

Software Requirements Specification Version 1

Point of Sale Management System

Team 6

Member Name	Member Roll #	Primary Responsibility
Ali Imran	15L-4148	Back-End Programming
Farhan Shoukat	15L-4292	Back-End Programming
Kamal Subhani	15L-4048	Database Programming
Hassan Alvi	15L-4203	Database Programming
Abdullah Asad	15L-4063	User Interface Development

Table of Contents

1. Introduction	4
1.1 Product	4
1.2 Scope	4
1.3 Business Goals	4
1.4 Document Conventions	5
1.5 References	5
2. Functional Requirements.....	6
2.1 Login	6
2.2 Change Password	7
2.3 Order Print.....	8
2.4 Update Details of Person.....	9
2.5 Approval of Changes in Product	9
2.6 Approval of Changes in Person.....	10
2.7 Change Inventory Item Details	11
2.8 Search the Person	11
2.9 View Information of the Person	12
2.10 Search Information of Product	13
2.11 View Monthly/Yearly Sales	13
2.12 View Information of Inventory Items.....	14
2.13 Delete Item Details	15
2.14 Delete Person Details.....	15
2.15 Remove User Access	16
2.16 Addition of Inventory Item.....	17
2.17 Addition of Person.....	18
3. Nonfunctional Requirements.....	19
3.1 Performance Requirements	19
3.2 Security Requirements	19
4. Other Requirements	20
4.1 Software Requirements	20
4.2 Hardware Requirements	20

Revision History

Name(s)	Date	Reason(s) For Change(s)	Version

1. Introduction

1.1 Product

Point of Sale Management System includes the hardware and software used in tandem to place orders, track sales, manage inventory and track revenue and profits. However, our product is only software based and is simply a POS desktop application designed to ensure a smooth transaction process between the retailer and the customer. Moreover, it will also effectively store all customers and sales data for further analysis like visualizing sales data or optimize pricing and also to check the sales trends of various products. These kind of applications are mainly used in small businesses such as departmental stores or supermarkets but they can be an efficient solution for most of the companies that maintain data in some specific sequence and require information regarding sales management.

1.2 Scope

The scope of the functionalities of this application revolves around the user i.e. the retailer in this case. Firstly, the user would have to login in order to gain access. After a successful login, the main interface will show where the user is given the option to provide customer and product details which adds them into the database. Moreover, the user will be given the options to either add a new customer or a product and also to update product information. Sales person is restricted only towards taking orders, updating his password and registration of customers into the database. Manager would have the access to approve the entry of the person or the product as well as perform actions of the sales person. Admin has most of the rights where he/she can perform the functionalities which are pretty basic yet sensitive enough for anyone to alter except the administrator such as add, update and delete. The application focuses towards having numerous panels to access multiple features offline such as summary of sale records, manage products and person information, searching records saved in the past etc.

1.3 Business Goals

This software will have huge impact on the business strategy of the client. The business goals which are expected to be achieved are listed below:

- Increased operating margins
- Quicker execution of company strategy
- Reduced employee turnover
- Establish a more effective and efficient goal-setting process
- Exponentially speed completion of company goals
- Boost employee cooperation company-wide with cascading goals
- Gain powerful insight by using collected data more effectively

1.4 Document Conventions

Conventions followed within this document are:

- For Title: Arial font, 32 font size and bold font style was used
- For Headings: Times New Roman font, 18 font size and bold font style was used.
- For Subheadings: Times New Roman font, 14 font size and bold font style was used.
- For Paragraphs: Times New Roman font, 11 font size was used.
- For Main terms: Times New Roman font, 11 font size and bold font style was used.

All text are in justified alignment. Bullet points have also been used where a list of features or description was required. Images are attached in the Glossary B for further explanation of the system.

1.5 References

- [1] "freshworks," [Online]. Available: <https://www.freshworks.com/freshsales-crm/solutions/sales-management-system/>.
- [2] "INSIGHTSQUARED," [Online]. Available: <https://www.insightsquared.com/solutions/sales-management-system/>.
- [3] "Software Suggest," [Online]. Available: <https://www.softwaresuggest.com/blog/basic-features-sales-management-software/>.
- [4] "3G Vision," [Online]. Available: <https://www.3gbusiness.com.au/blog/amazing-features-of-any-sales-management-software/>.
- [5] "Research Gate," [Online]. Available: https://www.researchgate.net/publication/307854899_Design_and_Development_of_a_Sales_Management_System_for_SMEs_in_Northern_Ghana.

