POS SYSTEM – ARCHITECTURE DRIVERS



HIT Team

Consulting

Sales

Staffing

Support

# Information of document

|  |  |
| --- | --- |
| **Title** | **Architecture Driver Document** |
| **Author(s)** | All team |
| **Reviewer(s)** | All team |
| **Team name** | HIT Team |
| **Team members** | Thanh Giang, Hiep Ta, Phuc Nguyen, Giang Nguyen, Dat Tran, Huy Huynh |
| **Project mentors** | Ms. Huong Nguyen, Mr. Huyen Pham |
| **Editor** |  |
| **Type of report** | Architecture Driver Document |
| **Software used** | MS Word |

# Document Reviewer Information

|  |  |  |
| --- | --- | --- |
| Reviewer Name | Review Attendance (R/S) | Comments |
|  |  |  |
|  |  |  |
|  |  |  |

# Document Approver Information

|  |  |  |
| --- | --- | --- |
| Approver Name | Approver Function | Comments |
|  |  |  |
|  |  |  |
|  |  |  |

# Document Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Revision | Status | Change Summary | Revised by |
| 6/3/2012 | 1.0 |  | Consume Team member’s tasks | Thanh Giang |
| 15/4/202 | 1.1 |  | Update Quality Attributes | Giang Nguyen |
|  |  |  |  |  |
|  |  |  |  |  |

1. **INTRODUCTION** 
   1. **Purpose of this document**

This section of the Architectural Driver gives an overview of the business context and the architectural drivers with their impact on the project. It also contains the project deliverables, the summary of the schedule the SPMP. We also show the development strategy that we choose to show how this may affect the design of a system.

* 1. **Project Summary**

* 1. **Document's intended audience**

|  |  |
| --- | --- |
| Intended Audience | Reading Suggestions |
| Project Manager | Section 2 – The architectural Drivers: List functions showed by use-case diagrams and constrains to make the Project Manager has an overview. So he can have the estimates for the project.  Section 3 – The development Strategy |
| Software Architecture and Designer | Section 2 – The architectural Drivers: This section describes Use-case diagram and Use-case descriptions. It makes easily to design and develop the proposed system. |
| Tester | The Overview section and Use-case: they will help to make the test plan and write the acceptance test |

1. **The architectural drivers**
   1. **High-level Functional Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID | Use Case Name | Important Level | Difficulty Level |
| System Management | | | |
| UC\_SM01 | Add New User | High | Medium |
| UC\_SM02 | Search/ View User List | Medium | Low |
| UC\_SM03 | View User Log | Low | High |
| UC\_SM04 | View User Detail Information | Normal | Low |
| UC\_SM05 | Assign Authorize | High | High |
| UC\_SM06 | Update User Information | Normal | Medium |
| Product Management | | | |
| UC\_PM01 | Add New Product | High | Medium |
| UC\_PM02 | Search/ View Product List | Medium | Low |
| UC\_PM03 | View Product Detail Information | Medium | Low |
| UC\_PM05 | Update Product Information | Medium | Medium |
| Bill Management | | | |
| UC\_RM01 | Add New Bill | High | High |
| UC\_RM02 | Search/ View Bill List | Medium | Low |
| UC\_RM03 | View Bill Detail Information | Medium | Low |
| UC\_RM04 | Print Bill | Medium | Medium |
| Retail Stores Management | | | |
| UC\_RSM01 | Add New Store | High | Medium |
| UC\_RSM02 | Search/ View Store List | Medium | Low |
| UC\_RSM03 | View Store Detail Information | Medium | Low |
| UC\_RSM04 | Update Store Information | Medium | Medium |
| Category Management | | | |
| UC\_CM01 | Add New Category | High | Medium |
| UC\_CM02 | Search/ View Category List | Medium | Low |
| UC\_CM03 | View Category Detail Information | Medium | Low |
| UC\_CM04 | Update Category Information | Medium | Medium |
| Customer Management | | | |
| UC\_C01 | Add New Customer | High | Medium |
| UC\_C02 | Search/ View Customer List | Medium | Low |
| UC\_C03 | View Customer Detail Information | Medium | Low |
| UC\_C04 | Update Customer Information | Medium | Medium |
| UC\_C05 | View Customer Point Log | High | Medium |
| Statistic | | | |
| UC\_S01 | Analysis Statistic | Low | High |
| POS Management | | | |
| UC\_P01 | Add New POS | High | Medium |
| UC\_P02 | Search/ View POS List | Medium | Low |
| UC\_P03 | View POS Detail Information | Medium | Low |
| UC\_P04 | Update POS Information | Medium | Medium |
| View Point | | | |
| UC\_VP | View Point | High | High |

