

Apparel Point of Sale (APOS) System

System Design Document

CMSC 495

University of Maryland University College (UMUC)

Team Tenacious Turtles

Signatures:

John Walsh

Kevin Boyer

Manoj Kumar

Ryan Harvell

Sean Walters

Vance Molhusen

Approval by Professor Mark Wireman:

Revision	Date	Description
SD-1	09/18/16	Initial Version

Table of Contents

2	INTRODUCTION	5
2.1	Purpose.....	5
2.2	Project Executive Summary.....	5
2.2.1	System Overview	5
2.2.2	Scope of the project.....	5
2.2.3	Assumptions.....	6
2.2.4	Overall approach in analysis and design.....	6
2.2.5	Design Constraints	10
2.2.6	Future Contingencies	11
2.3	Document Organization	11
2.4	Points of Contact.....	11
2.5	Project References.....	12
2.6	Glossary	12
3	SYSTEM ARCHITECTURE	12
3.1	System Hardware Architecture	12
3.2	System Software Architecture	13
3.3	Internal Communications Architecture	13
4	FILE AND DATABASE DESIGN	14
4.1	Database Management System Files	14
4.1.1	Entity Relationship Diagram (ERD)	14
4.1.2	Database Tables	15
4.1.3	Table Schema.....	15
4.2	Non-Database Management System Files	16
5	HUMAN-MACHINE INTERFACE	16
5.1	Inputs.....	16
5.1.1	Log-in Screen.....	17
5.1.2	Transaction Screen.....	18
5.1.3	Inventory Look-up Screen	20
5.1.4	Transaction Report Criteria Screen.....	20
5.2	Outputs	21
6	DETAILED DESIGN	24
6.1	Hardware Detailed Design	24
6.2	Software Detailed Design	26
6.2.1	Use Case Diagram.....	26
6.2.2	Use Case Scenarios	26
6.2.3	Activity Diagram.....	28
6.2.4	Sequence Diagram	29
6.2.5	Class Diagram.....	30
6.2.6	State Chart Diagrams	31
6.3	Internal Communications Detailed Design	32
7	EXTERNAL INTERFACES	33
7.1	Interface Architecture	33

7.2	Interface Detailed Design.....	33
8	SYSTEM INTEGRITY CONTROLS	33
9	PERFORMANCE ESTIMATES	34
9.1	Performance of APOS system functions.....	34
9.2	Performance Factors	34
10	APPENDIX.....	36

SYSTEM DESIGN DOCUMENT

2 INTRODUCTION

2.1 Purpose

The purpose of this SDD is to create a set of requirements for an application that will perform the functions of the operations and management of a point of sale system for an apparel company. These requirements will define the individual developmental components for this program, and will be used during the implementation phase of the system. This SDD is written for developers and others who are involved in the production of this software system.

2.2 Project Executive Summary

This section provides a description of the project from a management perspective and an overview of the framework within which the conceptual system design was prepared.

2.2.1 System Overview

The Tenacious Turtles Clothing Company requires a Point of Sale system to handle its cash register application. This application will have sales associate log-in, sales and returns processing, store inventory viewing, and the ability to view reports on daily sales transactions. The Apparel Point of Sale System is designed to meet the organization's requirements.

2.2.2 Scope of the project

The scope of the project is development of a cash register application only. It will have user interfaces for: sales associate log-in, sales processing, and store inventory viewing. The application will adjust the store inventory based on sales and returns. Daily sales transactions can be viewed on a report (displayed or printed). The system will display and print two receipts - one for the store and one for the customer after each transaction. Sales associate log-in will be required before performing any other interaction with the system.

The program will be written to be a single JAVA executable file located on the Windows operating system of the register PC.

2.2.3 Assumptions

A. Maintenance of the following information will be outside the scope of this project. The assumption is that these are populated by company's ERP (Enterprise Resource Planning) system. For this project, all these information will be pre-populated.

1. Store inventory: sales item, units, location, etc. Replenishment is handled by a separate ERP system.
2. Sales associate information: name, designation, etc. This information is handled by company's personnel management system.
3. Store information: store name, address, manager, etc.
4. Sales Item information: season, style, color, size, gender code, UPC code, etc.
5. Other master information.

B. Payment posting will be outside the scope of this application. The company's ERP system collects sales transactions from each store and then transfers detailed information to the company's database.

2.2.4 Overall approach in analysis and design

Tenacious Turtles team will use an Object-Oriented Analysis and Design (OOAD) approach with Unified Modeling Language for this project. For those team members not familiar with UML, the following information on UML can be helpful.

Unified Modeling Language (UML)

The Unified Modeling Language (UML) is a graphical language for OOAD that gives a standard way to write a software system's blueprint. It helps to visualize, specify, construct, and document the artifacts of an object-oriented system. It is used to depict the structures and the relationships in a complex system.

The conceptual model of UML has three major elements: basic building blocks, rules, and common mechanism. Basic building blocks are: things, relationship, and diagrams.

Things in UML include the following:

1. *Structural things*: Nouns to represent static elements that may be either physical or conceptual in a UML model.
2. *Behavioral things*: Verbs to represent the dynamic behavior over time and space. Two types of behavioral things are interaction and state.
3. *Grouping things*: Grouping has the organizational parts of the UML models such as packaging.
4. *Annotational things*: Explanations to represent the comments applied to describe elements.

Relationships in UML include the following:

1. *Dependency*: A semantic relationship between two things such that a change in one thing brings a change in another. The former is an independent thing, and the later is the independent thing.
2. *Association*: A structural relationship that represents a group of links having common structure and common behavior.
3. *Generalization*: This represents a generalization/specialization relationship in which subclasses inherit structure and behavior from super-classes.
4. *Realization*: This is a semantic relationship between two or more classifiers such that one classifier lays down a contract that the other classifiers ensure to abide by.

Diagrams in UML:

There are nine types of diagrams in UML. These are:

1. Class Diagram
2. Object Diagram
3. Use Case Diagram
4. Sequence Diagram
5. Collaboration Diagram
6. State Chart Diagram
7. Activity Diagram
8. Component Diagram
9. Deployment Diagram

In this project we will use only use case diagram, sequence diagram, activity diagram, class diagram, and state chart diagram.

Rules

UML has a number of rules so that the models are semantically self-consistent and related to other models in the system harmoniously. UML has semantic rules for names, scope, visibility, integrity, and execution.

Common Mechanisms: UML has four common mechanisms. These are specifications, adornments, common divisions, and extensibility mechanisms.

Specifications: In UML, behind each graphical notation, there is a textual statement denoting the syntax and semantics. These are the specifications. The specifications provide a semantic backplane that contains all the parts of a system and the relationship among the different paths.

Adornments: Each element in UML has a unique graphical notation. There are notations to represent the important aspects of an element like name, scope, visibility, etc.

Common Divisions: Object-oriented systems can be divided in many ways. The two common ways of division are:

Division of classes and objects: A class is an abstraction of a group of similar objects. An object is the concrete instance that has actual existence in the system.

Division of Interface and Implementation: An interface defines the rules for interaction. Implementation is the concrete realization of the rules defined in the interface.

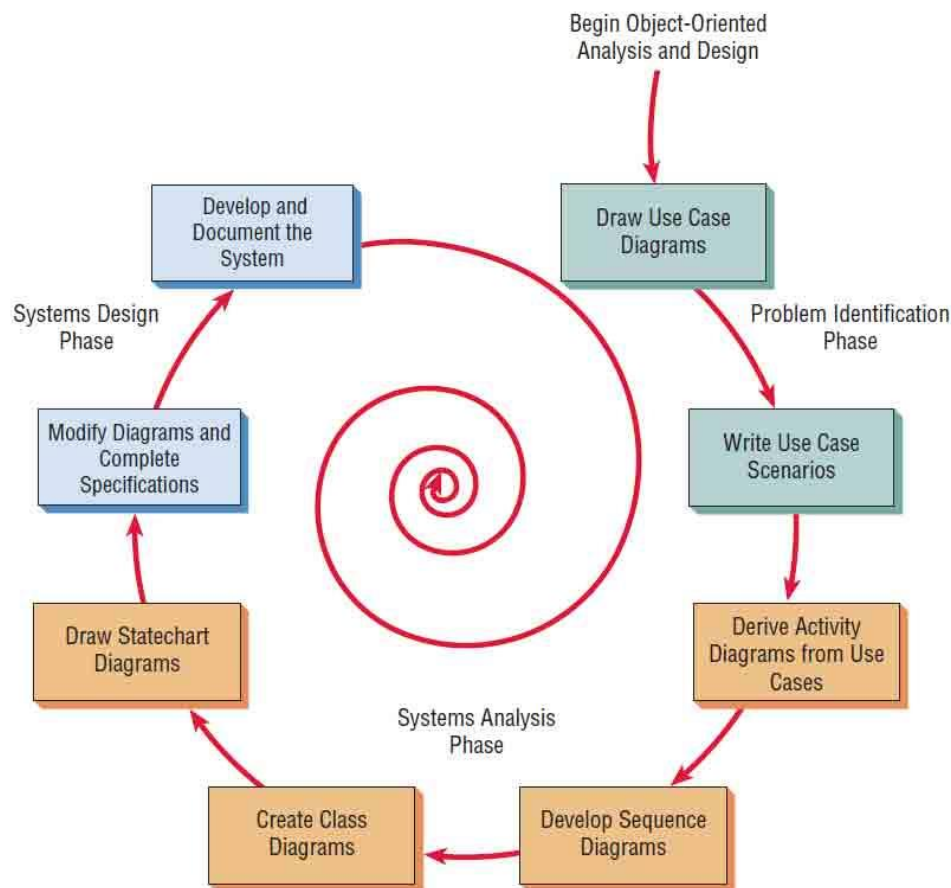
Extensibility Mechanisms: UML is an open-ended language. It is possible to extend the capabilities of UML in a controlled manner to suit the requirements of a system. The extensibility mechanisms are:

