

HWA PROJECT

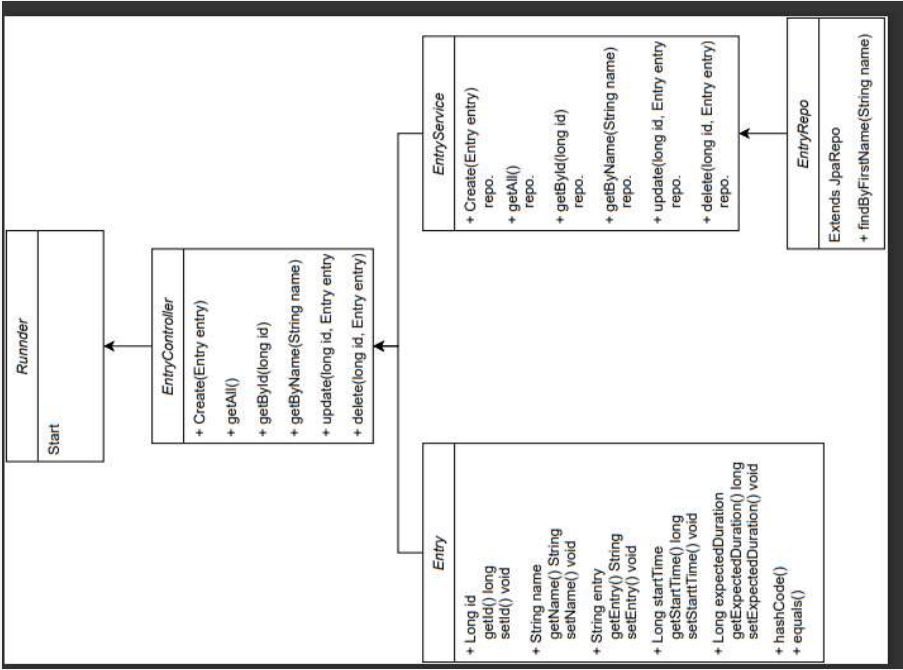
- Jesse Morgan

INTRODUCTION

- Who am I?
- What the specs are?
- How I approached the specs?

This was My starting ERD and UML

ToDo	
PK	toDo_id int NOT NULL AUTO_INCREMENT
	name char(50) NOT NULL
	entry char(255) NOT NULL
	startTime int NOT NULL
	expectedDuration int NOT NULL



Beginning Jira

Epic	MAY							JUN							JUN							JUN														
	T	F	S	S	S	M	T	W	T	F	S	S	S	M	T	W	T	F	S	S	S	M	T	W	T	F	S	S	S	M	T	W	T	F	S	S
▼																																				
▼	HP-29 Planning, Proof and Presenting																																			
	✓ HP-30 Pre-Production Docs To Do																																			
	✓ HP-31 ReadMe To Do																																			
	✓ HP-32 Diagrams To Do																																			
	✓ HP-33 Presentation file To Do																																			
▼	HP-28 Back-End																																			
	✓ HP-1 Java To Do																																			
	✓ HP-2 Testing, Junit/Mockito To Do																																			
	✓ HP-6 MySQL To Do																																			
	✓ HP-20 Bug-Fixes and Building To Do																																			
▼	HP-27 Front-End																																			
	✓ HP-3 Html To Do																																			
	✓ HP-4 Css To Do																																			
	✓ HP-5 JavaScript To Do																																			
▼	HP-34 User Stories																																			
	✓ HP-35 As a User I would like to be able to... To Do																																			
	✓ HP-36 As a User I would like to create e... To Do																																			
	✓ HP-37 As a User I would like to Update... To Do																																			
	✓ HP-38 As a User I would like to Delete e... To Do																																			
✓ HP-39 As a User I would like to view ent... To Do																																				
✓ HP-40 As a User I would like to have a b... To Do																																				

GIT!

Default branch			↻
main	Updated 13 minutes ago by JMorgan-git	Default	🔗
Your branches			
dev	Updated 5 days ago by JMorgan-git	3 0	#1 Merged 🔗
test	Updated 7 days ago by JMorgan-git	5 0	New pull request 🔗
feature1	Updated 7 days ago by JMorgan-git	6 0	New pull request 🔗
Active branches			
dev	Updated 5 days ago by JMorgan-git	3 0	#1 Merged 🔗
test	Updated 7 days ago by JMorgan-git	5 0	New pull request 🔗
feature1	Updated 7 days ago by JMorgan-git	6 0	New pull request 🔗

