



શ્રી સ્વામિનારાયણ ગુરુકુલ રાજકોટ સંસ્થાન

શાસ્ત્રી સ્વામી શ્રી ધર્મજીવનદાસજી

સાયન્સ & IT ગુરુકુલ કોલેજ

ગુરુકુલ કેમ્પસ, કોલેજ રોડ, જૂનાગઢ

Sales Report

Project Partners:

MR. Darshan P. Kalathiya, BCA-6TH

:: submitted to ::

BKNM University, Junagadh

:: GUIDED BY ::

Mr. Ripal V. Pandya

Mr. Milind V. Anandpara



શ્રી સ્વામિનારાયણ ગુરુકુલ રાજકોટ સંસ્થાન



શાસ્ત્રી સ્વામી શ્રી ધર્મજીવનદાસજી
સાયન્સ & IT ગુરુકુલ કોલેજ



ગુરુકુલ કેમ્પસ, કોલેજ રોડ, જૂનાગઢ

(Affiliated to **Bhakta Kavi Narsinh Mehta University, Junagadh**)

Project Completion Certificate

This certificate is awarded to

Kalathiya Darshan Premjibhai

BCA6-2024

in completion of project work

Asp.net

19/12/2023

SQL Server

11/03/2024

Mr. Ripal V. Pandya

Mr. Milind V. Anandpara

Project Guide

Director

www.sssdiit.junagadhgurukul.org



A

PROJECT REPORT ON

Sales Report

Submitted in Fulfillment of Requirements

For Completion of Semester - 6 in

Bachelor of computer application

Year 2023

To

SHASHTRI SWAMI SHREE DHARMAJIVANDASJI

INSTITUTE OF INFORMATION TECHNOLOGY

JUNAGADH

Guided By:

Ripal Pandya

Milind Anandpara

Prepared By:

Darshan Kalathiya

PREFACE

It is with great pleasure and a sense of accomplishment that we present this project report on the implementation of a mobile sales and supply record-keeping system. This project marks the culmination of our efforts during the sixth semester of our B.C.A at B.K.N.M University.

Project Overview:

The primary objective of this project is to design and develop a robust system for maintaining records of old mobile sales and supply. With the ever-evolving landscape of mobile technology, the need for efficient management of sales data and supply chain information becomes increasingly crucial. Our focus was on creating a user-friendly application that utilizes Swift as the programming language and CoreData for seamless data management.

Technology Stack:

In the pursuit of a reliable and scalable solution, we chose Swift as the programming language for the frontend development. Swift's expressiveness and safety, coupled with its compatibility with various iOS devices, made it the ideal choice for our application. CoreData, Apple's powerful and efficient framework for data storage and management, was integrated to handle the persistent storage of records.

Scope of the Project:

This project encompasses the entire life cycle of mobile sales and supply management, from the initial data entry to retrieval and analysis. We aimed to create a system that not only meets the basic requirements of record-keeping but also provides insightful analytics and reporting features for stakeholders.

Declaration:

We hereby declare that this project report represents our original work, and the codebase has not been submitted elsewhere. All external sources have been duly acknowledged, and the report adheres to the academic integrity standards set by B.K.M. University.

We hope that this project proves to be a valuable contribution to the domain of mobile sales and supply management. It is our sincere desire that this application will find practical utility in real-world scenarios.

Darshan Kalathiya

B.K.M. University

ACKNOWLEDGEMENT

We are very thankful to all whose have helped in preparing this project. We are feeling a great happiness to present this website project. First of all we would like to thank “**BKNM University**” who give me an opportunity to give a chance to prepare a project.

Before we get in to thick of the things we would to add a few heartfelt words for the people who were part of this project numerous ways, people who give unending support right from the stage project ideas was conceived. In particular we would like to thank **Ripal Pandya, Milind Anandpara (Project Guide)**, who has always inspired us and has directed us towards the successful completion of our project. They have been the guided through the project and their encouragement has left me indebted to them.

We are very thankful to the **Director Sadhu RushikeshdashjiSwami** and the **Asst. Director Mr. Rajesh Bharad of Shastri Swami Shree Dharmajivandasji Institute of Information Technology – Junagadh.**

Thank you...

Date:

Darshan Kalathiya

Place: Junagadh

INDEX

NO	Particulars	Page No
1	Project Profile	1
2	Use of System Development Life Cycle Model	2
3	Feasibility Study	5
4	Requirement Gathering	9
	Requirement Analysis 1) Hardware and Software Requirement 2) Front - End Tools 3) Back - End Tools 4) Other Tools & Technology Used	10
5	Project Abstracts (User Roles & Capabilities)	12
6	Proposed System	13
7	Advantages & Limitations of Proposed System	14
8	Evaluative Report Using Pert Chart and Gantt Chart	15
9	Data Flow Diagram 1) Context level 2) 1st Level 3) 2nd Level	18
10	Use Case Diagram	23
11	Flow Chart	25
12	Cost Estimation	27
13	Data Dictionary	28
14	Screen Layouts	30
15	Special Utilities	38
16	Testing	39
17	Implementation	45
18	Bibliography	46

PROJECT PROFILE

Project Title	Sales Report
Project Description	This project focuses on developing a streamlined record-keeping system for tracking old mobile sales and supply. The objective is to create a user-friendly application for efficient data management, offering a comprehensive solution to monitor and analyze transactions in the second-hand mobile market.
Front End	Swift (X-Code IDE)
Back End	SQLite (Core data)
Guide	Ripal Pandya Milind Anandpara
Submitted To	B.K.N.M University, Junagadh (S.S.S.D.I.I.T Collage)

USE OF SYSTEM DEVELOPMENT LIFE CYCLE MODEL

Software Development Life Cycle (SDLC) is a process for development of software. There are some steps to follow to create a software application.

