**Final Year Project Report**

**QuickStock**

****

**Project Advisor:**

**Sir Noaman Saleem**

**Submitted By:**

**Amaan Majeed (F2020266286)**

**Faseeh-ud-din (F2020266296)**

**Muhammad Hashim Khurshid (F2020266272)**

**Muhammad Usaid Afzal (F2020266271)**

**Munib Ahsan Khan (F2020266270)**

**Session**

**University of Management and Technology**

**C-II Johar Town Lahore Pakistan**

**Dedication**

## 

**Final Approval**

* **Head of Department**  Dr. Atif Alvi

Department of Informatics & Systems

School of Systems & Technology

UMT Lahore

* **Director (Final Year Projects-CS)** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Department of Computer Science.

School of Systems & Technology

UMT Lahore

* **Supervisor** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* Department of Informatics & Systems.
* School of Systems & Technology

UMT Lahore

* **Co-Supervisor** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Acknowledgment**

**Project Title: Quickstock**

**Objective**

**Undertaken by**

**Supervised by**

**Starting Date**

**Completion Date**

**Tools Used**

**Operating System**

**Documentation**

**Plagairism ReportAbstract**

Revision Chart

*This chart contains a history of this document’s revisions. The entries below are provided solely for illustration purposes. Those entries should be deleted until the revision/s they refer to have actually been created.*

*The document itself should be stored in revision control, and a brief description of each version should be entered in the Revision Control System. A brief description can be repeated in this section. Revisions need not be described elsewhere in the document, unless they explain the document.*

| **Version** | **Primary Author(s)** | **Description of Version** | **Date Completed** |
| --- | --- | --- | --- |
| *Draft* | TBD | Initial draft created for distribution and review comments | (To be decided) TBD |
| *Preliminary* | TBD | Second draft incorporating initial review comments, distributed for final review | TBD |
| *Final* | TBD | First complete draft, which is placed under change control | TBD |
| *Revision 1* | TBD | Revised draft, revised according to the change control process and maintained under change control | TBD |
| *Revision 2* | TBD | Revised draft, revised according to the change control process and maintained under change control | TBD |
| *Etc.* | TBD | TBD | TBD |

*Table 1: Revision Chart*

# Contents

*New paragraphs formatted as Heading 1, Heading 2, and Heading 3 will be added to the table automatically. To update this table of contents in Microsoft Word, put the cursor anywhere in the table and press F9. If you want the table to be easy to maintain, do not change it manually.*

[**Contents 1**](#_heading=h.17dp8vu)

[Definitions and Acronyms 3](#_heading=h.3rdcrjn)

[List of Figures 4](#_heading=h.lnxbz9)

[List of Tables 5](#_heading=h.35nkun2)

[**1. Introduction 7**](#_heading=h.1ksv4uv)

[1.1 Motivations 7](#_heading=h.44sinio)

[1.2 Project Overview 7](#_heading=h.2jxsxqh)

[1.3 Problem Statement 7](#_heading=h.z337ya)

[1.4 Objectives 7](#_heading=h.3j2qqm3)

[**2. Domain Analysis 9**](#_heading=h.1y810tw)

[2.1 Customer 9](#_heading=h.4i7ojhp)

[2.2 Stakeholders 9](#_heading=h.2xcytpi)

[2.3 Affected Groups with social or economic impact 10](#_heading=h.3whwml4)

[2.4 Dependencies/ External Systems 10](#_heading=h.2bn6wsx)

[2.5 Reference Documents 10](#_heading=h.qsh70q)

[*2.5.1 Related Projects 10*](#_heading=h.3as4poj)

[*2.5.2 Feature Comparison 11*](#_heading=h.1pxezwc)

[**3. Requirements analysis 12**](#_heading=h.49x2ik5)

[3.1 Requirements 12](#_heading=h.2p2csry)

[3.2 List of Actors 13](#_heading=h.147n2zr)

[3.3 List of use cases 13](#_heading=h.3o7alnk)

[3.4 System use case diagram 14](#_heading=h.23ckvvd)

[3.5 Extended use cases 15](#_heading=h.32hioqz)

[3.6 User interfaces (mock screens) 17](#_heading=h.1hmsyys)

[**4. Data flow diagram (optional) 18**](#_heading=h.2grqrue)

[4.1 Data Flow Diagram Level 0 18](#_heading=h.vx1227)

[4.2 Data Flow Diagram Level 1 18](#_heading=h.3fwokq0)

[4.3 Data Flow Diagram Level 2 18](#_heading=h.1v1yuxt)

[**5. System Design 20**](#_heading=h.4f1mdlm)

[5.1 System Architecture Diagram 20](#_heading=h.2u6wntf)

[5.2 Class Diagram 20](#_heading=h.3tbugp1)

[5.3 Sequence Diagrams 21](#_heading=)

[5.4 Collaboration Diagrams 26](#_heading=)

[5.5 Other UMLs 27](#_heading=h.nmf14n)

[5.6 ERD 28](#_heading=h.37m2jsg)

[5.7 Data Dictionary 28](#_heading=h.1mrcu09)

[**6. Implementation details 29**](#_heading=h.46r0co2)

[6.1 Development Setup 29](#_heading=h.2lwamvv)

[6.2 Deployment setup 29](#_heading=h.111kx3o)

[6.3 Algorithms 29](#_heading=h.3l18frh)

[6.4 Constraints 29](#_heading=h.206ipza)

[*6.4.1 Assumptions 29*](#_heading=h.4k668n3)

[*6.4.2 System constraints 29*](#_heading=h.2zbgiuw)

