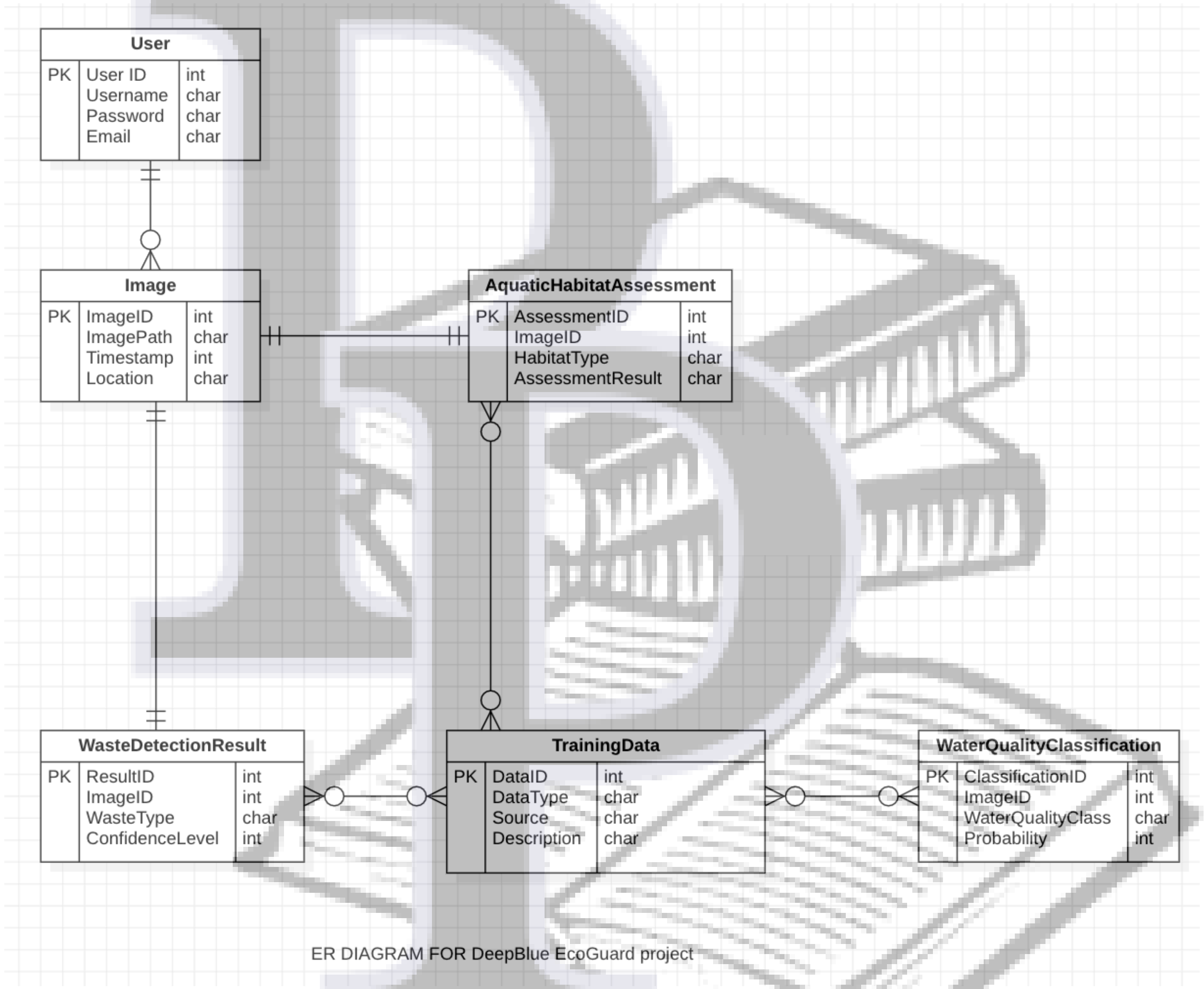
Software Engineering Lab
Assignment 2

1. Requirement modelling using Entity Relationship Diagram(Structural Modeling)



ER DIAGRAM FOR DeepBlue EcoGuard project

Entity-Relationship (ER) diagram for the DeepBlue EcoGuard software engineering project:

Entities:

User
Attributes: UserID (Primary Key), Username, Password, Email

Image
Attributes: ImageID (Primary Key), ImagePath, Timestamp, Location

WasteDetectionResult
Attributes: ResultID (Primary Key), ImageID (Foreign Key), WasteType, ConfidenceLevel

AquaticHabitatAssessment

Attributes: AssessmentID (Primary Key), ImageID (Foreign Key), HabitatType, AssessmentResult

WaterQualityClassification
Attributes: ClassificationID (Primary Key), ImageID (Foreign Key), WaterQualityClass, Probability

TrainingData
Attributes: DataID (Primary Key), DataType, Source, Description

Relationships:

User captures Image:
One-to-Many relationship from User to Image
Each User can capture multiple Images, but each Image is captured by one User.

Image is used for Waste Detection:
One-to-One relationship from Image to WasteDetectionResult
Each Image may have one WasteDetectionResult, and each WasteDetectionResult is associated with one Image.

Image is used for Habitat Assessment:
One-to-One relationship from Image to AquaticHabitatAssessment
Each Image may have one AquaticHabitatAssessment, and each AquaticHabitatAssessment is associated with one Image.

Image is used for Water Quality Classification:
One-to-One relationship from Image to WaterQualityClassification
Each Image may have one WaterQualityClassification, and each WaterQualityClassification is associated with one Image.

Waste Detection Result uses Training Data:
Many-to-Many relationship between WasteDetectionResult and TrainingData
A WasteDetectionResult is based on multiple TrainingData entries, and each TrainingData entry may be used in multiple WasteDetectionResults.

Aquatic Habitat Assessment uses Training Data:
Many-to-Many relationship between AquaticHabitatAssessment and TrainingData
An AquaticHabitatAssessment is based on multiple TrainingData entries, and each TrainingData entry may be used in multiple AquaticHabitatAssessments.

Water Quality Classification uses Training Data:
Many-to-Many relationship between WaterQualityClassification and TrainingData
A WaterQualityClassification is based on multiple TrainingData entries, and each TrainingData entry may be used in multiple WaterQualityClassifications.

This ER diagram represents the relationships between the main entities involved in the DeepBlue EcoGuard project, highlighting the connections between users, captured images, waste detection results, aquatic habitat assessments, water quality classifications, and the training data used for the machine learning models.

Creating ER diagram on StarUML:

To create a Entity-Relationship Diagram:
- Select first an element where a new Entity-Relationship Diagram to be contained as a child.
- Select Model | Add Diagram | ER Diagram in Menu Bar or select Add Diagram | ER Diagram in Context Menu.

Data Model
To create a Data Model (model element only) by Menu:

- Select an Element where a new Data Model to be contained.
- Select Model | Add | Data Model in Menu Bar or Add | Data Model in Context Menu.

Entity
To create an Entity:
- Select Entity in Toolbox.
- Drag on the diagram as the size of Entity.

To create a Entity (model element only) by Menu:
- Select a Data Model where a new Entity to be contained.
- Select Model | Add | Entity in Menu Bar or Add | Entity in Context Menu.

You can use QuickEdit for Entity by double-click or press Enter on a selected Entity.
- Name : Enter name.
- Add Note : Add a linked note.
- Add Column (Ctrl+Enter) : Add a column.
- Add One to One : Add an one-to-one relationship with an entity.
- Add One to Many : Add an one-to-many relationship with an entity.
- Add Many to Many : Add an many-to-many relationship with an entity.

Column
To add a Column:
- Select an Entity.
- Select Model | Add | Column in Menu Bar or Add | Column in Context Menu.