2. Functional Requirements

2.1 Login

Identifier	UC-1	
Purpose	User can log into the system.	
Priority	High	
Actors	Administrator, Manager and Salesperson	
Pre-conditions	Application should be running.	
Post-conditions	None	
Typical Course of Action		
S#	Actor Action	System Response
1	Enters username	
2	Enters password	
3	Clicks "Submit" button	
4		Checks username existence in the database
5		Username found
6		Matches user's password from the database
7		Password matched
8		Go to the "Note Orders" screen
Alternate Course of Action (User not found)		
S#	Actor Action	System Response
6		Username not found
7		Display error message: User not found. Please try again."
Alternate Course of Action (Password Mismatch)		
7		Password did not match
8		Display error message: "Incorrect password. Please try again"

Table 1: UC-1

2.2 Change Password

Identifier	UC-2	
Purpose	User can change his/her own password.	
Priority	Medium	
Actors	Administrator, Manager and Salesperson	
Pre-conditions	User must be logged in.	
Post-conditions	None	
Typical Course of Action		
S#	Actor Action	System Response
1	Enters username	
2	Enters old password	
3	Enters new password	
4	Enters confirm new password	
5	Clicks "Submit" button	
6		Checks username in database
7		Username found
8		Matches old password from the database
9		Old password matched
10		New and confirm password compared
11		Both new passwords match
12		Updates password of the user
Alternate Course of Action (User not found)		
S#	Actor Action	System Response
7		Username not found
8		Display error message: User not found. Please try again."
Alternate Course of Action (Old Password mismatch)		
9		Old password did not match
10		Display error message: "Incorrect password. Please try again"

Alternate Course of Action (New Passwords mismatch)		
10		Both of the new passwords did not match
11		Display error message: "New password did not match the confirm new password. Please try again."

Table 1: UC-2

2.3 Order Print

Identifier	UC-3	
Purpose	User can print the order once it has been registered into the database successfully.	
Priority	High	
Actors	Administrator, Manager and Salesperson	
Pre-conditions	User must be logged in. Order should have been registered before.	
Post-conditions	None	
Typical Course of Action		
S#	Actor Action	System Response
1	Clicks “Print Receipt” Button	
2		Receives data from Order Details table
3		Checks if the information is available in the database
4		Arranges the data in receipt form
5		Shows print preview
6	Clicks on “Print” button	
7		Sends the receipt to printer
Alternate Course of Action (Information not found)		
S#	Actor Action	System Response
4		Prompts an error message

Table 1: UC-3

2.4 Update Details of Person

Identifier		UC-4
Purpose		To change personal details of the person.
Priority		Medium
Actors		Administrator
Pre-conditions		User must be logged in.
Post-conditions		None
Typical Course of Action		
S#	Actor Action	System Response
1	Searches for the person	
2		Displays the details of the person
3	Changes any of the person details (name, address, location, phone number, mobile number)	
4	Clicks "Submit" button	
5		Forward changes to Manager's panel for approval
6		Displays confirmation message
Alternate Course of Action (Information not found)		
S#	Actor Action	System Response
2		Information not found
3		Displays error message

Table 1: UC-4

2.5 Approval of Changes in Product

Identifier	UC-5
Purpose	To approve the changes in the product details made by the salesperson.
Priority	High
Actors	Manager
Pre-conditions	User must be logged in. There should be at least one product that needs to

	be approved.	
Post-conditions	None	
Typical Course of Action		
S#	Actor Action	System Response
1		Displays the list of inventory items
2	Selects the item	
3	Clicks “Approve Product” button	
4		Adds product details into the database
5		Displays confirmation message

Table 1: UC-5

2.6 Approval of Changes in Person

Identifier	UC-6	
Purpose	To approve the changes in the person details made by the salesperson.	
Priority	High	
Actors	Manager	
Pre-conditions	Application should be running. There should be at least one person who needs to be approved.	
Post-conditions	None	
Typical Course of Action		
S#	Actor Action	System Response
2		Displays the list of all the people
4	Selects a person	
5	Clicks "Approve Person" button	
6		Adds person details into the database
7		Displays confirmation message

