



HOBBY PROJECT: FANTASY DATABASE

Callum Robinson
22FebEnable3

Intro & Concept

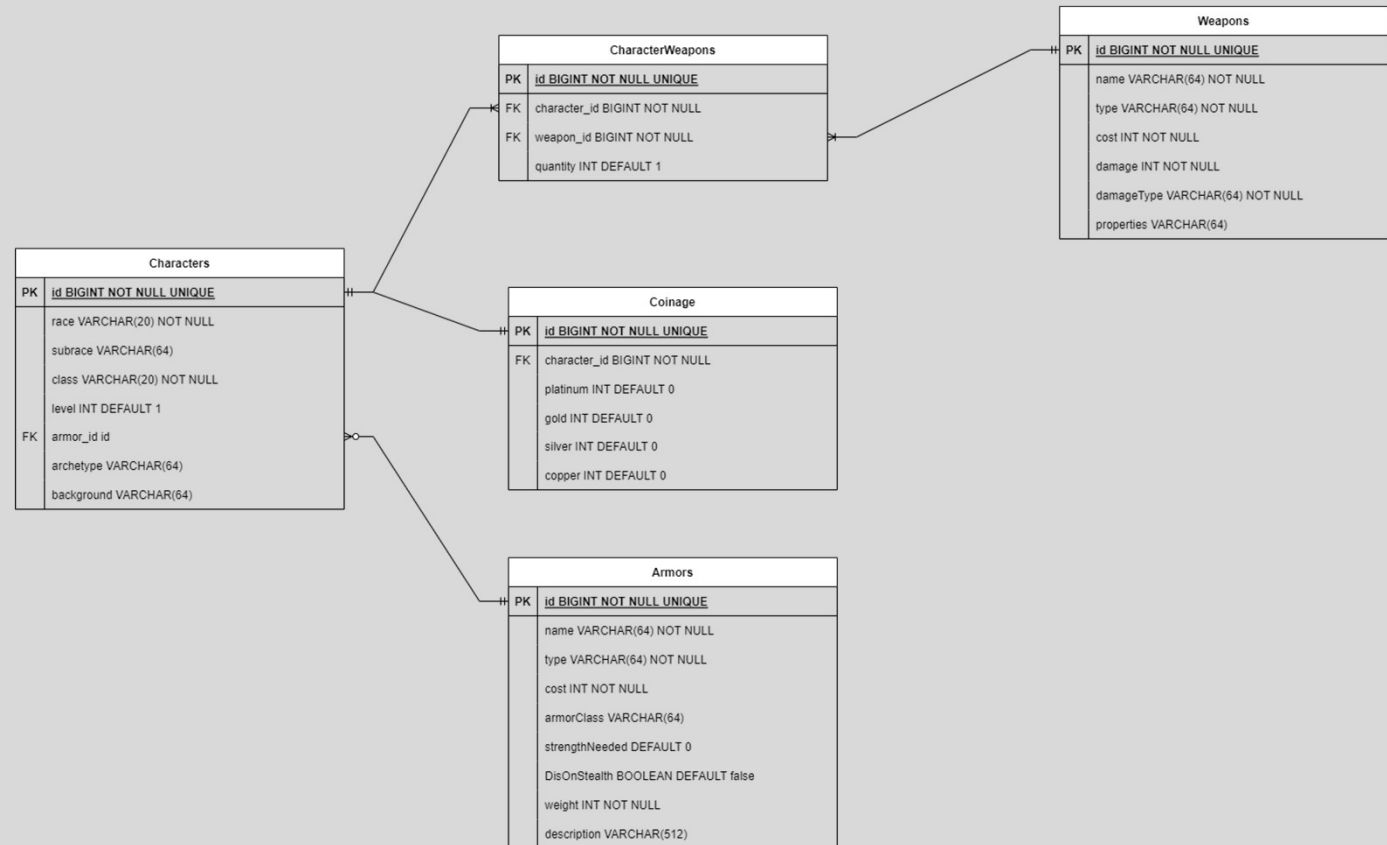
- One place to store all your ideas for characters and weapons so that they can be used in the future during campaigns or one-shots
- Can be used by both players and DMs running sessions
- The data can easily be read or updated when events happen during a session

