
Software Requirements Specification

For

<Blood Bank Management System>

Prepared by: Hamza Khattak

Ilhan Shah

Shaikh Mubashir Akhter

Abdul Wahab

Ammar Merchant

Organized SIK Corporation

<2/9/2021>

Table of Contents

Table of Contents	2
Revision History	2
1. Introduction	1
1.1 Purpose	1
1.2 Document Conventions	1
1.3 Intended Audience and Reading Suggestions.....	1
1.4 Product Scope	1
1.5 References	1
2. Overall Description	2
2.1 Product Perspective	2
2.2 Product Functions	2
2.3 User Classes and Characteristics	3
2.4 Operating Environment	3
2.5 Design and Implementation Constraints.....	4
2.6 User Documentation	4
3. External Interface Requirements	5
3.1 User Interfaces	5
3.2 Software Interfaces	5
3.3 Communications Interfaces	5
4. System Features	6
4.1 Blood Management module:.....	6
4.2 Blood Receiver Module:.....	6
4.3 Blood Test Module:	6
4.4 Login/Signup Module:.....	6
4.5 White Blood Cell Module:.....	6
4.6 Blood Grouping Module:.....	9
5. Other Nonfunctional Requirements	10
5.1 Performance Requirements	10
5.2 Safety Requirements	10
5.3 Security Requirements	11
6. Other Requirements	11
6.1 Interface Blueprint	12

Revision History

Name	Date	Reason For Changes	Version
SIK Corp	2/9/2021	Change in modules/ Add in Modules	1.2

--	--	--	--

1. Introduction

1.1 Purpose

The main goal of the Blood Bank and Donor Management System project is to monitor Blood Bank data, Blood cells, Blood stock, Donor List. It manages all the Blood Bank, Donor, and Blood stock data. The project is entirely administrative and therefore access is guaranteed only to the administrator. The project's aim is to develop an application system to minimize the manual work for Blood Bank, Donor, Blood Group management. It monitors all of the Blood Group information, Blood cells, Blood supply and Donor List Purpose

1.2 Document Conventions

Every requirement statement is to have its own priority.

1.3 Intended Audience and Reading Suggestions

- *Developers who can review project's capabilities and more easily understand where their efforts should be targeted to improve or add more features to it (design and code the application – it sets the guidelines for future development).*
- *Project testers can use this document as a base for their testing strategy as some bugs are easier to find using a requirements document. This way testing becomes more methodically organized.*
- *End users of this application who wish to read about what this project can do.(Section 2)*

1.4 Product Scope

The main aim of this project will therefore be to find more effective ways of managing the database of blood banks and blood donors and establish a forum for people connected to potential blood donors in the region. The donors of blood are recruited using a recruiting algorithm that is also a target of this project.

1.5 References

All for references requirement will be attached at the end for the document.

2. Overall Description

2.1 Product Perspective

OPS is aimed towards the vendors who want to reach out to the maximum cross-section of customer and common people who can be potential customer. This project envisages bridging the gap between the seller, the retailer and the customer. OPS should be user friendly, 'quick to learn' and reliable software for the above purpose. OPS is intended to be a stand-alone product and should not depend on the availability of other software. It should run on both UNIX and Windows based platform.

2.2 Product Functions

User: Administrator

- **Functions:**

*The Administrator is the super user and has complete control over all the activities that Can be performed. The application notifies the administrator of all shop Creation requests, and the administrator can then approve or reject them.
The administrator also manages the list of available product categories.*

User: Donor/Receiver

- **Functions:**

A Customer can browse through the shops and choose products to place in a virtual shopping cart. The shopping cart details can be viewed and items can be removed from the cart. To proceed with the purchase, the customer is prompted to login. Also, the customer can modify personal profile information (such as phone number and shipping address) stored by the application.

User: Admin

- **Functions:**

Department under a Purchase manager to overlook purchasing activities if warehousing needs arise.

- **Functions:**

Sales department under a Sales manager who will look after the sale of products and services, the most important activity.

- **Functions:**

Accounts department under an Accounts manager to look after the accounting activities of the enterprise

2.3 User Classes and Characteristics

Administrator:

- Database Management: Control the database and keep track of all records of customers and employee details.
- View all details: View the details of all employees and control the whole site.
- Advertising the Site: Responsible for making advertisements for the site.
- View customer details: View the personal details of the customer.
- Managing Sales to Customers: Responsible for properly allocating the selected product according to the customer's choice and delivering product to the customer.
- View Product Stocks: Keep track of each product item's stocks for selling purpose

Customers:

- Login: Customers must have a valid login id to enter into the site.
- Registration: New users can sign up by creating new ID.
- View and edit Own Details: Can view/edit his personal details, payment details, and details about services provided.
- Choosing products: Can view all available products make a choice for purchasing products.
- Purchasing: Can purchase any product through valid credit card.
- Logout: Customer must logout of the site after purchasing products.

