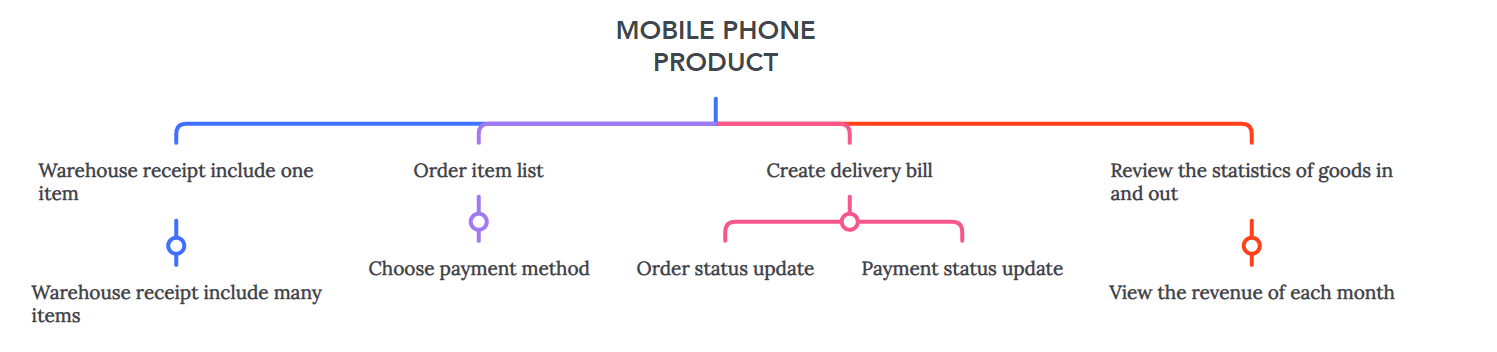
**CHAPTER 1 – INTRODUCTION**

**1.1 Purpose and Scope**

Purpose: Building management software for import/export, ordering, payment and revenue statistics for functional foods company for agents. And also we can get closer approach to the process of creating an actual software.

Scope: Work Breakdown Structure



**1.2 Product Overview**

- Accountants shall be able to create Goods Received when the company imports goods (a warehouse receipt will include many items).

- Agents shall be able to place an order of items and choose a payment method

(Cash, bank transfer, Momo...) (Webform).

- Accountants shall be able to create Goods Delivery Note to deliver goods to

agents (print delivery slips), update the status of orders as being transferred and

update the payment status of agents.

- Accountants shall be able to view incoming/outgoing stock report and revenue

report monthly.