Stereotypes: It extends the vocabulary of the UML, through which new building blocks can be created out of existing ones.

Tagged Values: It extends the properties of UML building blocks.

Constraints: It extends the semantics of UML building blocks.

2.2.4.1 Phases in UML



Picture 1. OOAD Phases. (w3Computing, 2016).

The following are the phases in an OOAD using UML.

1. Preparing Use Case Diagrams:

In this phase the analyst identifies the actors and the major events initiated by the actors. Often the analyst will start by drawing a diagram with stick figures

representing the actors and arrows showing how the actors relate. This is called a use case diagram and it represents the standard flow of events in the system.

2. Write use case scenarios: After creating use case diagrams, write use case scenarios, which describe in words the steps that are normally performed.
3. Activity Diagrams: Activity diagram illustrates all the major activities in the use case.
4. Sequence Diagrams: Create one or more sequence diagrams for each use case, which show the sequence of activities and their timing. This is an opportunity to go back and review the use cases, rethink them, and modify them if necessary.
5. Class Diagrams: In this phase, identify objects and classes, object relationships, attributes, methods. The nouns in the use cases are objects that can potentially be grouped into classes.
6. State Chart Diagrams: The class diagrams are used to draw state chart diagrams. This can help to understand complex processes that cannot be fully derived by the sequence diagrams. The state chart diagrams are extremely useful in modifying class diagrams, so the iterative process of UML modeling continues.
7. System Design and Specifications: In this phase, begin systems design by modifying the UML diagrams. After that, complete the specifications. Systems design means modifying the existing system and that implies modifying the diagrams drawn in the previous phase. These diagrams can be used to derive classes, their attributes, and methods. Class specifications for each class including attributes, methods, and their descriptions need to be completed. Method specifications detailing the input and output requirements, and internal processing need to be completed also.
8. Develop and document the system: Code development begins in this phase. Documentation is critical and correcting diagrams in previous phases.

2.2.5 Design Constraints

Software design constraint: Ideally OOD should have a presentation layer, a business logic layer, and a data access layer. But due to the complexity added to the development by having a data access layer and limited time for development, our team decided not to

have a separate data access layer. Separate data access layer will be in the alternate design.

Hardware constraint: As credit card validation is not feasible for this project, the application will be a “cash only” register. The system will not have other integrated peripherals such as UPC code scanner, electronic signature pad, or check reader.

2.2.6 Future Contingencies

Contingencies may cause a modification in the design of the system. Software modifications can and will happen. The team members responsible for developing the functionality components of the project will periodically work with other members of the team as they complete their tasks in order to review and assess how they are generating their components. This inter-collaboration will ensure that only reliable and accurate processes and techniques are being used to complete the project. Any deviations from these reliable processes will be identified and discussed within the team and corrected or left in place as determined by the team as a group.

The design document will be updated with every modification to the system. The document changes will be kept under version control.

2.3 Document Organization

The document layout is from UMUC system design template. But it is slightly re-organized and added more sections to accommodate OOAD model.

2.4 Points of Contact

Tenacious Turtles Team (TTT) members:

John Walsh	J.r.walsh.782@gmail.com
Kevin Boyer	kevinb011@yahoo.com
Manoj Kumar	manojkkumar@hotmail.com
Ryan Harvell	rlh04d@gmail.com
Sean Walters	spwalters@gmail.com
Vance Molhusen	aquasel@gmail.com

2.5 Project References

TutorialsPoint (2016). Object Oriented Analysis and Design (OOAD) Tutorial.

TutorialsPoint. Retrieved on 09/13/16 from

https://www.tutorialspoint.com/object_oriented_analysis_design/index.htm

w3Computing (2016). Object Oriented System Analysis and Design. w3Computing, A Developer Reference Website. Retrieved on 09/13/16 from

<http://www.w3computing.com/systemsanalysis/object-oriented-systems-analysis-design/>

IEEE Std 610.12-1990: IEEE Standard Glossary of Software Engineering Terminology. 1

IEEE Std 730-1998: IEEE Standard for Software Quality Assurance Plans.

IEEE Std 828-1998: IEEE Standard for Software Configuration Management Plans.

IEEE Std. 830-1998: IEEE Recommended Practice for Software Requirements Specifications.

IEEE Std. 1233-98: IEEE Guide for Developing System Requirements Specification

IEEE Std. 1058-1998: IEEE Standard for Software Project Management Plans

IEEE Std. 1016-1998: IEEE Recommended Practice for Software Design Descriptions

Tenacious Turtles APOS Test and Evaluation Master Plan

Tenacious Turtles APOS Project Plan

Tenacious Turtles APOS User's Guide

Tenacious Turtles Systems Requirements Spreadsheet

2.6 Glossary

POS: Point of Sale

IEEE: Institute of Electrical and Electronics Engineers

JDK: Java Development Kit

PMP: Project Management Plan

3 SYSTEM ARCHITECTURE

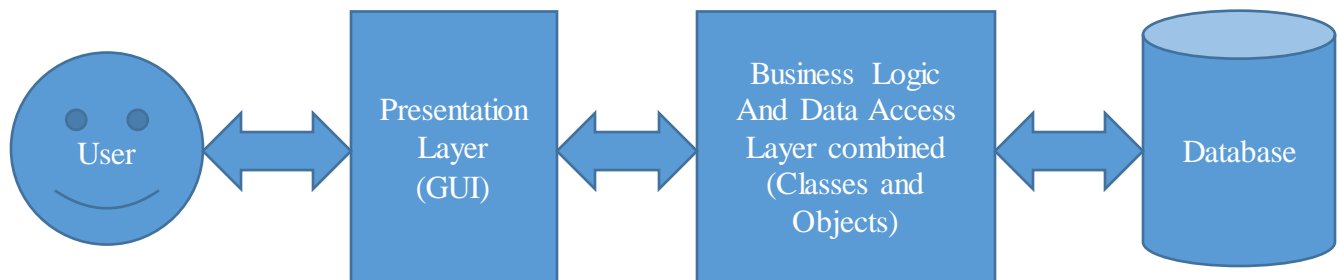
In this section, system architecture for the project is detailed.

3.1 System Hardware Architecture

Hardware includes:

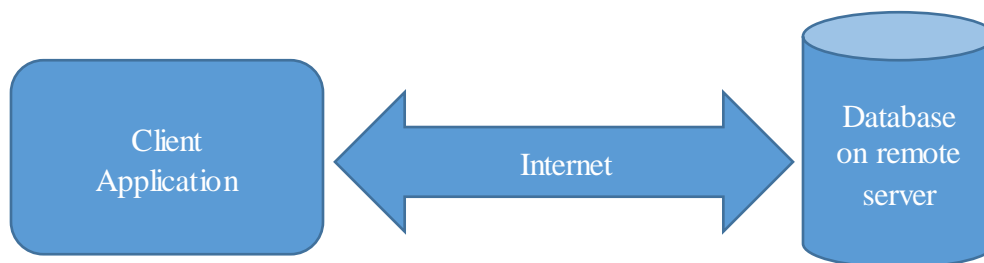
1. A Personal Computer (PC) running the operating system windows 10 for each register needed. This device must also have access to the internet.
 - Each of these systems includes: The computer “tower” itself
 - A monitor capable of at least resolution 1024x768
 - A keyboard and mouse
 - A printer
2. Currently the database is on a remoted server hosted by hosting24.comd. But in future, the database is going to be on company’s dedicated server for running the remote inventory database

3.2 System Software Architecture



Detailed design of software using UML is in section 5.2.

