Assignment 3: ADD and Frameworks

Step 1:

Category	**Details**
Design Purpose	To produce a sufficiently detailed design to support the construction of the system.
Primary Functional Requirements	UC 1 - Because it directly supports the core business. UC 2 - Because it directly supports the core business.
Quality Attributes	QA 1 - Portability

Iteration 2: Identifying structures to support primary functionality

Step 2: Establish Iteration Goal by Selecting Drivers

Our goal in this iteration is to allocate work to members of the development team and the following use cases will be addressed:

Use Case	**Description**
UC-1 Scanning Items	Prior to the scanning of any items the cashier must have commenced a purchasing session. The cashier scans each item that the customer contains. When a product has been identified using the barcode scanner, its name and price are displayed on a display.

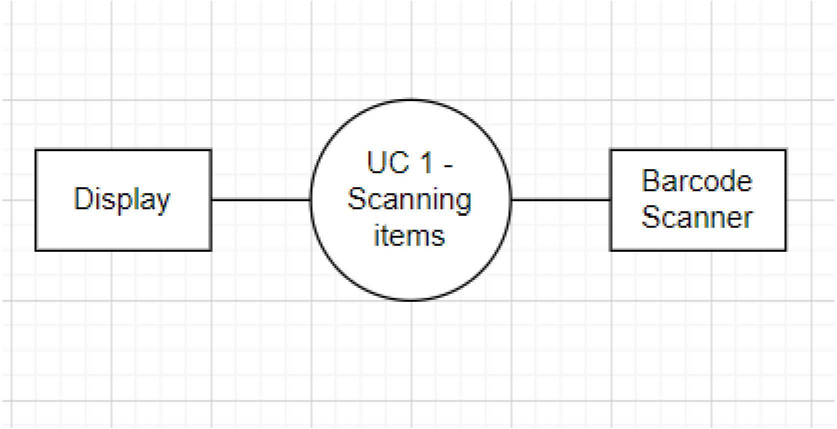
Quality Attributes

ID	**Quality Attribute**	**Description**
QA-1	Portability	The system must be portable to accommodate the need to run on different platforms and hardware.

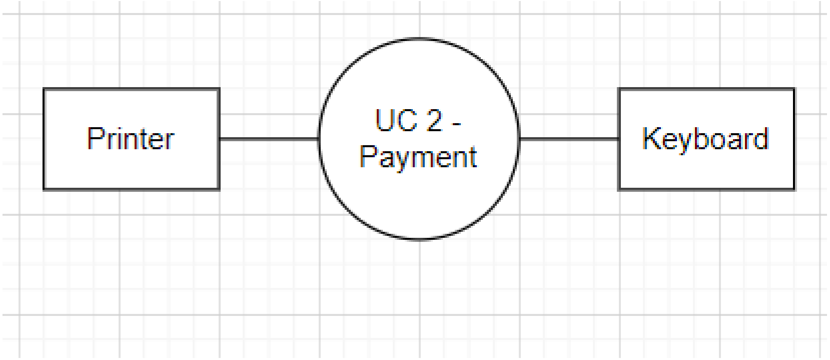
Constraints

ID	**Description**
CON-1	The Cash Register will contain a local database of products.

Structures that support Scanning items - Barcode scanner, display



Structures that support Payment - Keyboard, Printer



Step 3: Choose One or More Elements of the system to Refine

- We want to refine the UPC Access Module
- We want to refine the POS Access Module

Step 4: Choose One or More Design Concepts that Satisfy the Selected Drivers

Design Decisions and Location	**Rationale and Assumptions**
Create a Domain Model for the system.	Before starting functional decomposition, it is necessary to have an initial domain model of the system. The domain model will show the major entities like the Scanner, Display, etc and their relationships.
Domain object - Item Scanning	Responsibility of UC -1
Domain object - Payment	Responsibility of UC - 2
Decompose Item Scanning into UPC Scanner	UC - 1 is supported by UPC Scanner Module

Module	
Decompose Payment into POS Access Module	UC - 2 is supported by this module
Chosen frameworks: Spring and Hibernate	These frameworks will help with interfacing and database relation

Step 5: Instantiate Architectural Elements, Allocate Responsibilities and Define Interfaces

Design Decision	**Rationale Assumptions**
Create only an initial domain model	It shows the entities that participate in the primary use cases and helps accelerate design process as you would not have to create one
Map UC - 1 to Item Scanning	To identify domain objects, we analyzed the system's use case
Map UC - 2 to Payment	To identify domain objects, we analyzed the system's use case
Decompose Item Scanning to DisplayProductView on Presentation Layer	To identify all modules that support UC - 1
Decompose Item Scanning to DisplayProductController on Business Layer	To identify all modules that support UC - 1
Decompose Item Scanning to ItemScanningDataMapper on Data Layer	To identify all modules that support UC - 1
Decompose Payment to DisplayProductView on Presentation Layer	To identify all modules that support UC - 2
Decompose Payment to DisplayProductController on Business Layer	To identify all modules that support UC - 2
Decompose Payment to PaymentMapper on Data Layer	To identify all modules that support UC - 2
Associate Hibernate framework with RequestManager in the data layer.	Object to relational mapping (ORM) framework, like the Hibernate framework previously selected, is encapsulated in the modules contained in the data layer.

