
T22 - Exam Database



For: Marisa Duval, Donna Williams, Kim Rozon and the Faculty of Medicine.

Our team



- Alex Nadeau - Front-end developer / Prototype
- Elias Maalouf - Full stack developer / Code reviewer
- Joseph Abonasara - DevOps / Database design
- Michael Malek - Back-end developer / Architecture design
- Ricardo Saikali - Front-end developer / Prototype / Database design
- Stefano Stella - Back-end developer / Architecture design

Clients

- Marisa Duval - Project Manager
- Donna Williams - UG Medical Education Coordinator
- Kim Rozon - Postgraduate Program Manager
- Faculty of Medicine



Objective Overview (1/2)

- Create a secure and accessible exam database for the Department of Family Medicine (DFM) that is password protected, allowing only authorized personnel to access and modify exam data.
- Enabling easy creation of new exams and the ability to update existing questions in the exam database while maintaining a version history.
- Allowing exam administrators to manage examination learning objectives/requirements.



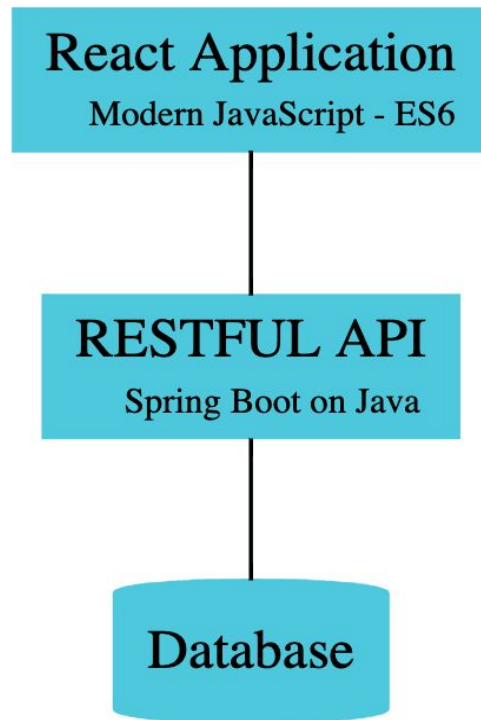
Objective Overview (2/2)

- Maintaining a historical record of all exams administered, including the date of administration, the questions asked, the exam's learning objectives, and any analysis information regarding the questions asked.
- Provide the ability to identify which questions have been used in previous exams and track how often they have been used to avoid redundancy and improve tests using analytics from the faculty.
- Exporting generated exams to word document format that follow client's specifications.

