

Introduction

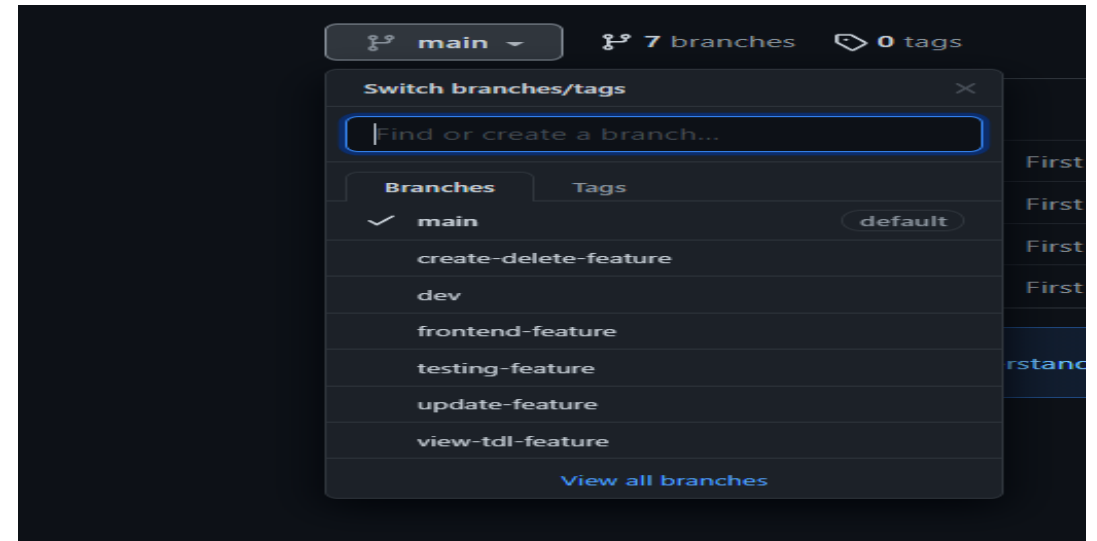
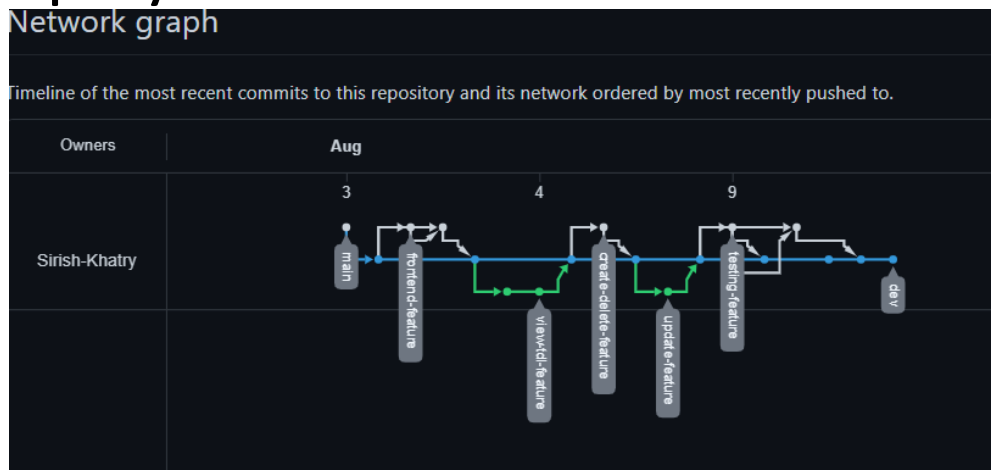
My name is Sirish Khatri and the way I approached the specification is by having a thorough read through and while doing that taking notes of important information's and requirements. Also, during the development I kept on reflecting back to the specs time to time to make sure the requirements were met and I was following the right procedures.

Technologies Learned

- Git for version control using feature-branch model
- Jira for story boards, epics and sprints
- Agile methodology
- HTML, CSS and JS for front-end development
- SQL Database using MySQL Workbench
- Unit testing using Junit framework
- Integration testing using Mockito
- User-Acceptance testing using Selenium

Version Control

For this project we learned to use Git for version control and that is what I used to implement the feature-branch model. Where we branch out from main to dev then to a different feature branch and then merging that with dev branch once completed and repeating this feature-branch model for every new feature we work on until every feature is completed and finally merging from dev to main for deployment.



