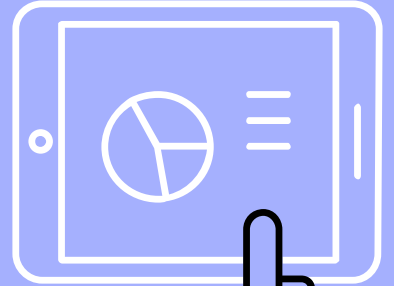
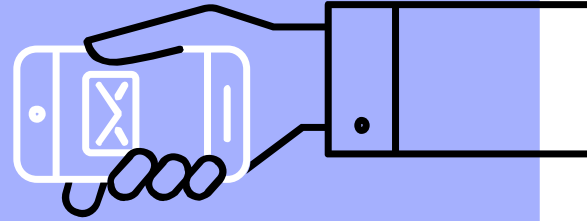
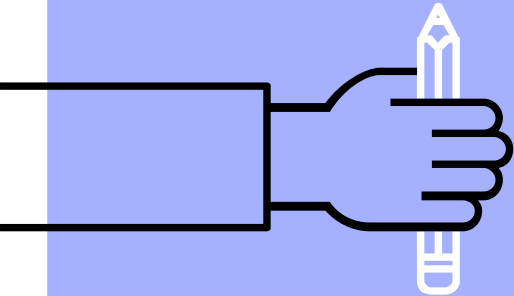
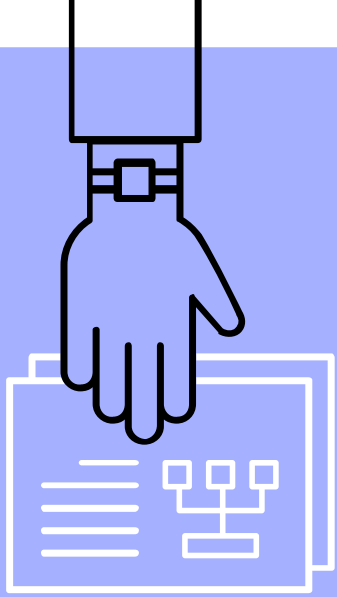


