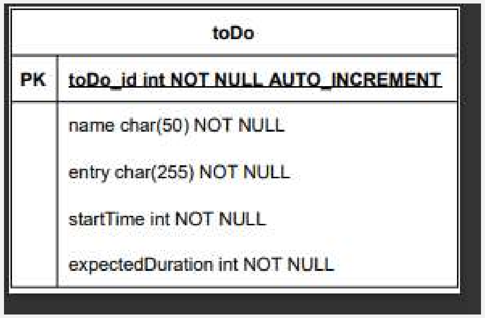
HWA PROJECT

- Jesse Morgan

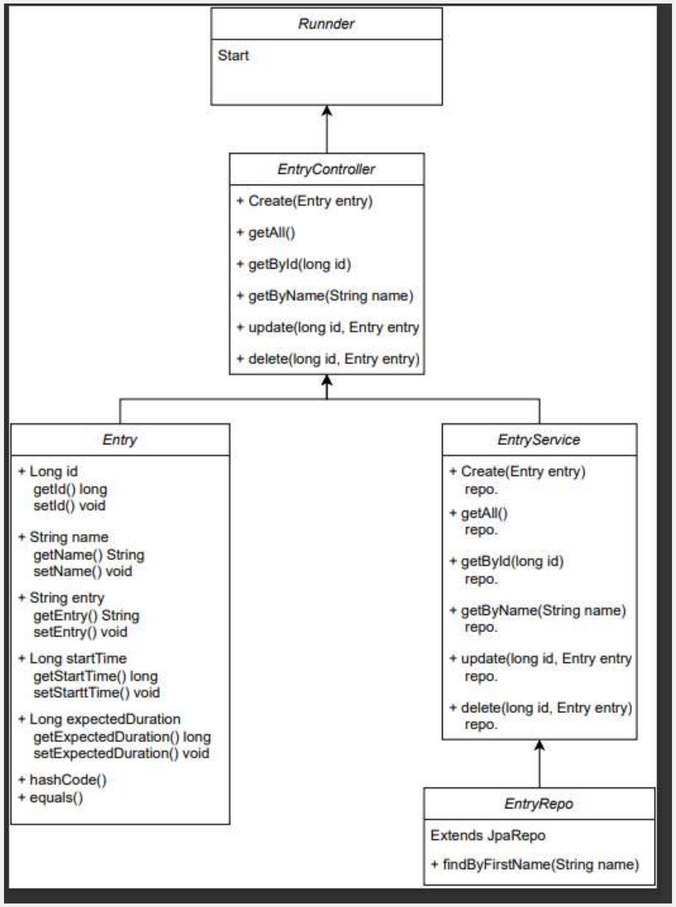
INTRODUCTION

- 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



Runnder

Start

EntryController

+ Create(Entry entry)
+ getAll()
+ getById(long id)
+ getByName(String name)
+ update(long id, Entry entry
+ delete(long id, Entry entry)

Entry

+ Long id
  getId() long
  setId() void
+ String name
  getName() String
  setName() void
+ String entry
  getEntry() String
  setEntry() void
+ Long startTime
  getStartTime() long
  setStartTime() void
+ Long expectedDuration
  getExpectedDuration() long
  setExpectedDuration() void
+ hashCode()
+ equals()

EntryService

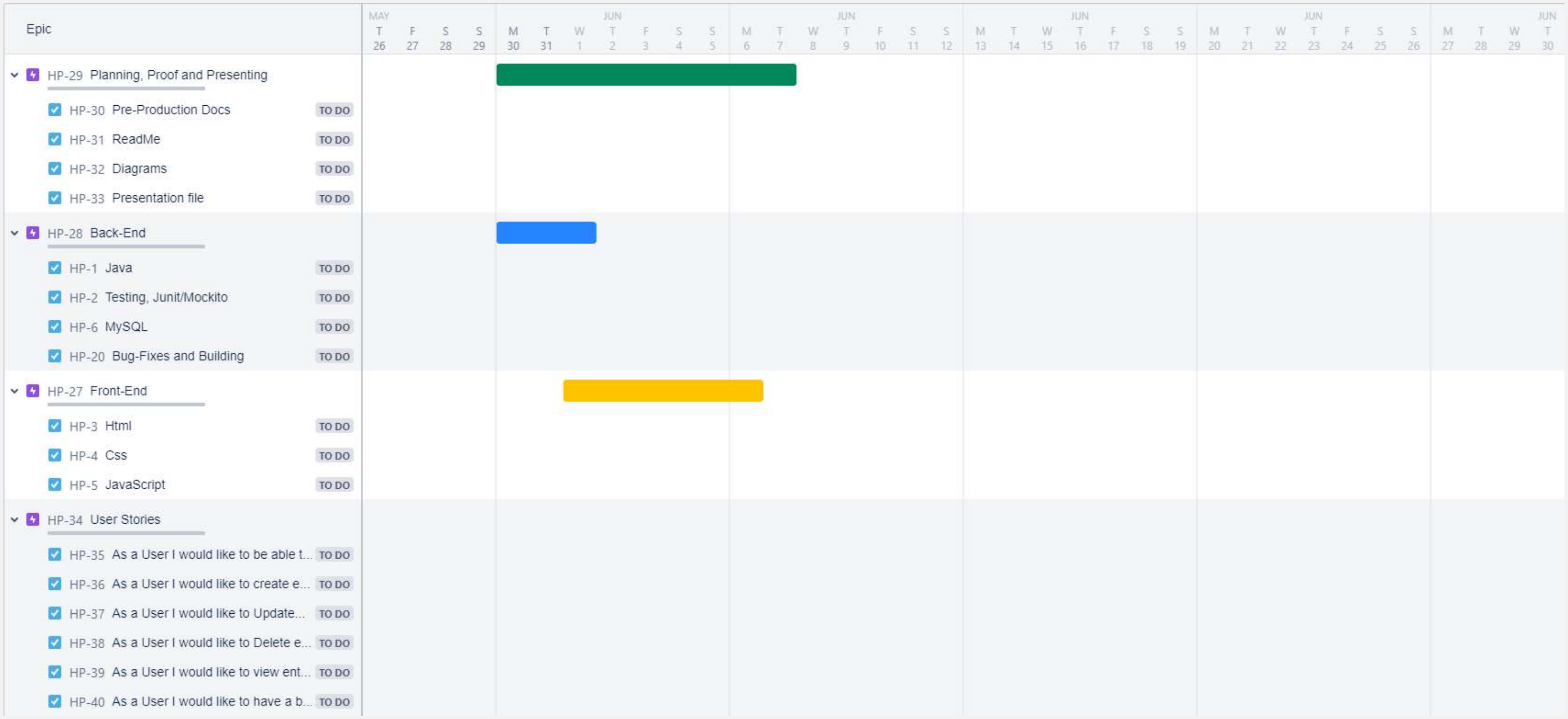
+ Create(Entry entry)
  repo.
+ getAll()
  repo.
+ getById(long id)
  repo.
+ getByName(String name)
  repo.
+ update(long id, Entry entry
  repo.
+ delete(long id, Entry entry)
  repo.

