Software Requirements Specification

for

POS with IOT.

Version 0.0

Prepared by Kshitij

AI Kairos

5th September 2020

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

3. External Interface Requirements 3

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

4. Features and working 4

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

5. Other Nonfunctional Requirements 4

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

6. Other Requirements 5

Appendix A: Glossary 5

Appendix B: Analysis Models 5

Appendix C: To Be Determined List 6

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

*A better way to record sales activity and more easily track and measure the business operations.*

It is an addition to the existing Point of Sales (POS) system to create, update, and maintain the inventory to meet customers’ requirements accordingly and reduce the sellers overhead to do the needful without the need of much labor.

<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.>

## Document Conventions

“ \*\* ” before a sentence denotes it is not confirmed or incomplete.

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>

## Intended Audience and Reading Suggestions

*This document is intended for use by current and future development teams and outside project advisers and stakeholders who are associated with the project.*

*This document contains technical information regarding the performance, specifications, functional requirements and also includes the User Interfaces.*

\*\*Sections that are most pertinent to each reader type.

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

## Product Scope

This system will be aiming at monitoring the sales with POS technology and use AI techniques which will help salesmen to map the needs of customers during different events and provide statistics & recommendations to the seller and customer to eventually boost sales and further improve inventory management.

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>

## References

\*\* currently None

# Overall Description

## Product Perspective

*It is a self-contained, standalone product & is mainly targeted to anyone using POS as the purpose is to integrate and improve upon already existing local POS systems*

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

## Product Functions

The project provides clients with:

* A user account, including a username, a password, and a list of owned stores
* A service that facilitates communication between the POS system and the servers[optional].
* A system which notifies the client of online transactions carried out by the customers.

Clients:

* Create and own multiple stores to display and sell products
* Create, read, update, and delete both local and online inventories
* Monitor Sales online through the web application

Customers:

* Create, read, update, and delete a personal, shopping cart
* View, purchase, and pay for items electronically through various mediums

Windows Syncing Service

* Resolves conflicts between online and offline inventories
* Provides transaction data to the client
* Provides real time statistics of sold products(sales) to the client
* Receives and transmits inventory data to and from each local client and the server

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high-level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top-level data flow diagram or object class diagram, is often effective.>

## User Classes and Characteristics

* *Client class:*

*Clients compose the primary user class who are going to be integrate the project into their stores.*

*Clients have personal accounts that keep records of all pertinent information.*

*Clients are provided with the Windows service that syncs their personal POS to their store.*

*Clients can create, read, update, and delete online and local inventories, sell items etc.*

* *Customer class:*

*Customers compose the secondary user class which primarily interacts with the clients.*

*Customers can create, read, update, and delete personal shopping carts.*

*Customers also can view and pay for online items through a client’s personal web store.*

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>

## Operating Environment

\*\*yet to be decided

The machine can use any compatible operating system like Linux/Windows which allows sync functionality to run smoothly without much requirements and problems.

And t*he syncing web service requires a web server like Apache etc. to run PHP.*

\*\*not yet decided to use whether Raspbian/Win10 IOT core if using Raspberry pi, else will be going with Win 10 if a PC will be used[for the machine carrying out the computation & syncing process].

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

## Design and Implementation Constraints

The main constraint is the computational resource as a single Raspberrypi won’t be able to handle such intensive jobs and implementing multiple of them will substantially increase the cost of the project.

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).>

## User Documentation

\*\*maybe a tutorial to setup the project

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

## Assumptions and Dependencies

*Assumptions:*

* *Clients are expected to be store owners themselves.*
* *Clients are expected to already have a POS set up.[mainly for the transaction part]*
* The sensors will be connected via a wired connection as of now.[wireless is future scope]

Ideal working assumptions:

* The project establishes a connection between a local POS and a remote web service.
* The remote web service in turn will establish a session for the client.
* The project will focus on maintaining and updating both local POS and online databases, while also facilitating online transactions for a given client.

