Web to do application notes:

28/01/21

minimal html code

use html primarily for design and layout of front-end

minimal functionality within html

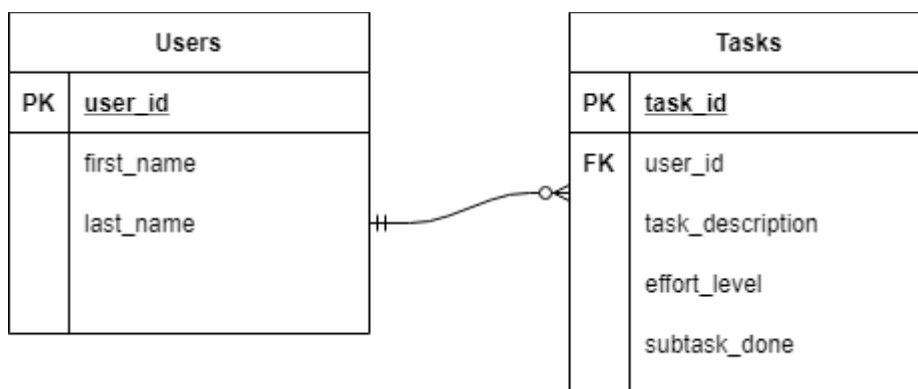functionality focused on in JS

30/2/21

Must have: ability to enter multiple entries into to list

Could have: categories for list entries, priority/effort amount for each task

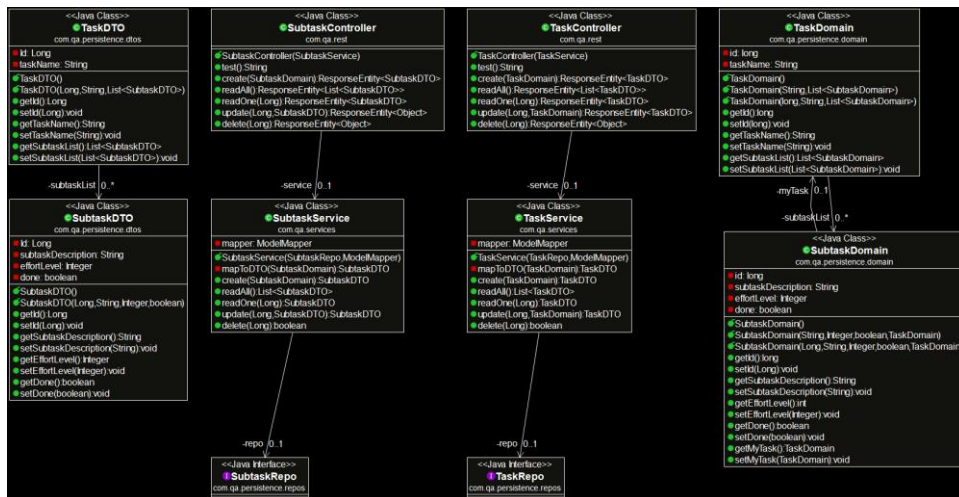Could have: subtasks

Sprint 1 (from 4/2/21):

Backlog

Backlog 14 issues		**Create sprint** ···
As a user, I want to create a task, so that I can store what needs to be done	TDL-1 ↑	3
As a user, I want to read a task, so that I can view tasks needed to be done	TDL-2 ↑	1
As a user, I want to update a task, so that I can change whether it has been completed	TDL-3 ↑	5
As a user, I want to delete a task, so that I can remove any unwanted tasks	TDL-4 ↑	2
As a user, I want to create a subtask, so that I can split a task into smaller ones to track	TDL-5 ↑	3
As a user, I want to view subtasks so I can see all subtasks within a task	TDL-6 ↑	1
As a user, I want to update a subtask, so that I can change whether it has been completed	TDL-7 ↑	5
As a user, I want to delete a subtask, so that I can remove unwanted subtasks	TDL-8 ↑	2
SubtaskService Unit Test	TDL-9 ↑	13
TaskService Unit Test	TDL-10 ↑	13
SubtaskController Integration Test	TDL-11 ↑	21
TaskController Integration Test	TDL-12 ↑	21
Task page	TDL-13 ↑	-
Subtask page	TDL-14 ↑	-

+ Create issue

Postman

File Edit View Help

Home   Workspaces ∨   Reports   Explore   Search Postman   No Environm

POST http:... GET http:/... GET http:/... PUT http:/... DEL http:/... GET http:/... POST http:... + ○○○

http://localhost:8080/task/readAll   Save

GET ∨   http://localhost:8080/task/readAll

Params   Authorization   Headers (6)   Body   Pre-request Script   Tests   Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body   Cookies   Headers (5)   Test Results   Status: 200 OK   Time: 18 ms   Size: 600 B