3.3 Internal Communications Architecture



4 FILE AND DATABASE DESIGN

This section details final design of all database management system (DBMS) files and data structures associated with the system under development.

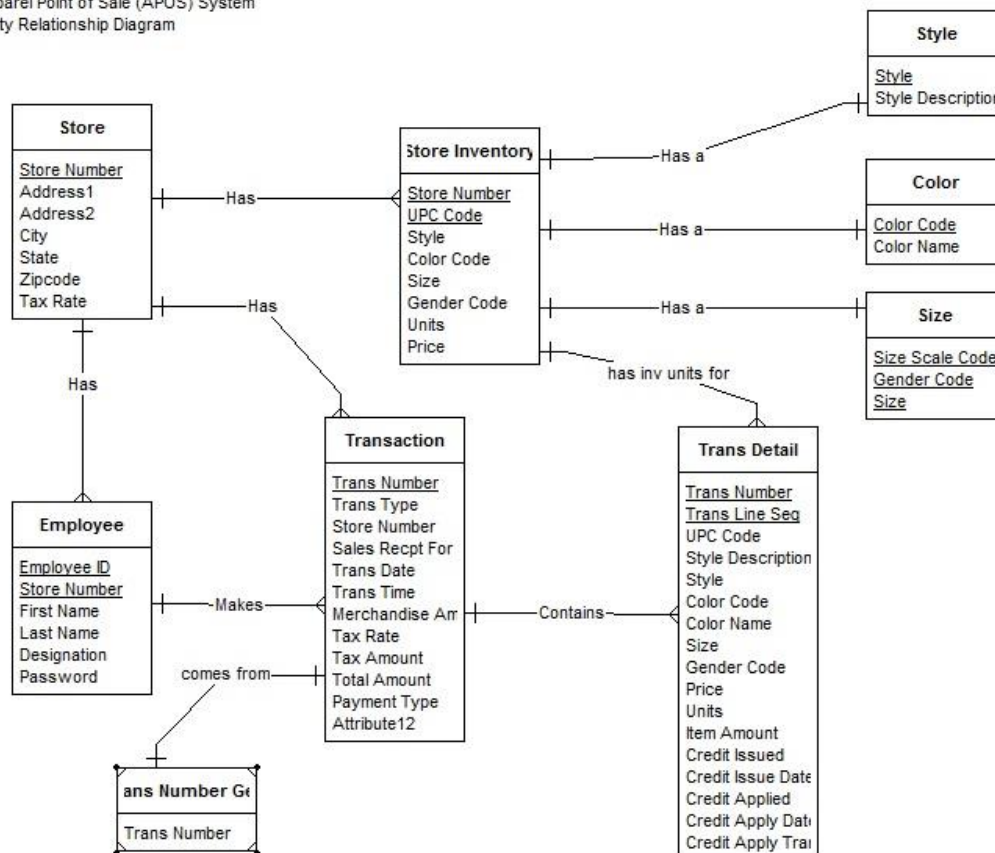
4.1 Database Management System Files

This section reveals the final design of the DBMS files and includes the following information.

4.1.1 Entity Relationship Diagram (ERD)

The following ERD diagram shows the data entities and their relationship to each other such as one to one, one to many, and many to many.

Apparel Point of Sale (APOS) System
Entity Relationship Diagram



4.1.2 Database Tables

1. store_master: Contains store attributes. Table size will be maximum 1000 rows.
2. employee_master: Contains employee attributes. Table size will be maximum 40000 rows.
3. style_master: Contains style attributes. Table size will be maximum 10000 rows.
4. color_master: Contains color attributes. Table size will be maximum 500 rows.
5. store_inventory: Contains store inventory items with price and units. Maximum 10000 rows.
6. transaction_header: Contains transaction header level information such as employee id, store, total sales amount, tax amount, etc. Maximum 50,000 rows (Archived beyond this).
7. transaction_detail: Contains transaction detail level information such as item, price, units, and item amount.
8. transaction_number: This table is for incrementing transaction number for transactions. Only one row.

4.1.3 Table Schema

Double click on the Excel sheet to view the table schema list. In case Excel does not open, please see Appendix A.

Database Tables						
Table Name	Description	Columns	Column Desc	Column type	Primary Key Index	
store_master	Contains store information	store_num	Unique store number	int	store_num	store_num
		store_address1	Store address – line 1	Varchar(50)		
		store_address2	Store address – line 2	Varchar(50)		
		store_city	City	Varchar(30)		
		store_state	State	Varchar(2)		
		store_zipcode	Zipcode	Varchar(10)		
		tax_rate	State Tax Rate	Decimal(10,2)		
		create_datetime	Created date and time	Date time		
		create_user	Created user	Varchar(10)		
		update_datetime	Updated date and time	Date time		
		update_user	Updated user	Varchar(10)		

4.2 Non-Database Management System Files

This section contains data structure for the system. Please double click on the Excel sheet to view data structure. If Excel does not open, please go to Appendix B to see data structure.

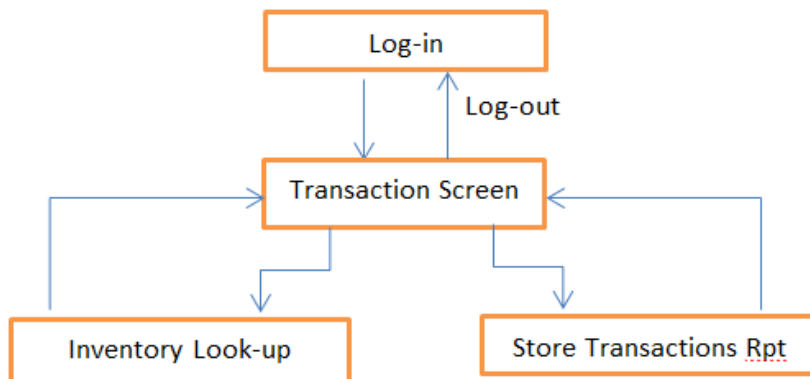
Data Structures			
Type	Name	Object	Attributes
Vector	vetctorTransItems	TransactionItem	UPC code
			Style
			Style Name
			Color
			Color Name
			Size

5 HUMAN-MACHINE INTERFACE

This section provides the detailed design of the system and subsystem inputs and outputs relative to the user/operator.

5.1 Inputs

GUI Navigation:



Screen Navigation

5.1.1 Log-in Screen

Upon launching the program, the user will be taken to the Log-in screen. This frame includes the name of the company, and will prompt the user for their store number, their employee log-in ID number, and their password using text fields and a password text field, as appropriate. This screen will also include a “Log-in” button. Upon clicking the button, a method will be called that attempts to verify the credentials entered by the user with those stored on the database. If the credentials do not match, an error is displayed that indicates incorrect credentials, and the text fields are cleared to allow for another attempt. If correct credentials are entered, the program progresses to the transaction screen.

See image below

A screenshot of a web application window titled "Tenacious Turtles Clothing Company". The window has a light blue background. In the center, there is a login form. The form consists of three text input fields stacked vertically, each preceded by a label: "Store Number:", "Employee Log-in ID:", and "Password:". Below these fields is a "Log-in" button. The window has standard OS window controls (minimize, maximize, close) in the top right corner.

Error Messages:

Error 1.1 - Invalid Store Number

When a user enters an invalid Store Number on Log-in screen, this error will appear. To resolve this error, enter correct Store Number.

Error 1.2 - Invalid Log-in ID or Password

When a user enters the wrong username or password, this error will appear. To resolve this error, ensure the user has the right login information available and inputs it correctly.

5.1.2 Transaction Screen

The transaction screen displays along the top the following information:

1. Store number
2. Store address
3. Currently logged-in employee
4. The employee's designation (Cashier, Assistant Manager, etc.)

Included near the top of the window is also a set of 2 radio buttons that the employee can select based on which type of transaction they wish to complete, either sales or returns.

Next to the returns radio button there is a text field where the employee can enter the original sales receipt that a customer has presented during a return transaction. If "returns" is not selected, this field is un-editable.