\*\*More will be added on more clear vision of expected results…

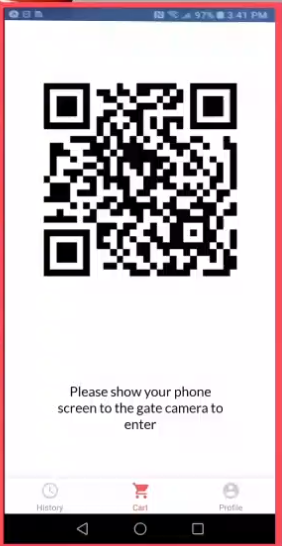
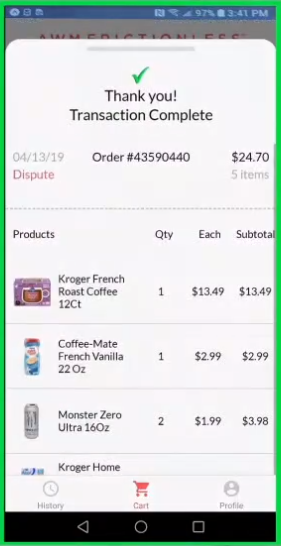
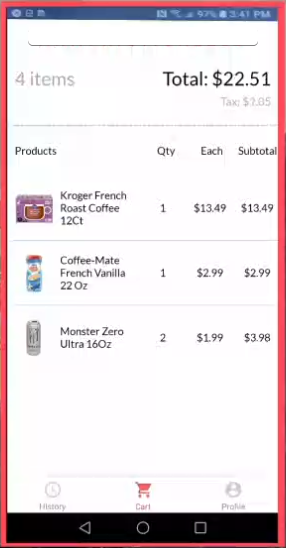
At current state there are many assumptions rather than the basic ones mentioned here

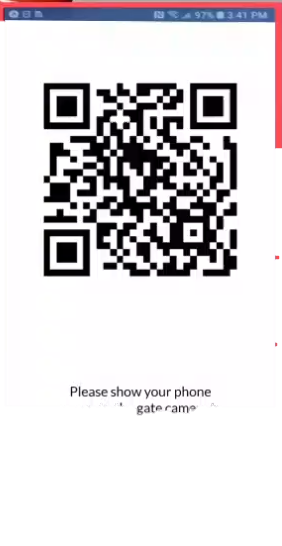
<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

# External Interface Requirements

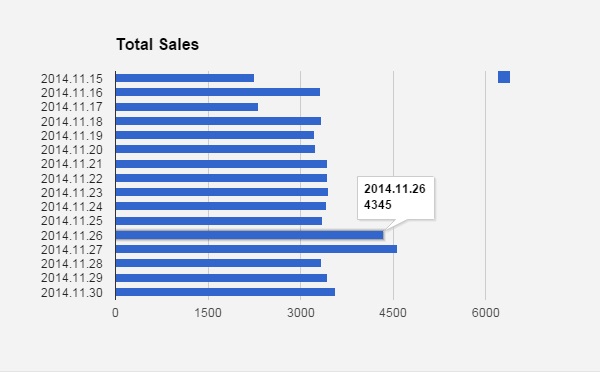
## User Interfaces

Customer oriented UI





Clients interface

Must be able to graphically display daily sales graph etc on timely basis.

And with AI also plot the prediction graph for planning the inventory ahead.

As for the processing/sync machine will be able to display the transaction databases and also the graphs available to the client.

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

## Hardware Interfaces

For the clients the minimal requirement for anyone wanting to use the service is any device capable of browsing the web and available internet connectivity.

As for the project the requirements are:

* A sensor for recording customers activities [a camera or IR sensor not decided yet]

like adding a product to cart, removing etc.

* A camera for reading QR codes for initiating the customers buying session.
* *Server-side machine service requires the ability to grow dynamically based upon number of clients and frequency of communications but does not initially require a large amount of space or resources to run.*
* As it is a web app, platform should not be an issue for customer, Input devices like a functional mouse and keyboard / a screen touch device are required for interacting with both the client and server-side devices.