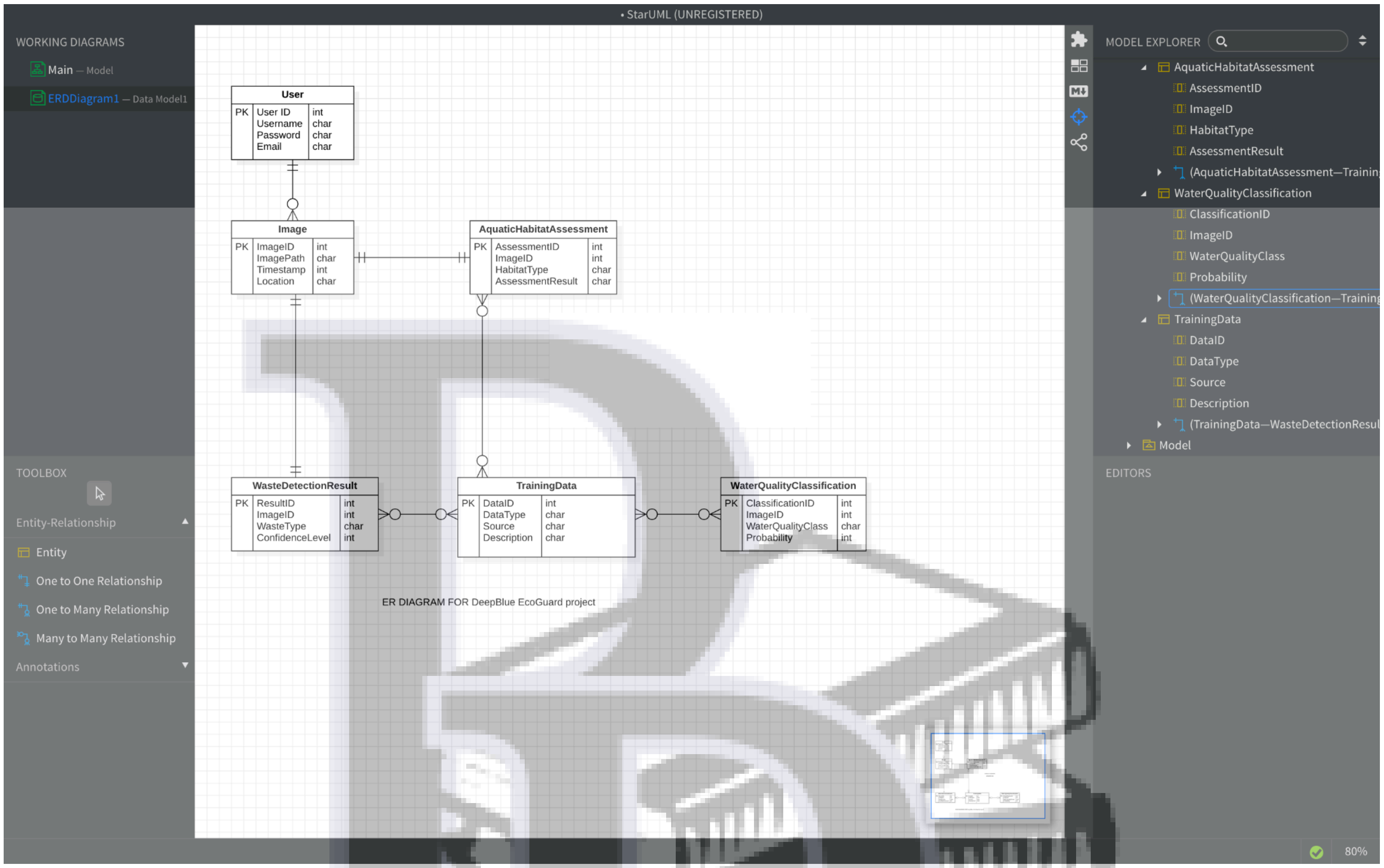
You can use QuickEdit for Column by double-click or press Enter on a selected Column.

- Column Expression : Edit column expression.
  Syntax of Column Expression
  ```
  column ::= name [ ':' type ] [ '(' length ')' ]
  name ::= (identifier)
  type ::= (identifier)
  length ::= (string)
  ```
- Primary Key : Check whether the column is primary key or not.
- Add (Ctrl+Enter) : Add one more column in the below.
- Delete (Ctrl+Delete) : Delete the column
- Move Up (Ctrl+Up) : Move the column up.
- Move Down (Ctrl+Down) : Move the column down.