EntryRepo

Extends JpaRepo

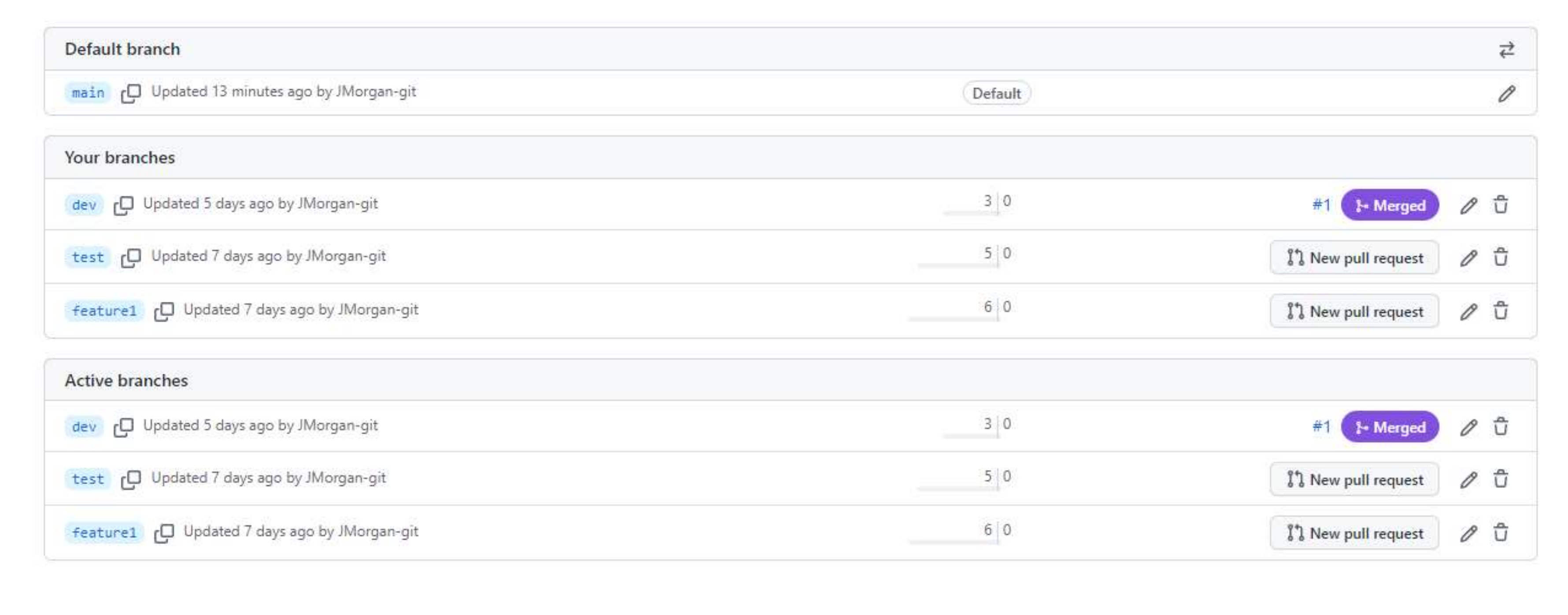
+ findByFirstName(String name)

Beginning Jira



Epic	
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 t...	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

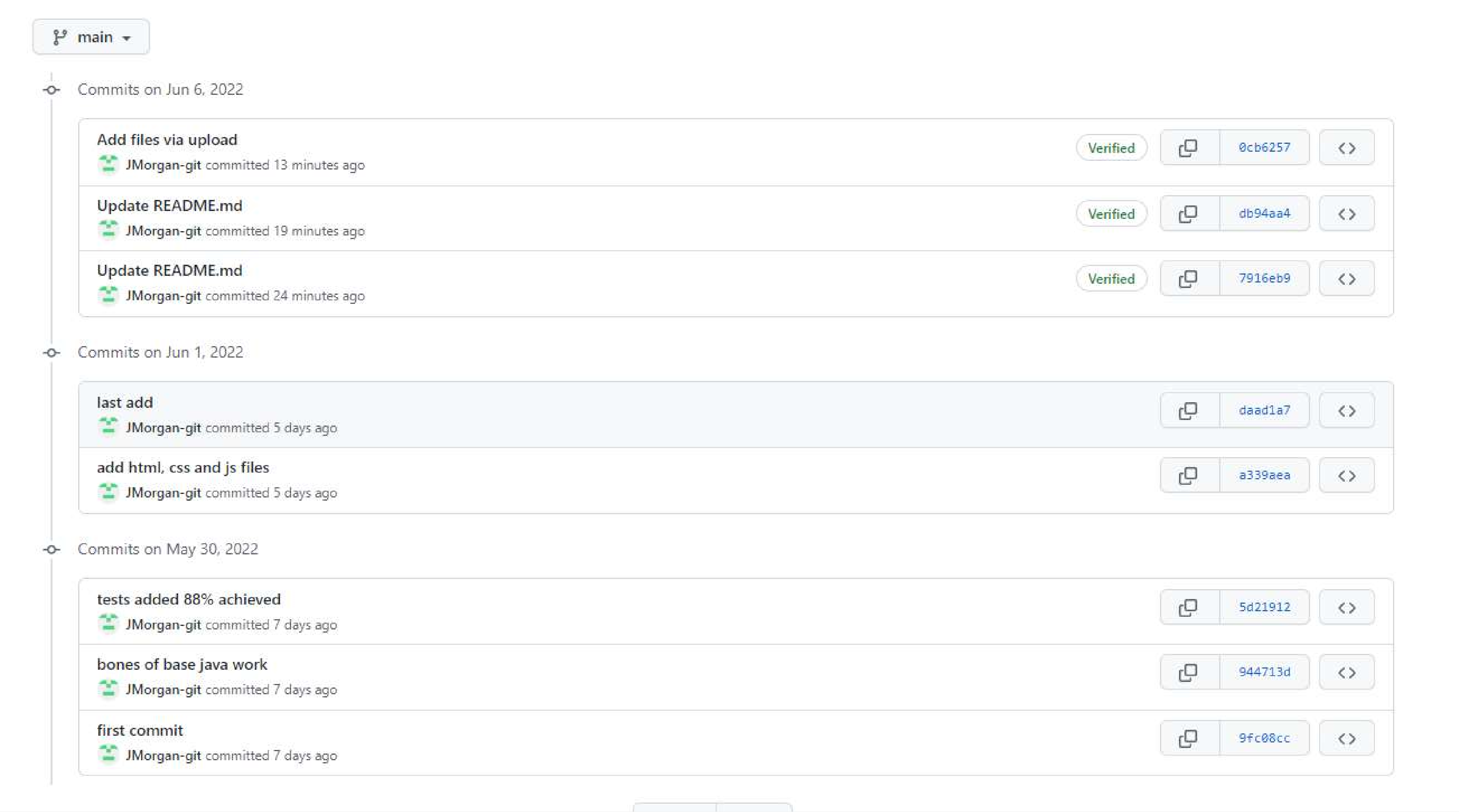
main	Updated 13 minutes ago by JMorgan-git	Default	