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

## Software Interfaces

The only requirement will be an OS with python installation on syncing machine.

And for customer side a handheld device with web surfing capabilities.[preferably android]

\*\*Mainly will try to keep it to only python, but if need arises will integrate other languages.

[as python is capable of almost everything even the interface with help of pyqt5 for windows and

Web development languages like HTML,CSS,JS,PHP etc for maintaining web application]

As for recording transactions MySQL or NoSQL will be used depending on data structure.

\*\*Currently not completely decided as it would depend on the type of data captured

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

## Communications Interfaces

* *All communications and syncing between the client and server will be through XML formatted messages preferably encrypted in HTTP protocol.(might be different initially)*
* *Exchange of any secure/confidential data will not take place in this system*.

[As for transactions an independent system will be governed]

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

# Features & working(described through UML)

\*\*will be added in iteration v1 when at least the basic prototype of project is made

Currently we can get the basic idea from uml diagrams.

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

## Feature 1

<Don’t really say “System Feature 1.” State the feature name in just a few words.>

4.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

4.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:

REQ-2:

## System Feature 2 (and so on)

# Other Nonfunctional Requirements

## Performance Requirements

* *Client-side service will require a machine capable to run the required scripts with Min. 2GB of free space further depending on size of the dataset size.*
* The platform will most probably will be a Raspberrypi or a more powerful machine. For E.g. Intel i5(min 4th gen) or equivalent because for the AI part a strong processor is required.
* *All transactions, whether online or offline, trigger a sync process. During this process, there is a chance of displaying conflicting information regarding the stock of an online store. A sync process should take a minute or less to complete in order to minimize the frequency these types of conflicts from occurring.*
* *All syncing communications timeout after one minute of no response from the either client or server. After this time, the non­responsive component is assumed to be unreachable, and any uncompleted actions are set aside for the next sync unless communication from the unreachable component is reestablished.*

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

## Safety Requirements

* The project might not work as intended at least at initial stages and might need a person to go thought the orders and results.
* Possibility of fraud on use of unconditional technology like IR sensors(if used).

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>

## Security Requirements

* *In regard to authorization, payment, and bank account information, all sensitive information is sent over encrypted channels and databases that are managed by other established services.*
* *The project does not store or directly manipulate any sensitive banking information. Rather, it only asks for and archives usernames, passwords, and information directly related to the session’s web service.*

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

## Software Quality Attributes

*The service must be tailored to the needs of those with little to no technical background.*  
*All interfaces and communications with clients are geared toward simplicity and ease of access.*

*\*\*If all goes well an API will be tailored for more extensive & flexible development.(integration)*

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

## Business Rules

* *Clients have full control over their personal store and local Window’s Service at all times.*
* *This includes, but is not limited to, editing the inventory, themes, and product information at any time.*
* *Client’s do not have access to any other client’s personal or web store information.*
* *Customers only have access to information displayed on a client’s physical store.*
* *Customers have access to web carts and the ability to make transactions on in stock products only when inside the store.*
* *Administrators have access to usernames, web store information, and all other*

*non-­ -secure/encrypted information, such as client passwords.*

* *Administrators, only with the permission and observation of respective client, have access to all features of a client’s web store.*
* *This is to help reduce client expertise and interaction with the system if the client does not*  
  *understand or wish to edit the store themselves*

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

# Other Requirements

\*\*currently none

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

**POS :** An acronym for Point of Sales, which is the place where a transaction occurs in exchange  
for goods or services

**PHP :** An open source server­-side scripting language designed for Web development to produce  
dynamic Web pages

**Client :** An owner and operator of a single instance of the installation

**Customer :** One of the users of a client’s service & buyer of the products.

**Administrator :** A person who manages the servers and products

.  
**GUI :** An acronym for graphical user interface, it is a type of user interface that allows users to  
interact with electronic devices using images rather than text commands

