Free Second-Hand Clothing Distribution System

Process Modeling and Data Modeling /   
System Proposal / Analysis Phase   
(Homework No.3)

Project team: Team 02

Instructor: Dr. Araz Yusubov

Submitted in partial fulfillment of the requirements of the INFT 2303: Systems Analysis and Design course project

| GitHub repository | <https://github.com/ADA-SITE-INFT2303-2023-Spring/sys-dev-project-team-02> |
| --- | --- |
| Version date | Version information |
| 22.04.23 | Initial draft |
| 23.04.23 | Final draft |

| Other documents in the package | |
| --- | --- |
| Homework 1B | Updated version of the document according to this homework’s progress. |
| Homework 2 (Requirements) | Updated version of the document according to this homework’s progress. |
| Context Diagram | PNG and drawio file |
| Activity Diagram | PNG and drawio file |
| Level 0 DIagram | PNG and drawio file |
| Level 1 Diagram | PNG and drawio files |
| ERD | PNG and drawio file |
| Crud Matrix | PNG and xlsx file |

| Team member | Contribution to this homework (NOT the project) | Estimated % |
| --- | --- | --- |
| Atlas Hamzali | Intro, Definitions, Process Modeling, Level 0 diagram, Level 1 diagram, Data Modeling, ER diagram, CRUD matrix, References | 28.3% |
| Farid Mammadli | Intro, Definitions, Process Modeling, Level 0 diagram, Level 1 diagram, Data Modeling, ER diagram, CRUD matrix, References | 28.3% |
| Nargiz Heybatova | Intro, Definitions, Process Modeling, Level 0 diagram, Level 1 diagram, Data Modeling, ER diagram, CRUD matrix, References | 28.3% |
| Azer Shukurlu | Intro, Definitions, Process Modeling, References | 15% |

# **Table of Contents**

# **Introduction**

This is part of the System Proposal for a hypothetical project  ***Free Second-Hand Clothing Distribution System*** submitted for partial fulfillment of the requirements of the Systems Analysis and Design course in the School of Information Technologies and Engineering at ADA University, Baku, Azerbaijan.

A project called the Free Second-Hand Clothing Distribution System seeks to develop a platform for the effective and long-lasting distribution of used clothes to people in demand. Through a variety of modeling tools, such as process modeling, data modeling, CRUD matrix, and UML activity diagrams, this article offers an in-depth understanding of the system. This document's main objective is to give readers a thorough knowledge of the functionality, data flow, and workflow of the system. During the system's development and deployment, it will act as a guide for programmers, clients, and project managers. The many modeling approaches employed will be described in depth in the sections that follow, along with descriptions of each part.

### Changes on the Previous Homeworks:

**Changes on the HW01:**

1. There were changes on the Definitions.
2. There were changes in the Overall Description.
3. Several irrelevant sentences were deleted from the Special issues or constraints.
4. From a product perspective, irrelevant points were deleted.
5. Several Product Functions were deleted since we had some changes in HW02 and HW03.

**Changes on the HW02:**

1. For the *“Register User”* use case: Actor is changed from *“User”* to *“Donor”*. Description and other sections are also updated accordingly.
2. *“Donate Clothes”* use case: Actor is changed from *“User”* to *“Donor”*. Description and other sections are also updated accordingly.
3. *“Remove Donation*” Use case: Actor is changed from *“User, Admin”* to *“Donor”*. Description and other sections are also updated accordingly.
4. *“Search Clothes”* Use case: Actor is changed from *“User”* to *“Donee”.* Description and other sections are also updated accordingly.
5. *“View inventory”* Use case: Actor is changed from *“User”* to *“Donee”.* Description and other sections are also updated accordingly.
6. *“Request Existing Donation”* Use case: Actor and name of the use case changed.

->Actor: changed from *“User”* to *“Donee”.* Description and other sections are also updated accordingly.

->Name of the use case is changed from *“Request Existing Donation”* to *“Request Donation”*

1. *“Request Special Donation”* Use case is removed from the system
2. *“Provide Donation Feedback”* Use case: Actor is changed from *“User”* to *“Donee”*. Description and other sections are also updated accordingly.

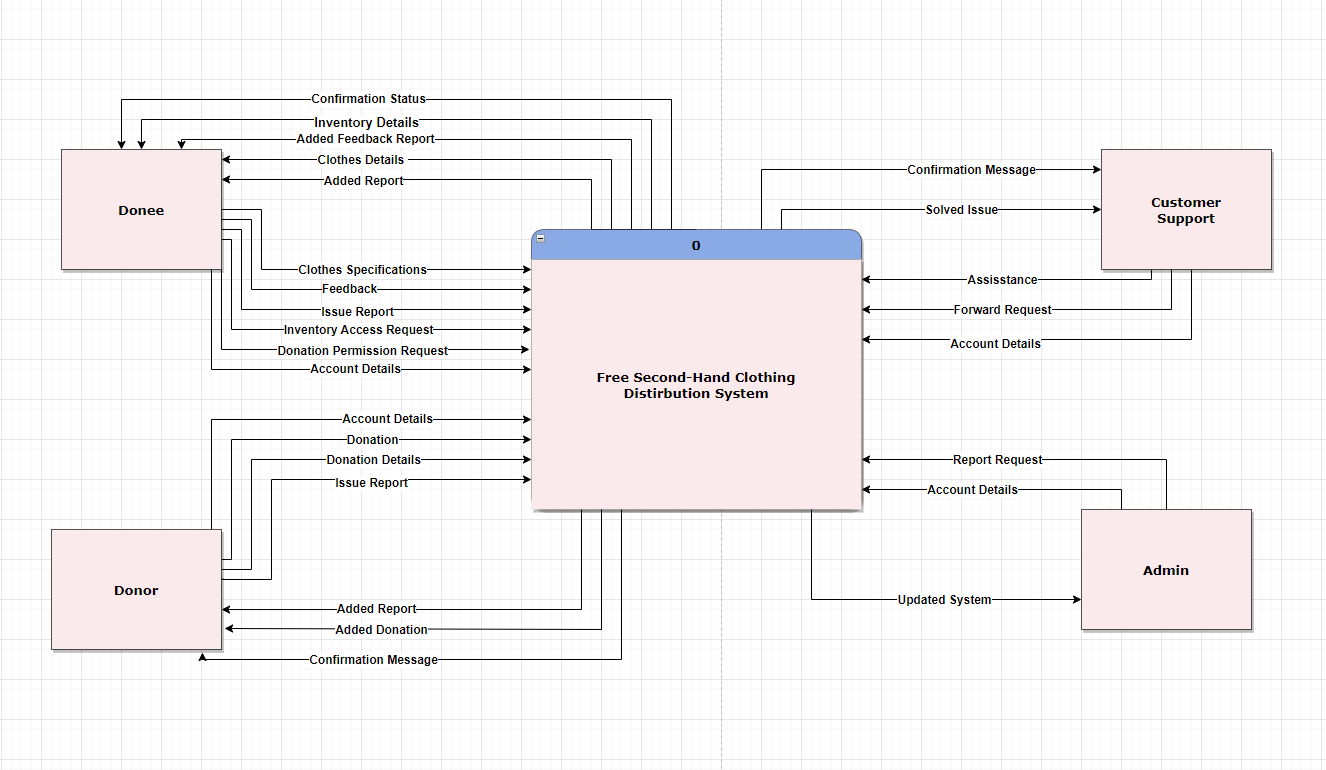
Name of the Use case is changed from *“Provide Donation Feedback”* to *“Provide Donation Feedback Report”*

