**Welcome to Refrigerator**

**Introduction to Software Engineering**

**Date: 1/24/2023**

**GROUP MEMBERS**

**HAFSA SHOAIB (21B-054-SE) SE-A**

**MAHREEN KHAN (21B-010-SE) SE-A**

**ALIZA KHAN (21B-205-SE) SE-A**

**COURSE INSTRUCTOR**

**Instructor Name: Sir Usman Waheed**

**Project manager: Amna Shahadat**

**Usman Institute of Technology University**

**Welcome to Refrigerator**

**MEMBERS**

**HAFSA SHOAIB (21B-054-SE) SE-A**

**MAHREEN KHAN (21B-010-SE) SE-A**

**ALIZA KHAN (21B-205-SE) SE-A**

# ABSTRACT / Executive Summary

The Fridge repairing system was fully manual, entries would be written on registers.

Fridge repairing is a shop where customers comes for the repairing of their fridge, because of this payment transaction continue day by day. The customer entries and payment was written on the transaction register and that created some difficulties. So we developed a software to solve these difficulties so now customer entries and payment transactions are done directly on the software. Additionally, technician and vender do automated tasks. And now the system is fully automated.

In future, there will be some updates to this software. It will be that every customer has their own accounts and the client will be able to view the issues with their refrigerator. This will greatly benefit for both the buyer and the system.

**TABLE OF CONTENTS**

Contents

[ABSTRACT / Executive Summary 1](#_Toc125404303)

[LIST OF FIGURES 3](#_Toc125404304)

[LIST OF TABLES 5](#_Toc125404305)

[Acknowledgments 5](#_Toc125404306)

[INTRODUCTION AND OVERVIEW 6](#_Toc125404307)

[METHODS, ASSUMPTIONS, and PROCEDURES 8](#_Toc125404308)

[AVAILABLE RELEVANT Solutions and Evaluation 10](#_Toc125404309)

[2. REFRIGERATOR REPAIR PROJECT 11](#_Toc125404310)

[Analysis AND DESIGN 11](#_Toc125404311)

[ACTOR USE CASE TABLE: 15](#_Toc125404312)

[REPORT DETAILED LIST 25](#_Toc125404313)

[CLASS DIAGRAM 62](#_Toc125404314)

[DATABASE DIAGRAM 63](#_Toc125404315)

[SUPPORT 63](#_Toc125404316)

[CONCLUSIONS AND SUGGESTIONS FOR FUTURE WORK 65](#_Toc125404317)

[REFERENCES 66](#_Toc125404318)

[APPENDICES SHOULD APPEAR AFTER THE LIST OF REFERENCES 66](#_Toc125404319)

[LIST OF SYMBOLS 69](#_Toc125404320)

[GLOSSARY 70](#_Toc125404321)

# LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| Figure No. | Description | Page No. |
| 1.1 Swimming lane diagram of Fridge repairing system | Swim lane diagrams are flowcharts that show both a process from start to finish and who is responsible for each step in the process. This diagram shows the whole process of Fridge repairing system project step by step. |  |
| 1.2  Context diagram | The Context Diagram shows the system under consideration as a single high-level process and then shows the relationship that the system has with other external entities. |  |
| 1.3  Object diagram | An object diagram is a graph of instances, including objects and data values. The purpose is to capture the static view of a system at a particular moment. |  |
| 1.4  Actor use case | The users that interact with a system. An actor can be a person, an organization, or an outside system that interacts with your application or system. They must be external objects that produce or consume data. |  |
| 1.5  Object diagram of each use case | An object diagram is a graph of instances, including objects and data values. The purpose is to capture the static view of a system at a particular moment. |  |
| 1.6 Sequence diagram | A sequence diagram is a Unified Modeling Language (UML) diagram that illustrates the sequence of messages between objects in an interaction |  |
| 1.7  Class diagram of each use case | Class diagrams are the blueprints of your system or subsystem. You can use class diagrams to model the objects that make up the system, to display the relationships between the objects, and to describe what those objects do and the services that they provide. |  |
| 1.8  Class diagram | Class diagrams are the blueprints of your system or subsystem. You can use class diagrams to model the objects that make up the system, to display the relationships between the objects, and to describe what those objects do and the services that they provide. |  |
| 1.9 Database diagram | Database diagramsgraphically show the structure of the database and relations between database objects*.* |  |
| 2.0 Deployment diagram | A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical components in a system. Deployment diagrams are used to visualize the topology of the physical components of a system. |  |

# LIST OF TABLES

|  |  |  |
| --- | --- | --- |
| Table No. | Description | Page No. |
| 1.1  Actor use case table | Actor use case table contains primary actor, business actor, participating actor etc. |  |
| 1.2  Report list table | The report list table mention detailed, summary and exception report against each use case. |  |
| 1.3  Analysis use case | It describes the way it is going at shop, bakery etc. |  |
| 1.4  Design use case | It describes the way it is going on our website. |  |

# Acknowledgments

Firstly, we are very thankful to Mr. Israr the owner of Welcome to Refrigerator. Who gave us permission to visit his shop related to our project (fridge repairing system). Sir you guided our team so well that we were able to complete the project without facing any problem.

Secondly we proudly say that we are UITians. We especially want to thank our institute, our director and faculty members. Quality education leads to the personality grooming through a balanced blend of mental, physical and moral training. And here, at UIT we as a student enjoys cooperative faculty, excellent facilities and congenial environment, which is second to none. Thank you UIT University.

Thirdly we want to thank our course instructor Sir Usman Waheed. Our gratitude to you for all you have done, which we will never forget. We truly appreciate you and your time you spent helping us in many occasions. Thank you very much for the course you teaches us. We enjoyed every minute of your lecture as well as your marvelous sense of humor. And also thank to our project advisor Amna Shahadat. She guided us throughout the project very well.

At the last, we are very thankful to our parents. Saying "thank you" is not enough for us to express how grateful we are for your support over the years. You are our mentor, and everything we have achieved today is because of you. May Allah grant you a long and happy life (Ameen).

**Welcome to Refrigerator**

**MEMBERS**

**HAFSA SHOAIB (21B-054-SE) SE-A**

**MAHREEN KHAN (21B-010-SE) SE-A**

**ALIZA KHAN (21B-205-SE) SE-A**

# INTRODUCTION AND OVERVIEW

Introduction to Welcome to Refrigerator:

Welcome to refrigerator is a shop that is responsible to repair refrigerators. They give professional repair services at home. For repairing if the parts of fridge are not available in their inventory they purchase parts from other vendors.

The problem was manual recording of payments received in journals, which is prone to human errors and is very tedious to trace errors and calculate, and it is not economical to in this modern age.

Overview of the project:

The application is in the form of “Website”. In the application, User first creates accounts for customers, shopkeeper and technician, then the user can create pages of provide details, problems in refrigerator, place order, vendor payment, and generate bill.

* Provide Detail :

The customer will enter his/her information along with the information of his/her refrigerator and all the data will be saved in the database.

* Problems in refrigerator :

The technician will detect the problem and tell the price according to it. The customer after reviewing the price will either accept the offer or reject it.

* Place order :

If the parts required are not available in the inventory then shopkeeper will place an order to vendor.

* Vendor Payment:

The vendor will generate the bill of the required part and the shopkeeper will pay the amount either through cash or card and this transaction also save in a database.

* Generate Bill:

The shopkeeper will generate the bill after technician repaired the refrigerator and the customer will receive the fridge and will pay the bill and this transaction also save in database.

**Background**

There is no existing system in WTR shop. The staff works manually. Shopkeeper met the customer and then technician attain the complaint of the customer. For the customer ease one of the website Welcome to refrigerator offers the customer to complain online and then they send the technician in customer house but there prices are too high that’s why they fail to satisfy the customer.

**Aim and Statement of Problem**

The Aim of the project is to solve organization problem by using problem solving approach.

Performance:

Sometimes when customers come to the shop, the shop is not open.

We solved the customer problem by creating a website. Customer now anytime contact the shopkeeper through the website.

Information:

• Input:

• Customer online request the shopkeeper for testing, then the customer is requested to input his details such as his name address etc.

• Output:

•The technician comes to the customer house for testing the fridge then the technician tells the problems amount to customer. According to price customer accept or reject, if the customer accept then the shopkeeper check his inventory, if repairing parts available in the inventory then the shopkeeper gives parts to technician for repairing the fridge and if the parts not available in the in the inventory then shopkeeper place the order to the vendor. If customer provide incomplete information prompt a message to the customer to provide complete information.

• Stored Data:

• Details of customer

•Details of fridge

• Order details

First customer have to come in the shop for requesting the shopkeeper for testing the fridge but now customer have website for requesting the shopkeeper for testing. We were face a lot of difficulties in the project. We met with the shopkeeper 2 times for customer data collection and facing the difficulty in coding because of the lack of the time.

Economic:

• This project is economically balanced as it will only require our time and effort in creating it and furthermore, we are not using any sort of hardware so our only resource is time.

Control:

• Customer can accept or reject the problems amount.

• Application will run by authorized person i.e., owner

• Data of customer must not be share with others.

Efficiency:

• Using of good data base

• Data of each customer will be stored in a database for easier access

Service:

• Customer can see the total amount of the bill.

• Software should be easy to understand for type of ages

# METHODS, ASSUMPTIONS, and PROCEDURES

Structured Analysis and Structured Design (SA/SD) is a diagrammatic notation that is designed to help people understand the system. The basic goal of SA/SD is to improve quality and reduce the risk of system failure. It establishes concrete management specifications and documentation.

STRENGTH OF STRUCTURE ANALYSIS:

* Lower Costs:

TQM lowers costs throughout the business infrastructure and organization. Because it is an all-encompassing quality management program, TQM helps different departments to communicate their needs, problems, and desires with each other, so that workable solutions can be found that will help the organization cut costs throughout the supply chain, distribution chain, shipping and receiving, accounting and management departments without losing productivity or the ability to operate rapidly in the face of change.

The concept of Total Quality Management is rooted in the idea of providing all of the tools, training, and experience necessary to measure the entire quality control of an organization.

Because it is an all-encompassing quality management program, TQM helps different departments to communicate their needs, problems, and desires with each other, so that workable solutions can be found that will help the organization cut costs throughout the supply chain, distribution chain, shipping and receiving, accounting and management departments without losing productivity or the ability to operate rapidly in the face of change.

* Improved Reputation:

TQM programs have the advantage of improving corporate as well as product reputations in the marketplace, because errors and defective products are discovered much more rapidly than under a non-TQM system, and often before they are ever sent to market or found in the hands of the public.

Weaknesses of Structure Analysis:

* Resistance to Change:

Workers may feel that their jobs or occupations within the company are at risk under a comprehensive TQM program, and as a result, they may be slow or resistant to making the necessary changes for the TQM program to work properly. In addition, skilled workers may be lost as they decide to leave because of their unease at the direction that things are headed within the company, or they may not implement things properly, causing increased costs.

* High Cost of Time:

The high cost of implementing a TQM program, and the fact that it may take several years for the program to be fully implemented before results and benefits are seen, can be a huge disadvantage to a TQM program, especially in today's uncertain economic conditions. TQM should be considered a long-term investment.

* Information Engineering:

Information engineering is a family of data-oriented analysis and techniques used to design, develop, and maintain information systems which support strategic missions, decision processes, and daily operations of a company. It is often regarded as a data-oriented methodology rather than a process-oriented methodology.

Advantages

The primary advantage of the IE methodology is that data are identified first, then the functions are identified second. The IE methodology does not foster the complete decomposition of the inputs, processes, and outputs. In fact it can be argued that IE does not hierarchically decompose the functions in the same way that the traditional methodologies do; this can be seen as an advantage not to have to train users to decompose the functionality of the system.

Disadvantages

The disadvantages of the IE methodology include the fact that users must be trained to understand the models and that users must be able to identify the data of the system first before identifying the functions of the system.

OBJECT ORIENTED:

Object-oriented (O-O) analysis and design is an approach that is intended to facilitate the development of systems that must change rapidly in response to dynamic business environments. Each object is a computer representation of some actual thing or event. Objects may be customers, items, orders, and so on.

Advantages

Focuses on data rather than the procedures as in Structured Analysis. The principles of encapsulation and data hiding help the developer to develop systems that cannot be tampered by other parts of the system. The principles of encapsulation and data hiding help the developer to develop systems that cannot be tampered by other parts of the system. It allows effective management of software complexity by the virtue of modularity.

Disadvantages

Functionality is restricted within objects. This may pose a problem for systems which are intrinsically procedural or computational in nature. It cannot identify which objects would generate an optimal system design. The object-oriented models do not easily show the communications between the objects in the system. All the interfaces between the objects cannot be represented in a single diagram.

# AVAILABLE RELEVANT Solutions and Evaluation

1. PROJECT REPAIR AND FABRICATION OF A REFRIGERATOR SYSTEM

[**https://www.grossarchive.com/project/18598/project-repair-and-fabrication-of-a-refrigerator-system**](https://www.grossarchive.com/project/18598/project-repair-and-fabrication-of-a-refrigerator-system)

Introduction

Refrigeration is a branch of science that deals with the process of removing heat from a substance or space in order to make it cooler. Refrigerator is defined as a device that is used in cooling the internal temperature below the room temperature (that is between 250C to 300C). Generally, however, for a space or substance to be cooler, it must loss that heat to another.  Also, for a space or substance to get hotter, it must absorb heat from another, which must be at higher temperature.  For both process to occur, heat must be absorbed or lost, thus heat is the characteristics agent of heating and cooling, consequently, for heat flow there is absorbs at a lower temperature region and rejected at a higher temperature region, which is the quantity being determined by the temperature gradient of the two regions. The modes of heat transmission are conduction concretion and radiation of which conduct convection and radiation, of which conduction and convection are extensively involved, in domestic refrigeration.  In refrigeration process, there is always a body employed as the heat absorber or indirect contact with the space or substance being cooled depending on the required final effect. Such cooling agents is known as refrigerant, which is known as the refrigerant, which is circulated around the evaporator that id high temperature region) and condensing region (that is higher temperature) in order to maintain a constant refrigeration process. It does it work be evaporating (when it absorbs heat up to the boiling point temperature) and by condensing when it losses the absorbed heat to return to its original liquid state, in the system. These heat when absorbed, may be classified as sensible heat or latent heat depending on its physical effect on the refrigerants.

# REFRIGERATOR REPAIR PROJECT

<https://circuitcellar.com/research-design-hub/projects/refrigerator-repair-project/>

The control PCB is basically a relay that enables power to the fridge thermostat or the defrost heater. The defrost heater is used periodically to warm the condenser coils. keeping them free of ice build-up. The defrost thermostat opens, removing power from the heater once the temperature of the condenser has sufficiently increased. Whenever the defroster is ON, the thermostat is OFF and so is the compressor—even if the thermostat should call for a lower temperature and attempt to turn on the compressor. I was a bit surprised to find that the compressor’s job was to remove heat from the freezer only. It is the door between the fridge and freezer that cools the fridge by circulating colder air from the freezer.

# Analysis AND DESIGN

For our project first we make the visit to the fridge repairing shop then we capture requirements

from their manager and instantly make a swim lane diagram which shows how everything work at the shop.

From that swim lane we make a context diagram and after that actor use case diagram which shows what each actor do. After that we write the analysis use case narration against each use case which describes the complete flow. And then we make an object diagram of each use case an after that complete object diagram, according to transaction pattern which contains participant, transaction, transaction line item, item, place etc.

And after that design use case narration which describe the complete flow of our website. And then we make class diagram by extracting verbs and noun from design use case. From design use case we make sequence diagram. And after that component, deployment and database diagram.

