Retrospective Report

09.06.2023 – Lasse Andre Stavland

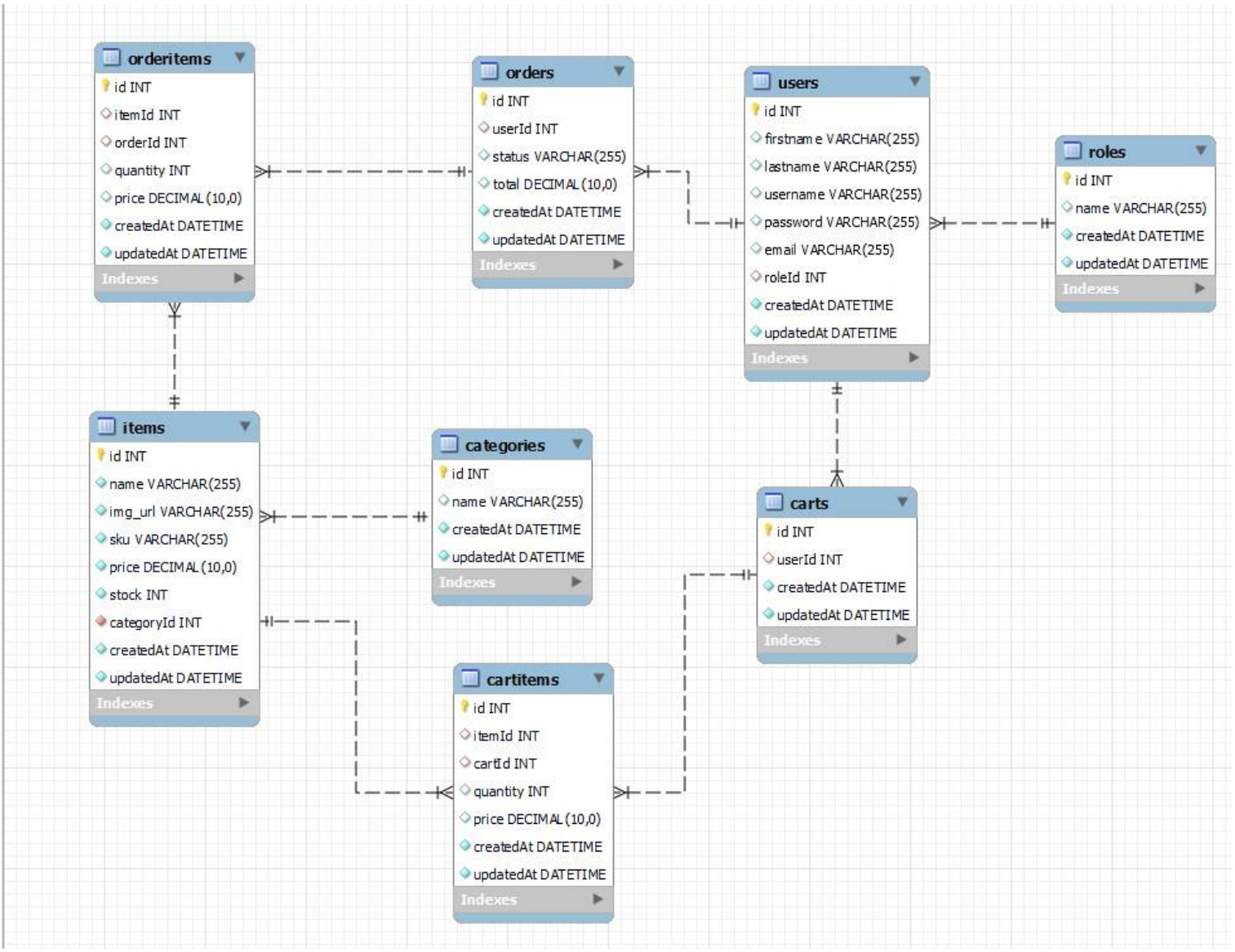
Class - AUG22FT || Back-end exam

Postman documentation for this project can be found by following this link:

- https://documenter.getpostman.com/view/26451965/2s93sc3XMr

There is also included a pdf of this in the documentation folder, but by following this link you get all the postman examples as well.

Screenshot of Database ERD



An explanation of the relationships between tables

User and Role:

- User belongs to Role, and Role has many Users.
- This represents a one-to-many relationship, where a Role can have multiple Users, but each User belongs to only one Role.

User and Cart:

- User has one Cart, and Cart belongs to User.
- This represents a one-to-one relationship, where each User has a Cart associated with them.

User and Order:

- User has many Orders, and Order belongs to User.
- This represents a one-to-many relationship, where a User can have multiple Orders, but each Order belongs to only one User.

Cart and CartItem:

- Cart has many CartItems, and CartItem belongs to Cart.
- This represents a one-to-many relationship, where a Cart can have multiple CartItems, but each CartItem belongs to only one Cart.

Item and Category:

- Item belongs to Category, and Category has many Items.
- This represents a one-to-many relationship, where a Category can have multiple Items, but each Item belongs to only one Category.

Item and CartItem:

- Item has many CartItems, and CartItem belongs to Item.
- This represents a one-to-many relationship, where an Item can be associated with multiple CartItems, but each CartItem belongs to only one Item.

Item and OrderItem:

- Item has many OrderItems, and OrderItem belongs to Item.
- This represents a one-to-many relationship, where an Item can be associated with multiple OrderItems, but each OrderItem belongs to only one Item.