1. *“Raise an Issue”* Use case: Actor is changed from *“User”* to *“Donee, Donor”.* Description and other sections are also updated accordingly.
2. *“Analyze Feedback”* Use case is removed from the system
3. *“Forward Feedback”* Use case is renamed to *“Forward Report”*
4. *“Work on Feedback”* Use case is renamed to *“Work on Report”*

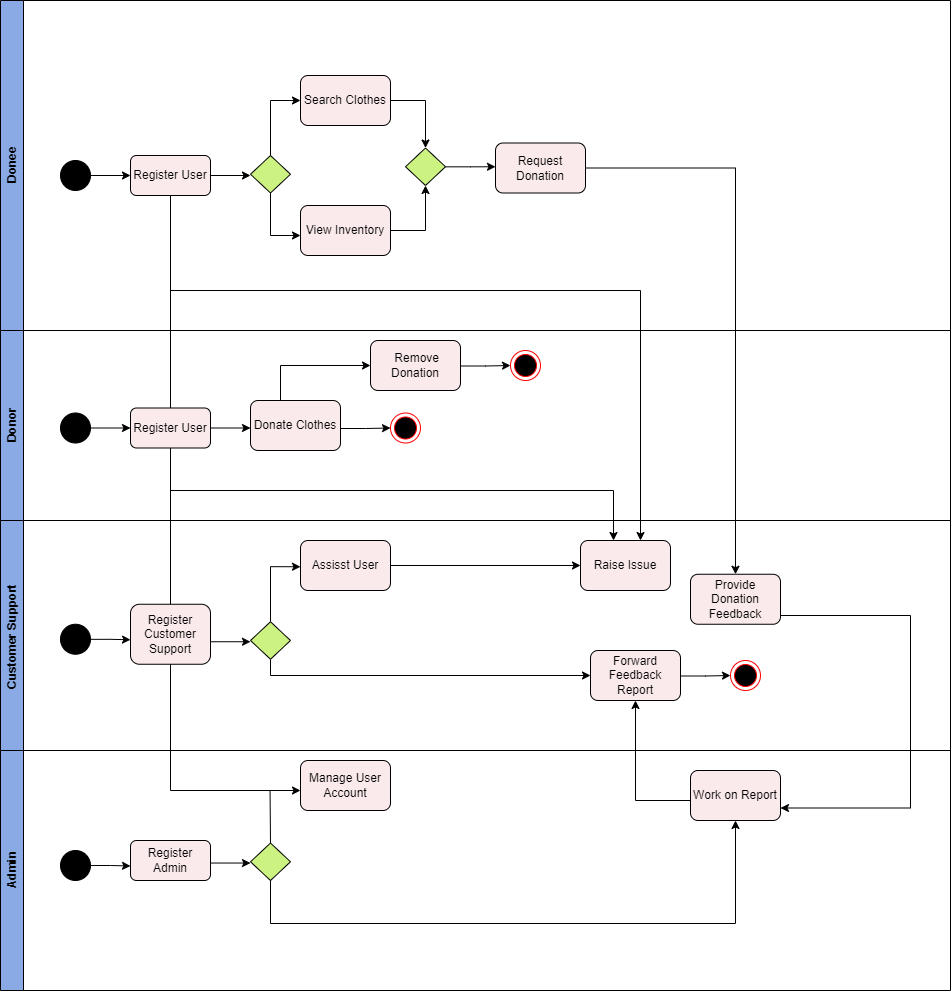
## **Definitions**

| Term | Definition |
| --- | --- |
| UML  DFD  ERD  CRUD matrix  Context Diagram | A UML diagram is a diagram based on the UML (Unified Modeling Language) with the purpose of visually representing a system along with its main actors, roles, actions, artifacts, or classes, in order to better understand, alter, maintain, or document information about the system.  A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles, and arrows, plus short text labels, to show data inputs, outputs, storage points, and the routes between each destination.  An entity relationship diagram (ERD), also known as an entity-relationship model, is a graphical representation that depicts relationships among people, objects, places, concepts, or events within an information technology (IT) system.  A CRUD matrix (Create, Read, Update, Delete) is a table that shows the links between processes and data, or between processes and resources. When a link exists, it shows whether the process performs a Create, Read, Update, or Delete operation on the data or resource.  The [Context Diagram](https://www.modernanalyst.com/Resources/Articles/tabid/115/ID/1355/Introduction-to-Context-Diagrams.aspx) shows the system under consideration as a single high-level process and then shows the relationship that the system has with other external entities (systems, organizational groups, external data stores, etc.). |