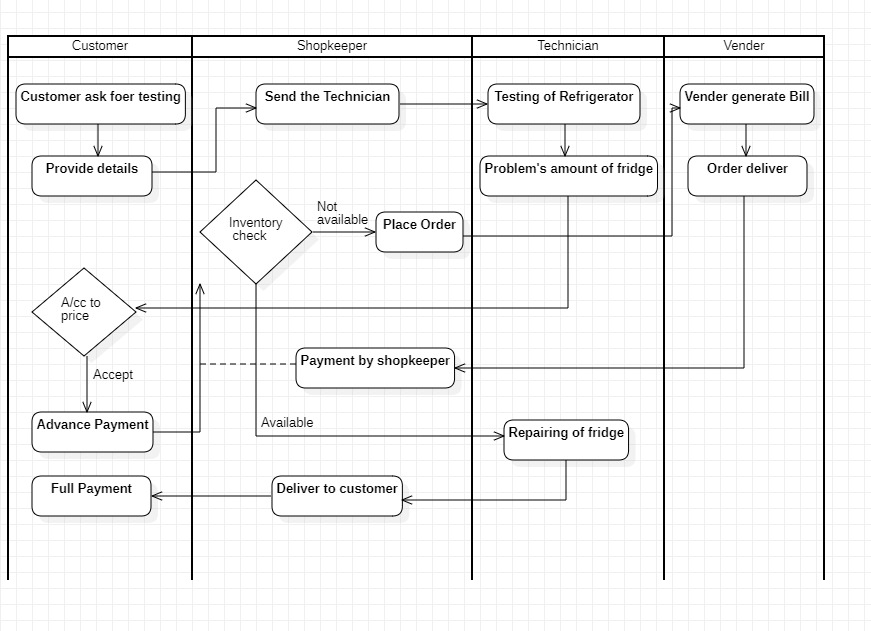


Figure 1.1 Swim Lane Diagram

In the above figure first customer ask for testing and then provide his/her details to shopkeeper. Then shopkeeper send the technician to customer house and the technician test the refrigerator and according to problem’s technician tells the problem’s amount of fridge, and then according to price customer accept or reject. If customer reject so the procedure is cancelled and if customer accept so give advance payment. Then shopkeeper check his inventory that the parts required for repairing is available or not. If available, then shopkeeper give parts to technician and technician repair the fridge and deliver fridge to shopkeeper. And then shopkeeper give fridge to customer and customer give advance payment. If not available, then shopkeeper place order to vender and then vender generate bill and deliver order, against delivery shopkeeper pay the bill. Now the parts are available in our inventory.

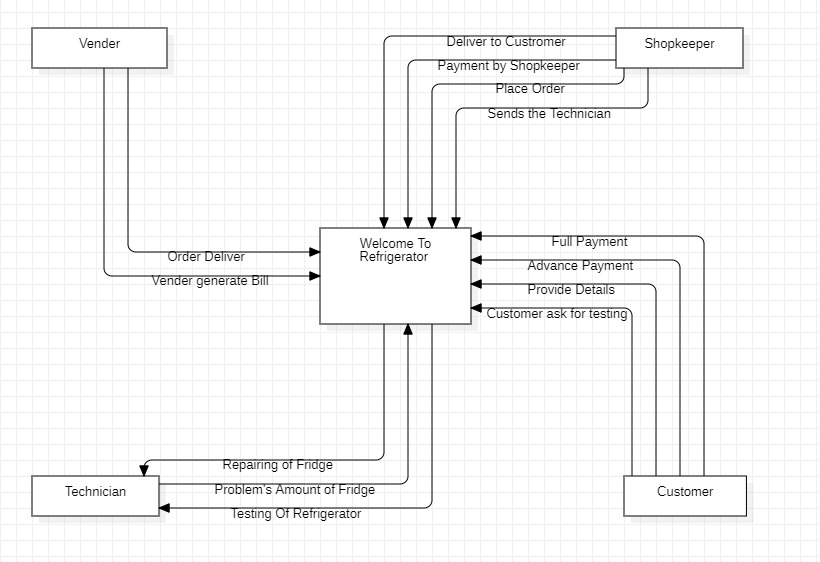


Figure 1.2 Context Diagram

In the above figure first customer ask for testing and then provide his/her details to shopkeeper. Then shopkeeper send the technician to customer house and the technician test the refrigerator and according to problem’s technician tells the problem’s amount of fridge, and then according to price customer accept or reject. If customer reject so the procedure is cancelled and if customer accept so give advance payment. Then shopkeeper check his inventory that the parts required for repairing is available or not. If available, then shopkeeper give parts to technician and technician repair the fridge and deliver fridge to shopkeeper. And then shopkeeper give fridge to customer and customer give advance payment. If not available, then shopkeeper place order to vender and then vender generate bill and deliver order, against delivery shopkeeper pay the bill. Now the parts are available in our inventory.

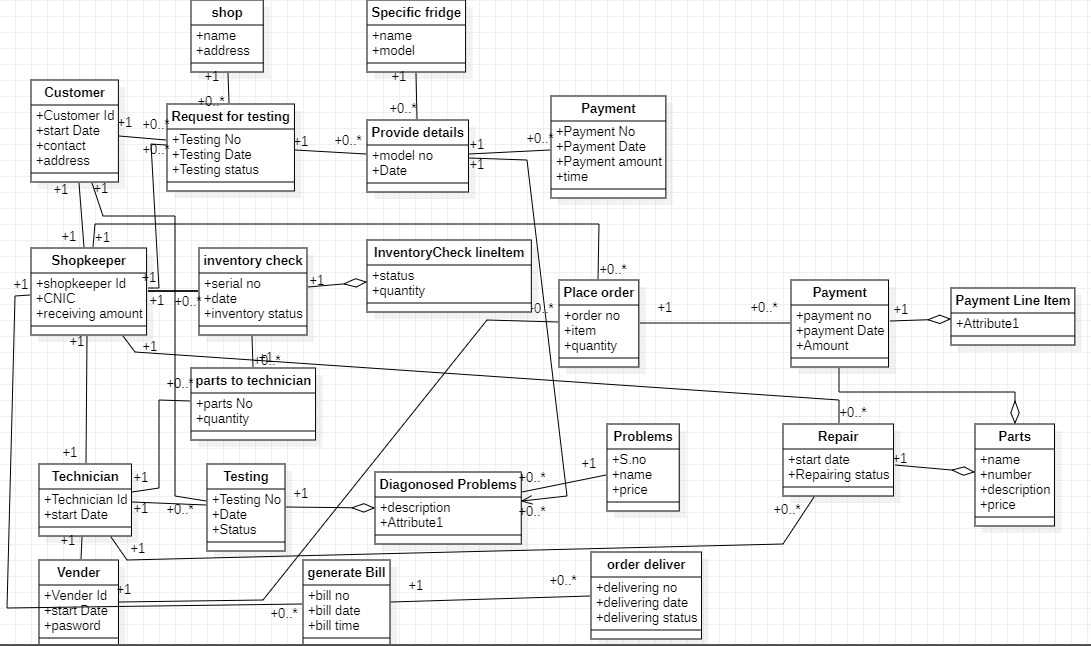


Figure 1.3 Object Diagram(complete)

In the above figure Customer, Shopkeeper, Technician, Vender are the participants. Transaction is request for testing whose subsequent transactions are provide details and payment with the specific item which is specific fridge and place which is shop. Second transaction is Testing whose other associate is Diagnosed problems and problems. Third transaction is Inventory check whose line item is inventory check line item and subsequent transaction is parts to technician. Fourth transaction is repair whose other associate is parts. Fifth transaction is place order whose subsequent transaction is payment and line item is payment line item. Sixth transaction is generate bill whose subsequent transaction is order deliver.

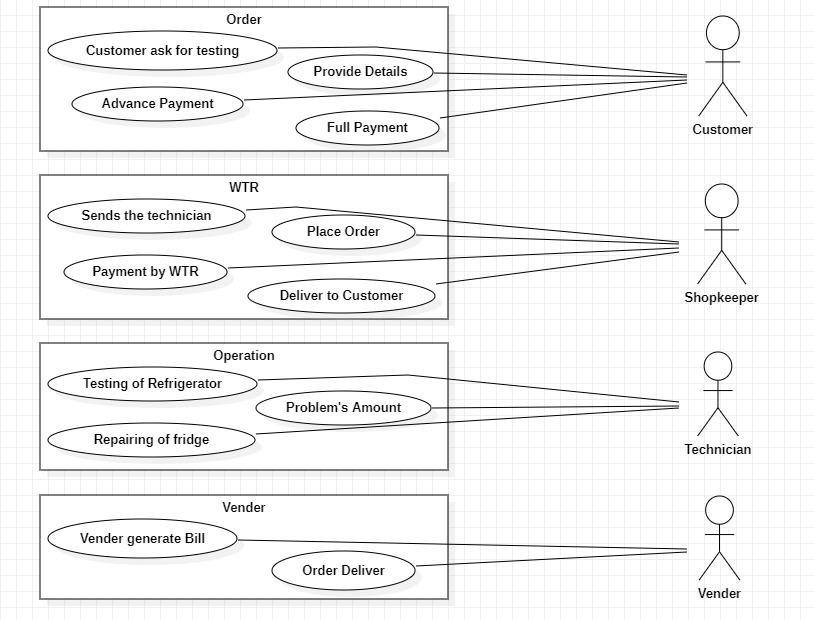


Figure 1.4 Actor Use case Diagram

The above figure represents the initiation of actors, customer ask for testing, provide detail, advance payment and full payment will be initiate by customer. Send the technician, place order, payment by shopkeeper and deliver to customer will be initiate by shopkeeper. Testing of refrigerator, problem’s amount and repairing of fridge will be initiate by technician. Vender generate bill and order deliver will be initiate by vender.

# ACTOR USE CASE TABLE:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Usecase | Primary Actor | System Actor | Other Participating Actor | Other Interested Stakeholder |
| Customer ask for testing | Customer | Customer | Shopkeeper | - |

Table 1.1 (a) Actor Use case Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Usecase | Primary Actor | System Actor | Other Participating Actor | Other Interested Stakeholder |
| Provide details | Customer | Customer | Shopkeeper | - |

Table 1.1 (b) Actor Use case Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Usecase | Primary Actor | System Actor | Other Participating Actor | Other Interested Stakeholder |
| Sends the technician | Shopkeeper | Shopkeeper | Customer, Technician | - |

Table 1.1 (c) Actor Use case Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Usecase | Primary Actor | System Actor | Other Participating Actor | Other Interested Stakeholder |
| Testing of refrigerator | Technician | Technician | Customer | - |

Table 1.1 (d) Actor Use case Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Usecase | Primary Actor | System Actor | Other Participating Actor | Other Interested Stakeholder |
| Problem’s amount | Technician | Technician | Customer | - |

Table 1.1 (e) Actor Use case Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Usecase | Primary Actor | System Actor | Other Participating Actor | Other Interested Stakeholder |
| Advance Payment | Customer | Customer | Shopkeeper | - |

Table 1.1 (f) Actor Use case Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Usecase | Primary Actor | System Actor | Other Participating Actor | Other Interested Stakeholder |
| Inventory check | Shopkeeper | Shopkeeper | - | - |

Table 1.1 (g) Actor Use case Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Usecase | Primary Actor | System Actor | Other Participating Actor | Other Interested Stakeholder |
| Place Order | Shopkeeper | Shopkeeper | Vender | - |

Table 1.1 (h) Actor Use case Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Usecase | Primary Actor | System Actor | Other Participating Actor | Other Interested Stakeholder |
| Generates Bill | Vender | Vender | Shopkeeper | - |

Table 1.1 (i) Actor Use case Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Usecase | Primary Actor | System Actor | Other Participating Actor | Other Interested Stakeholder |
| Order deliver | Vender | Vender | Shopkeeper | - |

Table 1.1 (j) Actor Use case Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Usecase | Primary Actor | System Actor | Other Participating Actor | Other Interested Stakeholder |
| Payment by Shopkeeper | Shopkeeper | Shopkeeper | Vender | - |

Table 1.1 (k) Actor Use case Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Usecase | Primary Actor | System Actor | Other Participating Actor | Other Interested Stakeholder |
| Repairing of fridge | Technician | Technician | Shopkeeper | - |

Table 1.1 (l) Actor Use case Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Usecase | Primary Actor | System Actor | Other Participating Actor | Other Interested Stakeholder |
| Deliver to customer | Shopkeeper | Shopkeeper | Customer | - |

Table 1.1 (m) Actor Use case Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Usecase | Primary Actor | System Actor | Other Participating Actor | Other Interested Stakeholder |
| Full payment | Customer | Customer | Shopkeeper | - |

Table 1.1 (n) Actor Use case Table

**REPORT LIST TABLE:**

|  |  |
| --- | --- |
| USECASE | Reports |
| Customer ask for Testing | **Detailed**  List of customers  List of customers ask for testing  **Summary**  Testing summary  Testing invoice  Customer wise testing  Customer Detail  **Exception**  Customer not ask for testing, many customers ask for testing |

Table 1.2 (a) Report List Table

|  |  |
| --- | --- |
| USECASE | Reports |
| Provide Details | **Detailed**  List of customers  Customer details and fridge details  **Summary**  Detailed summary  **Exception**  Detail have not saved |

Table 1.2 (b) Report List Table

|  |  |
| --- | --- |
| USECASE | Reports |
| Sends the Technician | **Detailed**  List of customer  Details about customer  Details about fridge  **Summary**  Detailed summary  Customer wise testing  Technician Details  **Exception**  Technician can’t examine problem |

Table 1.2 (c) Report List Table

|  |  |
| --- | --- |
| USECASE | Reports |
| Testing of Refrigerator | **Detailed**  List of customer  List of fridge for testing  Fridge details  **Summary**  Testing summary  Date of testing  Customer wise testing  Customer and fridge Details  **Exception**  Technician can’t repair fridge |

Table 1.2 (d) Report List Table

|  |  |
| --- | --- |
| USECASE | Reports |
| Problem’s Amount | **Detailed**  List of customer  List of problems  List of problem’s amount  **Summary**  Date of testing  Problem details  Problem Amount details  Customer and fridge details  **Exception**  Problem’s amount are too high  Customer can’t afford |

Table 1.2 (e) Report List Table

|  |  |
| --- | --- |
| USECASE | Reports |
| Advance Payment | **Detailed**  Detail of customer  List of problems  Amount to be paid  **Summary**  Date of payment  Advance Payment to be paid in cash to shopkeeper  Testing summary  Customer wise testing  **Exception**  Technician did not come for testing  Advance payment was delayed  Bill was generate with overwriting |

Table 1.2 (f) Report List Table

|  |  |
| --- | --- |
| USECASE | Reports |
| Inventory check | **Detailed**  Detail of fridge  List of problem  Detail about inventory  Detail about shop  **Summary**  Order summary  Order invoice  Fridge details  **Exception**  Parts are not available in market |

Table 1.2 (g) Report List Table

|  |  |
| --- | --- |
| USECASE | REPORT |
| Place Order | **Detailed**  List of products  Detail of Shopkeeper  List of Orders  **Summary**  Order summary  Order details  **Exception**  Product not ordered  Shopkeeper haven’t ordered any thing  Order not delivered on time.  Full payment was not given. |

Table 1.2 (h) Report List Table

|  |  |
| --- | --- |
| USECASE | REPORT |
| Generate Bill | **Detailed**  Details about Shopkeeper  List of products  Details about company  **Summary**  Order summary  Order invoice  Customer wise orders  Product wise orders  Monthly orders (total numbers, total cost, average order cost,etc.)  Order detail  **Exception**  Product not ordered  Customer haven’t ordered anything  Enough Stock was not available |

Table 1.2 (i) Report List Table

|  |  |
| --- | --- |
| USECASE | REPORT |
| Order deliver to shopkeeper | **Detailed**  List of products  Details about customer  Details about vendor  Details about company  **Summary**  Order summary  Order invoice  Customer wise orders  Product wise orders  Monthly orders (total numbers, total cost, average order cost,etc.)  Order detail  **Exception**  Product not ordered  customer haven’t ordered any thing  Order not delivered on time.  Enough Stock was not available |

Table 1.2 (j) Report List Table

|  |  |
| --- | --- |
| USECASE | REPORT |
| Payment by shopkeeper | **Detailed**  Details about company  Details about vendor  Amount to be paid.  **Summary**  Date of payment  Order summary  Order invoice    **Exception**  Vendor did not deliver the order  Payment was delayed  Bill was generated with overwriting. |

Table 1.2 (k) Report List Table

|  |  |
| --- | --- |
| USECASE | REPORT |
| Repairing of fridge | **Detailed**  Details about customer  Details about problems  **Summary**  Problem summary  Customer wise repairing  **Exception**  Late delivery  Parts for repairing not available |

Table 1.2 (l) Report List Table

|  |  |
| --- | --- |
| USECASE | REPORT |
| Deliver to customer | **Detailed**  Details about customer  Details about shop  Details about fridge  **Summary**  Customer wise orders  fridge detail  **Exception**  Late delivery |

