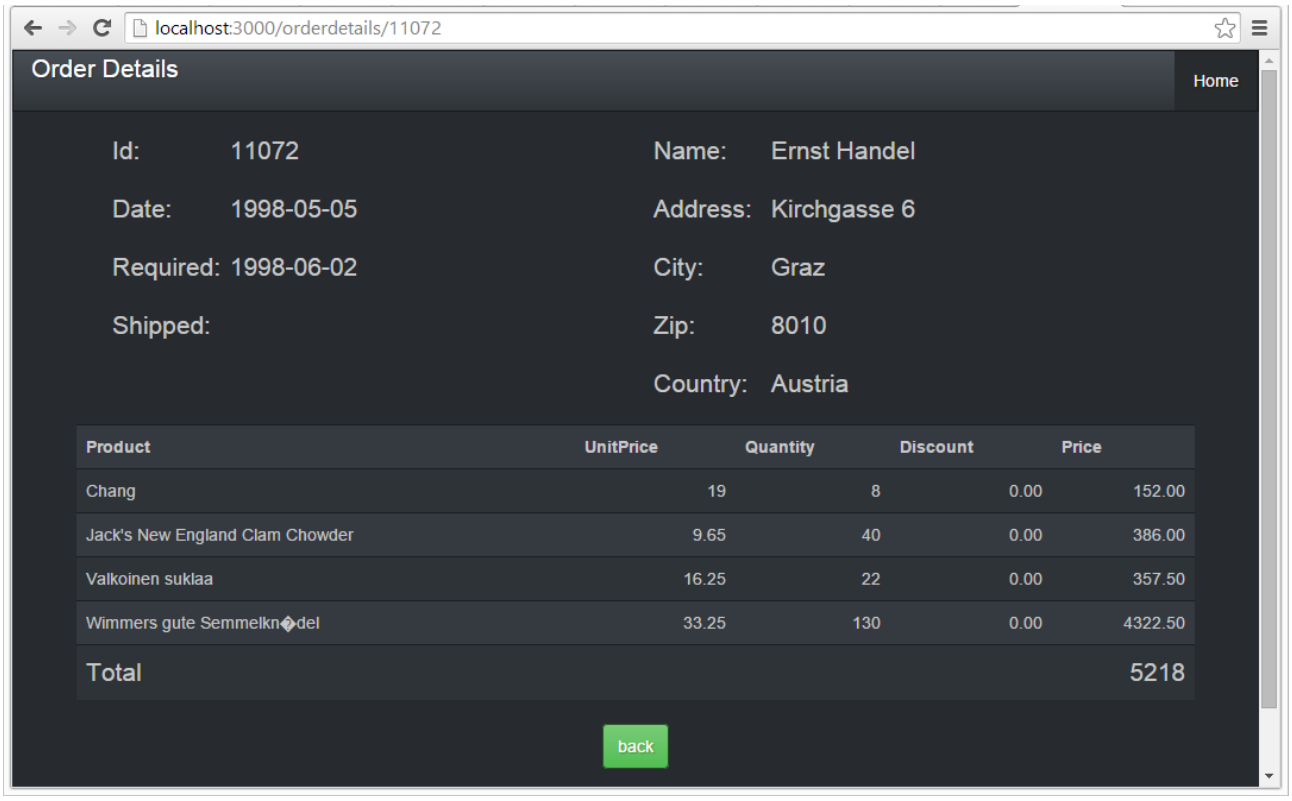


Order Viewer

Id	Date	Name	Address
11077	1998-05-06	Rattlesnake Canyon Grocery	2817 Milton Dr.
11076	1998-05-06	Bon app'	12, rue des Bouchers
11074	1998-05-06	Simons bistro	Vinb'ltet 34
11075	1998-05-06	Richter Supermarkt	Starenweg 5
11072	1998-05-05	Ernst Handel	Kirchgasse 6
11071	1998-05-05	LILA-Supermercado	Carrera 52 con Ave. Bol¡var #65-98 Llano Largo
11073	1998-05-05	Pericles Comidas cl sicas	Calle Dr. Jorge Cash 321
11070	1998-05-05	Lehmanns Marktstand	Magazinweg 7
11067	1998-05-04	Drachenblut Delikatessen	Walserweg 21
11069	1998-05-04	Tortuga Restaurante	Avda. Azteca 123
11068	1998-05-04	Queen Cozinha	Alameda dos Can…rios, 891
11064	1998-05-01	Save-a-lot Markets	187 Suffolk Ln.
11066	1998-05-01	White Clover Markets	1029 - 12th Ave. S.