Unit & Integration Testing

Unit testing and Integration method was used for testing with Junit and Mockito framework. All the controller, domain and service classes and methods within them were tested.

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
tdl-java	95.8 %	715	31	746
src/main/java	88.7 %	244	31	275
com.qa.tdl	37.5 %	3	5	8
com.qa.tdl.controller	80.4 %	74	18	92
com.qa.tdl.entity	100.0 %	105	0	105
com.qa.tdl.service	88.6 %	62	8	70
src/test/java	100.0 %	471	0	471

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

    Item item = new Item("Test-1", "Completed");

    String itemAsJSON = this.mapper.writeValueAsString(item);

    RequestBuilder mockRequest = post("/item/create")
        .contentType(MediaType.APPLICATION_JSON)
        .content(itemAsJSON);

    Item savedItem = new Item(2, "Test-1", "Completed");

    String savedItemAsJSON = this.mapper.writeValueAsString(savedItem);

    ResultMatcher matchStatus = status().isCreated();

    ResultMatcher matchBody = content().json(savedItemAsJSON);
    System.out.println(itemAsJSON);
    System.out.println(savedItemAsJSON);
    this.mock.perform(mockRequest).andExpect(matchBody).andExpect(matchStatus);

}

@Test
public void testGetAll() throws Exception {

    RequestBuilder mockRequest = get("/item/getAll")
        .contentType(MediaType.APPLICATION_JSON);

    Item savedItem = new Item(1, "Test-1", "Completed");
    List<Item> items = new ArrayList<>();
    items.add(savedItem);

    String savedItemAsJSON = this.mapper.writeValueAsString(items);

    ResultMatcher matchStatus = status().isOk();
    ResultMatcher matchBody = content().json(savedItemAsJSON);

    this.mock.perform(mockRequest).andExpect(matchBody).andExpect(matchStatus);

}
```

User Acceptance Testing

Selenium automated testing suite was used for testing the front-end for user acceptance.

```
@FindBy(xpath = "//*[@id=\"deleteBtn\"]")
private WebElement deleteButton;           // DELETE
@FindBy(xpath = "//*[@id=\"itemIDdel\"]")
private WebElement delIdField;
@FindBy(xpath = "//*[@id=\"deleteSubmit\"]")
private WebElement deleteSubmit;

public String pageTitleResult() {
    return pageTitle.getText();
}

public String createItem(String item) {
    createButton.click();
    textField.sendKeys(item);
    createSubmit.click();

    return prompt.getText();
}

public void updateItem(String item, String id, String status) {
    updateButton.click();
    idField.sendKeys(id);
    updateItemField.sendKeys(item);
    updateStatusField.sendKeys(status);
    updateSubmit.click();
}

public String updateItemReturn() {
    return prompt.getText();
}

public String deleteItem(String id) {
    deleteButton.click();
    delIdField.sendKeys(id);
    deleteSubmit.click();
    return prompt.getText();
}
```

```
@SpringBootTest
class homePageTest {

    private static WebDriver driver;

    @BeforeAll
    public static void setup() {
        driver = new ChromeDriver();
        driver.manage().window().maximize();
    }

    @Test
    public void homePageCheck() throws InterruptedException {

        String test = "To-Do-List";

        homePage page = PageFactory.initElements(driver, homePage.class);

        driver.get(page.URL);

        Thread.sleep(6000);

