Software Architecture
Tutorial 09

- **Group #1**

- **Student Names & ID:**

A – Ali Hakkani: 100867456
B –Lok Yung Chan, Carson - 100926449
C –Samuel Beamish - 100875987

Step 1: Review Input:

Category	Details
Design Purpose	This is a greenfield system designed for a retail environment. The purpose is to produce a detailed design for a Rich Client Cash Register Application that operates locally, processes sales transactions and remains portable across future hardware platforms
Primary Function Requirements	From the defined use cases, the primary ones were determined to be: **UC1:** Scanning Items: supports the core business process of identifying products and displaying prices. **UC2:** Payment: supports the core business process of completing customer purchases.
Quality Attribute Scenarios	The prioritized quality attribute is QA-1 Portability, rated high in importance. The system must be deployable on different hardware configurations with minimal modification, ensuring long-term adaptability and hardware independence.
Constraints	**CON-1**: The Cash Register must maintain a local database of products. **CON-2**: The system must operate as a standalone application independent of external servers or networks.
Architectural Concerns	Ensure hardware modularity so devices like scanners, printers and displays can be reconfigured without altering the main logic. Maintain offline reliability, fast response time and ease of maintenance within the Rich Client Environment

Step 2: Establish Iteration Goal

In this iteration, the goal is to refine the Cash Register subsystem so that the primary functionality of the system is portable, while following the Rich Client Architectural design from last iteration.

Step 3: Choose Elements to Decompose

In this iteration, the selected architectural drivers would be QA-1 and CON-2 as they focus on the independence of the system modules.

In QA-1, the stakeholders expect the system to be able to change hardware platforms in the future without affecting the primary functionality of the system. While in CON-2, the stakeholders requires the Cash Register to be a standalone application

Step 4: Choose Design Concept

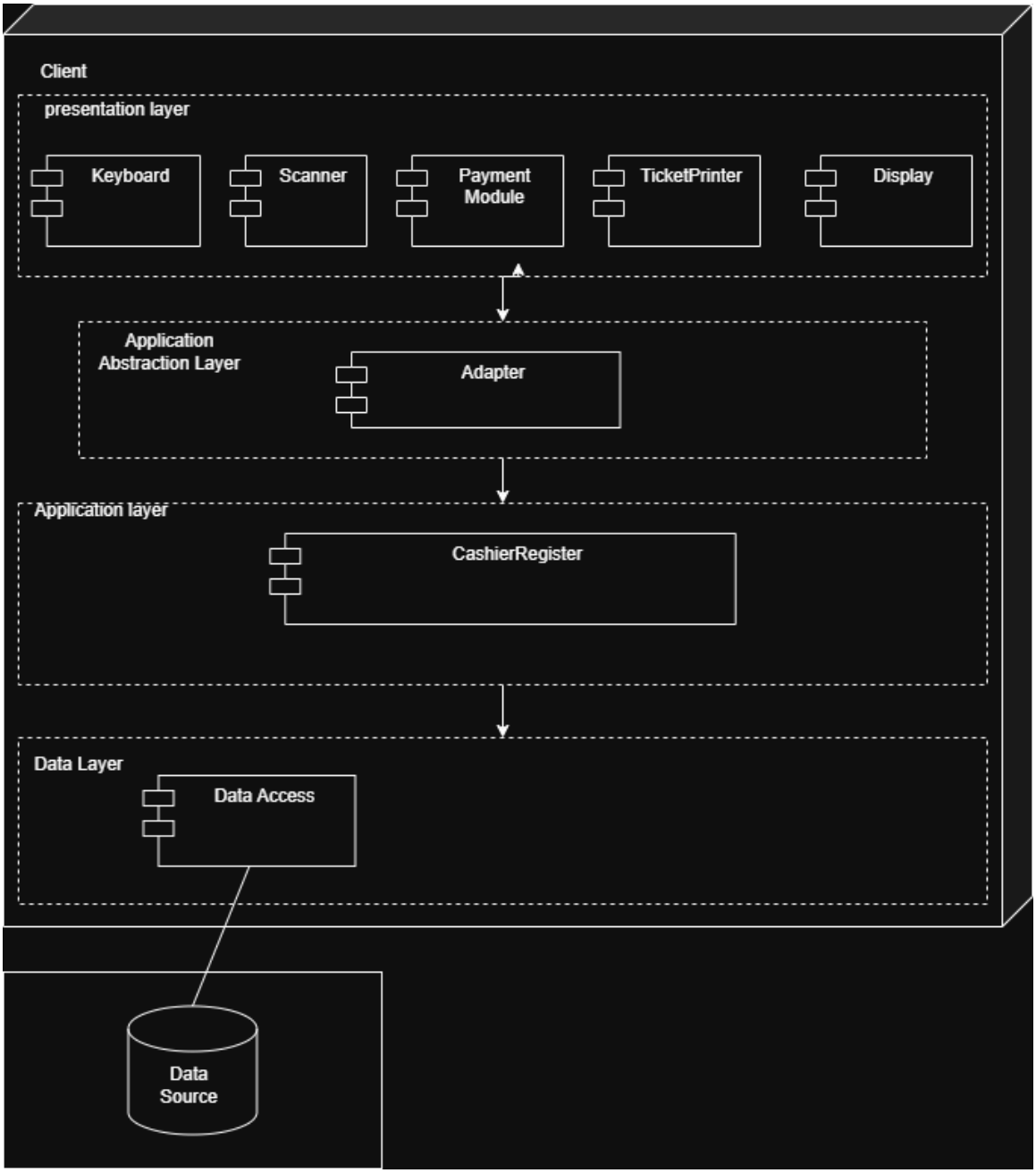
Adapter design pattern because it allows the system to connect with incompatible interfaces, meaning many different interfaces will be able to connect to the system. Although rich client patterns alone address the modularity of the business logic, it does not isolate platform specific hardware code. By following the adapter pattern, the Cash Register business logic can be easily ported to different hardware platforms without having to worry about compatibility between the logic and the hardware code.