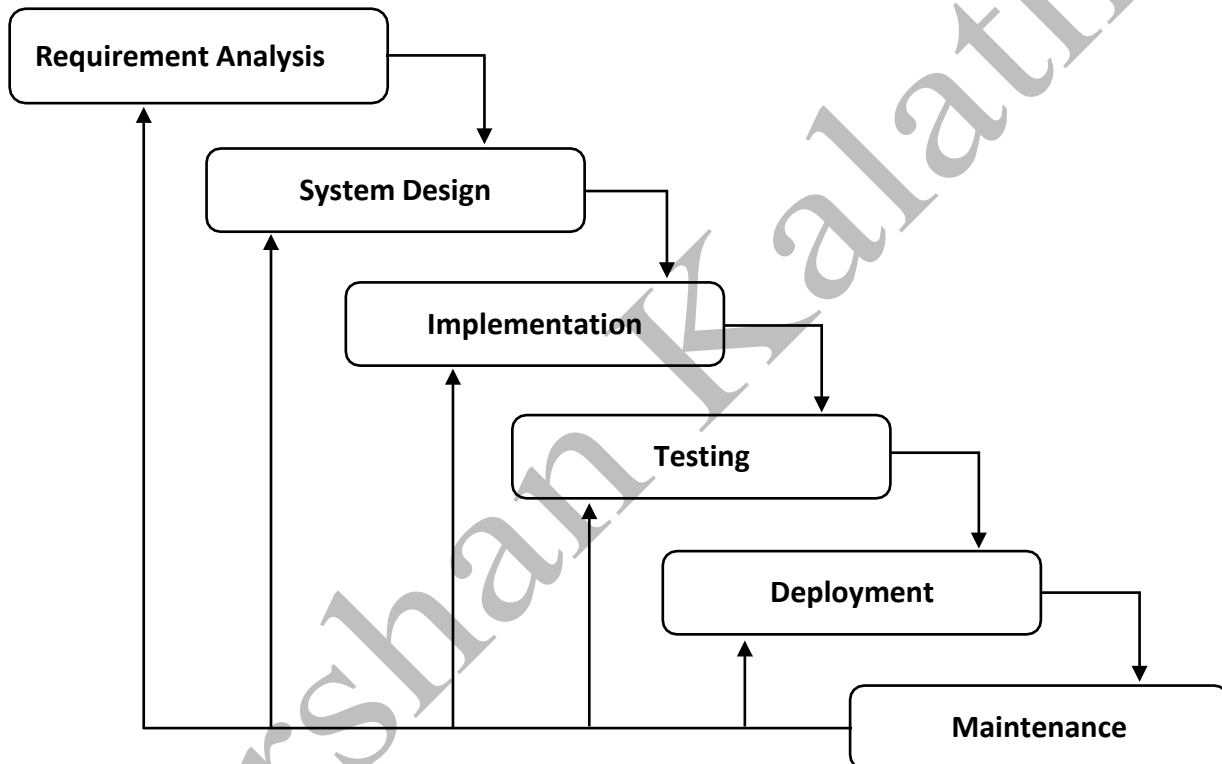
In an SDLC the steps follow requirement gathering. In requirement gathering questionnaire, personal interview etc. are the method for gathering information. Analysis phase includes creating Software Requirement Specification and analyze the gathered data. In design phase, design of Software application i.e. database design and GUI design have to be prepared. In coding phase, coding is done of different modules and forms. In testing phase, the different type of testing is done like integration testing, unit testing, system testing and at last the created software is implemented and maintained.

Following are the different Life Cycle Model example:

- Waterfall model
- Iterative waterfall model
- Prototyping model
- Evolutionary model
- Spiral model
- R.A.D. model (Rapid Application Development)

Waterfall Model:

In the Waterfall Model, software development progresses through sequential phases, each building upon the outputs of the previous phase. The model begins with requirements analysis, followed by system design, implementation, testing, deployment, and maintenance. Each phase is distinct and has specific deliverables, with progress flowing downwards like a waterfall. This linear approach provides clarity and structure but lacks flexibility for changes later in the development process.



1) Requirement Gathering and analysis:

All Possible requirements of the system to be developed are captured in this phase and documented in a requirement specific doc.

2) System Design:

The requirement specification from first phase is studied in this phase and system design is prepared, system Design helps in defining overall system architecture.

3) Implementation:

With Inputs from system design the system is first developed in small programs called nits which are integrated in the next phase, each unit is developed and tested for its functionality which is referred to as unit testing.

4) Integration and Testing:

All the Units Developed in the Implementation phase are integrated into a system after testing of each unit post integration the entire system is tested for any faults and failures.

5) Deployment of System:

Once the functional and non-functional testing is done the product is deployed in the customer environment or released into the market.

6) Maintenance:

There are some issues which commune in the environment to fix those issue patches are released, also to enhance the product some better versions are released.

FEASIBILITY STUDY

Feasibility of a project determines whether it is possible to develop the project. These are four main factors, which determine the feasibility of the project. They are discussed as follows.

The main aim of feasibility study is to determine whether developing the project is functionally and technically feasible or not.

The feasibility study involves analysis of the problem and collection of data which would be input to the system, the processing required to be carried out on this data, the output data required to be produced by the system, as well as study of various constraints on the behavior of the system.

An initial determination in a proposal that whether an alternative system is feasible or not. To determine feasibility of candidate system in all respects I need to consider following feasibility factors:

There are three types of feasibility study.

- 1) Technical
- 2) Operational
- 3) Economical

Technical feasibility considers whether the desired project can be completed within the framework of available technology. As our project is developing the website, this is not much of a problem because there are many advanced web editing tools available.

Operational feasibility was done to ensure that each model of mobile will come in the future will be changed so the database will be changed for each mobile model. Therefore, the site should provide all functionality for a proper website. Since this website has changeable resources by that we visit some informative website for the help to this website to satisfy the scope of the operational feasibility.

Technical Feasibility:

The main aim of technical feasibility study is to determine whether it is possible to develop the proposed system with the present technologies available and study the technical requirements and their availability in the organization & the technical equipment availability in market.

So, in this project technical requirements are: -

Hardware:

- 1) A Mac OR Computer With
- 2) 8 GB RAM
- 3) 256 GB SSD
- 4) Mac OS Operating System
- 5) IOS Simulator

Software:

- 1) XCode
- 2) IOS SDK
- 3) Swift
- 4) Developer Account

Economic Feasibility:

The economic feasibility takes into consideration the financial matters regarding the proposed system. The organization measures the cost effectiveness of the proposed system. The economic feasibility of the proposed system is as under budget of a company or not! This is checked in economic feasibility.

- The Cost of development is approximat – 7000
- Domain cost's – 0
- The design will a coadding to local and near to - 2000
- Other software cost's – 0
- The client can afford the cost of project.

Development Cost: -	7000
Domain & Hosing Cost: -	0
Design Cost: -	2000
Software & Other Cost: -	0
Total: -	9000/-

Operational Feasibility:

The Operational feasibility deals with the matter whether the proposed system fulfills the requirements of the organization. This feasibility determines whether the proposed system covers all the aspects of the current system & gives an extra facility which is nothing current system.

The project requires one person who has knowledge of basic computer fundamental. The client has one computer operator who can handle, the software. the operational feasibility is as follows. The proposed system will fulfill the organization's requirements.

- The proposed system covers all aspects of the working current manual system.
- The human sources required for proposed system.
- Staff is totally operational.
- Easy to manage with organization.

REQUIREMENT GATHERING

Questionnaire:

1) What do you want to create an Application, Software or Website?

- I want create an Application.

2) For Which purpose do you want to create app?

- The app is created to efficiently manage and record transactions in the market of old mobile sales and supply.

3) In the application, which specific data points are crucial for you to store when recording transactions in the old mobile sales and Purchase?

- IMEI, Mobile Number, Name, Address, Purchase price, Selling price, and Bill date.

4) Which platform do you intend to build the app for - iOS, Android, or both?

- The app will be built specifically for the iOS platform.

5) What is the desired timeframe for app development and launch?

- The target timeframe for app development and launch is six months.

6) Estimated cost range of Rs 13,000 to Rs 16,000 acceptable for the development of the application?

- Yes, the estimated cost range is acceptable for the development of the application.

7) Do you have any preferences for the user interface and design elements of the app? Any specific color schemes or branding guidelines to be followed?

- No, we prefer a clean and intuitive interface without specifying any particular color scheme, allowing for flexibility in design decisions.

8) Is there a need for user authentication and access control?

- No, user authentication and access control are not required for this application.

REQUIREMENT ANALYSIS

In the process of requirement analysis for the Sales Report project, the primary objective was to ascertain the specific needs and functionalities required by the project stakeholders, particularly focusing on the administrative aspect of the application.

The following considerations were made during this phase:

1. Stakeholder Engagement:

Engaging with stakeholders, particularly the project sponsor and end users, to understand their expectations and requirements from the Sales Report application.