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



main

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

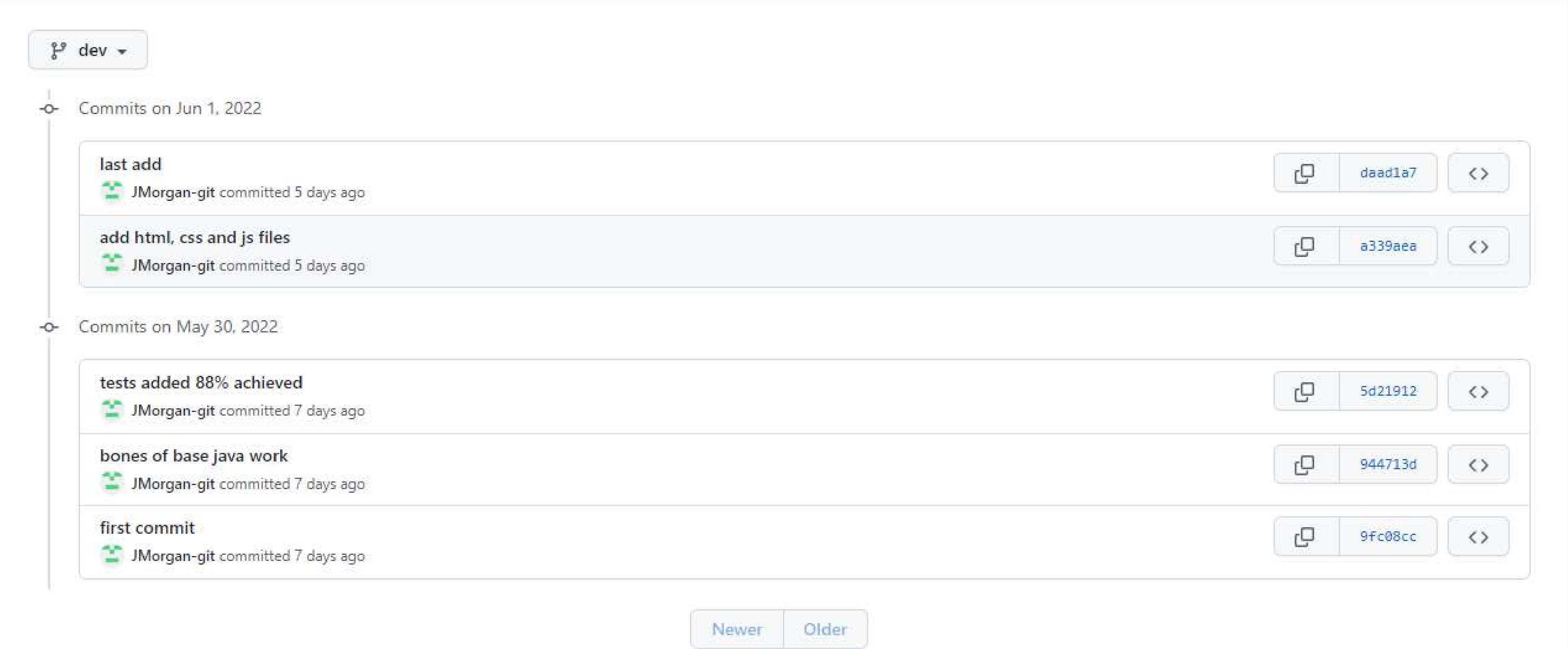
add html, css and js files
JMorgan-git committed 5 days ago
a339aea

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



Commits on Jun 1, 2022

last add
JMorgan-git committed 5 days ago daad1a7

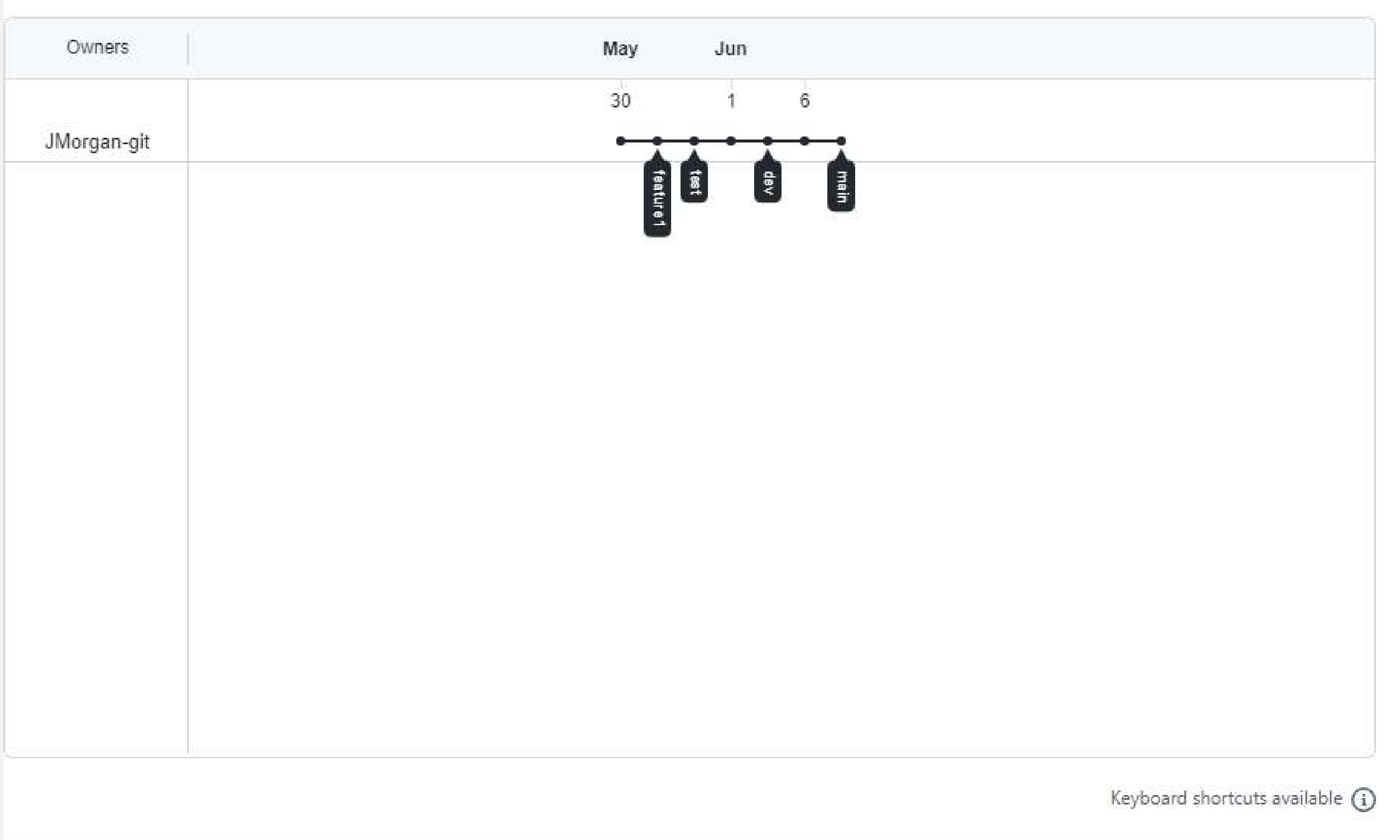
add html, css and js files
JMorgan-git committed 5 days ago a339aea

