Assignment - Task breakdown, implementation and Test

SENG301 Software Engineering II

Patricia de Andrade   Marina Filipovic   Fabian Gilson

20th March 2018

Assignment objectives

This assignment is meant to help students mastering the various software engineering and coding concepts taught during the first term of this course. In detail, by completing this assignment, the students will be able to

- break down a user story into tasks and give estimations of those tasks
- write and translate acceptance criteria for stories
- implement the tasks in a professional way with proper documentation and tests
- deal with basic database interactions

To this end, students will continue to work on the WoF case study used throughout the first term. For the sake of completeness, we include the description, domain models and use case diagram here.

Practical details

The students are required to work **individually** and submit their report and implementation files in one **zip** archive of the form **firstname_lastname-seng301.zip** through **Learn** by Sunday **1**st **April 2018** at **11.59PM** at the latest.

This assignment is intended to be feasible during labs 5 and 6 and students may use both sessions to ask individual questions to the tutors. We bring to the attention of the students that any obvious or proven cheating will result in a void grading for all involved parties. However, you are encouraged to discuss with **(not copy to)** each other in case you have questions prior to ask to a tutor.

The students may ask as many questions as they like to the Tutors during the labs, but no question will be answered outside these hours. During the labs, tutors may take a look at your tasks, additional acceptance criteria, etc., for validation purposes only upon request.

1 Reminder about the WoF case study

Important note: *the description has been shortened to concentrate on the relevant details for this assignment. On the other hand, details has been added regarding Vehicle properties.*

You are required to build a new system to help *Vehicle Testing New Zealand* (VTNZ) inspectors dealing with their day to day tasks. Most privately owned vehicles need a *Warrant of Fitness* (WoF) that is performed regularly. During a WoF inspection, many verifications must be done in order to ensure the vehicle is appropriately maintained and therefore being driven safely on Kiwi roads.

Those tests are either manual or semi-automated and may be conducted on different types of vehicles, *i.e* passenger cars (type MA), passenger vehicles (MB), off-road vehicles (MC), trailers (T) or any other special types of vehicle (O). Inspectors are solely responsible to encode the results of manual tests, where appropriate testing systems take care of the automated ones. Vehicles are identified by their plate number and have a model, a make and a manufacture date. All vehicles except trailers (type T) also have a fuel type being either *diesel, petrol, electric, gas or other*.

When going to a WoF centre, you must have a proof of address as well as your last WoF documents with the previous inspection date and status. When you arrive at the centre, you will pass through the verifications with an inspector that will be responsible to either encode the results or validate the output of automated tests. Prior the tests are run, the inspector must manually encode many details such as the VIN number, make, model, fuel type, odometer reading, first registration year in New Zealand, WoF expiry date, and so forth. As this encoding task is time-consuming and may be easily done beforehand, we would like to let people register on a website with all these details and make appointments in their preferred centre.

1.1 Conceptual diagram

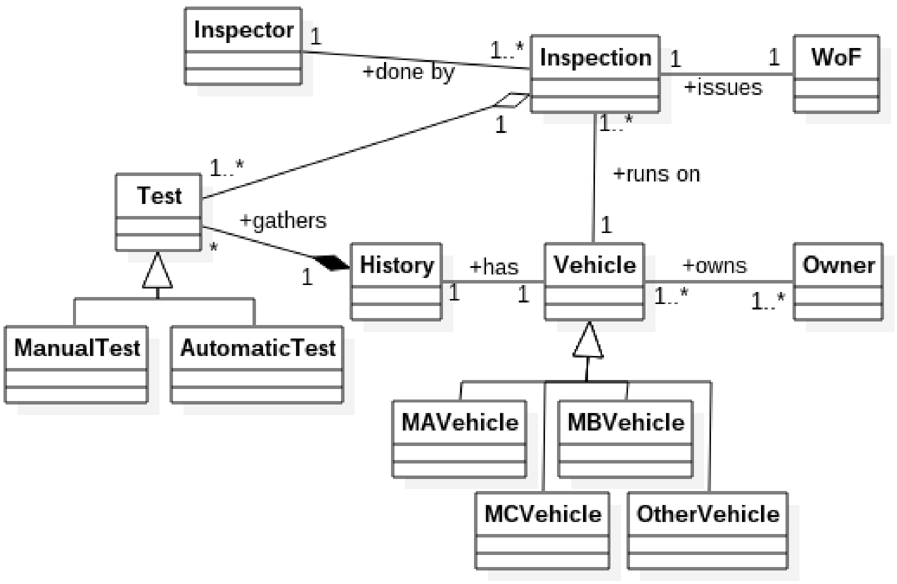

Figure 1: UML conceptual class diagram of WoF inspection

1.2 Use Case diagram of OnlineSystem

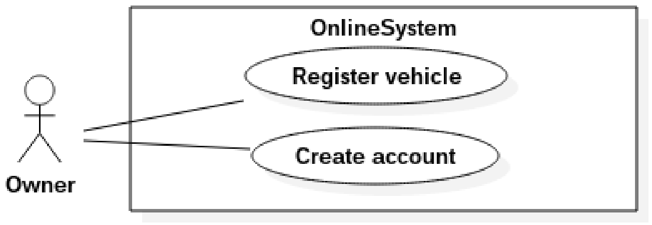

Figure 2: Simplified UML Use Case diagram for the WoF inspection system

2 Your tasks

From the two user stories below, you are required to

1. detail the breakdown in tasks with their estimations,
2. implement them,
3. add cucumber code for the acceptance criteria, and
4. write unit test classes.

The task breakdown referring to these stories must be documented in a separate report, where

- for both stories, you give the original estimate of the story,
- you add the acceptance criteria for the second story,
- you detail the task breakdown of each of them, with their descriptions and estimations, and
- you log the effective completion time for all tasks.

The final code is composed by

- the fully **documented** and commented source code in Java,
- accompanied by cucumber code regarding the acceptance criteria,
- a bunch of unit tests for your methods,
- a text file with the sql commands needed to create your *sqlite* database, and
- the generated javadoc is not mandatory, inline javadoc documentation is sufficient.

A small README file explaining the content of your archive file (its detailed structure) as well as how we can start your program.

Assumptions

1. You may develop either a simple command-line interface or *hardcode* a small running example in you main class to demonstrate your code.
2. You are not required to write a real web-based system, a plain Java program is sufficient.
3. Your main class must be easily identifiable (for example by giving a small explanation on how to run your code from your report).
4. Your *sqlite* file may contain test data which your test code relies on. In such case, appropriate documentation must be provided.

2.1 User story 1: Owner subscription

As an owner, I want to create an account on the system with my email address so that I can add and retrieve my vehicles later on.

Acceptance criteria

- An owner must submit his firstname(s), lastname, email address and password all at once.
- The submitted details must be stored into the database.
- The email address is used as the owner id on the system, as well as his/her login.
- When the registration of the owner is done, a confirmation message is displayed on the screen.
- After the registration of an email address, no new registration is allowed with the same email address.
- If an owner tries to register with an existing email, an error message saying that email is already in use must be displayed on the screen.

2.2 User story 2: Vehicle subscription

As an owner, I want to register my vehicle so that I can speed up my future WoF inspections.

Acceptance criteria

You have to provide the acceptance criteria for this story.