Relationship
To create a Relationship:
- Select One-to-One Relationship, One-to-Many Relationship or Many-to-Many Relationship in Toolbox.
- Drag from an entity and drop on another entity.

You can use QuickEdit for Relationship by double-click or press Enter on a selected Relationship.
- Name : Enter name.
- Identifying : Check whether the relationship is identifying or not.
- Add Note : Add a linked note.

You can also use QuickEdit for Relationship End by double-click at the end side of a Relationship.
- Name : Enter name.
- Cardinality : Select cardinality of the selected relationship end.

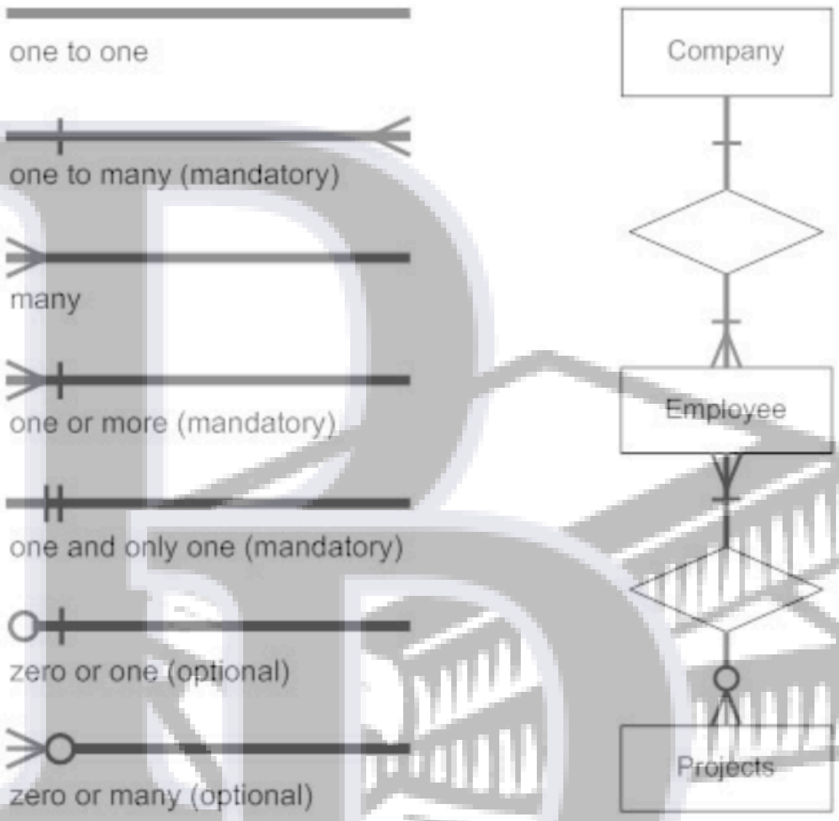


ER (Entity Relationship) Diagram:

- ER model stands for an Entity-Relationship model. It is a high-level data model. This model is used to define the data elements and relationship for a specified system.
- It develops a conceptual design for the database. It also develops a very simple and easy to design view of data.
- In ER modeling, the database structure is portrayed as a diagram called an entity-relationship diagram.

Notation of ER diagram

Database can be represented using the notations. In ER diagram, many notations are used to express the cardinality. These notations are as follows:



2. Requirement modelling using Context flow diagram, DFD (Functional Modeling)

Requirement modelling using Context Flow Diagram (CFD) and Data Flow Diagram (DFD) is a way to visually represent the functional requirements of a system.

Context Flow Diagram (CFD):

External Entities:
- User: Interacts with the system by capturing and submitting images for waste detection, habitat assessment, and water quality classification.
- Environment: Represents the underwater ecosystem where the system operates.

Processes:
- Waste Detection Process: Accepts user-input images, processes them using the YoloV8 Algorithm, and provides waste detection results to the User.
- Habitat Assessment Process: Takes user-input images, assesses aquatic habitat using a rule-based classifier, and outputs results to the User.
- Water Quality Classification Process: Accepts user-input images, classifies water quality using a machine learning model, and provides classification results to the User.