Table 1.2 (m) Report List Table

|  |  |
| --- | --- |
| USECASE | REPORT |
| Full payment | **Detailed**  Details about customer  Details about shop  Amount to be paid.  **Summary**  Date of payment  Payment details    **Exception**  Shopkeeper did not deliver the fridge on time  Payment was delayed  Bill was generated with overwriting. |

Table 1.2 (n) Report List Table

# REPORT DETAILED LIST

Customer ask for testing

|  |  |  |
| --- | --- | --- |
| Use Case Name | Customer ask for testing | Business requirement: System analysis: System design: |
| Use Case ID | -- |
| Priority | High |
| Primary Business Actor: | Customer | |
| Primary system Actor: | Customer | |
| Other Participating Actors: | Shopkeeper | |
| Descriptions | This use case describes the event of a customer ask for testing of his/her refrigerator. | |
| Precondition: | There must be a technical problem in refrigerator. | |
| Trigger: | This use case initiated when the customer request for testing. | |
| Typical Course of Events: | Actor Action | System Response |
| Step 1: Customer enters the shop.  Step 2: Customer request for testing. | Step 3: Shopkeeper accept the testing. |
| Alternate Course | -- | |
| Conclusion: | The use case concludes when the shopkeeper accept the testing. | |
| Postcondition: | The customer response has been recorded and customer provide information to shopkeeper. | |

Table 1.3 (a) Analysis Use case Table

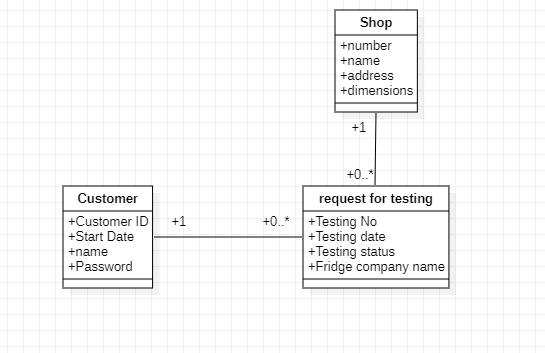
****

Figure 1.5 (a) Object Diagram

**Provide details**

|  |  |  |
| --- | --- | --- |
| Use Case Name | Provide details | Business requirement: System analysis: System design: |
| Use Case ID | -- |
| Priority | High |
| Primary Business Actor: | Customer | |
| Primary system Actor: | Customer | |
| Other Participating Actors: | Shopkeeper | |
| Descriptions | This use case describes the event in which customer is providing his/her details.. | |
| Precondition: | Customer availability. | |
| Trigger: | This use case initiated when the customer request is accepted. | |
| Typical Course of Events: | Actor Action | System Response |
| Step 1: Customer provide his/her details like: name, address, phone number and also fridge details (name, model no) etc. | Step 2: Shopkeeper saving the customer details. |
| Alternate Course | -- | |
| Conclusion: | The use case concludes when the shopkeeper receives the customer information. | |
| Postcondition: | The customer information has been recorded and shopkeeper sends the technician. | |

Table 1.3 (b) Analysis Use case Table

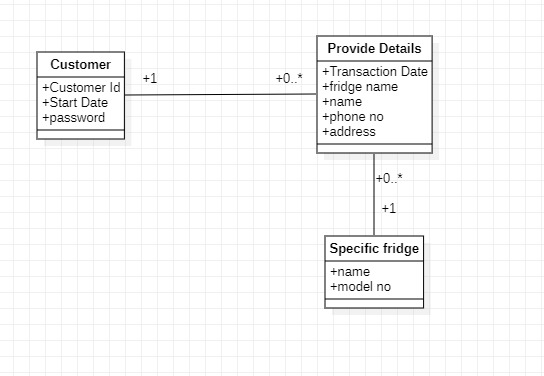


Figure 1.5 (b) Object Diagram

|  |  |  |
| --- | --- | --- |
| Use Case Name | Provide details | Business requirement: System analysis: System design: |
| Use Case ID | -- |
| Priority | High |
| Primary Business Actor: | Customer | |
| Primary system Actor: | Customer | |
| Other Participating Actors: | Shopkeeper | |
| Descriptions | This use case describes the event in which customer is providing his/her details. | |
| Precondition: | Customer must click on the provide detail button. | |
| Trigger: | This use case initiated when the customer request is accepted. | |
| Typical Course of Events: | Actor Action | System Response |
| Step 1: Customer open the website.  Step 2: Customer click on the customer’s interface button.  Step 4: Customer enter the details and click on the save button. | Step 3: System responds by displaying a window “w1- Provide details” to enter the customer and fridge information such as: Id, name, email, contact, address, gender, fridge company, fridge model, date, problems details.  Step 5: Message is displayed that your information has been saved. |
| Alternate Course | -- | |
| Conclusion: | The use case concludes when the shopkeeper receives the customer information. | |
| Postcondition: | The customer information has been recorded and shopkeeper sends the technician. | |

Table 1.4 (b) Design Use case Table

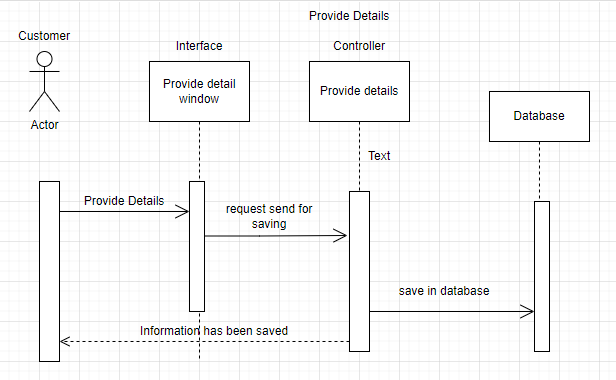
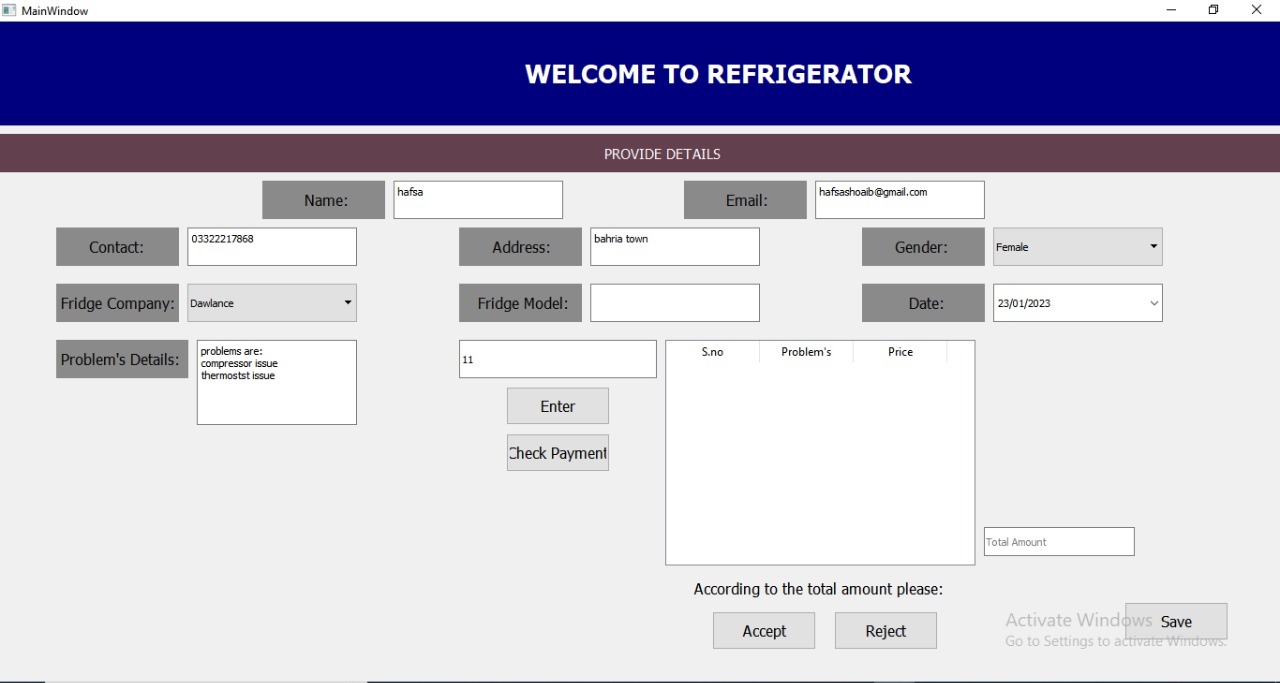


Figure 1.6 (b) Sequence Diagram



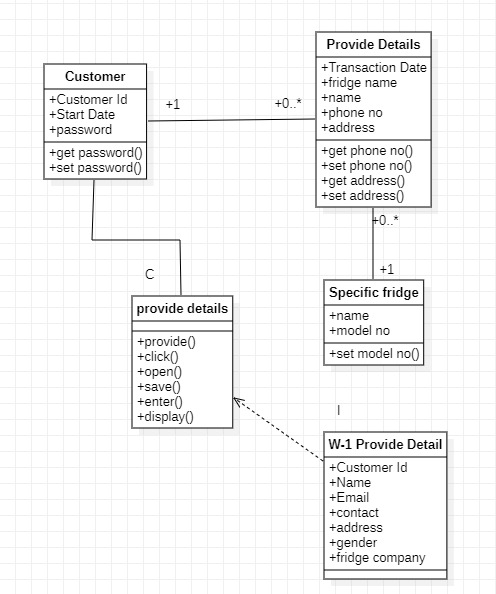


Figure 1.7 (b) Class Diagram

**Sends the technician**

|  |  |  |
| --- | --- | --- |
| Use Case Name | Sends the technician | Business requirement: System analysis: System design: |
| Use Case ID | MST-001 |
| Priority | High |
| Primary Business Actor: | Shopkeeper | |
| Primary system Actor: | Shopkeeper | |
| Other Participating Actors: | Customer, Technician | |
| Descriptions | This use case describes the event when the shopkeeper sends the technician. | |
| Precondition: | Technician availability at the shop. | |
| Trigger: | This use case initiated when the customer information is received. | |
| Typical Course of Events: | Actor Action | System Response |
| Step 1: Shopkeeper sends the technician. | Step 2: Technician go to customer house. |
| Alternate Course | -- | |
| Conclusion: | The use case concludes when the technician go to customer house. | |
| Postcondition: | Technician reach to customer house and test the refrigerator. | |

Table 1.3 (c) Analysis Use case Table

**Testing of refrigerator**

|  |  |  |
| --- | --- | --- |
| Use Case Name | Testing of refrigerator | Business requirement: System analysis: System design: |
| Use Case ID | MST-002 |
| Priority | High |
| Primary Business Actor: | Technician | |
| Primary system Actor: | Technician | |
| Other Participating Actors: | Customer | |
| Descriptions | This use case describes the event of a technician testing the refrigerator. | |
| Precondition: | Technician must have idea about his work. | |
| Trigger: | This use case initiated when the technician reach the customer house. | |
| Typical Course of Events: | Actor Action | System Response |
| Step 1: Technician test the refrigerator.    Step 2: Technician tells the list of problems of fridge to the customer like: compressor problem, water leakage, Thermostat, gas leakage etc. | Step 3: Customer asks for the amount. |
| Alternate Course | Step 2: If the problem is Compressor problem then the price is 5000 & if the problem is water leakage then the price is 2000 & if the problem is Thermostat then the price is 3000 & if the problem is Gas leakage then the price is 2000. | |
| Conclusion: | The use case concludes when the customer ask for problem’s amount. | |
| Postcondition: | Technician tells the problem’s amount of fridge. | |

Table 1.3 (d) Analysis Use case Table

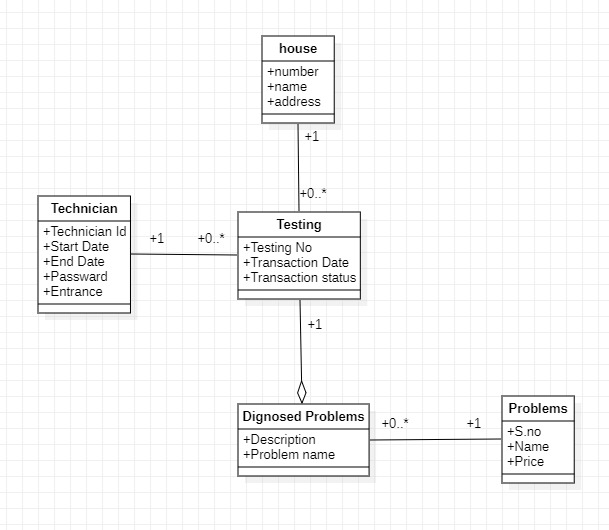


Figure 1.5 (d) Object Diagram

|  |  |  |
| --- | --- | --- |
| Use Case Name | Testing of refrigerator | Business requirement: System analysis: System design: |
| Use Case ID | MST-002 |
| Priority | High |
| Primary Business Actor: | Technician | |
| Primary system Actor: | Technician | |
| Other Participating Actors: | Customer | |
| Descriptions | This use case describes the event of a technician testing the refrigerator. | |
| Precondition: | Technician must have idea about his work. | |
| Trigger: | This use case initiated when the technician reach the customer house. | |
| Typical Course of Events: | Actor Action | System Response |
| Step 1: Testing is done.    Step 3: Technician select the problem’s from the list which are in the fridge which display in separate grid. | Step 2: System responds by displaying a window “w2- Problem’s list” which contains  compressor problem, water leakage, Thermostat, gas leakage etc. and their description and prices.  Step 4: Another window is displayed “w3-Total amount” which shows the total amount for repairing and a/cc to that customer accept or reject. |
| Alternate Course | Step 2: If the problem is Compressor problem then the price is 5000 & if the problem is water leakage then the price is 2000 & if the problem is Thermostat then the price is 3000 & if the problem is Gas leakage then the price is 2000. | |
| Conclusion: | The use case concludes when the customer ask for problem’s amount. | |
| Postcondition: | Technician tells the problem’s amount of fridge. | |

Table 1.4 (d) Design Use case Table

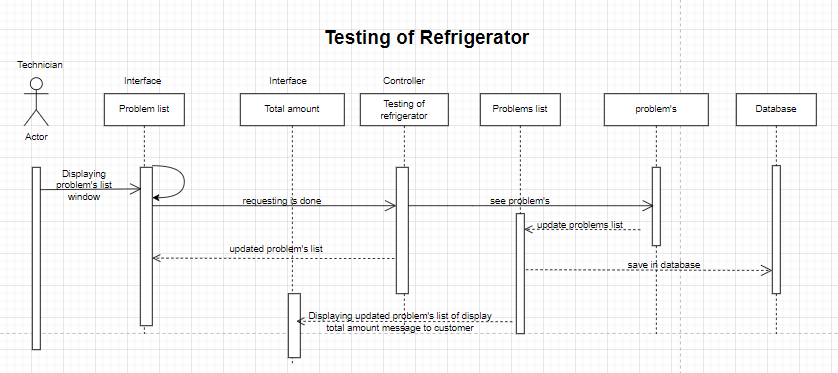
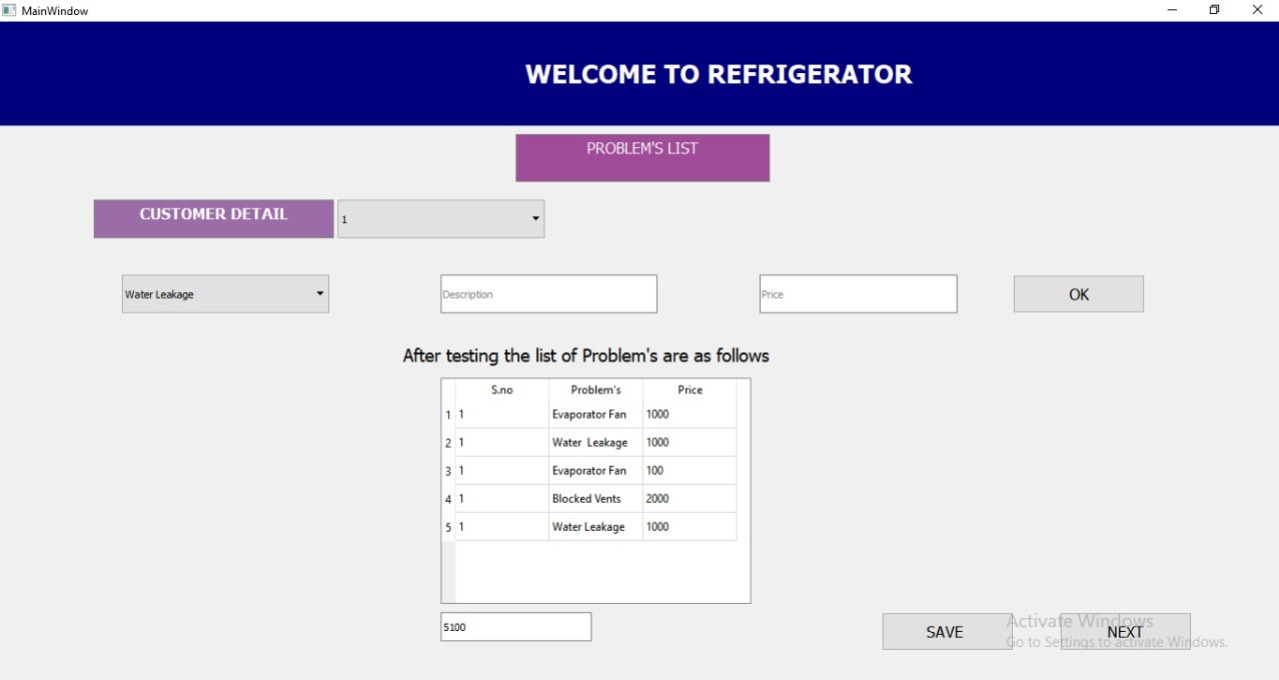