[*6.4.3 Restrictions 29*](#_heading=h.1egqt2p)

[*6.4.4 Limitations 29*](#_heading=h.3ygebqi)

[**7. Testing 30**](#_heading=h.2dlolyb)

[7.1 Extended Test Cases 30](#_heading=h.sqyw64)

[7.2 Decision Table 30](#_heading=h.1rvwp1q)

[*7.2.1 Code snippet 30*](#_heading=h.4bvk7pj)

[*7.2.2 Decision coverage table 30*](#_heading=h.2r0uhxc)

[7.3 Traceability Matrix 31](#_heading=h.1664s55)

[*7.3.1 RID vs UCID (requirements vs use cases) 31*](#_heading=h.3q5sasy)

[*7.3.2 Prototypes (RID vs PID) 32*](#_heading=h.25b2l0r)

[*7.3.3 Test Cases (RID vs TID) 32*](#_heading=h.kgcv8k)

[*7.3.4 Coverage (UCID vs TID) 32*](#_heading=h.34g0dwd)

[**8. Results/Output/Statistics 33**](#_heading=h.1jlao46)

[8.1 %completion 33](#_heading=h.43ky6rz)

[8.2 %accuracy 33](#_heading=h.2iq8gzs)

[8.3 %correctness 33](#_heading=h.xvir7l)

[**9. Conclusion 34**](#_heading=h.3hv69ve)

[**10. Future work 35**](#_heading=h.1x0gk37)

[**11. Bibliography 36**](#_heading=h.4h042r0)

[11.1 Books 36](#_heading=h.2w5ecyt)

[11.2 Journals 36](#_heading=h.1baon6m)

[11.3 Articles 36](#_heading=h.3vac5uf)

[11.4 Research papers 36](#_heading=h.2afmg28)

[11.5 Other References 36](#_heading=h.pkwqa1)

[**12. Appendix 37**](#_heading=h.39kk8xu)

[12.1 Glossary of terms 37](#_heading=h.1opuj5n)

[12.2 Pre-requisites 37](#_heading=h.48pi1tg)

## Definitions and Acronyms

*Provide definitions or references to all the definitions of the special terms and acronyms used within this document*

e.g

| **Acronym** | **Definition** |
| --- | --- |
| UMT | University of Management and Technology |
| POS | Point of Sale |

*Table 2: Table of acronyms and definitions*

## List of Figures

[Figure 1: 3.4](#_heading=h.3mzq4wv) System use case diagram 14

[Figure 2: 5.1 System Architecture Diagram](about:blank) 18

[Figure 3: Class Diagram](about:blank) 18

[Figure 4: Buy Item 19](#_heading=h.319y80a)

[Figure 5: Login 20](#_heading=h.1gf8i83)

[Figure 6: Stock Prediction 20](#_heading=h.40ew0vw)

[Figure 7: Product Recommendation 21](#_heading=h.2fk6b3p)

[Figure 8: Account Integration 21](#_heading=h.upglbi)

[Figure 9: Supplier Management 22](#_heading=h.3ep43zb)

[Figure 10: Inventory Counting 22](#_heading=h.1tuee74)

[Figure 11: Sales Reporting 23](#_heading=h.4du1wux)

[Figure 12: Multi-Location Monitoring 23](#_heading=h.2szc72q)

[Figure 13: Alerts for Low Stock 24](#_heading=h.184mhaj)

[Figure 14: Collaboration Diagrams 24](#_heading=h.279ka65)

[Figure 15: Database Diagrams 25](#_heading=h.meukdy)

[Figure 16: ERD Diagram 26](#_heading=h.36ei31r)

## List of Tables

[Table 1: Revision Chart 8](#_heading=h.19c6y18)

[Table 2: Table of acronyms and definitions 4](#_heading=h.lnxbz9)

[Table 3: List of Streak holders 10](#_heading=h.1ci93xb)

[Table 4: Feature Comparison 11](#_heading=h.1302m92)

[Table 5: Requirenments Table 13](#_heading=h.147n2zr)

[Table 6: Extended Use cases 17](#_heading=h.2250f4o)

[Table 7: Data Dictionary 26](#_heading=h.1ljsd9k)

[Table 8: RID vs UCID (requirements vs use cases) 29](#_heading=h.45jfvxd)

# Introduction

*This section should describe the project and the software product being to be built. No text is necessary between the heading above and the heading below unless otherwise desired.*

## Motivations

This project aims to simplify inventory management and optimize profits for small to medium businesses through an integrated cloud-based point-of-sale (POS) and accounting system. By analyzing past sales data, our machine learning-powered solution will recommend stock levels to avoid stock-outs or excess stock. Being cloud-based provides flexibility for owners to access their business data anywhere, anytime. Automating routine tasks through an all-in-one platform helps small shops streamline operations and focus on growth.

## Project Overview

Managing inventory manually is time-consuming and error-prone for small to medium retail shops. They often struggle with stock outs, excess stocks and tracking customer credit purchases. Cloud based system allows the data to be backed up in case of an outage, and helps streamline the project,optimize profits and simplify operations, helping in the process of growth and franchising.

## Problem Statement

Managing inventory is one of the biggest challenges for small retail shops. They often struggle to maintain optimal stock levels as manual tracking of stocks and sales is tedious and prone to errors. This leads to issues like stock-outs of popular items or maintaining excess stock of slow-moving products. These operational inefficiencies consume valuable time and resources of shop owners. With limited capital, small businesses cannot afford losses due to poor inventory management. There is a need for an integrated solution that automates inventory management, simplifies record-keeping, and helps optimize stock levels based on sales patterns to maximize profits. This project aims to develop a cloud-based point-of-sale and inventory management system with machine learning to address these issues faced by local small retail shops, and help them grow.