# **Process Modeling**

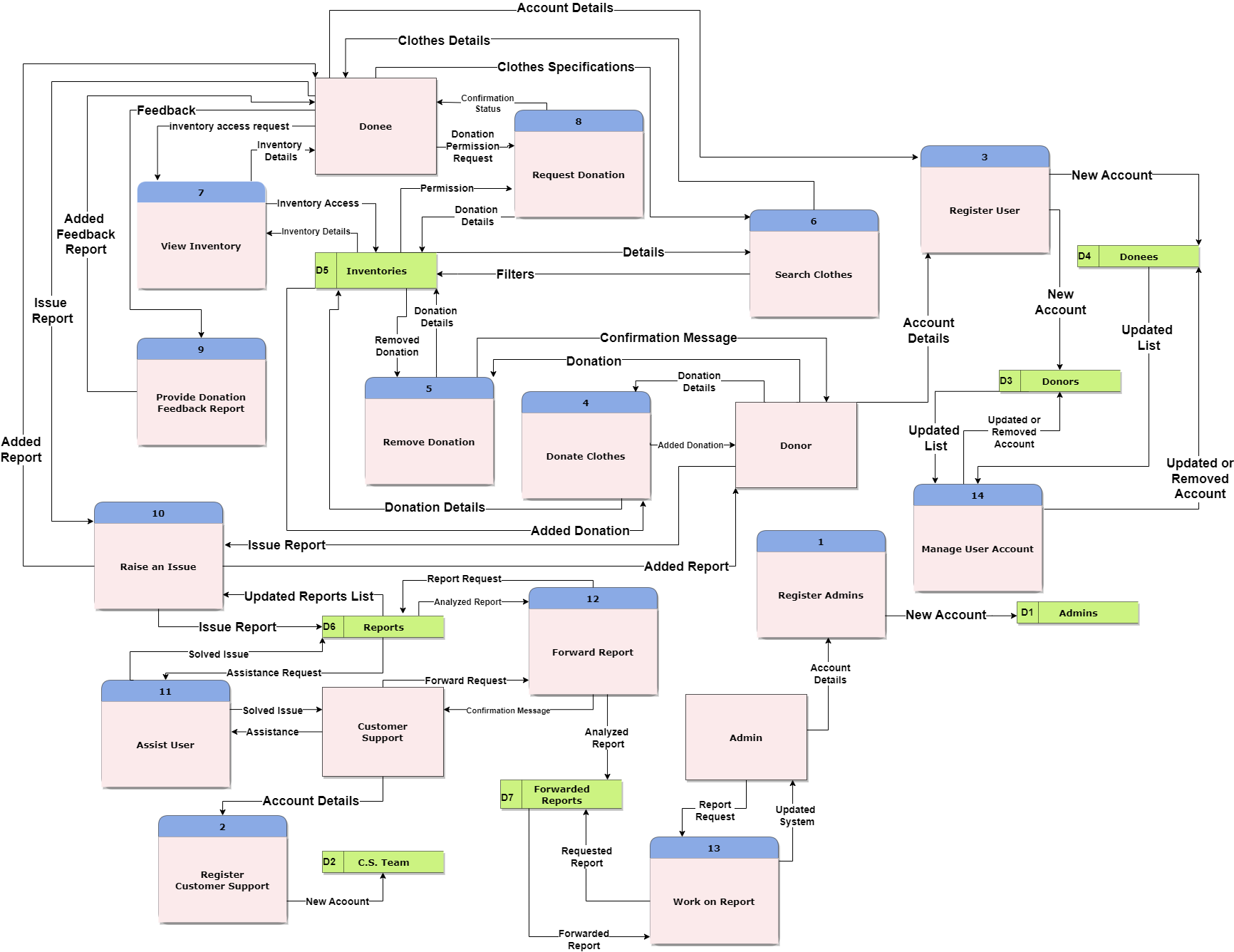


There are 4 external entities in the context diagram: Donee, Donor, Admin, and Customer Support. All entities register their accounts by providing their “Account Details” information. Donor provides its “Donation Details” to be able to make a “Donation” afterward, and receives “Confirmation Message” followed by “Added Donation”. Donee can choose the clothes needed by choosing “Clothes Specifications”, and getting “Clothes Details” displayed. Similarly, donee can submit “Inventory Access Request”, to review the state of inventory specified by getting “Inventory Details” displayed. After choosing a specific item, donee sends “Donation Permission Request” to be able to receive a donation by getting “Confirmation Status”. After getting needed donations, donee can leave “Feedback” and check it is being analyzed by viewing “Added Feedback Report”. Besides, if there is any issue encountered by either Donor or Donee, they can submit “Issue Report” to get help from Customer Support and get confirmation of it being reviewed through “Added Report”. Customer Support assists users in their needs through “Assistance” until getting confirmation of the issue being resolved via “Solved Issue”. Reports sent by users are analyzed by Customer Support and are related and sent through “Forward Request” to Admins for approval and then getting "Confirmation Message". Admin analyzes the reports sent by Customer Support, and after additional work, they get submitted through "Report Request" which results in the system being updated which is shown in "Updated Systems".



The activity diagram demonstrates that both Donee and Donor register as a User since they have the same registration process. Donee is able to Search Clothes or View Inventory to later Request Donation, and then provide Donation Feedback to Customer Support. Donors have the option of Donate Clothes and Remove Donation if needed. Both Donor and Donee can Raise Issue in case of any problem, which is sent to Customer Support, and is later reviewed by Admin together with feedback. After Admin approval, the analyzed reports are forwarded back to Donor or Donee by Customer Support. Admin has additional responsibilities of Managing User Accounts (Donees and Donors Accounts).