Data Flows:
- User Captured Images: Flow from the User to all three processes.
- Waste Detection Results: Flow from Waste Detection Process to the User.
- Habitat Assessment Results: Flow from Habitat Assessment Process to the User.

- Water Quality Classification Results: Flow from Water Quality Classification Process to the User.

Training Data Flow:
- Training Data: Flows from the system to all three processes, supporting their respective algorithms/models.

Data Flow Diagram (DFD) - Level 0:

Processes:
- User Interface: Represents the interaction point where the User submits images and receives results.
- Waste Detection System: Manages waste detection processing.
- Habitat Assessment System: Manages aquatic habitat assessment processing.
- Water Quality Classification System: Manages water quality classification processing.
- System Control: Coordinates and manages overall system control.

External Entities:
- User
- Environment

Data Flows:
- User Input Data: Flow from the User Interface to all three processing systems.
- Waste Detection Results: Flow from Waste Detection System to the User Interface.
- Habitat Assessment Results: Flow from Habitat Assessment System to the User Interface.
- Water Quality Classification Results: Flow from Water Quality Classification System to the User Interface.
- Training Data: Flow from System Control to all three processing systems.

Data Stores:
- Image Repository: Stores captured images submitted by the User.
- Waste Detection Results Store: Stores results of waste detection.
- Habitat Assessment Results Store: Stores results of habitat assessment.
- Water Quality Classification Results Store: Stores results of water quality classification.
- Training Data Repository: Stores training data for machine learning models.

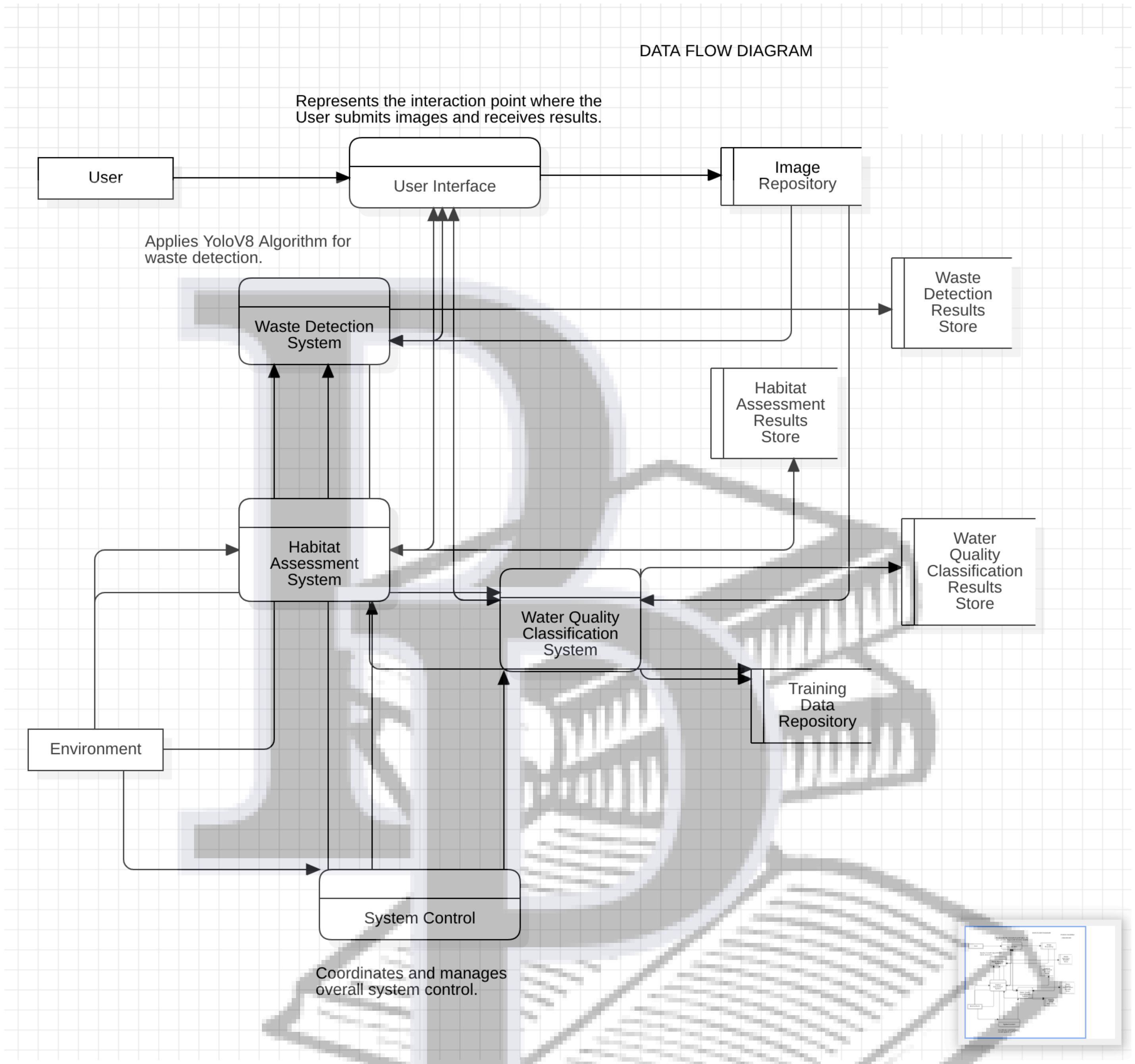
Level 1 DFD (Example: Waste Detection System):