## Objectives

1. Develop a cloud-based inventory management and point-of-sale system to automate core operations for small retail shops.
2. Implement machine learning algorithms to analyze historical sales data and recommend stocking levels to maximize profits and maintain minimum excess inventory.
3. Integrate inventory, sales, accounting and customer credit features into a single platform for ease of use.
4. Generate automated re-ordering of products based on recommended stock levels as sales occur to maintain optimal inventory levels.
5. Provide sales analysis reports and insights to help shop owners identify top-selling items, customize stocks, and maximize profits.
6. Develop a user-friendly dashboard interface for shop owners to access business data and manage day-to-day operations from any device with internet connectivity.
7. Conduct user evaluations to test the usability and effectiveness of the system in improving operational efficiency for small retailers.
8. Deploy a minimum viable product and obtain feedback from real shop owners to refine and enhance the system.

# Domain Analysis

## Customer

*A brief description of the client with whom you are working (or the potential customers). The organization, its products/services etc.****You will fill this section only if you have a client (contracted) .***

***The main customers for this POS and inventory system will be small shops. Things like grocery stores, provision stores, hardware stores and food shops will use it. These kinds of stores usually don't have a lot of money to spend on computers. But they need to keep track of lots of different products every day.***

***The things they sell can change a lot each week or month. Sometimes they have too many things left unsold. Other times some things run out fast. This system will help with that. It will learn from past sales and tell them what to order next.***

***Many small shops are in villages or small towns. They serve customers who come to the shop every day. Some people want things delivered or want to pay later. This system will help with those things too like keeping a record of credit customers.***

***By using this cloud-based system, small shops can get technology tools usually for big stores. It will make their work easier like managing stock, keeping records and accepting payments. This will help the small shops a lot with the difficult parts of running a store.***

## Stakeholders

*List of all stakeholders along with their roles in making of the system e.g*

| ***Stakeholder*** | ***Role in System*** |
| --- | --- |
| *Customers/users* | *The businesses and employees using the POS system to process transactions and manage their operations. Their needs and experience are top priority.* |
| *Vendors/merchants* | *Businesses selling goods/services that will be processed through the POS system. They need it to efficiently ring up sales and manage inventory.* |
| *Developers* | *The technical team building and maintaining the POS software. They need clear requirements and feedback to develop an effective product.* |
| *Sales/account managers* | *The team responsible for selling the POS system to new clients and supporting existing clients. They rely on the software meeting client needs.* |
| *Payment processors* | *Firms enabling payment processing through the POS like credit cards. Interface must meet their standards.* |

*Table 3: List of Streak holders*

**Table 2: list of stakeholders**

## Affected Groups with social or economic impact

*Here are some major affected groups that could face social or economic impacts from a point of sale system:*

*Small businesses/merchants - POS systems profoundly impact how they operate, engage customers, and track sales/inventory. Failed systems could seriously harm livelihoods.*

*Employees - Workers rely on POS systems to efficiently do their jobs like ringing up sales. Downtime shifts work burden or causes loss in sales/productivity.*

*Customers - Outages or issues impact the shopping experience and ability to purchase items in a convenient manner. Repeat customers may seek alternatives.*

## Dependencies/ External Systems

*None...*

## Reference Documents

*Provide references to all documents that have been consulted during the analysis phase.*

### Related Projects

*List of all the documents/ projects that you have looked up as reference material for this project along with their links/references. E.g*

*In order to develop UMTmanagementSystem, we looked up several similar systems. Their details are given below*

1. *FastManagementSystem(FMS)*

*Developed by XYZ. The partial documentation was obtained by the XYZ development team and the working of this management software was observed from abcFAST.com.pk*

1. *BeaconHouse Management System (BHMS)*

*Developed by ABC. The working of this management software was observed from abcbeaconhouse.com.pk. no relevant documentation was available.*

1. *“constructing and ideal academic system” (CIAS)*

*Research paper published by IEEE. The research paper is not available for free. It is only available to IEEE members*

### Feature Comparison

| *Features* | *QuickStock* | *Vypar* | *Quickbooks* |
| --- | --- | --- | --- |
| POS Accounting Software | ✔ | ✔ | ✔ |
| Data Backup | ✔ | ✔ | ✔ |
| Cloud Storage | ✔ | X | X |
| Stock Prediction | ✔ | X | X |
| Sales Report | ✔ | X | X |
| Sales Graph | ✔ | X | X |
| Credit System | ✔ | X | X |

*Table 4: Feature Comparison table*

# Requirements analysis

## Requirements

*This POS system with machine learning and camera credit features is being developed to help retail store owners efficiently manage their inventory, sales and customers.*

*System Functions:*

* *Record sales transactions and reduce inventory quantities*
* *Track customer purchase history and manage credit accounts*
* *Predict inventory demand based on past sales data*
* *Suggest optimal stocking levels to the owner*
* *Facilitate credit sales by matching customer photos*

*Non-Functional Requirements:*

* *System response time: Load price listings within 3 seconds, availability checks within 10 seconds*
* *Concurrent user load: Support at least 10 concurrent users*
* *Data storage: Inventory, sales and customer data will be stored in a cloud database*
* *Interface with accounting software to sync transactions and financial reports*

*Constraints:*

* *The system will be hosted as a web-based application*
* *Development platform will be Java/SpringBoot for the backend API and MySQL database*
* *The frontend will be developed as a responsive web interface*
* *Core functions must be available even during occasional server outages*
* *Privacy of customer data as per applicable legal requirements*