Commits on Jun 6, 2022

Add files via upload JMorgan-git committed 13 minutes ago	Verified		0cb6257	
Update README.md JMorgan-git committed 19 minutes ago	Verified		db94aa4	
Update README.md JMorgan-git committed 24 minutes ago	Verified		7916eb9	

Commits on Jun 1, 2022





last add JMorgan-git committed 5 days ago		daad1a7	
add html, css and js files JMorgan-git committed 5 days ago		a339aea	

Commits on May 30, 2022










tests added 88% achieved JMorgan-git committed 7 days ago		5d21912	
bones of base java work JMorgan-git committed 7 days ago		944713d	
first commit JMorgan-git committed 7 days ago		9fc08cc	

 dev

Commits on Jun 1, 2022

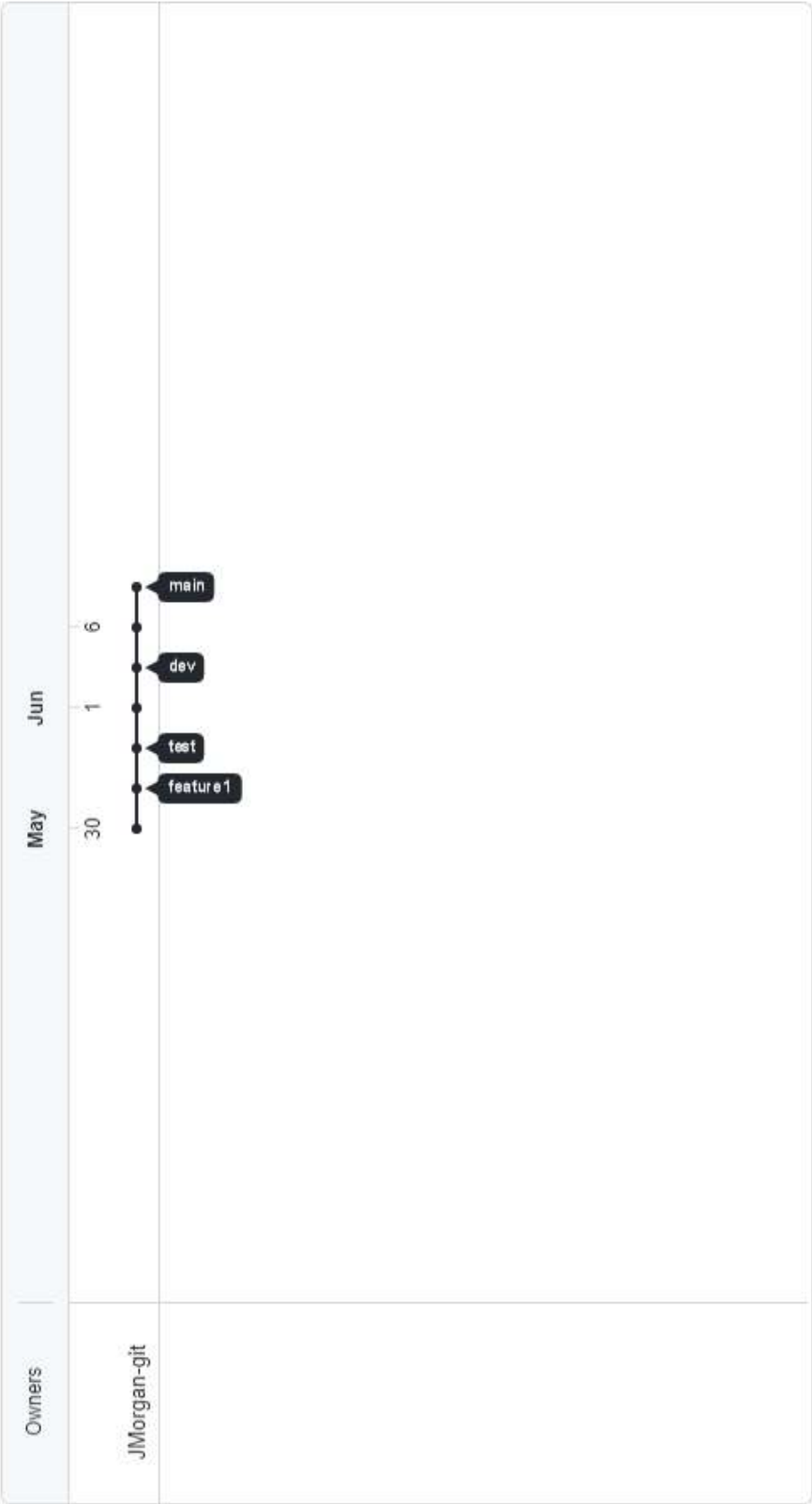
last add		JMorgan-git committed 5 days ago		daad1a7	
add html, css and js files		JMorgan-git committed 5 days ago		a339aea	