Processes:
- Waste Detection Algorithm: Applies YoloV8 Algorithm for waste detection.
- Training Data Retrieval: Retrieves relevant training data for the waste detection algorithm.
- Result Storage: Stores waste detection results.

Data Stores:
- Image Repository
- Waste Detection Results Store
- Training Data Repository

Data Flows:
- User Input Data: Flow from User Interface to Waste Detection Algorithm and Training Data Retrieval.
- Waste Detection Results: Flow from Waste Detection Algorithm to Result Storage.
- Training Data: Flow from System Control to Training Data Retrieval.

This detailed requirement modelling using CFD and DFD provides a structured representation of the system's functionalities, external entities, processes, data flows, and data stores.



DATA FLOW DIAGRAM
Represents the interaction point where the User submits images and receives results.
User
User Interface
Image Repository
Applies YoloV8 Algorithm for waste detection.
Waste Detection System
Waste Detection Results Store
Habitat Assessment Results Store
Habitat Assessment System
Water Quality Classification Results Store
Water Quality Classification System
Training Data Repository
Environment
System Control
Coordinates and manages overall system control.

Creating Data Flow Diagram on StarUML:

To create a Data Flow Diagram:

- Select first an element where a new Data Flow Diagram to be contained as a child.
- Select Model | Add Diagram | Data Flow Diagram in Menu Bar or select Add Diagram | Data Flow Diagram in Context Menu.

Data Flow Model

- To create a Data Flow Model (model element only) by Menu:
- Select an Element where a new Data Flow Model to be contained.
- Select Model | Add | Data Flow Model in Menu Bar or Add | Data Flow Model in Context Menu.

External Entity

To create an External Entity:

- Select External Entity in Toolbox.

- Drag on the diagram as the size of External Entity.

You can use QuickEdit for External Entity by double-click or press Enter on a selected External Entity.

- Name : Edit name.
- Add Note : Add a linked note.
- Add Outgoing Process : Add an outgoing data flow with a process.

Process

To create a Process:

- Select Process in Toolbox.
- Drag on the diagram as the size of Process.

You can use QuickEdit for Process by double-click or press Enter on a selected Process.

- Name : Edit name.
- Add Note : Add a linked note.
- Add Incoming External Entity : Add an incoming data flow with an external entity.
- Add Outgoing Process : Add an outgoing data flow with a process.
- Add Outgoing Datastore : Add an outgoing data flow with a datastore.

Datastore

To create a Datastore:

- Select Datastore in Toolbox.
- Drag on the diagram as the size of Datastore.

You can use QuickEdit for Datastore by double-click or press Enter on a selected Datastore.

- Name : Edit name.
- Add Note : Add a linked note.
- Add Incoming Process : Add an incoming data flow with a process.