Figure 1.6 (d) Sequence Diagram



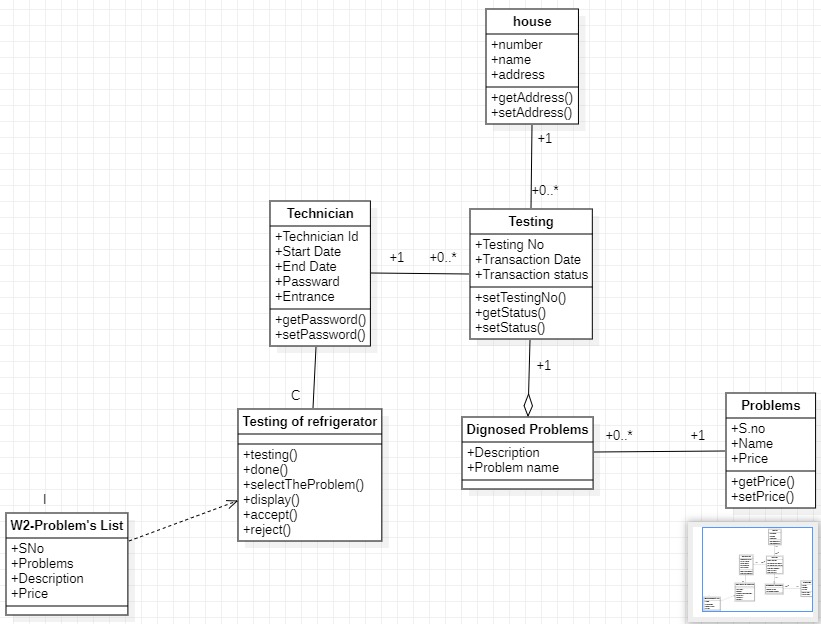
****

Figure 1.7 (d) Class Diagram

**Problem’s amount**

|  |  |  |
| --- | --- | --- |
| Use Case Name | Problem’s amount | Business requirement: System analysis: System design: |
| Use Case ID | MST-003 |
| Priority | High |
| Primary Business Actor: | Technician | |
| Primary system Actor: | Technician | |
| Other Participating Actors: | Customer | |
| Descriptions | This use case describes the event in which technician tells the problem’s amount. | |
| Precondition: | Technician should test the refrigerator first. | |
| Trigger: | This use case initiated when the customer asks for problem’s amount. | |
| Typical Course of Events: | Actor Action | System Response |
| Step 1: Technician tells the problem’s amount. | Step 2: Customer gives the advance payment. |
| Alternate Course | Step 2: If the customer does not accept the problem’s amount so the procedure is cancelled. | |
| Conclusion: | The use case concludes when the customer accept the problem’s amount. | |
| Postcondition: | Customer gives the advance payment. | |

Table 1.3 (e) Analysis Use case Table

**Advance payment**

|  |  |  |
| --- | --- | --- |
| Use Case Name | Advance payment | Business requirement: System analysis: System design: |
| Use Case ID | -- |  |
| Priority | High |
| Primary Business Actor: | Customer | |
| Primary system Actor: | Customer | |
| Other Participating Actors: | Shopkeeper | |
| Descriptions | This use case describes the event of an advance payment given by customer. | |
| Precondition: | Customer should have amount to pay. | |
| Trigger: | This use case initiated when the customer accept the problem’s amount. | |
| Typical Course of Events: | Actor Action | System Response |
| Step 1: Customer gives the advance payment. | Step 2: Shopkeeper accept the advance payment. |
| Alternate Course | -- | |
| Conclusion: | The use case concludes when the shopkeeper accept the advance payment. | |
| Postcondition: | After receiving advance payment, shopkeeper check his inventory. | |

Table 1.3 (f) Analysis Use case Table

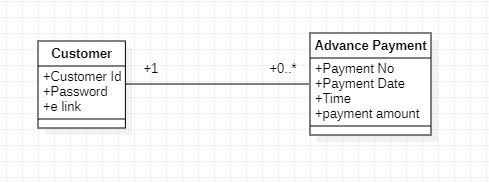
****

Figure 1.5 (f) Object Diagram

**Inventory check**

|  |  |  |
| --- | --- | --- |
| Use Case Name | Inventory check | Business requirement: System analysis: System design: |
| Use Case ID | MST-004 |
| Priority | High |
| Primary Business Actor: | Shopkeeper | |
| Primary system Actor: | Shopkeeper | |
| Other Participating Actors: | -- | |
| Descriptions | This use case describes the event of an inventory check of the shop by the shopkeeper. | |
| Precondition: | The shopkeeper must be available and aware of the problem. | |
| Trigger: | This use case initiated when the customer give the advance payment. | |
| Typical Course of Events: | Actor Action | System Response |
| Step 1: Shopkeeper check his inventory. | Step 2: If in inventory parts of fridge are available then shopkeeper give parts to technician. |
| Alternate Course | Step 2: If in inventory parts of fridge are not available then shopkeeper place order. | |
| Conclusion: | The use case concludes when the inventory check is done. | |
| Postcondition: | Shopkeeper place the order. | |

Table 1.3 (g) Analysis Use case Table

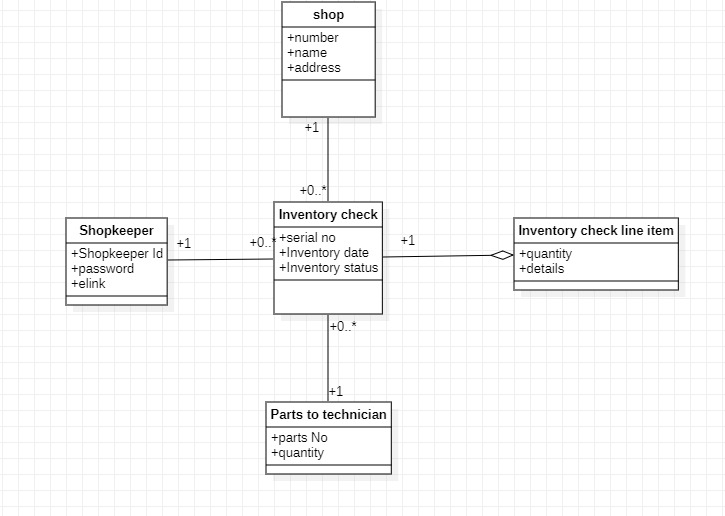


Figure 1.5 (g) Object Diagram

|  |  |  |
| --- | --- | --- |
| Use Case Name | Inventory check | Business requirement: System analysis: System design: |
| Use Case ID | MST-004 |
| Priority | High |
| Primary Business Actor: | Shopkeeper | |
| Primary system Actor: | Shopkeeper | |
| Other Participating Actors: | -- | |
| Descriptions | This use case describes the event of an inventory check of the shop by the shopkeeper. | |
| Precondition: | The shopkeeper must be available and aware of the problem. | |
| Trigger: | This use case initiated when the customer give the advance payment. | |
| Typical Course of Events: | Actor Action | System Response |
| Step 1: Shopkeeper check his inventory. | Step 2: System responds by displaying a window “w4-Inventory check” which shows items and status of items.    Step 3: If in inventory parts of fridge are available, another window is displayed “w5(a)-Items for repairing” which contains s.no, items and quantity. |
| Alternate Course | Step 2: If in inventory parts of fridge are not available, another window is displayed “w5(b)-Items to place order” which contains s.no, items and quantity, then shopkeeper place order. | |
| Conclusion: | The use case concludes when the inventory check is done. | |
| Postcondition: | Shopkeeper place the order. | |

Table 1.4 (g) Design Use case Table

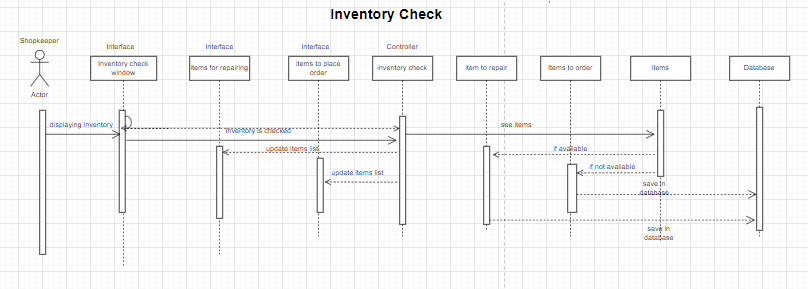
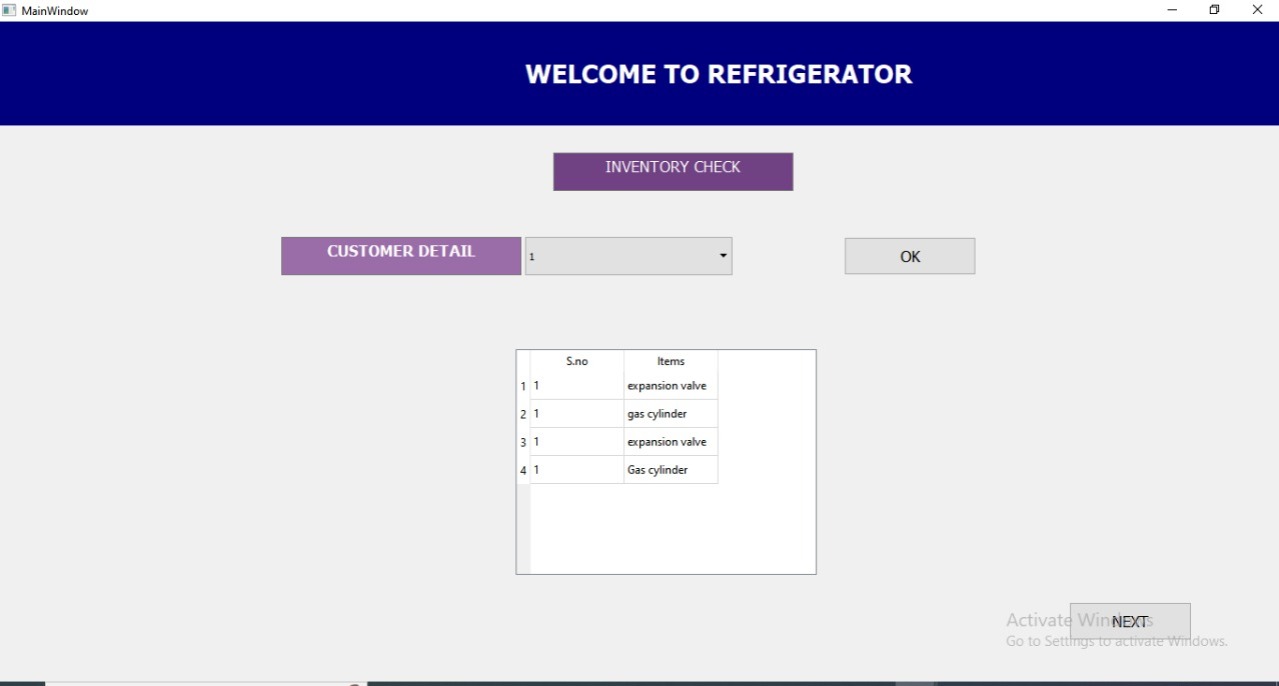


Figure 1.6 (g) Sequence Diagram



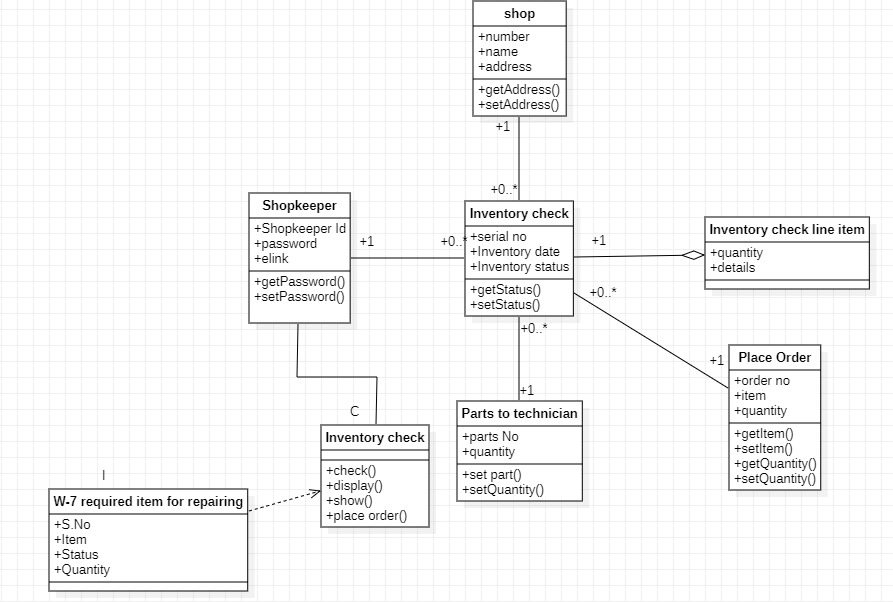


Figure 1.7 (g) Class Diagram

**Place order**

|  |  |  |
| --- | --- | --- |
| Use Case Name | Place order | Business requirement: System analysis: System design: |
| Use Case ID | MST-005 |
| Priority | High |
| Primary Business Actor: | Shopkeeper | |
| Primary system Actor: | Shopkeeper | |
| Other Participating Actors: | Vender | |
| Descriptions | This use case describes the event of a shopkeeper placing order for parts of fridge. | |
| Precondition: | The parts of fridge are not available in inventory. | |
| Trigger: | This use case initiated when the inventory is checked. | |
| Typical Course of Events: | Actor Action | System Response |
| Step 1: Shopkeeper place new order like: Compressor, Copper tube, gas thermostat, Bi-meter. | Step 2: Vender ensures all the necessary information has been provided for the product manufacture.  Step 3: The documentation of the order is prepared by Vender |
| Alternate Course | -- | |
| Conclusion: | The use case concludes when the vender prepared the order documentation. | |
| Postcondition: | Vender generates bill of the order. | |

Table 1.3 (h) Analysis Use case Table

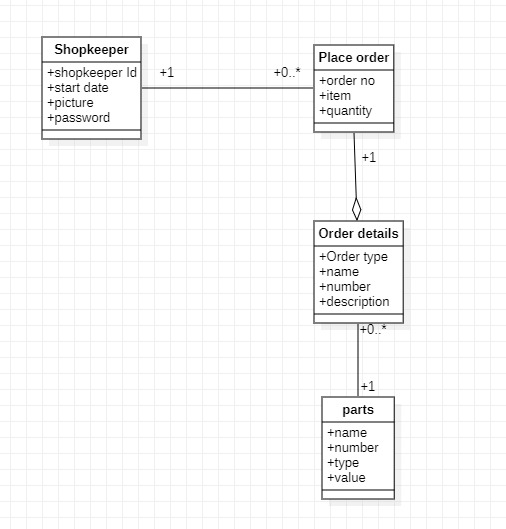


Figure 1.5 (h) Object Diagram

**Generates Bill**

|  |  |  |
| --- | --- | --- |
| Use Case Name | Generates Bill | Business requirement: System analysis: System design: |
| Use Case ID | -- |
| Priority | High |
| Primary Business Actor: | Vender | |
| Primary system Actor: | Vender | |
| Other Participating Actors: | Shopkeeper | |
| Descriptions | This use case describes the event of a vender generates bill of the order. | |
| Precondition: | Vender should have material ordered by the shopkeeper. | |
| Trigger: | This use case initiated when the shopkeeper place order. | |
| Typical Course of Events: | Actor Action | System Response |
| Step 1: Vender generates Bill of parts of fridge to shopkeeper. | Step 2: Shopkeeper accept the bill. |
| Alternate Course | -- | |
| Conclusion: | The use case concludes when the shopkeeper accept the bill. | |
| Postcondition: | Vender deliver order to shopkeeper. | |