Commits on May 30, 2022



tests added 88% achieved		JMorgan-git committed 7 days ago		5d21912	
bones of base java work		JMorgan-git committed 7 days ago		944713d	
first commit		JMorgan-git committed 7 days ago		9fc08cc	

Newer

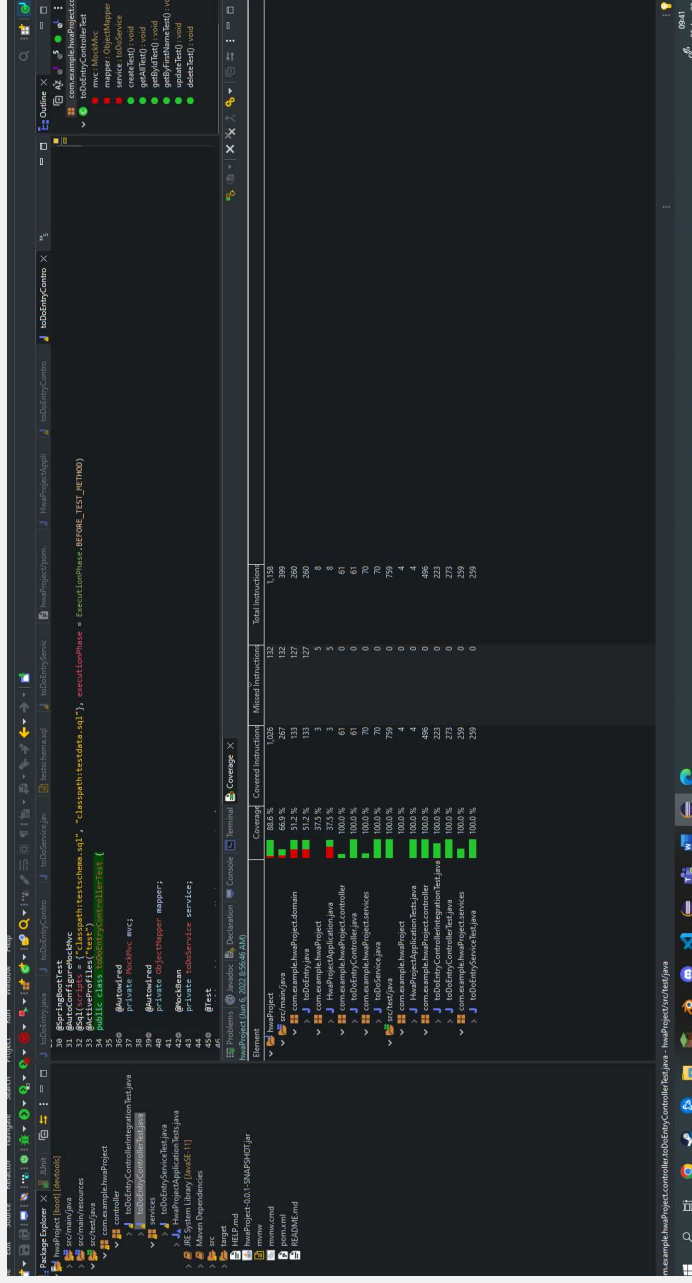
Older



Dependency graph

Dependencies	Dependents	Dependabot
These dependencies are defined in HWAPProject's manifest files, such as pom.xml .		
Dependencies defined in pom.xml 9		
 h2database / h2database	com.h2database:h2	
 mysql / mysql-connector-j	mysql:mysql-connector-java	
 projectlombok / lombok	org.projectlombok:lombok	
 spring-projects / spring-boot	org.springframework.boot:spring-boot-devtools	
 spring-projects / spring-boot	org.springframework.boot:spring-boot-maven-plugin	
 spring-projects / spring-boot	org.springframework.boot:spring-boot-starter-data-jpa	
 spring-projects / spring-boot	org.springframework.boot:spring-boot-starter-test	
 spring-projects / spring-boot	org.springframework.boot:spring-boot-starter-validation	
 spring-projects / spring-boot	org.springframework.boot:spring-boot-starter-web	

