16.17

Do you feel the order is appropriate? *No, they are in a random order. You want the simplest features to be first, so that it allows you to incrementally build the application up.*

Is the level of complexity of each too high, too low, or just right? *I would say its just right, as its at an early stage still, and that a lot of the classes have a similar structure, like if its either a taxi or shuttle.*

Are there any steps missing? *Probably but can’t think of any.*

Revise the list as you see fit, to suit your own view of the project:

■ Enable a single passenger to be picked up and taken to her destination by a single taxi.

■ Enable a single passenger to be picked up and taken to his destination by a single

shuttle.

■ Ensure that details are recorded of passengers for whom there is no free vehicle.

■ Provide sufficient taxis to enable multiple independent passengers to be picked up and

taken to their destinations concurrently.

■ Enable a single shuttle to pick up multiple passengers and carry them concurrently to

their destinations.

■ Ensure that taxis and shuttles are able to operate concurrently.

■ Provide a GUI to display the activities of all active vehicles and passengers within the

simulation.

■ Provide all remaining functionality, including full statistical data.

16.18

Are the completion criteria (tests on completion) for each

stage sufficiently obvious? If so, document some tests for each:

■ Enable a single passenger to be picked up and taken to her destination by a single taxi.

■ Enable a single passenger to be picked up and taken to his destination by a single

shuttle.

= single passenger is picked up and completes their final destination

■ Ensure that details are recorded of passengers for whom there is no free vehicle.

= Keeps track of passengers until they are assigned a taxi successfully

■ Provide sufficient taxis to enable multiple independent passengers to be picked up and

taken to their destinations concurrently.

= 2 or more passengers are picked up and taken to their final destination successfully

■ Provide a GUI to display the activities of all active vehicles and passengers within the

simulation.

= GUI has been created and all button selections work as expected

16.19

I have read through the act() method.

First it created a Location object and names it target, which then calls the getTargetLocation() method, that has the requested destination stored. It then creates an if loop that checks that the target isn’t null, basically checking that there’s a required destination. It will then create a variable ‘next’, which stores the destination. If the variable ‘next’ matches the target/destination, it takes the passenger to the destination, and then lets the passenger off. If there was no passenger, it picks up the next one. Then if there’s no passenger to pick up, then it will idle and increment the count until one is available.

16.20

Yes, they should keep seprate lists. They should be kept separate as this would help prevent scheduling issues, as you could keep a list for all available vehicles, and another for all occupied vehicles.

16.21

No, it doesn’t support this. All vehicles are held in a LinkedList, which would be impossible to create a unique identifier for each vehicle, or even if its either a taxi or shuttle. It would need reworked to allow the keeping track fo which customers have used which vehicle.

16.22

Review the way in which *vehicle:passenger* associations are stored in the assignments’ map in **TaxiCompany**. Can you see any weaknesses in this approach? = Yes, only one passenger can be assigned too one vehicle. Also, there’s no way of keeping track to the unique journeys.

Does it support more than one passenger being picked up from the same location? = No, only one registers passenger can be picked up, but they could 1 or more people with them, but no way of keeping track of them.

Could a vehicle ever need to have multiple associations recorded for it? = Yes, a shuttle would as it requires the capability to pick up more than one passenger from more than one destination. However, its not required for a taxi.

16.23

If you see any problems with the current way in which *vehicle:passenger* associations are stored, would creating a unique identification for each association help—say a “booking number”? = Yes, one way of achieving this would be to create a map of maps, that would allow you to store a unique identifier for each association.

16.24

Review the tests implemented in the test classes of *taxicompany-stage-one*. Should it be possible to use these as regression tests during the next stages or would they require substantial changes?

Yes, we can use the existing test and they wont require any rework at this stage. However, once we implement ned methods and classes then we would require to write further tests.

16.25

Implement additional tests and further test classes that you feel are necessary to increase your level of confidence in the current implementation.Fix any errors you discover in the process of doing this.

This is only required if we implement a new collection for storing the unique identifier, like booking id.