**UI :** An acronym for user interface. A UI is the space designed for interaction between humans and  
machines.

**MySQL :** The world's most used open source relational database management system

**Intuit :** An American software company that develops financial and tax preparation software and  
related services for small businesses, accountants and individuals

**Sync :** A term for the process for retrieving data, resolving conflict in the data, and sending the data back

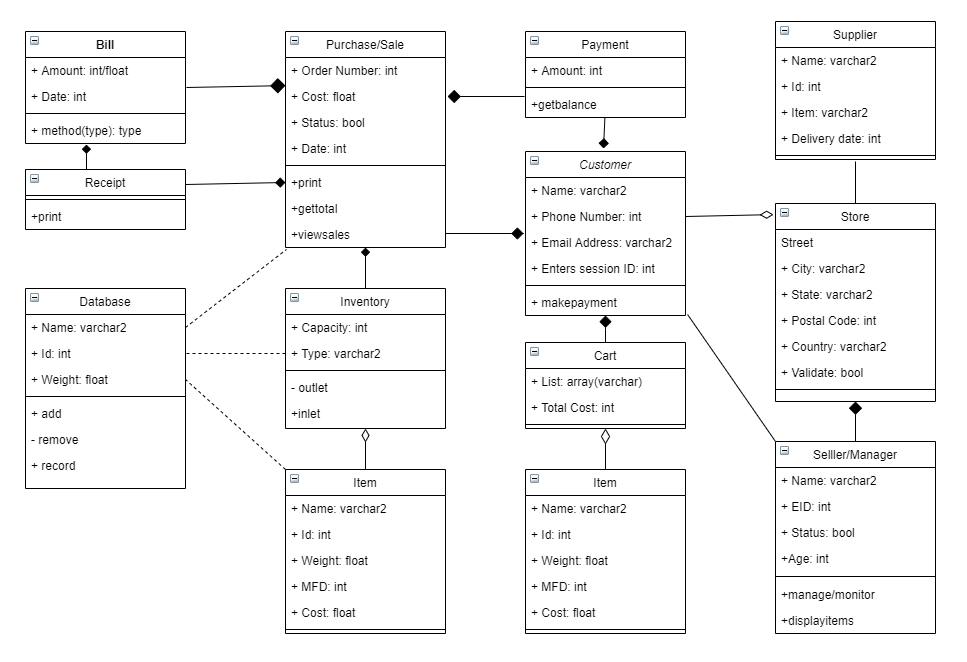
**OS :** Operating System. The main software that controls interactions from the hardware of the  
computer to the software that runs on the computer

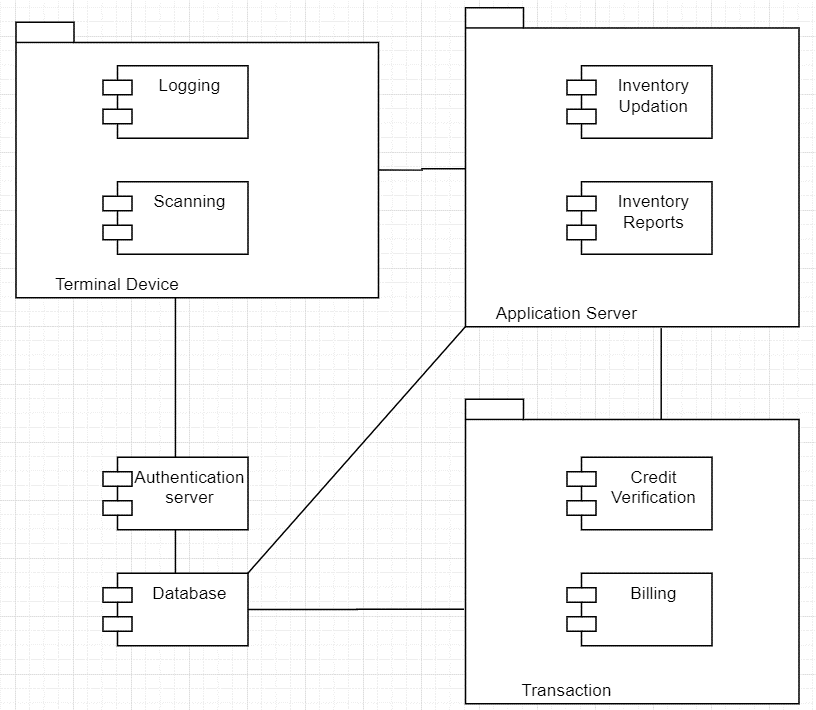
**API :** An acronym for application programming interface. An API is a protocol intended to be used  
as an interface by software components to communicate with each other