TESTING



SERVICE UNIT

```
com.example.hwaprojectse
toDoEntryServiceTest
service : toDoService
repo : toDoEntryRepo
createTest() : void
getAllTest() : void
getByIdTest() : void
updateTest() : void
deleteTest() : void

toDoEntryControllerTest.java
HwaprojectApplicationTe
toDoEntryServiceTest.java
application-prod-propert
testdata.sql
testschema.sql
package com.example.hwaproject.services;
import java.util.ArrayList;
17
18 @SpringBootTest
19 public class toDoEntryServiceTest {
20
21     @Autowired
22     private toDoService service;
23
24     @MockBean
25     private toDoEntryRepo repo;
26
27     // Create Test
28     @Test
29     public void createTest() {
30         toDoEntry input = new toDoEntry("John", "Walk dog", 1130, 30);
31         toDoEntry output = new toDoEntry(1L, "John", "Walk dog", 1130, 30);
32
33         Mockito.when(repo.saveAndFlush(output)).thenReturn(output);
34         assertEquals(output, service.create(output));
35
36         Mockito.verify(repo, Mockito.times(1)).saveAndFlush(output);
37     }
38
39     // Read all Test
40
41     @Test
42     public void getAllTest() {
43         List<toDoEntry> output = new ArrayList<>();
44         output.add(new toDoEntry(1L, "John", "Walk dog", 1130, 30));
45
46         Mockito.when(repo.findAll()).thenReturn(output);
47         assertEquals(output, service.getAll());
48
49         Mockito.verify(repo, Mockito.times(1)).findAll();
50     }
51
52     // Read by id Test
53
54     @Test
55     public void getByIdTest() {
56
57         toDoEntry output = new toDoEntry(1L, "John", "Walk dog", 1130, 30);
58         Mockito.when(repo.findById(1L)).thenReturn(Optional.of(output));
59         assertEquals(output, service.getById(1L));
60
61         Mockito.verify(repo, Mockito.times(1)).findById(1L);
62
63 }
```

CONTROLLER UNIT

```
e:\hwaProject\controller\toDoEntryControllerTest.java - Eclipse IDE
Project Run Window Help
application-prod.properties testdata.sql testschema.sql testschema.sql
toDoEntryServiceTest.java toDoEntryControllerIntegrationTest.java toDoEntryControllerTest.java
30 @SpringBootTest
31 @AutoConfigureMockMvc
32 @Sql(scripts = {"classpath:testschema.sql", "classpath:testdata.sql"}, executionPhase = ExecutionPhase.BEFORE_TEST_METHOD)
33 @ActiveProfiles("test")
34 public class toDoEntryControllerTest {
35
36     @Autowired
37     private MockMvc mvc;
38
39     @Autowired
40     private ObjectMapper mapper;
41
42     @MockBean
43     private toDoService service;
44
45     @Test
46     public void createTest() throws Exception {
47         toDoEntry input = new toDoEntry("John", "Walk dog", 1130, 30);
48         String inputAsJson = mapper.writeValueAsString(input);
49         toDoEntry output = new toDoEntry(2L, "John", "Walk dog", 1130, 30);
50         String outputAsJson = mapper.writeValueAsString(output);
51
52         Mockito.when(this.service.create(input)).thenReturn(output);
53
54         mvc.perform(post("/entries/create")
55             .contentType(MediaType.APPLICATION_JSON)
56             .content(inputAsJson))
57             .andExpect(status().isCreated())
58             .andExpect(content().json(outputAsJson));
59     }
60
61     @Test
62     public void getAllTest() throws Exception {
63         toDoEntry toDoEntry = new toDoEntry(1L, "John", "Walk dog", 1030, 30);
64         List<toDoEntry> output = new ArrayList<>();
65         output.add(toDoEntry);
66         String outputAsJson = mapper.writeValueAsString(output);
67
68         Mockito.when(this.service.getAll()).thenReturn(output);
69
70         mvc.perform(get("/entries/getall")
71             .contentType(MediaType.APPLICATION_JSON)
72             .andExpect(status().isOk())
73             .andExpect(content().json(outputAsJson)));
74     }
75
76     @Test
77     public void getEntryTest() throws Exception {
78         toDoEntry toDoEntry = new toDoEntry(1L, "John", "Walk dog", 1030, 30);
79         String outputAsJson = mapper.writeValueAsString(toDoEntry);
80         Mockito.when(this.service.getEntry(1L)).thenReturn(toDoEntry);
81         mvc.perform(get("/entries/{id}", 1L)
82             .contentType(MediaType.APPLICATION_JSON)
83             .andExpect(status().isOk())
84             .andExpect(content().json(outputAsJson)));
85     }
86
87     @Test
88     public void deleteTest() throws Exception {
89         toDoEntry toDoEntry = new toDoEntry(1L, "John", "Walk dog", 1030, 30);
90         Mockito.when(this.service.deleteEntry(1L)).thenReturn(true);
91         mvc.perform(delete("/entries/{id}", 1L)
92             .andExpect(status().isNoContent()));
93     }
94
95     @Test
96     public void updateTest() throws Exception {
97         toDoEntry input = new toDoEntry(1L, "John", "Walk dog", 1030, 30);
98         toDoEntry output = new toDoEntry(1L, "John", "Walk dog", 1130, 30);
99         String inputAsJson = mapper.writeValueAsString(input);
100        String outputAsJson = mapper.writeValueAsString(output);
101        Mockito.when(this.service.updateEntry(1L, input)).thenReturn(output);
102        mvc.perform(patch("/entries/{id}", 1L)
103            .contentType(MediaType.APPLICATION_JSON)
104            .content(inputAsJson))
105            .andExpect(status().isOk())
106            .andExpect(content().json(outputAsJson));
107    }
108}
```