Table 1: UC-6

2.7 Change Inventory Item Details

Identifier		UC-7
Purpose		User can update information of specific product.
Priority		Medium
Actors		Admin
Pre-conditions		User must be logged in.
Post-conditions		None
Typical Course of Action		
S#	Actor Action	System Response
1	Selects a product	
2		Shows item details to user
3	Clicks on Edit button	
4		Asks user to type in new values
5	Types in new values	
6	Clicks on submit button	
7		Prompts a message: "Item updated"
Alternate Course of Action 1 (No Product in Inventory)		
S#	Actor Action	System Response
2		Prompt a message: "No item in inventory"

Table 1: UC-7

2.8 Search the Person

Identifier		UC-8
Purpose		User can find a person in the database.
Priority		High
Actors		Admin, Manager
Pre-conditions		User is logged in.
Post-conditions		None
Typical Course of Action		
S#	Actor Action	System Response

1	Enters user's phone/mobile number	
2	Clicks on submit button	
3		Displays all the person information
Alternate Course of Action 1 (Unknown user phone number)		
S#	Actor Action	System Response
3		Prompt a message: "Person not found"
4	Clicks on Ok button	
5		Clears person phone number field
Alternate Course of Action 2 (Null phone)		
3		Prompts a message: "Please enter person's phone number"

Table 1: UC-8

2.9 View Information of the Person

Identifier	UC-9	
Purpose	User can view details of a person.	
Priority	Medium	
Actors	Admin, Manager	
Pre-conditions	User is logged in.	
Post-conditions	None	
Typical Course of Action		
S#	Actor Action	System Response
1	Clicks on "Person Information" tab	
2		Displays Person Name, Address, Mobile Phone and Payment Method
Alternate Course of Action 1 (No Person in Database)		
S#	Actor Action	System Response
2		Prompt a message: "No person in the database"

Table 1: UC-9

2.10 Search Information of Product

Identifier	UC-10	
Purpose	User can find a product in the inventory.	
Priority	High	
Actors	Admin	
Pre-conditions	User is logged in. Product search page is open.	
Post-conditions	None	
Typical Course of Action		
S#	Actor Action	System Response
1	Enters the name of item	
2	Clicks on submit button	
3		Displays product details
Alternate Course of Action 1 (Invalid Product Name)		
S#	Actor Action	System Response
3		Prompts a message: "Product not found"
Alternate Course of Action 2 (Null product name)		
3		Prompts a message: "Please enter product name"

Table 1: UC-10

2.11 View Monthly/Yearly Sales

Identifier	UC-11
Purpose	User can observe the statistics of all the sales
Priority	Medium
Actors	Admin, Manager
Pre-conditions	User is logged in. Sales Information Page is open.
Post-conditions	None
Typical Course of Action	

S#	Actor Action	System Response
1	Clicks "Sales Summary" menu item	
2		Display the total profit, total sales, total items sold, item names, quantities and month.
Alternate Course of Action 1 (No Sales ever made)		
S#	Actor Action	System Response
2		Prompt a message: "No sales have been made yet."

Table 1: UC-11

2.12 View Information of Inventory Items

Identifier	UC-12	
Purpose	User can view details related to product in inventory.	
Priority	Medium	
Actors	Admin, Manager	
Pre-conditions	User is logged in. Product Information Page is open	
Post-conditions	None	
Typical Course of Action		
S#	Actor Action	System Response
1	Selects product name from the drop down menu.	
2		Displays the total quantity ordered from the beginning, total cost, total profit, price of the product, quantity in stock.
Alternate Course of Action 1 (No Product in Database)		
S#	Actor Action	System Response
2		Prompt a message: "No product in the database"

Table 1: UC-12

2.13 Delete Item Details

Identifier	UC-13	
Purpose	Delete a specific product.	
Priority	Medium	
Actors	Administrator	
Pre-conditions	Product information is currently open.	
Post-conditions	None	
Typical Course of Action		
S#	Actor Action	System Response
1	Enters the product name	
2	Clicks the delete button	
3		Asks for the confirmation
4	Clicks on yes button	
5		Deletes the product
Alternate Course of Action 1 (Invalid Product ID)		
S#	Actor Action	System Response
2		Displays error message "No product found"
Alternate Course of Action 2 (User clicks on "No")		
4	Click on No button	
5		Clears the product name text field