*External Interfaces:*

* *RESTful API for inventory management by connected suppliers*
* *Sync with selected accounting packages (e.g. Tally) via standardized formats*

| *RID* | *description* | *Category* | *Attribute* | *Details & Boundary Constraints* |
| --- | --- | --- | --- | --- |
| *R1.1* | *Record the underway sale – the items purchased* | *non-functional* | *System Response time* | *Price listing within 3 seconds*  *Availability agreement in less than 10 sec* |
| *R1.2* | *Reduce inventory quantities when a sale is committed* | *non-functional* | *Concurrent user load* |  |

*Table 5: Requirenments Table*

## List of Actors

*Define the system boundary and list all actors with the use cases.****all the actors must also be mentioned in your list of stakeholders***

*For example:*

*Cashier; this person performs all the financial activities*

*Account Manager; this person supervises all financial activities*

*Customer: Main end user that buys items from the store*

*Store Owner: Manages one or multiple store locations and uses POS for reporting, monitoring, replenishment etc.*

*Supplier: Provides inventory to stores and accesses POS data for demand planning and logistics.*

*ML Model: Not a true actor but plays a role in automated processes like stock predictions and recommendations.*

## List of use cases

*List all the use cases, with a brief description (should not exceed two lines):*

*Buy Item; let users buy items from he pos software*

*Log In; allow user to provide account information and access the restricted services*

*Stock Prediction: ML model analyzes past sales data to determine optimum stock levels for maximizing sales while minimizing excess inventory.*

*Product Recommendations: System recommends additional products to customers during purchases based on their purchase history and preference patterns.*

*Accounting Integration: Inventory, sales and financial data seamlessly syncs with accounting software in real-time to eliminate redundant data entry and provide integrated insights.*

*Supplier Management: Authorized suppliers can access inventory data to better plan production and timely deliveries based on real demand patterns.*

*Inventory Counting: Digital stock records automatically facilitate accurate cycle counting with minimal manual effort.*

*Sales Reporting: Detailed reports provide actionable intelligence on best-sellers, trends etc. to optimize merchandising decisions.*

*Multi-Location Monitoring: Cloud-based accessibility allows remote operation and insights across multiple storefront locations.*

*Alerts for Low Stock: ML predictions notify owners in advance of low inventory levels to timely replenish and avoid stock-outs.*

## System use case diagram



*Figure 1: 3.4 System use case diagram*

**Figure 1: sample use case diagram with explanation**

## Extended use cases

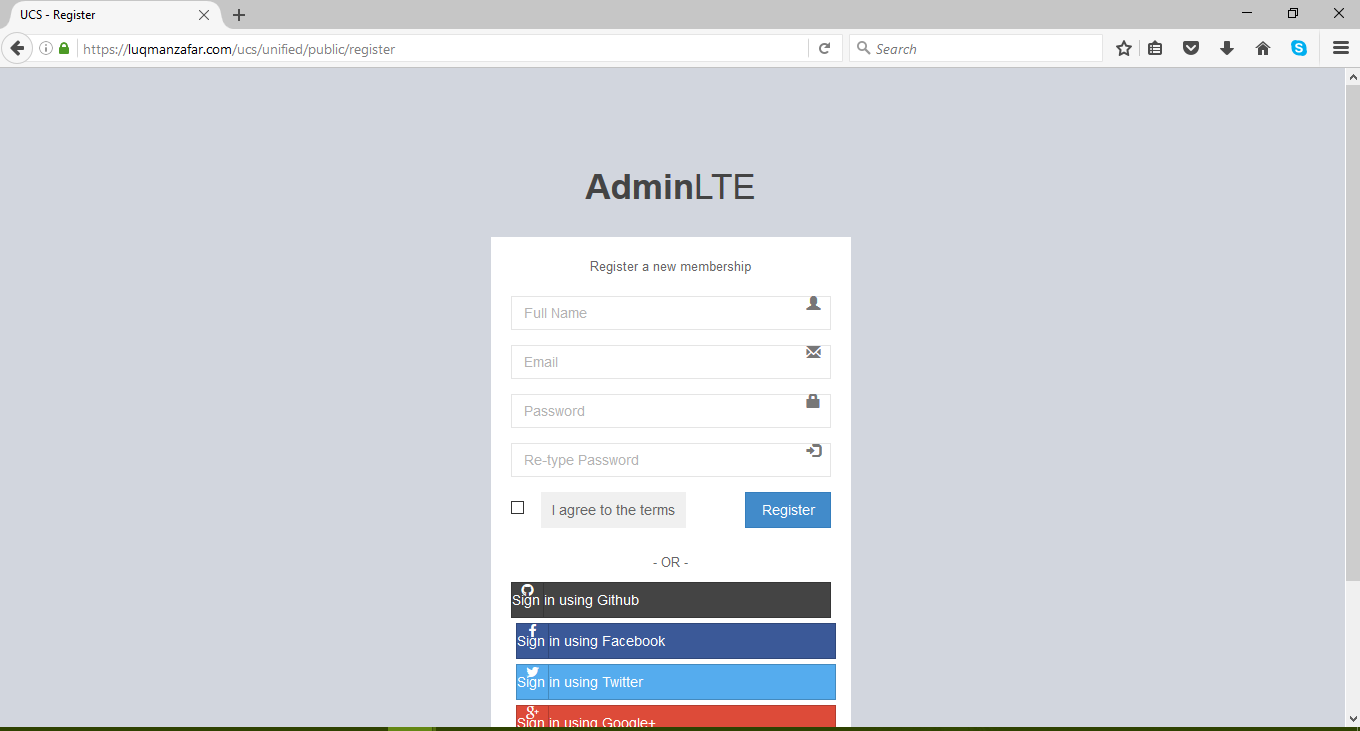
*Every use case form the list must be elaborated here. E.g*

