**Project description**

Each group has 4 deliverables: (**Phase2:** 50 grades)

* Total grades for the project should be **20 grades**. So, the grades will be scaled.
* The templates for your deliverables and your cover page are provided.
* Your deliverables will be submitted on **https://forms.gle/5tsks6cZURf1cwNUA**.
* This form is used to submit your different deliverables in course IS332 in the academic year (2022-2023)
* Please submit your file named T<Team number>-D<Deliverable number>.docx or .pdf. Example: T7-D2.docx
  + Do not add spaces in the filename.
  + Avoid zipping the file unless it is larger than 10 MB.
* Deliverable 2 **Firm Deadline is 26-11-2022**.

**Note that**

* Only one team member (**team leader**) can submit the project.
* Don’t submit multiple times.
* Cheating policy: Any copies will be graded zero.
* Clearly state your assumptions (only required ones).
* No handwritten reports and/or models are accepted [graded zero].
* You need to use any UML tool to depict your models and please mention the used tool(s) in your report.

[**Hint**: Review Chapters 4 - 5: domain class diagram & CRUD]

[**Tip**: Split the work among team members to help with time management]

**Project Details**

|  |  |
| --- | --- |
| Project number | *project 1* |
| Project title | *Mega store corporation* |
| Corresponding TA\LA | TA/Lamia Atef |
| Deliverable | *2* |

**Team Details**

|  |  |  |
| --- | --- | --- |
| **Student ID** | **Student name** | **Lab Group** |
| 20200366 | عمرو ابراهيم محمد ابراهيم سيد | S3/S4 |
| 20200645 | يوسف جلال الدين ناظم حيدر الطائي | S3/S4 |
| 20200367 | عمرو خالد احمد محمد طه | S3/S4 |
| 20200254 | شريف احمد محمد حسن عبده | S3/S4 |
| 20200239 | سيف الدين محمد خليل عبده | S3/S4 |
| 20201114 | عبد الرحمن سيد فؤاد سيد | S3/S4 |

# **Guidelines for Deliverable 2**

1. Apply the CRUD technique
2. Domain Class Diagram [Complete system]

* Domain Class Diagram based on the Noun technique.
* You need to follow the steps of the NOUN technique to identify the domain classes and document information using Tabular representations whenever possible. Please REFER to CHAPTER 4 in the book.
* You need to make use -as much as you can- of complex association types, i.e., generalization, aggregation, and composition.
* Your class diagram should also include multiplicities.
* Please specify the name of the associations in your model to increase readability.

1. State machine diagram (single object is enough). Please justify your selection.
2. Activity diagram(s) (at least 3 use cases)
3. System Sequence diagram(s) (for the same use cases in activity diagrams)

Please **clearly state** the name of the use case related to each model in steps 4 and 5.

**Notes:**

1. Updated use case Diagram with new use cases (Create account, Update profile details, View profile, deactivate account, Add product to cart, Add offer, Update offer details, View all offers, Remove offer, Add product, Update product details, View all products, Remove product, Check payment records)

**Assumptions:**

1. The different types of payment methods are: Cash, Credit card, Customer points.

2. A cart is assigned to a customer when an account is created.

3. An order must be placed before any payment procedure.

4. Products are assumed to be provided by a supplier, but the supplier is not necessarily connected to the system.

1. **CRUD technique:**

|  |  |  |
| --- | --- | --- |
| Domain Class | CRUD | Verified Use case |
| Customer | Create | Create account |
| Read | View profile |
| Update | Update profile details |
| Delete | Deactivate account |
| Cart | Create | Create account |
| Read | View cart |
| Update | Add product to cart |
| Delete | Deactivate account |
| Product | Create | Add product |
| Read | View profile |
| Update | View all products |
| Delete | Remove product |
| Offer | Create | Add offer |
| Read | View all offers |
| Update | Update offer details |
| Delete | Delete offer |
| Payment | Create | Checkout |
| Read | Check payment records |
| Update | None |
| Delete |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Use Case**  **VS**  **Domain Class** | **Customer** | **Cart** | **Product** | **Offer** | **Payment** | **Order** |
| **Create account** | C | C |  |  |  |  |
| **Update profile details** | U |  |  |  |  |  |
| **View profile** | R |  |  |  |  |  |
| **Deactivate account** | D |  |  |  |  |  |
| **Search products** |  |  | R |  |  |  |
| **Make purchase** |  |  |  |  |  | C |
| **Add product to cart** |  | U |  |  |  |  |
| **Browse products and offers** |  |  | R | R |  |  |
| **Request refund** |  |  | U |  |  | U |
| **Add offer** |  |  |  | C |  |  |
| **Update offer details** |  |  |  | U |  |  |
| **View all offer** |  |  |  | R |  |  |
| **Remove offer** |  |  |  | D |  |  |
| **Add product** |  |  | C |  |  |  |
| **Update product details** |  |  | U |  |  |  |
| **View all products** |  |  | R |  |  |  |
| **Remove product** |  |  | D |  |  |  |
| **Check payment records** |  |  |  |  | R |  |
| **Update customer preferences** | U | R |  |  |  | R |
| **Checkout** |  | U | U |  | C |  |
| **Package order** |  |  |  |  |  | U |
| **Package using normal bags** |  |  |  |  |  | U |
| **Package using special bags** |  |  |  |  |  | U |
| **Gain points** | U |  |  |  | R | R |

1. **Domain Class Model**

Diagram

Description automatically generated

1. **State Machine Model (Order object)**

Diagram

Description automatically generated

**Justification:**

We choose the order object because it’s a central object and it has the most states in the system, the other objects are static most of the time.

1. **Activity Diagrams** 
   1. **Checkout activity diagram**

Diagram

Description automatically generated

* 1. **Search Product Activity Diagram**

**Diagram

Description automatically generated**

* 1. **Add to cart Activity Diagram**

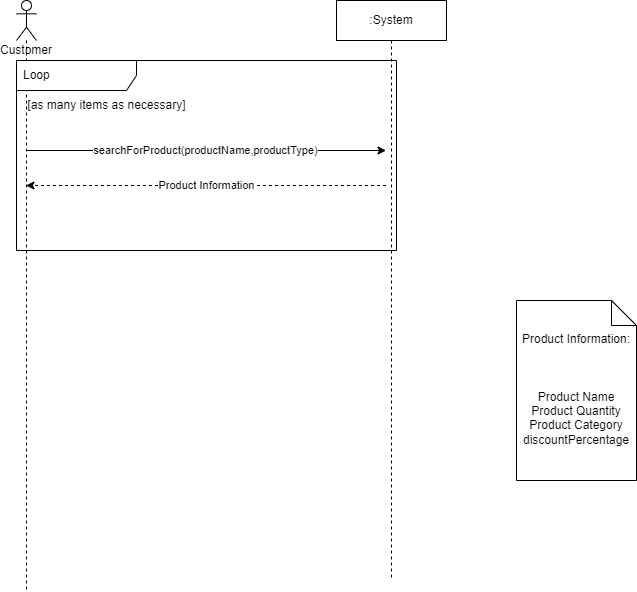
**Diagram

Description automatically generated**

1. **System Sequence Diagrams**
   1. **Checkout SSD**

**Diagram

Description automatically generated**

* 1. **Search Product SSD**
  2. **Add to Cart SSD**

**Diagram

Description automatically generated**