Individual Contribution and Reflective Report

Student Name: Lennox Kpakiwa
SID: 4869373

Assignment: Part 1

Activity Title: sprint labs: 1, 2

Contribution and Reflection:
Your report should include the following items:

- Evidence of your work produced for the above activity with an outline
- Research and reflection on your contribution to the work in relation to skills and knowledge acquired (in 500 words).
Sprint 1:

For sprint 1, my group along with myself had to create; A use case diagram which has to be showing the relationship between each use case, and also we had to create a use case and use case description for each process within the system.

Below is the UML diagram that my group and I created. The UML diagram shows each process within our system and also shows all the actors who interact with the system, for example the booking operator and credit agency. My individual contribution to the task was the use case description. I was tasked with having to look at each process and describing them, for example; which actors interact with this process, the pre-condition etc.

Below is a screenshot of the ‘Get member ID’ process which I had to create a use case description for.
As you can see I described the actor who was interacting with the process, I described the trigger, post conditions, preconditions, I described the full flow of how the system should flow if everything goes right etc.

<table>
<thead>
<tr>
<th>Actors</th>
<th>Booking operator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Descriptions</td>
<td>The get member ID process is necessary in order for a booking to be made. The booking needs to have all the information of the user, this information is made easily accessible by simply searching the member ID.</td>
</tr>
<tr>
<td>Triggers</td>
<td>Get member ID is triggered when a user wishes to make a booking, and the booking operator needs to access their information</td>
</tr>
</tbody>
</table>
| Preconditions | 1. Customer has chosen to book a seat 
2. Booking operator needs to find customer's information |
| Post conditions | 1. Booking operator gets access to member's information. 
2. Booking is made |
| Normal Flow | 1. Customer enters site 
2. Customer decides they want to book seat 
3. Booking operator begins process of making booking 
4. Booking operator requests member ID 
5. Member ID is used to search information 
6. Booking is selected 
7. Customer gives payment details 
8. Payment is verified 
9. Trip booking is completed 
10. Seat and booking is verified |
| Alternative Flows | [Alternative Flow 1 – Not in Network] |
| Error Conditions | 1. Member ID is not given 
2. Message sent to booking operator 
3. Booking operator requests full information 
4. Booking operator must have customer ID sent to system 
5. Use case resumes on step 3.6 |

Sprint 2:
For the 2\textsuperscript{nd} sprint, my group and I had to create a UML class diagram of our system. The Class diagram has to show all the classes along with their responsibilities, their associations such as inheritance, aggregation, and composition. Once we had done that we also had to make a list of all the non-functional requirements for the system.

Below is the finished UML Class diagram which myself and my group created. In order to save time, we decided that we should each do one class for the diagram with its associations and attributes etc, and then we would combine them all together and finish it off. I decided to work on the journey class.
On the left is a screenshot of the classes I created. As you can see I made a parent class which is ‘Journey’ and then I had to make sub classes also, these classes were ‘Domestic Booking’ and ‘Europe Booking’. Creating the journey class I had to add the operations and attributes to it, then class them as either private or public. In the UML use case diagram, there were two include processes which connected to the journey, these were home trips and european trips, so I had to make thes the sub classes of the Journey class and give them a direct association to their parent class.
**Reflection on my individual work**

Doing this assignment I realised just how much I thoroughly enjoy UML use case diagrams, descriptions, and class diagrams. This assignment allowed me to test myself at creating work which showed the best of my ability, this made it enjoyable which allowed me to produce high quality work. The part of the assignment which was difficult, was working with my group. Working individually is easy as you have no one else to depend on, but working with a group was challenging as we had to arrange meeting outside of uni, everyone wanted their own ideas and opinions to be accepted which just made the whole assignment draining. Overall I am happy with how assignment one went as it allowed me to experience and improve upon my team working skills, but also improve and challenge myself in relation to my UML work.
Individual Contribution and Reflective Report

Student Name: Lennox Kpakiwa

SID: 4869373

Assignment: Part 2

Activity Title: sprint labs: 4, 6, 7

Contribution and Reflection:
Your report should include the following items:
- Evidence of your work produced for the above activity with an outline
- Research and reflection on your contribution to the work in relation to skills and knowledge acquired (in 500 words).
Sprint 4:

For the 4\textsuperscript{th} sprint, my group and I had to create another UML class diagram, but this class diagram needed to have; a layered architecture, constructor methods, additional attributes and operations, Data types for attributes and finally Operation signatures for operations. My group and I discussed who would do what within the sprint, since I felt I was most confident with layered architecture; I put myself forward to do that part of the sprint.

Below is a screenshot of one of my class diagrams which I made using the 3 layered architecture method. For the diagram I first identified and separated the layers, I first identified the user interface logic layer, then the application logic layer and lastly the Domain/Data logic layer. Next I had to start deciding what goes into each section. I put the actor, who interacts with the system in the UI logic layer, then I put the actual booking class in the application logic layer, then last of all I put all the things which the booking class controls into the Domain/Data logic layer. Last thing I had to do was add the associations and attributes and operations.
Sprint 6:

Task 6 was all about automatic code generation and Pseudocode. For this week’s sprint my group and I had to produce Pseudocode to process customers booking requests based on the details below.