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

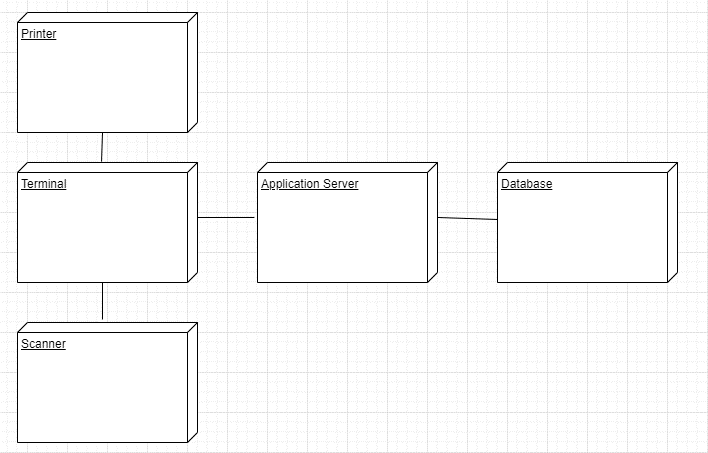
Appendix B: Analysis Models

UML Diagrams

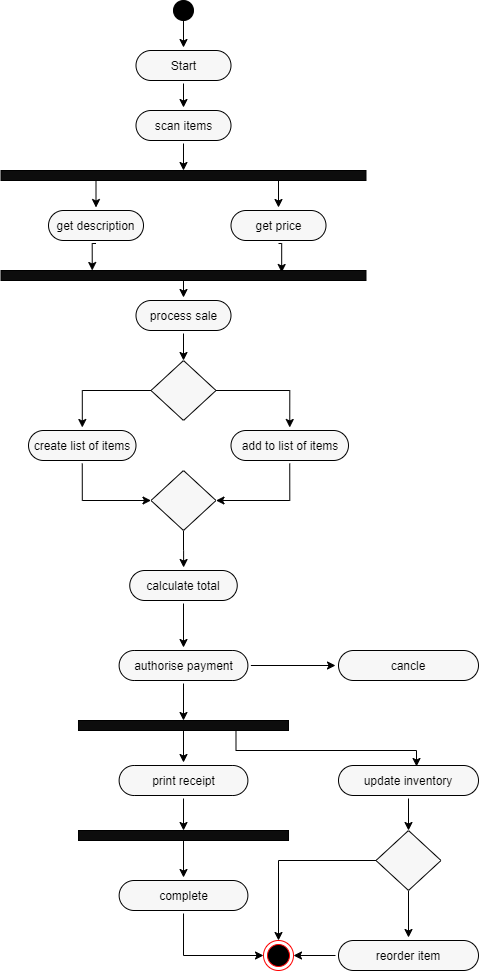
* Structural Diagrams
  + Class Diagram
  + Component Diagram