* + 1. **Use-case Diagrams and Use-case Descriptions**

|  |  |
| --- | --- |
| **Actor** | **Description** |
| Administrator | Responsible for manage user of the system such as: Create new, assign authorize. He can also sync information between |
| Manager | Responsible for manage information of products, categories, customer, retail stores, POS and they can statistic sales by many criterion |
| Cashier | Responsible for check bills (bills) |
| User | Includes Manager, cashier and admin: they can use basic function like: log in, logout, change password |

***Note Table:***





**Level 1 – Retail Sales System**



**Level 2 - System Management**



**Level 2 - Customer Management**



**Level 2 - Product Management**



**Level 2 - Category Management**



**Level 2 - Bill Management**



**Level 2 - Retail Stores Management**



**Level 2 – POS Management**



* + 1. **Use-case Descriptions**

|  |  |
| --- | --- |
| **Use Case Title:** Add New Store | **Use Case ID:** UC\_RSM01 |
| **General Use Case Description:** This use case help the manager add new sale store. This function can also split products of retail stores that are consumed, as well as regulate the product price. | |
| **Entities Involved:** Manager, Retail Store | |
| **Preconditions:**   * User is assigned authorized to use this function * User has chosen Retail Stores Management function * The system is available | |
| **Primary Use Case Flow of Events:**   1. User choose “Add new store” button 2. Program displays Add new store interface 3. User fills in Store name text field 4. User fills in Address text field 5. User chooses a Category 6. Program displays Products which has in that Category 7. User chooses a Product 8. Program add the product which user just has chosen to the Product List with the standard Price 9. User can edit the Price base on how much the store want to sell that product. 10. User clicks "OK". 11. Program shows message "Are you sure you want to create new retail store’s information as above?" 12. User clicks "OK" to confirm 13. The system stores the retail store’s information that has just been created and returned to retail store management interface 14. End Use-case | |
| **Primary Use Case Post Conditions:**  Successful: Create new successful. The system stores the retail store’s information that has just been created and returned to retail store management interface  Fail: Failed to create new | |
| **Alternate Use Case #1 Flow of Events:**  There is a store with already name (starting from step 11 of the main flow)   1. Program shows message "Are you sure you want to create new retail store’s information as above?" 2. User clicks "OK" to confirm 3. Program check in database and found that it has a store with already name 4. Program display warning dialog “There is a store with already name. Please fill in again” 5. End Use-case | |
| **Alternate Use Case #1 Post Events:** | |

|  |  |
| --- | --- |
| **Use Case Title:** Sync Information | **Use Case ID:** UC\_SM01 |
| **General Use Case Description:** This use case help the administrator synchronize information between retail stores and head office: mostly customer’s points needs to be updated to be able to use immediately the next day | |
| **Entities Involved:** Administrator | |
| **Preconditions:**   * User use this function at the end of the day when all sessions are finished * The system is available | |
| **Primary Use Case Flow of Events:**   1. User choose “Sync Information” button 2. Program displays Sync Information interface 3. User choose the retail store want to sync with the head office 4. User choose type what to sync: all or just customer point 5. User choose “Sync” button 6. Program syncs information between the head office with those retail stores 7. Program shows message “Sync Information Successful” 8. End Use-case | |
| **Primary Use Case Post Conditions:**  Successful: Sync Successful  Fail: Failed to sync OK | |
| **Alternate Use Case #1 Flow of Events:**  Can’t access the retail store’s database (starting from step 6 of the main flow)   1. Program syncs information between the head office with those retail stores 2. Program shows message “Sync Information Successful” 3. End Use-case | |
| **Alternate Use Case #1 Post Events:** | |

