

REFLECTION REPORT FOR EXAM PROJECT.

JIRA ROADMAP AND SPRINTS

I want to start with showing the roadmap since that also was where I started my project. I really appreciated that this was a part of the task since it helps break the quite large project into smaller sprints. This allowed me to plan the workload to fit with my calendar and help me keep a sense of where I stood in relation to my goal. It was also a good way to keep track and know if I was ahead of schedule and could take a few days off and focus on my family for a weekend or a couple of nights.

At the same time, it wasn't that straight forward. I'm not very experienced so knowing what would take more time and what will take less time wasn't that easy. More than once did I meet a fork in the road (or in my code) that didn't reveal itself at once. And when I solve one issue, another soon appeared.

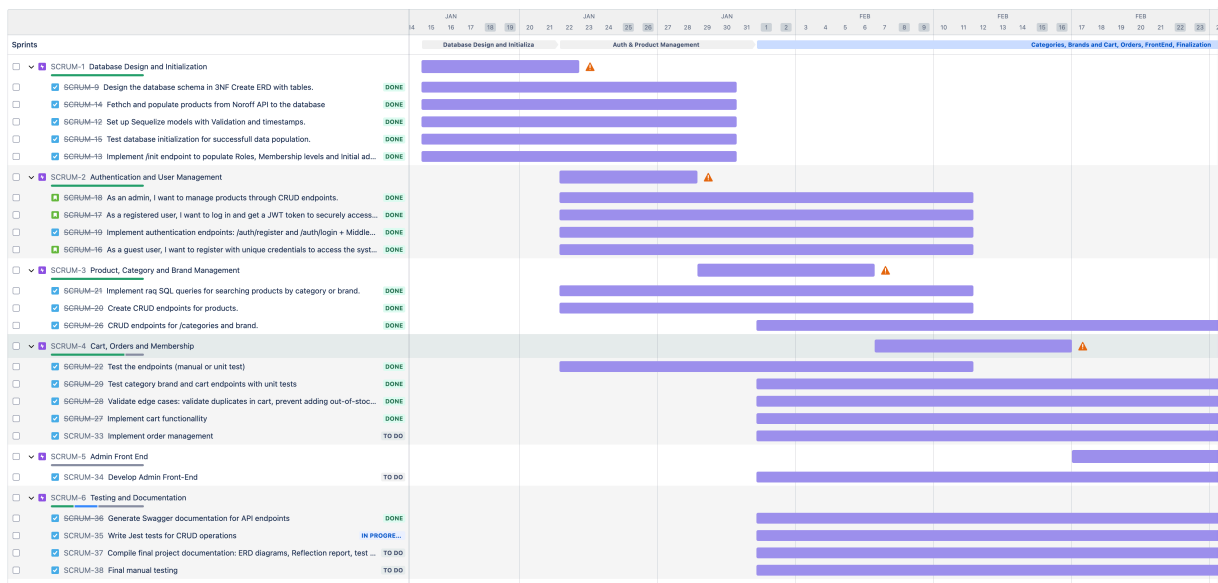


FIGURE 1JIRA ROADMAP WITH SPRINTS

DATABASE ERD

Users table:

- Each Users has one Role. The roleId is a foreign key in the user table.
- Each User has one Cart from the Carts table. userId is a foreign key in Carts.
- Each Users can have many Orders.
- Each Users can have one or more Memberships via UserMemberships.

Carts table:

- Each Users has one Cart.
- Each Cart can have many CartItems.

CartItems table:

- Each CartItem belongs to one Cart.
- Each CartItem references one Product.

Products table:

- Each Product belongs to one Brand.
- Each Product belongs to one Category.
- Each Product can be linked to many OrderItems.

Brands table:

- Each Brand can have many Products.

Categories table:

- Each Category can have many Products.

OrderItems table:

- Each OrderItem can be linked to many Products.
- Each OrderItem belongs to one Order.

