Hobby Web Application (HWA)

Amandeep Bassi

Introduction

- Agile Planning methodology
- Jira User stories and targets
- Git / Github Keep track of the development
- MySQL Data driven application, database hosted on a local instance
- Spring Boot Java framework to build backend API
- HTML / JS Build web frontend
- CSS / Bootstrap Apply styles and modal functionality
- JUnit / Mockito Testing the application
- Selenium Web browser automation for frontend testing
- Swagger Visualise and describe API structure

Concept

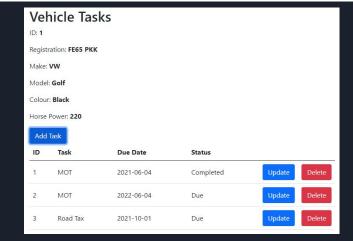
- A web application which helps users keep track of vehicle maintenance.
- Many vehicles can be stored in to the system.
- Each vehicle can have its own record of tasks e.g.(Insurance, Road tax)
- Each task has a due date and status



Vehicles



Fleet Management System · © 2021



Risk assessment

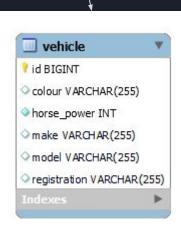
	70.23	X	Impact Level	
		Low	Medium	High
8	Low	1	3	4
Likelihood	Medium	3	4	5
3	High	4	5	6

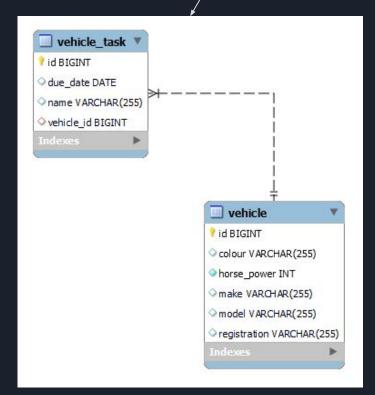
Project deadline moves forward	Application is required earlier than expected	Low	High	Stakeholder	Work overtime to complete the project faster	Plan some extra time in to the project schedule to allow for unexpected issues
Late feature request	More project complexity	Medium	Medium	Stakeholder	Use new code branch to implement and test new feature	Communicate and discuss complete project requirements before development and have regular sprint meetings
Poor development strategy	Software development takes longer for the same task	Medium	Medium	Developers	Change development strategy to improve coding efficiency	Complete the backend first then move on to the frontend to save time having to go back and forth between code

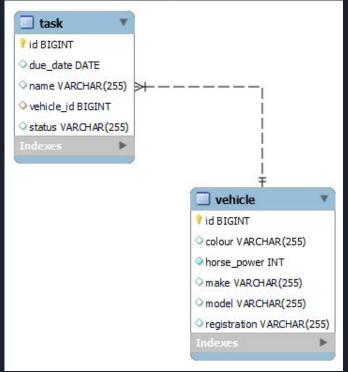
Sprint Plan

~ 🗗	HWA-5 Vehicle							
	■ HWA-7 As a user, I want to record a vehicle so that I can assign and track vehicle tasks	DONE	AB					
	☐ HWA-8 As a user, I want to view a list of all vehicles so that I can see which vehicle has the next task to complete	DONE	AB					
	■ HWA-9 As a user, I want to update vehicle details so that the system can adapt to vehicle modifications	DONE	AB					
	■ HWA-10 As a user, I want to delete a vehicle so that the system does not track vehicles that are no longer part o	DONE	AB					
~ []	HWA-6 Vehicle Tasks							
	■ HWA-11 As a user, I want to add tasks to a vehicle so that I can see what tasks a vehicle has	DONE	AB					
	☐ HWA-12 As a user, I want to see a list of all tasks associated to a vehicle so that I can see when and what tasks a	DONE	AB					
	☐ HWA-13 As a user, I want to be able to update task details so that amend task details	DONE	AB					
	HWA-14 As a user, I want to be able to delete a task so that if a task was created by mistake on the wrong vehicl	DONE	AB					