2.4 Operating Environment

The OPS is a website that shall operate in all famous browsers, for a model we are taking Microsoft Internet Explorer versions 7.0. 8.0 And 9.0. And Google Chrome

2.5 Design and Implementation Constraints

There are some of the thing which have to mentioned but it will overcome till the end, that are

- There is no maintainability of back up so availability will get affected.
- Limited to HTTP/HTTPS.
- Real-life credit card validation and Banking system is not implemented.
- No multilingual support

2.6 User Documentation

□ PROJECT HOST LINK: <http://www.AbdulWahab9287.somee.com> □

GitzHub Link:

3. External Interface Requirements

3.1 User Interfaces

- C#: C#.Net is a programming platform— part of the MVC 5 framework for developing and running distributed multi-task architecture web application
- HTML, XML: Hyper Text Markup Language and Extensible markup Language are the predominant markup languages for web pages. It provides a means to describe the structure of text-based information in a document and to supplement that text with interactive forms, embedded images, and other objects.
- SQL Server: SQL server is used Tools & Development Environment
- Microsoft Visual Studio: Microsoft Visual Studio is a toolkit which is designed for the creation of complex projects, providing fully dynamic web application.

3.2 Software Interfaces

Database:

- *MySQL*

Operating system:

- *Web*

Frontend:

- *C#*
- *MVC.*

3.3 Communications Interfaces

- Should run on 500GHz and 64MB machine.
- Should have a proper internet connection.
- The response time for occurs a change will be more than 3 seconds.
- The response time for access the database will be no more 5 seconds

4. System Features

There are total of many modules in our project some of the methods are

4.1 Blood Management module:

- Add Blood Record: A donor's blood can be added in the inventory.
- Delete Blood Record: A donor's blood can Deleted in the inventory.
- View Blood Record: A donor's blood can be viewed from the inventory.

4.2 Blood Receiver Module:

- Add Blood Receiver Record: A Blood Receiver Record can be added in the inventory.
- Delete Blood Receiver Record: A Blood Receiver Record can Deleted in the inventory. □
- View Blood Receiver Record: A Blood Receiver Record can be viewed from the inventory.

4.3 Blood Test Module:

- Blood test Request can be made.
- Add that request in the system.
- Forwarded that request to the donation module or receivers list if blood tester requested to donate.
- Request can be deleted.

4.4 Login/Signup Module:

- User can be logged is receiver
- User can logged in as the donor.
- User can be logged as anonymous user until further request made for test, receiver or donor.

4.5 White Blood Cell Module:

- **Receiver Module:**
- Add White Blood Receiver Record: A White Blood Receiver Record can be added in the inventory.
- Delete White Blood Receiver Record: A White Blood Receiver Record can Deleted in the inventory.

- View White Blood Receiver Record: A White Blood Receiver Record can be viewed from the inventory.
- **Donor Module:**
- Add White Blood Donor Record: A White Blood Donor Record can be added in the inventory.
- Delete White Blood Donor Record: A White Blood Donor Record can Deleted in the inventory.
- View White Blood Donor Record: A White Blood Donor Record can be viewed from the inventory.

4.6 Blood Grouping Module:

- A tabular system which is produced by system from inventory this information of all blood groups available and which blood group is matched with the user requested.
- A table system which will provide a systematic display of Blood groups and types which will make receiver to obtain his required blood type.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- *The application should load and be usable within 3 seconds*
- *The application should update the interface on interaction within 2 seconds*
- *The database should be normalized to prevent redundant data and improve performance*
- *The database should be distributed to prevent outages*

5.2 Safety Requirements

·System use shall not cause any harm to human users.

5.3 Security Requirements

- System will use secured database
- Normal users can just read information but they cannot edit or modify anything except their personal and some other information.
- System will have different types of users and every user has access constraints.
- Error handling: OPS shall handle expected and non-expected errors in ways that prevent loss in information and long downtime period.

6. Interface Blueprints.

Following are the blueprints of some functions.

Interface for User(Receiver/Donor):

6.1 Registration Interface:

The user will enter his personal details like Name, User Name, Password, Date of Birth, Address, Registration Type, etc. Users will be warned about any mistakes on data format or any other constraints by validation notes and error messages. When the button "save" button is clicked, the server will check if the username or email is already taken and alert the user. If everything is entered correctly and saved a new user will be created.