Table 1.3 (i) Analysis Use case Table

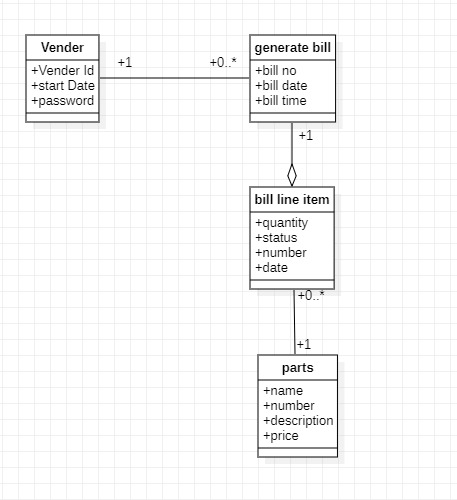


Figure 1.5 (i) Object Diagram

|  |  |  |
| --- | --- | --- |
| Use Case Name | Generates Bill | Business requirement: System analysis: System design: |
| Use Case ID | -- |  |
| Priority | High |
| Primary Business Actor: | Vender | |
| Primary system Actor: | Vender | |
| Other Participating Actors: | Shopkeeper | |
| Descriptions | This use case describes the event of a vender generates bill of the order. | |
| Precondition: | Vender should have material ordered by the shopkeeper. | |
| Trigger: | This use case initiated when the shopkeeper place order. | |
| Typical Course of Events: | Actor Action | System Response |
| Step 1: Vender generates Bill of parts of fridge to shopkeeper. | Step 2: System responds by displaying a window “w6-Bill” to Shopkeeper which contains name, bill id, description, quantity and price.  Step 3: Message is displayed that Welcome to refrigerator’s order has been delivered” |
| Alternate Course | -- | |
| Conclusion: | The use case concludes when the shopkeeper accept the bill. | |
| Postcondition: | Vender deliver order to shopkeeper. | |

Table 1.4 (i) Design Use case Table

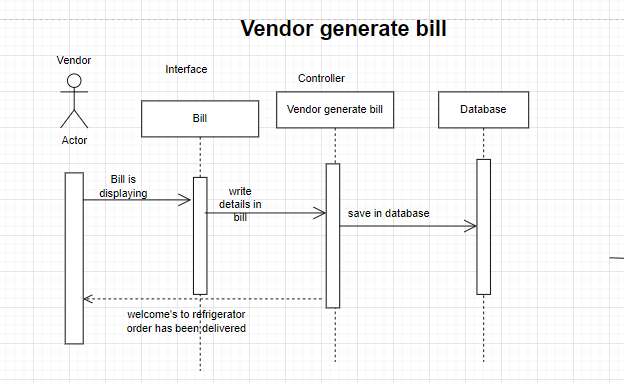


Figure 1.6 (i) Sequence Diagram

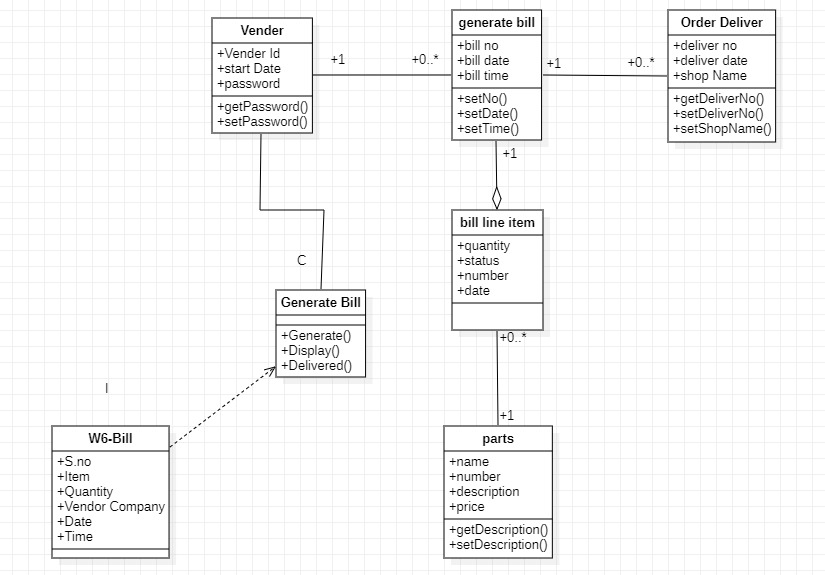


Figure 1.7 (i) Class Diagram

**Order deliver**

|  |  |  |
| --- | --- | --- |
| Use Case Name | Order deliver | Business requirement: System analysis: System design: |
| Use Case ID | -- |
| Priority | High |
| Primary Business Actor: | Vender | |
| Primary system Actor: | Vender | |
| Other Participating Actors: | Shopkeeper | |
| Descriptions | This use case describes the event of an order deliver by vender. | |
| Precondition: | Shopkeeper must give correct information of shop. | |
| Trigger: | This use case initiated when the vender generates bill. | |
| Typical Course of Events: | Actor Action | System Response |
| Step 1: Vender deliver order (parts of fridge) to shopkeeper. | Step 2: Shopkeeper accept the delivery of order. |
| Alternate Course | -- | |
| Conclusion: | The use case concludes when the shopkeeper accept the delivery of order. | |
| Postcondition: | The order has been delivered and the shopkeeper pays the bill. | |

Table 1.3 (j) Analysis Use case Table

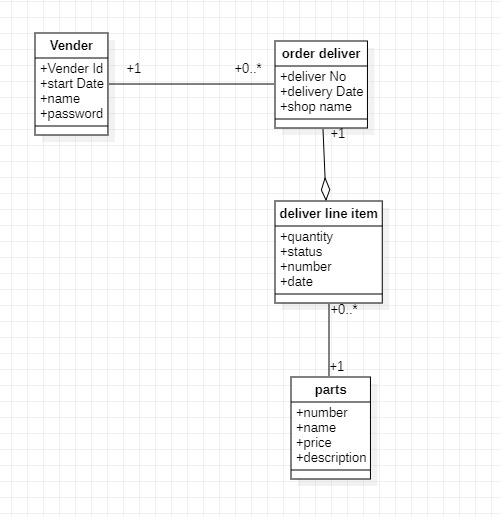


Figure 1.5 (j) Object Diagram

**Payment by Shopkeeper**

|  |  |  |
| --- | --- | --- |
| Use Case Name | Payment by Shopkeeper | Business requirement: System analysis: System design: |
| Use Case ID | MST-006 |
| Priority | High |
| Primary Business Actor: | Shopkeeper | |
| Primary system Actor: | Shopkeeper | |
| Other Participating Actors: | Vender | |
| Descriptions | This use case describes the event of a payment given by shopkeeper. | |
| Precondition: | An order should be placed. | |
| Trigger: | This use case initiated when the vender deliver order. | |
| Typical Course of Events: | Actor Action | System Response |
| Step 1: Shopkeeper give the payment and now parts of fridge are available in our inventory. | Step 2: Vender accept the payment. |
| Alternate Course | -- | |
| Conclusion: | The use case concludes when the vender accept the payment. | |
| Postcondition: | Technician repair the fridge. | |

Table 1.3 (k) Analysis Use case Table

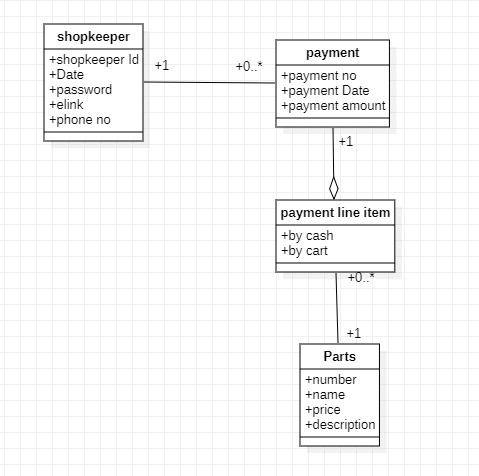


Figure 1.5 (k) Object Diagram

|  |  |  |
| --- | --- | --- |
| Use Case Name | Payment by Shopkeeper | Business requirement: System analysis: System design: |
| Use Case ID | MST-006 |
| Priority | High |
| Primary Business Actor: | Shopkeeper | |
| Primary system Actor: | Shopkeeper | |
| Other Participating Actors: | Vender | |
| Descriptions | This use case describes the event of a payment given by shopkeeper. | |
| Precondition: | An order should be placed. | |
| Trigger: | This use case initiated when the vender deliver order. | |
| Typical Course of Events: | Actor Action | System Response |
| Step 1: Shopkeeper pay the payment and now parts of fridge are available in shopkeeper’s inventory. | Step 2: Bill is displayed and also system responds by displaying a window “w7-Payment” which contains amount, balance, status and mode of payment. |
| Alternate Course | -- | |
| Conclusion: | The use case concludes when the vender accept the payment. | |
| Postcondition: | Technician repair the fridge. | |

Table 1.4 (k) Design Use case Table

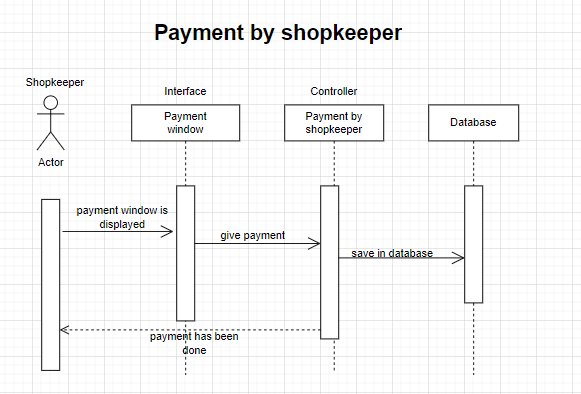


Figure 1.6 (k) Sequence Diagram

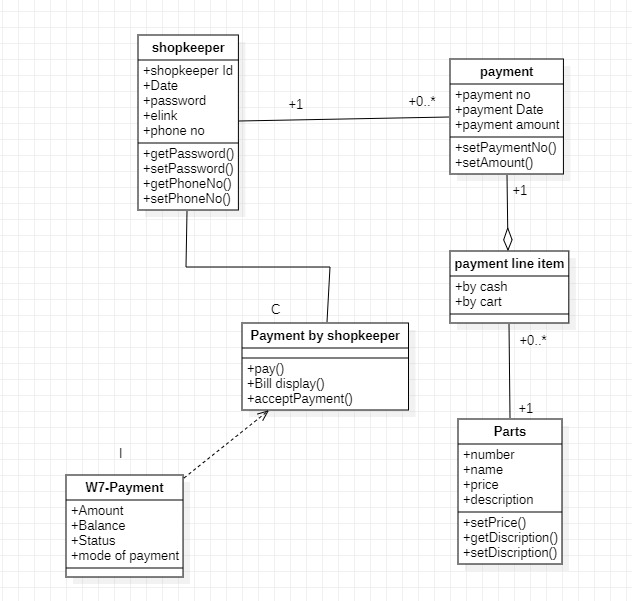


Figure 1.7 (k) Class Diagram

**Repairing of fridge**

|  |  |  |
| --- | --- | --- |
| Use Case Name | Repairing of fridge | Business requirement: System analysis: System design: |
| Use Case ID | MST-007 |
| Priority | High |
| Primary Business Actor: | Technician | |
| Primary system Actor: | Technician | |
| Other Participating Actors: | Shopkeeper | |
| Descriptions | This use case describes the event of a repairing of fridge. | |
| Precondition: | Technician availability. | |
| Trigger: | This use case initiated when the parts are available. | |
| Typical Course of Events: | Actor Action | System Response |
| Step 1: Technician repair the fridge.  Step 2: Technician give fridge to shopkeeper. | Step 3: Shopkeeper accept the fridge. |
| Alternate Course | -- | |
| Conclusion: | The use case concludes when the shopkeeper accept the fridge. | |
| Postcondition: | Shopkeeper deliver fridge to customer. | |

Table 1.3 (l) Analysis Use case Table

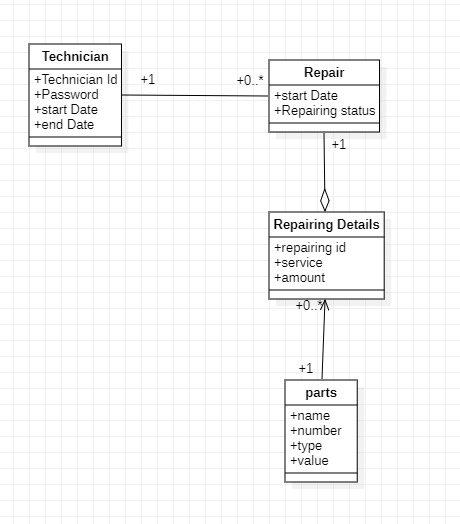


Figure 1.5 (l) Object Diagram

|  |  |  |
| --- | --- | --- |
| Use Case Name | Repairing of fridge | Business requirement: System analysis: System design: |
| Use Case ID | MST-007 |
| Priority | High |
| Primary Business Actor: | Technician | |
| Primary system Actor: | Technician | |
| Other Participating Actors: | Shopkeeper | |
| Descriptions | This use case describes the event of a repairing of fridge. | |
| Precondition: | Technician availability. | |
| Trigger: | This use case initiated when the parts are available. | |
| Typical Course of Events: | Actor Action | System Response |
| Step 1: Technician repair the fridge. | Step2: System responds by displaying a window “w8-Repair status” which shows a message that repairing of fridge is done and fridge is deliver to shopkeeper. |
| Alternate Course | -- | |
| Conclusion: | The use case concludes when the shopkeeper accept the fridge. | |
| Postcondition: | Shopkeeper deliver fridge to customer. | |

Table 1.4 (l) Design Use case Table

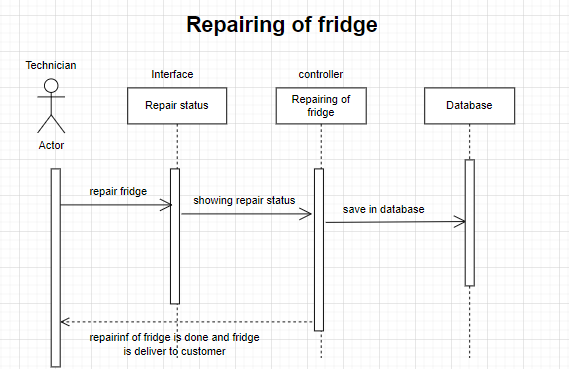
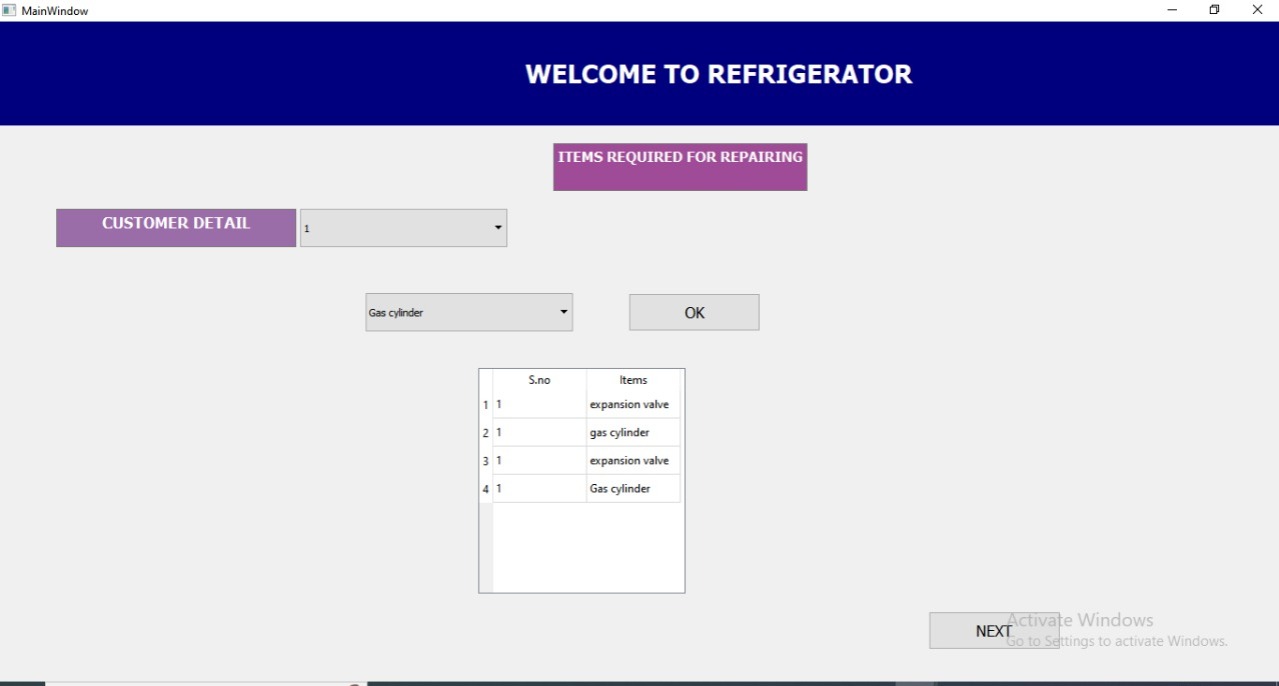