| **Use Case ID:** | Enter a unique numeric identifier for the Use Case. e.g. UC-1.2.1 | | | |
| --- | --- | --- | --- | --- |
| **Use Case Name:** | Enter a short name for the Use Case using an active verb phrase. e.g. Withdraw Cash | | | |
| **Created By:** |  | | **Last Updated By:** |  |
| **Date Created:** |  | | **Last Revision Date:** |  |
| **Actors:** | | [An actor is a person or other entity external to the software system being specified who interacts with the system and performs use cases to accomplish tasks. Different actors often correspond to different user classes, or roles, identified from the customer community that will use the product. Name the actor that will be initiating this use case (primary) and any other actors who will participate in completing the use case (secondary).] | | |
| **Description:** | | [Provide a brief description of the reason for and outcome of this use case.] | | |
| **Trigger:** | | [Identify the event that initiates the use case. This could be an external business event or system event that causes the use case to begin, or it could be the first step in the normal flow.] | | |
| **Preconditions:** | | [List any activities that must take place, or any conditions that must be true, before the use case can be started. Number each pre-condition. e.g.   1. Customer has active deposit account with ATM privileges 2. Customer has an activated ATM card.] | | |
| **Post conditions:** | | [Describe the state of the system at the conclusion of the use case execution. Should include both *minimal guarantees* (what must happen even if the actor’s goal is not achieved) and the *success guarantees* (what happens when the actor’s goal is achieved. Number each post-condition. e.g.   1. Customer receives cash 2. Customer account balance is reduced by the amount of the withdrawal and transaction fees] | | |
| **Normal Flow:** | | [Provide a detailed description of the user actions and system responses that will take place during execution of the use case under **normal, expected** conditions. This dialog sequence will ultimately lead to accomplishing the goal stated in the use case name and description.   1. Customer inserts ATM card 2. Customer enters PIN 3. System prompts customer to enter language performance English or Spanish 4. System validates if customer is in the bank network 5. System prompts user to select transaction type 6. Customer selects Withdrawal From Checking 7. System prompts user to enter withdrawal amount 8. … 9. System ejects ATM card] | | |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | | [Document **legitimate** branches from the main flow to handle special conditions (also known as extensions). For each alternative flow reference the branching step number of the normal flow and the condition which must be true in order for this extension to be executed. e.g. Alternative flows in the *Withdraw Cash* transaction:  4a. In step 4 of the normal flow, if the customer is not in the bank network   1. System will prompt customer to accept network fee 2. Customer accepts 3. Use Case resumes on step 5   4b. In step 4 of the normal flow, if the customer is not in the bank network   1. System will prompt customer to accept network fee 2. Customer declines 3. Transaction is terminated 4. Use Case resumes on step 9 of normal flow   Note: Insert a new row for each distinctive alternative flow. ] | | |
| **Exceptions:** | | [Describe any anticipated **error conditions** that could occur during execution of the use case, and define how the system is to respond to those conditions.  e.g. Exceptions to the Withdraw Case transaction  2a. In step 2 of the normal flow, if the customer enters and invalid PIN   1. Transaction is disapproved 2. Message to customer to re-enter PIN 3. Customer enters correct PIN 4. Use Case resumes on step 3 of normal flow] | | |
| **Includes:** | | [List any other use cases that are included (“called”) by this use case. Common functionality that appears in multiple use cases can be split out into a separate use case that is included by the ones that need that common functionality. e.g. steps 1-4 in the normal flow would be required for all types of ATM transactions- a Use Case could be written for these steps and “included” in all ATM Use Cases.] | | |
| **Frequency of Use:** | | [How often will this Use Case be executed. This information is primarily useful for designers. e.g. enter values such as 50 per hour, 200 per day, once a week, once a year, on demand etc.] | | |
| **Special Requirements:** | | [Identify any additional requirements, such as nonfunctional requirements, for the use case that may need to be addressed during design or implementation. These may include performance requirements or other quality attributes.] | | |
| **Assumptions:** | | [List any assumptions that were made in the analysis that led to accepting this use case into the product description and writing the use case description.  e.g. For the *Withdraw Cash* Use Case, an assumption could be:  The Bank Customer understands either English or Spanish language.] | | |
| **Notes and Issues:** | | [List any additional comments about this use case or any remaining open issues or TBDs (To Be Determined) that must be resolved. e.g.   1. What is the maximum size of the that a use can have?] | | |

*Table 6: Extended Use cases*

## User interfaces (mock screens)

*Initial mockup screens (even hand drawn drafts) will be inserted here. Each screen will be given an appropriate prototype id.e.g.*