Table 1: UC-13

2.14 Delete Person Details

Identifier	UC-14
Purpose	Delete information of a person permanently.
Priority	Low
Actors	Administrator
Pre-conditions	User information is currently open.
Post-conditions	None

Typical Course of Action		
S#	Actor Action	System Response
1	Enters the id of a person	
2	Clicks "Delete Person" button	
3		Ask for confirmation
4	Clicks on yes button	
5		Checks if the person's session is active
6		Deletes the information of the person
Alternate Course of Action 1 (Invalid User ID)		
S#	Actor Action	System Response
2		Display error message "No person found"
3		Clear the id field
Alternate Course of Action 2 (User clicks on "No")		
4	Clicks on no button	
5		Clear the id field

Table 1: UC-14

2.15 Remove User Access

Identifier	UC-15	
Purpose	Deletes the username of a specific user so that he/she cannot login but the data remains within the database.	
Priority	Low	
Actors	Administrator	
Pre-conditions	User information is currently open.	
Post-conditions	None	
Typical Course of Action		
S#	Actor Action	System Response
1	Enters id of a user	

2	Selects delete username button	
3		Asks for confirmation
4	Clicks on yes button	
5		Checks if the person's session is active
6		Deletes the information of the person username only
Alternate Course of Action 1 (Invalid User ID)		
S#	Actor Action	System Response
2		Displays error message "No user found"
3		Clears the user ID text field
Alternate Course of Action 2 (User clicks on "No")		
4	Clicks on no button	
5		Clears the id field

Table 1: UC-15

2.16 Addition of Inventory Item

Identifier	UC-16	
Purpose	Stores the new product within the database	
Priority	High	
Actors	Administrator	
Pre-conditions	Admin is logged in.	
Post-conditions	None	
Typical Course of Action		
S#	Actor Action	System Response
1	Enter product and its information	
2		Searches the product to find redundancy
3		Store product information
4		Prompt a message of successful entry

Alternate Course of Action 1 (Duplication)		
S#	Actor Action	System Response
3		Prompt the user a failed duplication message
Back to 1		

Table 1: UC-16

2.17 Addition of Person

Identifier	UC-17	
Purpose	Stores the new user inside the storage.	
Priority	High	
Actors	Administrator	
Pre-conditions	Admin is logged in.	
Post-conditions	None	
Typical Course of Action		
S#	Actor Action	System Response
1	Enter person and its information	
2		Searches the person to find redundancy
3		Store person information
4		Prompt a message of successful entry
Alternate Course of Action 1(Redundancy)		
S#	Actor Action	System Response
3		Prompt the user a failed duplication message

Table 1: UC-17

3. Nonfunctional Requirements

3.1 Performance Requirements

Performance requirement of POSS include:

- A user friendly user interface will be provided so that the user finds it easy to navigate whatever he/she is searching for.
- Allowing user to press enter will move user from one textbox to another. This way it will be convenient for the user to swiftly shift to another area of typing without going through the hassle of selecting the textbox via mouse
- Using combo boxes, in places such as: location or product, will help user in quickly selecting the specific option instead of typing the whole text.
- User will be allowed to use hotkeys instead of clicking the buttons, for instance: f4 to submit, f2 to delete the product. Again, this will increase the speed of the selection of an object, providing performance benefits to the user by reducing his task execution time.

3.2 Security Requirements

Security enhances can be made within the application through these techniques:

- Blacklisting of characters will be done so that the user cannot accidentally type in unknown keywords for example: digits in name field, alphabets in mobile field etc. This will provide user the security that his wrong entry cannot cause problem within the system.
- Asterisks will be used in password field to hide the password while the user is typing.
- Dialog Boxes will be prompted if user commits an action that can seriously affect the system. This way user will be provided an alert before he/she commits a wrong action.
- Disabling the options that are not needed in a specific scenario will also secure application from being damaged. For example: if the user submits the order, the submit button will be disabled. Once the user clicks the clear whole screen button, the submit button will be enabled again. Since when the user has added the order into the database, he/she can accidentally press the submit button twice (if it remains enabled) making two orders that are same which can cause an issue in sale details. This disabling of those buttons that are not needed in a scenario will prevent user from making mistakes.
- Undo and Redo buttons will be provided so that user can resort to a state where there was no mistake being made. User will feel secure if he has the undo option; he will be aware that his mistake is not permanent and can be reverted. For example: if the user deletes an important product from the list, he will have the undo button available to move the system back to its previous state.

