

Contents

1	introduction	3
2	context	3
3	objective	3
4	concepts and technologies	3
4.1	front-end	3
4.1.1	typescript	3
4.1.2	html	3
4.1.3	css	3
4.1.4	scss	3
4.1.5	angular	3
4.1.6	angular material	3
4.1.7	Sb-Angular-Material	3
4.2	back-end	3
4.2.1	java	3
4.2.2	spring	3
4.2.3	rest	4
4.2.4	mariadb	4
4.2.5	ldap	4
4.3	development	4
4.3.1	apache netbeans	4
4.3.2	maven	4
4.3.3	lombok	4
4.3.4	apache directory studio	4
4.3.5	visual studio code	4
4.3.6	docker	4
4.3.7	docker-compose	4
4.3.8	mariadb	4
4.3.9	chinook database	4
4.3.10	angular cli	4
4.3.11	firefox	4
4.3.12	postman	5
5	project	5
5.1	use-cases	5
5.2	class diagram	5

5.3	template angular-material	5
5.4	TODO multi-database support	5
6	implementation and results	5
6.1	front-end	5
6.1.1	component structure	5
6.1.2	generic form controll builder	5
6.1.3	spring HATEOAS classes	5
6.1.4	temporal caching for repository index method	5
6.1.5	error handler	5
6.1.6	database reader	5
6.2	back-end	5
6.2.1	java classes	5
6.2.2	spring configuration	6
6.2.3	custom controllers & view models	6
6.2.4	spring repositories	6
6.2.5	orm generated database	6
6.2.6	multi-database support	6
6.3	development environment	6
6.3.1	Apache Directory	6
6.3.2	multi-database support	6
6.3.3	postman tests	6
7	conclusion	6
8	future work	6

- 1 introduction
- 2 context
- 3 objective
- 4 concepts and technologies
 - 4.1 front-end
 - 4.1.1 typescript
 - 4.1.2 html
 - 4.1.3 css
 - 4.1.4 scss
 - 4.1.5 angular
 - 4.1.6 angular material
 - 4.1.7 Sb-Angular-Material
 - 1. project structure
 - 4.2 back-end
 - 4.2.1 java
 - 4.2.2 spring
 - 1. dependency injection
 - 2. boot
 - 3. data
 - 4. web
 - (a) HATEOAS
 - 5. security

4.2.3 rest

4.2.4 mariadb

4.2.5 ldap

4.3 development

4.3.1 apache netbeans

4.3.2 maven

4.3.3 lombok

4.3.4 apache directory studio

4.3.5 visual studio code

4.3.6 docker

4.3.7 docker-compose

4.3.8 mariadb

4.3.9 chinook database

4.3.10 angular cli

4.3.11 firefox

1. debugger

2. webpack

4.3.12 postman

5 project

5.1 use-cases

5.2 class diagram

5.3 template angular-material

5.4 TODO multi-database support

6 implementation and results

6.1 front-end

6.1.1 component structure

1. services
2. modules
3. dialog

6.1.2 generic form controll builder

6.1.3 spring HATEOAS classes

1. entity class
2. acessor class
3. repository class
4. repository service class

6.1.4 temporal caching for repository index method

6.1.5 error handler

6.1.6 database reader

6.2 back-end

6.2.1 java classes

1. view models

6.2.2 spring configuration

1. ldap
2. security
 - (a) admin resources
3. cors mapping
4. user details context mapper

6.2.3 custom controllers & view models

1. database reader
2. permission tree delete method
3. user
4. queries
5. permissions

6.2.4 spring repositories

6.2.5 orm generated database

6.2.6 multi-database support

6.3 development environment

6.3.1 Apache Directory

6.3.2 multi-database support

1. microsoft sql server docker image
2. oracle:xe driver access
3. testing file

6.3.3 postman tests

7 conclusion

8 future work