2. Identification of User Roles:

Recognizing that the project is solely intended for administrative purposes, the emphasis was placed on understanding the specific roles and responsibilities of the admin user.

3. Functional Requirements:

Gathering detailed functional requirements, such as the ability to add, update, and delete sales records, manage user accounts, and perform administrative tasks within the platform.

4. Non-functional Requirements:

Identifying non-functional requirements, including security measures to safeguard sensitive data, user interface considerations for an intuitive admin interface, and performance criteria to ensure efficient system operation.

5. Documentation:

Documenting all gathered requirements in a clear and concise manner to serve as a reference throughout the development lifecycle and facilitate communication among project team members.

6. Validation and Verification:

Ensuring that the gathered requirements align with the project objectives and are validated and verified by stakeholders to mitigate any misunderstandings or discrepancies.

Through meticulous requirement analysis, the project team aims to establish a comprehensive understanding of the admin's needs and expectations, laying the groundwork for the successful development and implementation of the Sales Report application.

PROJECT ABSTRACTS

The Sales Report project aims to streamline and automate the process of tracking sales data for an organization. In today's competitive market, efficient sales management is paramount for business success. This project provides a centralized platform for administrators to manage sales records effectively, facilitating informed decision-making and optimizing sales strategies.

Key Features:

- Admin can execute purchasing and sales transactions utilizing the supplied models, incorporating IMEI tracking.
- Admin Can both submit and access customer feedback for enhanced engagement and service improvement.

With its user-friendly interface, comprehensive features, and focus on data accuracy and security, the Sales Report project serves as a valuable tool for organizations seeking to enhance their sales management processes and drive business growth.

PROPOSED SYSTEM

The proposed system is an intuitive and user-friendly platform designed to revolutionize the process of managing sales records and customer feedback. Key features of the proposed system include:

1) Streamlined Sales Record Management:

The system provides administrators with a centralized interface to efficiently add, update, and delete sales records. Through seamless integration with IMEI tracking, it ensures accurate and detailed transaction records.

2) Comprehensive Customer Feedback Module:

Users can easily submit feedback through a dedicated module, enabling businesses to gather valuable insights into customer satisfaction and preferences. Administrators can view and analyze feedback data to identify areas for improvement and enhance customer experience.

3) Interactive Dashboard:

The system offers an interactive dashboard for administrators, providing real-time insights into sales performance, customer feedback trends, and inventory management. Customizable widgets and charts empower users to visualize data and make informed decisions.

4) Scalability and Flexibility:

Built with scalability in mind, the system can adapt to the evolving needs of businesses of all sizes. Its modular architecture allows for seamless integration with additional features and third-party applications as required.

Overall, the proposed system offers a comprehensive solution for businesses to streamline sales record management, gather valuable customer feedback, and drive continuous improvement in customer satisfaction and sales performance.

ADVANTAGES & LIMITATIONS OF PROPOSED SYSTEM

Special Features:

- Efficient transaction tracking with IMEI integration ensures accurate sales and purchase records.
- Seamless management of customer feedback fosters proactive service enhancement and customer engagement.
- Streamlined administrative processes enhance productivity and decision-making.
- Comprehensive data insights enable informed strategic planning and business growth.
- Enhanced customer satisfaction through personalized feedback interaction and swift resolution of issues.

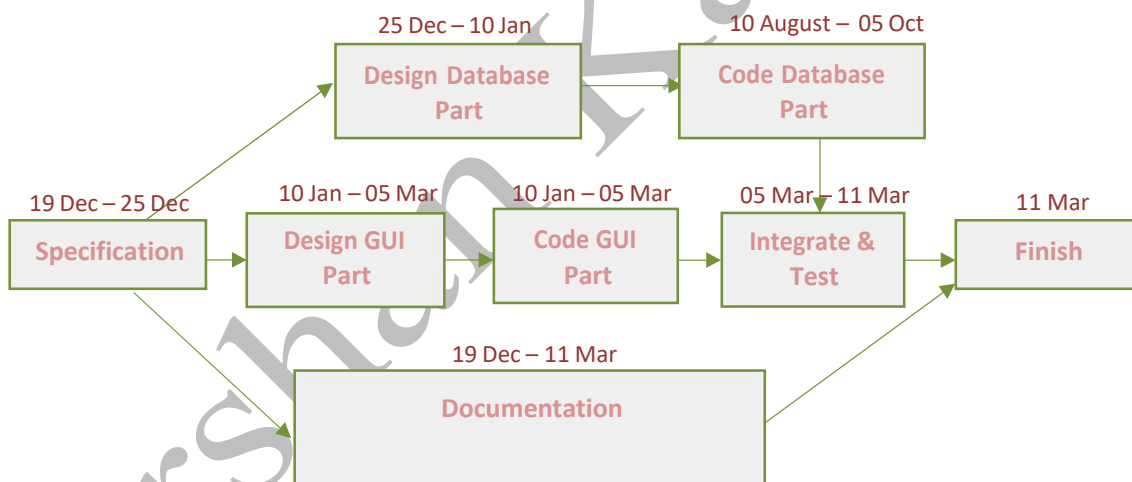
Disadvantage:

- Limited scalability and performance compared to robust database systems like MySQL or PostgreSQL.
- Potential data loss or corruption risks in case of system crashes or failures due to the lack of robust transaction management features.

PERT CHART AND GANTT CHART

Pert Chart:

PERT (Project Evaluation and Review Technique) charts consist of a network of boxes and arrows. The boxes represent activities and the arrows represent task dependencies. PERT charts are a more sophisticated form of activity chart. Where instead of making a single estimate for each task, pessimistic, likely and optimistic estimates are made. The boxes of PERT charts are usually annotated with the pessimistic, likely, and optimistic estimates for every task. There are thus not one but many critical paths, depending on the permutations of the estimates for each task. This makes analysis of critical path show by using shaded boxes. The PERT chart representation of the MIS problem of show follows.



Gantt chart can be derived automatically from PERT charts. However, PERT charts cannot be automatically derived from Gantt charts because PERT charts incorporate additional information about the time when an engineer does a task. This information is not available is helpful in planning the utilization of resources, while the PERT charts is more useful for monitoring the timely progress of activities. Also, parallel activities in a project can be easily identified using a PERT chart.

Gantt CHART:

PERT Gantt charts are mainly use of scheduling, budgeting, and resource planning. It allocates resource to activity include Staff, Hardware, Software, etc...

A Gantt chart is a special type of bar chart where each bar represents an activity. The bars are drawn along a time line. The length of each bar is proportional to the duration of time planned for the corresponding activity.

19/12/2023	25/12/2023	10/01/2024	05/03/2024	05/03/2023	11/03/2024
START					
Requirement Specification					
	Design Database				
		GUI Design			
		Code of GUI & Database			
					FINISH
			Integration & Testing		
		Documentation			

Gantt charts used in software project management are actually an enhanced version of the software project management. Each bar consists of a white part and a shaded part. The white part of the bar shows the length of time each task is estimated to take. The shaded part of the bar shows the slack time.

In order to estimate the time durations for various activities, usually managers let the engineers themselves estimate the time for an activity they might be assigned to. However, some managers prefer to estimate the time for various activities themselves. Many managers believe that an aggressive schedule motivates the engineers to do a job better and faster.