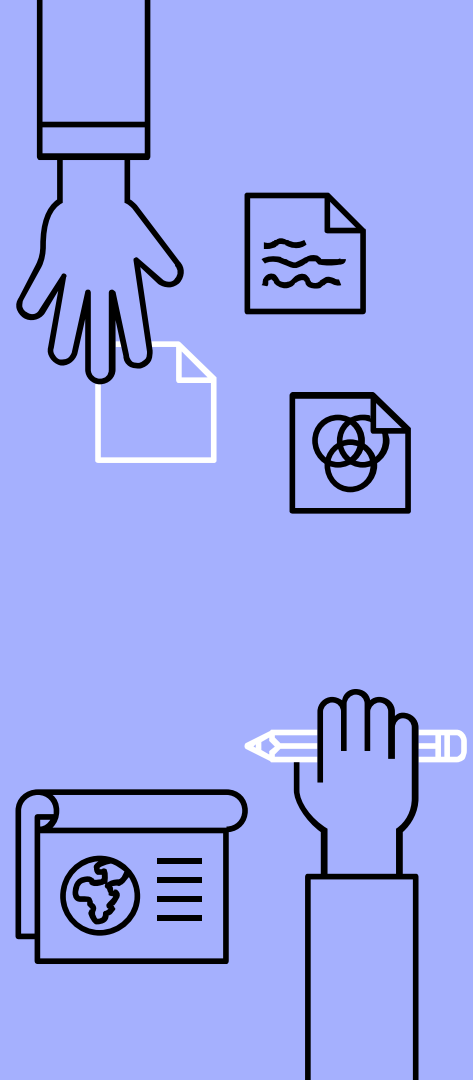
# To Do List Web App Project

By Gie-Anne Fortuna



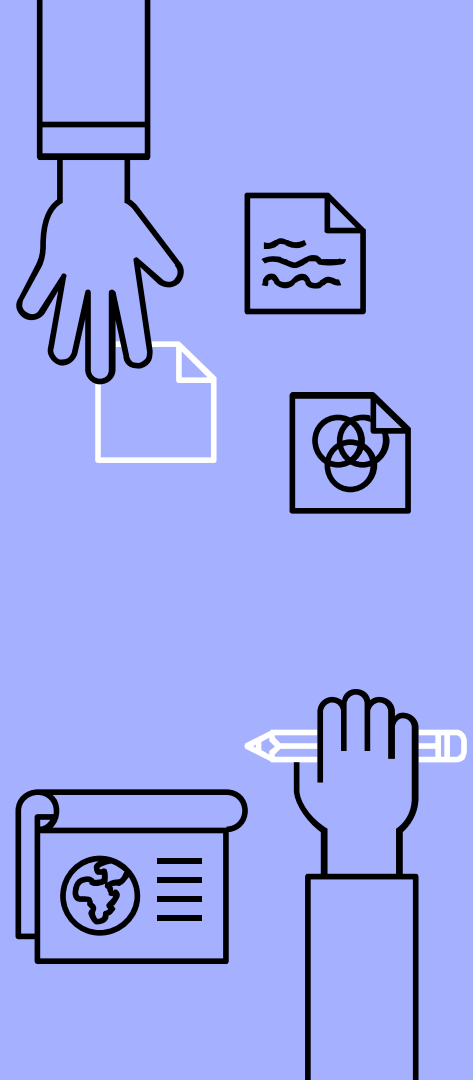
# MUST HAVE

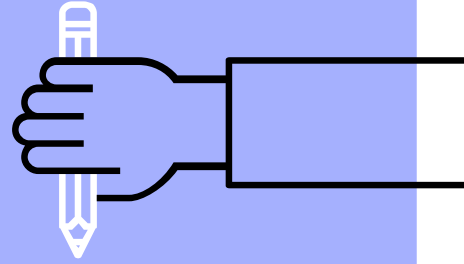
A functional web app that allows the user to perform CRUD functionalities to the two entities-Item and To Do List.



# SHOULD HAVE

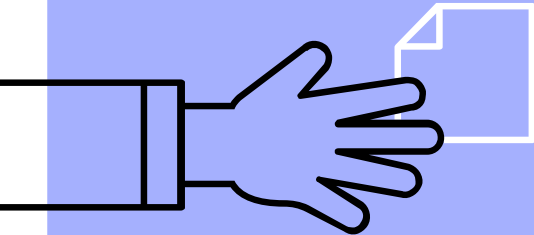
A web app that allows the user to CRUD tasks in the To Do List they belong to.





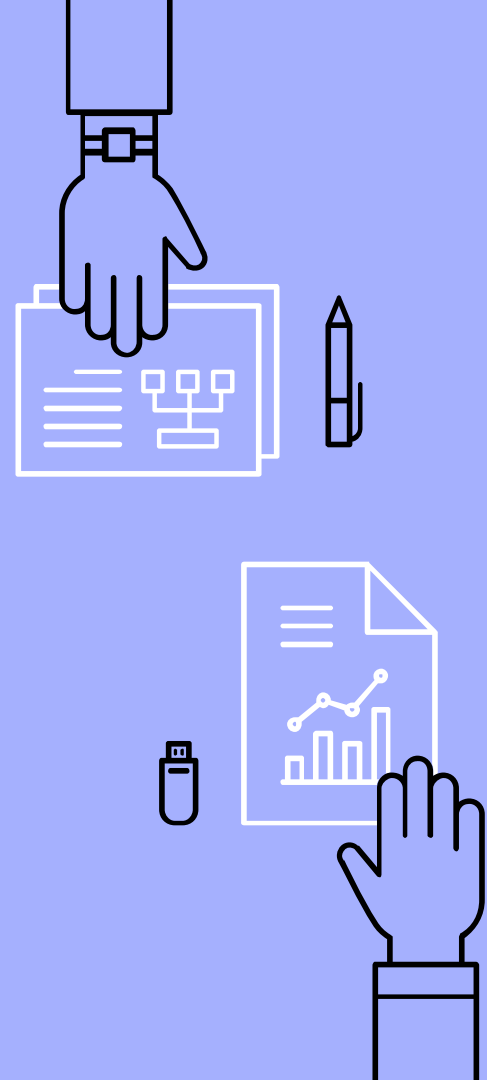
# TECHNOLOGIES

Here are the software used



- ▶ VCS/Code Management: Git/Github
- ▶ Project Management: Jira-Scrum
- ▶ Database Management: MySQL
- ▶ Back-end: Java
- ▶ **API Development: Spring**
- ▶ **Front-end: HTML,CSS,JavaScript**
- ▶ Build Tool: Maven
- ▶ Testing
  - Static Analysis: SonarQube
  - Unit/Integration: JUnit/**Mockito**
  - **User-Acceptance: Selenium**