Below this upper section is the main table for item entry. To the left is an editable column where the employee enters the UPC code for items the customer wishes to purchase or return. Upon entering the UPC code and pressing enter, a method call queries the remote database in order to populate the other fields: style, style name, color, color name, size, gender, and price. The employee will then need to enter the quantity and press enter again. The program then calculates the final column, amount (the total price for all items of that type). It will then move to the next line and back to the UPC code column to allow for the next item entry.

Under this table are located several buttons:

- Complete Transaction

This button finishes either the sale or the return transaction that was entered.

Sales Transaction:

[illegible]

Returns Transaction:

[illegible]

Error messages:

Error 2.1 – Invalid Item Entered

When a user enters an invalid UPC number for an item, this error will appear. To resolve this error, enter the correct UPC code.

Error 2.2 - Item not available in inventory.

After entering units, the system checks store inventory for item availability for the entered

units. If store inventory does not have enough units, this error will appear. To resolve this error, enter units available by doing inventory look-up operation.

Error 2.3 – Invalid sales receipt for returns

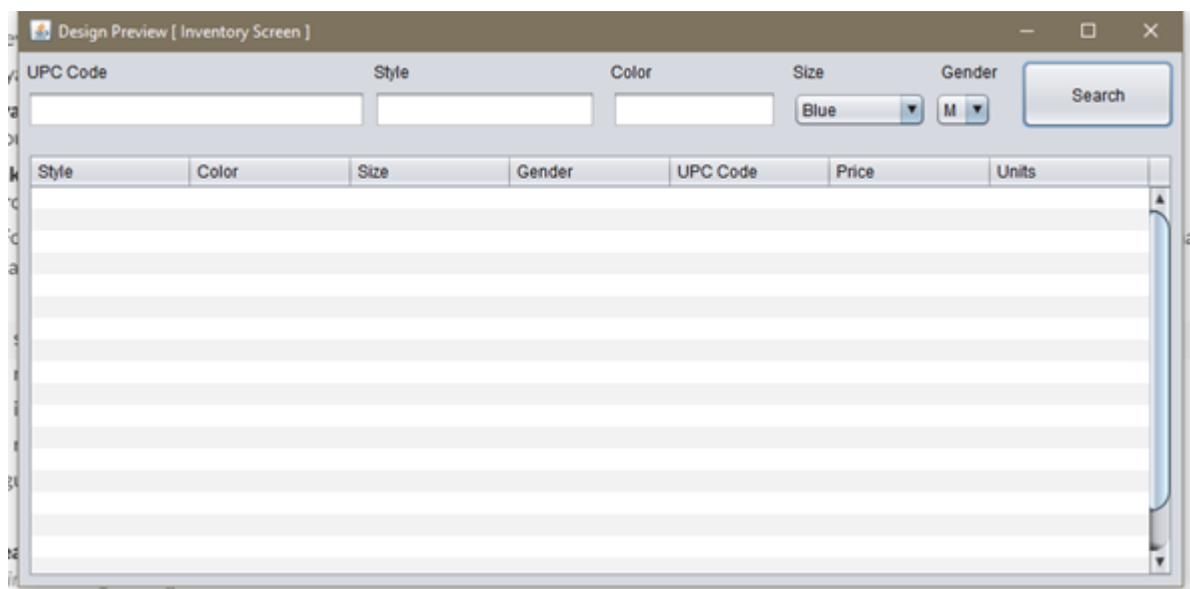
When a user enters an invalid sales receipt number for returns, this error will appear. To resolve this error, enter the correct sales receipt number.

Error 2.4 – Not items to return

After entering a valid sales receipt number for returns, if all items in that sales receipt are already returned, then this error message appears.

5.1.3 Inventory Look-up Screen

From transaction screen, the user can access inventory lookup screen. The user will need to input the UPC code and any other identifying information (Style, Color, Size, and Gender) for the product they are inquiring about and then selecting the search button that is provided. See image below.



5.1.4 Transaction Report Criteria Screen

From Transaction screen, user can access Transaction Report Criteria screen. The user will need to enter a date range and then any of the other fields that apply to the situational report. Once all of the appropriate fields have been filled in, the user needs to select the Generate Report button located at the bottom of the menu. See image below



The screenshot shows a software window titled "Tenacious Turtles Clothing Company". Inside the window, the title "Store Transaction Report" is displayed. Below the title, there are several input fields for filtering transactions: "Date Range:" with two adjacent text boxes, "Transaction Type:" with a dropdown menu currently showing "All", "UPC Code:" with a single text box, "Style:" with a single text box, "Color:" with a single text box, "Size:" with a single text box, "Gender:" with a single text box, and "Associate:" with a single text box. At the bottom center of the form is a button labeled "Generate Report".

5.2 Outputs

This system includes the following outputs:

1. Monitor output. Required to show user the GUI for the software. This GUI includes a login screen and a transaction screen that includes a myriad of functions. Accessing the transaction screen requires valid login credentials to be entered in the login screen first.

2. Upon completion of both sales and return transactions, a receipt is printed for both the customer and the store's records. Sales receipts are needed by customers for

<p>Tenacious Turtles Clothing Co. Store #: 10001 1234 Main Street Adelphi, MD 20783 321-123-4567</p> <p>Transaction #: 000000001 Date: 09/06/2016 04:22:37 PM Payment Type: Cash</p> <table> <thead> <tr> <th>QTY</th> <th>ITEM</th> <th>SIZE</th> <th>PRICE</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>red M Shorts</td> <td>32</td> <td>15.99</td> </tr> <tr> <td>2</td> <td>grn G skirt</td> <td>S</td> <td>06.99</td> </tr> <tr> <td colspan="2">3 Items</td> <td>Sub-Total:</td> <td>29.97</td> </tr> <tr> <td colspan="2">Tax 6%: 1.80</td> <td>Total:</td> <td>31.77</td> </tr> </tbody> </table> <p>Thank you for shopping at Tenacious Turtles Clothing Co.!</p>				QTY	ITEM	SIZE	PRICE	1	red M Shorts	32	15.99	2	grn G skirt	S	06.99	3 Items		Sub-Total:	29.97	Tax 6%: 1.80		Total:	31.77	<p>Tenacious Turtles Clothing Co. Store #: 10001 1234 Main Street Adelphi, MD 20783 321-123-4567</p> <p>RETURN RECEIPT</p> <p>Original Transaction #: 000000001 Return Transaction #: 000000023</p> <p>Return Date: 09/13/2016 09:15:04 AM</p> <table> <thead> <tr> <th>QTY</th> <th>ITEM</th> <th>SIZE</th> <th>PRICE</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>red M Shorts</td> <td>32</td> <td>-15.99</td> </tr> <tr> <td>1</td> <td>grn G skirt</td> <td>S</td> <td>-06.99</td> </tr> <tr> <td colspan="2">2 Items</td> <td>Sub-Total:</td> <td>-22.98</td> </tr> <tr> <td colspan="2">Tax 6%: 1.38</td> <td>Total:</td> <td>-24.36</td> </tr> </tbody> </table> <p>Return payment type: Cash</p> <p>Thank you for shopping at Tenacious Turtles Clothing Co.!</p>				QTY	ITEM	SIZE	PRICE	1	red M Shorts	32	-15.99	1	grn G skirt	S	-06.99	2 Items		Sub-Total:	-22.98	Tax 6%: 1.38		Total:	-24.36
QTY	ITEM	SIZE	PRICE																																												
1	red M Shorts	32	15.99																																												
2	grn G skirt	S	06.99																																												
3 Items		Sub-Total:	29.97																																												
Tax 6%: 1.80		Total:	31.77																																												
QTY	ITEM	SIZE	PRICE																																												
1	red M Shorts	32	-15.99																																												
1	grn G skirt	S	-06.99																																												
2 Items		Sub-Total:	-22.98																																												
Tax 6%: 1.38		Total:	-24.36																																												

their own records or in the event they need to perform a return. The return receipts are for record keeping for both the store and the customer.

3. At the completion of the work day, or whenever appropriate, the system can print a Daily Report that shows a comprehensive list of the items sold, returned, and the totals for return and sales amounts. This report is used by managers to track sales and returns figures, as well as to perform store inventory to ensure that inventory matches sales.