**1.3 Structure of the Document**

1. Introduction

1.1. Purpose and Scope

1.2. Product Overview (including capabilities, scenariosfor using the product, etc.)

1.3. Structure of the Document

1.4. Terms, Acronyms, and Abbreviations

2. Project Management Plan

2.1. Project Organization

2.2. Lifecycle Model Used

2.3. Risk Analysis

2.4. Hardware and Software Resource Requirements

2.5. Deliverables and Schedule

2.6. Monitoring, Reporting, and Controlling Mechanisms

2.7. Professional Standards

2.8. Evidence all the artifacts have been placed under configuration management

2.9. Impact of the project on individuals and organizations

3. Requirement Specifications

3.1. Stakeholders for the system

3.2. Use case model

3.2.1. Graphical use case model

3.2.2. Textual Description for each use case

3.3. Functional requirements

3.4. Non-functional requirements

4. Architecture

4.1. Architectural style(s) used

4.2. Architectural model

4.3. Technology, software, and hardware used

4.4. Rationale for your architectural style and model

5. Design

5.1. Database design

5.2. Static model – class diagram

5.3. Dynamic model – sequence diagrams

5.4. Rationale for your detailed design model

5.5. Traceability from requirements to detailed design model

6. Test Plan

6.1. Requirements/specifications-based system level test cases

6.2. Traceability of test cases to use cases

6.3. Techniques used for test generation

6.4. Assessment of the goodness of your testsuite

7. Demo

7.1. Database

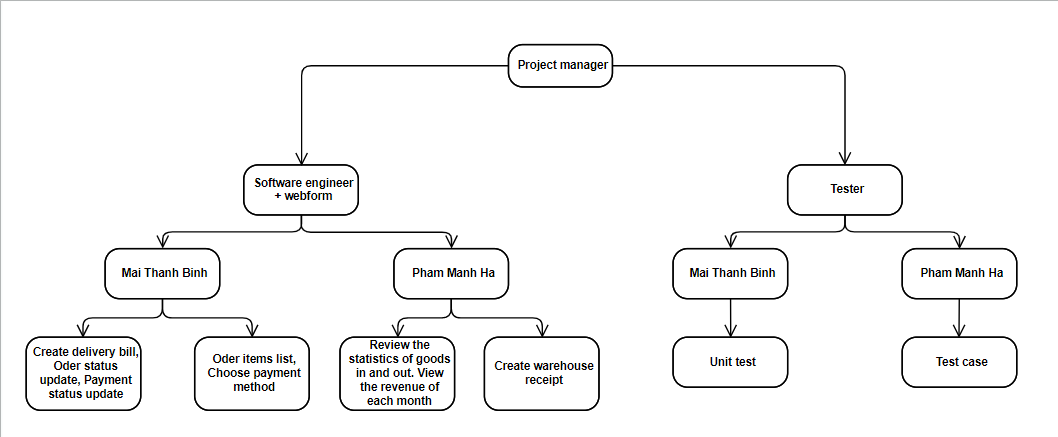
7.2. Source code

7.3. Testing

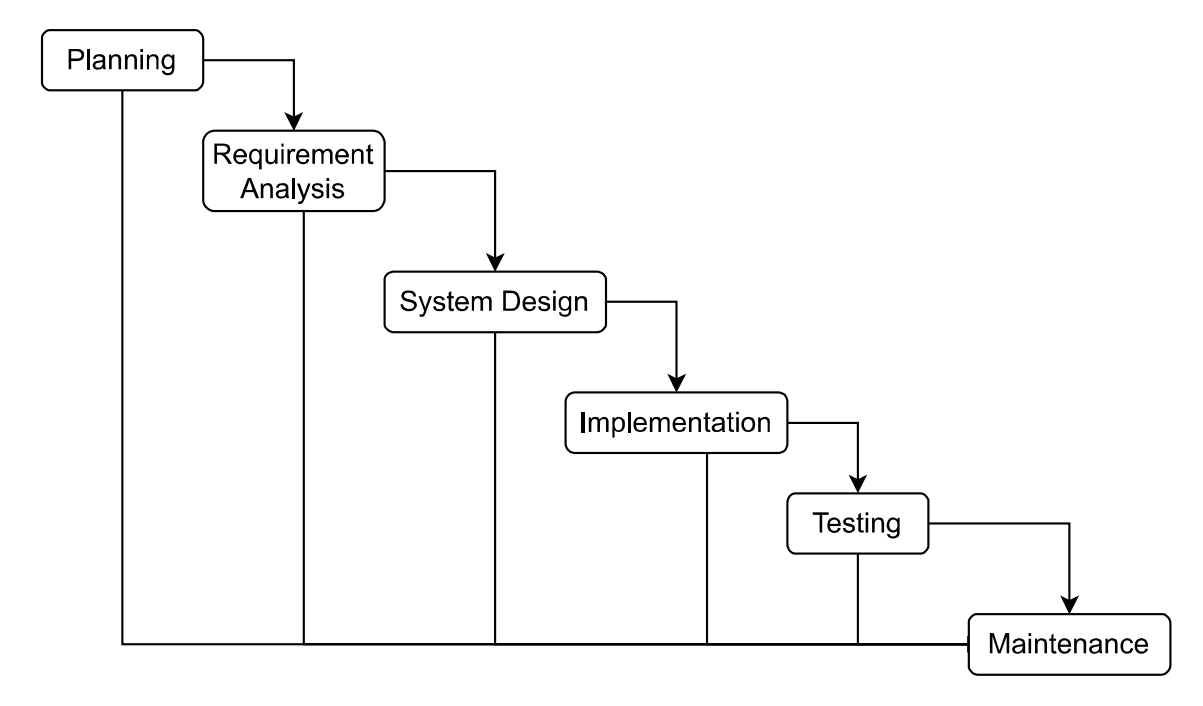
**1.4 Terms, Acronyms, and Abbreviations**