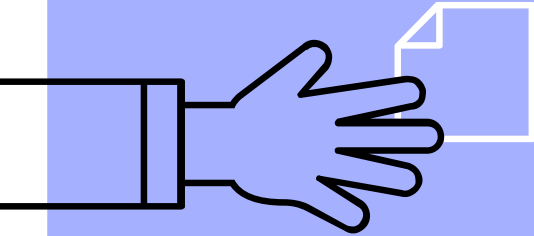
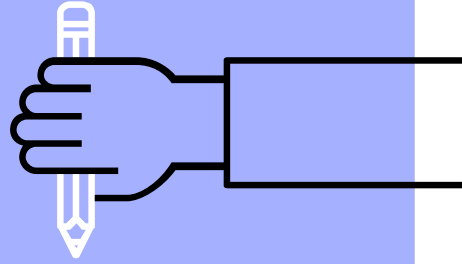
(New for this project)



“

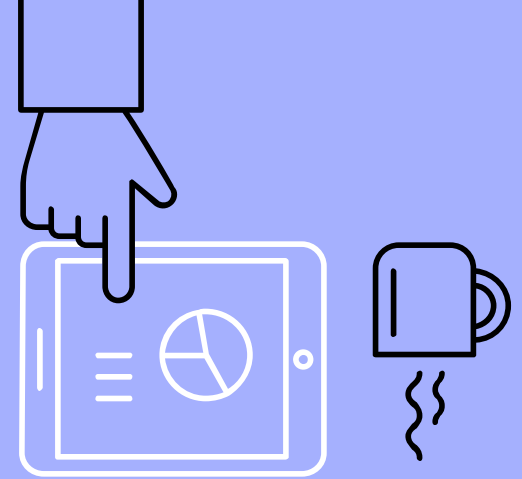
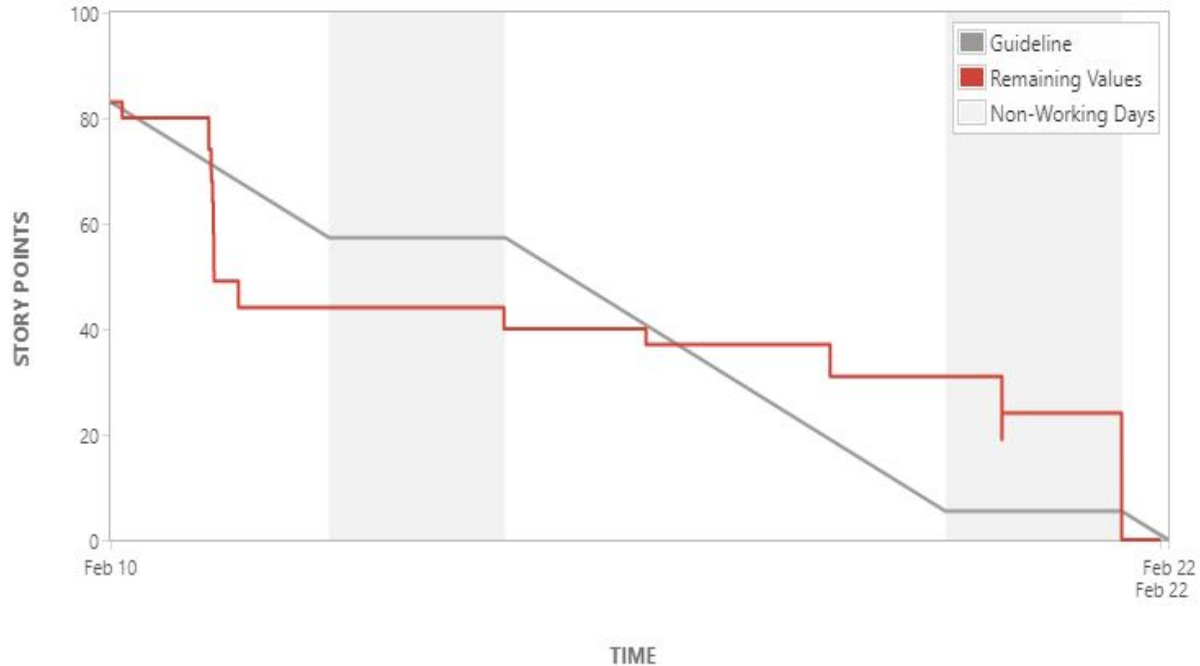
*A goal without a plan is  
just a wish.*