However, careful aspects, but also cause schedule compromise on intangible quality aspects, but also cause schedule delays. A good way to achieve accuracy without creating problems is to let people set their own schedules.

We can see that one engineer can do the database design and then code the database design whereas another engineer can design the GUI part, code the GUI part, and still have time left for writing the user manual. Thus, Gantt charts are very useful in scheduling resources.

So here, I have to follow the scheduling steps for my project.

Gantt chart is really useful us for planning software application resources.

DATA FLOW DIAGRAM

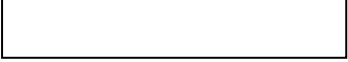
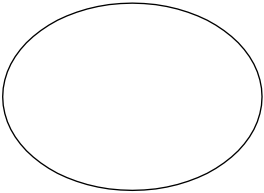
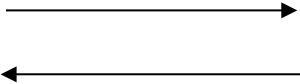
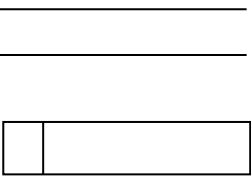
Detailed Life Cycle of Project:

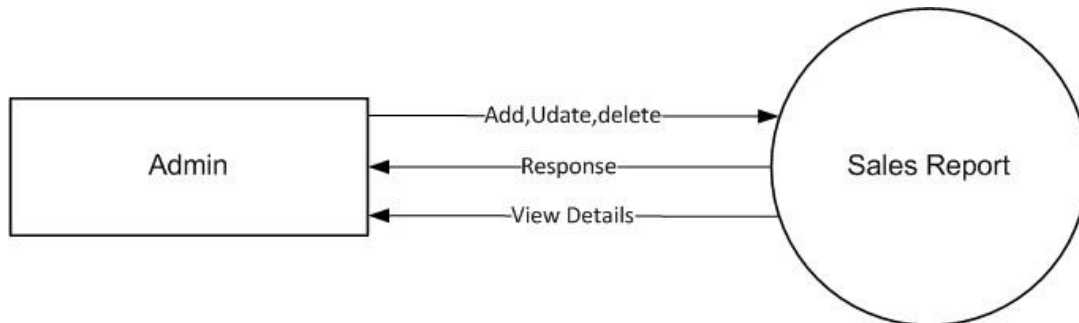
In the discussion of “Detailed Life Cycle of Project” we have to concentrate on DFD (Data Flow Diagram). Here we have work on it while developing this software project.

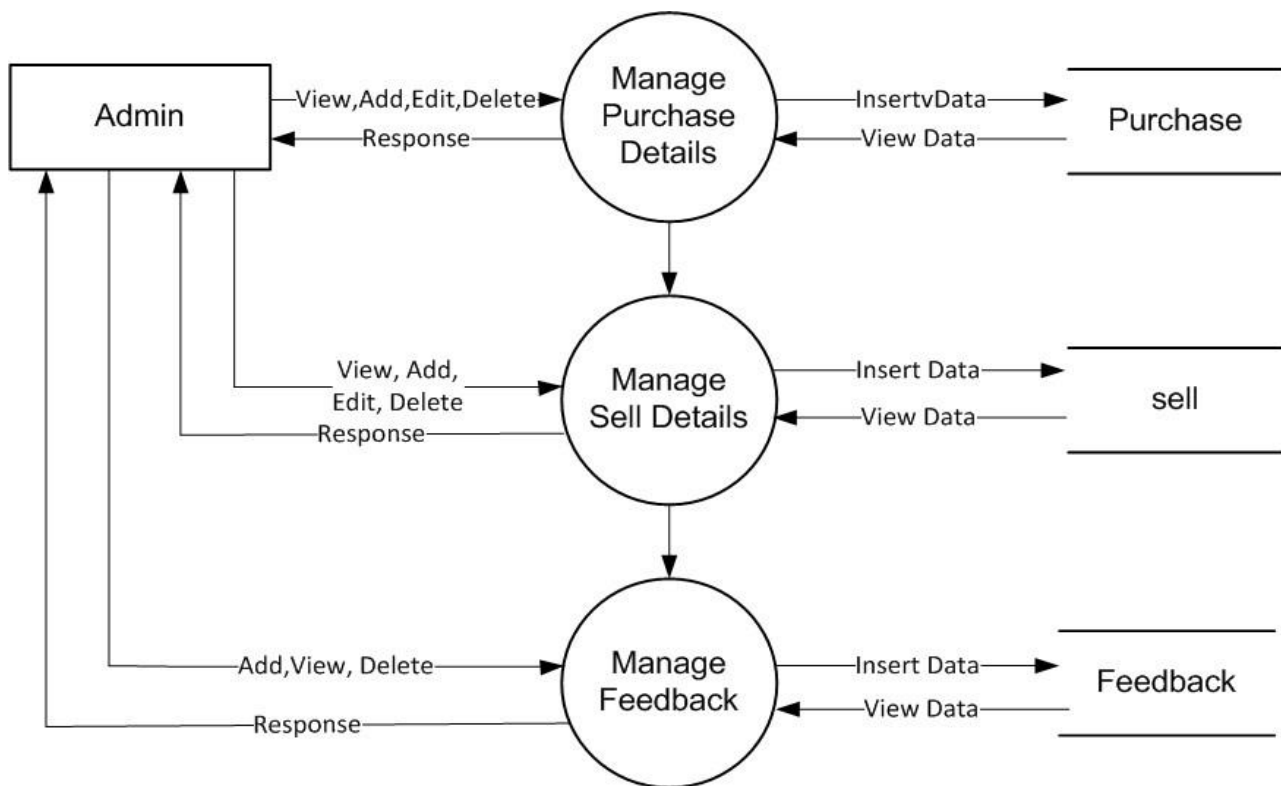
DFD (Data Flow Diagram):