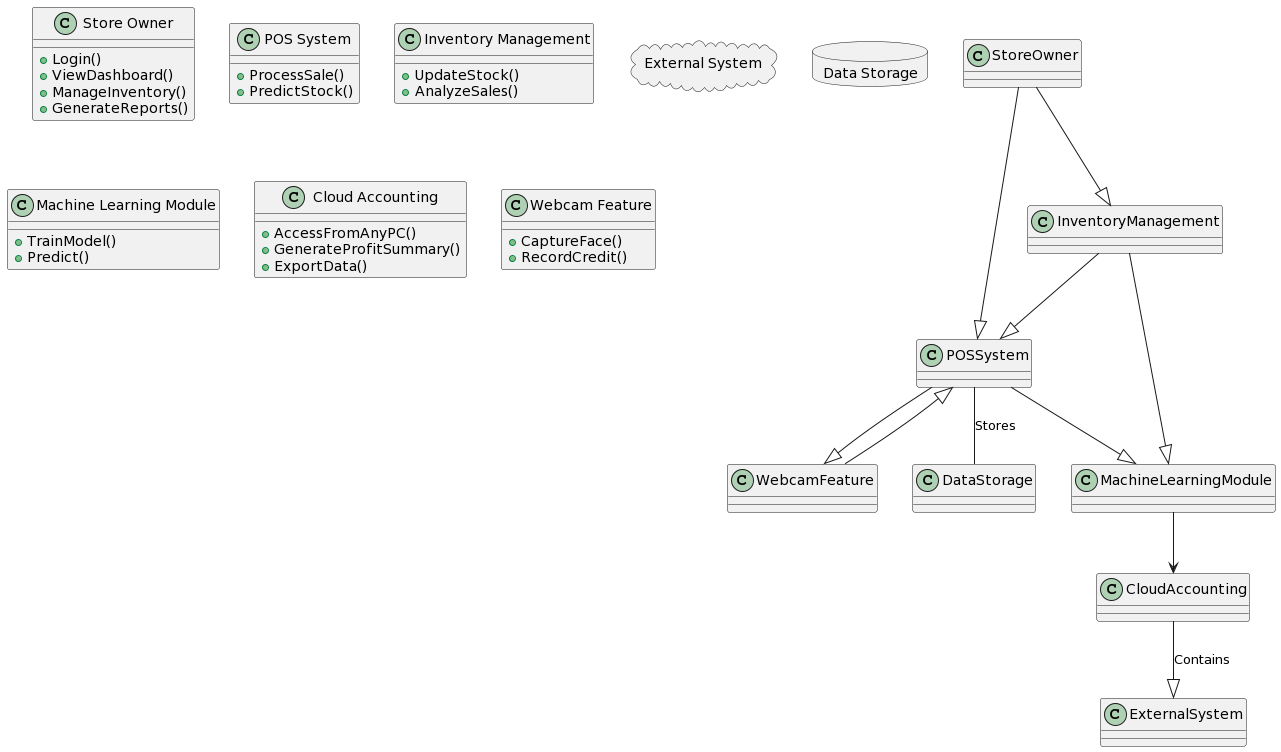
******

**Prototype1: (P1) register a new member**

# System Design

*Describe the system architecture, or simply provide the architecture diagram. For School system it may include web based front end, webserve , database etc. Don’t worry too much about it just give a simple diagram of a typical web based project.*