        assertEquals(test, page.pageTitleResult());
    }

    @Test
    public void createItemCheck() throws InterruptedException {

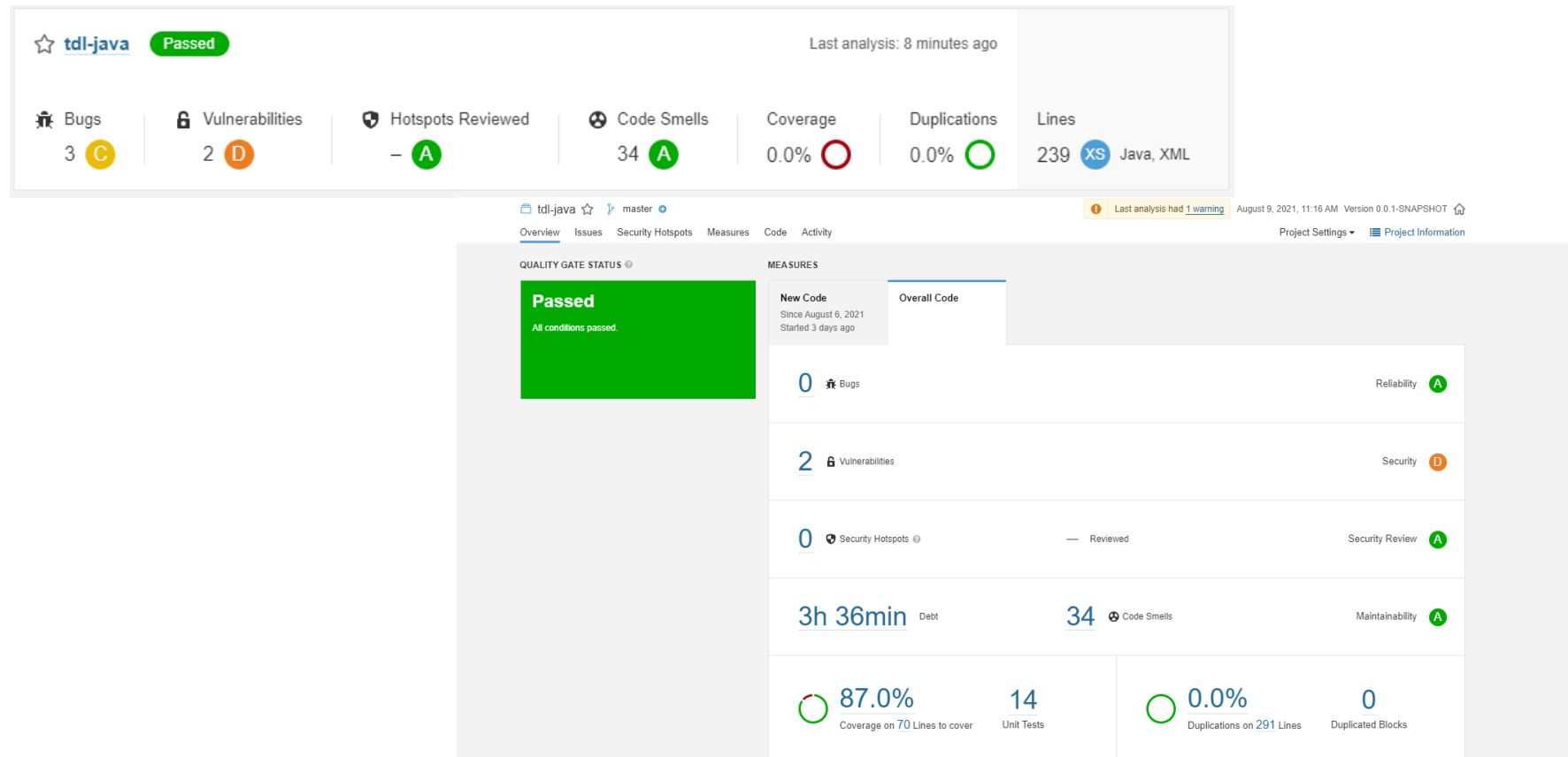
        String expected = "Created";

        homePage page = PageFactory.initElements(driver, homePage.class);
        driver.get(page.URL);
        Thread.sleep(6000);