Risk Assessment

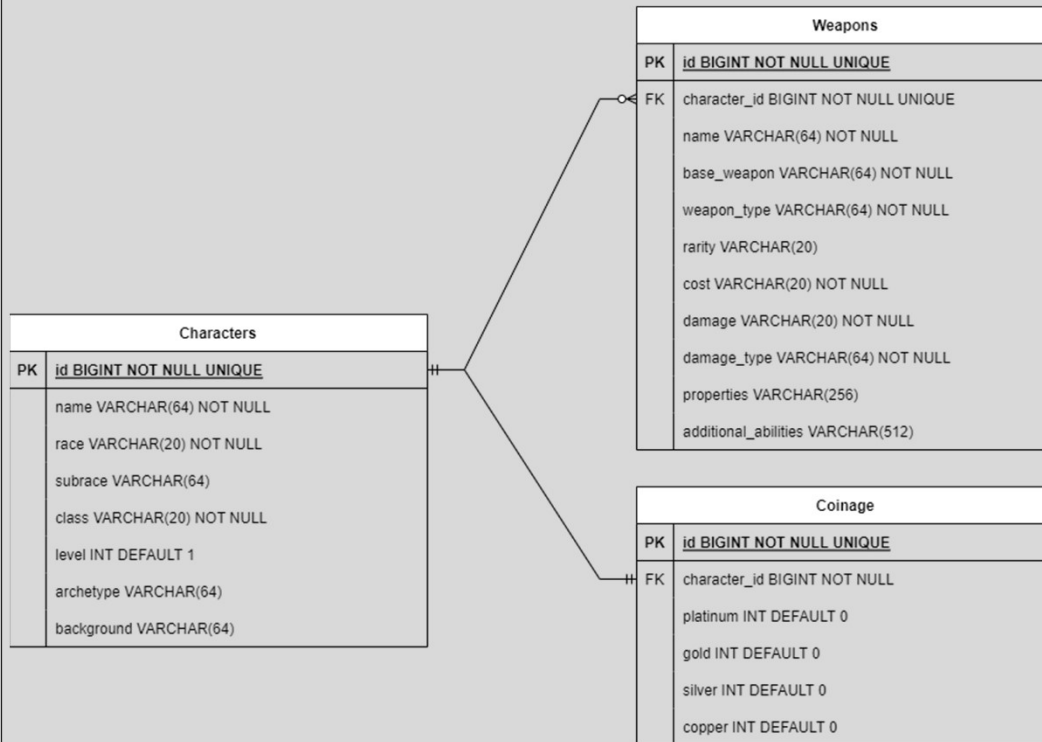
Description	Evaluation	Likelihood	Impact Level	Responsibility	Response	Control Measures
Internet outage	Lose access to repository and online resources	Medium	Medium	Internet Provider	A mobile hotspot maybe used as backup	Ensure a local copy is kept up to date with the repository
Github outage	Lose access to repository	Low	Medium	Github/Microsoft	Work on a local copy until able to reconnect	Ensure a local copy is kept up to date with the repository
Power outage	Lose any changes made since last save and any local changes lost until power is restored	Low	High	Electricity Supplier	Work on laptop from repository after IDEs setup	Ensure any work is saved, committed and pushed regularly
Computer crash	Lose any changes made since last save and any local changes if computer cannot be restarted	Low	High	Myself and supplier of equipment	Attempt to get the PC up and running again and continue from where possible. If PC cannot be restarted then find an alternative	Ensure work is pushed to repository regularly and ensure equipment is looked after and kept updated
Bugged version pushed to main branch	Application may cause errors until resolved or rolled back	High	Medium	Myself	Roll the version back until the problem is resolved	Keep to a main/develop/feature branch model with regular small commits
Run out of time	Features missing and MVP not met	Medium	High	Myself	Explain the failure in the requirements	Good use of kanban board with prioritisation for effective time management
Requests blocked by CORS	Fetch requests blocked by CORS policy	High	High	Myself	Give back the error response	Add permissions for cross origin requests on the controllers
User inputs invalid	User inputs invalid data into dataform	High	Low	User	Send back an error response	Add restrictions on inputs and ensure exception handling in controllers