Tenacious Turtles Clothing Company									
Store Transactions Report									
Store: 10001 1234 Main Street, Adelphi, MD 20783							Report Date: 09/09/16		
Seq	Style	Style Name	Color	Size	Gender	UPC Code	Price	Units	Amount

Receipt#: 10010001		Associate: Dan Edmonston		Trans Type: S		Receipt#: 38922000129 for returns		Date:09/09/16	
1	A10001	Long Sleeve Shirt	BLK	XL	M	38922000129	65.99	2	131.98
2	C10025	Skirt	BLU	S	W	38929000013	115.15	1	115.15
							Sub Total:		3 247.13
Tax: 14.83							Sales Amt:		261.96

Receipt#: 10010002		Associate: Larry David		Trans Type: R		Receipt#: 10010001 for returns		Date:09/09/16	
1	A10001	Long Sleeve Shirt	BLK	XL	M	38922000129	65.99	2	-131.98
2	C10025	Skirt	BLU	S	W	38929000013	115.15	1	-115.15
							Sub Total:		3 -247.13
Tax: -14.83							Sales Amt:		-261.96

Total sales:		3	261.96	Total returns:		3	261.96		

6 DETAILED DESIGN

This section provides the information needed for our team to actually build and integrate the hardware components, code and integrates the software modules, and interconnects the hardware and software segments into a functional product.

6.1 Hardware Detailed Design

All devices must be located where they can be plugged into and operate from the standard 120V wall outlet, if applicable. If needed, an appropriate power strip should be installed to allow all the needed devices to be located in the appropriate register location. All register computers must have internet access. The needed devices and their requirements are as follows:

1. The PC should be capable of running windows 10 (32Bit) sufficiently. A store-bought PC will typically suffice, however the recommended requirements to run the software sufficiently are as follows:
 - Dual-Core Processor of at least 1GHz with integrated graphics capability
 - Power supply of at least 450W that functions on a 120V outlet
 - At least 2GB of RAM
 - A hard drive of at least 50GB
 - Must provide a DVA output and inputs for a keyboard, mouse, and at least 2 USB ports for the receipt and report printers
2. The monitor should be a color display of at least resolution 1024x768. A greater resolution is acceptable, with a display ratio of 4:3. Monitor is recommended to have screen size of 14" to 16". Must function on standard 120V outlet.
3. A standard, 2-button optical mouse for navigating both the Windows operating system before the program run, and the menus of the program itself. Must be compatible with the mouse input on the PC.
4. A standard keyboard that is compatible with the keyboard input on the PC. While not required, having a numerical pad on the keyboard greatly facilitates number entering by employees, and should be included if possible.
5. Receipt printer capable of printing receipts of the standard width of 2 1/4". This device must be able to connect to the register PC through a USB input. Must also operate on the standard 120V outlet.

6. Standard printer that is capable of printing full sheets (8 ½” by 11”) for printing transaction reports. This printer can be located away from the register if needed, such as in the manager’s office. This printer must be able to connect to at least 1 register, either physically or wirelessly (if both the printer and the PC support this function). If a wireless printer is used, it should be set up in a secure fashion so that unauthorized individuals cannot access this device. Must operate on a standard 120V outlet.
7. Currently the database is hosted remotely and accessed through the Internet. If deciding to host the database on a server for storing and handling the inventory database, that PC must meet or exceed the requirements of the PC indicated above. It is also essential for this device to have internet access at all times, and should have security measures in place to prevent unauthorized access, both digitally and physically.
8. Graphical representation depicting the number of hardware items (for example, monitors, printers, servers, I/O devices), and the relative positioning of the components to each other

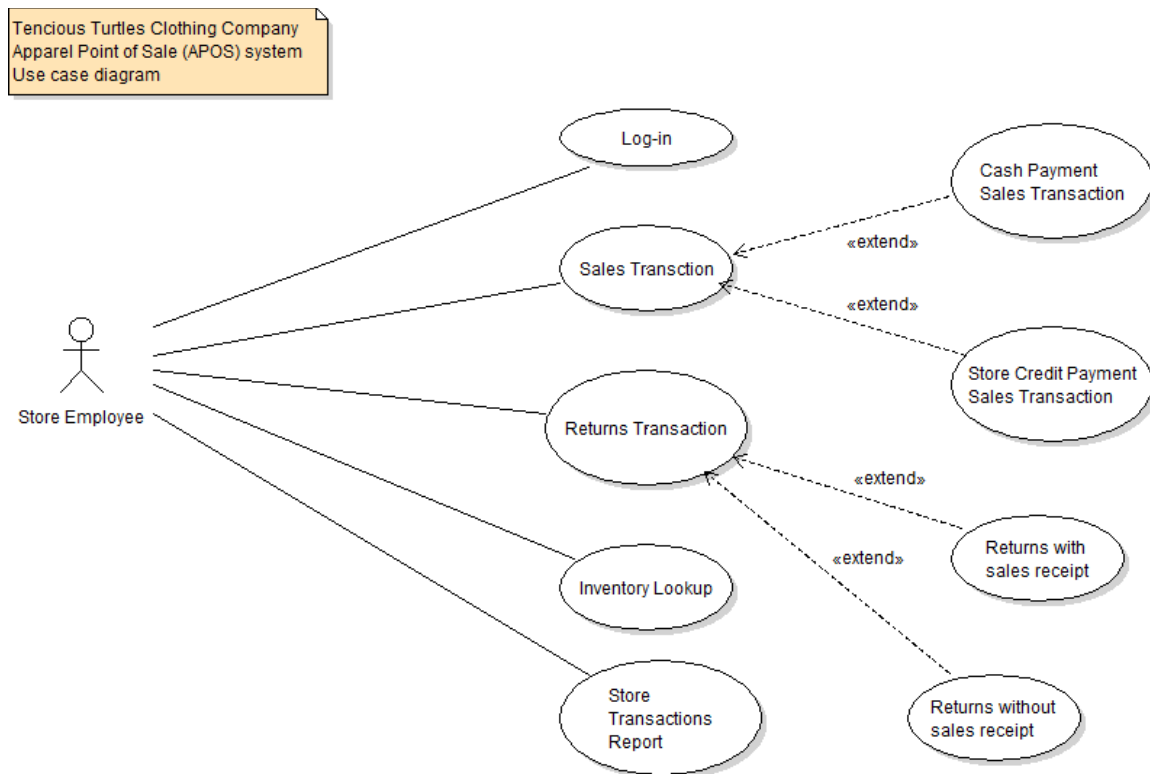
6.2 Software Detailed Design

This section has complete Object Oriented Analysis and Design (OOAD) model using UML.

The phases in OOAD are already explained in section 1.2.4 Overall Approach.

The following are each phases in OOAD:

6.2.1 Use Case Diagram



6.2.2 Use Case Scenarios

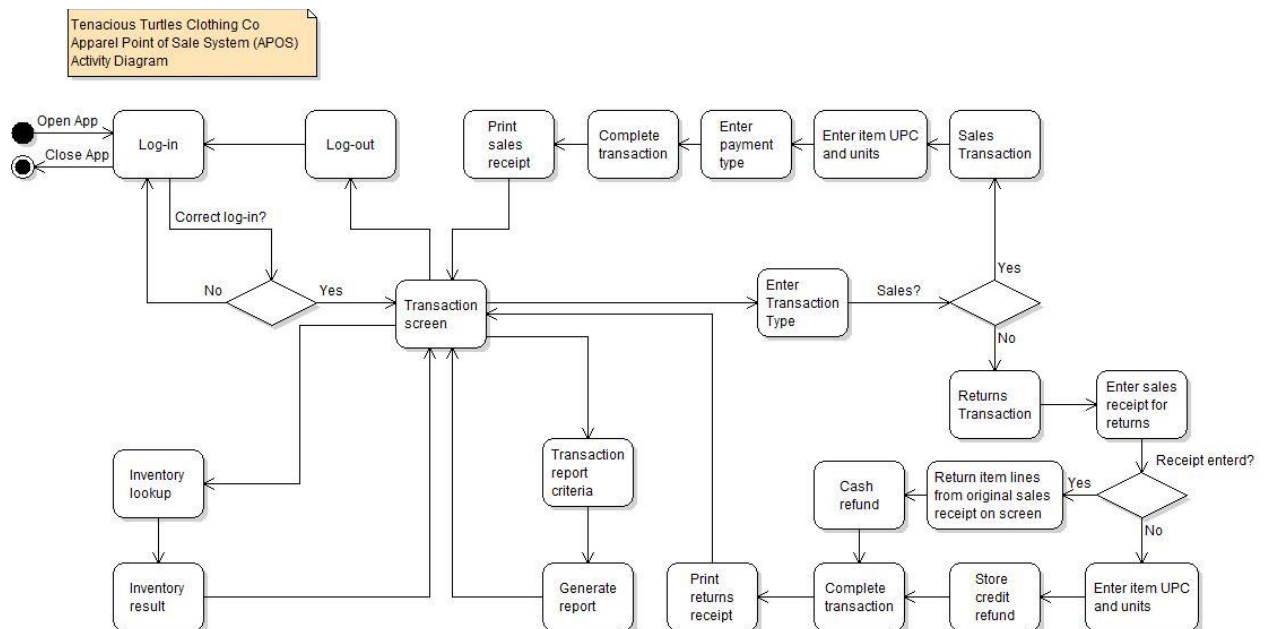
Requirement ID	Description	Scenarios / Use Cases
SR01	System will provide employee login/logout interface.	SR01: Employee log-in - Employee enters login credential - System validates credentials - System differentiates between employee and