        assertEquals(expected, page.createItem(expected));
    }
}
```

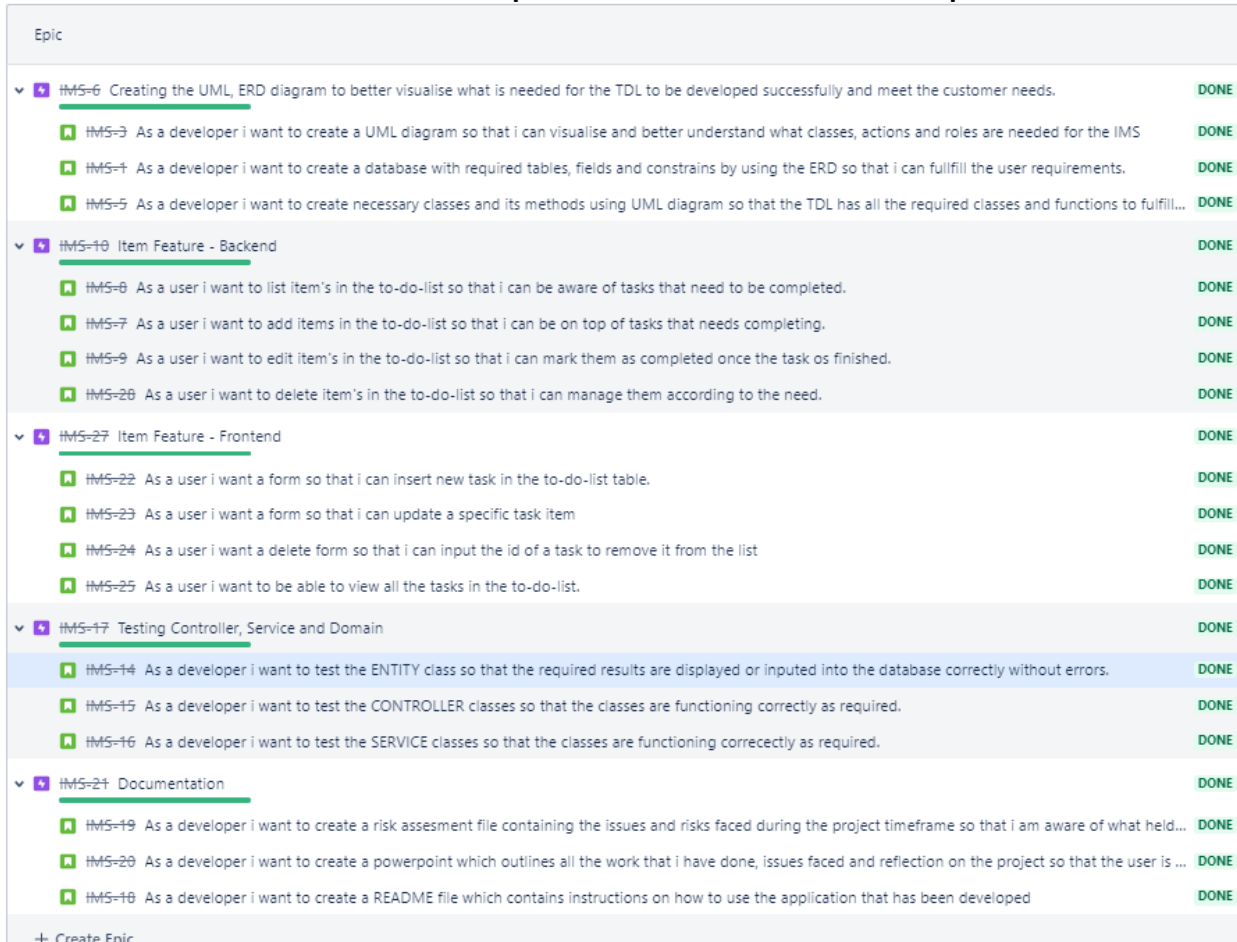
Static Analysis

SonarQube was used to perform static analysis on the codes to detect bugs, code smells and security vulnerabilities.



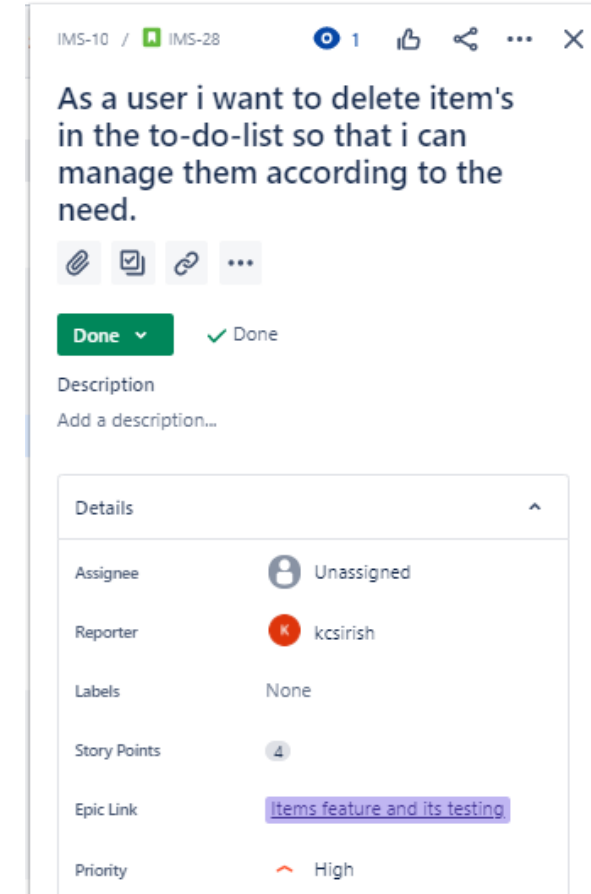
Epics and Stories

Jira software was used to create Kanban board containing epics, users stories and their sprints. The stories were prioritised and stories points were also implemented.



The screenshot shows a Jira Epic view for an epic titled "Epic". It contains a list of user stories, each with a status of "DONE". The stories are organized into sections: "Item Feature - Backend", "Item Feature - Frontend", "Testing Controller, Service and Domain", and "Documentation".

- Item Feature - Backend**
 - HM5-6 Creating the UML, ERD diagram to better visualise what is needed for the TDL to be developed successfully and meet the customer needs. DONE
 - HM5-3 As a developer i want to create a UML diagram so that i can visualise and better understand what classes, actions and roles are needed for the IMS. DONE
 - HM5-4 As a developer i want to create a database with required tables, fields and constrains by using the ERD so that i can fulfill the user requirements. DONE
 - HM5-5 As a developer i want to create necessary classes and its methods using UML diagram so that the TDL has all the required classes and functions to fulfill... DONE
- Item Feature - Frontend**
 - HM5-8 As a user i want to list item's in the to-do-list so that i can be aware of tasks that need to be completed. DONE
 - HM5-7 As a user i want to add items in the to-do-list so that i can be on top of tasks that needs completing. DONE
 - HM5-9 As a user i want to edit item's in the to-do-list so that i can mark them as completed once the task os finished. DONE
 - HM5-28 As a user i want to delete item's in the to-do-list so that i can manage them according to the need. DONE
- Testing Controller, Service and Domain**
 - HM5-22 As a user i want a form so that i can insert new task in the to-do-list table. DONE
 - HM5-23 As a user i want a form so that i can update a specific task item. DONE