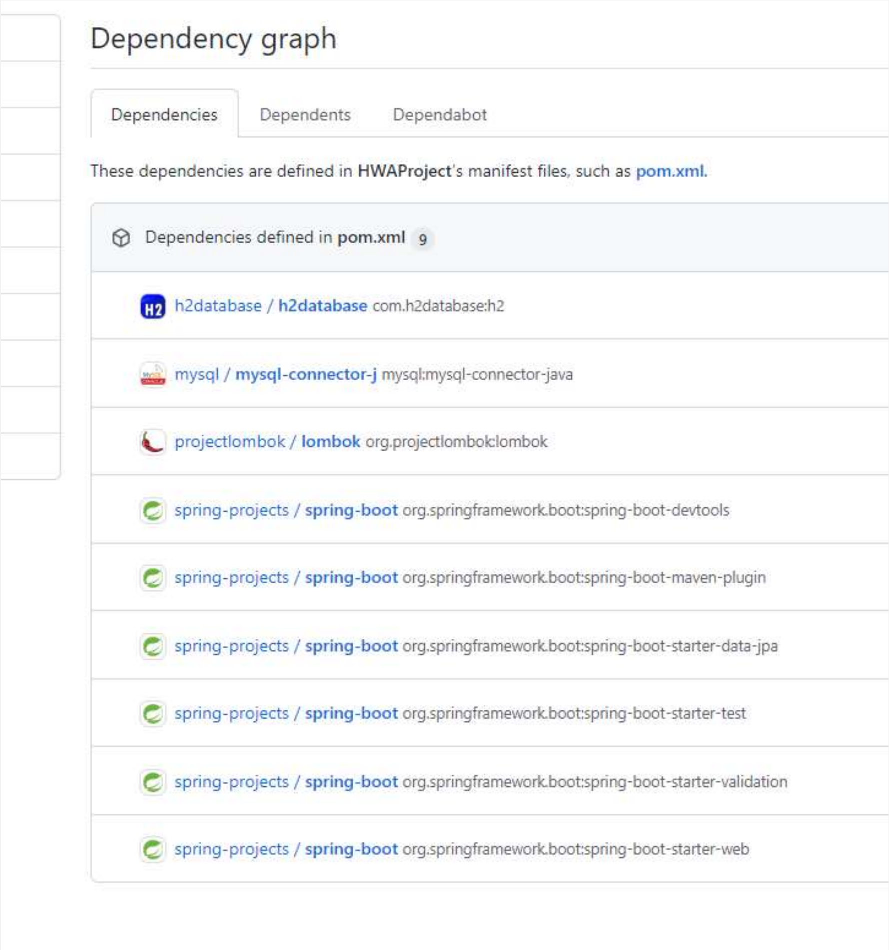
Commits on May 30, 2022

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


Owners
May
Jun
30
1
6
JMorgan-git
feature1
test
dev
main
Keyboard shortcuts available



Dependency graph

Dependencies Dependents Dependabot

These dependencies are defined in **HWAProject**'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

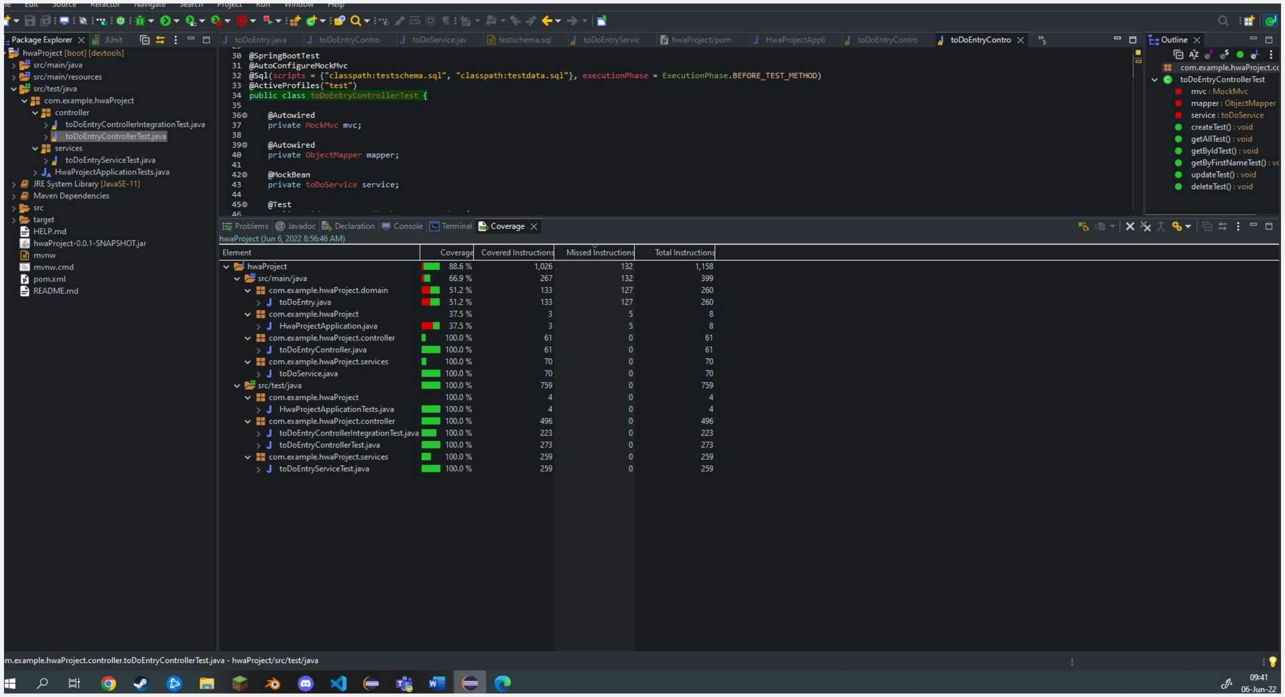
spring-projects / **spring-boot** org.springframework.boot:spring-boot-starter-data-jpa