# PLANNING



PLANNING:

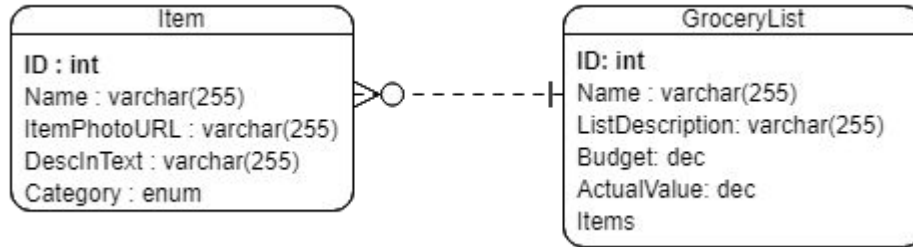
## Jira: Scrum board



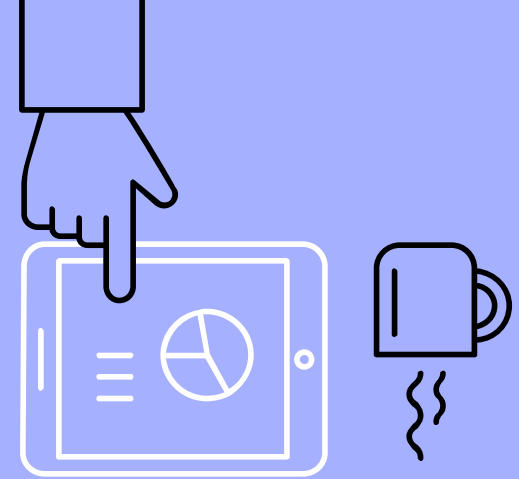


PLANNING:

# Entity Relationship Diagram (start)



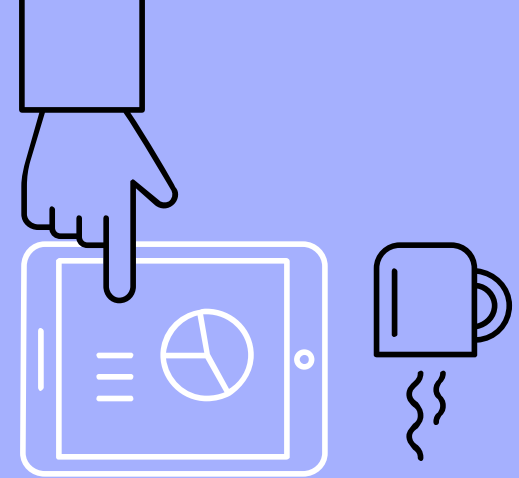
[final](#)



PLANNING:

# RISK ASSESSMENT

Risk	Likelihood	Impact	Risk Level
Incomplete Project	M	H	M
Equipment issues	M	M	L
Broken Frontend	L	H	L
Jar file	L	M	L
Project Spec Misinterpretation	M	H	M
Merge conflicts	M	M	L
Not understanding starter code	M	H	M