 - HM5-24 As a user i want a delete form so that i can input the id of a task to remove it from the list. DONE
 - HM5-25 As a user i want to be able to view all the tasks in the to-do-list. DONE
- Documentation**
 - HM5-17 Testing Controller, Service and Domain. DONE
 - HM5-14 As a developer i want to test the ENTITY class so that the required results are displayed or inputed into the database correctly without errors. DONE
 - HM5-15 As a developer i want to test the CONTROLLER classes so that the classes are functioning correctly as required. DONE
 - HM5-16 As a developer i want to test the SERVICE classes so that the classes are functioning correctly as required. DONE
 - HM5-21 Documentation. DONE
 - HM5-19 As a developer i want to create a risk assesment file containing the issues and risks faced during the project timeframe so that i am aware of what held... DONE
 - HM5-20 As a developer i want to create a powerpoint which outlines all the work that i have done, issues faced and reflection on the project so that the user is ... DONE
 - HM5-18 As a developer i want to create a README file which contains instructions on how to use the application that has been developed. DONE



The screenshot shows a Jira Story view for a story titled "As a user i want to delete item's in the to-do-list so that i can manage them according to the need." The story is assigned to "kcsirish" and has a status of "Done". The details section shows the following information:

- Assignee: Unassigned
- Reporter: kcsirish
- Labels: None
- Story Points: 4
- Epic Link: Items feature and its testing
- Priority: High

Sprint Review

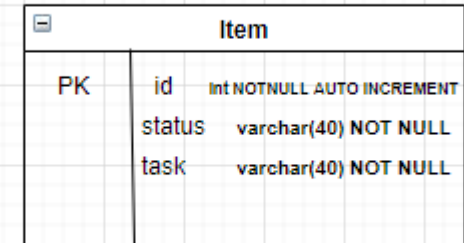
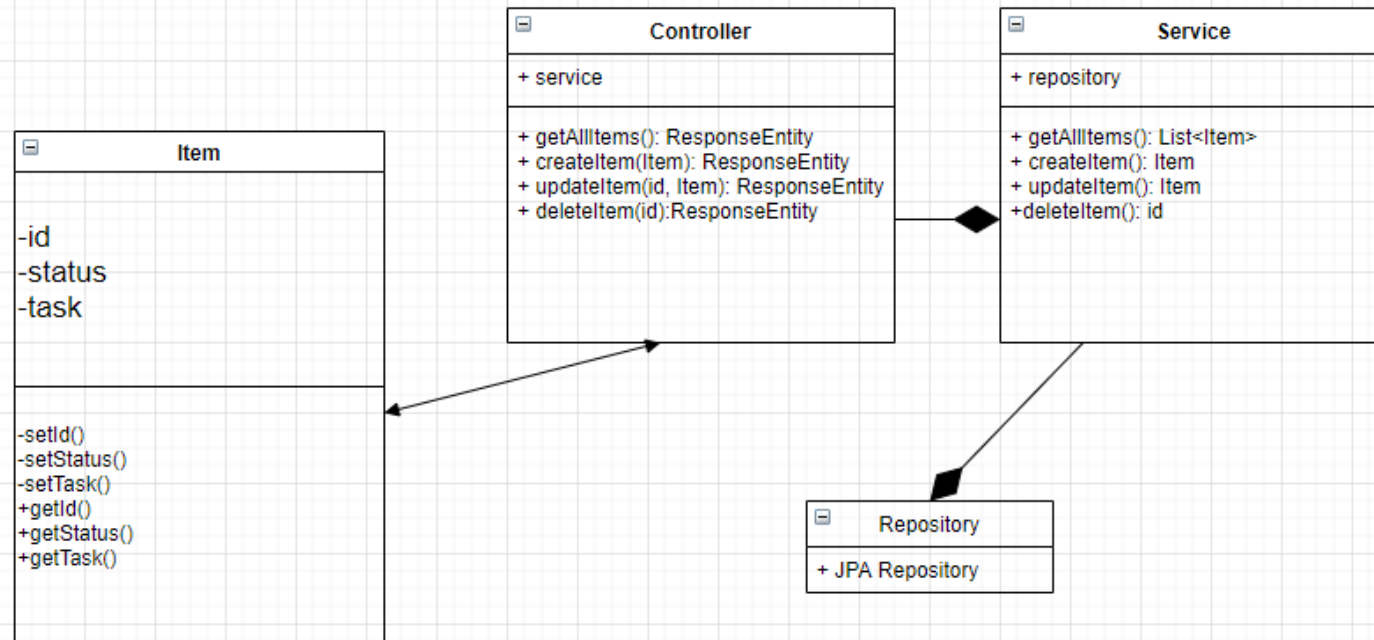
All the user stories relating the minimum viable product were completed however, I had thought of adding more complex features such as accounts feature. If there was more time, the front-end could be improved to look more neat and full of information rather than just one page with all the features.

Sprint Retrospective

Most of the sprints were completed on the set deadline, only few were delayed by few hours due to unexpected errors.