## **Level 0 diagram**



**External Entities:**

* Admin
* Customer Support
* Donor
* Donee

**Processes:**

* **Register Admin** - The process takes the account details to register the admin account. The account information is saved on the data store as a new account.
* **Register Customer Suppor**t - The process takes the account details to register the “Customer Support” account. The account information is saved on the data store as a new account.
* **Register User** - The process takes the account details to register both donor and donee accounts. The information is saved on the data store as a new account.
* **Donate Clothes** - Donor entity sends the donation details to the process for saving it as a new donation on the “inventories” data store. Then the data store sends the updated list of the new “inventories” list.
* **Remove Donation** - Donor entity sends donation details to the process for sending a request to the data store “inventories” for deleting it.
* **Search Clothes** - Donee sends specified donation details to the process which sends a request to the “inventories'' data store for filtering and finding the related clothes on the data store. Then the Data store sends related donation details which are saved on the system.
* **View Inventory** - Donee sends the request to the process which also sends a request to the “inventories” data store for getting the inventory details.
* **Request Donation** - Donee sends the donation details to the process for sending a request to the data store for getting permission on requested donation.
* **Provide Donation Feedback** - Donee sends the feedback report about the donation to the process. After that, the process saves the report on the “Reports” data store and sends the confirmation message to the Donee entity.
* **Raise an Issue** - Both donee and donor sends the Issue Report to the process which saves them in the “Reports” data store. Then the confirmation message is sent to the users.
* **Assist User** - Customer support sends an assistance request to the process. Process takes the Reports (which donor and donee reported for feedback or issue) from the data store and realizes the assistance process. Then the solved issue details are sent to the customer support.
* **Forward Feedback Report** - Customer support analyzes the reports on the data store “Reports” and sends the analyzed reports to the process. Then the process saves the reports on the “Forwarded Reports” data store.
* **Work on Report -** Admin sends a request to the process which takes the forwarded reports from the data store “forwarded reports”. Next, the process works on the report, improves or solves the related issue and sends the updated system details to the admin entity.
* **Manage User Account** - Admin sends details about the process should be taken on the account to the process. Process refers to the data stores (donee, donor) to get the required account and make operations on the data store. Then the data store sends the updated list of the donor/donee to the admin entity.

To properly list all the processes we used the following reference: Wasson, C. S. (2006). System Use Cases and Scenarios. In C. S. Wasson, *System Analysis, Design and Development* (p. 167).

**Data Stores**

* Admins
* Donors
* Donees
* C.S Team (Customer Support)
* Inventories
* Reports
* Forwarded Reports

**Evaluating DFDs**

**Process**

* Every process has a unique, action-oriented verb phrase names, and a number
* Every process has at least one input and output data flow.
* Output data flows have different names than input data flows

**Data Flow**

* Every data flow has a name that is a noun, and a description.
* Every data flow connects to at least one process.

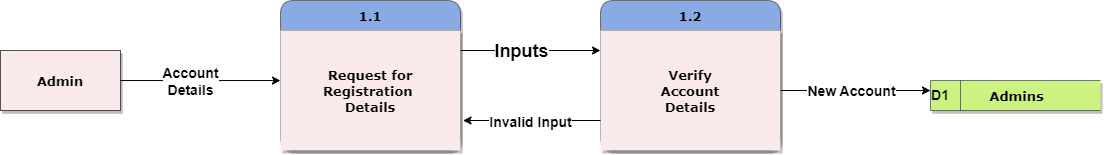
**Data Store**

* Every data store has a unique name that is a noun
* Every data store has at least one input data flow.
* Every data store has at least one output data flow (except “Forwarded report”, “Admins”, “C.S Team”)

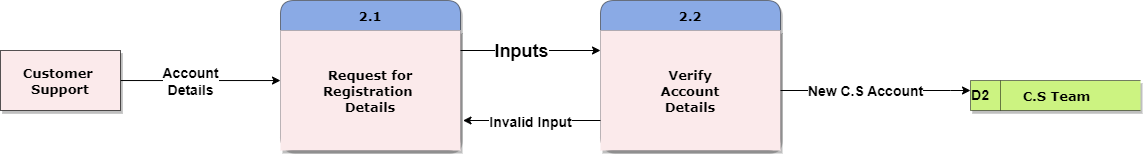
**External Entity**

* Every external entity has a unique name that is a noun
* Every external entity has at least one input or output data flow.

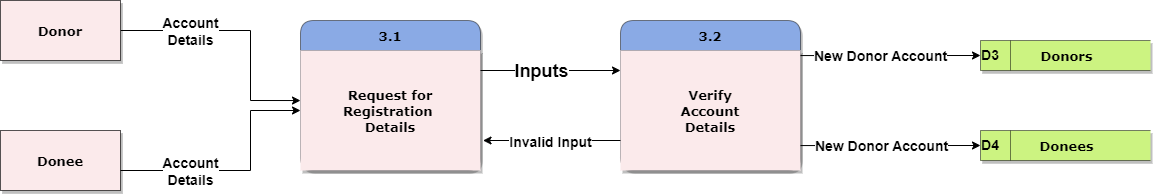
## **Level 1 diagrams**

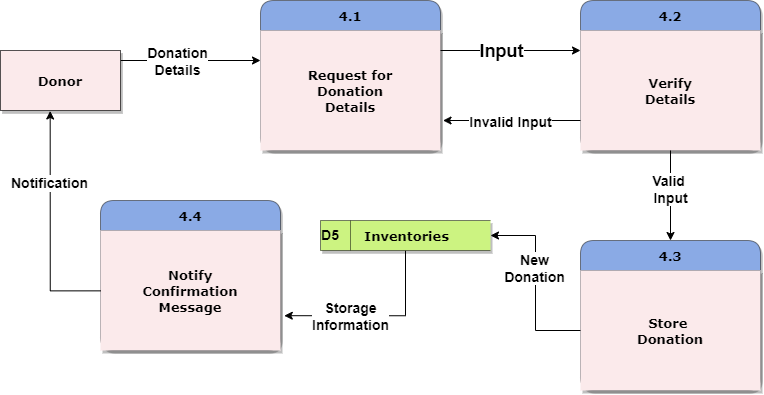


Admin Entity includes registration details for the system to verify and save as a new account on the data store.