4. Other Requirements

POSS requires the following specification:

4.1 Software Requirements

- Windows 7,8 or 10
- Microsoft Visual Studio 2010 or onwards
- Microsoft SQL Server 2008 or onwards
- Microsoft SQL Management Studio R2 2008 or onwards
- Microsoft Report Viewer 2012

4.2 Hardware Requirements

- RAM 2GB or more
- CPU 3rd Generation or more
- Processor Dual Core 1.9 GHz or more

Appendix A: Glossary

POSS: *POINT OF SALE SYSTEM*

Blacklisting: *Not allow certain characters to be pressed via keyboard.*

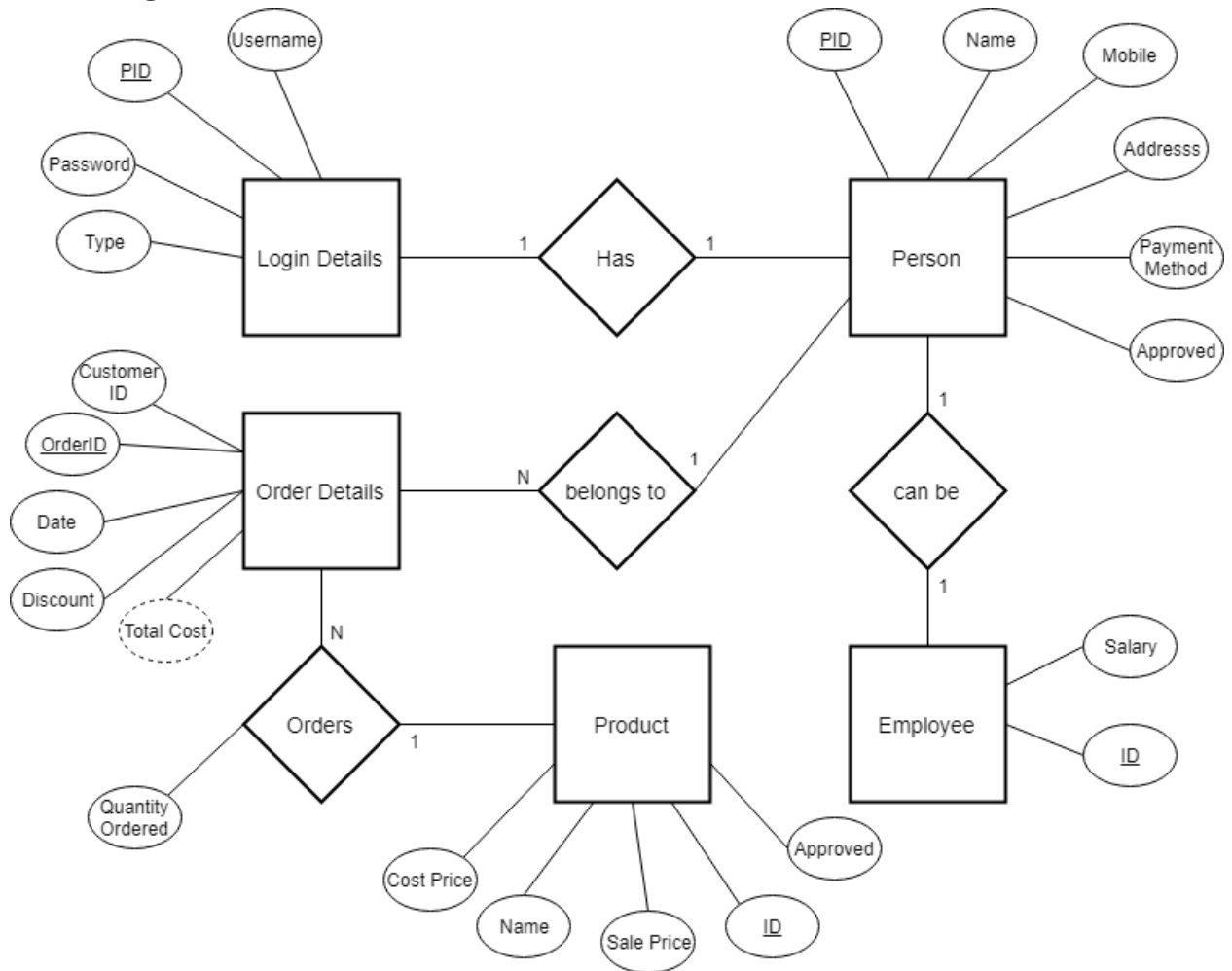
Execution Time: *Time taken for the user to perform a certain task.*