**CHAPTER 2 - PROJECT MANAGEMENT PLAN**

**2.1 Project Organization Structure**

Project Organization ****

**2.2 Lifecycle Model Used**



**2.3 Risk Analysis**

- Leave the input blank

- Duplicate primary key

- Delete foreign key

- Edit foreign key

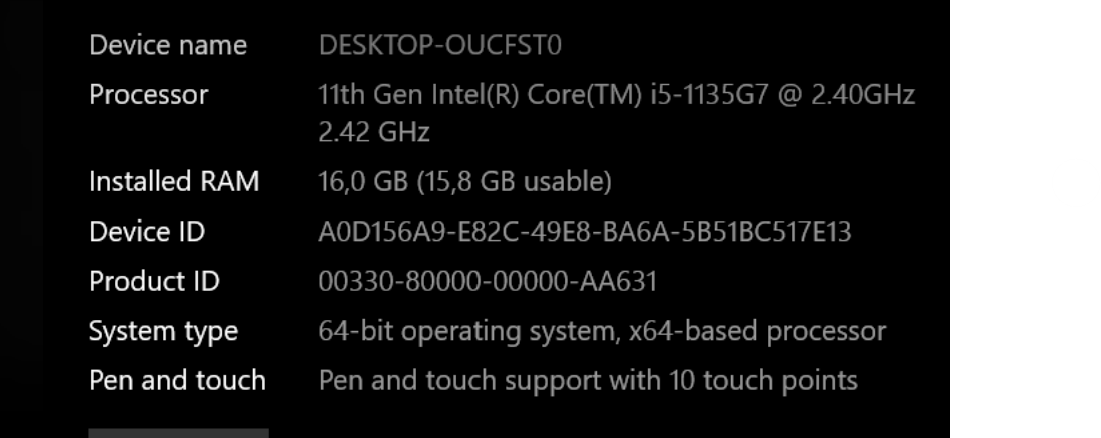
**2.4 Hardware and Software Resource Requirements**

Hardware:

* 1.8 GHz or faster processor. Quad-core or better recommended
* 1.8 GB of RAM; 8 GB of RAM recommended (3.5 GB minimum if running on a virtual machine)
* Hard disk space: Minimum of 800MB up to 210 GB of available space, depending on features installed; typical installations require 20-50 GB of free space.
* Hard disk speed: to improve performance, install Windows and Visual Studio on a solid state drive (SSD).
* Video card that supports a minimum display resolution of 720p (1280 by 720); Visual Studio will work best at a resolution of WXGA (1366 by 768) or higher.

Example:

Ảnh có chứa văn bản

Mô tả được tạo tự động

Software:

- Microsoft SQL Server Management Studio 19

- Visual Studio 2020

- Visual Studio Code

**2.5 Deliverbles and Schedule**

- Delivery schedule will be decided on project discussion date. If the appointment time is exceeded, there is no product or according to the customer’s request. We will take full responsibility.

- Before delivery, we will schedule a demo report date, presenting the project’s completion schedule. Collect comments and edit to suit the requirements of software users.

- By the scheduled delivery date. All functions will be completed. The delivered software will match the customer’s wishes.

- Deadline: 7/5/2023

**2.6 Monitoring, Reporting, and Controlling Mechanisms**

**Monitoring and Controlling**

**Monitor and control project work:** The generic step under which all other monitoring and controlling activities fall under.

**Perform integrated change control:** The functions involved in making changes to the project plan. When changes to the schedule, cost, or any other area of the project management plan are necessary, the program is changed and re-approved by the project sponsor.

**Validate scope:** The activities involved with gaining approval of the project's deliverables.

**Control scope:** Ensuring that the scope of the project does not change and that unauthorized activities are not performed as part of the plan (scope creep).

**Control schedule:** The functions involved with ensuring the project work is performed according to the schedule, and that project deadlines are met.

**Control costs:** The tasks involved with ensuring the project costs stay within the approved budget.

**Control quality:** Ensuring that the quality of the project?s deliverables is to the standard defined in the project management plan.

**Control communications:** Providing for the communication needs of each project stakeholder.

**Control Risks:** Safeguarding the project from unexpected events that negatively impact the project's budget, schedule, stakeholder needs, or any other project success criteria.

**Control procurements:** Ensuring the project's subcontractors and vendors meet the project goals.

**Control stakeholder engagement:** The tasks involved with ensuring that all of the project's stakeholders are left satisfied with the project work.

The report will be completed when the project is completed.

**2.7 Professional Standards**

- Experience in using Windows Form Applications.

- Experience in using Visual Studio.

- Experience in using Microsoft SQL Server Management Studio 19.

- Have SQL database knowledge.

- Have C# knowledge.