Version 1



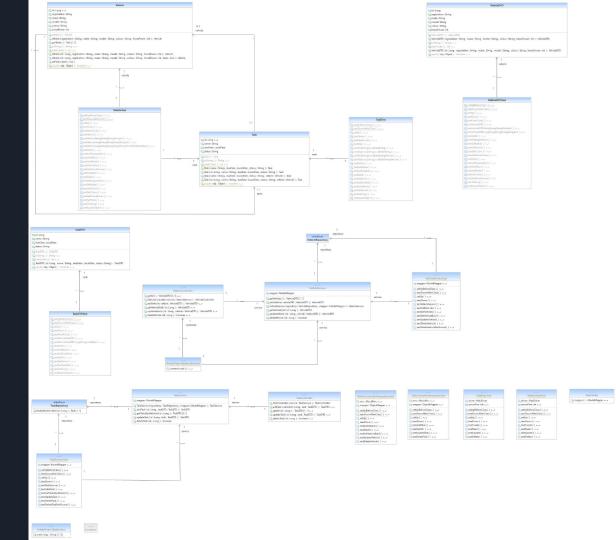




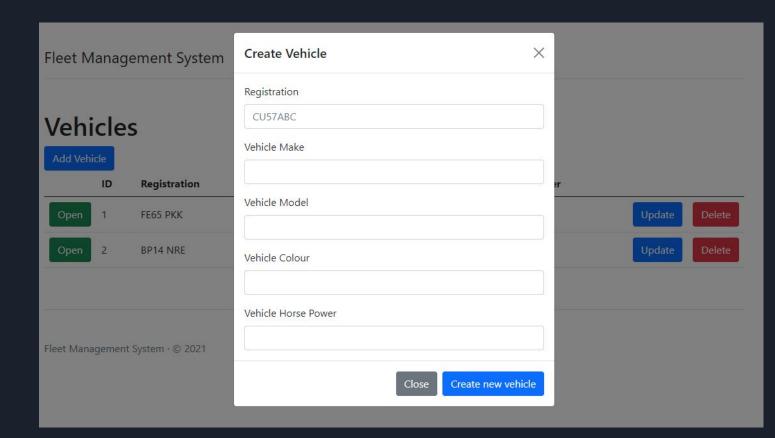
Rest API

Java Spring Boot framework

Postman



HTML/JS/CSS



Git / GitHub

```
* abd6c69 - Updated README file - Amandeep Bassi (origin/dev, dev)
* c4edccd - Create README.md - AmanBassi
 ac81604 - Merge branch 'dev' - Amandeep Bassi
   26ba02a - Merge pull request #15 from AmanBassi/refactoring-2 - AmanBassi
  * 2baf70b - Declared some local variables as var following sonarqube recommendation - Amandeep Bassi (origin/refactoring-2, refactoring-2)
  6delaef - Merge pull request #14 from AmanBassi/dev - AmanBassi
   be736fc - Merge pull request #13 from AmanBassi/task-status - AmanBassi
  * 329e47b - Added status variable to task object and tested - Amandeep Bassi (origin/task-status, task-status)
 422c903 - Merge pull request #12 from AmanBassi/dev - AmanBassi
   Odd6fb4 - Merge pull request #11 from AmanBassi/refactoring - AmanBassi
 * 2f1618d - Sonarqube refactor possible suggestions carried out - Amandeep Bassi (origin/refactoring, refactoring)
 * 37fc09a - Vehicle create method updated to take in vehicle DTO - Amandeep Bassi
 * 7925cbb - Task create method updated take in DTOs - Amandeep Bassi
 * ef39f2a - Vehicle and Task update method take in DTOs - Amandeep Bassi
 * 883b8ef - Renamed VehicleTask to Task better readability - Amandeep Bassi
 * b97624e - Updated web test for task update - Amandeep Bassi
 * 5e4d96a - Added dependancy for sonargube test coverage feature - Amandeep Bassi
 * cd5ed44 - Added dependancy for sonargube test coverage feature - Amandeep Bassi
   bb2489a - Merge pull request #10 from AmanBassi/selenium - AmanBassi
 * 5053418 - Added chrome driver config so it does not open a new window while testing - Amandeep Bassi (origin/selenium, selenium)
 * 5db0aa9 - Created selenium test case for updating a task - Amandeep Bassi
 * 65bcebb - Created selenium test case for reading vehicle tasks in the system - Amandeep Bassi
 * Gecbddb - Created selenium test case for creating a vehicle task in the system - Amandeep Bassi
 * eedaad6 - Created new file for web testing vehicle tasks - Amandeep Bassi
 * 5b27e72 - Created selenium test case for deleting a vehicle in the system - Amandeep Bassi
 * dbd3cdd - Created selenium test case for updating a row - Amandeep Bassi
 * 48ef3ca - Fixed Bug where on update changes will only show on page refresh - Amandeep Bassi
 * 2702125 - Updated create vehicle test case to check details in whole table row - Amandeep Bassi
 * c549138 - Added selenium to the project and created a test case for creating a vehicle - Amandeep Bassi
   f6c38d7 - Merge pull request #9 from AmanBassi/testing - AmanBassi
 * 2abd9bd - Created and implemented task dto test cases - Amandeep Bassi (origin/testing, testing)
 * 1d53db9 - Create test cases for to string methods to improve coverage - Amandeep Bassi
 * 999d78e - Created and implemented test cases for vehicle dto - Amandeep Bassi
 * 816fa86 - Created and implemented unit test for the vehicle task class - Amandeep Bassi
 * 486ec82 - Created and implemented test cases for the vehicle domain - Amandeep Bassi
 * 5d5e329 - Created and implemented test cases for getting single object by id - Amandeep Bassi
 * 43bca49 - Create test case for task delete service - Amandeep Bassi
 * c514899 - Create test case for task update service - Amandeep Bassi
 * fcf5b9e - Implemented get all tasks test - Amandeep Bassi
 * dd5ec57 - Removed some unused imports - Amandeep Bassi
 * 60402ba - Implemented test cases for all methods in the vehicle task service class - Amandeep Bassi
 * 6b957b2 - Create JUnit test file for vehicle task service and tested the create method - Amandeep Bassi
 * 261cb2a - Create a construtor without the id to help testing - Amandeep Bassi
  3c4b090 - Merge pull request #8 from AmanBassi/task-delete - AmanBassi
 * 9f4d7e2 - Created delete buttons for vehicle tasks - Amandeep Bassi (origin/task-delete, task-delete)
 * ae9f286 - Created and implemented task delete in controller and service - Amandeep Bassi
   32d77ff - Merge pull request #7 from AmanBassi/task-update - AmanBassi
 * a5349e2 - Created web form to update task details - Amandeep Bassi (origin/task-update, task-update)
 * f2cdf64 - Created controller and service methods for the task update feature - Amandeep Bassi
   29f2dbb - Merge pull request #6 from AmanBassi/task-read - AmanBassi
1 * 0195aa9 - Created table on task page to show all tasks one vehicle has - Amandeep Bassi (origin/task-read, task-read)
```