Figure 1.6 (l) Sequence Diagram



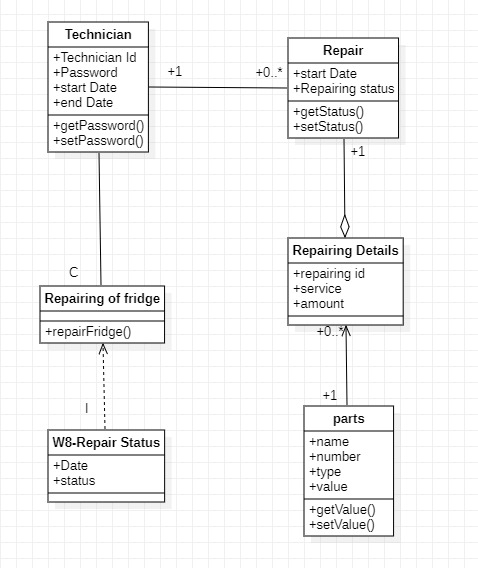


Figure 1.7 (l) Class Diagram

**Deliver to Customer**

|  |  |  |
| --- | --- | --- |
| Use Case Name | Deliver to customer | Business requirement: System analysis: System design: |
| Use Case ID | MST-008 |
| Priority | High |
| Primary Business Actor: | Shopkeeper | |
| Primary system Actor: | Shopkeeper | |
| Other Participating Actors: | Customer | |
| Descriptions | This use case describes the event of a shopkeeper deliver fridge to customer. | |
| Precondition: | Customer information must be correct. | |
| Trigger: | This use case initiated when the technician repair the fridge. | |
| Typical Course of Events: | Actor Action | System Response |
| Step 1: Shopkeeper deliver fridge to customer. | Step 3: Customer accept the fridge. |
| Alternate Course | -- | |
| Conclusion: | The use case concludes when the customer accept the fridge. | |
| Postcondition: | Customer give full payment. | |

Table 1.3 (m) Analysis Use case Table

**Full payment**

|  |  |  |
| --- | --- | --- |
| Use Case Name | Full payment | Business requirement: System analysis: System design: |
| Use Case ID | -- |
| Priority | High |
| Primary Business Actor: | Customer | |
| Primary system Actor: | Customer | |
| Other Participating Actors: | Shopkeeper | |
| Descriptions | This use case describes the event of a full payment given by customer. | |
| Precondition: | Repairing of the refrigerator must be done successfully. | |
| Trigger: | This use case initiated when the fridge is delivered to customer. | |
| Typical Course of Events: | Actor Action | System Response |
| Step 1: Customer give full payment. | Step 3: Shopkeeper accept the payment. |
| Alternate Course | -- | |
| Conclusion: | The use case concludes when the shopkeeper accept the payment. | |
| Postcondition: | -- | |

Table 1.3 (n) Analysis Use case Table

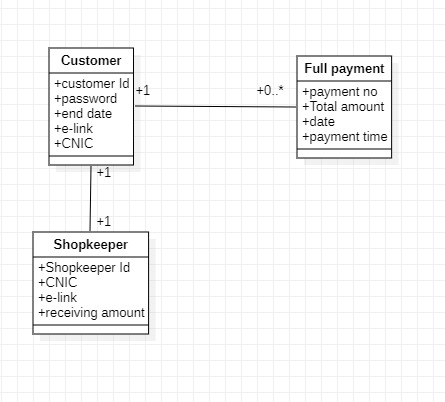


Figure 1.5 (n) Object Diagram

|  |  |  |
| --- | --- | --- |
| Use Case Name | Full payment | Business requirement: System analysis: System design: |
| Use Case ID | -- |
| Priority | High |
| Primary Business Actor: | Customer | |
| Primary system Actor: | Customer | |
| Other Participating Actors: | Shopkeeper | |
| Descriptions | This use case describes the event of a full payment given by customer. | |
| Precondition: | Repairing of the refrigerator must be done successfully. | |
| Trigger: | This use case initiated when the fridge is delivered to customer. | |
| Typical Course of Events: | Actor Action | System Response |
| Step 1: Customer pay full payment. | Step 3: When customer accept the price then system responds by displaying a window “w9-Payment” which contains name, bill no, phone no, fridge name, address and amount. |
| Alternate Course | -- | |
| Conclusion: | The use case concludes when the shopkeeper accept the payment. | |
| Postcondition: | -- | |

Table 1.4 (n) Design Use case Table

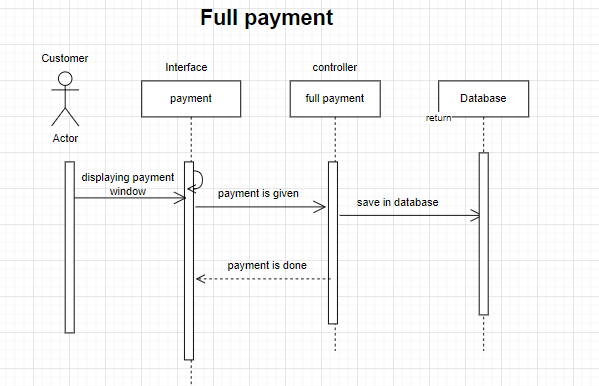
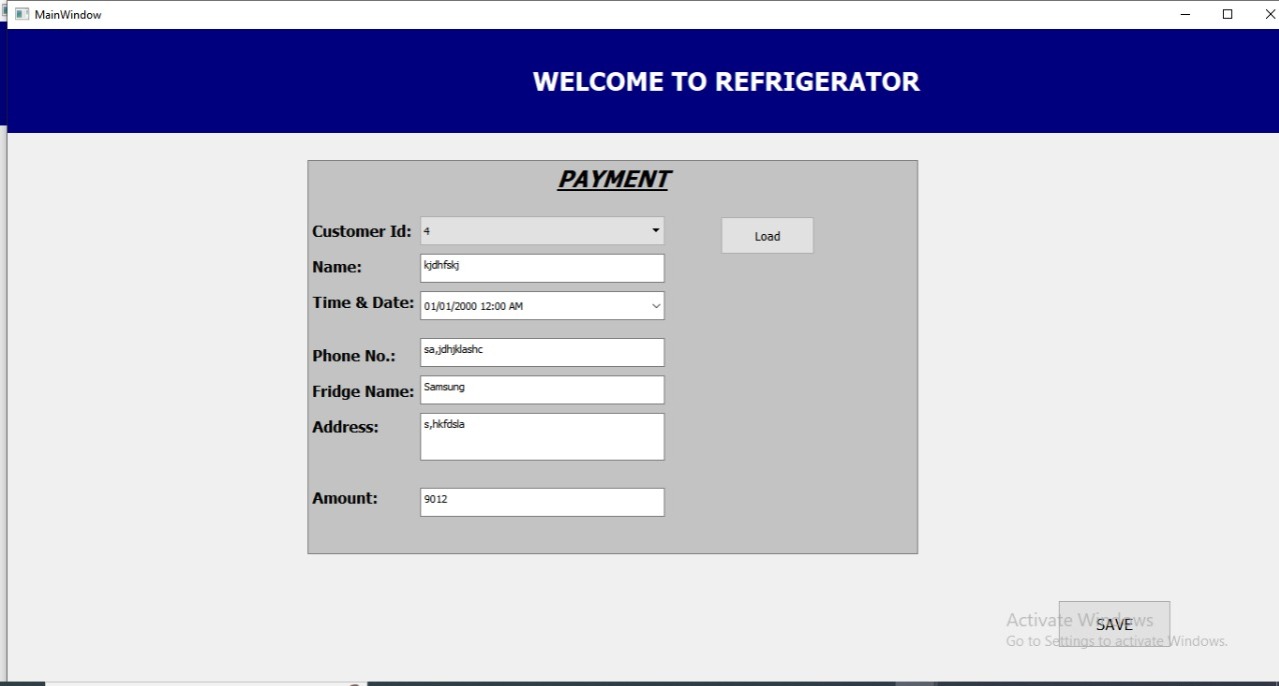


Figure 1.6 (n) Sequence Diagram



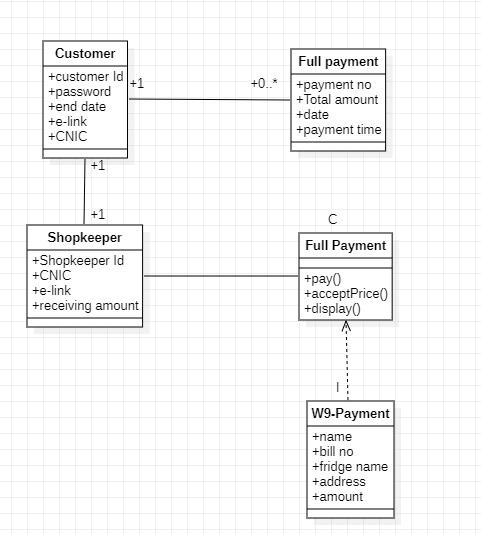


Figure 1.7 (n) Class Diagram

# CLASS DIAGRAM

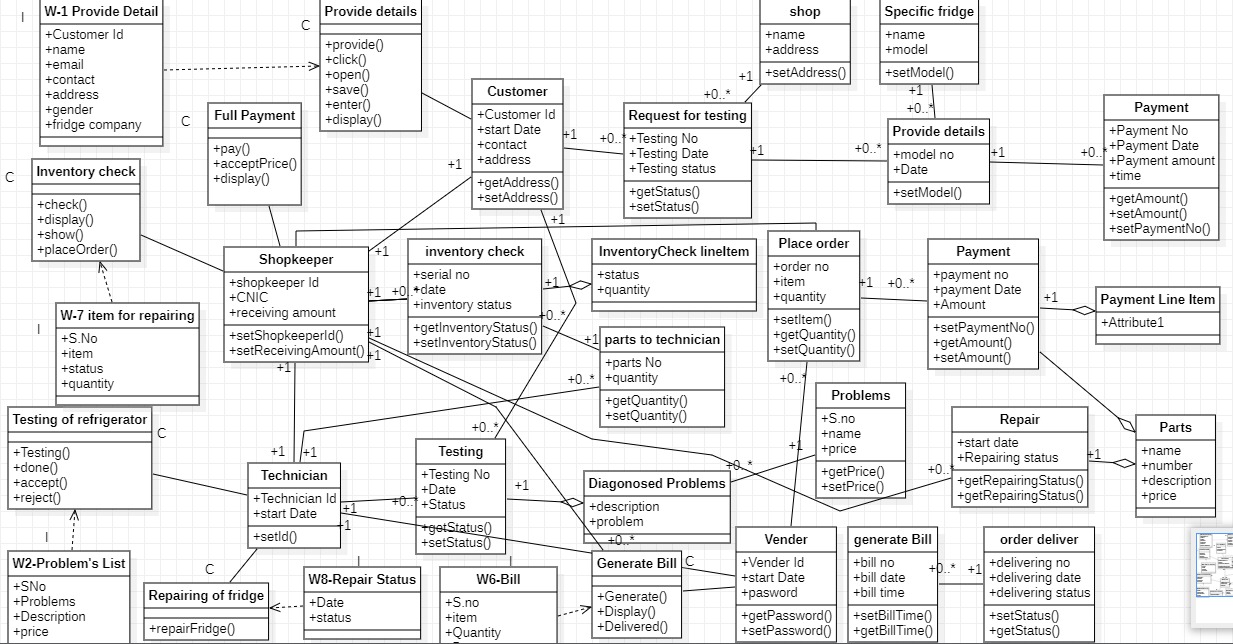
****

Figure 1.8 Class Diagram (Complete)

# DATABASE DIAGRAM

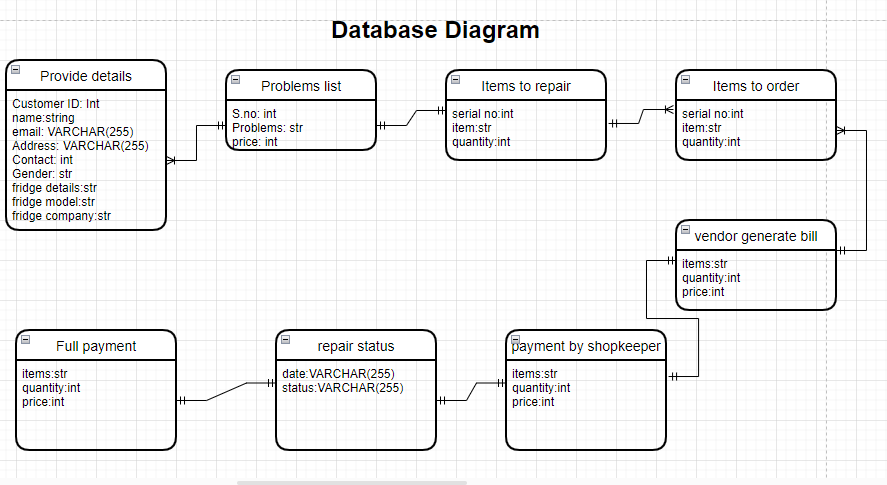


Figure 1.9 Database Diagram

**I**MPLEMENTATION

Actual Form:

The application is in the form of “Website”. In the application, User first creates accounts for customers, shopkeeper and technician, then the user can create pages of provide details, problems in refrigerator, inventory check, place order, vendor payment, and generate bill.