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



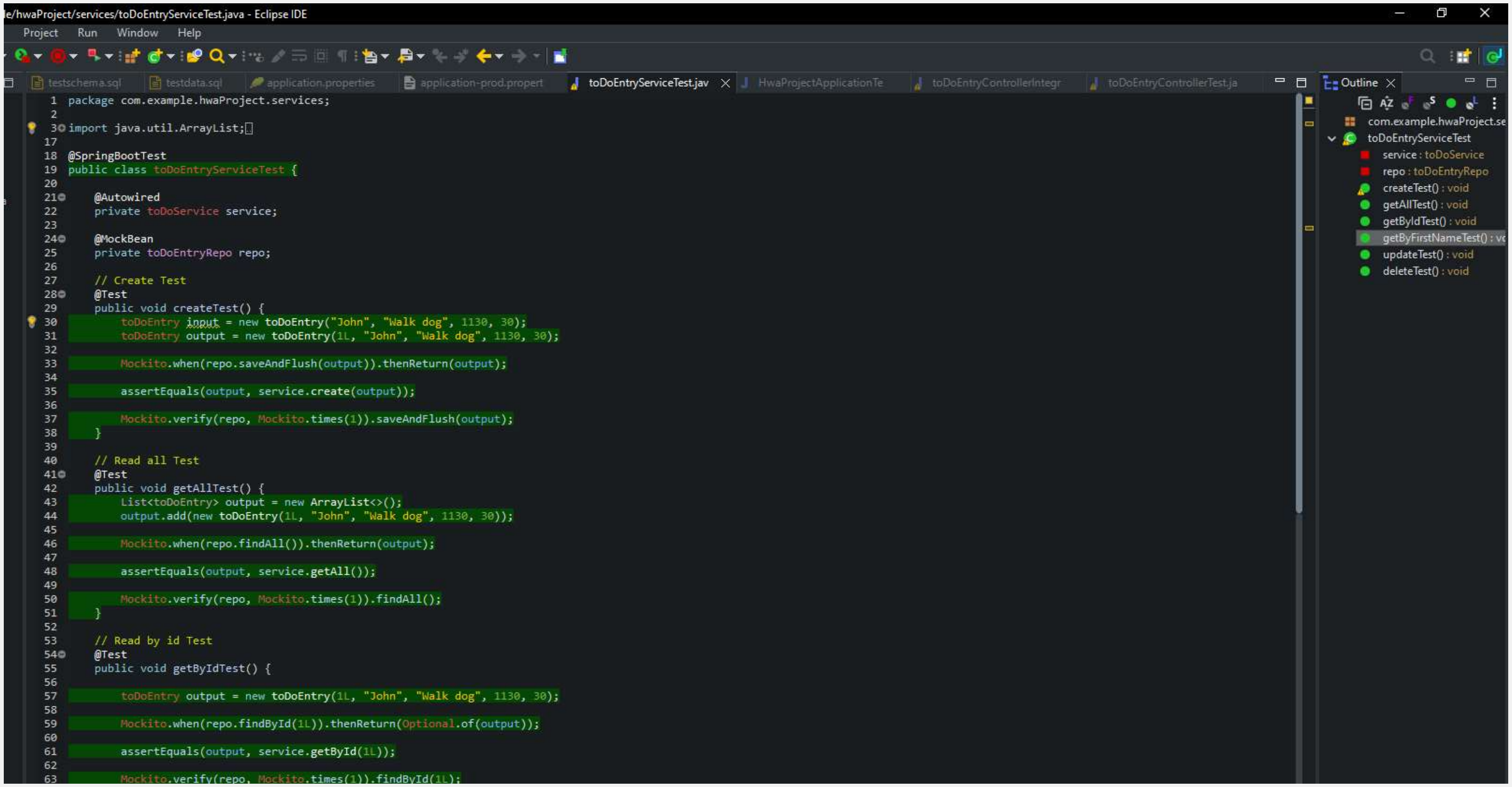
```
@SpringBootTest
@AutoConfigureMockMvc
@Sql(scripts = {"classpath:testschema.sql", "classpath:testdata.sql"}, executionPhase = ExecutionPhase.BEFORE_TEST_METHOD)
@ActiveProfiles("test")
public class toDoEntryControllerTest {

    @Autowired
    private MockMvc mvc;

    @Autowired
    private ObjectMapper mapper;

    @MockBean
    private toDoService service;

    @Test
```

hwaProject (Jun 6, 2022 8:56:46 AM)

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
hwaProject	88.6 %	1,026	132	1,158
src/main/java	66.9 %	267	132	399
com.example.hwaProject.domain	51.2 %	133	127	260
toDoEntry.java	51.2 %	133	127	260
com.example.hwaProject	37.5 %	3	5	8
HwaProjectApplication.java	37.5 %	3	5	8
com.example.hwaProject.controller	100.0 %	61	0	61
toDoEntryController.java	100.0 %	61	0	61
com.example.hwaProject.services	100.0 %	70	0	70
toDoService.java	100.0 %	70	0	70
src/test/java	100.0 %	759	0	759
com.example.hwaProject	100.0 %	4	0	4
HwaProjectApplicationTests.java	100.0 %	4	0	4
com.example.hwaProject.controller	100.0 %	496	0	496
toDoEntryControllerIntegrationTest.java	100.0 %	223	0	223
toDoEntryControllerTest.java	100.0 %	273	0	273
com.example.hwaProject.services	100.0 %	259	0	259
toDoEntryServiceTest.java	100.0 %	259	0	259

