WILFRID LAURIER UNIVERSITY

LAURIER

Inspiring Lives.

CP 317 – Software Engineering

Spring 2024

Group 19

Assignment #3

Member 1 Name:	Christopher Matheson	ID:169019620
Member 2 Name:	Shayan Dhillon	ID:169026351
Member 3 Name:	Aidan Lin	ID:169022628
Member 4 Name:	Zakariya Boukhezza	ID:169024296
Member 5 Name:	Amuthan Vigneswaran	ID:169043624

Submission Date:

All the listed members have contributed, read, and approved this submission.

Member 1 Name: Christopher Matheson Signature: Christopher Matheson
Member 2 Name: Zakariya Boukhezza Signature:
Member 3 Name: Aidan Lin Signature:
Member 4 Name: Amuthan Vigneswaran Signature:
Member 5 Name: Shayan Dhillon Signature:

Please provide your answer starting form this page

1. Determine the cohesion (Explain why this type of cohesion exists) of the following modules: (1.1)
    a. EditProfitAndTaxRecord
        - This module likely falls under functional cohesion, since it only has one singular purpose/mission; editing the profit and tax record together- while also maintaining the flexibility for reusability, making it a reliant function to call upon as many times needed.
    b. EditProfitRecordAndTaxRecord
        - This module likely falls under logical cohesion, as the profile and tax record are logically related. The code for both operations will be intertwined, which falls under one of the negatives when it comes to logical cohesion- demonstrating that it's one of the more avoidable degrees of cohesion.
    c. ReadDeliveryRecordAndCheckSalaryPayments
        - This module likely falls under procedural cohesion, because the operations performed in this module have no relation or correlation with each other whatsoever, but it performs a series of operations related procedurally to one another.
    d. ComputeTheOptimalCostUsingAksen'sAlgorithm
        - This module likely falls under functional cohesion for the same reasoning as part a) did- it only serves one specific and narrowed down purpose, in this case being computing the optimal cost using Aksen's algorithm.
    e. MeasureVaporPressureAndSoundAlarmIfNecessary
        - This module likely falls under communicational cohesion, seeing as how the measurement of the vapor pressure is essential towards sounding the alarm if necessary, demonstrating the series of operations executed while being related to each other procedurally.

2. What is the influence of cohesion on maintenance? (1.2)

Cohesion is the degree of interaction within a module. This could range from a module performing multiple unrelated operations to a module performing exactly only one operation.

A module performing multiple tasks has zero reusability. Said module will only be used in the specific instance it was made for. If the entire project is made with poorly designed modules like this, then the number of modules needed will rise which already increases the time for maintenance.

When the opposite happens, and a module only performs a single task, it can be reused multiple times in the project. If other modules aren't harshly dependent on this one, maintenance could be performed once on the single module to improve the entire project.

3. Your client has stipulated that open-source software is to be used. Is this a functional or nonfunctional requirement? How early in the life-cycle model can this requirement be handled? Explain your answer.

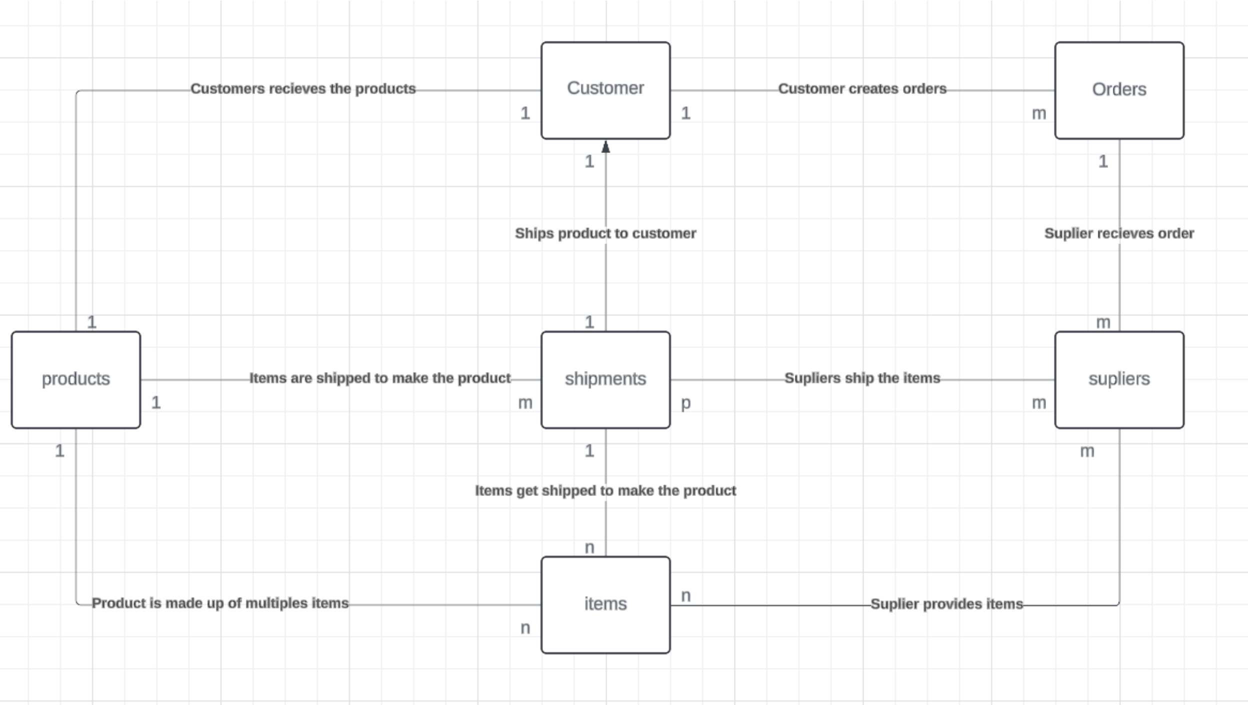
This is a non-functional requirement. Non-functional requirements are defined by constraints that must be satisfied during development. Required open-source software to be used does not describe what the system should use but what constraint must be satisfied.