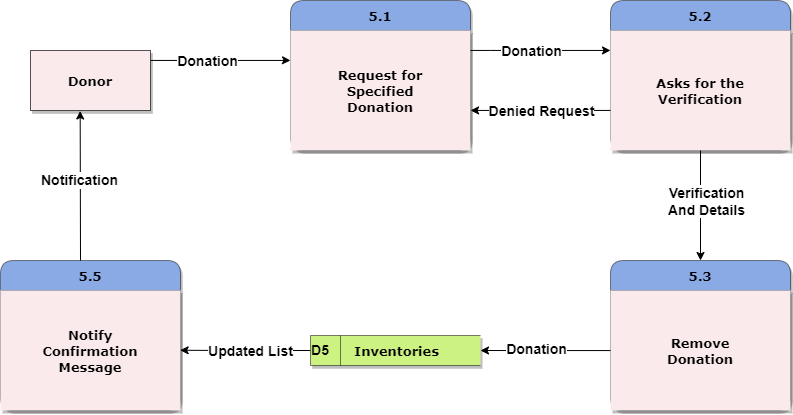


Customer Support includes registration details for the system to verify and save as a new account on the data store.

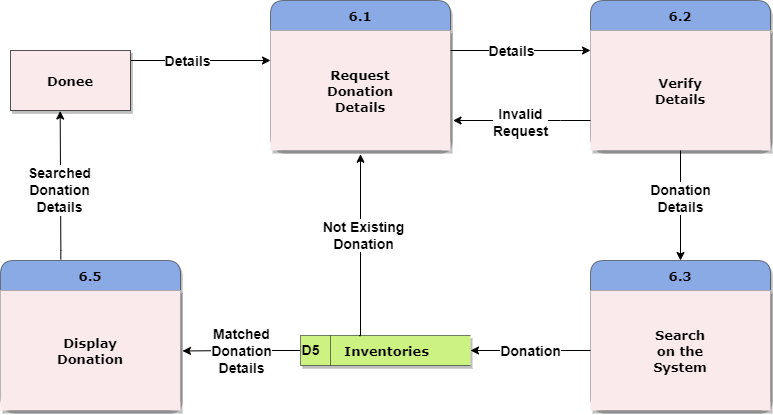




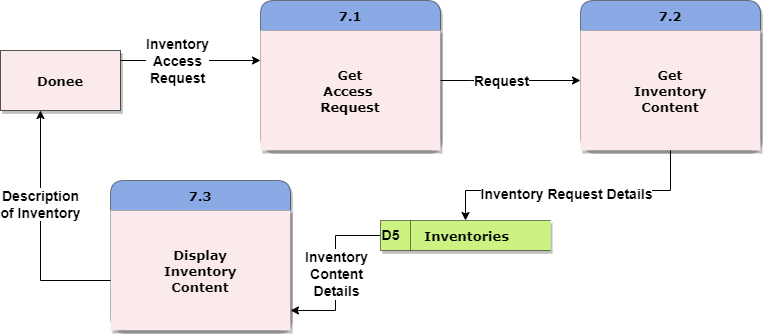
Donor provides donation details which system verifies them according to the specified criterias. After that, the system stores the donation specifications and sends completion status messages to the Donor.



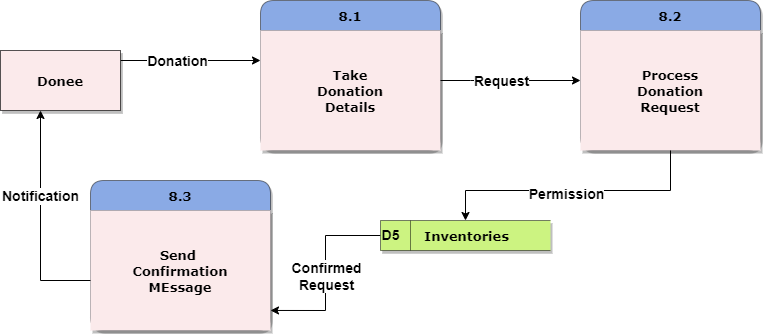
Donors provide specific donations to the system. System asks for the verification to make sure if the donor is sure to remove the donation from the system. If the verification is not provided, then the request is denied and Donor is again requested for the specific donation. Else, verification and details are sended to the system to be removed. After the process, the system updates the list and sends the notification message to the donor.



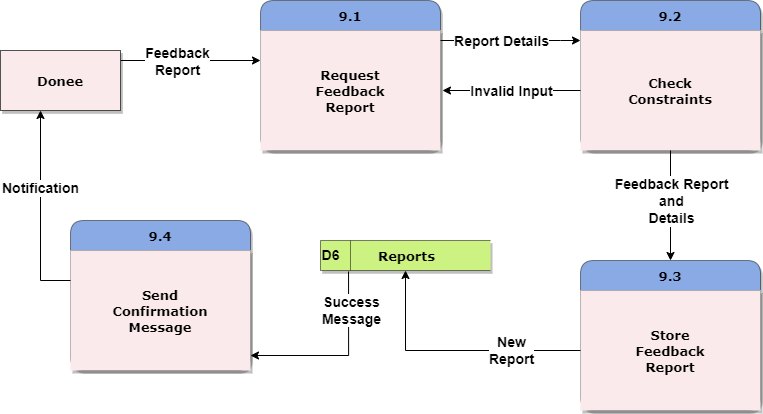
Donee provide donation details which are verified by the system first. After that, the specifications are passed to the system to make searches based on the specified criterias. If the criteria is not found on the system’s database, then the system turns to the first page which asks the donor to provide donation details. Else, the system passes the donation details which are later displayed to the donee.