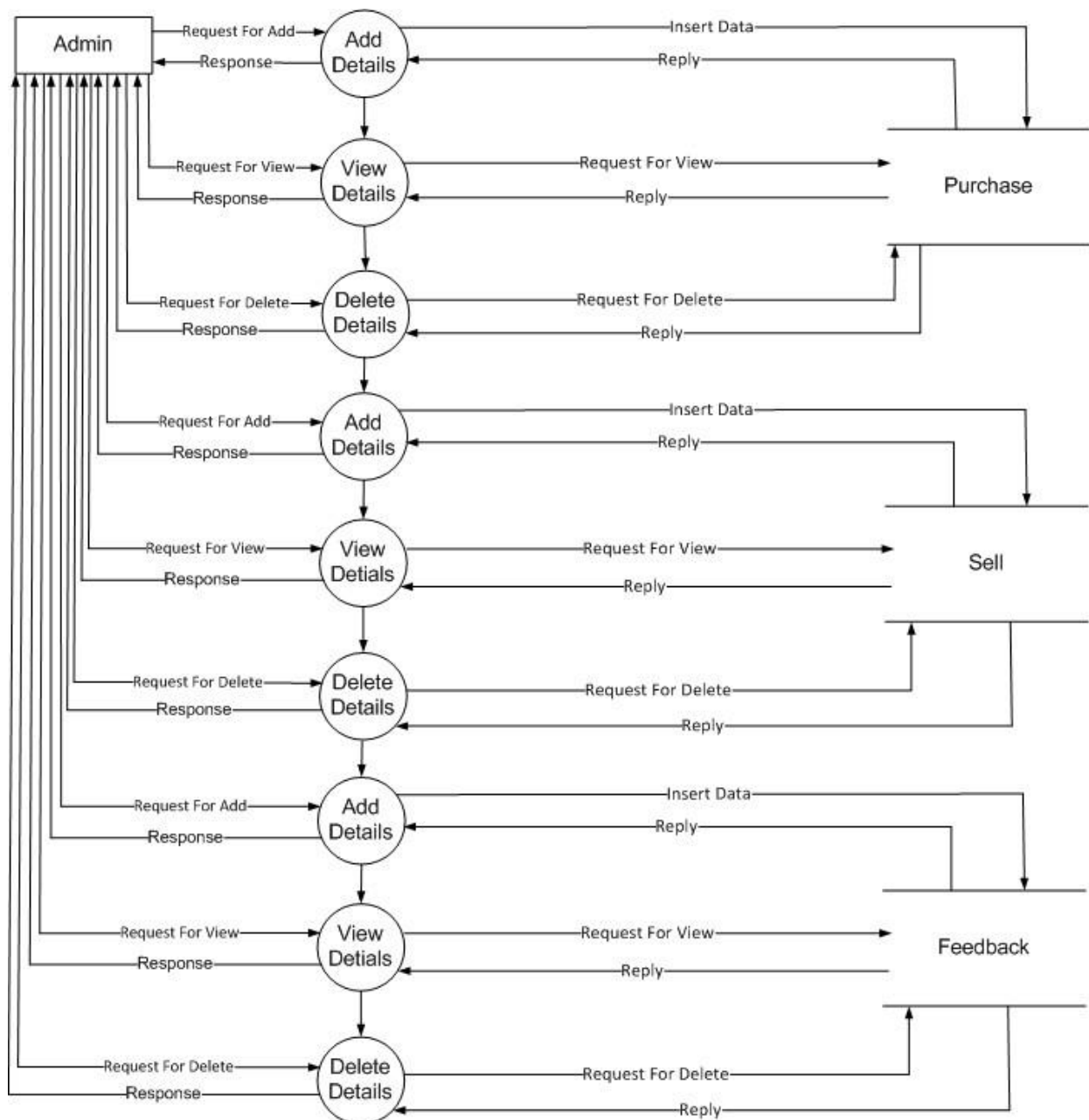
DFD is a graphical view of all system processes and transactions. With the DFD an End-User also can easily understand the system in a short time period. Also, it is useful to find out problems or any complications with the system we are going to develop. We can easily get that whether we have understood the system as per the requirements of the customer or not by showing them this diagram. Thus, DFD is a necessary phase while developing software.

For Understanding:

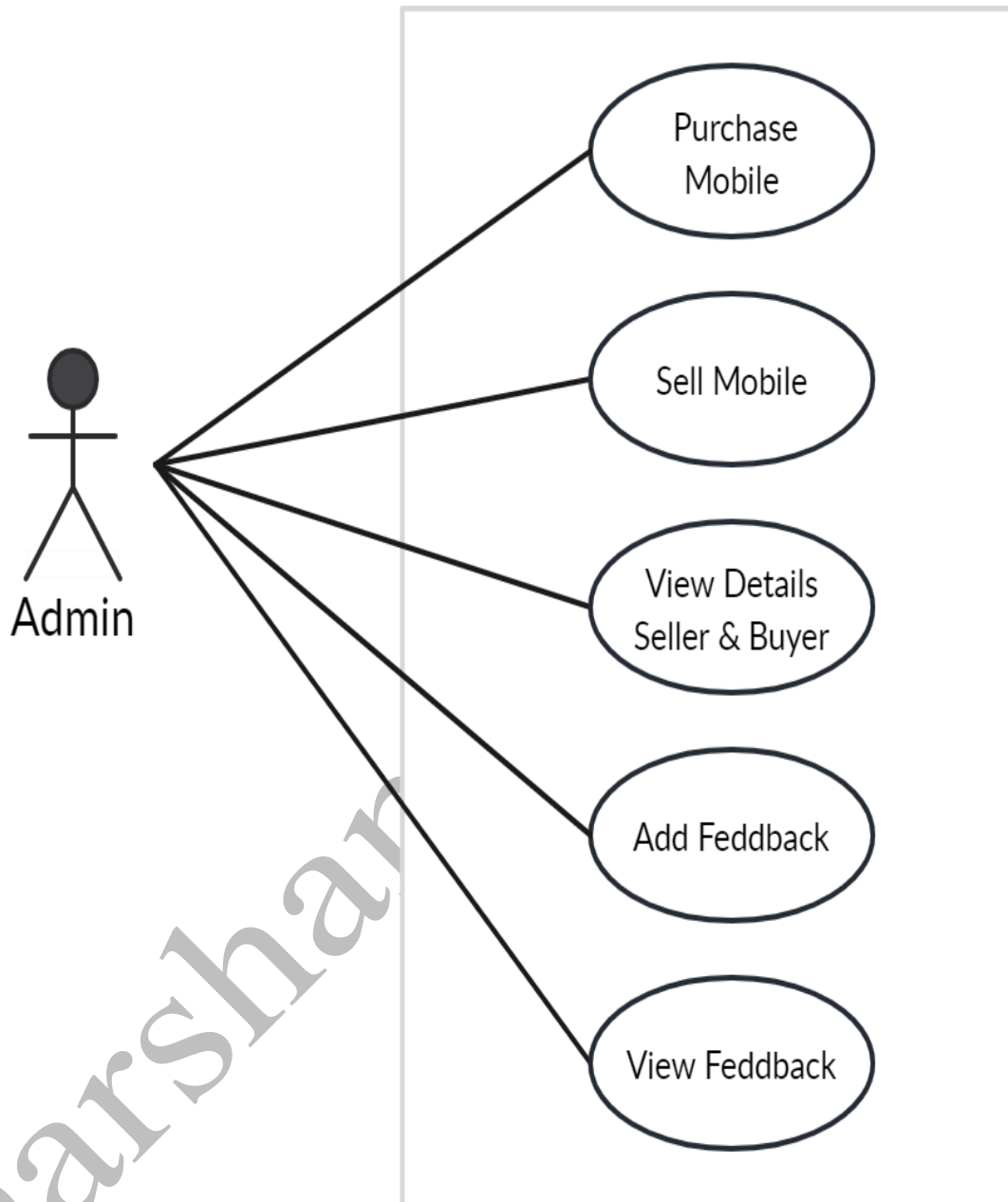
Symbol	Name	Use
	External Entity	Rectangle source and / sink destination data.
	Process / Function	Transformed, Store, or Distribute. Annotated with number and name of function.
	Data Flow	Direction of data flow single piece of data or logical collection of data.
	Data Store	Open Rectangle Parallel Lines Data Structure, File, Table, Database.

Context Level Diagram:

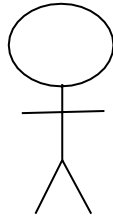
1st Level Diagram:

2nd Level Diagram:

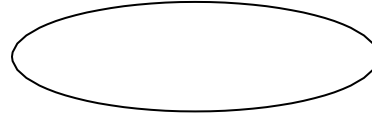
USE CASE DIAGRAM



A use case is a set of scenarios that describing an interaction between a user and a system. A use case diagrams the relationship among actors and use cases. The two main components of a use case diagram are use cases and actors.



Actor



Use Case

An actor is representing a user or another system that will interact with the system you are modeling. A use case is an external view of the system that represents some action the user might perform in order to complete a task.