Outline

- com.example.hwaProject
- toDoEntryControllerTest
- mvc: MockMvc
- mapper: ObjectMapper
- service: toDoService
- createTest(): void
- getAllTest(): void
- getEntryTest(): void
- updateTest(): void
- deleteTest(): void

CONTROLLER INTEGRATION

```

1 package com.example.hwaProject.controller;
2
3 import static org.springframework.test.web.servlet.request.MockMvcRequestBuilders.delete;
4
5 import org.springframework.boot.test.autoconfigure.web.servlet.AutoConfigureMockMvc;
6 import org.springframework.boot.test.context.SpringBootTest;
7 import org.springframework.data.jpa.repository.JpaRepository;
8 import org.springframework.data.jpa.repository.Query;
9 import org.springframework.data.repository.query.Param;
10 import org.springframework.http.MediaType;
11 import org.springframework.security.test.web.servlet.request.SecurityMockMvcRequestBuilders;
12 import org.springframework.test.web.servlet.MockMvc;
13 import org.springframework.test.web.servlet.MvcResult;
14 import org.springframework.test.web.servlet.request.MockMvcRequestBuilders;
15 import org.springframework.test.web.servlet.result.MockMvcResultMatchers;
16
17 import java.util.List;
18
19 @SpringBootTest
20 @AutoConfigureMockMvc
21 @SpringBootTest(classes = {ToDoEntryController.class, ToDoEntryRepository.class, ToDoEntryMapper.class, ToDoEntryService.class})
22 @ActiveProfiles("test")
23 public class ToDoEntryControllerIntegrationTest {
24
25     @Autowired
26     private MockMvc mvc;
27
28     @Autowired
29     private ObjectMapper mapper;
30
31     @Test
32     public void createTest() throws Exception {
33         ToDoEntry input = new ToDoEntry("John", "Walk cat", 1130, 30);
34         String inputJson = mapper.writeValueAsString(input);
35
36         mockMvc.perform(post("/entries/create")
37             .contentType(MediaType.APPLICATION_JSON)
38             .content(inputJson)
39             .andExpect(status().isCreated())
40             .andExpect(content().json(outputJson)))
41             .andExpect(status().isCreated());
42
43         String outputJson = mapper.writeValueAsString(output);
44
45         mockMvc.perform(post("/entries/create")
46             .contentType(MediaType.APPLICATION_JSON)
47             .content(inputJson)
48             .andExpect(status().isCreated())
49             .andExpect(content().json(outputJson)))
50             .andExpect(status().isCreated());
51
52         mockMvc.perform(post("/entries/create")
53             .contentType(MediaType.APPLICATION_JSON)
54             .content(inputJson)
55             .andExpect(status().isCreated())
56             .andExpect(content().json(outputJson)))
57             .andExpect(status().isCreated());
58
59         mockMvc.perform(post("/entries/create")
60             .contentType(MediaType.APPLICATION_JSON)
61             .content(inputJson)
62             .andExpect(status().isCreated())
63             .andExpect(content().json(outputJson)))
64             .andExpect(status().isCreated());
65
66         mockMvc.perform(post("/entries/create")
67             .contentType(MediaType.APPLICATION_JSON)
68             .content(inputJson)
69             .andExpect(status().isCreated())
70             .andExpect(content().json(outputJson)))
71             .andExpect(status().isCreated());
72
73         mockMvc.perform(post("/entries/create")
74             .contentType(MediaType.APPLICATION_JSON)
75             .content(inputJson)
76             .andExpect(status().isCreated())
77             .andExpect(content().json(outputJson)))
78             .andExpect(status().isCreated());
79
80         mockMvc.perform(post("/entries/create")
81             .contentType(MediaType.APPLICATION_JSON)
82             .content(inputJson)
83             .andExpect(status().isCreated())
84             .andExpect(content().json(outputJson)))
85             .andExpect(status().isCreated());
86
87         mockMvc.perform(post("/entries/create")
88             .contentType(MediaType.APPLICATION_JSON)
89             .content(inputJson)
90             .andExpect(status().isCreated())
91             .andExpect(content().json(outputJson)))
92             .andExpect(status().isCreated());
93
94         mockMvc.perform(post("/entries/create")
95             .contentType(MediaType.APPLICATION_JSON)
96             .content(inputJson)
97             .andExpect(status().isCreated())
98             .andExpect(content().json(outputJson)))
99             .andExpect(status().isCreated());
100
101         mockMvc.perform(post("/entries/create")
102             .contentType(MediaType.APPLICATION_JSON)
103             .content(inputJson)
104             .andExpect(status().isCreated())
105             .andExpect(content().json(outputJson)))
106             .andExpect(status().isCreated());
107
108         mockMvc.perform(post("/entries/create")
109             .contentType(MediaType.APPLICATION_JSON)
110             .content(inputJson)
111             .andExpect(status().isCreated())
112             .andExpect(content().json(outputJson)))
113             .andExpect(status().isCreated());
114
115         mockMvc.perform(post("/entries/create")
116             .contentType(MediaType.APPLICATION_JSON)
117             .content(inputJson)
118             .andExpect(status().isCreated())
119             .andExpect(content().json(outputJson)))
120             .andExpect(status().isCreated());
121
122         mockMvc.perform(post("/entries/create")
123             .contentType(MediaType.APPLICATION_JSON)
124             .content(inputJson)
125             .andExpect(status().isCreated())