6.2 Search:

The customer can enter the type of item he is looking for and the specifications he is interested in them he can click on "Search". User can also use advanced search for more options. For e.g.: the user can filter the results basing on various aspects such as area or nearest spot etc. And also they can sort the product display according to their wish (relevance, price in ascending or descending, popularity).

6.3 Cart:

This will be a space for the customer where he/she can store the items he/she wishes to buy. The user can also remove items from cart prior to checkout. Once the user decides to buy the items it cart, the user is directed to the payment page for making payment.

6.4 Payment:

The user given options with various modes of payment (online payment through credit/debit cards, via net or mobile banking or cash on delivery) out of which he chooses one. The chose mode of transaction is carried therefore by proper verification and authentication of bank details.

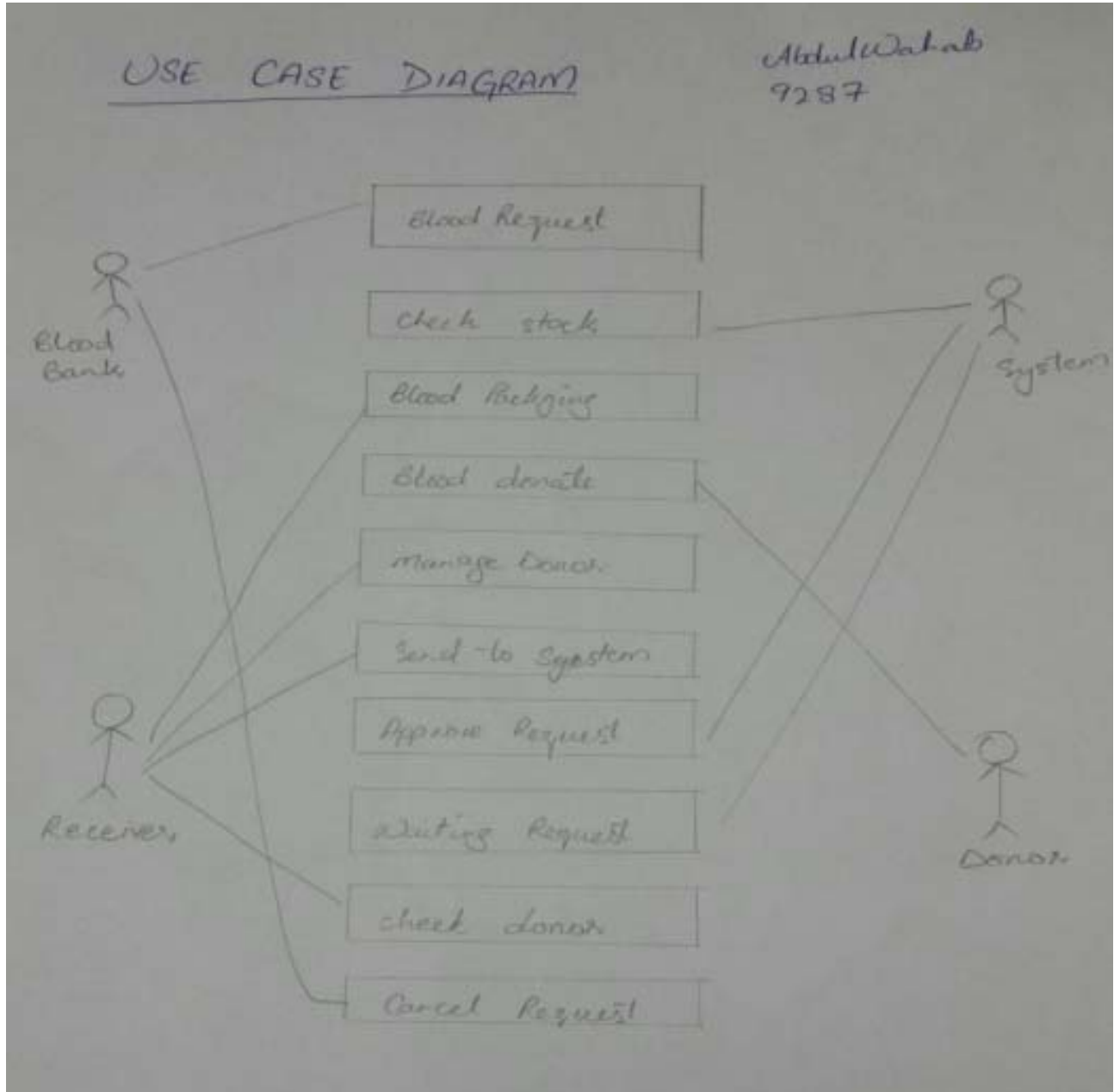
Interface for Administrator:

The shop owner will have a different login id using which he can access his account that contains a control panel that allows him to contact the administrator, set up/ maintain shop etc. This control panel will allow the shop owner to do the following things:

- Add/Remove items. The corresponding changes will be done in the database by the Admin.
- Create advertisements for his products.
- Generate a sales report.
- Discontinue the shop.