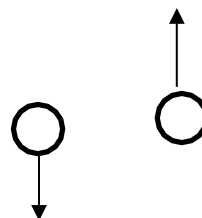
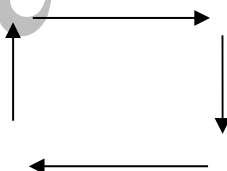
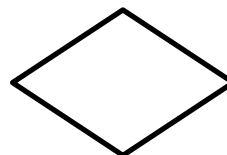
Actor and use case description show the details description of interaction between the actor.

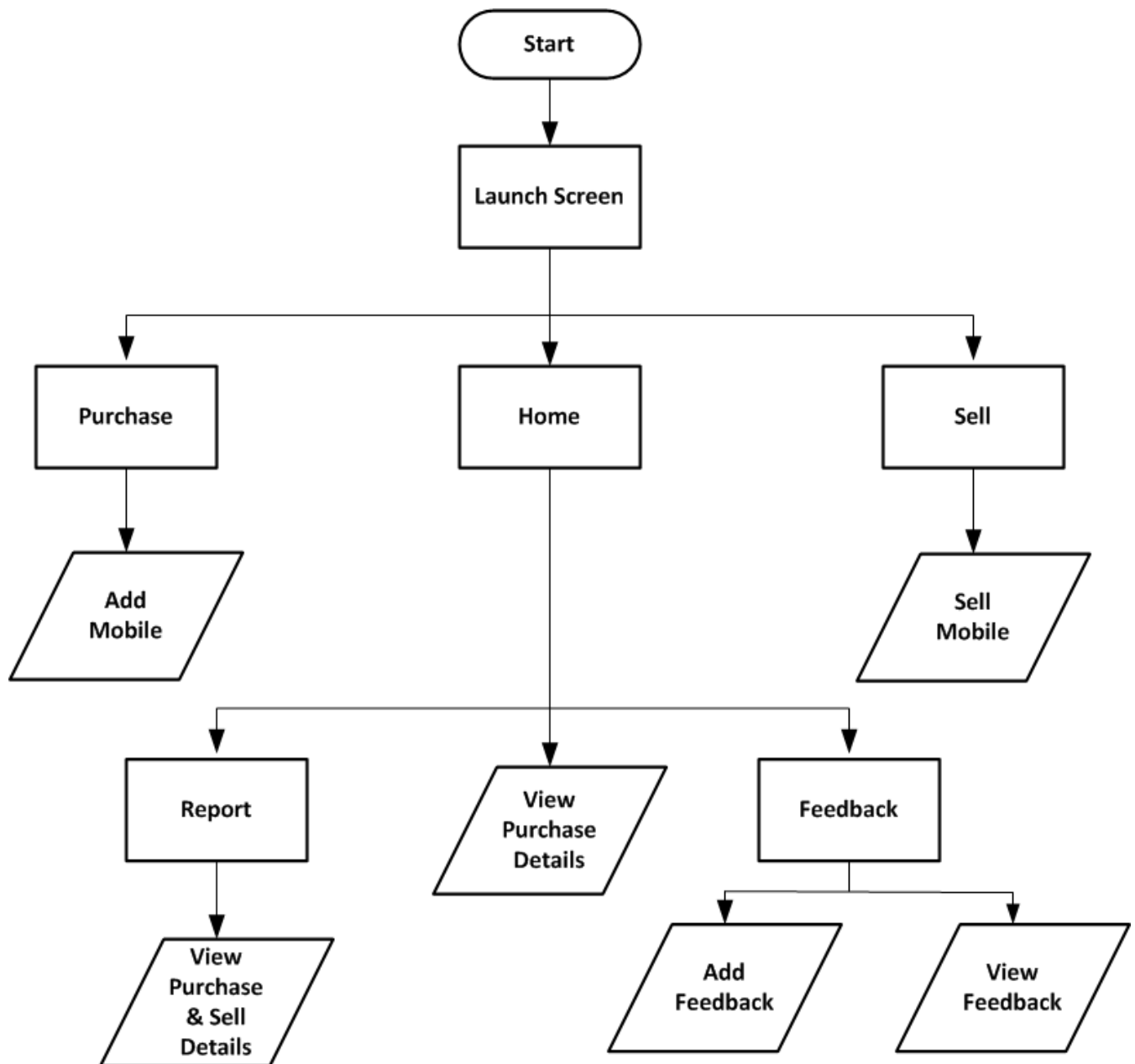
And their use case the description enables have a proper understanding of how actor interact.

FLOW CHART

A Flow is a pictorial representation of an algorithm. Programmers often use it as a program-planning tool for visually organizing a sequence of steps necessary to solve a problem using computer. It uses boxes of different shapes to denote different type of instructions. The actual instructions are written within these boxes using clear and concise statements. Solid lines having arrow marks connect these boxes to indicate the flow of operation, that is, the exact sequence in which to execute the instructions. The process of drawing a flowchart for an algorithm is known as flowcharting.

Basic Flowchart Symbols:



Administrative Login Flowchart:

COST ESTIMATION

In my project, I am implementing a Module Wise to calculate and evaluate the per-Screen average cost. This approach allows for a comprehensive assessment of cost allocation across various components, resulting in a more accurate and informative cost analysis.

Module Wise:

Per-Screen 1500Rs Cost.

Total Price of Screens

$$6 * 1500 = 9,000.00/-$$

$$\text{Total} = 9,000.00 \text{ /-}$$

The final cost of the application project is Rs 9,000.00/-.

DATA DICTIONARY & NORMALIZATION

Database Name: Sales_report

Table 1: Purchase

Column Name	Datatype (Size)	Constraints	Remarks
id	UUID	A.I, P. K	
cname	String		
cmobile	String		
caddress	String		
aadharcard	String		
imei	String		
billdate	String		
mname	String		
price	String		
purchasedate	String		
status	String		

Table 2: Sell

Column Name	Data Type (Size)	Constraints	Remarks
id	UUID	A.I, P. K	
name	String		
mobilenno	String		
address	String		
aadharcard	String		
imei	String		
mname	String		
price	String		
selldate	String		

Table 3: Feedback

Column Name	DataType (Size)	Constraints	Remarks
id	UUID	A.I, P. K	
name	String		
mobilenno	String		
address	String		
message	String		

SCREEN LAYOUTS

LaunchScreen

8:25



Sales Report

```
func application(application: UIApplication, didFinishLaunchingWithOptions
launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
Thread.sleep(for Time Interval: 3)

return true
}
```

HomeScreen



```

Let context=(UIApplication.shared.delegate as!
AppDelegate).persistentContainer.viewContext

let fetchRequest: NSFetchRequest<Purchase> = Purchase.fetchRequest()
let predicate = NSPredicate (format: "status CONTAINS %@",
"Purchase") fetchRequest.predicate = predicate
let sortDescriptor = NSSortDescriptor(key: "id", ascending: true)
fetchRequest.sortDescriptors = [sortDescriptor]
do {
    fetchedData = try context.fetch(fetchRequest)
    tableView.reloadData()
    return fetchedData.first
} catch{ }

