

## Experiment No 8

### Identifying System requirements for an Architecture for any specific domain.

43\_BECOMP-B\_Sagar Pachare

**Learning Objective:** Student should be able to understand System requirements for an Architecture for any specific domain.

#### **Theory:**

The purpose of a requirements architecture is to structure and organize requirements in such a way that the requirements are stable, usable, adapt to changes, and are elegant. When a requirements architecture is sound, it helps facilitate better design of the system it attempts to describe. When a requirements architecture is faulty, it can cause problems. When the requirements architecture is poor, the following problems result:

- i. No one knows why a requirement was changed
- ii. Requirements cannot be reused
- iii. Traceability is superficial or unused by other teams
- iv. Requirements reviews involve irrelevant information
- v. Big picture of the system being built and reasons for building it are not well-understood
- vi. It is important to keep in mind that the purpose of a good requirements architecture is to build working software that meets business objectives.

Software requirements must be testable, unambiguous, and concise, a requirements architecture must also possess certain attributes. The above blueprint provides some general guidelines for how to structure requirements, but keeping in mind the following attributes:

- a. **Maintainable:** Whatever choices you make in organizing requirements, ensure that you create a structure that can adapt to changes in requirements.
- b. **Traceable:** Do you know which requirements any given process flow step is traced to?
- c. **Usable:** Consider the stakeholders in the org chart—are the requirements architected in such a way that you could either produce output for each of them or such that they could navigate to the requirements in the tool and find the requirements objects that are relevant to them? The hierarchies and traces you create should be consistent: Don't create one hierarchy where the FRs are children of the models and another hierarchy for the same project where FRs are not children of the models but are traced to

them. The absolute worst thing to do is to list all requirements objects in a flat list or to manage your requirements in word or excel.

- d. **Scalable:** Imagine your requirements architecture with 10 times the number of requirements it has. Now imagine it with 100 times the number of requirements. Architectures should be able to support the addition of new requirements with minimal overhead.
- e. **Elegant:** Are there just enough hierarchies to facilitate use? Are you repeating hierarchies just to make traceability easier? Does your architecture contain duplicate models or requirements?
- f. **Generalizable:** The architecture approach should be repeatable. You ought to be able to go into any project and no matter the domain uses the same approach to requirements architecture.

All architectures are tradeoffs – like in software architecture, you may need to sometimes sacrifice aesthetics for robust traceability or reuse. Or you may sacrifice usability for ease of exporting to external formats. Understand the tradeoffs you are making with your requirements architecture.

Domain-specific Software Development Various mechanisms of domain-specific software development are under investigation within the projects. Compositional mechanisms facilitate reuse of existing artifacts, including software. Generative mechanisms are used when needed components are not available.1 Constraint-based reasoning systems and module interconnection languages are critical underlying technologies for software composition. Prototyping technologies underlie generation. The TRW project serves as a technology conduit from the prototyping community into the DSSA program.

Following are the system requirement of Domain-specific Software Development:

- Performance
- Scalability
- Availability
- Reliability
- Security
- Maintainability
- Flexibility
- Configurability
- Personalizability
- Usability
- Portability
- Conformance to standard

# *Software Engineering Software Requirements Specification (SRS) Document*

*Sagar Pachare  
15/10/2022*



## Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Final Draft	Sagar Pachare	All sections being Filled	15/10/2022

## Review & Approval

### Requirements Document Approval History

Approving Party	Version Approved	Signature	Date
Sagar Pachare			15/10/22

### Requirements Document Review History

Reviewer	Version Reviewed	Signature	Date
Ms Drashti Shrimal			15/10/22



## Contents

1. Introduction	3
2. General Description	3
3. Functional Requirements	4
4. Interface Requirements	5
4.1 User Interfaces	5
4.2 Hardware Interfaces	5
4.3 Communications Interfaces	5
4.4 Software Interfaces	5
5. Performance Requirements	5
6. Other non-functional attributes	6
6.1 Usability	6
6.2 Performability	6
6.3 Reliability	6
6.4 Maintainability	6
6.5 Security	6
7. Operational Scenarios	7
8. Preliminary Data Flow Diagram	8
8.1 Data Flow Diagram	8
9. Updated Schedule	10
10. Updated Budget	10
11. Appendices	10
11.1 Definitions, Acronyms, Abbreviations	10