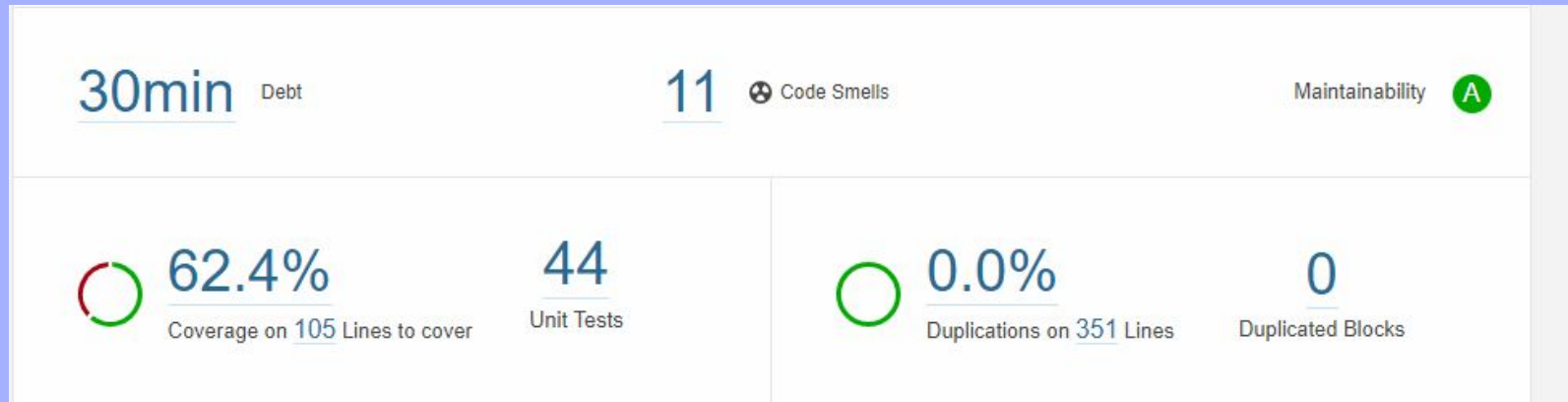
✓ Spring Boot Start

h2 database

demo































# DEMONSTRATION

# SONARQUBE COVERAGE



# STS COVERAGE as JUNIT 5

TDL\_Project (2) (22 Feb 2021 14:29:20)

Element	Coverage	Covered Instructio...	Missed Instructions	Total Instructions
 TDL_Project	 71.1 %	3,161	1,282	4,443
 src/test/java	 72.2 %	2,461	947	3,408
 com.qa.acceptance	 47.1 %	839	941	1,780
 com.qa.main	 85.7 %	36	6	42
 com.qa.main.controller	 100.0 %	1,010	0	1,010
 com.qa.main.service	 100.0 %	576	0	576
 src/main/java	 67.6 %	700	335	1,035
 com.qa.main.persistence.domain	 53.8 %	215	185	400
 com.qa.main.dto	 51.8 %	144	134	278
 com.qa.main.utils	 87.5 %	49	7	56
 com.qa.main	 37.5 %	3	5	8
 com.qa.main.exceptions	 0.0 %	0	4	4
 com.qa.main.config	 100.0 %	7	0	7
 com.qa.main.controller	 100.0 %	112	0	112
 com.qa.main.service	 100.0 %	170	0	170






# STS COVERAGE as JUNIT 5

Runs: 44/44	Errors: 0	Failures: 0
ItemControllerTest [Runner: JUnit 5] (1.836 s)		
testReadById() (0.137 s)		
testReadAll() (0.013 s)		
testCreate() (0.008 s)		
testDelete() (0.010 s)		
testUpdate() (1.668 s)		
ItemIntegrationTest [Runner: JUnit 5] (2.043 s)		
testReadById() (0.285 s)		
testFindItemsInList() (0.105 s)		
testReadAll() (0.189 s)		
testDeletePass() (0.080 s)		
testCreate() (0.110 s)		
testUpdate() (1.274 s)		
ToDoListControllerTest [Runner: JUnit 5] (0.089 s)		
testReadById() (0.011 s)		
testCreatePass() (0.013 s)		
testReadAll() (0.009 s)		
testDeleteFail() (0.013 s)		
testDeletePass() (0.011 s)		
testUpdate() (0.031 s)		
ToDoListIntegrationTest [Runner: JUnit 5] (1.506 s)		
testReadById() (0.043 s)		
testReadAll() (0.037 s)		
testDeletePass() (0.041 s)		
testCreate() (0.019 s)		
testUpdate() (1.366 s)		
TdlApplicationTests [Runner: JUnit 5] (0.020 s)		
contextLoads() (0.020 s)		
POJOTest [Runner: JUnit 5] (1.098 s)		
testOnCascadeDelete() (1.098 s)		