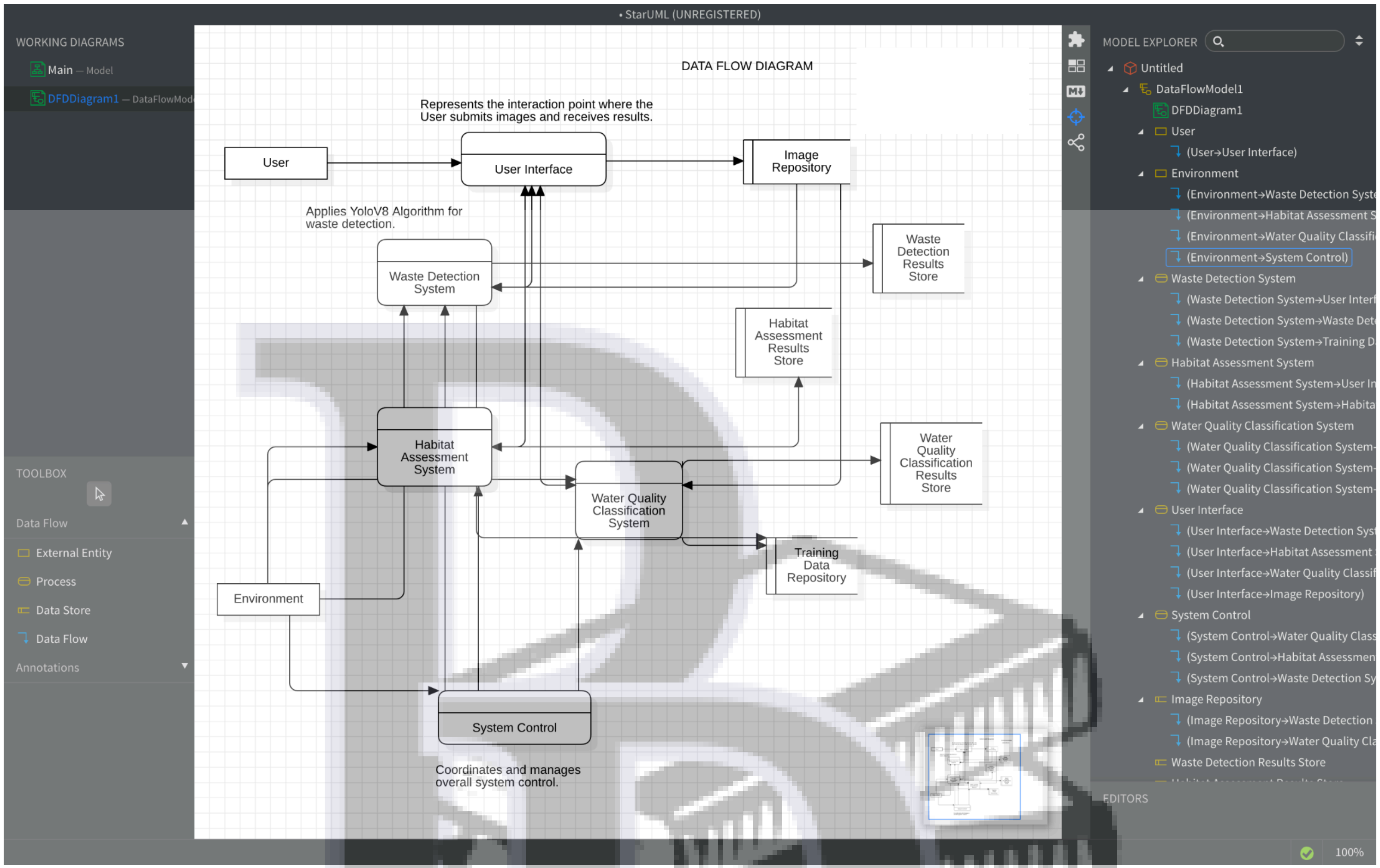
Data Flow

To create a Data Flow:

- Select Data Flow in Toolbox.
- Drag from an element and drop on another element.

You can use QuickEdit by double-click or press Enter on a selected data flow.

- Name : Edit name.
- Add Note : Add a linked note.