Pretty   Raw   Preview   Visualize   JSON ∨

```
12          },
13          {
14              "taskName": "Cleaning",
15              "subtaskList": [
16                  {
17                      "subtaskDescription": "Buy vacuum",
18                      "done": true,
19                      "id": 2
20                  },
21                  {
22                      "subtaskDescription": "Clean the kitchen",
23                      "done": true,
24                      "id": 3
25                  }
26              ],
27              "id": 2
28          },
29          {
30              "taskName": "Admin Work",
31              "subtaskList": [
32                  {
33                      "subtaskDescription": "Sign documents",
34                      "done": true,
35                      "id": 4
36                  },
37                  {
38                      "subtaskDescription": "Update calendar",
39                      "done": true,
40                      "id": 5
41                  }
```

Find and Replace   Console

End of sprint 1:

Service unit tests completed for both Task and Subtask

Integration test completed for both Task and Subtask

Element	Coverage	ed Instructions	ed Instructions	tal Instructions
TDL-Project	98.6 %	1,957	28	1,985
src/main/java	94.2 %	456	28	484
com.qa.rest	88.7 %	94	12	106
com.qa	37.5 %	3	5	8
com.qa.persistence.doma	96.6 %	112	4	116
com.qa.services	97.5 %	156	4	160
com.qa.persistence.dtos	96.3 %	79	3	82
com.qa.config	100.0 %	12	0	12
src/test/java	100.0 %	1,501	0	1,501
com.qa	100.0 %	4	0	4
com.qa.rest	100.0 %	612	0	612
com.qa.services	100.0 %	885	0	885

Sprint 2 10/02/21 - 12/02/21:

Project Name:	TDL Project								
Project Description:	Functional web application to interact with a managed database								
Project Owner:	Claes Alfonso								

Risk ID	Risk area	Risk description	Probability			Risk Level			Project impact	Proximity / Length of impact	Response strategy
1	covid-19	Illness due to the covid-19 virus			X	X		X	100%	~14 days	Social distancing if in public areas
2	SQL data	Using test data instead of dev data		X				X	20%	Minimal project effect, however may take longer to get the correct results	Check the Spring application properties before booting the application
3	Computer crashing	Loss of data due to computer crashing or damage		X				X	100%	Long	push to github frequently
4	Merge conflicts within git	Git merge could create a merge conflict			X	X			20%	Short	Minimise duplicate items
5	Application breaks	Application could break due to errors or other software issues		X			X		100%	Medium	Break up larger functions into smaller pieces

Projects / TDL-Project / TDL board

Backlog

VERSIONS EPICS

TDL Sprint 2 10 issues   **Start sprint** | Plan sprint ⌄ | •••

🔖	Task page frontend layout	TDL-13 ↑	8
🔖	Subtask page frontend layout	TDL-14 ↑	8
🔖	GET method for Subtask	TDL-15 ↑	8
🔖	POST method for Subtask	TDL-16 ↑	13
🔖	PUT method for Subtask	TDL-17 ↑	21
🔖	DELETE method for Subtask	TDL-18 ↑	8
🔖	GET method for Task	TDL-19 ↑	8
🔖	POST method for Task	TDL-20 ↑	13
🔖	PUT method for Task	TDL-21 ↑	21
🔖	DELETE method for Task	TDL-22 ↑	8

+ Create issue

Trying to get values into separate table columns:

To Do List

Task Description:

Effort Level:

Create

subtaskDescription effortLevel done id

1

Buy utensils

Buy vacuum

Clean the kitchen

Sign documents

Update calendar

123

3

Row 1 Column 1   Row 1 Column 2   Row 1 Column 3

SonarQube build:

sonar:sonar -Dsonar.login=627e360c9d971e3ac6afa1c73c5fd7fb1cffe6b2

End of sprint 2

Backlog

🔍	CA 👤 Only My Issues Recently Updated

TDL Sprint 2 14 issues   ⓪ 0 144   Plan sprint ⌄   •••
08/Feb/21 9:00 AM • 15/Feb/21 5:30 PM