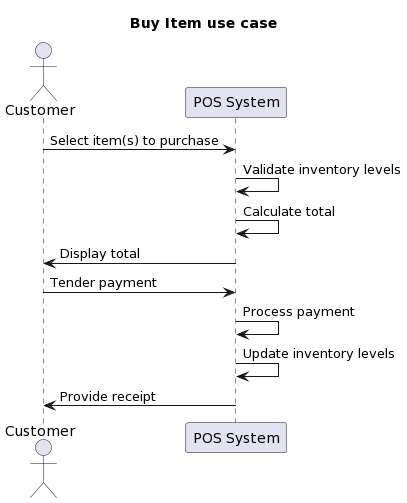
## System Architecture Diagram



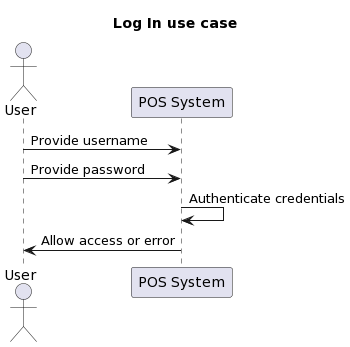
## Class Diagram

## Sequence Diagrams

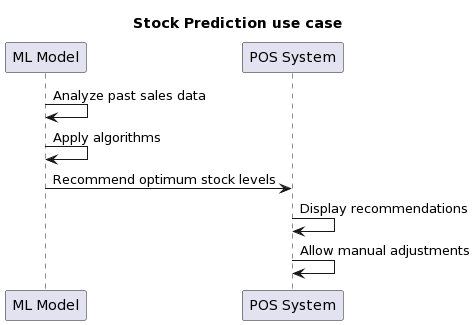
5.3.1 Buy Item



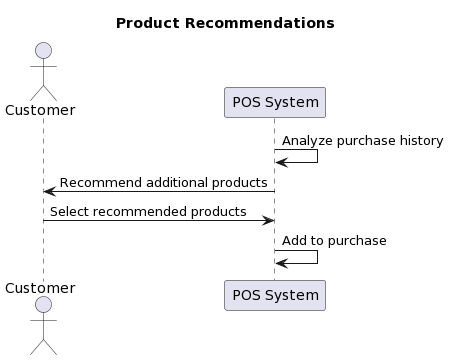
*Figure 4: Buy Item*



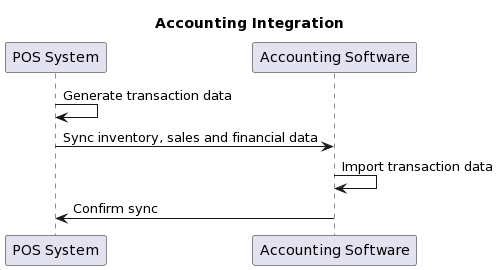
*Figure 5: Login*



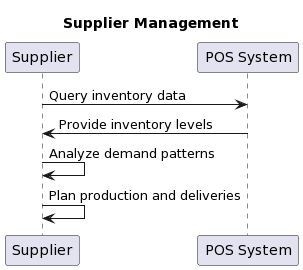
*Figure 6: Stock Prediction*



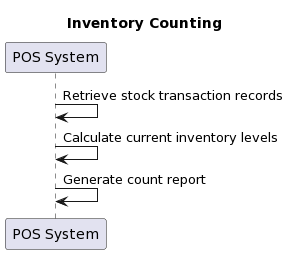
*Figure 7: Product Recommendation*



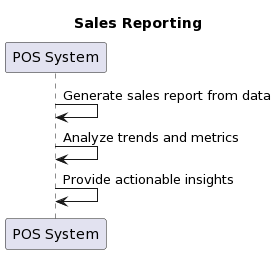
*Figure 8: Account Integration*



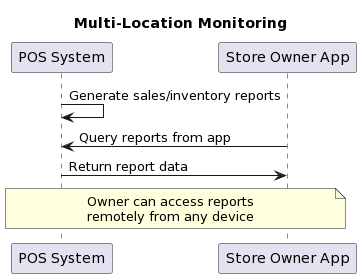
*Figure 9: Supplier Management*



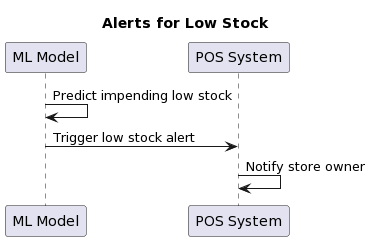
*Figure 10: Inventory Counting*



*Figure 11: Sales Reporting*

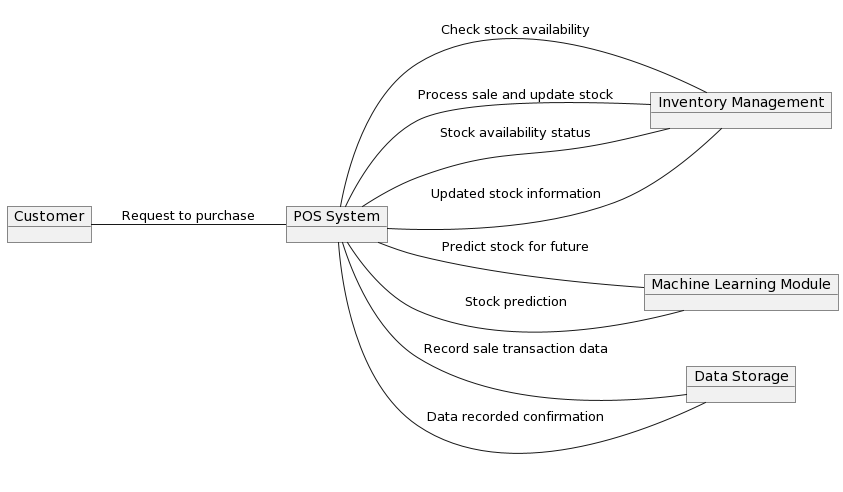


*Figure 12: Multi-Location Monitoring*



*Figure 13: Alerts for Low Stock*

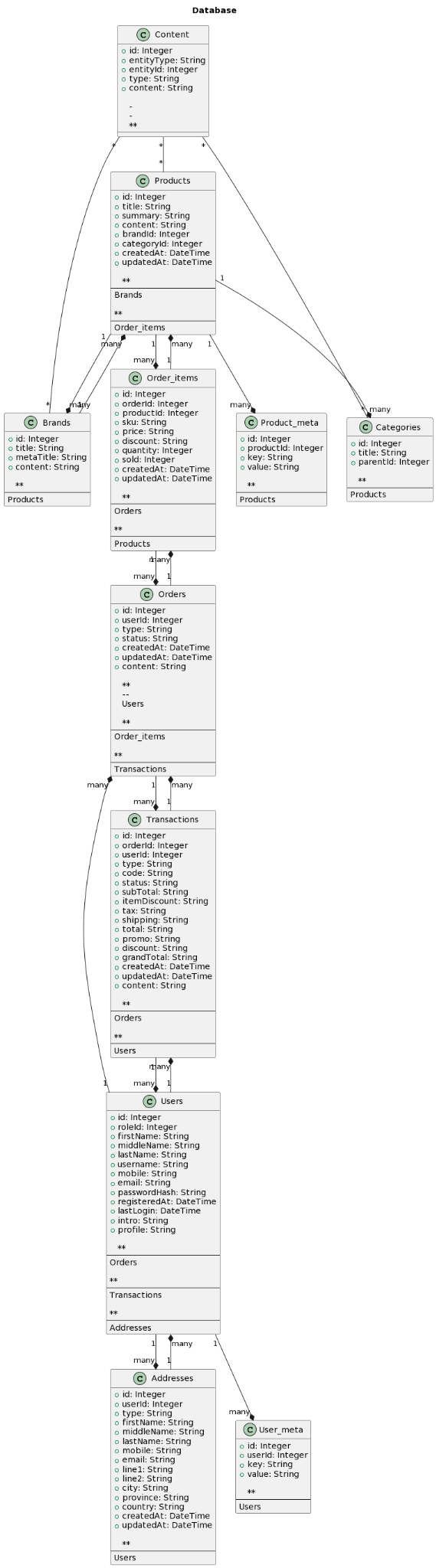
## Collaboration Diagrams



*Figure 14: Collaboration Diagrams*

## Other UMLs

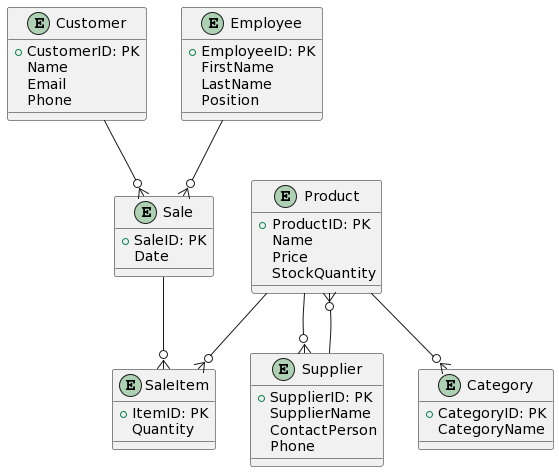
**DataBase Diagram**

****

*Figure 15: Database Diagrams*

**

## ERD



*Figure 16: ERD Diagram*

## Data Dictionary

*This section may be used to provide the details of interface elements that are present on the screenshots.*