|  |  |
| --- | --- |
| **Use Case Title:** Add New Bill | **Use Case ID:** UC\_RM01 |
| **General Use Case Description:** This use case helps cashiers who work at retail stores make bill-paying for customers. Cashier can use barcode reader or directly enter the product code and product number by keyboard, then use the payment function to save to database and printed out bills for customers. | |
| **Entities Involved:** Cashier, Retail Store | |
| **Preconditions:**   * User is assigned authorized to use this function * User has chosen Bill Management function * The system is available. | |
| **Primary Use Case Flow of Events:**   1. User choose “Add new bill” button 2. Program displays Add new store interface 3. User fills in Bar Code text field 4. User choose “Add” button 5. The system load information from database to the “Product List” table on the interface. 6. User fills in Quantity text field 7. User scan or fill in customer Loyalty card ID 8. The system load information from database and show customer name with their point on the interface 9. If customer want to use their point to pay bill, User choose “Type to pay bill” 10. There are three ways to pay bill:     1. Pay by cash: User does not fill in “Point” text field. The system auto shows all the money customer has to pay in “Cash” text field.     2. Pay by point: User fills in “Point” text field if user has enough point to pay that bill, the system auto shows “0 VND” in “Cash” text field.     3. Pay by point and cash: User fills in “Point” text field, the system shows money that customer still has to pay with that bill in “Cash” text field. 11. User choose “Pay by Cash” 12. User choose Pay bill button 13. The system stores the bill’s information that has just been created and prints the bill. Then the program returns to bill management interface 14. End Use-case | |
| **Primary Use Case Post Conditions:**  Successful: Create new successful. The system stores the bill’s information that has just been created and prints the bill. Then the program returns to bill management interface  Fail: Can’t print the bill | |
| **Alternate Use Case #1 Flow of Events:**  The bar code is invalid (starting from step 5 of the main flow)   1. Program loads information from database and shows message “The bar code is invalid” 2. User clicks "OK" to confirm 3. Program allows user to retype the bar code 4. End Use-case | |
| **Alternate Use Case #1 Post Events:** | |

* 1. **Constrains:**
     1. **Business Constrains:**

|  |  |  |
| --- | --- | --- |
| Consideration | ID | Business Constraints. |
| Schedule limitations. | BC01 | 1 Project Management – 120h (4h/day) |
| 2 Programmer – 120h (4h/day) |
| 1 Architect – 90h (3h/day) |
| 2 Tester – 90h (3h/day) |
| 1 Requirement -180h (6h/day) |
| Mandatory regulatory restrictions and demands. | BC02 | Customers who have become point service members are issued point cards. Points can be used in all stores.  The member the either pays with cash, points, or a combination of the two |
| Market restrictions and demands. | BC03 | The actual retail price must be set in advance. |
| Organizational restrictions and demands. | BC04 | One team with 6 members |

* + 1. **Technical Constrains:**

|  |  |  |
| --- | --- | --- |
| Consideration. | ID | Technical Constraints |
| Peripheral or network hardware. | TC01 | Database server |
| Commercial hardware or software products. | TC02 | Bar code readers  Keyboard  Computer |
| Tools and methods. | TC03 | SQL Database Server |
| Protocols, interfaces, standards. | TC04 | TCP/IP protocol |

* 1. **Quality Attributes**
* **Quality Attribute Ranking Table**

|  |  |  |  |
| --- | --- | --- | --- |
| Quality Attribute | Important (base on customer) | Difficult level (to implement) | Priority |
| Performance | **High** | **High** | **High** |
| Security | **High** | **High** | **High** |
| Availability | **Medium** | **Medium** | **Medium** |
| Usability | **Medium** | **Low** | **Low** |