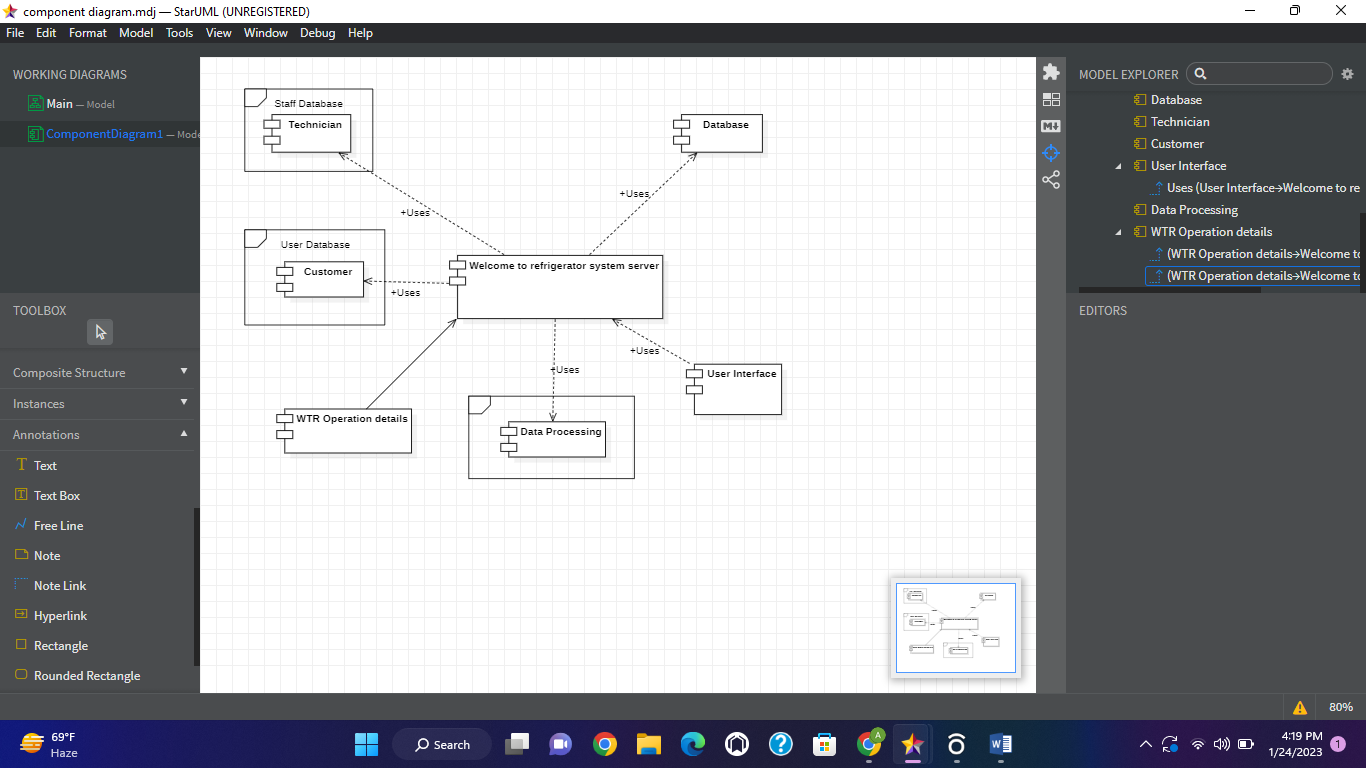


Figure 2.0 Component Diagram

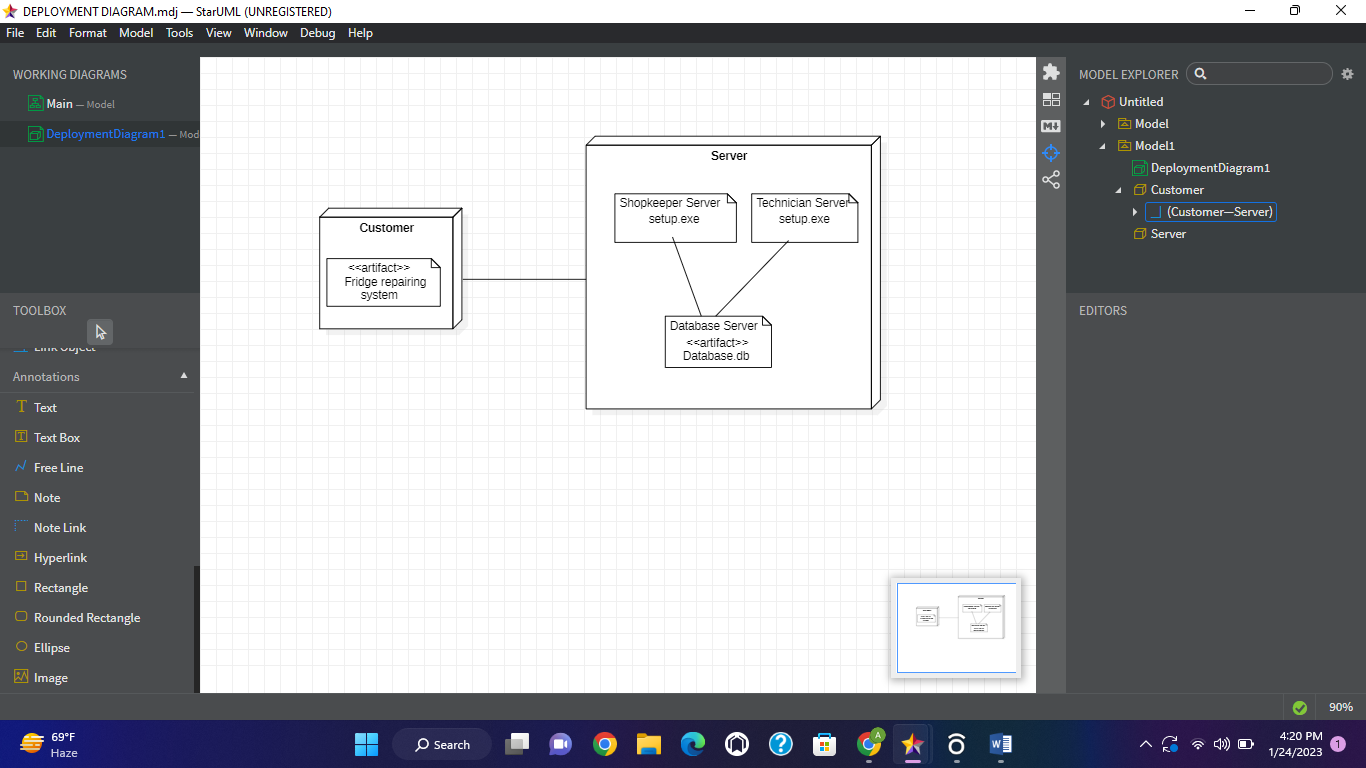
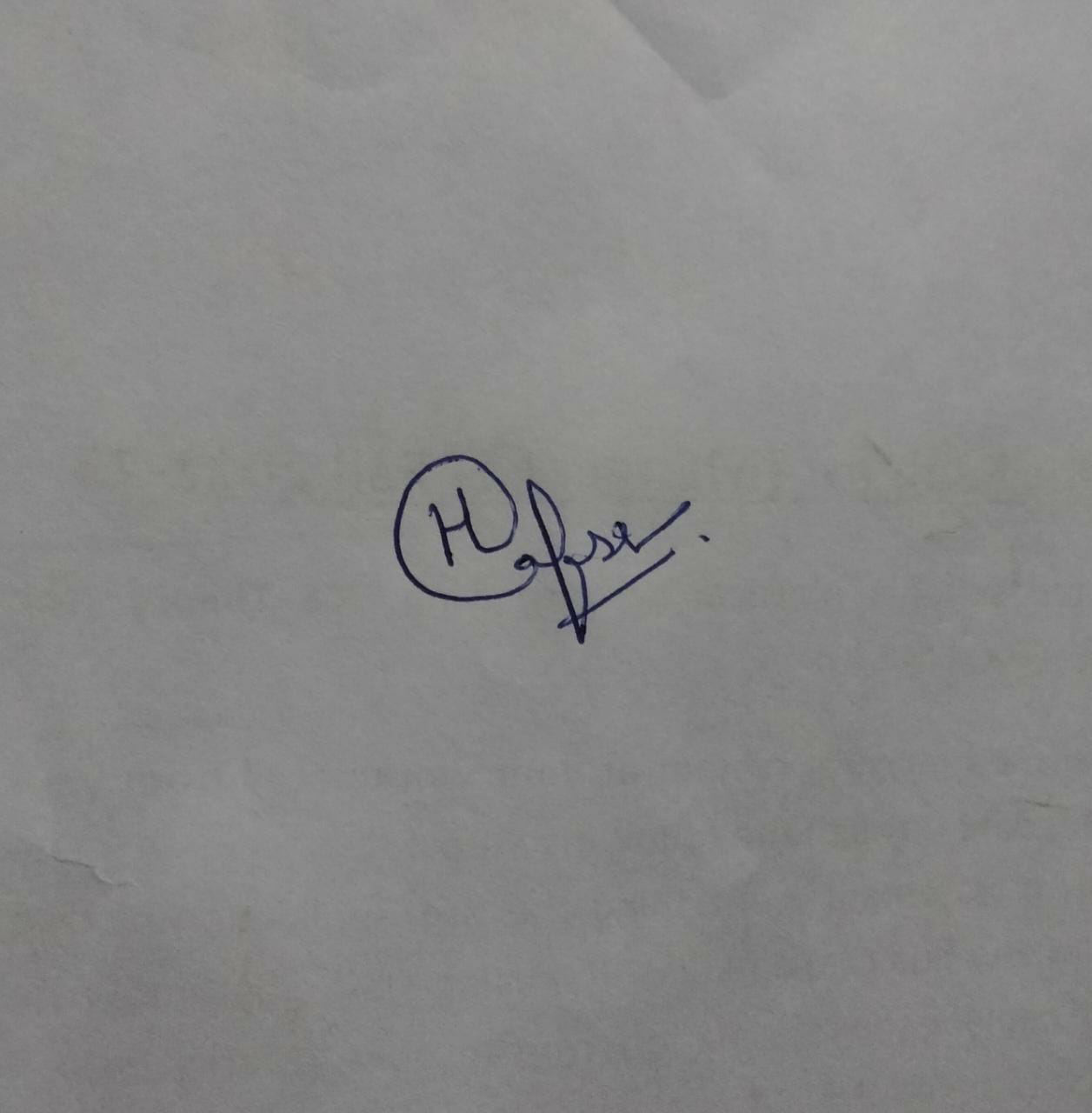


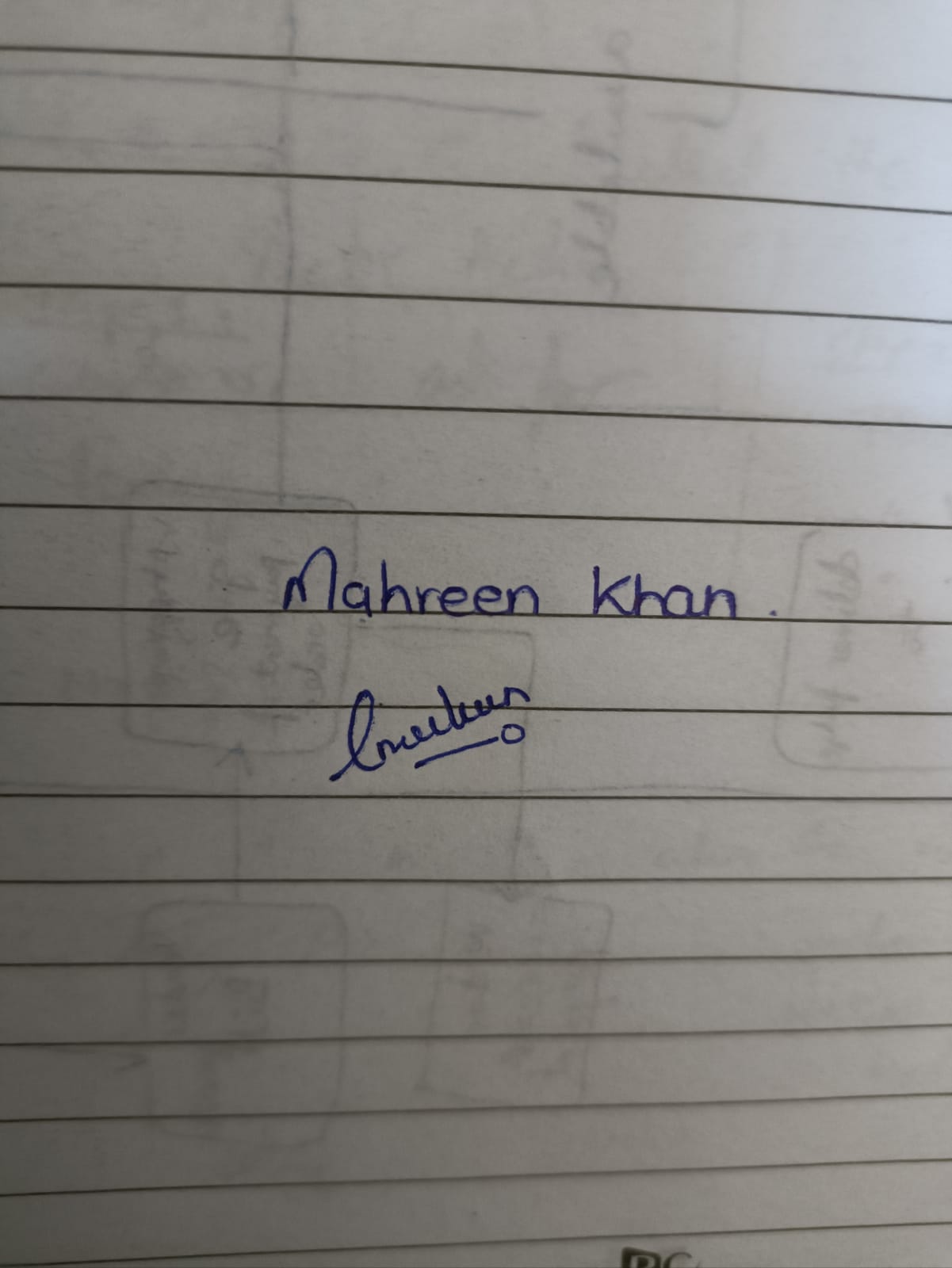
Figure 2.1 Deployment Diagram

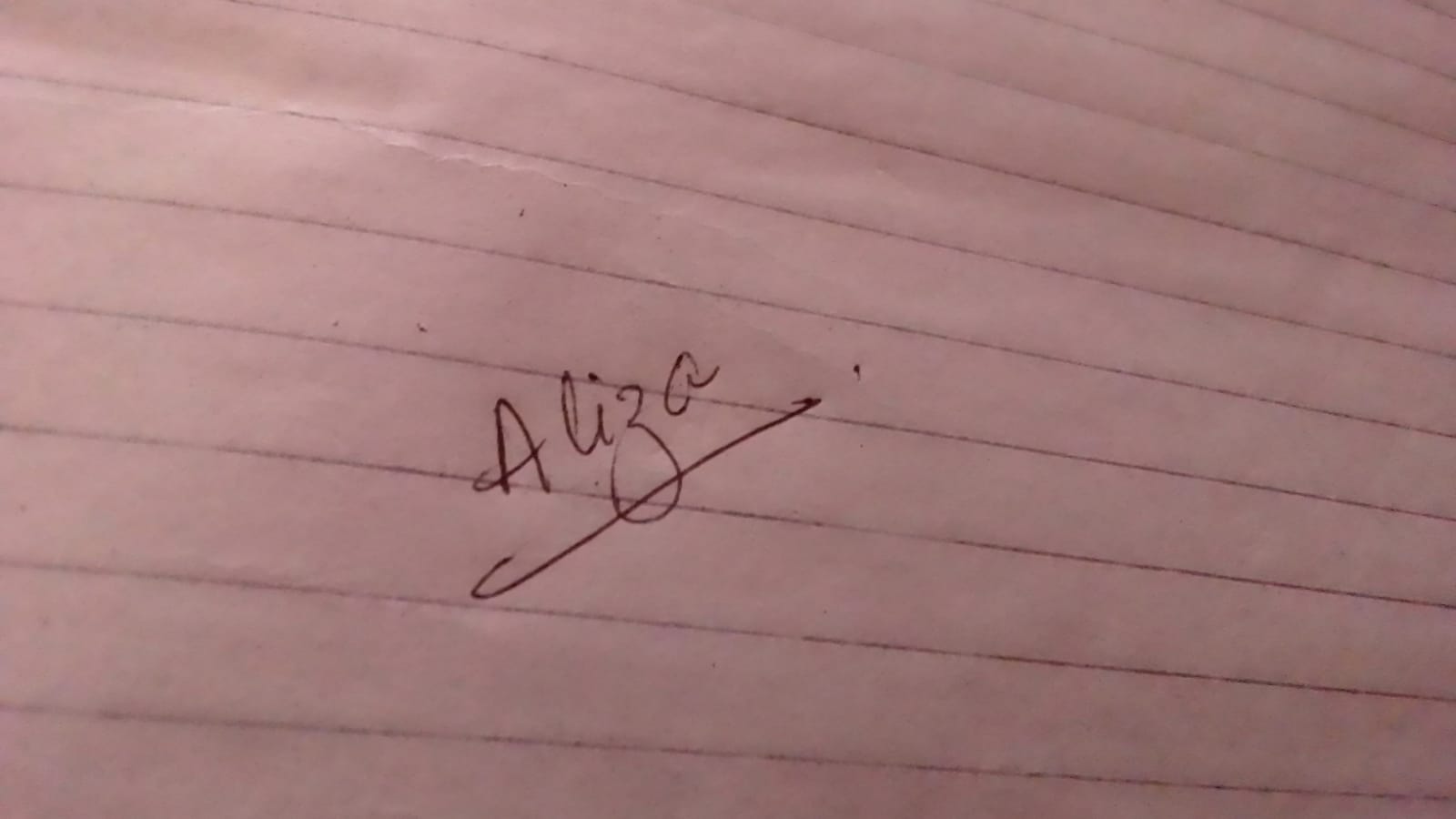
# SUPPORT

Thank you for your support. We truly appreciate your business and look forward to serving you again. Thank you for being our valued customer. We are so grateful and hope we met your expectations. If you ever face any problem regarding anything whether it’s related to bug, maintenance or even enhancement, so feel free to call us for help. We shall provide you our service till six months at free of cost.