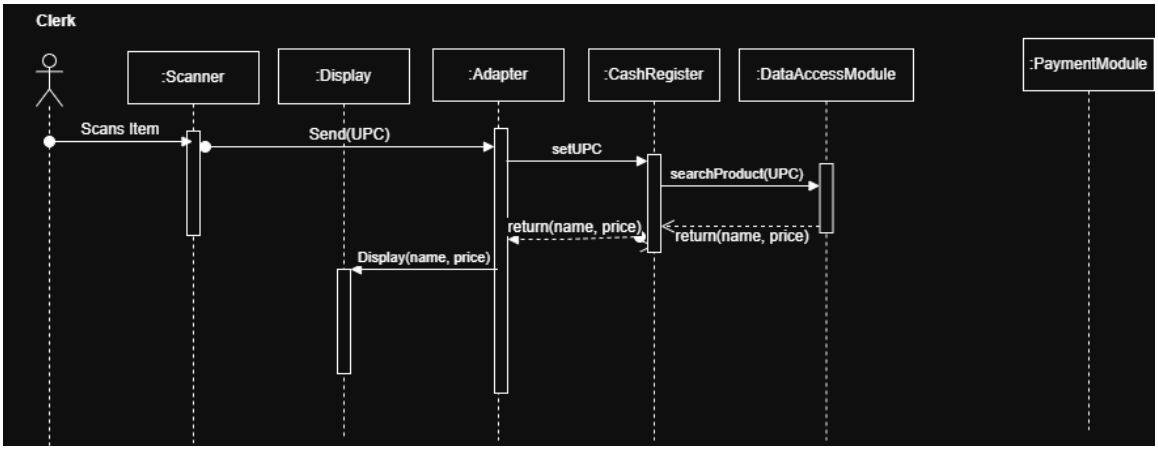
Step 5: Instantiate architectural element, allocate responsibility and define interfaces

Architectural element	Responsibility	Interface
Barcode Scanner	Read barcode and passes UPC to the application layer	Scanner Interface
Keyboard	Read user input and passes to application layer	Keyboard Interface
Product Display	Display product data received	Display interface
Ticket Printer	Print ticket	Printer interface
Payment Module	Select payment method, and payment process	Payment interface
Cash Register Application	Coordinate workflow of scanning, display, payment and data query	Controller interface
Data Access Module	Handle product data retrieval and log data query	Database interface
Adapter	Allow business logic to be easily installed in different hardware systems.	-Scanner interface: provides scanUPC() to get UPC reading for the business logic -Keyboard interface:

		provides readInput() to get UPC from text input field -Printer Interface provides setProducts to pass product from Business Logic layer to Printer -Display Interface: also provides set product to define scanned product for display -Payment Interface: provides getPrice() to pass product price to payment module

Step 6: Sketch Views And Record


Client
presentation layer
Keyboard
Scanner
Payment Module
TicketPrinter
Display
Application Abstraction Layer
Adapter
Application layer
CashierRegister
Data Layer
Data Access
Data Source



Step 7: Analyze the current design and evaluate whether the iteration goal and design objectives have been successfully met.

After going through the iteration and evaluation whether it reached the iteration goal, it was determined that the iteration goal was reached, as the hardware pieces were isolated, by using the adapter pattern, which ensures that the hardware can be switched without the main functionality being compromised which supports QA-1 which is portability.

The Cash Register also uses a local database, and does not rely on the network to access it which satisfies CON-2, hence it addresses stakeholders concerns of offline reliability, hardware modularity, maintainability, and fast responses, thus this iteration reaches its goals.

Iteration 3 analysis is not necessary, as there is not enough information to cover it extensively, just yet.