## 1. Introduction

### 1.1 Introduction

Inventory Control System has become an important part in today's world as managing different inventories is becoming quite tedious. This system will help the people to streamline the management tasks and provide real time data. Building this system as a web application will further help with ease of accessibility. The study findings enable the definition of the project problem statement, its objective, scope, and advantages of the Inventory Control System.

### 1.2 Scope of this Document

In this document, we are covering requirements for the Inventory Control System. This system will allow users to perform various operations like storing, updating, and retrieving Product information. This would be very helpful to other developers in selecting a design that will be able to accommodate the full-scale application. This system will capture information about customer's personal details, products, and their quantities. Storing, updating, and retrieving in a fast and accurate way.

### 1.3 Overview

An inventory control system tracks inventory availability at all points of your supply chain — from purchasing, to production, and finally sales to consumers. An inventory control system is important for companies with large amounts of inventory in order to keep it at optimal levels.

### 1.4 Business Context

This document is intended for both end users and the developers of the software.

## 2. General Description

### 2.1 Product Functions

Inventory Control System has some features, which will help a shop owner to manage their shops effectively. The features are,

- Web based interface
- User based Login - password based authentication for data protection.
- Dynamic Modular structure – Different users have access to different modules of the system.
- Details of buyer, supplier and vendor will be available at one place.
- Different Products tracking.
- Inventory stock tracking.
- Low stock notifier.
- Product category and product stock information with value.
- Reports include purchase, issues, stocks & categories.
- Notification System in case of any process delay.
- Feedback from customers.

## 2.2 Similar System Information

The product has been developed by Zoho Inventory.

## 2.3 User Characteristics

ICS has three types of accessing modes,

- 1) Administrator: Administrator has the highest level of privilege. They have access over all the system operations. Administrator has to update and monitor the registered product details, add a new product, provide product number for all products, assign each product quantity and GST etc., Administrator can update his profile, and also can give help to the customers.
- 2) Employee: Employees will be taking care of the billing section and will have access to some of the functionality of the system. They will be responsible for checking the stocks in the store and report the same to the admin. Employees will also be taking care of the billing section.
- 3) Customer: Customers are the end users. Customers can purchase the products and make payment accordingly. All his data will be saved into the database by the system to keep record of the sold products. They can also provide feedback

## 2.4 User Problem Statement

A grocery control system where users can order the items, get the receipt and make payment. The employees can update the stocks and generate reports.

### 3. Functional Requirements

- Functional Requirements for Administrator
  - 1) Login.
  - 2) Accessibility to the whole system.
  - 3) Monitoring the total system as well as the database.
  - 4) Giving the total goal of a deal as the input.
  - 5) Clarifying the resources.
  - 6) Distributing the job segments to the controllers.
  - 7) Getting updated information about the production.
  - 8) Getting update information about production delay.
  - 9) Getting update information about operation.
  - 10) Invoice information collection.
  - 11) Change settings
- Functional Requirements for Employee
  - 1) Login
  - 2) Bill Generation
  - 3) Accessibility of availability of stocks
  - 4) Updates about resources

### 4. Interface Requirements

#### 4.1 User Interfaces

User interface is one of the most important elements in any software. Most of the software is used by non- technical persons. So, they always seek a user-friendly environment in their system. And the user interface makes the system more familiar to its user.

#### 4.2 Hardware Interfaces

- 1GHz or High processor
- 512 MB RAM
- 500 MB Hard Disk



### 4.3 Software Interface

The Current version of this system will be built on the following software:

Server:

- Node JS
- Express JS

Client:

- HTML
- CSS
- JavaScript (synchronous/asynchronous) Enable Browser.

