

# Hobby Web Application

# Introduction

Approaching the project:

- Read Specification and list all things to be completed.
- Order the list in terms of what needs to be completed first.
- Link tasks that are dependent on other tasks being completed.
- With each task give a brief detail of what needs to be done.
  - Item Controller example:
    - Create: Take inputs and store into a database.
    - Update: Specify which item and what needs to be amended then update the changes to the existing item.
    - Delete: Select which item to remove and delete from database.

# Concept

Before applying a topic to the web application listed out what were the necessary components.

- At least one table in database
- Interface where users can enter information (Front-end)
- CRUD functionality (Buttons for each)

Next was to think of a concept that made use of these “must haves”

Design basic structure of all the components.

- What the ERD would look like
- What the Front-end would include
  - Single Page/Multiple Pages
  - Styling (Headers must be in bold)

Spending Tracker	
PK	<u>ID</u>
	Amount
	Type
	Description

# Sprint






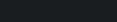






## Must achieve

- CRUD functionality
- Basic Front-end user interface
- Back-end

## Hoping to achieve






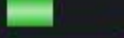



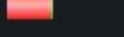
- Aesthetic design more pleasing on the eye
- A more sophisticated design of the html
- Some auto inputted data like date

# Testing Back-End

Element		Coverage	Covered Instructions	Missed Instructions	Total Instructions
▼ HWA		59.0 %	627	436	1,063
▼ src/test/java		55.2 %	416	338	754
> com.qa.hwa.selenium.test		0.0 %	0	243	243
> com.qa.hwa.selenium.page		0.0 %	0	95	95
▼ com.qa.hwa		100.0 %	4	0	4
> HwaApplicationTests.java		100.0 %	4	0	4
▼ com.qa.hwa.controller		100.0 %	174	0	174
> SpendControllerIntegrationTest.java		100.0 %	174	0	174
▼ com.qa.hwa.service		100.0 %	238	0	238
> SpendServiceTest.java		100.0 %	238	0	238
> src/main/java		68.3 %	211	98	309

# Testing Front End

Combined Coverage 92.5%

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
▼ HWA	 33.5 %	356	707	1,063
▼ src/test/java	 44.8 %	338	416	754
> com.qa.hwa.service	 0.0 %	0	238	238
> com.qa.hwa.controller	 0.0 %	0	174	174
> com.qa.hwa	 0.0 %	0	4	4
▼ com.qa.hwa.selenium.page	 100.0 %	95	0	95
> Home.java	 100.0 %	95	0	95
▼ com.qa.hwa.selenium.test	 100.0 %	243	0	243
> HomeTest.java	 100.0 %	243	0	243
> src/main/java	 5.8 %	18	291	309

# Technologies Used

Continued with building on the foundations laid from previous IMS project gaining further knowledge through more experience implementing the techniques

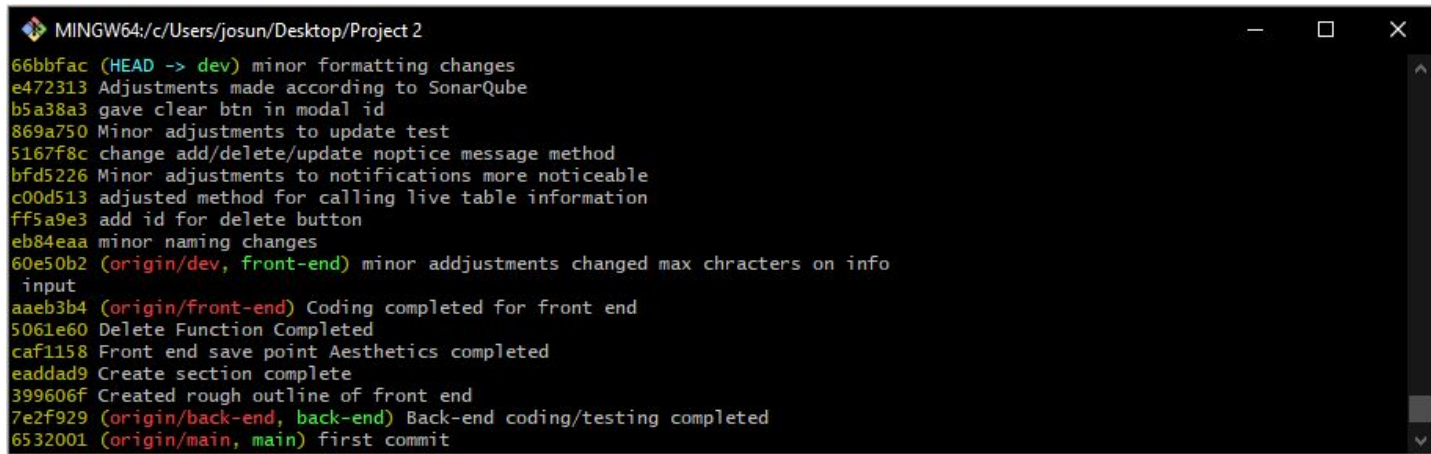
As well as adding the below concepts

- HTML,CSS,JAVASCRIPT
- SonarQube
- Selenium

# Version Control

Looked to complete the back-end first as that was probably the quickest to complete.