This requirement can be handled as early as the analysis phase. Ensuring that the project aligns with the constraint of needing to use open-source software is critical to moving forward. This influences the decisions made during the design phase for the tools, technologies and frameworks to be used to complete the product.

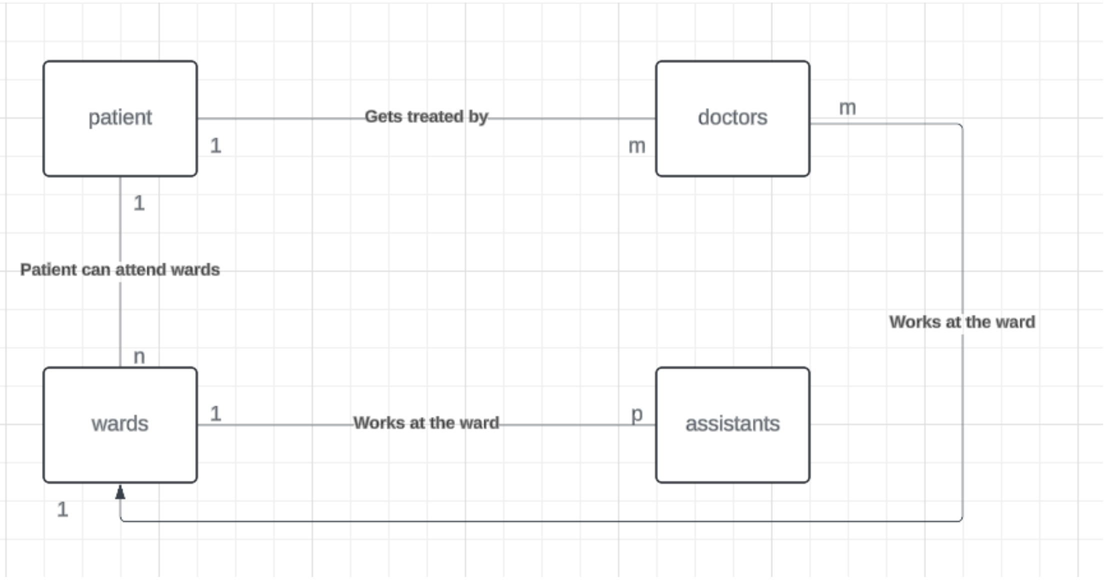
4. Develop an entity-relationship diagram for each of the following database systems: (3)

https://lucid.app/lucidchart/e74a6f43-133e-4161-9ab4-c3e766fd0d81/edit?viewport_loc=-1815%2C92%2C2219%2C1145%2C0_0&invitationId=inv_0de1cd3f-dccb-458b-984e-6bd8b0689952

a. Entities: Customers, orders, products, items, shipments, suppliers.
Customers submit orders for products. A product consists of many items. Different suppliers may supply
each item. A supplier may ship multiple items within a single shipment.


Customers recieves the products
Customer
Customer creates orders
Orders
1
1
m
1
1
Ships product to customer
Suplier recieves order
1
1
m
products
Items are shipped to make the product
shipments
Supliers ship the items
supliers
1
m
p
m
1
1
m
Items get shipped to make the product
n
Product is made up of multiples items
items
n
Suplier provides items
n

b. Entities: patients, doctors, wards, assistants.
Patients can be seen by different doctors and treated in other wards. Each ward has many assistants who can see the patients.


patient
Gets treated by
doctors
1
m
m
1
Patient can attend wards
n
Works at the ward
wards
1
Works at the ward
p
assistants
1