• StarUML (UNREGISTERED)
WORKING DIAGRAMS
Main — Model
DFDDiagram1 — DataFlowModel1
DATA FLOW DIAGRAM
Represents the interaction point where the User submits images and receives results.
User
User Interface
Image Repository
Applies YoloV8 Algorithm for waste detection.
Waste Detection System
Waste Detection Results Store
Habitat Assessment Results Store
Habitat Assessment System
Water Quality Classification System
Water Quality Classification Results Store
Training Data Repository
Environment
System Control
Coordinates and manages overall system control.
TOOLBOX
Data Flow
External Entity
Process
Data Store
Data Flow
Annotations
MODEL EXPLORER
Untitled
DataFlowModel1
DFDDiagram1
User
(User→User Interface)
Environment
(Environment→Waste Detection Syste
(Environment→Habitat Assessment S
(Environment→Water Quality Classifi
(Environment→System Control)
Waste Detection System
(Waste Detection System→User Interf
(Waste Detection System→Waste Dete
(Waste Detection System→Training D
Habitat Assessment System
(Habitat Assessment System→User In
(Habitat Assessment System→Habita
Water Quality Classification System
(Water Quality Classification System-
(Water Quality Classification System-
(Water Quality Classification System-
User Interface
(User Interface→Waste Detection Syst
(User Interface→Habitat Assessment
(User Interface→Water Quality Classif
(User Interface→Image Repository)
System Control
(System Control→Water Quality Class
(System Control→Habitat Assessmen
(System Control→Waste Detection Sy
Image Repository
(Image Repository→Waste Detection
(Image Repository→Water Quality Cla
Waste Detection Results Store
EDITORS
100%

Standard symbols for DFDs

Circle: A circle (bubble) shows a process that transforms data inputs into data outputs.

Data Flow: A curved line shows the flow of data into or out of a process or data store.

Data Store: A set of parallel lines shows a place for the collection of data items. A data store indicates that the data is stored which can be used at a later stage or by the other processes in a different order. The data store can have an element or group of elements.

Source or Sink: Source or Sink is an external entity and acts as a source of system inputs or sink of system outputs.

Levels in Data Flow Diagrams (DFD)

The DFD may be used to perform a system or software at any level of abstraction. Infact, DFDs may be partitioned into levels that represent increasing information flow and functional detail. Levels in DFD are numbered 0, 1, 2 or beyond. Here, we will see primarily three levels in the data flow diagram, which are: 0-level DFD, 1-level DFD, and 2-level DFD.

0-level DFDM

It is also known as the fundamental system model, or context diagram represents the entire software requirement as a single bubble with input and output data denoted by incoming and outgoing arrows. Then the system is decomposed and described as a DFD with multiple bubbles. Parts of the system represented by each of these bubbles are then decomposed and documented as more and more detailed DFDs. This process may be repeated at as many levels as necessary until the program at hand is well understood. It is essential to preserve the number of inputs and outputs between levels, this concept is called levelling by DeMacro. Thus, if bubble "A" has two inputs x1 and x2 and one output y, then the expanded DFD, that represents "A" should have exactly two external inputs and one external output as shown in fig:

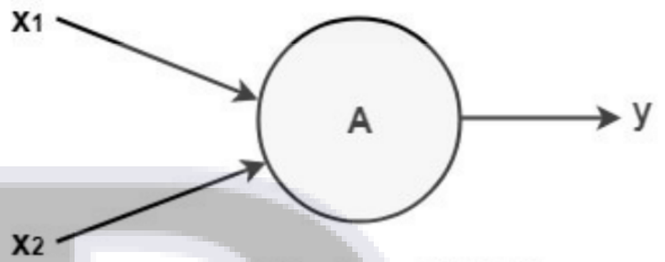


Fig: Level-0 DFD.

1-level DFD

In 1-level DFD, a context diagram is decomposed into multiple bubbles/processes. In this level, we highlight the main objectives of the system and break down the high-level process of 0-level DFD into subprocesses.

2-Level DFD

2-level DFD goes one process deeper into parts of 1-level DFD. It can be used to project or record the specific/necessary detail about the system's functioning.