| *Element Name* | *Type* | *Validation* | *Mandatory* | *Remarks* |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

*Table 7: Data Dictionary*

# Implementation details

## Development Setup

*List your tools and technologies and their role in development.*

## Deployment setup

*How and where was your software deployed? Did you face any problems, how did you overcome these problems.*

## Algorithms

*Entire code of software is not required. Just highlight your important (user defined/ improved) algorithms.*

## Constraints

### Assumptions

*Things we assume will be true.*

*e.g.:*

* *We will receive all necessary technical support from the engineers at cMeRun, Select and Mellon Bank to help design the interfaces between their systems and enGyro.*
* *All database maintenance will be handled by the client.*
* *There will be no real-time interfacing with any accounting systems.*

### System constraints

*A constraint specifies how the system must operate or how it must be built*

### Restrictions

*Constraints applied on the system by the client*

### Limitations

*Services your software is unable to provide*

# Testing

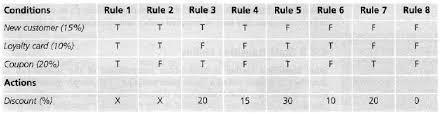
## Extended Test Cases

## 

## Decision Table

### Code snippet

### Decision coverage table



## Traceability Matrix

### RID vs UCID (requirements vs use cases)

| **UCID/RID** | **R**  **1** | **R**  **2** | **R**  **3** | **R**  **4** | **R**  **5** | **R**  **6** | **R**  **7** | **R**  **8** | **R**  **9** | **R**  **10** | **R**  **11** | **R**  **12** | **R**  **13** | **R**  **14** | **R**  **15** | **R**  **16** | **R**  **17** | **R**  **18** | **R**  **19** | **R**  **20** | **R**  **21** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| UC 1 | ✔ | ✔ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC 2 |  | ✔ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC 3 | ✔ | ✔ |  |  |  |  |  |  |  |  |  |  | ✔ |  |  |  |  |  |  |  |  |
| UC 4 | ✔ | ✔ |  |  |  |  |  |  |  |  |  |  |  | ✔ |  |  |  |  |  |  |  |
| UC 5 | ✔ | ✔ | ✔ |  | ✔ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC 6 | ✔ | ✔ |  | ✔ |  | ✔ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC 7 | ✔ | ✔ | ✔ |  |  |  |  |  | ✔ |  |  |  |  |  |  |  |  |  |  |  |  |
| UC 8 | ✔ | ✔ |  | ✔ |  |  |  |  |  | ✔ |  |  |  |  |  |  |  |  |  |  |  |
| UC 9 | ✔ | ✔ | ✔ |  | ✔ |  |  |  |  |  | ✔ |  |  |  |  |  |  |  |  |  |  |
| UC 10 | ✔ | ✔ |  | ✔ |  | ✔ |  |  |  |  |  | ✔ |  |  |  |  |  |  |  |  |  |
| UC 11 | ✔ | ✔ | ✔ |  | ✔ |  | ✔ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC 12 | ✔ | ✔ |  | ✔ |  | ✔ |  | ✔ |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC 19 | ✔ | ✔ | ✔ |  | ✔ |  |  |  |  |  |  |  |  |  | ✔ |  |  |  |  |  |  |
| UC 20 | ✔ | ✔ |  | ✔ |  | ✔ |  |  |  |  |  |  |  |  |  | ✔ |  |  |  |  |  |
| UC 21 | ✔ | ✔ | ✔ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC 22 | ✔ | ✔ |  | ✔ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC 23 | ✔ | ✔ | ✔ |  |  |  |  |  |  |  |  |  |  |  |  |  | ✔ |  |  |  |  |
| UC 24 | ✔ | ✔ |  | ✔ |  |  |  |  |  |  |  |  |  |  |  |  |  | ✔ |  |  |  |
| UC 25 | ✔ | ✔ | ✔ | ✔ |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ✔ |  |  |
| UC 26 | ✔ | ✔ | ✔ | ✔ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ✔ |  |
| UC 27 | ✔ | ✔ | ✔ | ✔ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ✔ |

*Table 8: RID vs UCID (requirements vs use cases)*

### Prototypes (RID vs PID)

### Test Cases (RID vs TID)

### Coverage (UCID vs TID)

# Results/Output/Statistics

## %completion

*Use the matrix & values from 7.3.1 to show that all requirements are being fulfilled.*

## %accuracy

*Use the matrix & values from 7.3.3 to show that all requirements have been implemented correctly.*

## %correctness

*Use the matrix & values from 7.3.4 to show that all requirements have been tested to be conforming to requirements.*

# Conclusion

# Future work

# Bibliography

*Use IEEE or ACM format for citations*

## Books

## Journals

## Articles

## Research papers

## Other References

# Appendix

## Glossary of terms

## Pre-requisites

*Must use contents of development/ deployment setup & external system dependencies*