126             .andExpect(content().json(outputJson)))
127             .andExpect(status().isCreated());
128
129         mockMvc.perform(post("/entries/create")
130             .contentType(MediaType.APPLICATION_JSON)
131             .content(inputJson)
132             .andExpect(status().isCreated())
133             .andExpect(content().json(outputJson)))
134             .andExpect(status().isCreated());
135
136         mockMvc.perform(post("/entries/create")
137             .contentType(MediaType.APPLICATION_JSON)
138             .content(inputJson)
139             .andExpect(status().isCreated())
140             .andExpect(content().json(outputJson)))
141             .andExpect(status().isCreated());
142
143         mockMvc.perform(post("/entries/create")
144             .contentType(MediaType.APPLICATION_JSON)
145             .content(inputJson)
146             .andExpect(status().isCreated())
147             .andExpect(content().json(outputJson)))
148             .andExpect(status().isCreated());
149
150         mockMvc.perform(post("/entries/create")
151             .contentType(MediaType.APPLICATION_JSON)
152             .content(inputJson)
153             .andExpect(status().isCreated())
154             .andExpect(content().json(outputJson)))
155             .andExpect(status().isCreated());
156
157         mockMvc.perform(post("/entries/create")
158             .contentType(MediaType.APPLICATION_JSON)
159             .content(inputJson)
160             .andExpect(status().isCreated())
161             .andExpect(content().json(outputJson)))
162             .andExpect(status().isCreated());
163
164         mockMvc.perform(post("/entries/create")
165             .contentType(MediaType.APPLICATION_JSON)
166             .content(inputJson)
167             .andExpect(status().isCreated())
168             .andExpect(content().json(outputJson)))
169             .andExpect(status().isCreated());
170
171         mockMvc.perform(post("/entries/create")
172             .contentType(MediaType.APPLICATION_JSON)
173             .content(inputJson)
174             .andExpect(status().isCreated())
175             .andExpect(content().json(outputJson)))
176             .andExpect(status().isCreated());
177
178         mockMvc.perform(post("/entries/create")
179             .contentType(MediaType.APPLICATION_JSON)
180             .content(inputJson)
181             .andExpect(status().isCreated())
182             .andExpect(content().json(outputJson)))
183             .andExpect(status().isCreated());
184
185         mockMvc.perform(post("/entries/create")
186             .contentType(MediaType.APPLICATION_JSON)
187             .content(inputJson)
188             .andExpect(status().isCreated())
189             .andExpect(content().json(outputJson)))
190             .andExpect(status().isCreated());
191
192         mockMvc.perform(post("/entries/create")
193             .contentType(MediaType.APPLICATION_JSON)
194             .content(inputJson)
195             .andExpect(status().isCreated())
196             .andExpect(content().json(outputJson)))
197             .andExpect(status().isCreated());
198
199         mockMvc.perform(post("/entries/create")
200             .contentType(MediaType.APPLICATION_JSON)
201             .content(inputJson)
202             .andExpect(status().isCreated())
203             .andExpect(content().json(outputJson)))
204             .andExpect(status().isCreated());
205
206         mockMvc.perform(post("/entries/create")
207             .contentType(MediaType.APPLICATION_JSON)
208             .content(inputJson)
209             .andExpect(status().isCreated())
210             .andExpect(content().json(outputJson)))
211             .andExpect(status().isCreated());
212
213         mockMvc.perform(post("/entries/create")
214             .contentType(MediaType.APPLICATION_JSON)
215             .content(inputJson)
216             .andExpect(status().isCreated())
217             .andExpect(content().json(outputJson)))
218             .andExpect(status().isCreated());
219
220         mockMvc.perform(post("/entries/create")
221             .contentType(MediaType.APPLICATION_JSON)
222             .content(inputJson)
223             .andExpect(status().isCreated())
224             .andExpect(content().json(outputJson)))
225             .andExpect(status().isCreated());
226
227         mockMvc.perform(post("/entries/create")
228             .contentType(MediaType.APPLICATION_JSON)
229             .content(inputJson)
230             .andExpect(status().isCreated())
231             .andExpect(content().json(outputJson)))
232             .andExpect(status().isCreated());
233
234         mockMvc.perform(post("/entries/create")
235             .contentType(MediaType.APPLICATION_JSON)
236             .content(inputJson)
237             .andExpect(status().isCreated())
238             .andExpect(content().json(outputJson)))
239             .andExpect(status().isCreated());
240
241         mockMvc.perform(post("/entries/create")
242             .contentType(MediaType.APPLICATION_JSON)
243             .content(inputJson)
244             .andExpect(status().isCreated())
245             .andExpect(content().json(outputJson)))
246             .andExpect(status().isCreated());
247
248         mockMvc.perform(post("/entries/create")
249             .contentType(MediaType.APPLICATION_JSON)
250             .content(inputJson)
251             .andExpect(status().isCreated())
252             .andExpect(content().json(outputJson)))
253             .andExpect(status().isCreated());
254