348282e - Merge pull request #16 from AmanBassi/dev - AmanBassi (HEAD -> main, origin/main)

Testing

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
▼ I HobbyProject Market Market	95.8 %	4,530	199	4,729
✓	86.6 %	1,249	194	1,443
com.qa.hobbyproject.domain	84.9 %	549	98	647
> 🤳 Task.java	83.1 %	250	51	301
> J Vehicle.java	86.4 %	299	47	346
com.qa.hobbyproject.dto	81.8 %	408	91	499
> J VehicleDTO.java	82.0 %	233	51	284
> J TaskDTO.java	81.4 %	175	40	215
com.qa.hobbyproject	37.5 %	3	5	8
J HobbyProjectApplication.java	37.5 %	3	5	8
com.qa.hobbyproject.config	100.0 %	7	0	7
> J AppConfig.java	100.0 %	7	0	7
com.qa.hobbyproject.controller	1 100.0 %	59	0	59
> J TaskController.java	100.0 %	28	0	28
> J VehicleController.java	100.0 %	31	0	31
com.qa.hobbyproject.service	100.0 %	223	0	223
> J TaskService.java	100.0 %	105	0	105
> J VehicleService.java	100.0 %	118	0	118
> 📂 src/test/java	99.8 %	3,281	5	3,286

Mockito

```
@Test
void testGetVehicleById() {
    Optional<Phicle optionalVehicle = Optional.ofNullable(new Vehicle(1L, "FE65 PKK", "VW", "Golf", "Black", 220));
    Vehicle foundVehicle = new Vehicle(1L, "FE65 PKK", "VW", "Golf", "Black", 220);
    VehicleDTO vehicleDTO = new VehicleDTO(1L, "FE65 PKK", "VW", "Golf", "Black", 220);
    // GIVEN
    Long id = 1L;
    // WHEN
    Mockito.when(this.repository.findById(id)).thenReturn(optionalVehicle);
    Mockito.when(this.mapper.map(foundVehicle, VehicleDTO.class)).thenReturn(vehicleDTO);
    // THEN
    assertThat(this.service.getVehicleById(1L)).isEqualTo(vehicleDTO);
}</pre>
```

```
void testGetVehicleById() throws Exception {
    Vehicle vehicle = new Vehicle(1L, "PB08 BSB", "Porsche", "Macan", "Blue", 258);
    String vehicleAsJSON = this.mapper.writeValueAsString(vehicle);

    RequestBuilder mockRequest = get("/vehicle/getById/1");

    ResultMatcher checkStatus = status().isOk();

    ResultMatcher checkBody = content().json(vehicleAsJSON);

    this.mvc.perform(mockRequest).andExpect(checkStatus).andExpect(checkBody);
}
```