Hotkeys: *Special shortcut keys that are used to perform specific actions.*

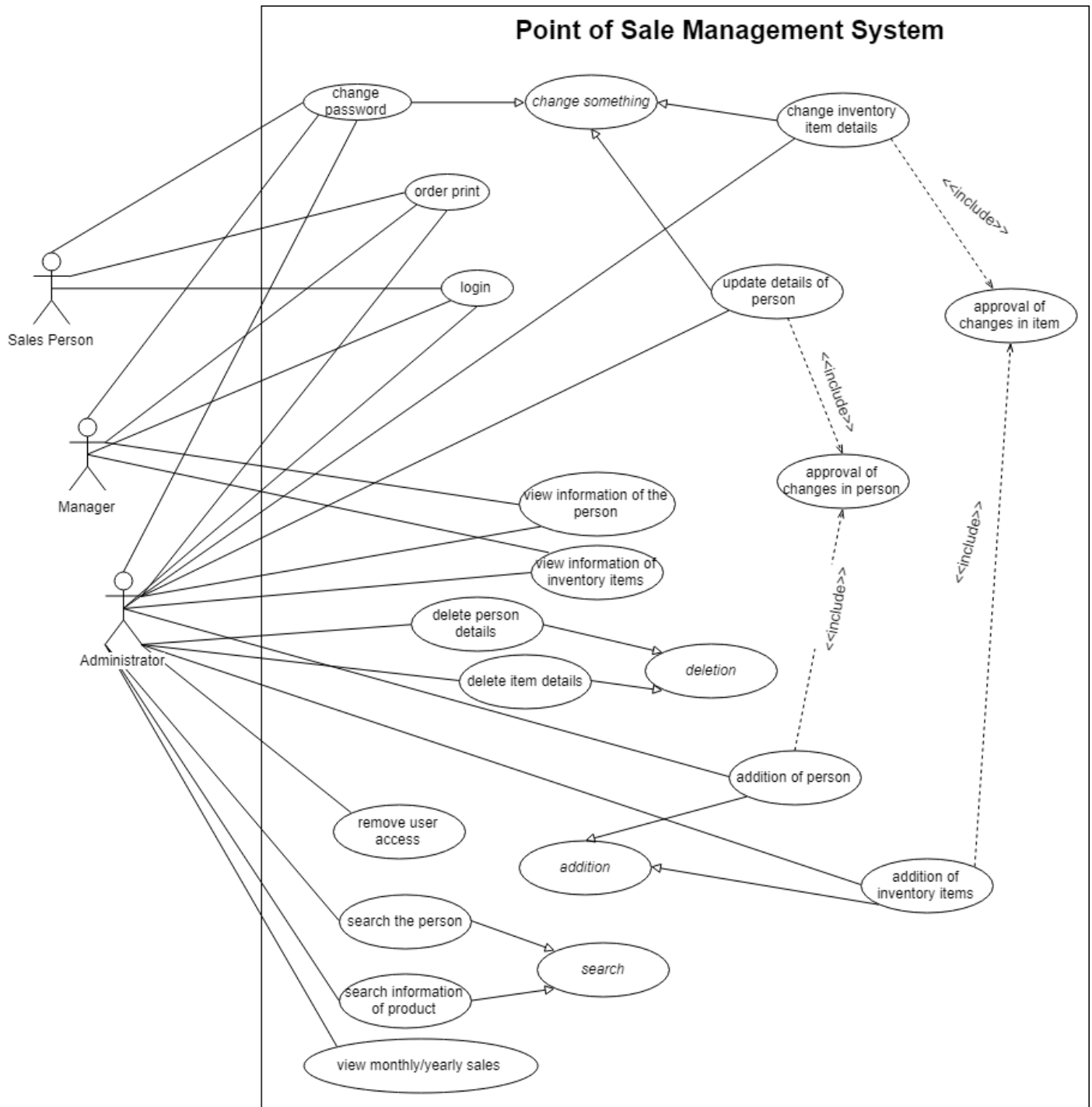
Turnover: *Rate at which the employees leave or join any company or organization.*