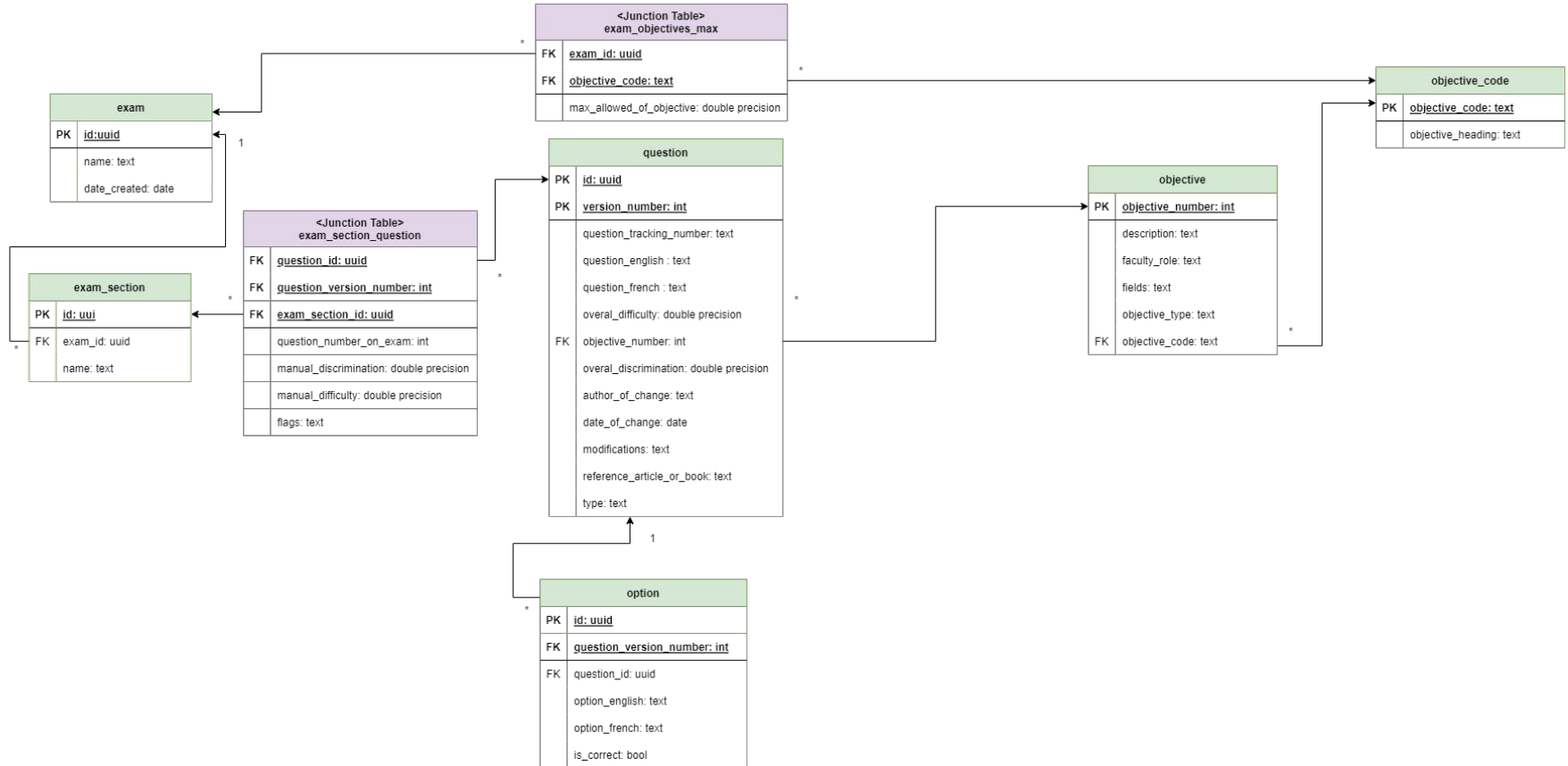


Design and Architecture (1/4)

- How our final db design came to be:
 - Many meetings with the clients → Requirements
 - Many changes to our database design (Several Iterations)
- The architecture of our product:
 - **Backend:** Spring Boot
 - **FrontEnd:** ReactJS
 - **Database:** PostgreSQL