We hereby undertake that our team will be available for you for the next six months for any kind of problem.







# CONCLUSIONS AND SUGGESTIONS FOR FUTURE WORK

The application is in the form of “Website”. In the application, User first creates accounts for customers, shopkeeper and technician, then the user can create pages of provide details, problems in refrigerator, place order, vendor payment, and generate bill.

* Provide Detail :

The customer will enter his/her information along with the information of his/her refrigerator and all the data will be saved in the database.

* Problems in refrigerator :

The technician will detect the problem and tell the price according to it. The customer after reviewing the price will either accept the offer or reject it.

* Place order :

If the parts required are not available in the inventory then shopkeeper will place an order to vendor.

* Vendor Payment:

The vendor will generate the bill of the required part and the shopkeeper will pay the amount either through cash or card and this transaction also save in a database.

* Generate Bill:

The shopkeeper will generate the bill after technician repaired the refrigerator and the customer will receive the fridge and will pay the bill and this transaction also save in database.

FUTURE WORK

• The rights could be given to customer so that he could track his order.

• The panel will be given to the customers so he could pay online.

• We could build an android application as well; through which everyone can monitor everything at anywhere anytime.

• An option can be added such that the purchase order be sent directly to company/vendor's mail instead of manually having to print the document and giving it to them. Email is more economical.

# REFERENCES

Conference

[Bentley7] Jeffrey L. Whitten, Lonnie D. Bentley (1986) “system analysis and design methods” *process of examining a business situation with the intent of improving it through better procedures and methods*., McGraw-Hill, Inc., Professional Book Group 11 West 19th Street New York, NY, United States.

Book example:

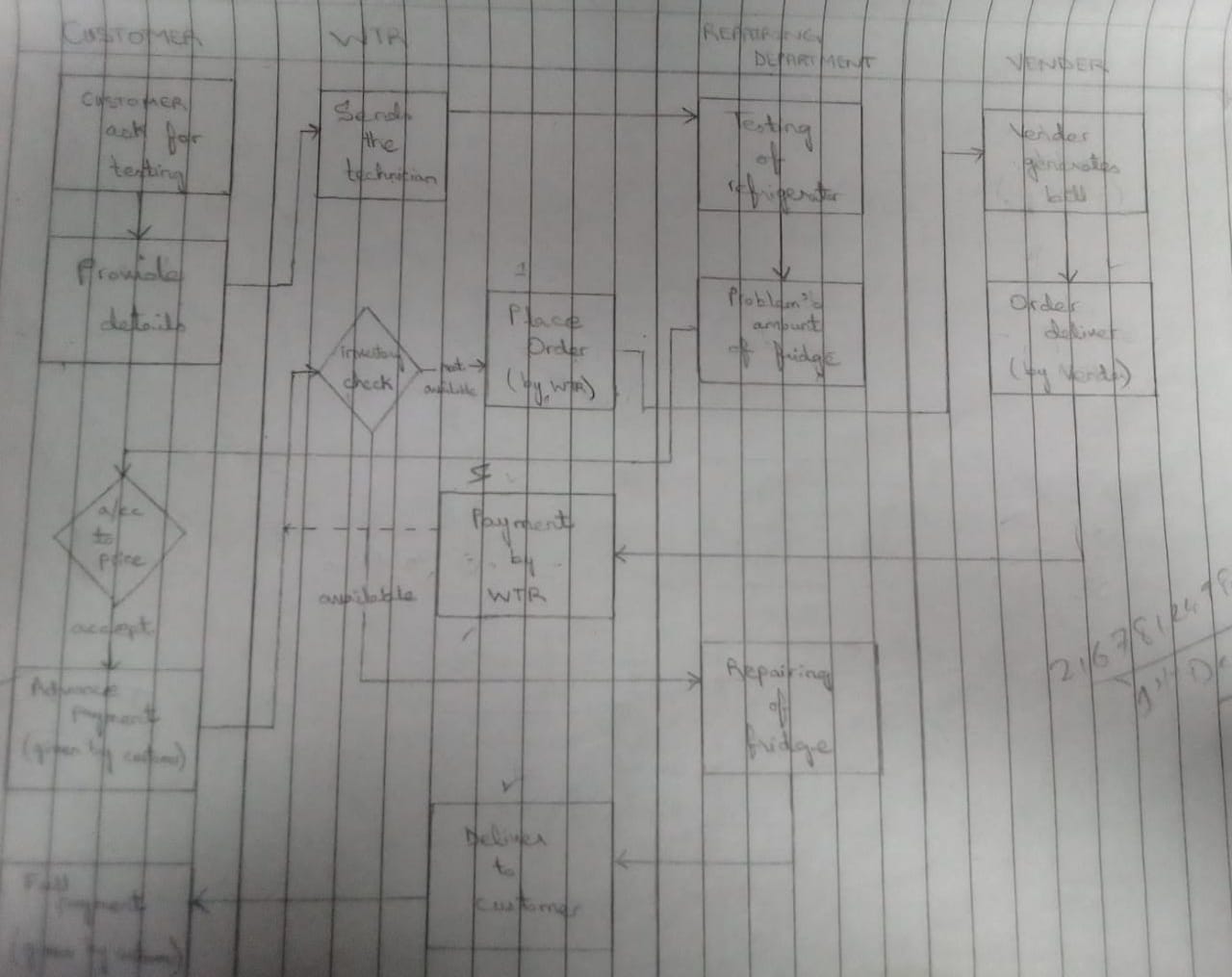
1. [Pressman7] Roger S. Pressman (2019) "Software engineering a practitioner’s approach,”

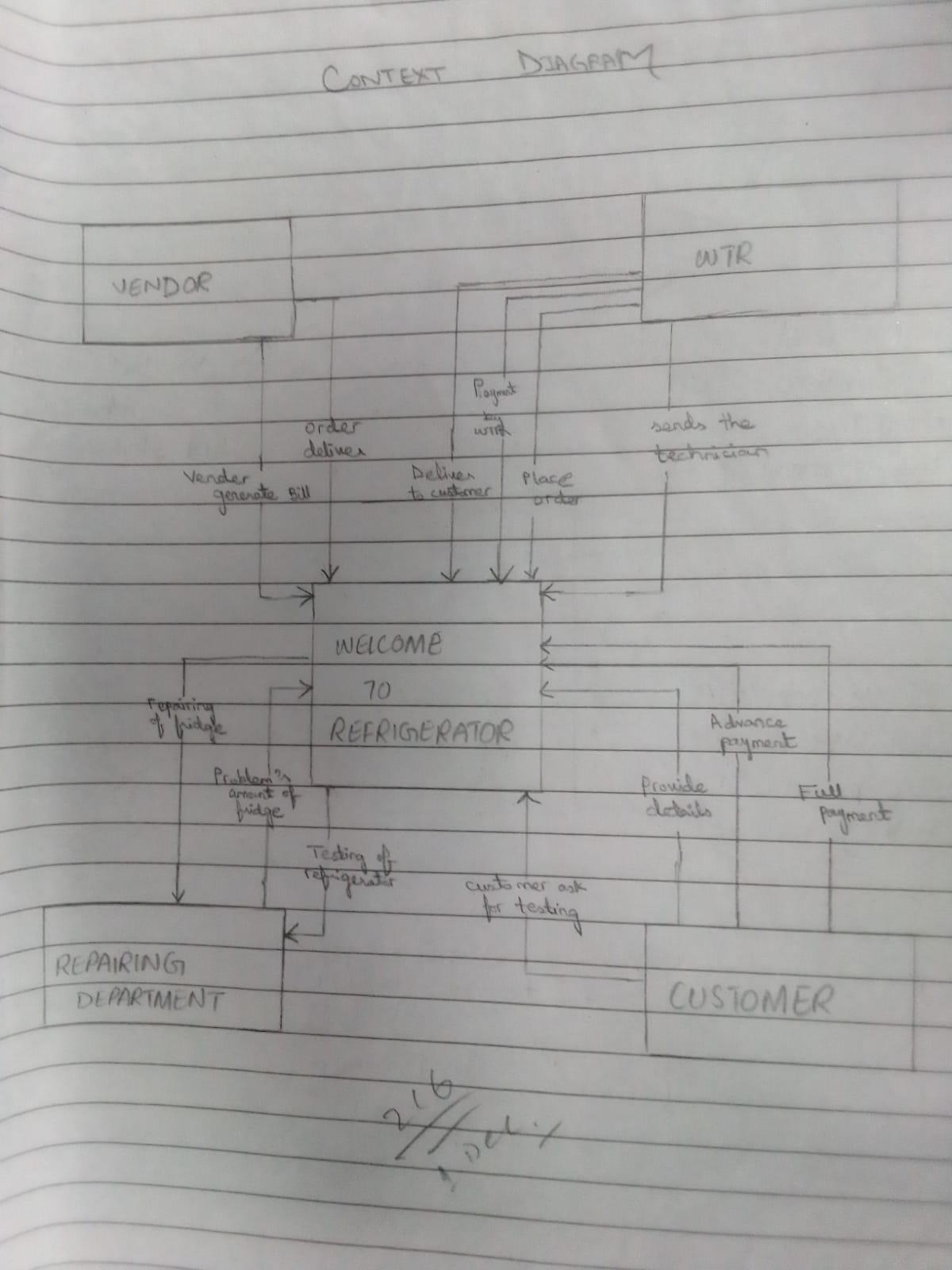
Website Reference

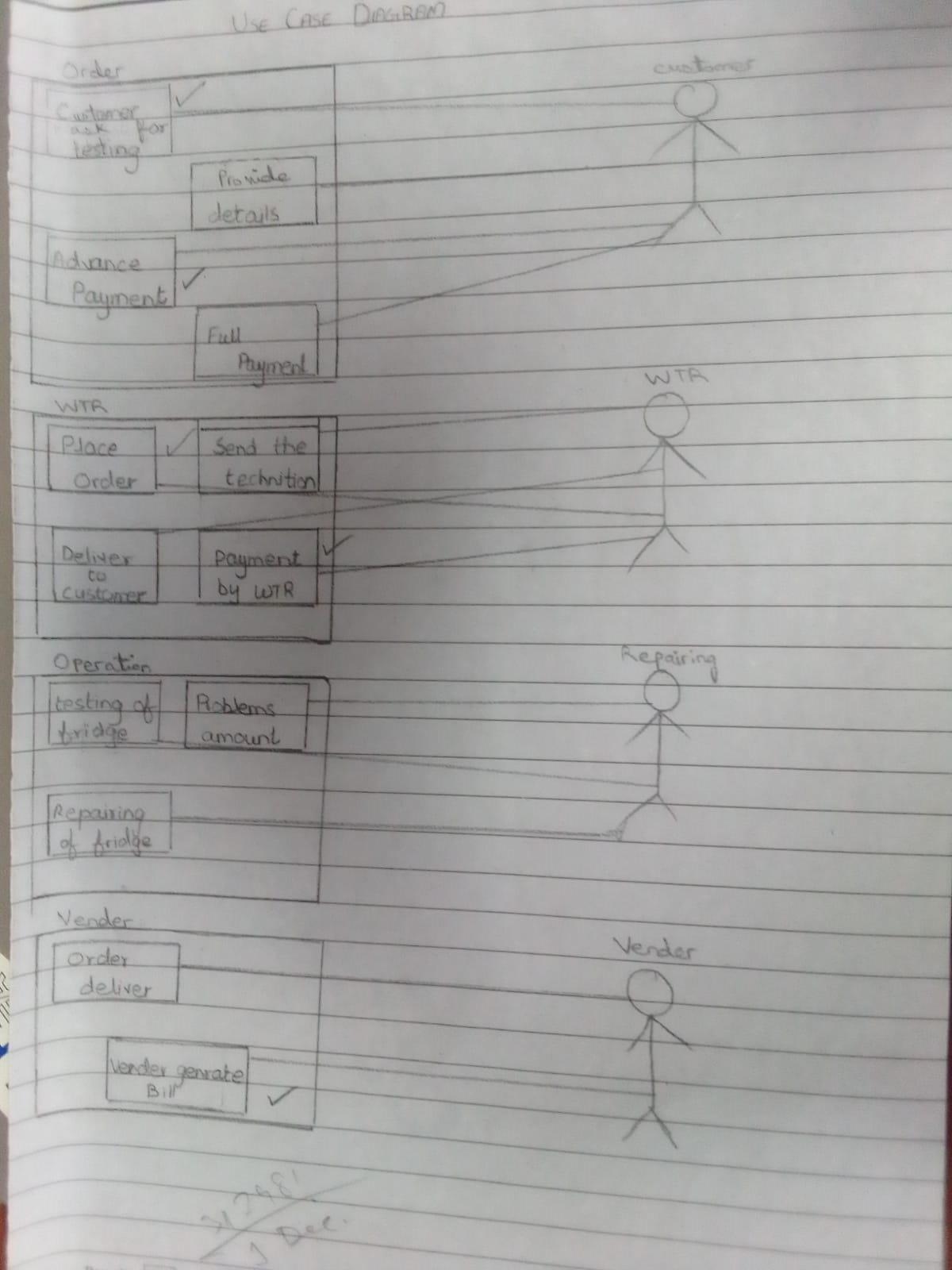
1. <https://www.youtube.com/channel/UCyHta2dyCTkf29AB67AYn7A>

General information for Object oriented diagrams.

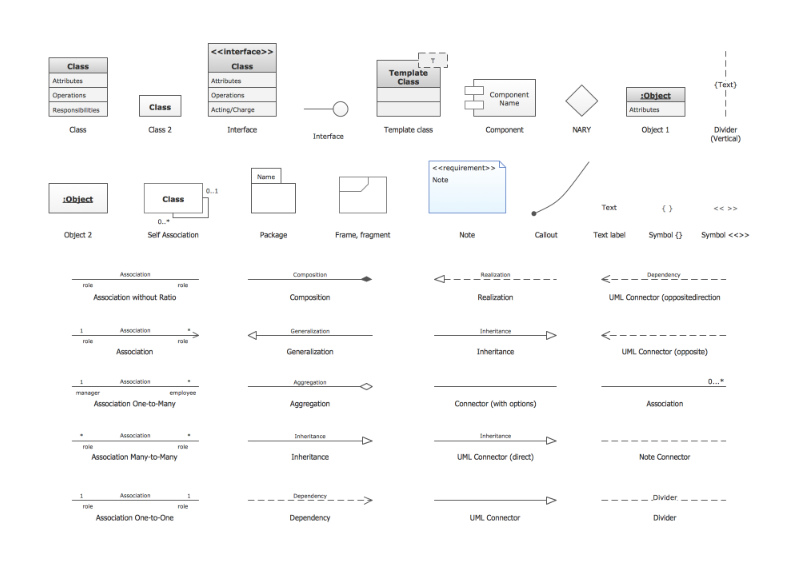
# APPENDICES SHOULD APPEAR AFTER THE LIST OF REFERENCES







# LIST OF SYMBOLS



# GLOSSARY

Context Diagram: the interactions between a system and other actors (external factors) with which the system is designed to interface.

Use case Diagram: model the behavior of a system and help to capture the requirements of the system.

Narrations: describe the desired response of a system when it receives external requests.

UML: unified modeling language.

Object Diagram: it is a graph of instances, including objects and data values.

SSD: system sequence diagram.

Sequence Diagram: shows object interactions arranged in time sequence in the field of software engineering.

Component Diagram: describes the organization and wiring of the physical components in a system.

Deployment Diagram: to visualize the topology of the physical components of a system, where the software components are deployed.

PIECES: Performance, Information (and data), Economics, Control, Efficiency, Services.

PIECES Framework: framework containing the categories of classification and problem-solving problems.