Appendix B: Analysis Models

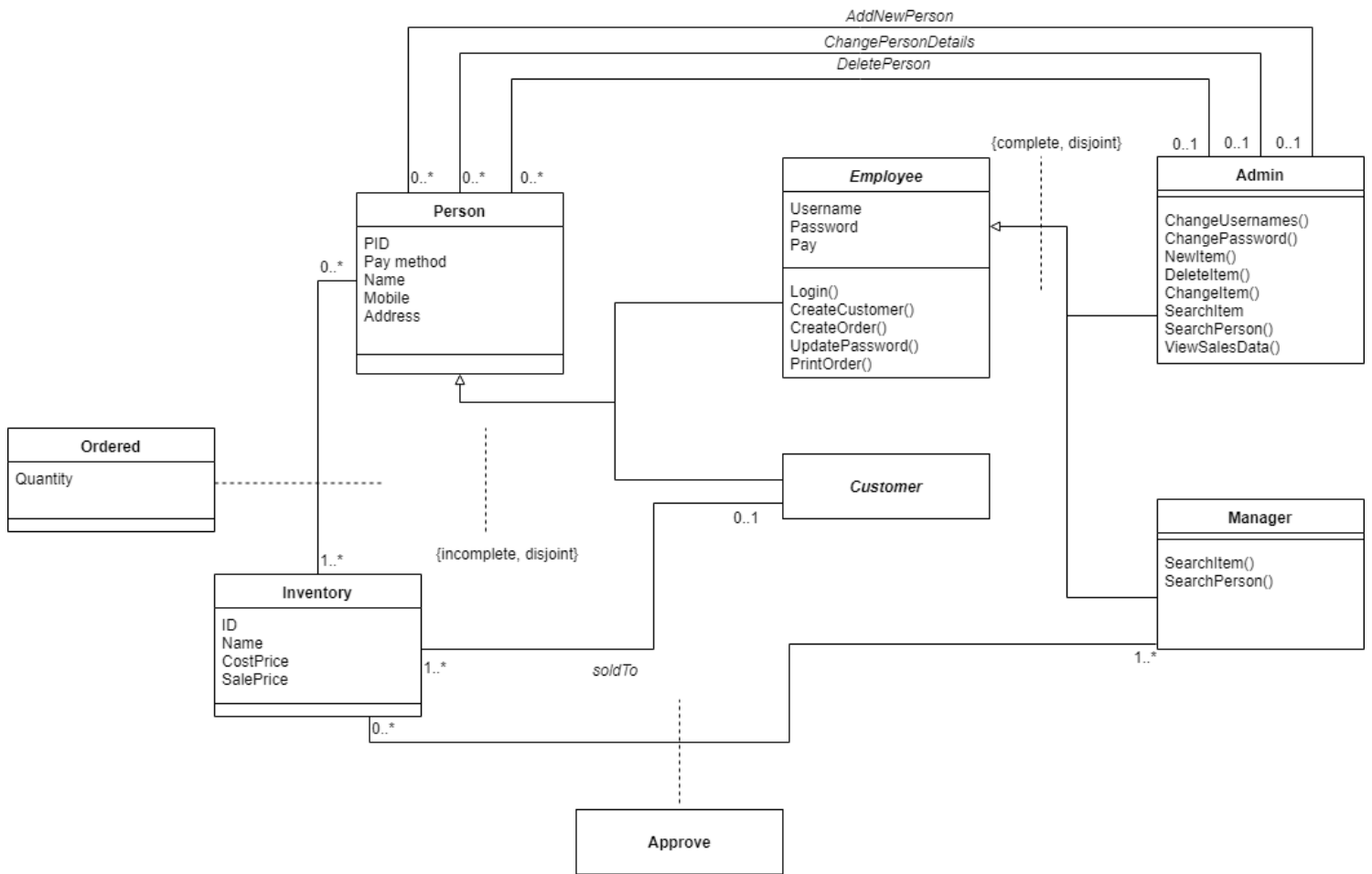
1. ER DIAGRAM



2. USE-CASE DIAGRAM