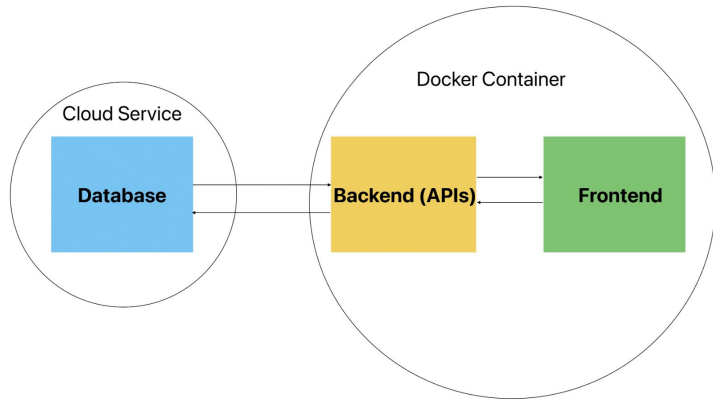


Database Design

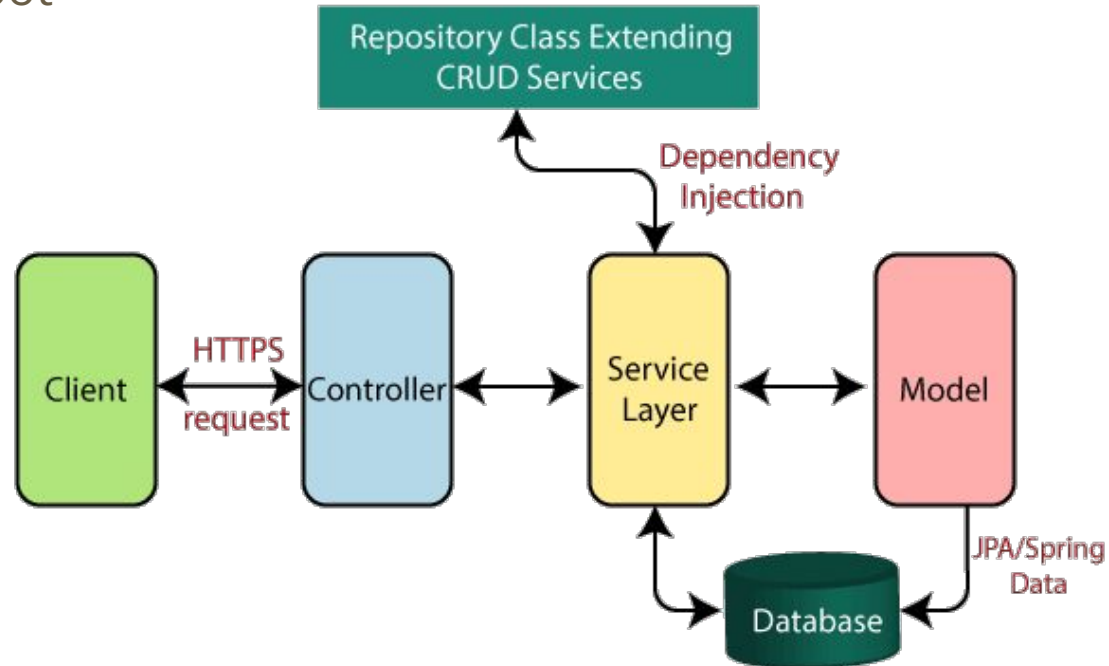


Design and Architecture (2/4)

- **Backend/Server:** Spring Boot



Spring Boot flow architecture



Design and Architecture (3/4)

- **Frontend:** React
- **Backend/Server:** Java Spring
- **Database:** PostgreSQL



Design and Architecture (4/4)

- Why these technologies?

- Spring Boot:**

- Robust, scalable, and maintainable back-end for the exam database
 - Previous experience with Spring

- React:**

- Library for building modern, responsive, and user-friendly front-ends
 - Component-based architecture for creating reusable UI elements

- PostgreSQL:**

- Relational database management system
 - Provides reliable, scalable, and feature-rich data storage solution
 - Previous experience with PostgreSQL

Challenges and lessons learned (1/2)

Challenges:

- Not familiar with the tech used / learning curve
- Many technical challenges faced on each step of the project
- UI implementation was much harder than anticipated
- Making a backend design that suits the frontend, domain, use cases was challenging
- Gathering requirements and defining the scope of the project
- Creating an efficient and user-friendly search engine that allows for filtering and sorting exam questions can be complex.
- Designing UI for optimal UX
- Designing database for optimisation and changing requirements

Challenges and lessons learned (2/2)

Lessons learned:

- Learned how to make a full scale web app from scratch
- Always try to use reliable libraries when possible (don't reinvent the wheel)
- Learned many new technical skills while working with unfamiliar technologies
- Learned how to set up professional client meetings and extract clear requirements and objectives from them
- Learned agile development methodologies, can help break down the project into manageable chunks and allow for frequent communication and feedback with stakeholders.

DEMO



LOADING...