Diagrams:

Use Case Diagram:



Admin can maintain portal.

Primary Actor:

Admin, User, Receiver.

Goal in context:

Uses some web site to make Donate and receives

Preconditions:

View Items, Make Purchase and Client Register. View

Items use case could be used by customer as top level if

customer only wants to find and see some products.

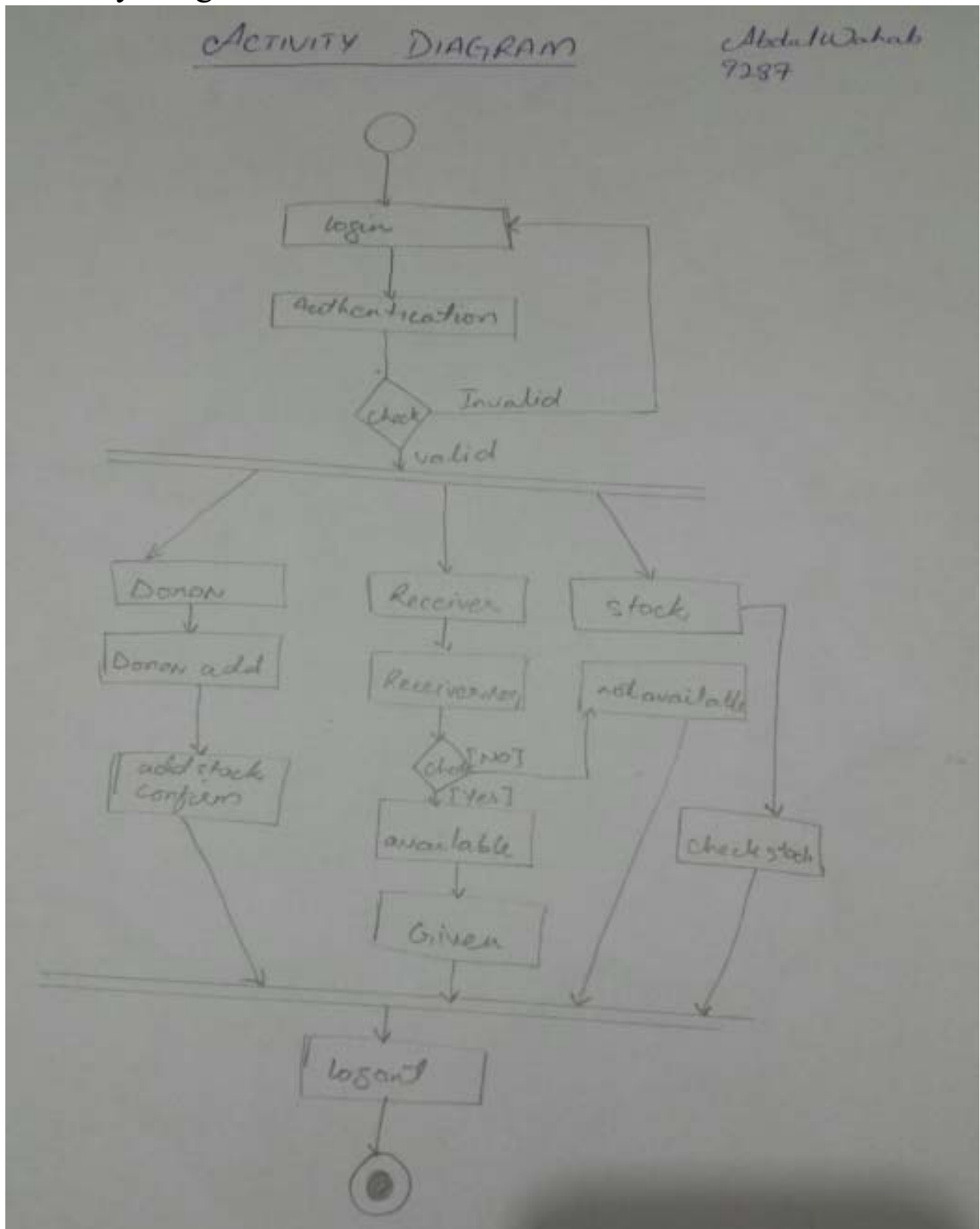
This could also be used as a part of Make Purchase.