Database :

- MongoDB

### 4.4 Communication Interface

There are:

- Should run on 500GHz, 64MB Machine.
- Should have a proper internet connection.
- The response time for occurs a change will be more than 4 seconds

## 5. Performance Requirements

Our proposed inventory control system will have a huge amount of data and this data is very precious. So, keeping this in mind there are several performance factors we need to consider. There are a lot of internal and external operations which are interrelated with each other for fruit full production. So, the communication among each end system should be tightly scheduled and the notification should be sent in a timely manner.

## 6. Other non-functional attributes

- **Usability:** The product will be made as simple as possible with respect to usability and this will be achieved by making a lucid UI.
- **Performability:** Our proposed inventory control system will have a huge amount of data and this data is very precious. So, keeping this in mind there are several performance factors we need to consider. There are a lot of internal and external operations which are interrelated with each other for fruit full production. So, the communication among each end system should be tightly scheduled and the notification should be sent in a timely manner.
- **Reliability:** System we are building must be reliable as there are various operations, which generate a huge amount of data. So, we need to prevent this data from loss because these lost data can cause high damage to a specific module as well as to the total process. So, if the reliability is not confirmed, this deficiency will affect the production performance.
- **Maintainability:** In software engineering, software maintenance is the modification of a software product after delivery to correct faults, to improve performance or other attributes. Maintainability of software is categorized in four classes:
  - 1) Adaptive – dealing with changes and adapting in the software environment.
  - 2) Perfective – accommodating with new or changed user requirements, which concern functional enhancements to the software.
  - 3) Corrective – dealing with errors found and fixing it.
  - 4) Preventive – concerns activities aiming on increasing software maintainability and preventing problems in the future.

Therefore, to maintain our system we will try to concentrate on these four classes.

- **Security:** This system will deal with huge amounts of data. So, security is a prime issue of our system. So, the system should be secured from external interference by providing efficient security to the entire system.

## 7. Operational Scenarios

### Scenario A: Admin Login

The admin will login into the portal and perform operations related to customers and employees.

### Scenario B: Customer Login

The customer will login and make payment.

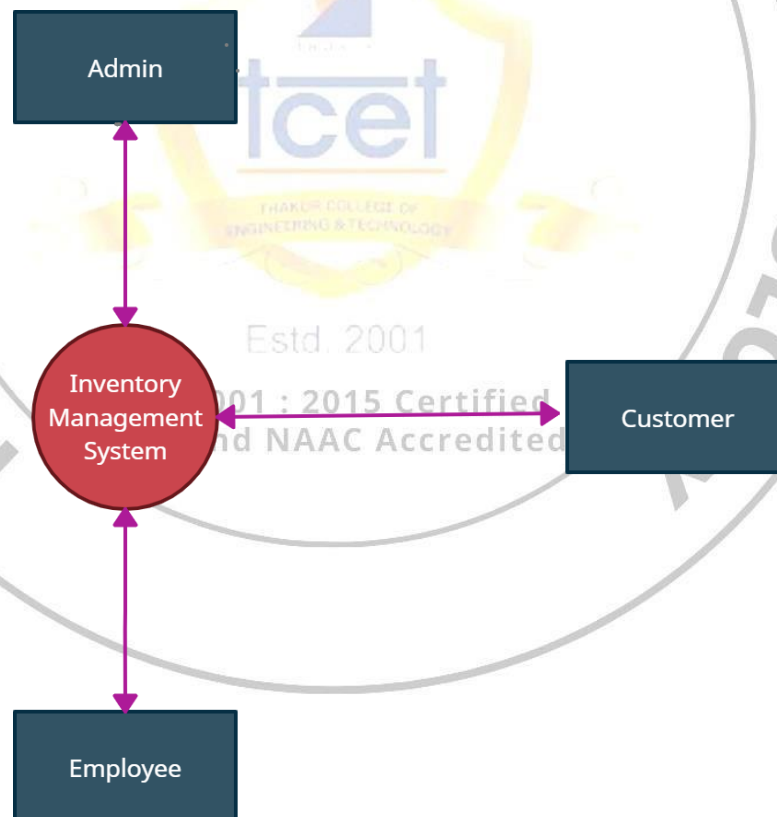
### Scenario C: Employee Login

The employee will login and make billing receipt and update stocks.