Stories within some of my sprints could have been more detailed and focused on a specific feature rather than two or more features.

UML & ERD



Risk Assessment

	Risk	Statement	Response	Objective	Likelihood	Impact
1	Prioritising tasks	If i focus on a feature which is not important i wont have the minimum viable product ready.	While working on the project minimum viable product should be main priority.	Prioritising tasks to leave any unimportant task for last.	8	20
2	Testing Service Layer	While testing the service layer I struggled with testing the update and delete.	I went through a lot of research online and tried various different ways but couldn't sort out the issue.	Looking through related classes to find issues and solving them effectively.	8	12
3	Test-Data Error	While using the h2 database for testing, the test result wasn't as expected.	I asked for help with the trainers and did my own research aswell.	Making sure I seek help when stuck on a problem for too long.	15	20
4	Sonarqube	I was having issue getting the coverage of the test.	I went through the community page and asked help for with the trainers	Seek for help and go through the materials on community page to get more information on the topic.	6	12

Conclusion

This project has been really good as it has allowed me to integrate different environments to the application using the N-tier architecture and its Presentation, Business and Data layers. This has given me more knowledge on how we can implement front-end, back-end and database into a application using different environments. In regards to testing, this project has taught me how user-acceptance testing can be performed in the front-end using Selenium IDE. More practise on continuous integration using Git has also made me more comfortable with using the tool. In the future this project could be improved with more than one entity allowing users to login in and store their own to-do-list and perform actions with them. The front-end could also be improved using more JS to make it look more like an application rather than a website.