
COMP 3059 – Capstone Project I**Software Requirements Analysis and Design Assignment**

This assignment is an overview to gather the software needs with requirements analysis and help to proceed with the design.

The requirements analysis helps to break down functional and nonfunctional requirements to a basic design view to provide a clear system development process framework. It involves various entities, including business, stakeholders and technology requirements.

The design is the activity following requirements specification and before programming. Software design usually involves problem solving and planning a software solution.

To work on this assignment you could use the references and a sample template given below. The sample template can be customised to suit the nature of your project.

Reference Readings/Example:

http://www.uacg.bg/filebank/acadstaff/userfiles/publ_bg_397_SDP_activities_and_steps.pdf

www.cse.msu.edu/~chengb/RE-491/Papers/SRSEExample-webapp.doc

Reference template:

www.tricity.wsu.edu/~mckinnon/cpts322/cpts322-srs-v1.doc

1.0 Introduction	3
1.1 Purpose	3
1.2 Scope	3
2.0 System Overview	3
2.1 Project Perspective	3
2.2 System Context	4
2.3 General Constraints	4
2.4 Assumptions and Dependencies	4
3.0 Functional Requirements	4
3.1 Functional Requirements	5
3.2 Use Cases	6
3.2.1 Use Case	6
3.3 Data Modelling and Analysis	6
3.4 Process Modelling	6
4.0 Non-Functional Requirements	6
4.1 Performance	6
4.2 Scalability	6
4.3 Portability	6
4.4 Reliability	7
4.5 Security	7
4.6 Usability	7
5.0 Logical Database Requirements	7
7.0 Approval	8

- 1.0 Introduction

The Introduction section provides an overview of the system using software requirements analysis

1.1 Purpose

The purpose of this document is to collect, analyze and provide a clear definition to Jarvis Inventory System's high-level scope, the business opportunity of this project and the intended user. Its main focus is to indicate what this project intends to solve. The Inventory System created in this document is to emphasize the importance of this project in improving the inventory system of grocery stores. In it, the management must be able to add, edit, or remove information about a product order, and its supplier must be able to view orders. It also allows users to process orders automatically and also would notify them when the product is low on stock, and would also give daily reviews if the product is good to use or not. Managers can also order shipments for products through the system.

1.2 Scope

The scope of this project is to provide an automatic, efficient and visual inventory system for retail companies such as grocery stores. The system would help employees keep the inventory count more accurate by having it automated, allowing users to scan in products to add to the database and scan out products to remove anything from the inventory count. The app will allow users to check product information such as aisle number, stock count, vendor information, and processed orders. It will also allow employees to find a product by its name from product availability and alert the manager when a product is low on stock.

2.0 System Overview

The System Overview section introduces the system context and design.

2.1 Project Perspective

The Project Perspective describes the context and origin of the system by defining whether the system is:

An inventory management system that stores the following information:

- **Employee details:**

- It includes the login details of employees such as name, id, phone number, address and salary.

- **Inventory description:**

- It includes the products – barcode, name, aisle number, expiration date, quantity/amount

- **Vendor description:**

- It includes the vendor's details such as name, company, date of delivery, date of shipment, email, and phone number

- **Orders:**

- It includes the plu/barcode, product name, product quantity, and cost

2.2 System Context

Grocery stores face the inconvenience of having to keep track of their inventory manually. There's no reliable software system that allows them to help organize and keep all the information in one database. The Jarvis Inventory System, unlike other inventory management systems, will allow stores to manage their inventory and provide real-time tracking, store information about the product and access that information to help customers. This system will also allow users to analyze which products sell and which are low in stock and process orders from the vendor, improving the business flow.

2.3 General Constraints

- There are some tasks that can be assigned to certain employees as per their qualifications.
- Employees learning how to operate the system can be lengthy, cumbersome, and complex.
- Although the system provides amazing features to make the entire business much more efficient, it comes with a cost. For example, big-time businesses can cover the cost, but in small or medium-sized businesses, it is not feasible to maintain the software at times.
- The system helps control many risks, but still, there's a risk of encountering many others. Hence, many risks are restricted, but the entire process can't be risk-proof.

2.4 Assumptions and Dependencies

Let us assume that the system is used in the following:

- an internet connection and a web browser to use the system.
- the user knows using a computer and simple software or application
- once a customer buys the item, it will automatically deduct from the database.
- when there is a discrepancy in the inventory count that it is damaged, recalled or shrank products.
- the vendor would always approve the orders.

- 3.0 Functional Requirements

This section describes specific features of the software project. If desired, some requirements may be specified in the use-case format and listed in the Use Cases Section.