		manager - Employee logs in
SR02	System will process sales transactions (cash and store credit only). This will include updating store inventory and printing receipts.	SR02: Employee completes sales transaction - Employee enters UPC and Units into sales interface - System queries database for inventory information - Database responds with inventory and price information - Employee completes transaction - System assigns receipt number to sales transaction in database - System displays/prints receipts for store and customer - System adjusts store inventory
SR03	System will process product returns that will result in cash back or store credit.	SR03: Employee completes returns transaction - Employee enters receipt number - Employee enters item information into sales interface if sales receipt is not available - If valid sales receipt, cash refund. Otherwise, store credit. - Employee completes transaction - System assigns receipt number to returns transaction in database - System displays/prints receipts for store and customer - System adjust store inventory

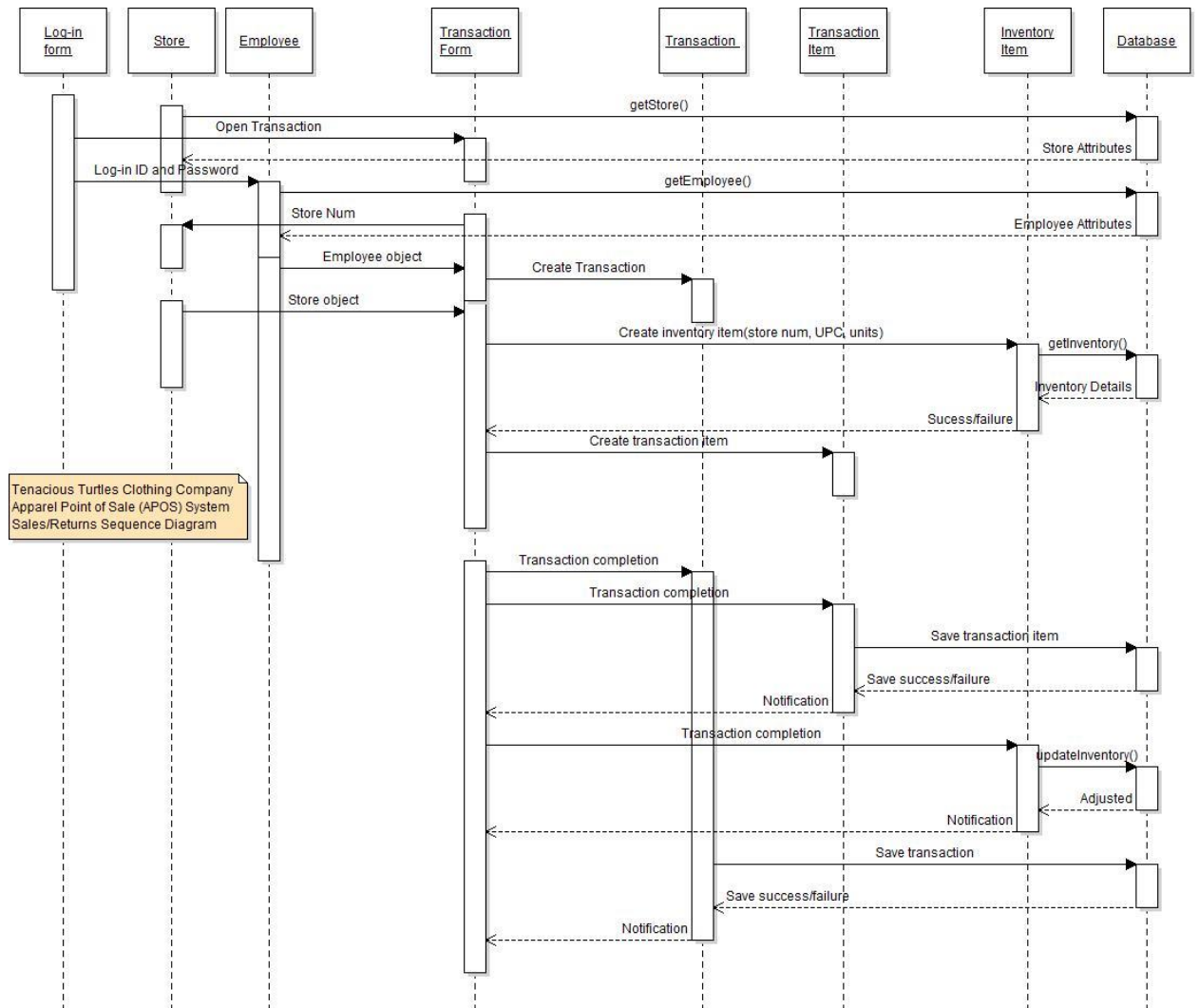
SR04	System will display or print daily sales transactions.	SR04: Store manager checks sales transactions - System checks employee designation, only displays option if employee logged in as manager - System allows to enter report criteria - System generates report
SR05	System will be able to query and review store inventory information.	SR05: Employee checks inventory - Search by item, gender, etc. - System queries back-end to get inventory information - System reports current inventory status
SR06	System will provide log-out functionality.	SR06: Employee logs out

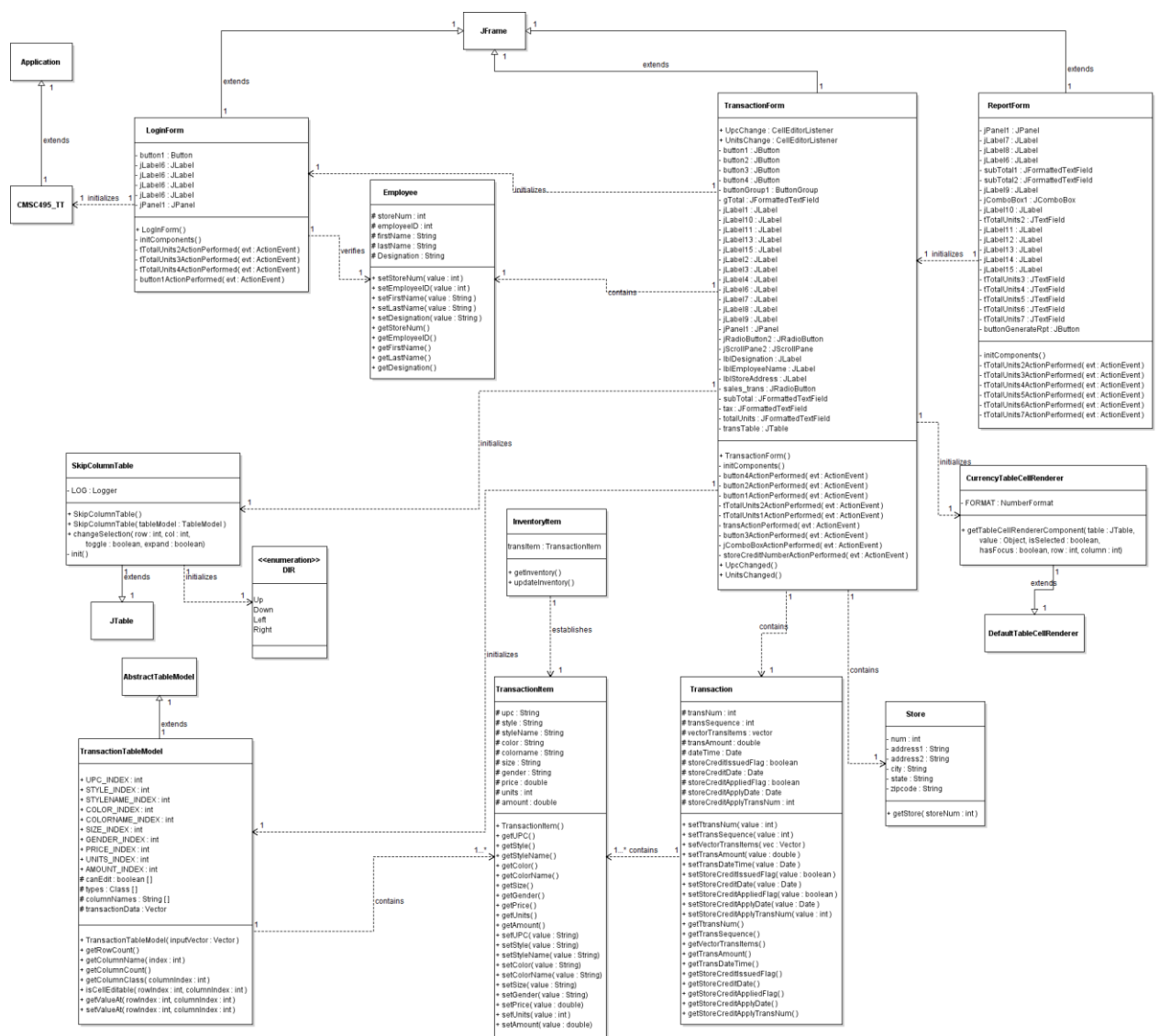
6.2.3 Activity Diagram

This has all activity identified in use cases and not involving database access in the application.