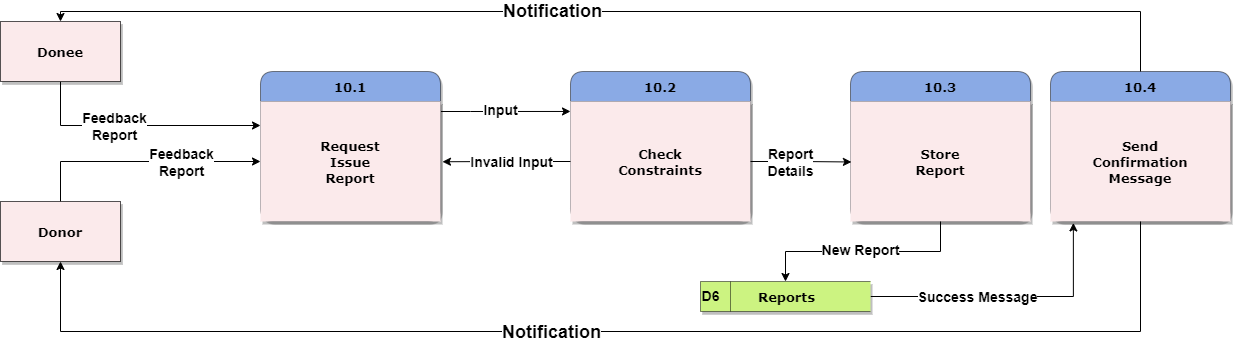
Donee sends the access request to the system. System forwards this request to the “inventories” data store to display the content of the specified Inventory.



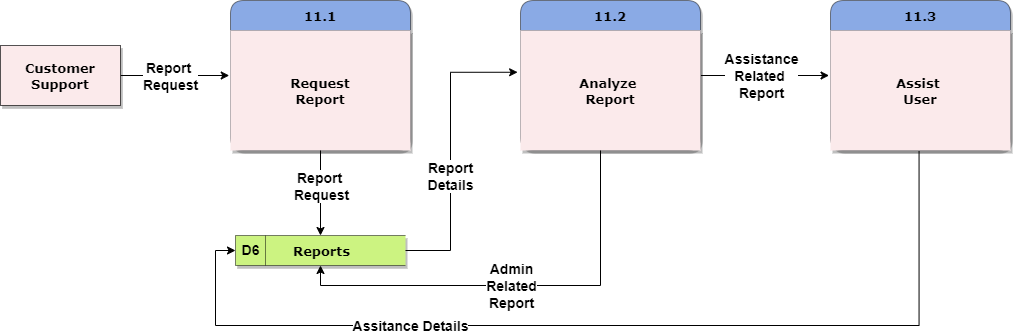
Donee provides specific donation details to the system. After that, the request is processed by the system and the permission is granted. System Displays the confirmation message

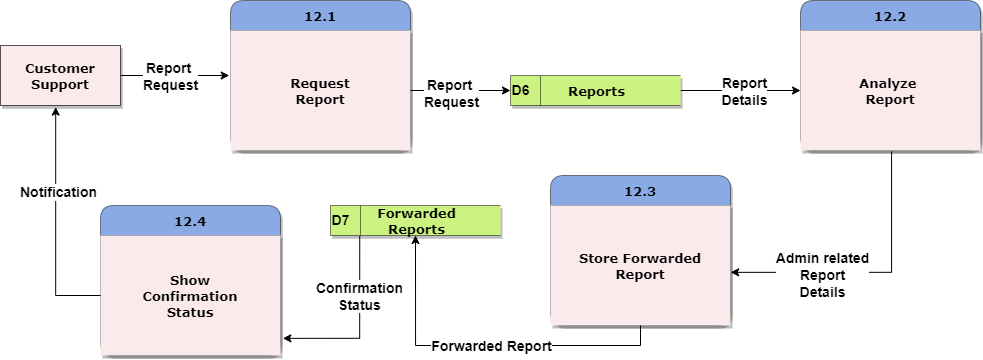


Donee Provides donation feedback report. System checks if there are any constraint issues. If there are any, then the system again returns to the first stage where it requests for the feedback report. Else, Feedback report details are stored on the data store and confirmation status is displayed to the Donee.

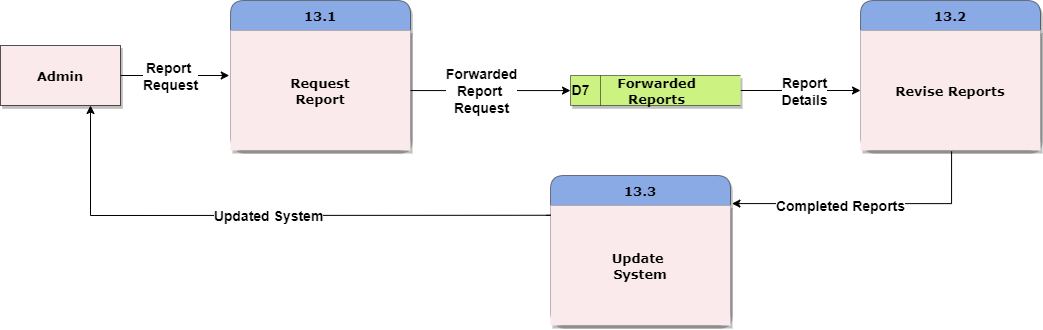


Both Donor and Donee provide Issue Reports to the system. System checks for the constraint issues accordingly. If there is any, then the system again returns to the first stage where it requests for the Issue report. Else, report details are stored on the data store and confirmation status is displayed to the Donee and Donor accordingly.