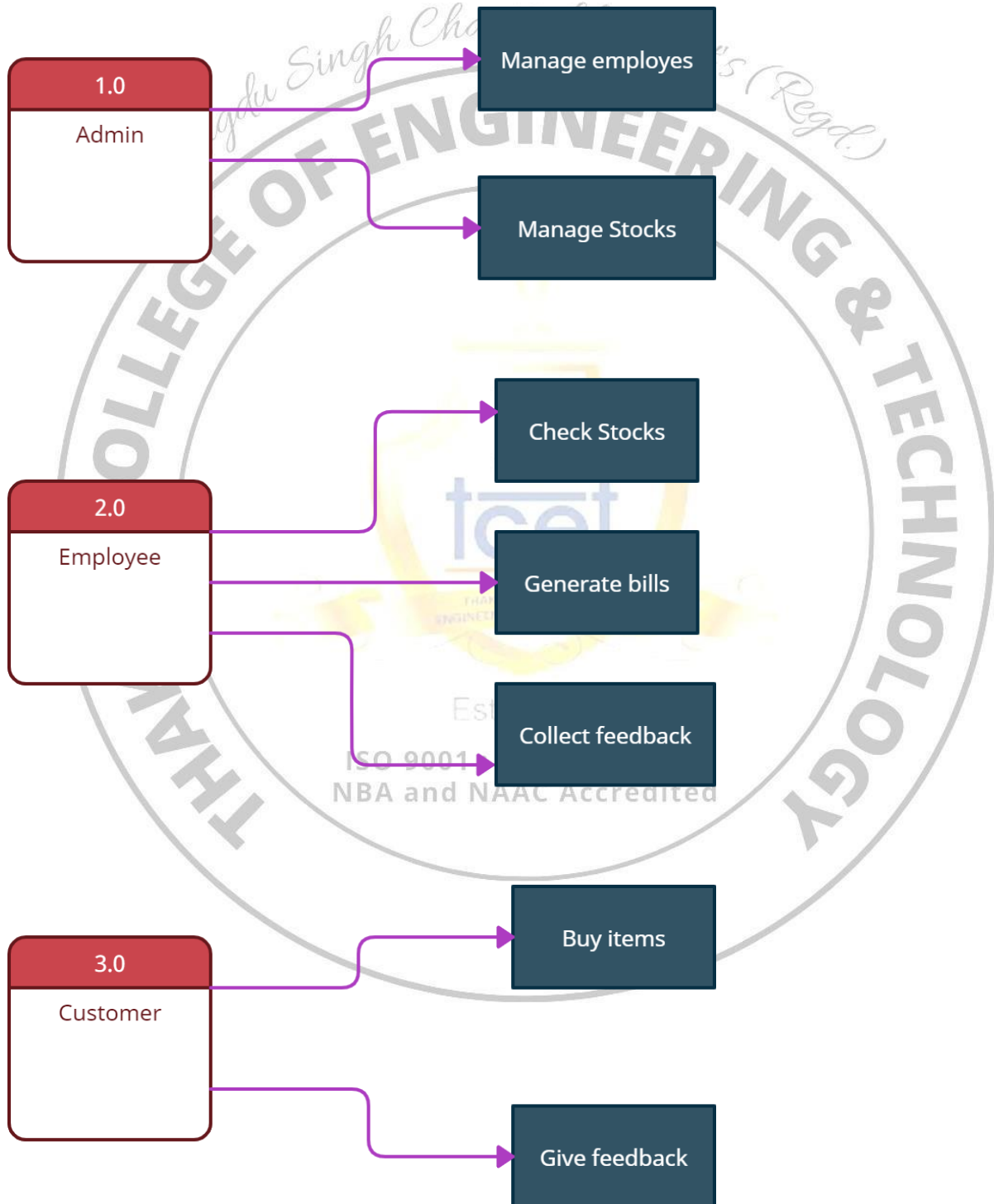
## 8. Preliminary Data Flow Diagram

This section presents a list of the fundamental Data Flow Diagram that satisfy the system's requirements. The purpose is to provide an alternative, "structural" view of the requirements stated above and how they might be satisfied in the system.

