# ARCHITECTURE BACKGROUND – MODULE VIEW

Why decision design meets the requirement will be explained as below:

* A product have a specific point that will be awarded when the member purchase products with cash.
* Users can enter these codes by using computer keyboard if they don’t have any barcode reader.
* The loyal member either pays with cash, points, or a combination of the two on Sale UI.
* When the points are used, the number of points used is immediately subtracted from the number of points accrued by the member
* Products are classified into categories (kind of product types) such as food, general merchandise. And user can manage categories
* For each product, its standard price, common to all stores, is set as a part of the product data. Each store, however, can set and use its own actual retail price instead of the standard price. The actual retail price must be set in advance. User can

# ARCHIETCTURE BACKGROUND – C&C VIEW

The application divided into 3 tiers: client tier, server tier, database tier.

On database tier, we have a cluster contain two databases and those databases will sync with each other in a period of time. If one of those databases is crash, another one is active, ensuring available for system.

The reason that our design meets quality attributes that will be explained in the following table

|  |  |  |  |
| --- | --- | --- | --- |
| **Quality** | **Quality ID** | **Concern** | **Response measure** |
| Performance | QA\_P01 | Display product’s information | The system will display product’s information within 1 second |
| **Rationale**: When the member pay a bill, the cashier scanning products code, the system will display product’s information within 1 second | | | |
| Performance | QA\_P02 | Displays statistical reports | The system displays information reported within 5 seconds |
| **Rationale:** To know a business status, the manager performs the statistical reports, the system displays information reported within 5 seconds | | | |
| Performance | QA\_P03 | Displays Member’s information | The system displays the Member information within 2 s |
| **Rationale**: When the member goes to the store, choose some product and go to the POS to pay a bill. The cashier scans the Member's card; the system displays the Member information within 2s | | | |
| Performance | QA\_P04 | Making a bill | the system is operating normally, system confirm request, save billing information in the database and notify successful payment within 2 s |
| **Rationale:** The cashier sends confirmation of payment request, system confirm request, save billing information in the database and notify successful payment within 2s | | | |
| Availability | QA\_A01 | Database | The system will accept payment requests, stores the information in the database backup and respond to machines in retail store with no downtime |
| **Rationale:** Computer in retail store send a request for bill payment to server in head office while the database has been crash. The system will accept payment requests, stores the information in the database backup and respond to machines in retail store with no downtime | | | |

|  |  |
| --- | --- |
| Connector | Properties |
| Request/ Reply | Connector between client and server style, used by a client to invoke services on a server. |
| Call and return | Responsible for conveying the service request from the requester to the provider and for returning any results. Use by interface to request data from Filter/Object |
| Synchronous Replication | The technique for replicating data by two or more databases (or file systems) where the system being replicated does waits for the data to have been recorded on the duplicate system before proceeding. |
| ODBC | Stands for Open Database Connectivity. It is the standard method which allows any application to connect data. ODBC uses a middle layer called the database driver to handle the connection in between the application and the relational database management system. |