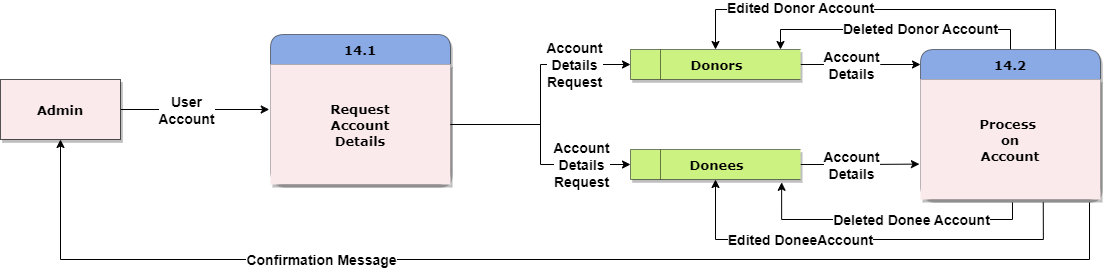
Customer Support requests the report from the system. After that, the report is analyzed, and if it is admin related (admins have to solve) then nothing changed on the report and saved to its place again. Else, if there is a small issue that customer service can handle, then they assist the users and assistance details are stored on the specified report. .



Customer Support again requests for the report. After getting and analyzing reports from the data store, it takes admin related reports and saves them into forwarded reports data store. Next, the system shows a confirmation message.



Admin requests for the report from the forwarded reports data store. After that, the admin works on the reports (fixes if there is an issue/creates new functionalities if needed) and updates the system. Updated system is displayed for the admin.



Admin requests for the User(donee, donor) account details which then have processes (delete/update). Confirmation message is sent afterwards.

This section was based on the materials obtained from the following reference: [https://www.google.com/url?sa=t&source=web&rct=j&url=https://www.cs.uct.ac.za/mit\_notes/software/pdfs/Chp06.pdf&ved=2ahUKEwjdgr7tkcD-AhUjSvEDHVTNDaYQFnoECBIQAQ&usg=AOvVaw13DEmzM6p\_VfoDS37cYi2-](https://www.cs.uct.ac.za/mit_notes/software/pdfs/Chp06.pdf)

# **Data Modeling**

***Admin:*** This entity stands in for the system's manager or administrator, who has complete authority over and access to all system functions. An Admin may manage and supervise numerous reports since the Admin entity has a one-to-many relationship with the Report entity.

*Attributes:*

-admin id

-admin name

-admin email

-admin password

-admin phone number

***Donor:*** This Entity stands for people or businesses that provide the system with used clothing donations. The relationship between the Donor-> Clothing, and Report entities is one-to-many. Because of this relation, a Donor may provide many pieces of clothing.

*Attributes:*

-donor id

-donor name

-donor email

-donor phone number

-donor password

-donor address

***Customer Support:*** This entity stands for the assistance provided by staff that helps users with any questions, problems, or issues they may have with the system. One customer support service representative may manage numerous Reports since the Customer Support entity has a one-to-many relationship with the Report entity.

*Attributes:*

-customer support id

-customer support name

-customer support email

-customer support password

-customer support phone

***Donee:*** This entity stands for people or groups who get used clothes via the system. A Donee can get numerous pieces of clothes from the Clothing and can have numerous reports since there is a one-to-many relationship between the Donee-> the Clothing and Report entities.

*Attributes:*

-donee id

-donee name

-donee email

-donee phone

-donee password

-donee address

***Clothing:*** This entity illustrates the donated used clothing items. The Donee entity and the Clothing entity have a many-to-many relationship, which allows for both the Donee to get numerous clothing items and for several Donees to receive the same clothes. But between the Donor and Clothing entities, we have one-to-many relationships since one Donor can give several clothing items.

*Attributes:*

-clothing id

-clothing category

-clothing gender

-clothing size

-clothing color

-clothing condition

***Inventory:*** This entity stands for every bit of the system's available supply of donated clothes. Multiple Donees may get clothes from the inventory since there is a one-to-many relationship between the Inventory and the Donee entity.

*Attributes:*

-donated clothing id

-donated clothing category

-donated clothing gender

-donated clothing size

-inventory location

-donated clothing color

-donated clothing condition

***Report:*** This entity reflects the comments made by Donors and Donees regarding the functionality and user experience of the system. A single Donor and Donee may provide several reports since the Donor and Donee and the Report entity have a one-to-many connection.