```

PurchaseScreen

12:16

Purchase Mobile

Model Name
Required

Imei
Required

Bill Date
08-02-2024

Customer Name
Required

Mobile Number
Required

Customer Address
Required

Aadhar Card Number
Required

Purchase Amount
Required

Purchase

+

Purchase Home Sell

```

let purchase Purchase(context: context)
purchase.id = UUID()
purchase.mname = mname
purchase.imei = imei
purchase.cname = name
purchase.billdate = billdate
purchase.cmobile = mobile
purchase.caddress = address
purchase.aadharcard = aadharcard
purchase.price = productprice
purchase.status = status
purchase.purchasedate = formattedDate
// database reflect
do {
    try context.save() }catch{ }

```

SellScreen

12:26

Sell Mobile

Imei
Required

Model Name

Customer Name
Required

Mobile Number
Required

Customer Address
Required

Aadhar Card Number
Required

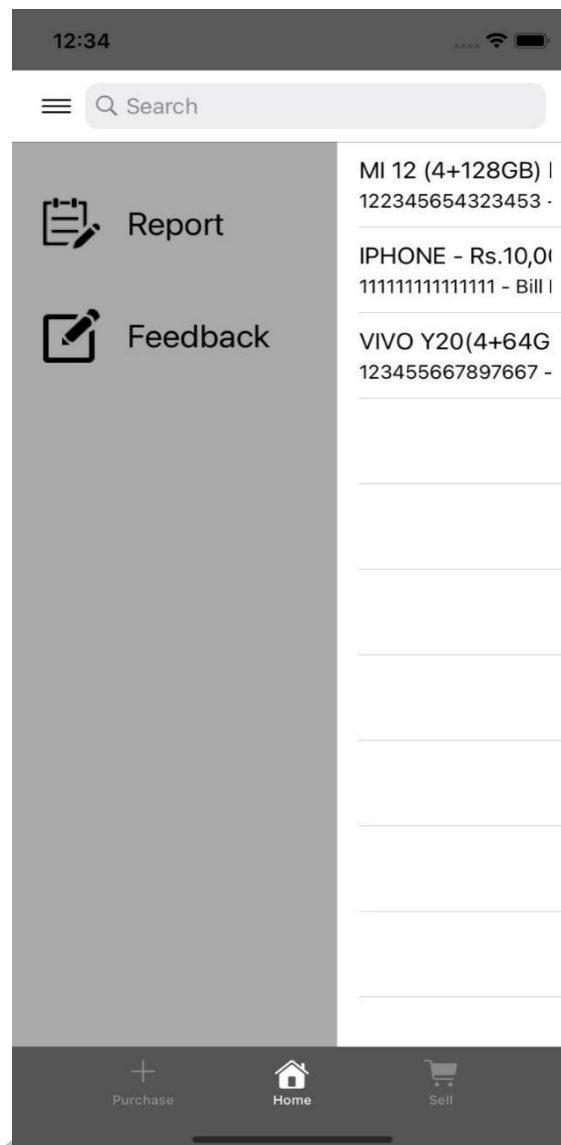
Selling Amount
Required

Sell

Purchase Home Sell

```
do {
//update status
let matching = try context.fetch(fetchRequest)
    if let found Purchase = matching.last
    {
        found Purchase.status = "Sold"
        try context.save()
    }
    else
    {
        print("staus note upate ")
    }
}
try context.save()
} catch {}
```

SlideMenu



```
@IBAction func openMenu(_ sender: Any) {
    if(showMenu)
    {
        leadingConstraint.constant = -240;
    }
    else{
        leadingConstraint.constant = 0;
    }
    showMenu = ! showMenu;
}
```

ReportScreen

```
customCell.PurchaseDetailLabel.text = "Purchase Detail"
```

```
let fetchimei: NSFetchRequest<Purchase> = Purchase.fetchRequest()
```

```
fetchimei.predicate = NSPredicate (format: "imei == %@", conformimei!)
```

```
do {
```

```
    let fetchpurchaseimei = try context.fetch(fetchimei) if
```

```
    let ipurchase = fetchpurchaseimei.last
```

```
    customCell.CustomerName.text = "Name: \(ipurchase.cname ?? "")"
```

```
    customCell.MobileNo.text = "Mobile No: \(ipurchase.cmobile ?? "")"
```

```
    customCell.Address.text = "Address: \(ipurchase.caddress ?? "")"
```

```
    customCell.AddarcadNO.text = "Addarcad: \(ipurchase.aadharcard ?? "")"
```

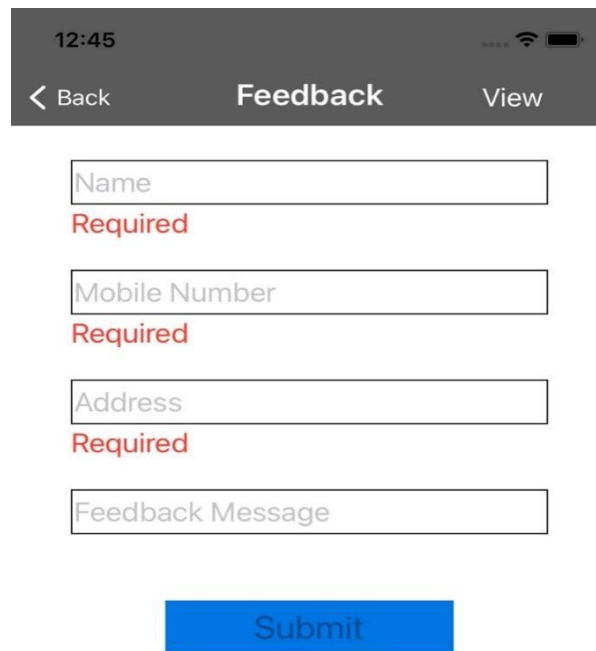
```
    customCell.DateTime.text = "Date & Time: \(ipurchase.purchasedate ?? "")"
```

```
    customCell1.PurchasePrice.text = "Rs. \(ipurchase.price ?? "").00"
```

```
}
```

```
}catch{ }
```