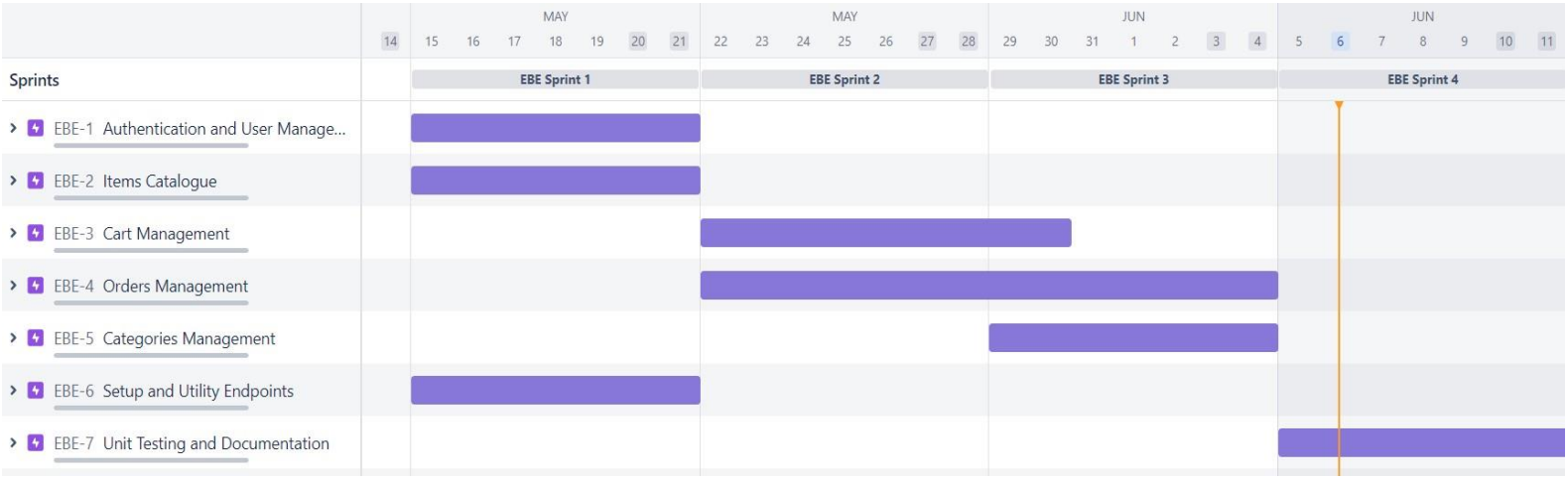
Order and OrderItem:

- Order has many OrderItems, and OrderItem belongs to Order.
- This represents a one-to-many relationship, where an Order can have multiple OrderItems, but each OrderItem belongs to only one Order.

Role and User:

- Role has many Users, and User belongs to Role.
- This represents a one-to-many relationship, where a Role can have multiple Users, but each User belongs to only one Role.

Screenshot of Jira Roadmap



Retrospective thoughts!

Jira – Scrum:

First I created a Jira scrum board for this project, I divided everything into epics and issues. This made it easier for me to work on different parts of the app as I could just choose what to work on from the tasks on the board.

The app and database:

I started the project by creating a small skeleton app that populated the database with the api provided by Noroff. I ran this with a small test script at first, just to see that my sequelize code worked.

Then I created the user registration, the models and the associations. I created the middleware for the admin and middleware for the user and I created all routes and did some thorough testing.

The unit testing:

I started the unit testing by writing all the «describe» and «it» blocks/code. Then I installed **Jest** and **Supertest** to test the code I wrote.

There were a lot of failed tests at first and I tried changing a lot of things in my code, I tried creating test databases etc. to fix it.

But I realized after a while that the error was with the way I populated the database, it tried to populate the database before it had finished initializing the database. So I added a timeout to the /setup request in my unit testing and after that, everything worked perfectly. So it was an easy fix in the end.

Postman documentation:

Most of my testing during database/endpoint setup was done with Postman, after I was done with the project I created the documentation with a Postman collection and I tested everything. The advantage I had with using Postman for documentation was that I got to test a lot of edge cases and that made me find multiple requests that would crash the application so which made me create more validation and error catching code to keep the app from crashing due to wrong body json etc.

I created and published every endpoint in my project with Postman, and I have included a pdf file of that in the project files under the documentation folder. I also provided a link to the actual documentation with examples at the top of this pdf.

I also created an extensive ReadMe file with all information needed for running and installing the project. I put a lot of effort into creating and styling the ReadMe file.

Challenges faced during the development of this project:

Here are some challenges I had:

- **Populating the database during unit testing:**

  I had a lot of trouble with the unit testing, with all tests failing no matter what I did.

  I then realized with some console.log statements that the error was with the /setup part of my test. It tried to run that endpoint before the database had finished its initialization, so I added a timeout to it and it worked perfectly.

- **Fullname was needed in user registration:**

  Found out late in development that full name was needed in user registration for the admin to be able to see the full name on orders.

  This was easily fixed by editing the user model and some endpoints.

- **I created the wrong endpoints**

  At first I created endpoints like:

  - /authentication/login
  - /items/item/:id
  - /categories/category

  But then after some more fine reading of the exam task and talking to teachers on teams, I found out that most endpoint routes were wrong so I fixed it by splitting all routes into their own router files.

Summary:

After being done with the exam, I feel that it was a fine project to test everything we have learned. I got to repeat some of the stuff we learned at the start of this study year and I think the project itself was very beneficial to my understanding of programming.

It did not contain any front-end, and that would be fun to do as well but I do realize that this is a Back-end class so that is fine.

I hope that what I have delivered here has lived up to what is expected from a student after a year of study. I did my best.

Best regards

Lasse André Stavland