255         mockMvc.perform(post("/entries/create")
256             .contentType(MediaType.APPLICATION_JSON)
257             .content(inputJson)
258             .andExpect(status().isCreated())
259             .andExpect(content().json(outputJson)))
260             .andExpect(status().isCreated());
261
262         mockMvc.perform(post("/entries/create")
263             .contentType(MediaType.APPLICATION_JSON)
264             .content(inputJson)
265             .andExpect(status().isCreated())
266             .andExpect(content().json(outputJson)))
267             .andExpect(status().isCreated());
268
269         mockMvc.perform(post("/entries/create")
270             .contentType(MediaType.APPLICATION_JSON)
271             .content(inputJson)
272             .andExpect(status().isCreated())
273             .andExpect(content().json(outputJson)))
274             .andExpect(status().isCreated());
275
276         mockMvc.perform(post("/entries/create")
277             .contentType(MediaType.APPLICATION_JSON)
278             .content(inputJson)
279             .andExpect(status().isCreated())
280             .andExpect(content().json(outputJson)))
281             .andExpect(status().isCreated());
282
283         mockMvc.perform(post("/entries/create")
284             .contentType(MediaType.APPLICATION_JSON)
285             .content(inputJson)
286             .andExpect(status().isCreated())
287             .andExpect(content().json(outputJson)))
288             .andExpect(status().isCreated());
289
290         mockMvc.perform(post("/entries/create")
291             .contentType(MediaType.APPLICATION_JSON)
292             .content(inputJson)
293             .andExpect(status().isCreated())
294             .andExpect(content().json(outputJson)))
295             .andExpect(status().isCreated());
296
297         mockMvc.perform(post("/entries/create")
298             .contentType(MediaType.APPLICATION_JSON)
299             .content(inputJson)
300             .andExpect(status().isCreated())
301             .andExpect(content().json(outputJson)))
302             .andExpect(status().isCreated());
303
304         mockMvc.perform(post("/entries/create")
305             .contentType(MediaType.APPLICATION_JSON)
306             .content(inputJson)
307             .andExpect(status().isCreated())
308             .andExpect(content().json(outputJson)))
309             .andExpect(status().isCreated());
310
311         mockMvc.perform(post("/entries/create")
312             .contentType(MediaType.APPLICATION_JSON)
313             .content(inputJson)
314             .andExpect(status().isCreated())
315             .andExpect(content().json(outputJson)))
316             .andExpect(status().isCreated());
317
318         mockMvc.perform(post("/entries/create")
319             .contentType(MediaType.APPLICATION_JSON)
320             .content(inputJson)
321             .andExpect(status().isCreated())
322             .andExpect(content().json(outputJson)))
323             .andExpect(status().isCreated());
324
325         mockMvc.perform(post("/entries/create")
326             .contentType(MediaType.APPLICATION_JSON)
327             .content(inputJson)
328             .andExpect(status().isCreated())
329             .andExpect(content().json(outputJson)))
330             .andExpect(status().isCreated());
331
332         mockMvc.perform(post("/entries/create")
333             .contentType(MediaType.APPLICATION_JSON)
334             .content(inputJson)
335             .andExpect(status().isCreated())
336             .andExpect(content().json(outputJson)))
337             .andExpect(status().isCreated());
338
339         mockMvc.perform(post("/entries/create")
340             .contentType(MediaType.APPLICATION_JSON)
341             .content(inputJson)
342             .andExpect(status().isCreated())
343             .andExpect(content().json(outputJson)))
344             .andExpect(status().isCreated());
345
346         mockMvc.perform(post("/entries/create")
347             .contentType(MediaType.APPLICATION_JSON)
348             .content(inputJson)
349             .andExpect(status().isCreated())
350             .andExpect(content().json(outputJson)))
351             .andExpect(status().isCreated());
352
353         mockMvc.perform(post("/entries/create")
354             .contentType(MediaType.APPLICATION_JSON)
355             .content(inputJson)
356             .andExpect(status().isCreated())
357             .andExpect(content().json(outputJson)))
358             .andExpect(status().isCreated());
359
360         mockMvc.perform(post("/entries/create")
361             .contentType(MediaType.APPLICATION_JSON)
362             .content(inputJson)
363             .andExpect(status().isCreated())
364             .andExpect(content().json(outputJson)))
365             .andExpect(status().isCreated());
366
367         mockMvc.perform(post("/entries/create")
368             .contentType(MediaType.APPLICATION_JSON)
369             .content(inputJson)
370             .andExpect(status().isCreated())
371             .andExpect(content().json(outputJson)))
372             .andExpect(status().isCreated());
373
374         mockMvc.perform(post("/entries/create")
375             .contentType(MediaType.APPLICATION_JSON)
376             .content(inputJson)
377             .andExpect(status().isCreated())
378             .andExpect(content().json(outputJson)))
379             .andExpect(status().isCreated());
380
381         mockMvc.perform(post("/entries/create")
382             .contentType(MediaType.APPLICATION_JSON
```