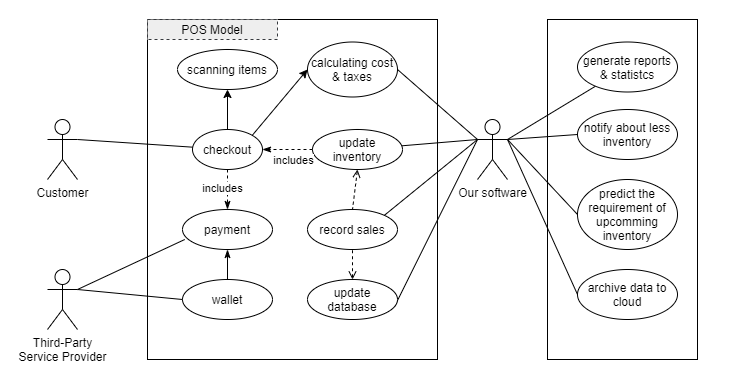
* + Deployment Diagram



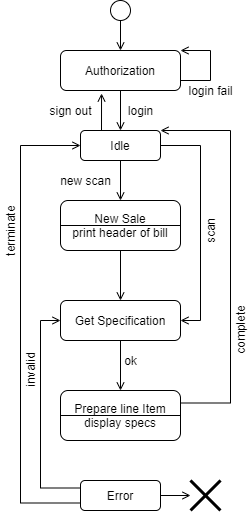
* + Package Diagram
  + Composite Structure Diagram
  + Object Diagram
  + Profile Diagram
* Behavioral Diagram
  + Activity Diagram



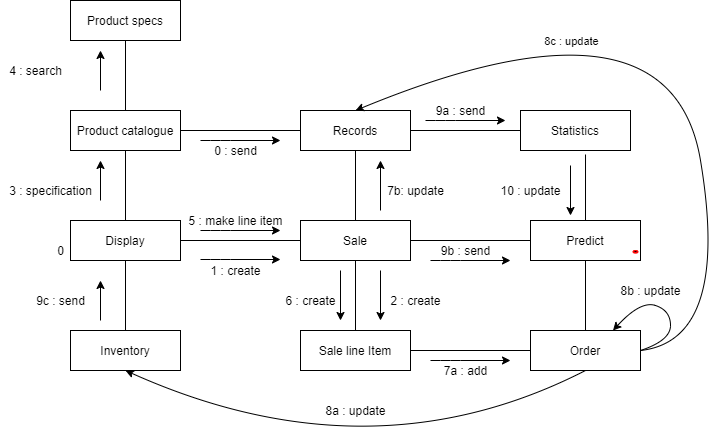
* + Use case Diagram



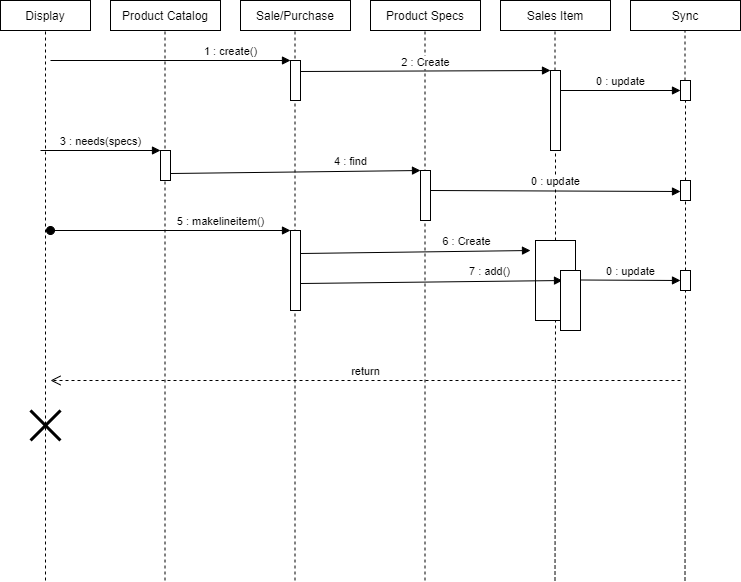
* + State Machine Diagram

**

* + Interaction/Communication Diagram



* + Sequence Diagram



* + Interaction Overview Diagram
  + Timing Diagram

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>