Orders table:

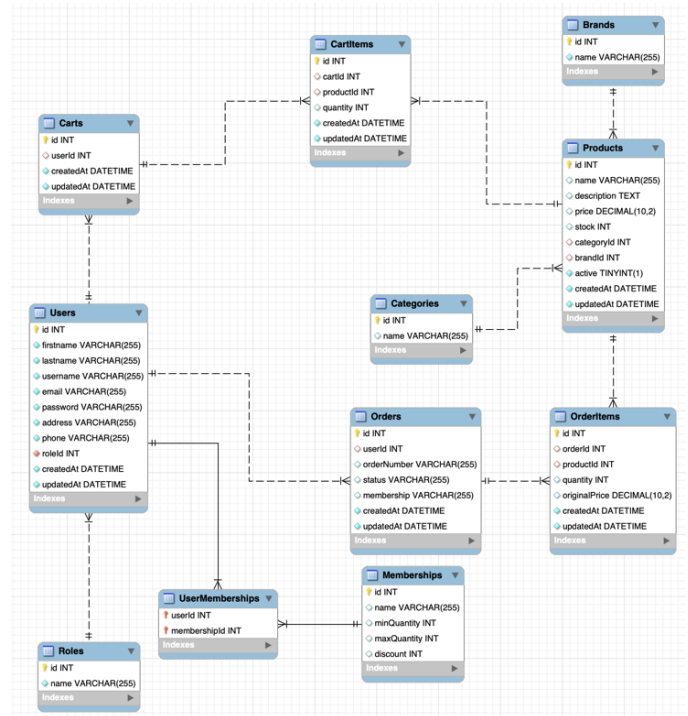
- Each Order can have many OrderItems.
- Each Order belongs to a User.

Roles table:

- Each Role can be assigned to many Users.

Memberships table:

- Users can have one or more Memberships via UserMembership.



The membership table defines the different levels of membership levels, while the UserMembership table takes each userId and puts it together with the Membership defined in the membership table after first order has been placed. Orders is associated with the Membership table by having a membership field which records the user's membership level at the time of purchase.

THE PROCESS

I really enjoyed the start, setting up tables, thinking about how to make it as efficient and good as possible. This is the skeleton everything else is leaning on. Once the database was setup and was working I could move on to work on setting up the Back-End.

During CRU implementation, I had to make some decisions around how to divide what actions admin was going to be able to perform and what the regular user was going to be able to do. I decided to keep all admin CRUDs inside admin.js while the other CRUDs in their respective routes. I'm not sure if that will be my go-to next time, because it became a rather large file. But it worked out nicely and I actually found it quite easy to work with when implementing the Swagger documentation as well.

TESTING AND DOCUMENTATION

Testing was a bit of a struggle. I got help from ChatGPT to steer me in the right direction to debug and make the tests pass. Writing the swagger documentation was something I quite enjoyed and found it useful to do at the same time as preparing Postman tests that I kept using throughout the entire project.

FRONT-END

Front-End was mostly fun. I went at it and prepared endpoints for most CRUDS. Later I noticed that according to the task description this was not optimal. Due to a house leak in our home, I had been spending some time renovating and upon time of noticing the miss reading of the task, I decided to stay at it with the endpoints just to keep the workflow and not having to change the setup I had prepared.

While working with implementing all the Front-End CRUD's I noticed some issues that were not so obvious when working Back-End. One thing was that membership levels didn't update correctly. ChatGPT suggested to implement another table called UserMembership. This table would serve as a bridge between Users and Memberships so the system could track a user's total purchased items and assign the correct membership. This solved it enough to move on.

USE OF CHATGPT

AI showed up out of nowhere shortly after starting this study. It has been a learning partner/teacher that always has time. When studying in the evenings and weekends, when teachers are unavailable, AI has been a really helpful tool. It has helped to elaborate on difficult subjects, check my explanations of things when I think I got it, debugging my code, and helping me find the right direction when I'm stuck. However, it has also been quite useful to see that sometimes AI just goes in circles in trying to locate your problem and you are actually dependent on having to figure some things out the hard way. I have found it surprising that AI has not been mentioned or discussed in our courses (though I guess it is implemented for future students already). Given how useful AI is as a coding tool and has quickly become is such a monumental part of the coding industry, I believe it is essential to teach students how to use it effectively. When it has not been discussed how to use it in a good way, we have had to figure out the way our selves.

All in all, it has been a good learning experience setting up such a large project!