SPRINT

Epic	MAY W T F S S 25 26 27 28 29	JUN M T W T F S S 30 31 1 2 3 4 5	JUN M T W T F S S 6 7 8 9 10 11 12	JUN M T W T F S S 13 14 15 16 17 18 19	JUN M T W T F S S 20 21 22 23 24 25 26	T 27	F 28
▼ HP-29 Planning, Proof and Presenting							
✓ HP-30 Pre-Production Docs	DONE						
✓ HP-33 Presentation file	DONE						
✓ HP-34 ReadMe	DONE						
✓ HP-32 Diagrams	DONE						
▼ HP-28 Back-End							
✓ HP-4 Java	DONE						
✓ HP-2 Testing, Junit/Mockito	DONE						
✓ HP-6 MySQL	DONE						
✓ HP-20 Bug-Fixes and Building	DONE						
▼ HP-27 Front-End							
✓ HP-3 Html	DONE						
✓ HP-4 Css	DONE						
✓ HP-5 JavaScript	DONE						
▼ HP-34 User Stories							
✓ HP-35 As a User I would like to be able t...	DONE						
✓ HP-36 As a User I would like to create e...	DONE						
✓ HP-38 As a User I would like to Delete e...	DONE						
✓ HP-37 As a User I would like to Update e...	DONE						
✓ HP-39 As a User I would like to view entl...	DONE						
✓ HP-40 As a User I would like to have a b...	DONE						

PRODUCT EXAMPLE + EXPLANATION

28

John: Walk the dog

Start at: 1030 Taking 30 mins

29

John: Walk the dog

Start at: 430 Taking 30 mins

30

Jane: Washing up

Start at: 730 Taking 10 mins

31

Jane: Washing up

Start at: 1240 Taking 10 mins

32

Tommy: Washing up

Start at: 530 Taking 20 mins

33

Jimmy: Wash cloaths

Start at: 1800 Taking 10 mins

Refresh Entries

Add Entry

Name

Start Time

Expected Duration

Entry

Update an entry

Type id value

View an entry

Type id value

Select all entries by name

Type name

Delete By id

Type id value

IMPROVEMENTS

- Button Interactivity
- Delete, update ect on each entry
- More visuals I.e representation of time on the sticky notes

DEMO

CONCLUSION

QUESTIONS?