Version <1.0>

[The texts in blue are guidance for filling in the information for each section. Remove everything in blue when writing the document.

This template is a simplified version of the Software Architecture Document from the RUP model. ]

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 23/11/2019 | 1.0 |  | <name> |
| 5/12/2019 | 1.5 | Deployment View | Nguyen Ly Nhat Phuong |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 4

2. Architectural Goals and Constraints 4

3. Use-Case Model 4

4. Logical View 4

4.1 Component: abc 4

5. Deployment 4

6. Implementation View 4

# Introduction

[The introduction of the **Software Architecture Document** provides an overview of the entire **Software Architecture Document**. It includes the purpose, scope, definitions, acronyms, abbreviations, references, and overview of the **Software Architecture Document**.]

# Architectural Goals and Constraints

[This section describes the software requirements and objectives that have some significant impact on the architecture; for example, safety, security, privacy, use of an off-the-shelf product, portability, distribution, and reuse. It also captures the special constraints that may apply: design and implementation strategy, development tools, team structure, schedule, legacy code, and so on.]

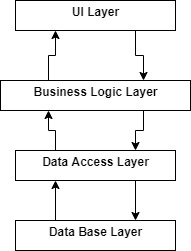
# Use-Case Model

[This section includes the use case diagrams that are already modeled and presented in the use-case specification document.]

# Logical View

The logical view of the Convenience Store Sale Management Software follow the layered architecture with 4 main layer:

* **UI Layer**: Contains all the boundary classes that represent the software screens that the users use to communicate with the software.
* **Business Logic Layer**: Contains class that supports logic processing. It process user input from UI and transfer to the Data Access Layer. Process the data receive from the Data Access Layer and transfer it to UI.
* **Data Access Layer**: Contains class that supports access to the data base. It receive the statement from the Business Logic Layer and access to the data base toget the data. Transfer the data get from the data base to the Business Logic Layer.
* **Database Layer:** Contains the data base of the store.



## Component: UI Layer

A close up of a map

Description automatically generated

* Login class: Login screen class, supports login for all users.
* ChangePassword class: Change password screen, supports all users changing password.
* MainMenu class: Abstract class for all users main menu classes.
* StaffMenu class, WarehouseStaffMenu, AdministratorMenu: Main menu screen for staffs, warehouse staff, administrator. Displays menu to choice which action users want to do.
* MemberMenu class: Menu displays for staffs and administrator all action can do on the members.
* BillMenu class: Menu displays for staffs and administrator all action can do on the bills.
* ProductMenu class: Displays screen support search and watch product information for all user.
* InventoryMenu class: Display screen supports warehouse staff choose to check inventory, to import or to export product.
* StaffMenu: Displays screen support administrator to add a staff, remove a staff or search for a staff.
* StatisticMenu: Diaplays screen support administrator to make statistic report in many type.

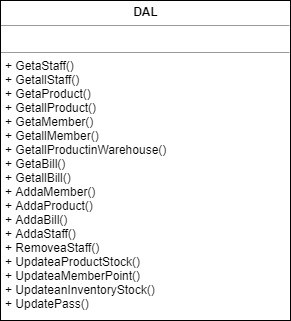
## Component: Business Logic Layer

A screenshot of a cell phone

Description automatically generated

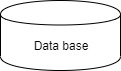
* BLL class: Supports logical process in the software like process input from UI Layer for Data Access Layer or process data from Data Access Layer for UI.

## Component: Data Access Layer

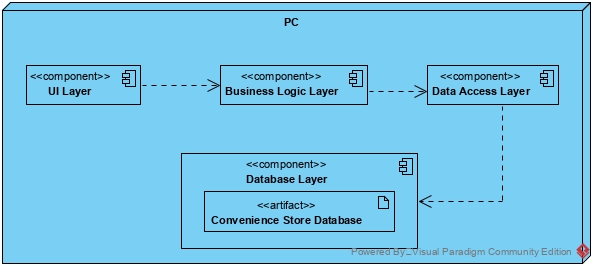


* DAL class: Supports access to the data base to get data insert data, delete data, update data on the databse follow the require from the Business Logic Layer.

## Component: Data Base Layer



# Deployment



**PC:** Staffs, warehouse staff, administrator using the store PC to manage almost processes in the the store.

# Implementation View

[Leave this section blank for PA3.]