ToDoListServiceTest [Runner: JUnit 5] (0.995 s)
testReadById() (0.016 s)
testReadAll() (0.033 s)
testDeleteFail() (0.007 s)
testCreate() (0.004 s)
testUpdate() (0.935 s)
ItemServiceTest [Runner: JUnit 5] (2.008 s)
testReadById() (0.007 s)
testFindItemsInList() (0.006 s)
testReadAll() (0.005 s)
testDeleteFail() (0.009 s)
testDeletePass() (0.005 s)
testCreate() (0.005 s)
testUpdate() (1.971 s)
WebAppTest [Runner: JUnit 5] (26.474 s)
testUpdateItem() (5.256 s)
testUpdateList() (3.308 s)
testCreateItem() (2.839 s)
testCreateList() (2.575 s)
testReadListById() (3.020 s)
testDeleteItem() (3.027 s)
testDeleteList() (3.021 s)
testReadAllLists() (3.428 s)

# ACCEPTANCE TESTS with SELENIUM






TESTS



Update list by Id Test	Pass
Create a item test	Pass
Create a list test	Pass
Read list by Id Test	Pass
Delete list Test	Pass
Read all lists test	Pass

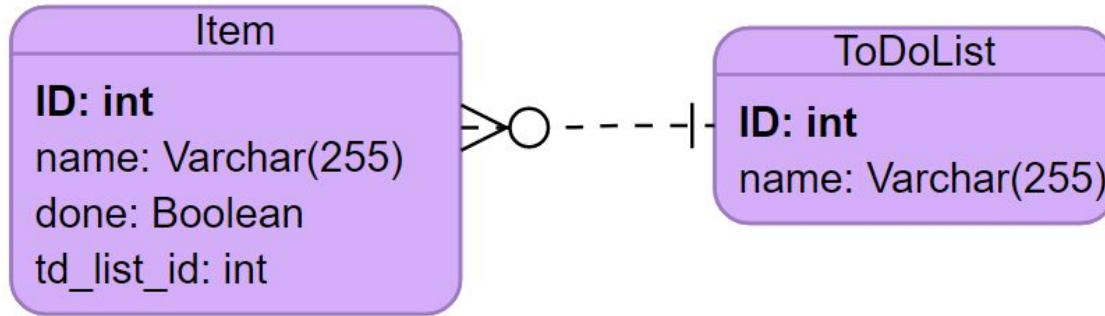
Update list by Id Test

2021-02-21 20:17:13 2021-02-21 20:17:16 0h 0m 2s+534ms

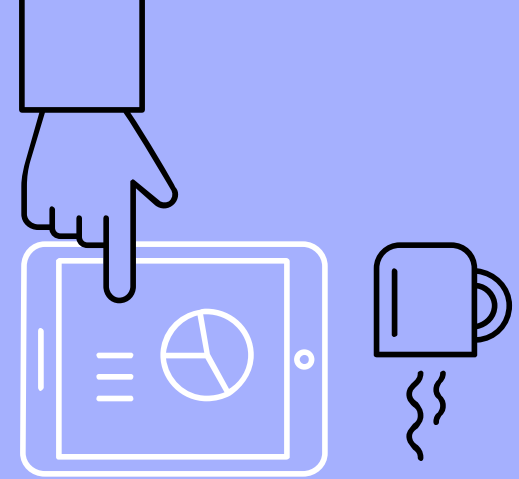
STATUS	TIMESTAMP	DETAILS
	20:17:13	Given - we can access the To Do List webpage
	20:17:14	When we read the first list
	20:17:15	And when we click the edit/pencil button
	20:17:16	Then - I should see this list with new name
	20:17:16	List updated successfully.

RETROSPECTIVE:

# Entity Relationship Diagram (final)



[start](#)

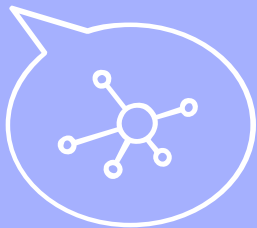




[illegible]

17

**F  
I  
N  
A  
L**



What went well?