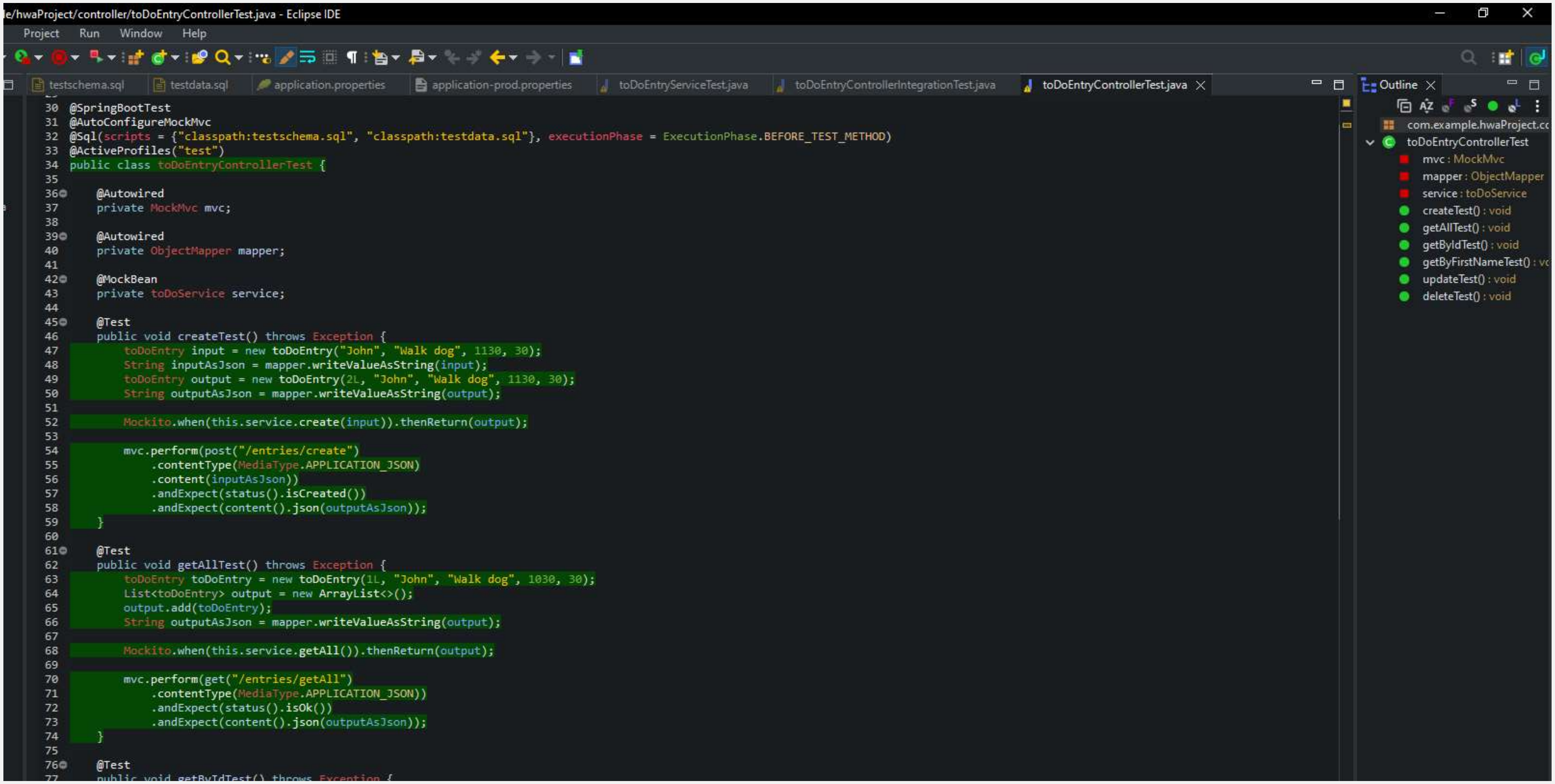
m.example.hwaProject.controller.toDoEntryControllerTest.java - hwaProject/src/test/java

SERVICE UNIT



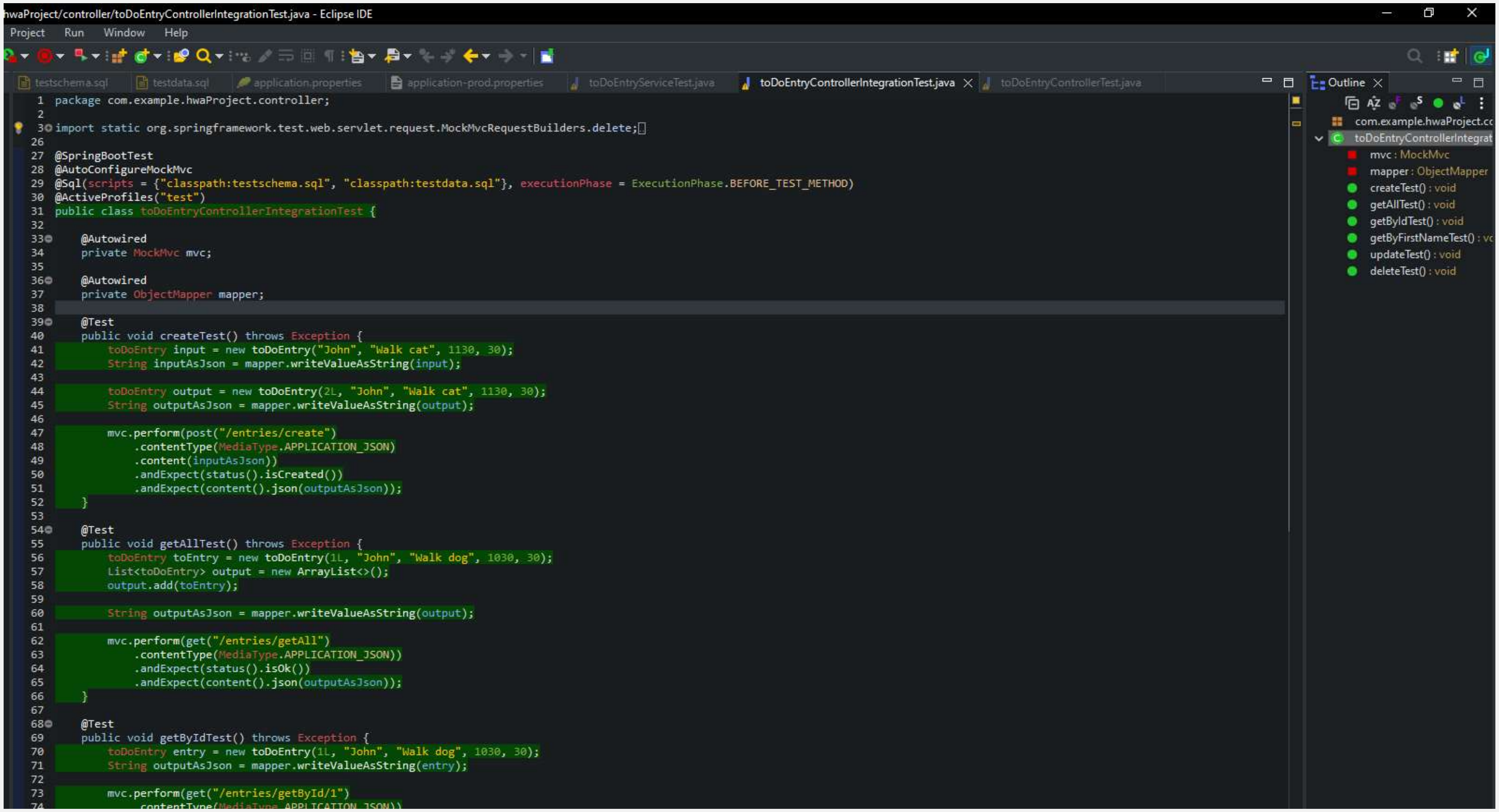
```java
package com.example.hwaProject.services;

import java.util.ArrayList;[]

@SpringBootTest
public class toDoEntryServiceTest {

    @Autowired
    private toDoService service;

    @MockBean
    private toDoEntryRepo repo;

    // Create Test
    @Test
    public void createTest() {
        toDoEntry input = new toDoEntry("John", "Walk dog", 1130, 30);
        toDoEntry output = new toDoEntry(1L, "John", "Walk dog", 1130, 30);

        Mockito.when(repo.saveAndFlush(output)).thenReturn(output);

        assertEquals(output, service.create(output));

        Mockito.verify(repo, Mockito.times(1)).saveAndFlush(output);
    }

    // Read all Test
    @Test
    public void getAllTest() {
        List<toDoEntry> output = new ArrayList<>();
        output.add(new toDoEntry(1L, "John", "Walk dog", 1130, 30));

        Mockito.when(repo.findAll()).thenReturn(output);

        assertEquals(output, service.getAll());

        Mockito.verify(repo, Mockito.times(1)).findAll();
    }

    // Read by id Test
    @Test
    public void getByIdTest() {

        toDoEntry output = new toDoEntry(1L, "John", "Walk dog", 1130, 30);

        Mockito.when(repo.findById(1L)).thenReturn(Optional.of(output));

        assertEquals(output, service.getById(1L));

        Mockito.verify(repo, Mockito.times(1)).findById(1L);
```