6.2.4 Sequence Diagram

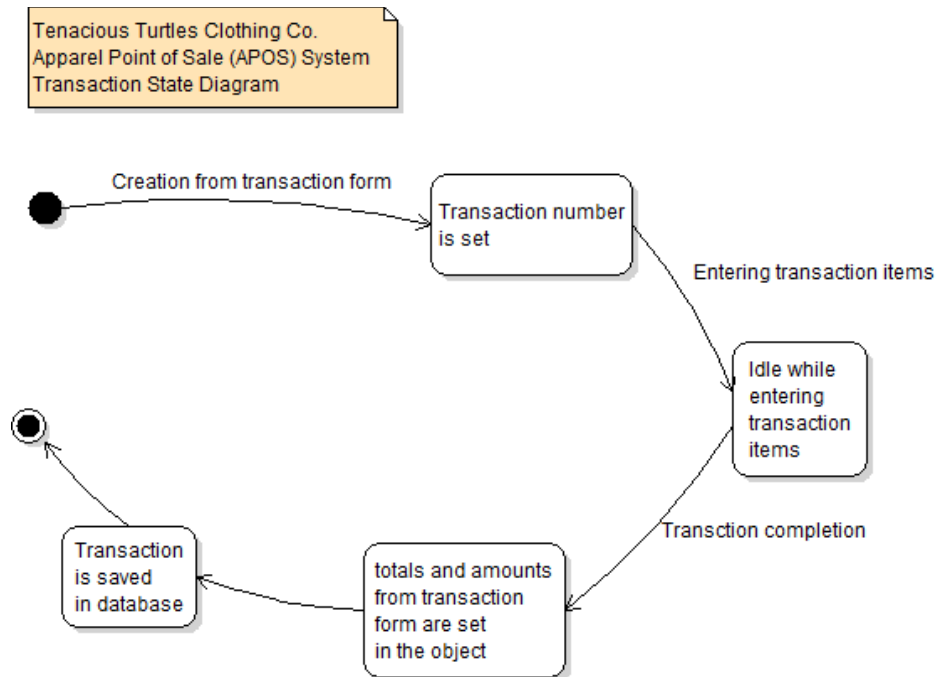




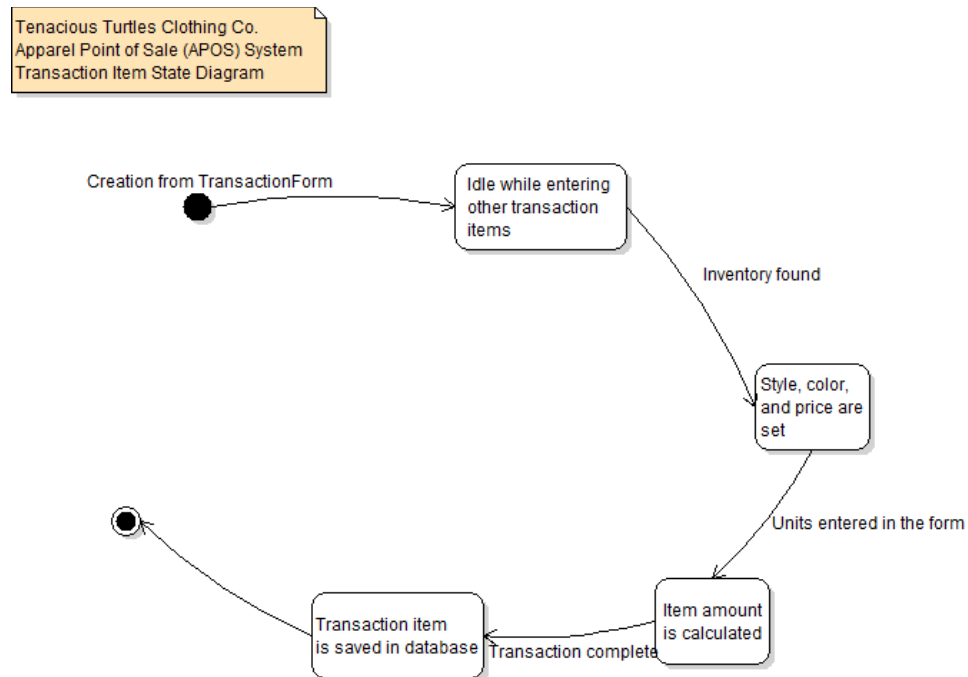
6.2.6 State Chart Diagrams

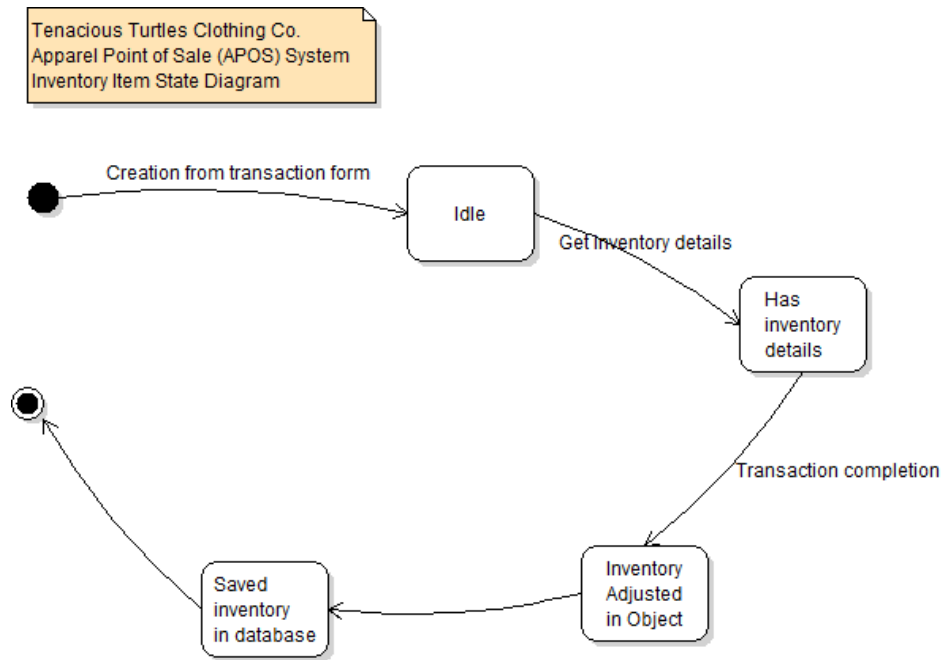
This section has state chart diagrams of three important objects: transaction, transaction item, and inventory item.

Transaction Object:

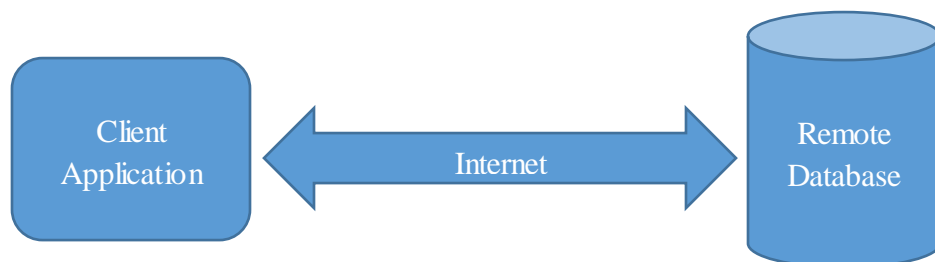


Transaction Item Object:



Inventory Item Object:**6.3 Internal Communications Detailed Design**

The client software in store and the database on remote server have to communicate through the Internet. A high speed internet connection is required for better performance. The IP address of the client system has to be registered on the remote database server for security reasons. The client systems in stores are not connected through Local Area Network (LAN).