🟩 Task page frontend layout	TDL-13	↑	8
🟩 Subtask page frontend layout	TDL-14	↑	8
🟩 GET method for Subtask	TDL-15	↑	8
🟩 POST method for Subtask	TDL-16	↑	13
🟩 PUT method for Subtask	TDL-17	↑	21
🟩 DELETE method for Subtask	TDL-18	↑	8
🟩 GET method for Task	TDL-19	↑	8
🟩 POST method for Task	TDL-20	↑	13
🟩 PUT method for Task	TDL-21	↑	21
🟩 DELETE method for Task	TDL-22	↑	8
🟩 Selenium Automated Testing for Subtask	TDL-23	↑	21
🟩 Risk assessment	TDL-25	↑	3
🟩 UML Diagram after	TDL-26	↑	2
🟩 ERD diagram after	TDL-27	↑	2

VERSIONS

EPICS

Task

PK	item_id
	task_title

Subtask

PK	subtask_id
FK	task_id
	task_description
	effort_level
	subtask_done

<<Java Class>>
TaskDTO
com.qa.persistence.dtos

- id: Long
- taskName: String

- TaskDTO()
- TaskDTO(Long,String,List<SubtaskDTO>)
- getId():Long
- setId(Long):void
- getTaskName():String
- setTaskName(String):void
- getSubtaskList():List<SubtaskDTO>
- setSubtaskList(List<SubtaskDTO>):void

-subtaskList 0..*

<<Java Class>>
SubtaskDTO
com.qa.persistence.dtos

- id: Long
- subtaskDescription: String
- effortLevel: Integer
- done: Boolean

- SubtaskDTO()
- SubtaskDTO(Long,String,Integer,Boolean)
- getId():Long
- setId(Long):void
- getSubtaskDescription():String
- setSubtaskDescription(String):void
- getEffortLevel():Integer
- setEffortLevel(Integer):void
- getDone():Boolean
- setDone(Boolean):void

<<Java Class>>
SubtaskController
com.qa.rest

- SubtaskController(SubtaskService)
- test():String
- create(SubtaskDomain):ResponseEntity<SubtaskDTO>
- readAll():ResponseEntity<List<SubtaskDTO>>
- readOne(Long):ResponseEntity<SubtaskDTO>
- update(Long,SubtaskDTO):ResponseEntity<Object>
- delete(Long):ResponseEntity<Object>

-service 0..1

<<Java Class>>
SubtaskService
com.qa.services

- mapper: ModelMapper

- SubtaskService(SubtaskRepo,ModelMapper)
- mapToDTO(SubtaskDomain):SubtaskDTO
- create(SubtaskDomain):SubtaskDTO
- readAll():List<SubtaskDTO>
- readOne(Long):SubtaskDTO
- update(Long,SubtaskDTO):SubtaskDTO
- delete(Long):boolean

-repo 0..1

<<Java Interface>>
SubtaskRepo
com.qa.persistence.repos

<<Java Class>>
TaskController
com.qa.rest

- TaskController(TaskService)
- test():String
- create(TaskDomain):ResponseEntity<TaskDTO>
- readAll():ResponseEntity<List<TaskDTO>>
- readOne(Long):ResponseEntity<TaskDTO>
- update(Long,TaskDomain):ResponseEntity<TaskDTO>
- delete(Long):ResponseEntity<Object>

-service 0..1

<<Java Class>>
TaskService
com.qa.services

- mapper: ModelMapper

- TaskService(TaskRepo,ModelMapper)
- mapToDTO(TaskDomain):TaskDTO
- create(TaskDomain):TaskDTO
- readAll():List<TaskDTO>
- readOne(Long):TaskDTO
- update(Long,TaskDomain):TaskDTO
- delete(Long):boolean

-repo 0..1

<<Java Interface>>
TaskRepo
com.qa.persistence.repos

<<Java Class>>
TaskDomain
com.qa.persistence.domain

- id: long
- taskName: String

- TaskDomain()
- TaskDomain(String,List<SubtaskDomain>)
- TaskDomain(long,String,List<SubtaskDomain>)
- getId():long
- setId(long):void
- getTaskName():String
- setTaskName(String):void
- getSubtaskList():List<SubtaskDomain>
- setSubtaskList(List<SubtaskDomain>):void

-myTask 0..1

-subtaskList 0..*

<<Java Class>>
SubtaskDomain
com.qa.persistence.domain

- id: long
- subtaskDescription: String
- effortLevel: Integer
- done: Boolean

- SubtaskDomain()
- SubtaskDomain(String,Integer,Boolean,TaskDomain)
- SubtaskDomain(Long,String,Integer,Boolean,TaskDomain)
- getId():Long
- setId(Long):void
- getSubtaskDescription():String
- setSubtaskDescription(String):void
- getEffortLevel():int
- setEffortLevel(Integer):void
- getDone():Boolean
- setDone(Boolean):void
- getMyTask():TaskDomain
- setMyTask(TaskDomain):void