Step 6: Sketch Views and Record

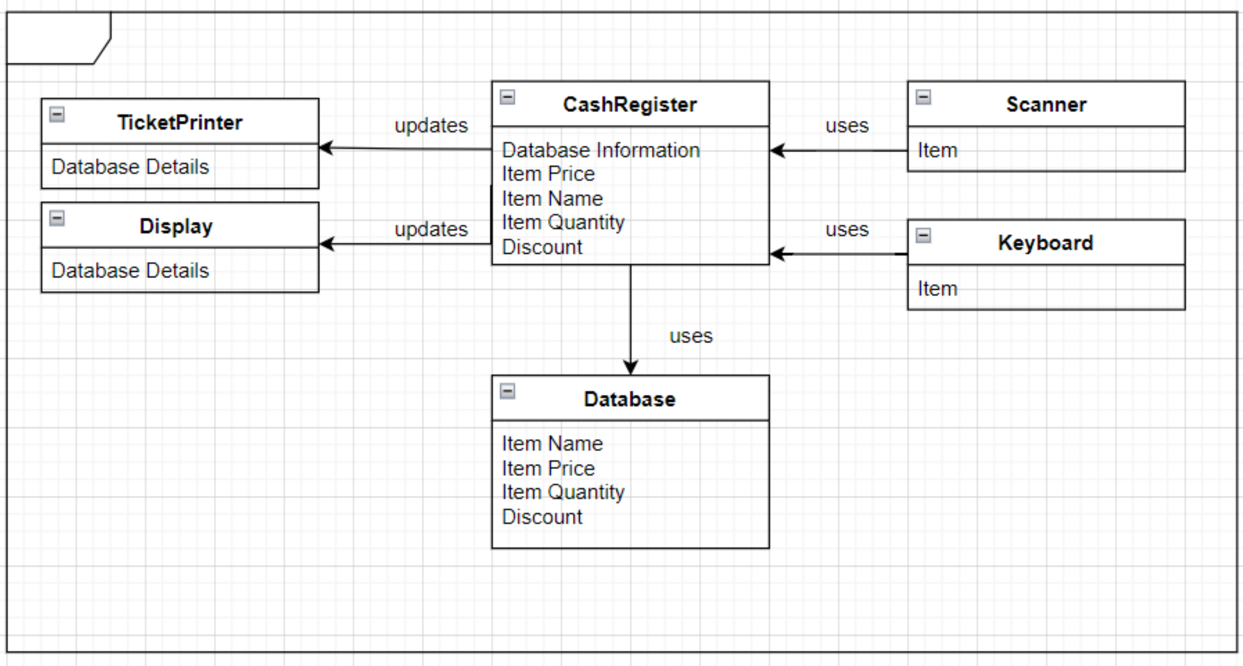


Figure 1 - Initial Domain Model (Class Diagram UML)