### 8.1 Data Flow Diagram

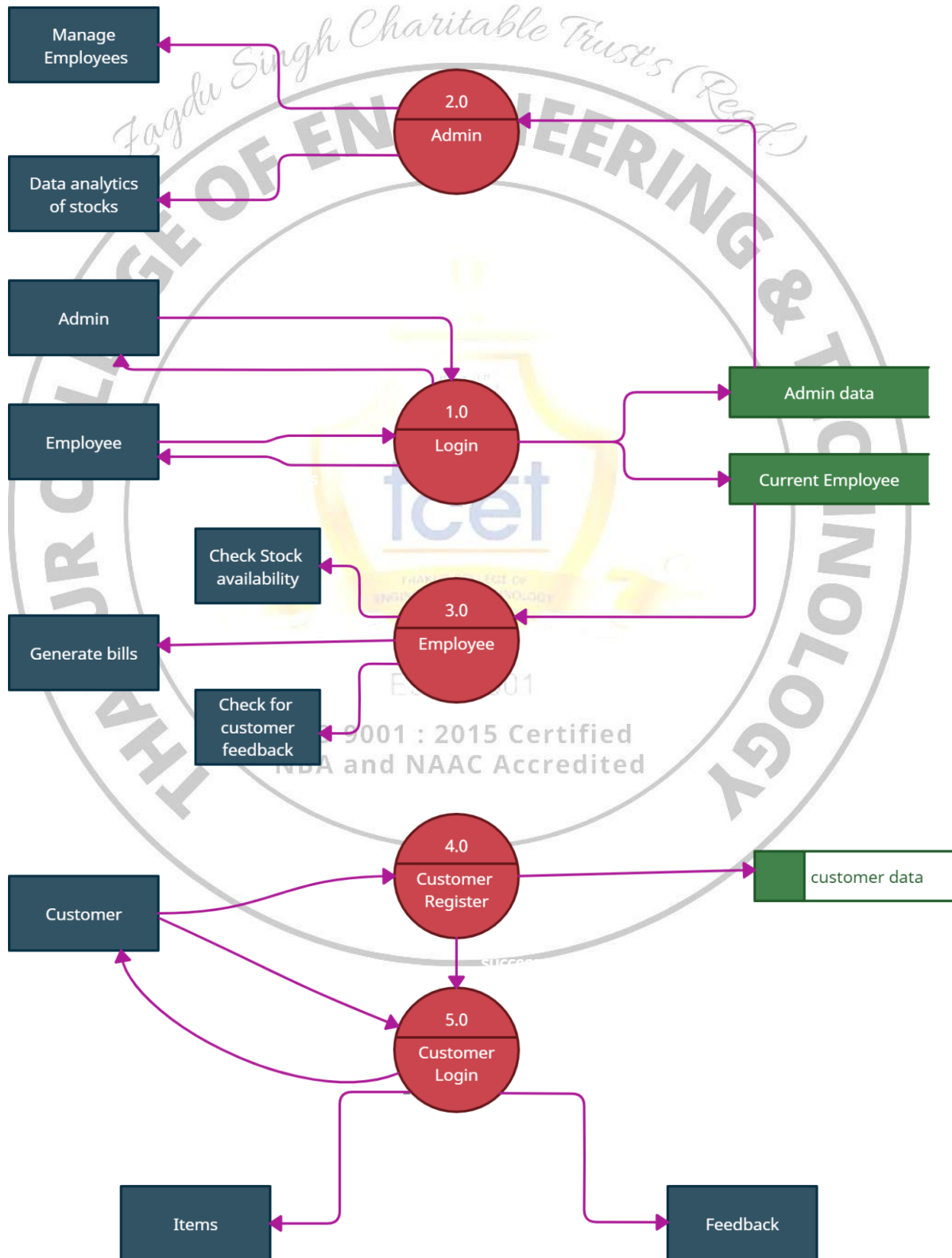


Level 1 DFD



Level 1 DFD





Level 3 DFD

## 9. Updated Schedule

The updated PERT/GANTT chart is attached at the end of the document

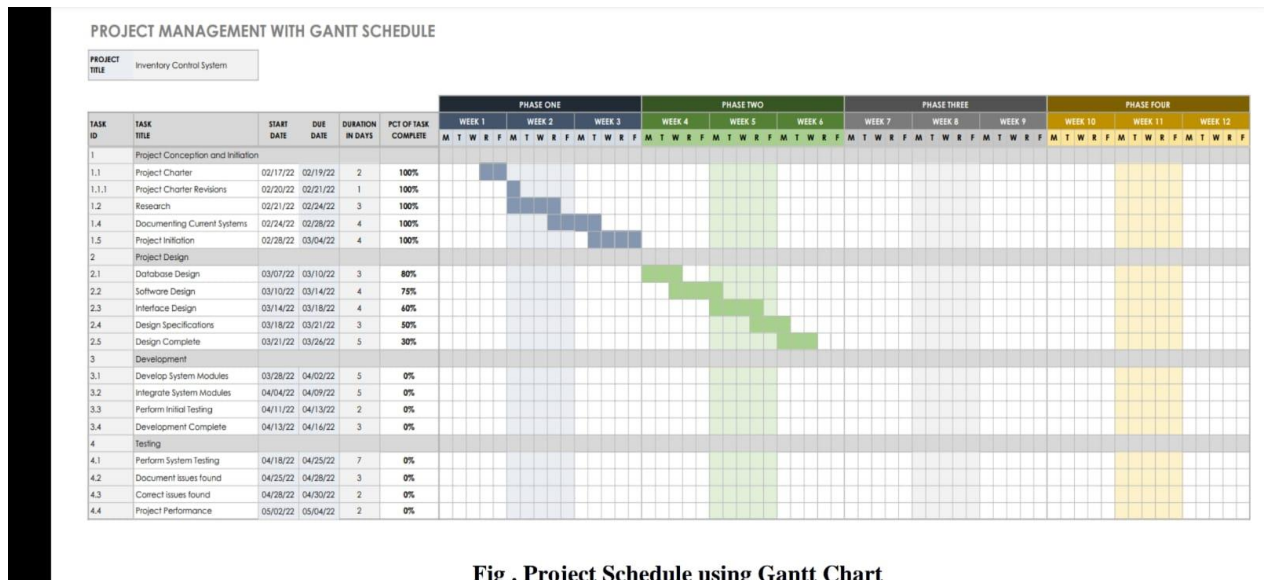


Fig . Project Schedule using Gantt Chart

## 10. Updated Budget

An updated budget is attached at the end of this document.

## 11. Appendices

### 11.1 Definitions, Acronyms, Abbreviations

- ICS - Inventory Control System
- DFD - Data Flow Diagram

## **Result and Discussion:**

**Learning Outcomes:** Students should have been able to understand

LO1: Define software System.

LO2: Identify different system requirements of software Architecture.

LO3: Explain system requirement of Domain-specific Software Development.

**Course Outcomes:** Upon completion of the course students will be able to understand System requirements for an Architecture for any specific domain.

## **Conclusion:**

## **Viva Questions:**

1. Define Software System.
2. Explain different requirements for creating a software Architecture.
3. Explain any five-system requirement.

For Faculty Use

Correction Parameters	Formative Assessment [40%]	Timely completion of Practical [ 40%]	Attendance / Learning Attitude [20%]	