ERDs

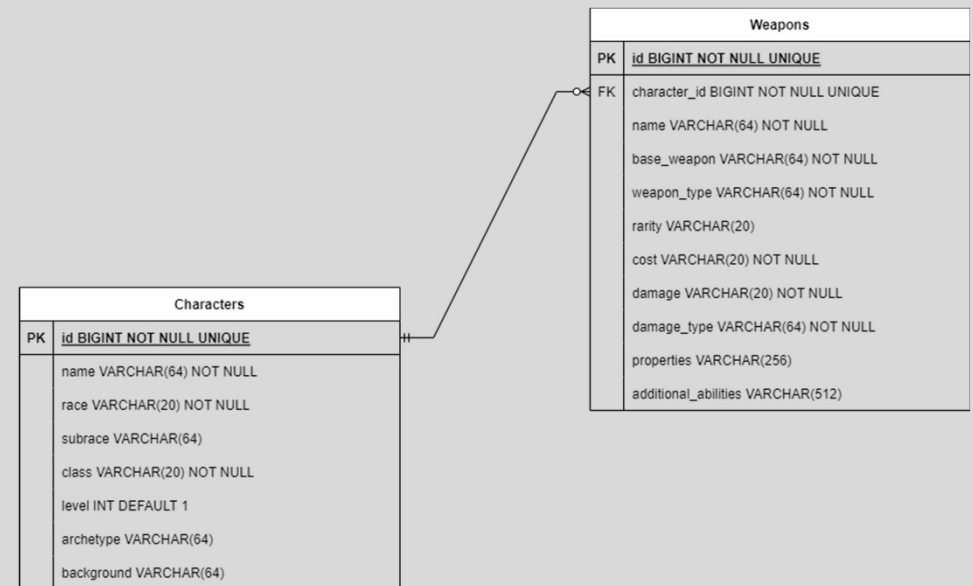
- Initial ERD concept
- Out of scope for this project



- Planned ERD with 3 Entities



- Final ERD with 2 Entities



Sprint planning

Let us go to the created Kanban Board:

<https://callumjrobinson.atlassian.net/jira/software/projects/HP/boards/3>

Journey

- Version Control
- Sprint Planning / Planning Boards
- Database Management
- Java
- Spring
- HTML
- CSS
- JavaScript
- Maven
- Junit and Mockito
- Lombok (independent)

Version Control

- Created two empty repositories on Github
- Instead of cloning the back-end it made more sense to push a starter spring application

```
$ git init|
```

```
$ git remote add origin https://github.com/Callum-Robinson/hobby-project-backend.git|
```

- Then create new branches following a main/dev/feature type model

```
$ git checkout -b configuration-and-profiles|
```

```
$ git push -u origin configuration-and-profiles|
```

- Stage, commit and push and changes

```
$ git add .|
```

```
$ git commit -m "added model mapper to the pom and created configuration file"|
```

```
$ git push|
```

- <https://github.com/Callum-Robinson/hobby-project-backend/tree/develop>

Testing

- Unit Tests were created for both service classes
- This include all CRUD functionality and their exceptions
- Achieved 82% coverage across src/main/java only missing controllers
- Lombok annotations were ignored by the coverage using the following:

```
1 config.stopBubbling = true  
2 lombok.addLombokGeneratedAnnotation = true
```

- A quick demonstration

Demonstration Time

- The locally-hosted front end: <http://127.0.0.1:5500/index.html>
- Lets try-out some of the user stories for the different CRUD functionalities

Sprint Review

- What was completed
 - The MVP of the application concept
 - Full CRUD functionality for both entities
 - Styling added to front end with some links to additional resources a user could want
- What was left behind
 - The third entity 'Coinage' and fourth entity 'Armour'
 - Integration tests for the controllers
 - Use of a static analysis tool

Sprint Retrospective

- What went well
 - Board and diagrams planned well
 - All wanted CRUD working as intended
 - Good progress and time management working through the back-end
- What could be improved
 - Better planning to deal with bugs especially in the front end
 - Integration tests to be done
 - Added in the 3rd entity functionality

Conclusion

- Very happy with current application using everything used so far in this course
- Increased understanding of everything covered so far
- Increased experience with debugging and finding errors
- The application can be expanded upon with further functionality
- More practice with the front-end is needed especially JavaScript and CSS

Questions?