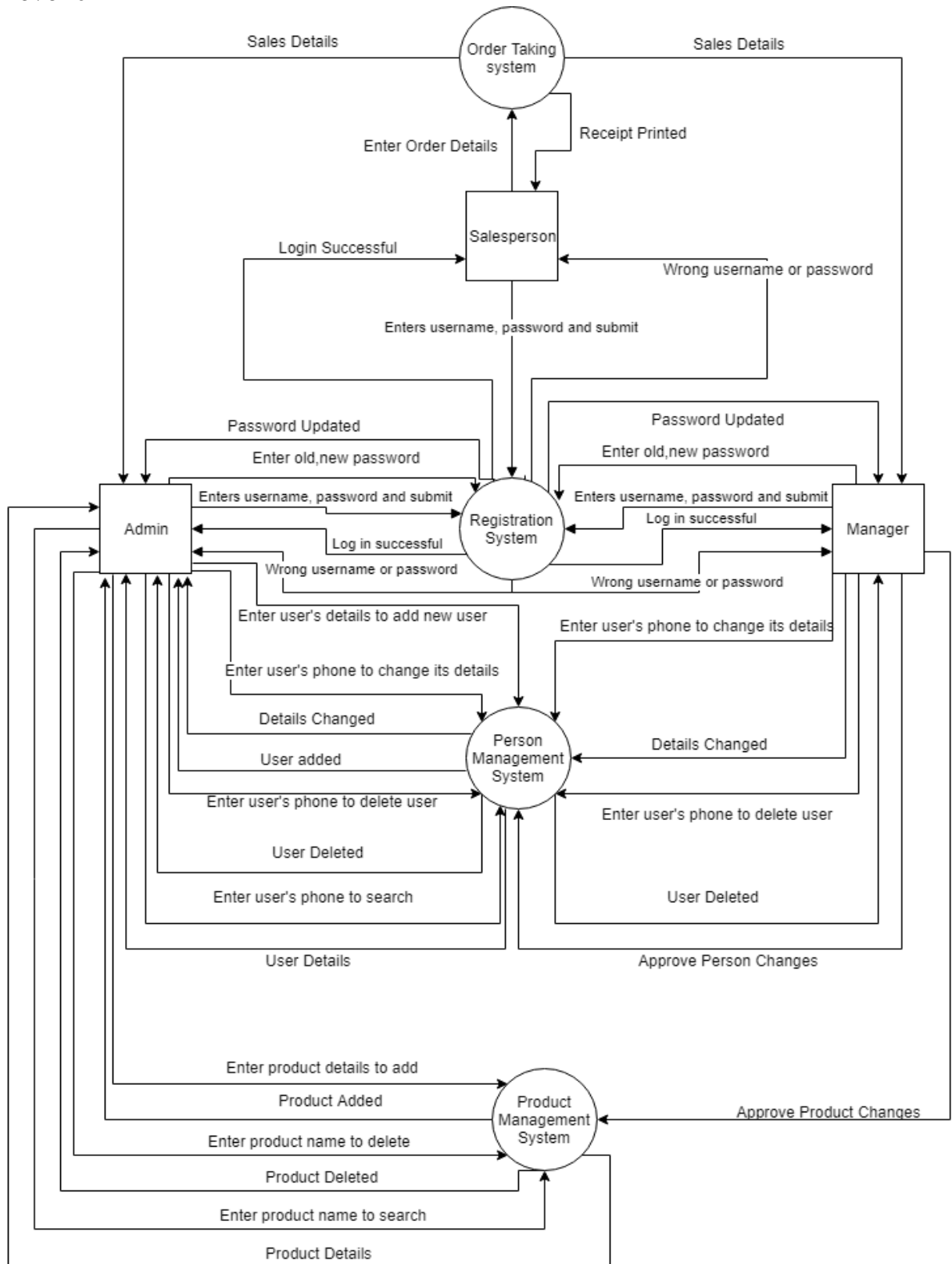


3. CLASS DIAGRAM



4. DFD DIAGRAM

- Level 0



- Level 1