7 EXTERNAL INTERFACES

APOS does not have any external interfaces. Company's ERP system and the Personnel Management systems are external. But these systems populate master table information needed for the system directly in the APOS database. Store inventory in the database is replenished by the ERP system directly. Also, company's ERP system loads store transactions and payments directly from APOS database. The application itself does not provide these interfaces.

7.1 Interface Architecture

Not applicable

7.2 Interface Detailed Design

Not applicable

8 SYSTEM INTEGRITY CONTROLS

The APOS system will be requiring all users to authenticate to the system through via a log-in. This will verify what role the user/s have with the system (clerk or manager) which will in turn, identify what system actions they are able to perform and what store data they are able to view or modify. Users with no log-in credentials will not be able to use any of the APOS system functions or access any of the data stored within. User access control to the system will be audited and maintained by APOS system administrators.

Internally, the APOS system has identified the appropriate requirements and/or datatypes that will be accepted as input from its external interfaces. If incoming data does not match these parameters, it will not be recorded or processed within the system until corrections have been made. Internal data constraints will also be utilized to ensure any information that is critical to system functionality is fully retrieved before the storing or processing of the data takes place.

As needed, APOS system administrators will review audit events and logs pertaining to functionality of the APOS system. This audit data will be able to provide the specifics (date, time, data modified...etc.) associated with these events. These reviews will be used by the administrators to highlight and correct any data modification issues that may impact the integrity of the data being stored and processed within the system. Any integrity issues that are discovered

will be documented and reported to the appropriate developers within the Tenacious Turtles team to aid in immediate correction. Documented issues will be retained for a specified timeframe so that they can be incorporated into future APOS system versions or updates.

9 PERFORMANCE ESTIMATES

9.1 Performance of APOS system functions

System Log-in – Client system queries database for existence of user record, user is presented with transaction screen. Estimated completion time: 1-3 secs

Sales transaction – Client system processes transaction, modifies store inventory and generates receipt. Estimated completion time: 2-5 secs

Return transaction – Client system processes transaction, modifies store inventory and generates receipt. Estimated completion time: 2-5 secs

Review of store inventory – Client system queries database for existence of apparel records and displays results. Estimated completion time: 2-5 secs

Generation of sales report – Client system queries database for existence of sales records and generates visual report. Estimated completion time: 2-5 secs

9.2 Performance Factors

The performance of the APOS system will be dependent on several factors that are described below:

Client system resources: 2 GB RAM with minimal multitasking which should provide a surplus of system resources to easily handle any APOS system request. Adding additional systems or applications to the client system may eventually degrade performance.

Network connectivity between client and database: Acquiring high speed internet connection for this purpose. Outages or internet connection issues may temporarily impact performance of APOS system.

Number of users using APOS system at the same time: Only one user will be using the system at once. Simultaneous use of the system will negatively impact the availability of system resources.

The volume of data in the database: The efficiency of the database is also a factor. The database is common for all stores. So by increasing number of stores and transactions, the data volume

will also increase. When volume increases, data access time will also increase. However, with proper indexing this can be avoided up to some extent. The database used in this project is MySQL hosted by hosting24.com. So optimizing database is beyond the scope of this project.

10 APPENDIX

A. Database table schema

Database Tables						
Table Name	Description	Columns	Column Desc	Column type	Primary Key	Index
store_master	Contains store information	store_num	Unique store number	int	store_num	store_num
		store_address1	Store address – line 1	Varchar(50)		
		store_address2	Store address – line 2	Varchar(50)		
		store_city	City	Varchar(30)		
		store_state	State	Varchar(2)		
		store_zipcode	Zipcode	Varchar(10)		
		tax_rate	State Tax Rate	Decimal(10,2)		
		create_datetime	Created date and time	Date time		
		create_user	Created user	Varchar(10)		
		update_datetime	Updated date and time	Date time		
		update_user	Updated user	Varchar(10)		
employee_master	Contains employee master information	store_num	Store number from Store Master	int	store_num	store_num
		employee_id	Unique employee ID	Varchar(10)	employee_id	employee_id
		employee_first_name	Employee first name	Varchar(30)		
		employee_last_name	Employee last name	Varchar(30)		
		employee_designation	Designation of employee	Varchar(20)		
		employee_password	Encrypted password	Varchar(30)		
		create_datetime	Created date and time	Date time		
		create_user	Created user	Varchar(10)		
		update_datetime	Updated date and time	Date time		
		update_user	Updated user	Varchar(10)		
style_master	Contains style information	style	Unique style code	Varchar(10)	style	style
		style_description	Description of the style	Varchar(50)		
		create_datetime	Created date and time	Date time		
		create_user	Created user	Varchar(10)		
		update_datetime	Updated date and time	Date time		
		update_user	Updated user	Varchar(10)		

color_master	Contains color information	color_code	Unique color code	Varchar(10)	color_code	color_code
		color_name	Description of color	Varchar(20)		
		create_datetime	Created date and time	Date time		
		create_user	Created user	Varchar(10)		
		update_datetime	Updated date and time	Date time		
		update_user	Updated user	Varchar(10)		
size_master	Contains size information	size_code	Size code for size ranges	Varchar(2)		
		gender_code	Gender code	Varchar(1)	gender_code	gender_code
		size	size	Varchar(10)	size	size
		create_datetime	Created date and time	Date time		
		create_user	Created user	Varchar(10)		
		update_datetime	Updated date and time	Date time		
		update_user	Updated user	Varchar(10)		
store_inventory	Contains store inventory items	store_num	Store number from Store Master	int	store_num	store_num
		upc_code	UPC Code	Char(12)	upc_code	upc_code
		style	Style code	Varchar(10)		
		color_code	Color code	Varchar(10)		
		size	size	Varchar(10)		
		gender_code	Gender code	Varchar(1)		
		units	Units in inventory	int		
		price	Price of the item	Decimal(10,2)		
		create_datetime	Created date and time	Date time		
		create_user	Created user	Varchar(10)		
		update_datetime	Updated date and time	Date time		
		update_user	Updated user	Varchar(10)		
transaction_header	Contains transaction header information	trans_num	Unique transaction number / receipt number	int	trans_num	trans_num
		trans_type	Transaction type: Returns / Sales	Varchar(6)		
		store_num	Store number from Store Master	int		
		sales_receipt_for_ret	Sales receipt number for returns	int		
		trans_date	transaction date	Date		
		trans_time	transaction time	Time		
		trans_merch_amt	Merchandise Amount – sub total	Decimal(12,2)		
		trans_tax_rate	Tax rate	Decimal(10,2)		

		trans_tax_amount	Tax amount	Decimal(12,2)		
		trans_total_amount	Total amount	Decimal(12,2)		
		payment_type	Payment type: Cash / Store credit	Varchar(15)		
		create_datetime	Created date and time	Date time		
		create_user	Created user	Varchar(10)		
		update_datetime	Updated date and time	Date time		
		update_user	Updated user	Varchar(10)		
transaction_detail	Contains transaction item details	trans_num	Transaction number	int	trans_num	trans_num
		trans_seq	Sequence number within transaction	int	trans_seq	trans_seq
		upc_code	UPC Code	Varchar(12)		
		style	Style code	Varchar(10)		
		style_description	Style Description	Varchar(50)		
		color_code	Color code	Varchar(10)		
		color_name	Color description	Varchar(20)		
		size	size	Varchar(10)		
		price	Price of the item	Decimal(10,2)		
		units	Number of units sold / returned	int		
		trans_item_amount	Line amount	Decimal(12,2)		
		store_credit_issued	Store credit issue flag	Boolean		
		store_credit_issue_date	Store credit issue date	Date		
		store_credit_applied	Store credit apply flag	Boolean		
		store_credit_apply_date	Store credit apply date	Date		
		store_credit_apply_trans_num	Store credit applied transaction number	Int		
		create_datetime	Created date and time	Date time		
		create_user	Created user	Varchar(10)		
		update_datetime	Updated date and time	Date time		
		update_user	Updated user	Varchar(10)		
transaction_numbers	Contains transaction number to increment	trans_num	Transaction number to increment	int		

B. Data Structure

Data Structures			
Type	Name	Object	Attributes
Vector	vectorTransItems	TransactionItem	UPC code
			Style
			Style Name
			Color
			Color Name
			Size
			Gender
			Price
			Units
			Total Amount
Vector	vectorIntenvoryItem	InventoryItem	Store Number
			UPC Code
			Style
			Style Description
			Color
			Color Name
			Price
			Units