Paid more attention to the uses of branches and merging for better version control

A screenshot of a terminal window with a black background and white text. The window title is 'MINGW64:/c/Users/josun/Desktop/Project 2'. It displays a list of git commit messages in reverse chronological order. The messages are color-coded: green for HEAD, red for origin/dev, and blue for origin/main. The commits include descriptions of formatting changes, adjustments for SonarQube, button updates, test updates, message method changes, notification adjustments, live table information calls, delete button IDs, naming changes, character limit adjustments, front-end coding completion, delete function completion, aesthetics completion, section creation, front-end outline creation, back-end coding/testing completion, and the first commit.

```
MINGW64:/c/Users/josun/Desktop/Project 2
66bbfac (HEAD -> dev) minor formatting changes
e472313 Adjustments made according to SonarQube
b5a38a3 gave clear btn in modal id
869a750 Minor adjustments to update test
5167f8c change add/delete/update noptice message method
bfd5226 Minor adjustments to notifications more noticeable
c00d513 adjusted method for calling live table information
ff5a9e3 add id for delete button
eb84eaa minor naming changes
60e50b2 (origin/dev, front-end) minor adjustments changed max chracters on info
input
aaeb3b4 (origin/Front-end) Coding completed for front end
5061e60 Delete Function Completed
caf1158 Front end save point Aesthetics completed
eaddad9 Create section complete
399606f Created rough outline of front end
7e2f929 (origin/back-end, back-end) Back-end coding/testing completed
6532001 (origin/main, main) first commit
```



# Review

All must achieve objectives in sprint are completed.

Some minor styling has been used.

Testing has meet the 80% criteria.

More could be done to make it more colourful.

Pay more attention to details (Testing)

# Retrospective

- More relaxed style of coding
  - Potentially lead to more minor bugs cropping into the programming
  - But problem solving in my opinion was easier due to there being more feedback when bugs did appear.
- Scheduling
  - Happy with my planning and estimation of roughly how long each part would take
  - Due to good scheduling full functionality of project was reached by the beginning of this week
  - Had all of Thursday to review everything and make any adjustments necessary

# Conclusion

From feedback/experience of previous project:

- Still not 100% happy with version control.
- Progression in testing was smoother and better than previous project
- Managed to keep documentation up to date throughout the project.

Future Steps for project:

- Add addition features like auto date input or attachment feature that can give pictures.

# First SonarQube Run

QUALITY GATE STATUS ⓘ

**Passed**

All conditions passed.

MEASURES

New Code

Overall Code

1

🐛 Bugs

Reliability

C

2

🔒 Vulnerabilities

Security

D

1

🛡️ Security Hotspots ⓘ

0.0% Reviewed

Security Review

E

32min

Debt

10

🧪 Code Smells

Maintainability

A

0.0%

0.0%

Coverage on 79 Lines to cover

-

Unit Tests

0.0%

0.0%

Duplications on 265 Lines

0

Duplicated Blocks

# Second SonarQube Run

## QUALITY GATE STATUS ⓘ

**Passed**

All conditions passed.

## MEASURES

### New Code

Since May 13, 2021  
Started 2 hours ago

### Overall Code

0

🐛 New Bugs

Reliability

A

0

🔒 New Vulnerabilities

Security

A

0

🛡️ New Security Hotspots ⓘ

— Reviewed

Security Review

A

0

Added Debt

3

🐻 New Code Smells

Maintainability

A

 0.0%

Coverage on 7 New Lines to cover

 0.0%

Duplications on 289 New Lines















▼ Backlog (7 issues)

0

0

0

Create sprint

-  HOB-3 Create an add function SPENDING TRACKER 
-  HOB-4 Create a read function SPENDING TRACKER 
-  HOB-5 Create a delete function SPENDING TRACKER 
-  HOB-6 Create an update function SPENDING TRACKER 
-  HOB-7 Complete a risk assessment SPENDING TRACKER 
-  HOB-8 Complete ERD SPENDING TRACKER 
-  HOB-9 Create a Front end to interact with SPENDING TRACKER 

+ Create issue

## Create an add function

📎 Attach

🔗 Add a child issue

🔗 Link issue



### Description

As the user

I want to be able to add new spending info into the database

So that the database is always up to date

### Add new entry: acceptance criteria

**Given** the user is on add section

**When** new all details have been entered

**And** pressed enter

**Then** the new entity will be added to the database

Activity



To Do ▾

Assignee



Unassigned

Labels

None

Sprint

None

Story point estimate

2

Reporter



Kevin Tse

Created 26 minutes ago

Updated 39 seconds ago



Configure

	Description	Evaluation	Likelihood	Impact	Response	Control Measure
1	Modifying completed coding	Can possibly break things or create new issues that become difficult to resolve	High	Medium	If done you can always use the undo method to revert changes	Either have a separate copy on local machine that can be used as a safety net or create new branch solely for testing ideas and pull from that
2	Incorrect Deletion	Lost of work	Medium	Medium	Find method to undo deletion	Rather than delete straight use comments to slowly phase the use of this folder out
3	Requirements are not met as specified	App would not be perfect and may not perform as intended	Low	High	Try to adapt the features so they cover the specified tasks	Always plan new features while referencing the specifications
4	Unresolved issues	When problems arise in coding i.e. not function as intended and then being unable to resolve by yourself	High	Medium	Always reference your documents like the MoSCoW your acceptance criteria and your tests so you have a clear end goal	By using the Test Driven Development methods you can ensure reductions in deviations during the project
5	Uploading without fully testing	Bugs may cause the application to be faulty and possibly unusable	Low	High	Some errors can leak through	When coding each function have the tests coded in parallel