* **List of quality attributes**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Quality** | **Quality ID** | **Concern** | **Use case** | **Attribute** |
| **Performance** | QP01 | Response time | UC\_SM02 - Search/ View User List  UC\_SM03 - View User Log  UC\_SM04 - View User Detail Information  UC\_PM02 - Search/ View Product List  UC\_PM03 - View Product Detail Information  UC\_PM04 - Choose Category  UC\_RM02 - Search/ View Bill List  UC\_RM03 - View Bill Detail Information  UC\_RM04 - Print Bill  UC\_RSM02 - Search/ View Store List  UC\_RSM03 - View Store Detail Information  UC\_CM02 - Search/ View Category List  UC\_CM03 - View Category Detail Information  UC\_C02 - Search/ View Customer List  UC\_C03 - View Customer Detail Information  UC\_P03 – Search/ View POS Detail Information  UC\_C04 - Update Customer Information  UC\_C05 - View Customer Point Log  UC\_CM01 - View Point | System will return results within 0.5s if there is any interaction with it (search, view…) |
| QP02 | Delay time | UC\_SM05 - Assign Authorize  UC\_SM06 - Update User Information  UC\_SM07 - Sync Information  UC\_PM05 - Update Product Information  UC\_RSM04 - Update Store Information  UC\_CM04 - Update Category Information  UC\_CM04 - Update Category Information  UC\_T01 - Update Information  UC\_P04 – Update POS Information | For interactive into systems requires more time to handling, these transactions are processed and with an average latency of two seconds. |
| **Availability** | QA01 | Easy to configure | UC\_SM01 - Add New User  UC\_PM01 - Add New Product  UC\_RM01 - Add New Bill  UC\_RSM01 - Add New Store  UC\_CM01 - Add New Category  UC\_C01 - Add New Customer  UC\_P01 – Add new POS  UC\_SM05 - Assign Authorize | The user interface is easy for users to add new anything in the system. |
| **Security** | QS01 | Security customer’s information | UC\_S01- Statistic Information | Customer’s information will be protection with high level |

* + 1. **Performance:**

This is a retail system, so the performance we discuss in here is about timing. It means that we must always ensure that customer service was not delayed. Intervals such as scan the point card, bar code scanning, displaying results... will be done in the shortest time possible to avoid the dissatisfy customer. Things that need attention are: increased client response time, reduced throughput, and server resource over utilization. Ensure that you structure the application in an appropriate way and deploy it onto a system or systems that provide sufficient resources.

**Example of Performance Attribute:**

**Example** 1: Users initiate 100 transactions per minute at 100 POS terminal under overload operations, and these transactions are processed with an average latency of two seconds.

|  |  |
| --- | --- |
| **Portion of scenario** | **Possible values** |
| **Source** | User |
| **Stimulus** | 100 transactions per minute at 100 POS terminal |
| **Artifact** | System |
| **Environment** | Overload operation |
| **Response** | These transactions are processed |
| **Response Measure** | 2 seconds |

**Example** 2: Users scan the point card under normal operating conditions and the result will be show with an average latency of 0.5 seconds

|  |  |
| --- | --- |
| **Portion of scenario** | **Possible values** |
| **Source** | User |
| **Stimulus** | Scan the point card |
| **Artifact** | System |
| **Environment** | Normal operating conditions |
| **Response** | The result will be show |
| **Response Measure** | 0.5 seconds |

* + 1. **Availability:**

This system has two servers located at the head office server and POS terminal, if the failure occurs at head office server or the network; we need the system still available and working normal at the POS terminal.

**Example of Availability Attribute**: The temporary failure of the head office server, the POS terminal will be still working securely. The POST terminals can carry out the sales operation efficiently using locally stored data as much as possible.

|  |  |
| --- | --- |
| **Portion of scenario** | **Possible values** |
| **Source** | Head office database |
| **Stimulus** | Temporary failure of the head office server |
| **Artifact** | System |
| **Environment** | Normal operation |
| **Response** | To be still working securely and carry out the sales operation efficiently using locally stored data. |
| **Response Measure** | No down time |

## -- The End --