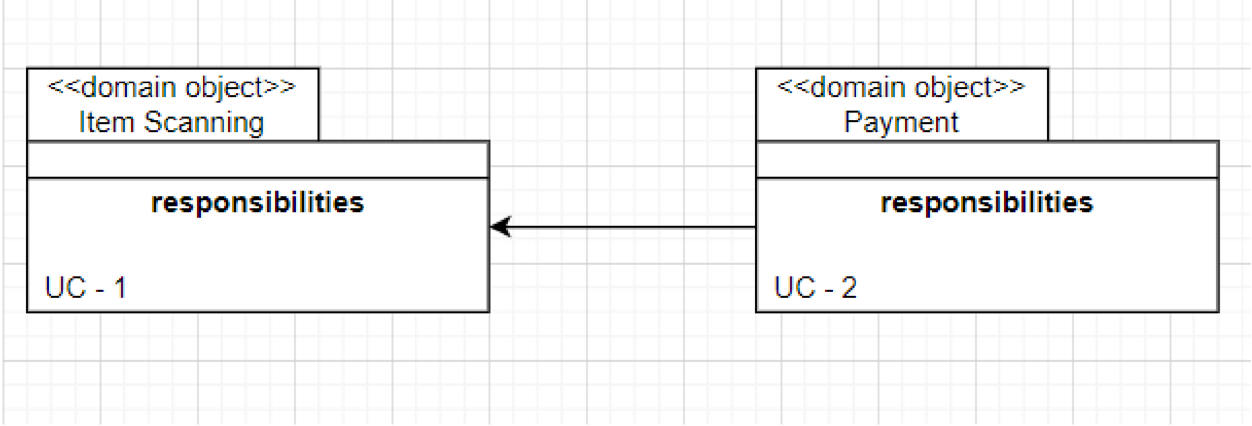


Figure 2 - Domain objects associated with the use case model

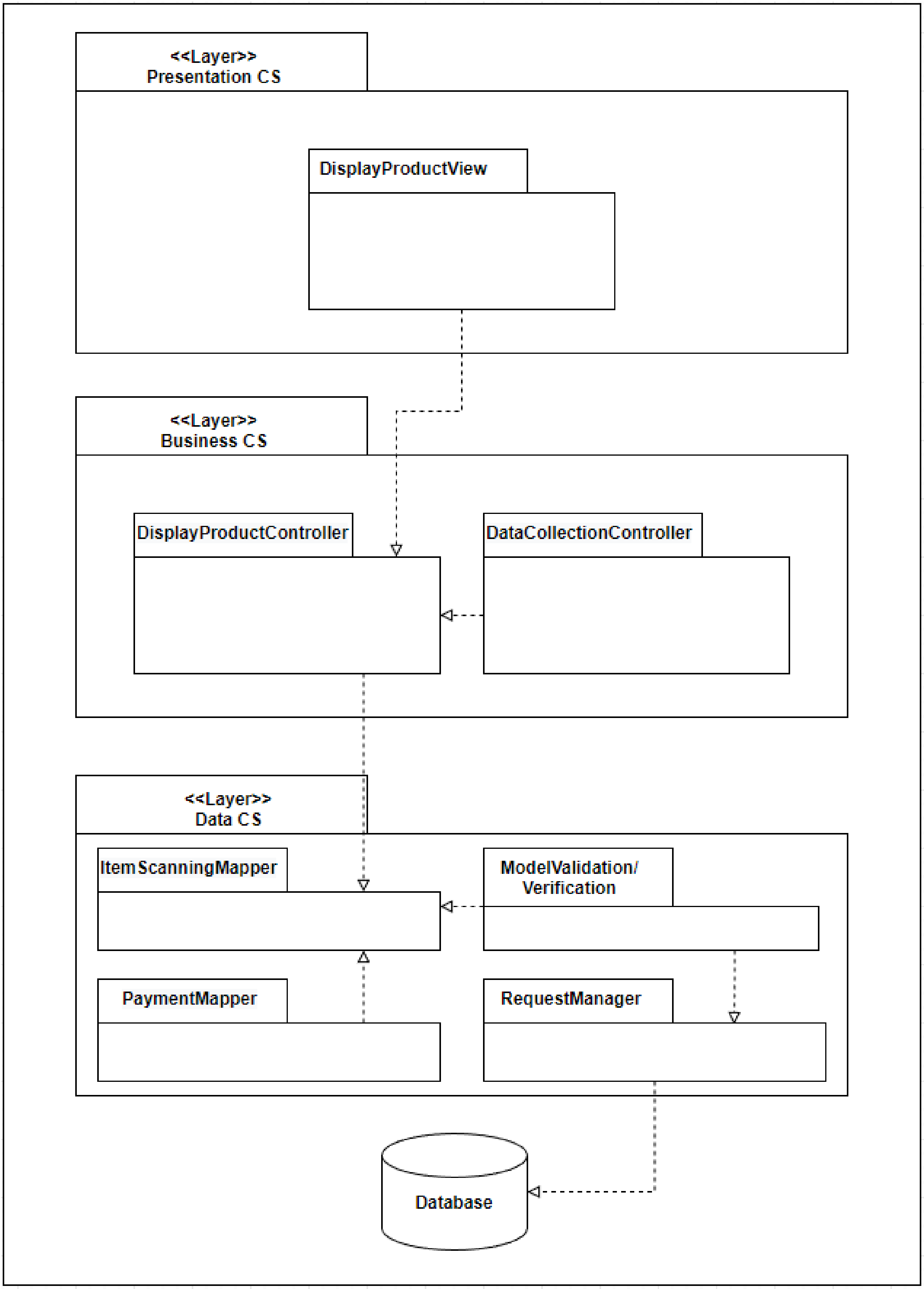


Figure 3 - Modules that support the primary use cases

Element	**Responsibility**
DisplayProductView	Displays product details of items purchased by the customer and updates it when events are received.
DisplayProductController	Responsible for providing the necessary information to the presentation layer for displaying products scanned.
ItemScanningMapper	Responsible for operations relating to scanning items for purchasing.
PaymentMapper	Responsible for operations relating to payment of items.

Step 7: Perform Analysis of Current Design

Not Addressed	**Partially Addressed**	**Completely Addressed**	**Design Decisions Made during Iteration**
		UC-1	
	QA-1	UC-2	
	CON-1		