Part II

The clickable id on the first page should link to another page showing all details of the order, e.g.



Id: 11072

Date: 1998-05-05

Required: 1998-06-02

Shipped:

Name: Ernst Handel

Address: Kirchgasse 6

City: Graz

Zip: 8010

Country: Austria

Product	UnitPrice	Quantity	Discount	Price
Chang	19	8	0.00	152.00
Jack's New England Clam Chowder	9.65	40	0.00	386.00
Valkoinen suklaa	16.25	22	0.00	357.50
Wimmers gute Semmelkn�del	33.25	130	0.00	4322.50
Total				5218

You must provide a route to this page /orderdetails that can take a parameter, the order id, a show the order details.

You must add a "back" link to the page and a home link in a menu in the top of the page.

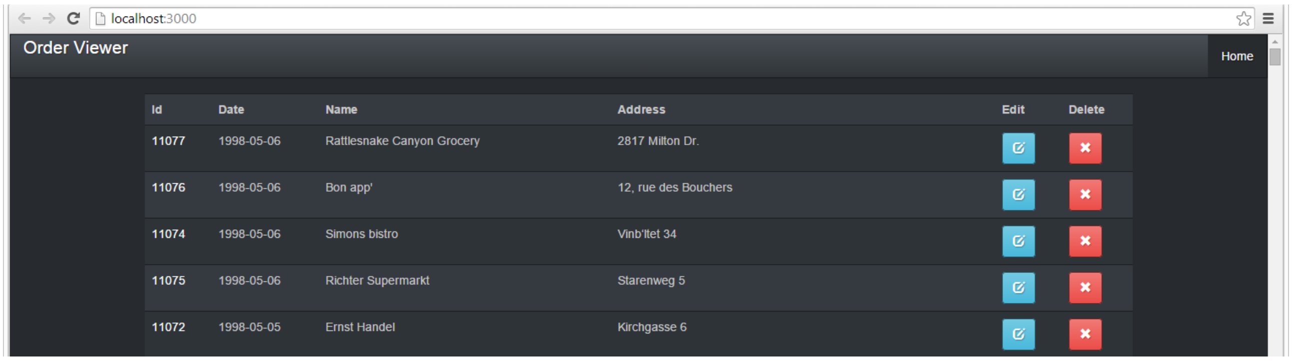
Bonus I: Add the customer and employee name to the details and make the clickable such that a click will lead to the details of customers and employee, respectively. Add a back link to these pages, too.

Bonus II: Add pages for listing categories, products, employees, and customers. Clicking on elements in the categories page should lead you to a list of products related to this category, i.e. clicking on the Beverages category should give you all products in this category. Clicking on elements in the products, employees, and customers list should lead to a list of orders related to these elements, i.e. clicking on the product Chai will give you all orders with that product.

Part III (optional)

Make the list pages[1] editable. Add links to the list such that you can choose to edit or delete the element (use bootstrap glyphs). A click on the delete button must ask for some kind of confirmation from the user, otherwise a web crawler could delete all you data ☹

[1] By list page, we mean a page that list elements, i.e. orders if you did not make any of the bonus tasks, otherwise categories, products, employees and customers.



Id	Date	Name	Address	Edit	Delete
11077	1998-05-06	Rattlesnake Canyon Grocery	2817 Milton Dr.	✎	✖
11076	1998-05-06	Bon app'	12, rue des Bouchers	✎	✖
11074	1998-05-06	Simons bistro	Vinb'ltet 34	✎	✖
11075	1998-05-06	Richter Supermarkt	Starenweg 5	✎	✖
11072	1998-05-05	Ernst Handel	Kirchgasse 6	✎	✖

Add a "add" button on top of each list pages that will bring you to a form where you can create new elements.