**2.8 Evidence all the artifacts have been placed under configuration management**

**2.9 Impact of the project on individuals and organizations**

For individuals:

- Accounting: Increase the speed and reduce the time to manage goods when entering and leaving warehouse. Easy revenue statistics.

- Customers: Convenient to buy goods remotely or get invoices. Easy to choose payment method, transparent.

- Organization: Reduce human waste, modernize management process.

For society:

- Developing the country's economy.

**CHAPTER 3 - REQUIREMENT SPECIFICATIONS**

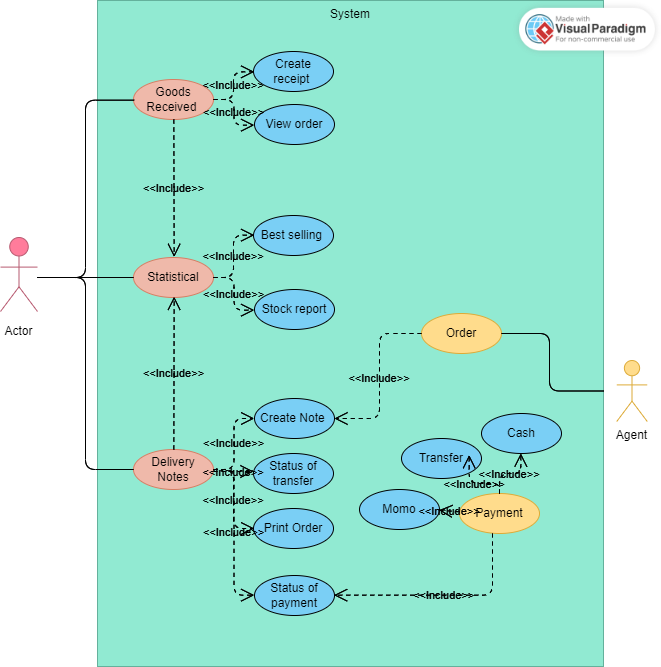
**3.1 Stakeholders for the system**

- Stakeholders for the system are developers, users, customers, management, project management, implementation management, operation, infrastructure management, accounting staff, security staff, hotline support staff.

- While it is not possible to maximize the satisfaction of each individual party, the system will balance the parties to maximize the overall satisfaction of all stakeholders.

**3.2 Use case model**

***3.2.1 Graphical use case model***

******

### *3.2.2 Use case specification*

**Use case for Goods Received**

|  |  |
| --- | --- |
| Usecase name | Goods Received |
| Context | Directly at the store’s warehouse |
| Trigger | Select “Manage goods import” -> Select “Create purchase order” |
| Description | Usecase is used to creat receipt information when there is a new batch of goods and save it in the database. |
| Actor | Accounting |
| Related | Login |
| Stakeholders | Manage |
| Pre - Condition | Login with access right according to the position of accountant |
| Post - Condition | Update information in the database |
| Basic Flow | |  |  | | --- | --- | | Agent | System | | 1. Log in to the system. 2. Select “Manage goods import” -> Select “Receipt of goods” 3. The accouting enter the goods receipt information and click “ Save” | 1. Display the form enter the information of the receipt 2. Database update system. | |
| Result | Goods receipt created successfull |
| Exception | None |

**Use case for Delivery bill**

|  |  |
| --- | --- |
| Usecase name | Delivery bill |
| Context | Directly at the store’s warehouse |
| Trigger | Select “Manage goods export” -> Select “Create delivery note” |
| Description | Usecase is used to create delivery slip information when there is a shipment and save it in the database |
| Actor | Accounting |
| Related | Login |
| Stakeholders | Manage |
| Pre - Condition | Login with access right according to the position of accountant |
| Post - Condition | Update information in the database |
| Basic Flow | |  |  | | --- | --- | | Agent | System | | 1. Log in to the system. 2. Select “Manage goods export” -> Select “Delivery note” 3. The accouting enter the goods export information and click “ Save” | 1. Display the form export the information of the receipt 2. Database update system. 3. Update order status and payment | |
| Result | Create delivery note successfull |
| Exception | None |