CONTROLLER UNIT



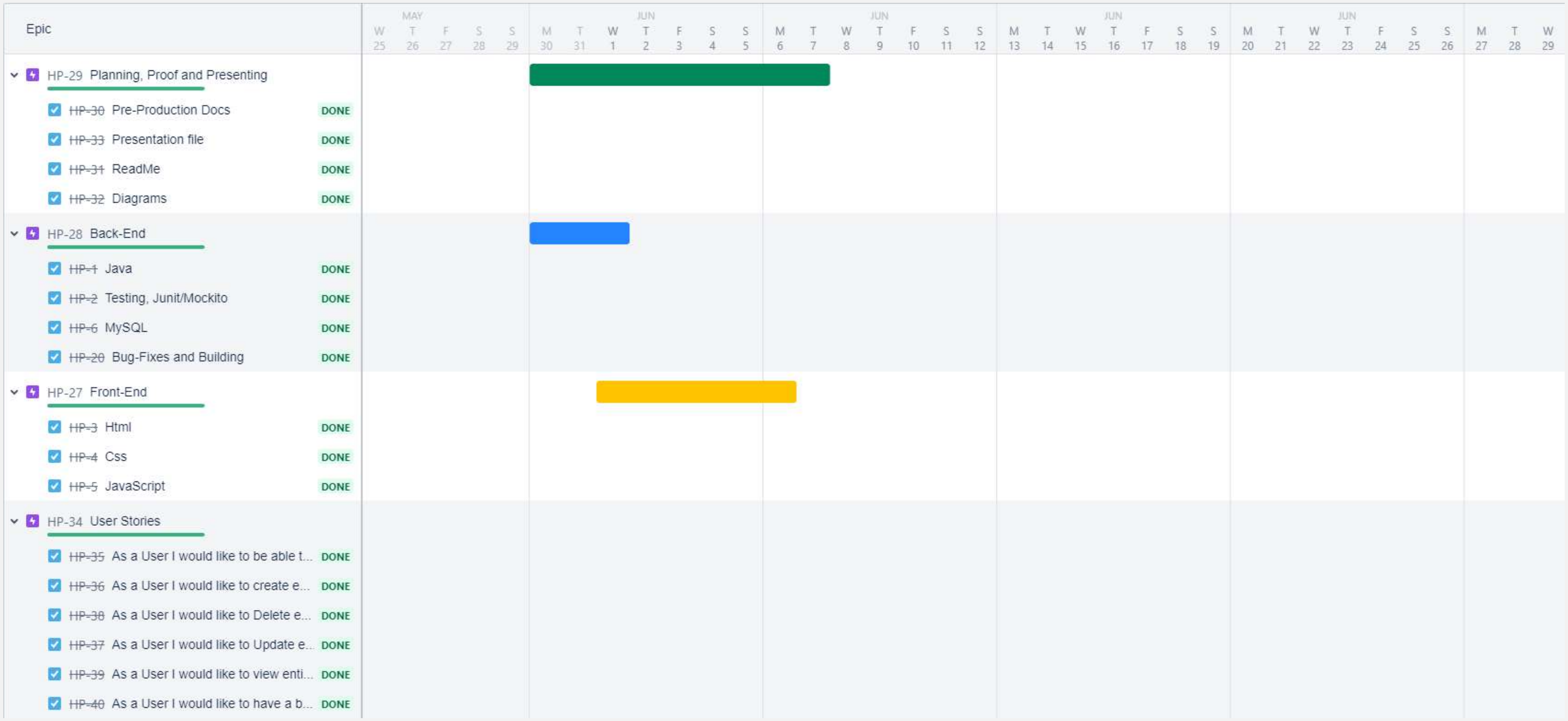
```java
@SpringBootTest
@AutoConfigureMockMvc
@Sql(scripts = {"classpath:testschema.sql", "classpath:testdata.sql"}, executionPhase = ExecutionPhase.BEFORE_TEST_METHOD)
@ActiveProfiles("test")
public class toDoEntryControllerTest {

    @Autowired
    private MockMvc mvc;

    @Autowired
    private ObjectMapper mapper;

    @MockBean
    private toDoService service;

    @Test
    public void createTest() throws Exception {
        toDoEntry input = new toDoEntry("John", "Walk dog", 1130, 30);
        String inputAsJson = mapper.writeValueAsString(input);
        toDoEntry output = new toDoEntry(2L, "John", "Walk dog", 1130, 30);
        String outputAsJson = mapper.writeValueAsString(output);

        Mockito.when(this.service.create(input)).thenReturn(output);

        mvc.perform(post("/entries/create")
            .contentType(MediaType.APPLICATION_JSON)
            .content(inputAsJson))
            .andExpect(status().isCreated())
            .andExpect(content().json(outputAsJson));
    }

    @Test
    public void getAllTest() throws Exception {
        toDoEntry toDoEntry = new toDoEntry(1L, "John", "Walk dog", 1030, 30);
        List<toDoEntry> output = new ArrayList<>();
        output.add(toDoEntry);
        String outputAsJson = mapper.writeValueAsString(output);

        Mockito.when(this.service.getAll()).thenReturn(output);

        mvc.perform(get("/entries/getAll")
            .contentType(MediaType.APPLICATION_JSON))
            .andExpect(status().isOk())
            .andExpect(content().json(outputAsJson));
    }

    @Test
    public void getByIdTest() throws Exception {
```