Below is the Pseudocode my group and I created for the task:

PROGRAM Booking Request

Take details of journey.

Take number of seats required

Take date of required journey.

Check system for journeys that fit requirements.

If Journey = Available

Make booking

System Requests card details

If transaction is ok

Accept transaction
Else reject transaction

If customer wants to look up information
System finds departure information

If customer wants to cancel
Check membership
If customer=member
Cancel = true
else
Cancel = false

If awaymatch = true
If customer = member
check how many trips they have been on in last year.
Additional fare required.
else if customer = nonmember
awaymatch = false

If user=operations manager
new services can be made
services can be amended
else
Services cannot be edited.
Sprint 7:

For the final sprint, sprint 7, my group and I had to create test scripts which would help us to evaluate whether or not our system was a success. The tests scripts would be used as a guideline for the tester which they will use to mark if the tested sections are a good standard or not. As a group we decided we would do our tests before the handover section of the project, so basically we would test the product before handing it over to the client. I believed for this task that I had the most experience with projects and project managing and testing so I put myself forward to make the test scripts.

Below is a print screen of the test script I made.

<table>
<thead>
<tr>
<th>Subject</th>
<th>Test Name</th>
<th>Description</th>
<th>Step Name</th>
<th>Step Description</th>
<th>Expected Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>LPEC Reservations System</td>
<td>Customer Identification</td>
<td>Customer ID is successfully recognised and their details are made available</td>
<td>Step 1</td>
<td>Enter site</td>
<td>Site agent</td>
</tr>
<tr>
<td>LPEC Reservations System</td>
<td>Customer Identification</td>
<td></td>
<td>Step 2</td>
<td>Select booking</td>
<td>Allow customers booking and be redirected to login page</td>
</tr>
<tr>
<td>LPEC Reservations System</td>
<td>Customer Identification</td>
<td></td>
<td>Step 3</td>
<td>Enter Member ID</td>
<td>Allow customer to enter in their membership ID</td>
</tr>
<tr>
<td>LPEC Reservations System</td>
<td>Customer Identification</td>
<td></td>
<td>Step 4</td>
<td>View member details</td>
<td>All of the member details are displayed</td>
</tr>
<tr>
<td>LPEC Reservations System</td>
<td>Customer Registration</td>
<td>Customer who are non-members can successfully and easily create an account</td>
<td>Step 1</td>
<td>Click on register</td>
<td>User should be directed to account registration page</td>
</tr>
<tr>
<td>LPEC Reservations System</td>
<td>Customer Registration</td>
<td>Customer who are non-members can successfully and easily create an account</td>
<td>Step 2</td>
<td>Type in required details</td>
<td>User should be directed to account registration page</td>
</tr>
<tr>
<td>LPEC Reservations System</td>
<td>Customer Registration</td>
<td>Customer who are non-members can successfully and easily create an account</td>
<td>Step 3</td>
<td>Get member ID</td>
<td>User is given a unique membership ID</td>
</tr>
<tr>
<td>LPEC Reservations System</td>
<td>Processing a customer's booking request</td>
<td>Customer booking request should be successfully processed and confirmed</td>
<td>Step 1</td>
<td>Complete booking</td>
<td>User is redirected to payment page</td>
</tr>
<tr>
<td>LPEC Reservations System</td>
<td>Processing a customer's booking request</td>
<td>Customer booking request should be successfully processed and confirmed</td>
<td>Step 2</td>
<td>Enter payment details</td>
<td>Payment should be successfully verified and user booking is confirmed</td>
</tr>
<tr>
<td>LPEC Reservations System</td>
<td>Processing a customer's booking request</td>
<td>Customer booking request should be successfully processed and confirmed</td>
<td>Step 3</td>
<td>Booking confirmed and details of booking shown</td>
<td>Booking confirmed and information of booking displayed</td>
</tr>
</tbody>
</table>
Reflections on my work

Doing this second assignment I got the opportunity to further expand on my system/software design skills and my UML skills as sprint 4 challenged me to remember what I had previously learnt and showed me how much I had improved since I first learnt software engineering. The task was really interesting and enjoyable, this made it enjoyable which allowed me to produce high quality work. Just like in the first assignment the part of the assignment which was difficult, was working with my group. Working individually is easy as you have no one else to depend on, but working with a group was challenging as we had to arrange meeting outside of uni, everyone wanted their own ideas and opinions to be accepted which just made the whole assignment draining. I also found that I need to improve and do further research on pseudocode as found that I struggled a little bit with sprint 6. Overall I am happy with how assignment one went as it allowed me to experience and improve upon my team working skills, and I learnt about overcoming tasks which I find complicated and pushing myself to improve and make my weakest skills my strongest.