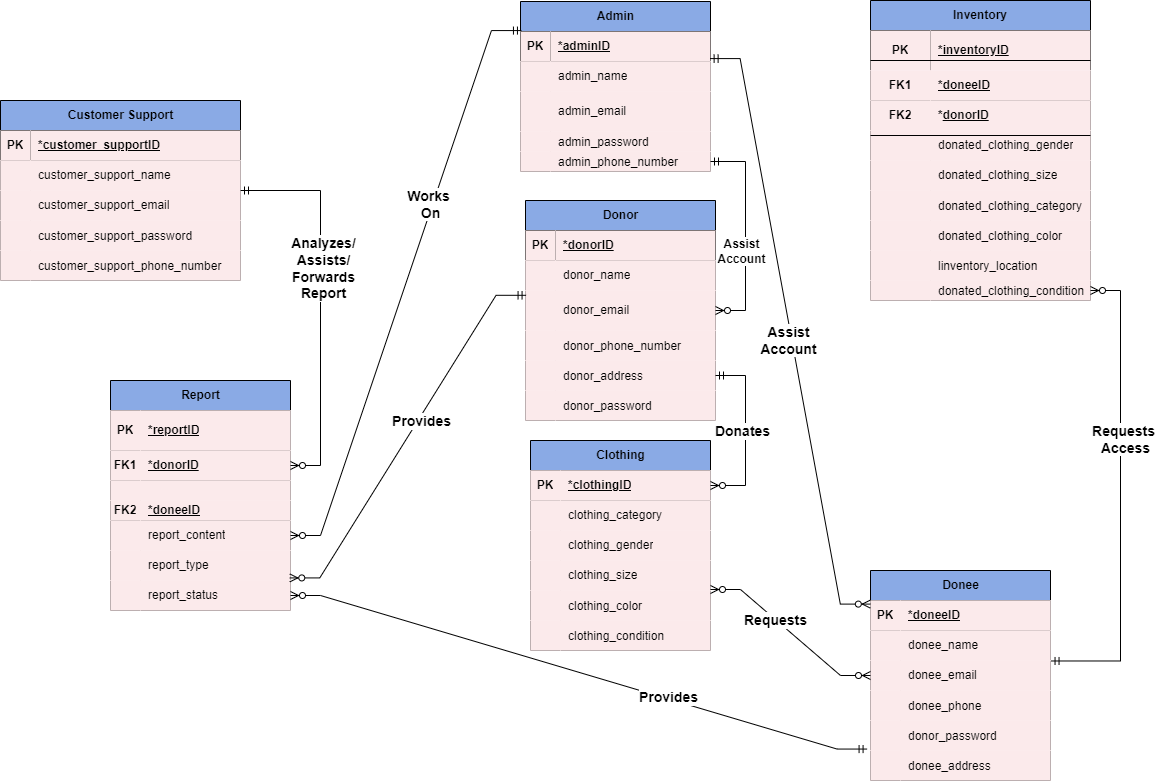
*Attributes:*

-report id

-report content

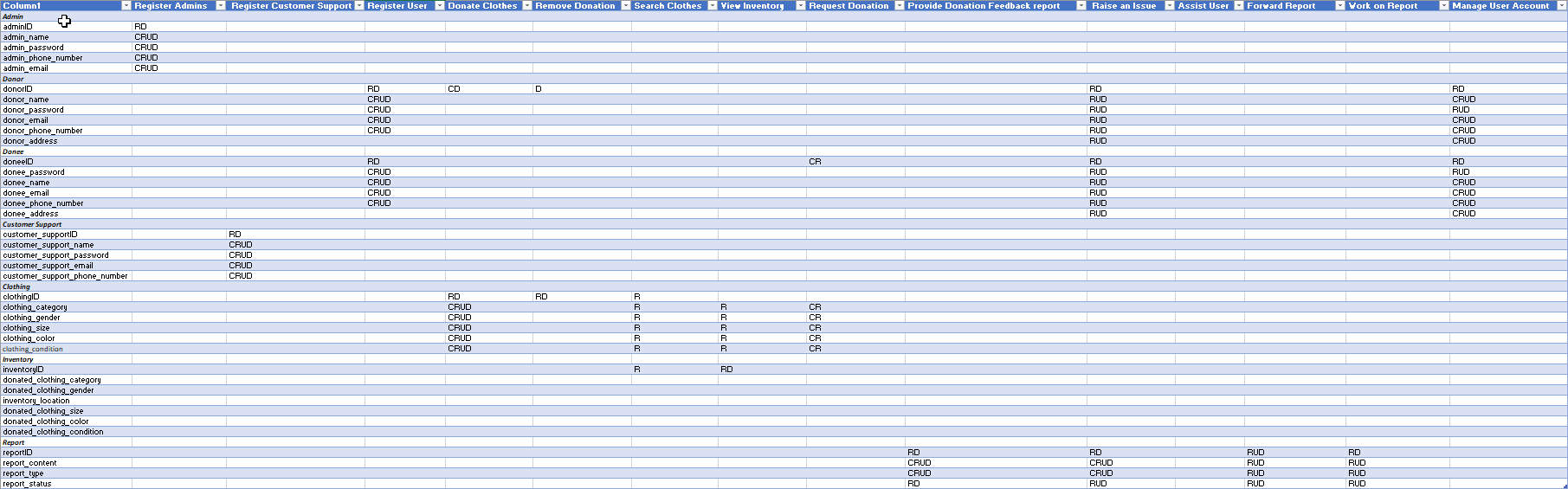
-report type

-report status

******

This section was based on the materials obtained from the following reference: Entity-Relationship Diagrams,” Information Sciences: An International Journal 29, no. 2–3 (1983): 127–149; Ian Graham, Migrating to Object Technology (Reading, MA: Addison Wesley Longman, 1995).

## **CRUD matrix**



This section was based on the materials obtained from the following reference: *Visual Expert Documentation*. (n.d.). Retrieved from Visual Expert: <https://www.visual-expert.com/EN/visual-expert-documentation/code-cross-references/crud-matrix-generator.html>

# **References**

* Wasson, C. S. (2006). System Use Cases and Scenarios. In C. S. Wasson, System Analysis, Design and Development (p. 167).
* Russell J. Abbott, “Program Design by Informal English Descriptions,” Communications of the ACM 26, no. 11 (1983): 882–894; Peter P-S Chen, “English Sentence Structure and Entity-Relationship Diagrams,” Information Sciences: An International Journal 29, no. 2–3 (1983): 127–149; Ian Graham, Migrating to Object Technology (Reading, MA: Addison Wesley Longman, 1995).
* Visual Expert Documentation. (n.d.). Retrieved from Visual Expert: https://www.visual-expert.com/EN/visual-expert-documentation/code-cross-references/crud-matrix-generator.html
* Retrieved from: https://www.google.com/url?sa=t&source=web&rct=j&url=https://www.cs.uct.ac.za/mit\_notes/software/pdfs/Chp06.pdf&ved=2ahUKEwjdgr7tkcD-AhUjSvEDHVTNDaYQFnoECBIQAQ&usg=AOvVaw13DEmzM6p\_VfoDS37cYi2-