CONTROLLER INTEGRATION


hwaProject/controller/toDoEntryControllerIntegrationTest.java - Eclipse IDE
Project
Run
Window
Help
testschema.sql
testdata.sql
application.properties
application-prod.properties
toDoEntryServiceTest.java
toDoEntryControllerIntegrationTest.java
toDoEntryControllerTest.java
Outline
com.example.hwaProject.cc
toDoEntryControllerIntegrat
mvc : MockMvc
mapper : ObjectMapper
createTest() : void
getAllTest() : void
getByIdTest() : void
getByFirstNameTest() : vc
updateTest() : void
deleteTest() : void

```java
package com.example.hwaProject.controller;

import static org.springframework.test.web.servlet.request.MockMvcRequestBuilders.delete;

@SpringBootTest
@AutoConfigureMockMvc
@Sql(scripts = {"classpath:testschema.sql", "classpath:testdata.sql"}, executionPhase = ExecutionPhase.BEFORE_TEST_METHOD)
@ActiveProfiles("test")
public class toDoEntryControllerIntegrationTest {

    @Autowired
    private MockMvc mvc;

    @Autowired
    private ObjectMapper mapper;

    @Test
    public void createTest() throws Exception {
        toDoEntry input = new toDoEntry("John", "Walk cat", 1130, 30);
        String inputAsJson = mapper.writeValueAsString(input);

        toDoEntry output = new toDoEntry(2L, "John", "Walk cat", 1130, 30);
        String outputAsJson = mapper.writeValueAsString(output);

        mvc.perform(post("/entries/create")
            .contentType(MediaType.APPLICATION_JSON)
            .content(inputAsJson))
            .andExpect(status().isCreated())
            .andExpect(content().json(outputAsJson));
    }

    @Test
    public void getAllTest() throws Exception {
        toDoEntry toEntry = new toDoEntry(1L, "John", "Walk dog", 1030, 30);
        List<toDoEntry> output = new ArrayList<>();
        output.add(toEntry);

        String outputAsJson = mapper.writeValueAsString(output);

        mvc.perform(get("/entries/getAll")
            .contentType(MediaType.APPLICATION_JSON))
            .andExpect(status().isOk())
            .andExpect(content().json(outputAsJson));
    }

    @Test
    public void getByIdTest() throws Exception {
        toDoEntry entry = new toDoEntry(1L, "John", "Walk dog", 1030, 30);
        String outputAsJson = mapper.writeValueAsString(entry);

        mvc.perform(get("/entries/getById/1")
            .contentType(MediaType.APPLICATION_JSON))
```

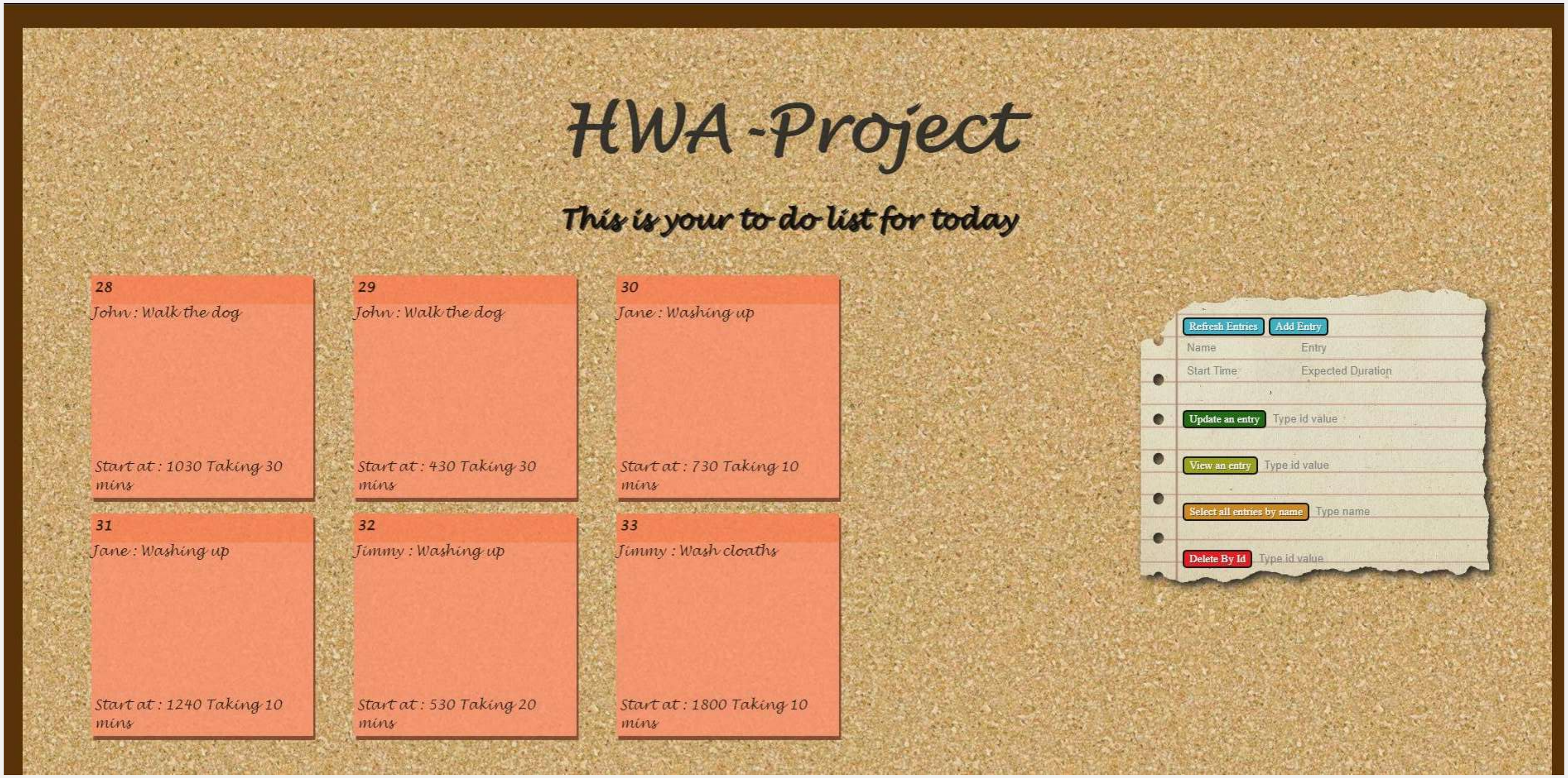
SPRINT



Epic

- HP-29 Planning, Proof and Presenting
  - ☑ ~~HP-30~~ Pre-Production Docs DONE
  - ☑ ~~HP-33~~ Presentation file DONE
  - ☑ ~~HP-31~~ ReadMe DONE
  - ☑ ~~HP-32~~ Diagrams DONE
- HP-28 Back-End
  - ☑ ~~HP-1~~ 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 enti... DONE
  - ☑ ~~HP-40~~ As a User I would like to have a b... DONE

PRODUCT EXAMPLE + EXPLANATION



HWA -Project

This is your to do list for today

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
Jimmy : Washing up
Start at : 530 Taking 20 mins

33
Jimmy : Wash cloaths
Start at : 1800 Taking 10 mins

IMPROVEMENTS

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

CONCLUSION

QUESTIONS?