**Use case for Statistical management**

|  |  |
| --- | --- |
| Usecase name | Statistical management |
| Context | Manage operations directly on the system |
| Trigger | Select “Statistics” -> Select the statistics sheet -> select “View statistics sheet” |
| Description | Usecase used to view statistics sheet |
| Actor | Accounting |
| Related | Login |
| Pre - Condition | -Login with access right according to the position of accountant  -There is at least one statistical vote in the list |
| Post - Condition | View information about statistics |
| Basic Flow | |  |  | | --- | --- | | Agent | System | | 1. Login to the system. 2. Select “Statistics” 3. Select the statistics sheet you want to see then select “View statistics sheet” | 1. Check login permission 2. Display “Statistics” interface 3. Display information of the selected statistics sheet. | |
| Exception | None |

**Use case for Order**

|  |  |
| --- | --- |
| Usecase name | Order Information |
| Context | Agency order at the company’s website |
| Trigger | Select “Order” -> Select “Create Order” |
| Description | Usecase is used for ordering |
| Actor | Agency |
| Related | Login |
| Stakeholders | Company |
| Pre – Condition | -Login to the system with agency access  -Create order |
| Post – Condition | Update information in the database |
| Basic Flow | |  |  | | --- | --- | | Agent | System | | 1. Log in to the system 2. Select “Orders” -> Select “Create Order” 3. Enter order information 4. Click “Save” -> Order | 1. The system updates the database 2. Display order on the company’s order management system | |
| Result | Order successfull |
| Exception | None |

**Use case for Payment**

|  |  |
| --- | --- |
| Usecase name | Payment |
| Context | Agency payment at the company’s website |
| Trigger | Select “Payment” -> Choose payment method |
| Description | Usecase is used for payment |
| Actor | Agency |
| Related | Login |
| Stakeholders | Accounting |
| Pre - Condition | -Login to the system with agency access  -Agency order has been successfully placed |
| Post - Condition | Update information in the database |
| Basic Flow | |  |  | | --- | --- | | Agent | System | | 1. Log in to the system. 2. Select “Payment” -> Select payment method 3. Agency choose payment method and “pay” | 1. Show form paid 2. Database update system. | |
| Result | Payment successfull |
| Exception | None |

## 3.3 Functional requirements

- User should be able to enter Sales Data.

- Sales Reports should be genarated every 24 hours.

- Good service system and works according to the wished of the user.

- Details of operations conducted in every screen.

- Data handing logic should be entered into the system.

- It should have descriptions of system reports or other outputs.

- Complete information about the workflows performed by the system.

- It should clearly define who will be allowed to create/ modify/ delete the data in the system.

- How the system will fulfill applicable regulatory and compliance needs should be captured in the functional document.

- Goods Received:

+ Save goods information: product name, price, date of import, ...

+ Can edit, update, delete types of items if entered incorrectly.

+ The function of searching for imported items.

- Delivery bill:

+ Save export information.

+ Print the export slip.

+ Can add, delete, edit orders.

+ Always update the status of orders: moving, where, payment status.

+ Search for exported goods: can display the total and separately.

- Payment management:

+ After being paid, calculate and check the balance, automatically save and have notifications about the system.

- Management and statistics of goods, revenue of each month:

+ Each item when sold will be saved and calculated by the system. It is possible to count each type of item or all items sold during the month.

+ Monthly revenue will be calculated by the system, in addition to being able to calculate the revenue of each type of item.

+ Can add, delete, edit quantity if necessary.

## Non – functional requirements

- Good service system, capable of working well 24/24 hours and 7 days a week.

- Make sure the system is ready to be used: Time can be used by the system and the dependent elements to operate the system.

- There is periodic maintenance with each system upgrade that will not last more than 30 minutes.

- The system must ask for confirmation when there are data deletion operations.

- The product must comply with the provisions of law, state, company, financial law, ...

- The scalability of the system:

+ Server has the ability to upgrade configuration.

+ It is possible to separate the database on a separate server and backend on a separate server.

- The features of import, export, payment, statistics of goods, revenue of each month must be fast and accurate, ensuring security. Avoid the case of leaked information affecting the company.

# CHAPTER 4 - ARCHITECTURE

## 4.1 Architectural style used

**Architectural model use: Layered Architecture**

- The layered architecture pattern closely matches the traditional IT.