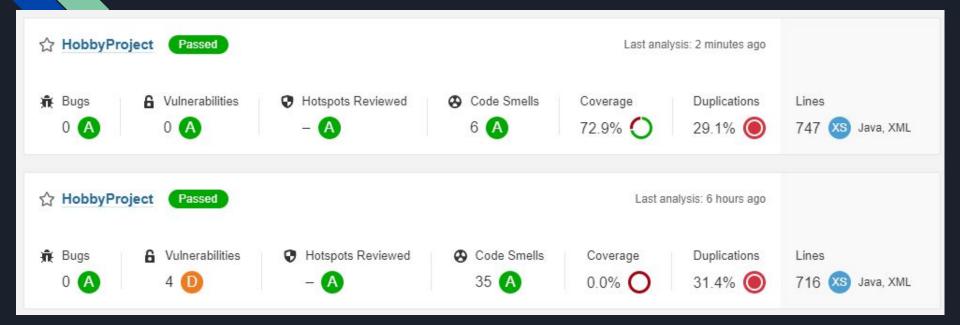
Selenium

```
@SpringBootTest(webEnvironment = WebEnvironment.RANDOM PORT)
@Sql(scripts = { "classpath:vehicle-schema.sql", "classpath:vehicle-data.sql" }, executionPhase = ExecutionPhase.BEFORE_TEST_METHOD)
@ActiveProfiles("test")
class VehiclePageTest {
    private static WebDriver driver;
   @LocalServerPort
   int serverPort;
   @BeforeAll
    static void setUpBeforeClass() throws Exception {
        System.setProperty("webdriver.chrome.driver", "src/test/resources/chromedriver.exe");
        ChromeOptions config = new ChromeOptions().setHeadless(true);
        driver = new ChromeDriver(config);
   @AfterAll
   static void tearDownAfterClass() throws Exception {
        driver.quit();
   @BeforeEach
   void setUp() throws Exception {
        driver.get("http://localhost:" + serverPort + "/");
       assertEquals("Hobby Project", driver.getTitle());
   @AfterEach
   void tearDown() throws Exception {
```

Selenium

```
@Test
void testUpdate() {
   driver.findElement(By.linkText("Enter system")).click();
   new WebDriverWait(driver, 3).until(ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div/main/div[1]/table/tbody/tr")))
   driver.findElement(By.cssSelector(".btn-primary:nth-child(1)")).click();
   String originalRow = driver.findElement(By.xpath("/html/body/div/main/div[1]/table/tbody/tr")).getText();
   new WebDriverWait(driver, 3).until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//*[@id=\"updateVehicleModal\"]")));
   driver.findElement(By.id("updateVehicleModal")).click();
   driver.findElement(By.id("updateRegistration")).clear();
   driver.findElement(By.id("updateRegistration")).sendKeys("BP14 NRE");
   driver.findElement(By.id("updateMake")).clear();
   driver.findElement(By.id("updateMake")).sendKeys("Suzuki");
   driver.findElement(By.id("updateModel")).clear();
   driver.findElement(By.id("updateModel")).sendKeys("Swift");
   driver.findElement(By.id("updateColour")).clear();
   driver.findElement(By.id("updateColour")).sendKeys("Grey");
   driver.findElement(By.id("updateHorsePower")).clear();
   driver.findElement(By.id("updateHorsePower")).sendKeys("90");
   driver.findElement(By.id("updateVehicleButton")).click();
   new WebDriverWait(driver, 3).until(ExpectedConditions.invisibilityOfElementWithText(By.xpath("/html/body/div/main/div[1]/table/tbody/tr"
   assertEquals("1", driver.findElement(By.xpath("/html/body/div/main/div[1]/table/tbody/tr/td[2]")).getText());
   assertEquals("BP14 NRE", driver.findElement(By.xpath("/html/body/div/main/div[1]/table/tbody/tr/td[3]")).getText());
   assertEquals("Suzuki", driver.findElement(By.xpath("/html/body/div/main/div[1]/table/tbody/tr/td[4]")).getText());
   assertEquals("Swift", driver.findElement(By.xpath("/html/body/div/main/div[1]/table/tbody/tr/td[5]")).getText());
   assertEquals("Grey", driver.findElement(By.xpath("/html/body/div/main/div[1]/table/tbody/tr/td[6]")).getText());
   assertEquals("90", driver.findElement(By.xpath("/html/body/div/main/div[1]/table/tbody/tr/td[7]")).getText());
```

Sonarqube



Demonstration

Sprint Retrospective

The Good

- Achieved MVP
- Codebase completed on time after falling behind schedule early in the project
- Learned new skills (Selenium, Postman, Axios JS)
- Added new status feature after I thought the project was completed

The Bad

- Should have set up the project structure
 in a separate branch then added features
- Implemented cascade delete on the
 Vehicle page
- Used Lombok to generate code as this would have improved the Sonarqube code smells

Questions