**DELIVERED MVP**



**ADDED EXTRA FEATURES**

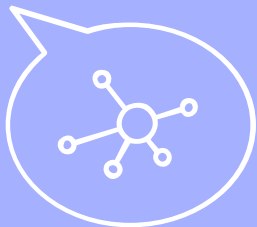


**LEARNED NEW TECHNOLOGIES**

**R  
E  
V  
I  
E  
W**



**F  
I  
N  
A  
L**



What could be done to improve?



**MORE FREQUENT GIT COMMITS**



**BETTER TIME MANAGEMENT**



**MVP FIRST**

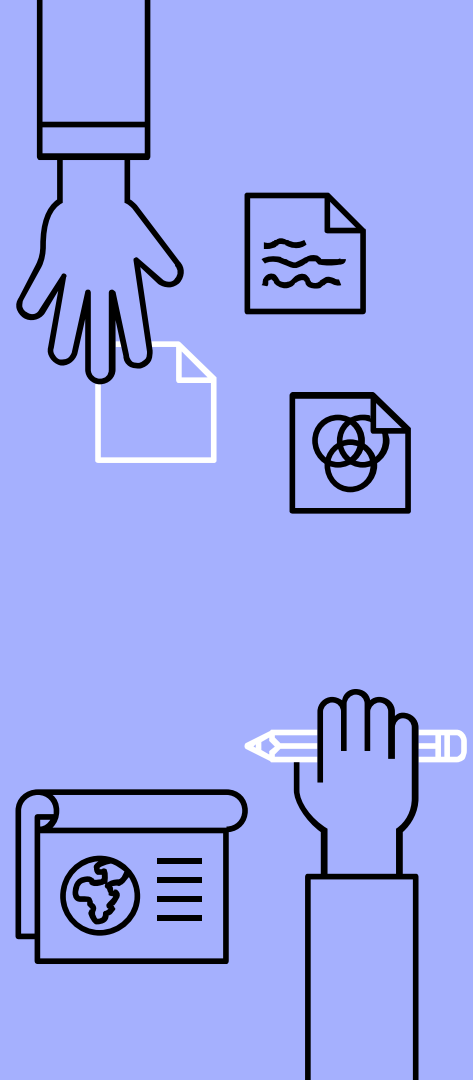
**R  
E  
V  
I  
E  
W**



## COULD HAVE

A web app that has extra features such as searching for a list by name or description.

Mark task as done using checkboxes.

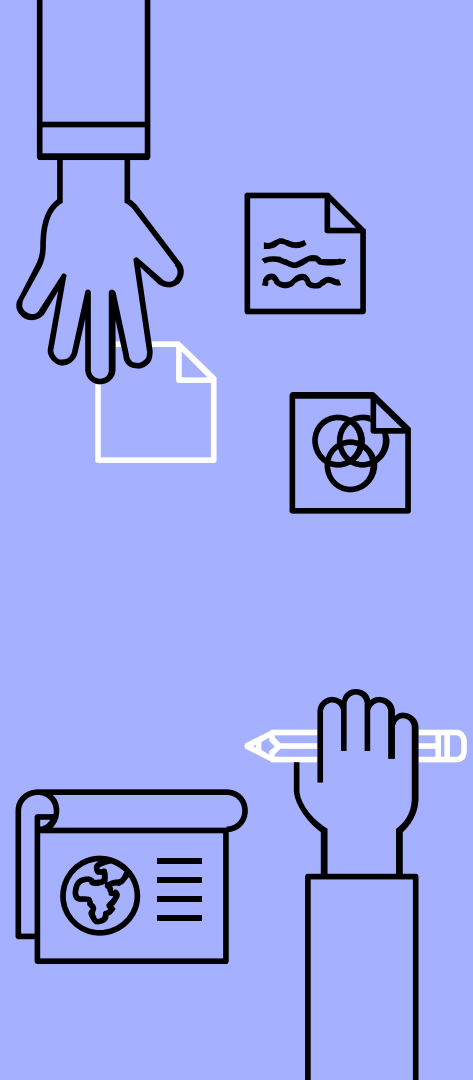


## WOULD HAVE

A web app that asks for user identification in order to access the lists they previously created.

Sessions.

Email notifications.



# THANKS!

Any questions?