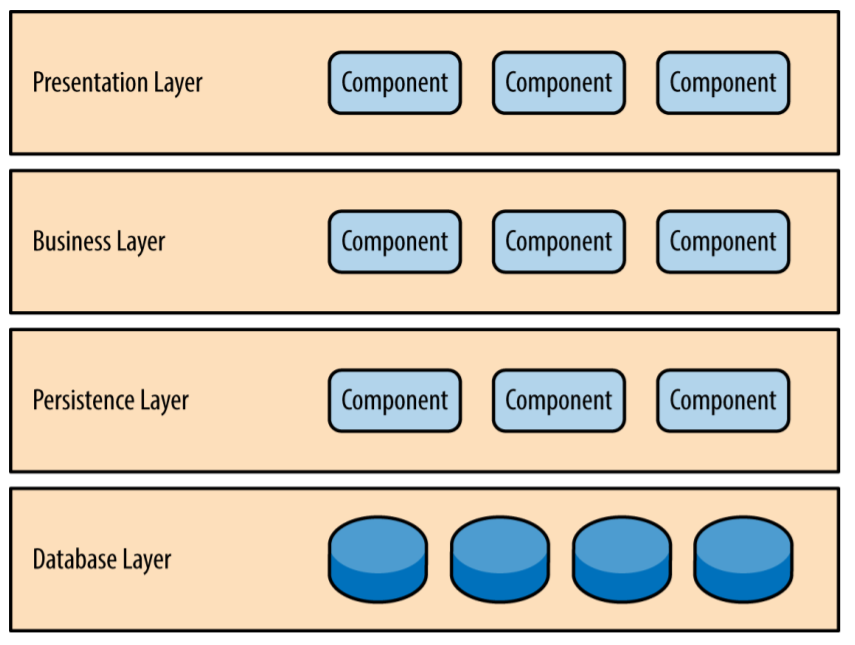
- The different layers are defined in the architecture. It consists of outer and inner layer.

- The components of outer layer manage the user interface operations (Presentation Layer).

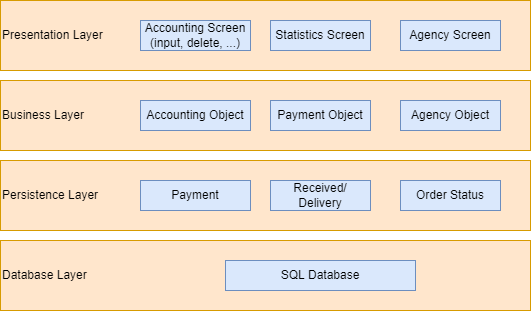
- Components excute the operating system interfacing at the inner layer (Accounting Screen, Agency Screen, Statistics Screen).

- The inner layers are application layer, utility layer and the core layer (Payment, Delivery bill, Goods Received, Database).

- In many cases, it is possible that more than one pattern is suitable and the alternate architectural style can be designed and evaluated.



## 4.2 Architectural model

****

## 4.3 Technology, software, and hardware used

**Technology**

- SQL Server

- C#

- Github

- ASP.NET

**Software**

-Microsoft SQL Server Management Studio 19.

-Visual Studio 2020.

-Visual Studio Code.

**Hardware**

Ảnh có chứa văn bản

Mô tả được tạo tự động



## 4.4 Rationale architectural style and model

- Layer Architecture allow we just need to understand what classes below the class are doing.

- Each class can be replaced by an equivalent class without affecting the other. A class can be used by a number of different senior classes.

**According to the model above:**

+ Presentation Layer: allows users to see the interface and basic operations such as adding, deleting, editing, order ...

+ Business Layer: shows that the people who have the right to try to access and use the system (Accounting Object, Agents Object, Payment Object).

+ Persistence Layer: intermediates between the functions of the system and the stored database (Payment, Order Status, Received/ Delivery).

+ Database Layer: Data storage layer (SQL Databse).

**Advantages of Layered architecture:**

+ Layered architecture increases flexibility, maintainability, and scalability

+ Multiple applications can reuse the components.

+ Layered architecture enables teams to work on different parts of the application parallely with minimal dependencies on other teams.

+ Layered architecture enables develop loosely coupled systems.

+ Different components of the application can be independently deployed, maintained, and updated, on different time schedules.

**Disadvantages of Layered architecture:**

+ Lack of inbuilt scalability.

+ Hidden use cases.

+ No dependency inversion.

**Testability:** Because components belong to specific layers in the architecture, other layers can be mocked or stubbed, making this pattern is relatively easy to test. A developer can mock a presentation component or screen to isolate testing within a business component, as well as mock the business layer to test certain screen functionality.

**Ease of development:** Ease of development gets evaluate high, mostly because this pattern is so well known and is not overly complex to implement. Because most companies develop appli‐ cations by separating skill sets by layers (presentation, business, database), this pattern becomes a natural choice for most business-application development.