FeedbackForm



12:45

< Back Feedback View

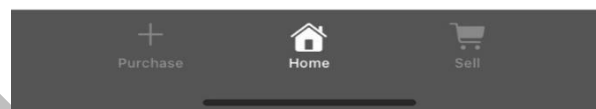
Name
Required

Mobile Number
Required

Address
Required

Feedback Message

Submit

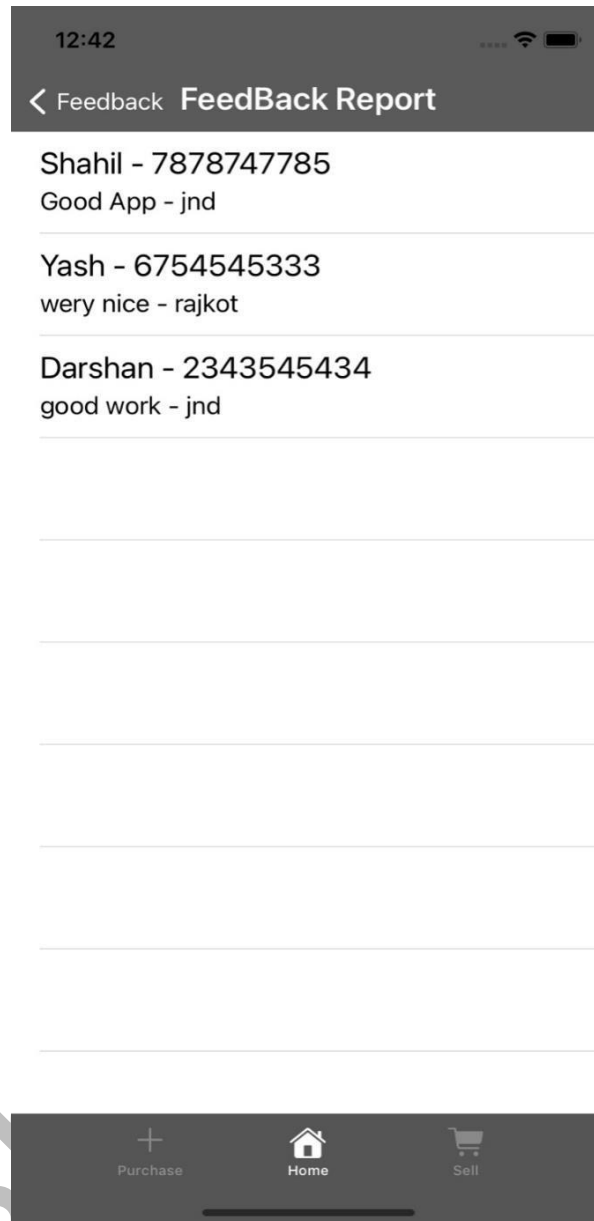


```

let name = nameTF.text
let address = addressTF.text
let mobile = phoneTF.text
let message = messageTF.text
let feedback = Feedback(context: context)
feedback.id = UUID()
feedback.name = name
feedback.address = address
feedback.mobilenno = mobile
feedback.message = message
// database reflect
do {
try context.save()
} catch { }

```

Feedback Report



```

@objc func fetchItems()
let context = (UIApplication.shared.delegate as!
AppDelegate).persistentContainer.viewContext
let fetchRequest: NSFetchedRequest<Feedback> = Feedback.fetchRequest()
let sortDescriptor = NSSortDescriptor(key: "id", ascending: false)
fetchRequest.sortDescriptors = [sortDescriptor]
do{
    fetchedData = try context.fetch(fetchRequest)
    tableView.reloadData()
} catch {}

```


SPECIAL UTILITIES

Following are special utilities provided by Sales Report.

- Seamless IMEI tracking integration ensures accurate inventory management.
- Mobile-responsive interface for on-the-go sales and inventory management.
- Intuitive data visualization tools facilitate informed decision-making.
- Real-time customer feedback collection for immediate service improvements.
- Seller and purchaser data are seamlessly linked with sales transactions for comprehensive reporting and in-depth analysis.

TESTING

Software Development Life Cycle (SDLC) includes a series of production activities one of this is testing.

Testing is a process of executing a program with the intent of finding an error.

Testing is the most important element to be considered for providing quality software and it represents the ultimate review of specification, design and coding.

The success or failure of the software as a system mainly depends on testing.

Software Developer spends 40% to 50% of their total development time on testing.

There are several SDLC techniques and development model. I have focused on Prototype Model. I have followed the prototyping model to develop this Software.

The development of software system involves a series of production activities where opportunities for injection of human fallibility are enormous. Error may begin to occur at the very inception of the process where the objectives may be erroneously or imperfectly specified, as well as later design and development states. Because of human inability to perform and communicate with perfection, software development is accompanied by a quality assurance activity.

Testing is program consists of providing the program with a set of test inputs and observing if the programs behave as expected. Under which a failure occurs are noted for debugging and correction. The following are some commonly used terms associated with testing.

A failure is manifestation of an error. But, the mere presence of an error may not necessarily lead to a failure.

A fault is an incorrect intermediate state that may have been entered during program execution. A fault may or may not lead to a failure.

A test case is the triplet [I.S.O.], where I is the data input to the system, S is the state of the system at which the data is input, and O is the expected output of the system.

A test suite is the set of all test cases with which a given software product is to be tested.

Many types of testing techniques are describing as follows.

Unit Testing:

Unit testing is under taken when a module has been coded and successfully reviewed in this section we first discuss the environment needed to perform unit testing.

Here in this project we test each and every module and forms of software application individually when it is completely coded.

There are some methods for unit testing are as follows.

Black-Box Testing:

- Equivalence Class Partitioning
- Boundary Value Analysis

White-Box Testing:

- Statement coverage
- Branch Coverage
- Condition Coverage

- Path Coverage
- Linearly independent Path
- Data Flow - Based Testing
- Mutation testing

Integration Testing:

The primary objectives of the integration testing is to test the module interface in order to ensure that there are no error in parameter passing when one module invokes another module.

During integration testing different module of system as per integration plane the integration plan specify the steps and the order in which module are combine to realize the full system.

After each integration test the practical integrated system is tested. Following are the integration testing Methods & Approaches

- Big bang approach
- Top down approach
- Bottom up approach
- Mixed approach

System Testing:

In the system testing the whole application is tested and the error and failure possibility is carried out in it.

Following are the method & approach of system testing.

- Alpha testing
- Beta testing
- Acceptance testing

- Performance testing
- Error seeding

Testing is a process of executing a program with the intent of finding an error.

A good test case is one that has a high probability of finding an as yet in discovered error.

A successful test is one that uncovers a yet undiscovered error.

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. The increasing visibility of software as a system element and the attendant “Cost” associated with a software failure are motivating forces for well – planned, thorough testing. It is not unusual for a software development organization to expend between 30 to 40 percent of total project effort on testing. In the extreme, testing of human-rated software can cost three to five times as much as all other software engineering activities combined.

TEST CASE

Sell Mobile Screen:

Imei Field:

No	Input	Process	Message	Reason
1	ABCD	Is Number	Not Valid	Enter only Number
2	12345	Length <15 && >15	Not Valid	Number with min 15 and Max 15
3	123....15	True Query(reject)	Not Valid Imei	Not Found in Database
4	12345...15	True	Valid	Database Valid

Price Field:

No	Input	Process	Message	Reason
1	ABCD	Is Number	Not Valid	Enter only Number
2	0	Amount >=0	Not Valid	Price only Min 1
4	1500	True	Valid	Valid price

Purchase Mobile Screen:**Imei Field:**

No	Input	Process	Message	Reason
1	ABCD	Is Number	Not Valid	Enter only Number
2	12345	Length <15 && >15	Not Valid	Number with 15 min and 15 Max
3	123....15	True Query(reject)	Imei Already used	Found in Database
4	12345...15	True	Valid	Database Valid

Mobile Number Field:

No	Input	Process	Message	Reason
1	ABCD	Is Number	Not Valid	Enter only Number
2	12345	Length <10 && >10	Not Valid	Number with 10 min and 10 Max
4	12345...10	True	Valid	Database Valid

IMPLEMENTATION

Implementation refers to the entire effort associated with a new system. The implementation of a web application involves longer term issues after the system has been designed and installed. Implementation is a part of the design of a web application, and is an organizational change process. It is a part of the process that begins with the very first idea for a web application has been successfully integrated with the operations of the organization. We expect most of the implementation to be concerned with behavioral phenomena since people are expected to change their information processing activities.

The implementation is processed from review and reports from developer cover the following areas:

- Good working conditions.
- Useful for gathering information.
- Update details easily.
- Attractive layouts.
- Working for as per requirements.

BIBLIOGRAPHY

Websites used:

- www.swift.org
- www.google.com
- www.youtube.com
- chat.openai.com

Darshan Kalathiya