- **3.1 Functional Requirements**

1. The system should provide an efficient interface for managing the inventory.
2. The inventory count is continuously monitored by the system by updating the inventory every time a product expires, is purchased, or is damaged.
3. The system should notify managers when a product is low in stock or expiring.
4. The system will hold all the details of the product that are in stock in the store.
5. The system will hold all the details of the employees who work in the organization.
6. The managers should be the only ones with access to the vendor and employee lists.
7. It allows managers to manage different roles for employees and create login details based on their roles.
8. The system should allow managers to create, update and cancel a purchase order.
9. The system should allow clerks to create a purchase order.
10. The system should allow clerks or managers to input the quantity from the invoice of incoming shipments and update the count accordingly.
11. The system will create a purchase order when a product is low in stock.
12. When a refund is done by a cashier, the stock number will be automatically updated, but the product will be considered as damage.
13. The system will allow users to reset their password.
14. The system will allow users to change their password.
15. The system will allow managers to print a product sales report.
16. The system will allow employees to create a new product.
17. The system will allow employees to update information about a product.
18. The system will allow managers to add a new employee.
19. The system will allow managers to update information about an employee.
20. The system will allow managers to delete an employee from the system.
21. The system will allow the users to log out of the system.

- 3.2 Use Cases

- 3.2.1 Use Case

<https://drive.google.com/file/d/12fSDuCuxcjDx1DWfCGMXo-9RM90xQEWv/view?usp=sharing>

- 3.3 Data Modelling and Analysis

- Normalized Data Model Diagram
 - https://drive.google.com/file/d/1ocZMV2LfmZKjt-sbL9Oy21EH_JHQrEMH/view?usp=sharing
- Activity Diagrams(Open with diagrams.net to see multiple diagrams)
 - <https://drive.google.com/file/d/1-xLHrETUk5Ihyo1vAdCpuN9yA1mVM-8h/view?usp=sharing>
- Sequence Diagrams(Open with diagrams.net to see multiple diagrams)
 - https://drive.google.com/file/d/1T2HqP33ltiY0maggH_QlHXJWZ1IPfrAF/view?usp=sharing
- UML Class Diagram
 - <https://drive.google.com/file/d/1Dk1Ql2uQWzkcPoPMw00qztEZGcUlfUsP/view?usp=sharing>

- 3.4 Process Modelling

- Data Flow Diagram
 - https://drive.google.com/file/d/1GEPFFru71WBugwF0UAWGTdd_zzM5HBOc/view?usp=sharing

- 4.0 Non-Functional Requirements

- 4.1 Performance

- The website must work efficiently despite having multiple users using the system simultaneously.
- The website's load time should not take more than a few seconds.
- The system must not lag because the users don't have downtime to watch until it finishes a function.

- 4.2 Scalability

- The system must be scalable to handle any volume of simultaneous visits while maintaining peak performance.

- 4.3 Portability

- The website must adapt to the user's device, i.e. when they are accessing it on a mobile device.

- **4.4 Reliability**
 - The user must constantly receive accurate inventory status from the system. Any accuracy is prevented by routinely verifying real levels with the levels shown by the computer system.
 - If any operations fail, the system shouldn't modify the information in any databases.
 - To prevent any outside entity from altering the system's data, the system must offer the user a password-protected login.
- **4.5 Security**
 - The information about user accounts must be kept accurately by the software.
 - The system's behaviour must be accurate and predictable.
 - A user without authorization cannot be given access to a restricted web page by the server.
 - A browser that the server is unable to authenticate must not receive a restricted web page.
- **4.6 Usability**
 - All important data and relationships must be shown by the system in a straightforward and understandable manner.
 - The users (i.e. Manager) must be able to access the system promptly.
 - The users (i.e. Managers) must find the system simple to use, so they don't have to read a lot of instructions.
- **5.0 Logical Database Requirements**

The database we will use is H2. H2 is the most suitable because it can work with any suitable database in a smart way.

The main features of H2 are it is very fast, open source has JDBC API, it's a browser-based console application, and has a small footprint of around 2.5 MB jar file size.

Data retention:-

- Personal data wouldn't be kept longer than it is necessary for the purpose for which the personal data are processed.
- Organizations will cut down media when it is no longer needed for business or legal reasons.
- It would have automated inventory rebalancing to simplify complex inventory optimization processes.
- Set a practice of keeping records for a certain period of time to comply with business needs, industry guidelines and regulations.

Data integrity:-

Data integrity is the difference between being able to keep promises to end customers or running around trying to find items you were sure you had in stock. It can also be defined as a concept and process ensuring an organization's data's accuracy, completeness, consistency, and validity.

Entity integrity-

There would be no duplicate records within the table in our database. In the primary key, the elements will be as much as needed for the data to be accurate, every element would be unique, and none of these elements would be null.

User-defines integrity-

The rules and constraints around data are created by users to align with their specific requirements, and it's used when other integrity processes will not provide an organization's data, and that would be classified as a "user-defined" safeguard.

- **7.0 Approval**

The signatures below indicate their approval of the contents of this document.

Project Role	Name	Signature	Date
Lead Developer	Marie Vianca Pagaduan	Marie	11-08-2022
Developer	Ellyn Francess Bibon	Ellyn	11-08-2022
Developer	Janine Mae Usana	Janine	11-08-2022