Trigger:

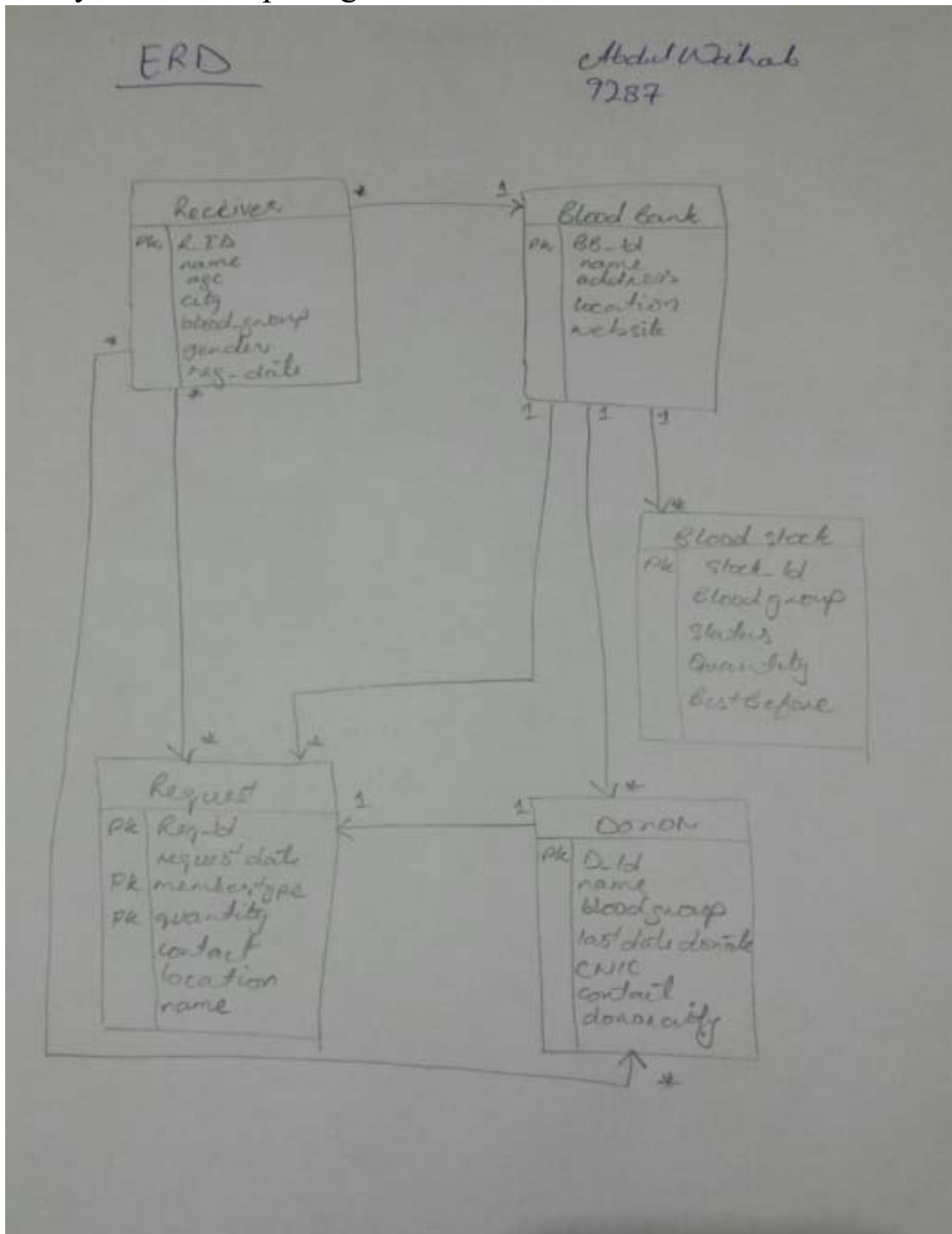
The Administrator decides to view their product inventory. Administrator:

1. The Administrator logs onto the Online furniture portal.
2. The Administrator enters his/her ID.
3. The Administrator enters his/her password.
4. The system displays all major function lists.
5. The Administrator selects the function from "Administrator Panel".
6. The Administrator perform "CRUD" operations.
7. The system displays the list of stock category, stock name, product price.
8. The Administrator update his profile, Receivers accounts.
9. The system displays the updated stock.

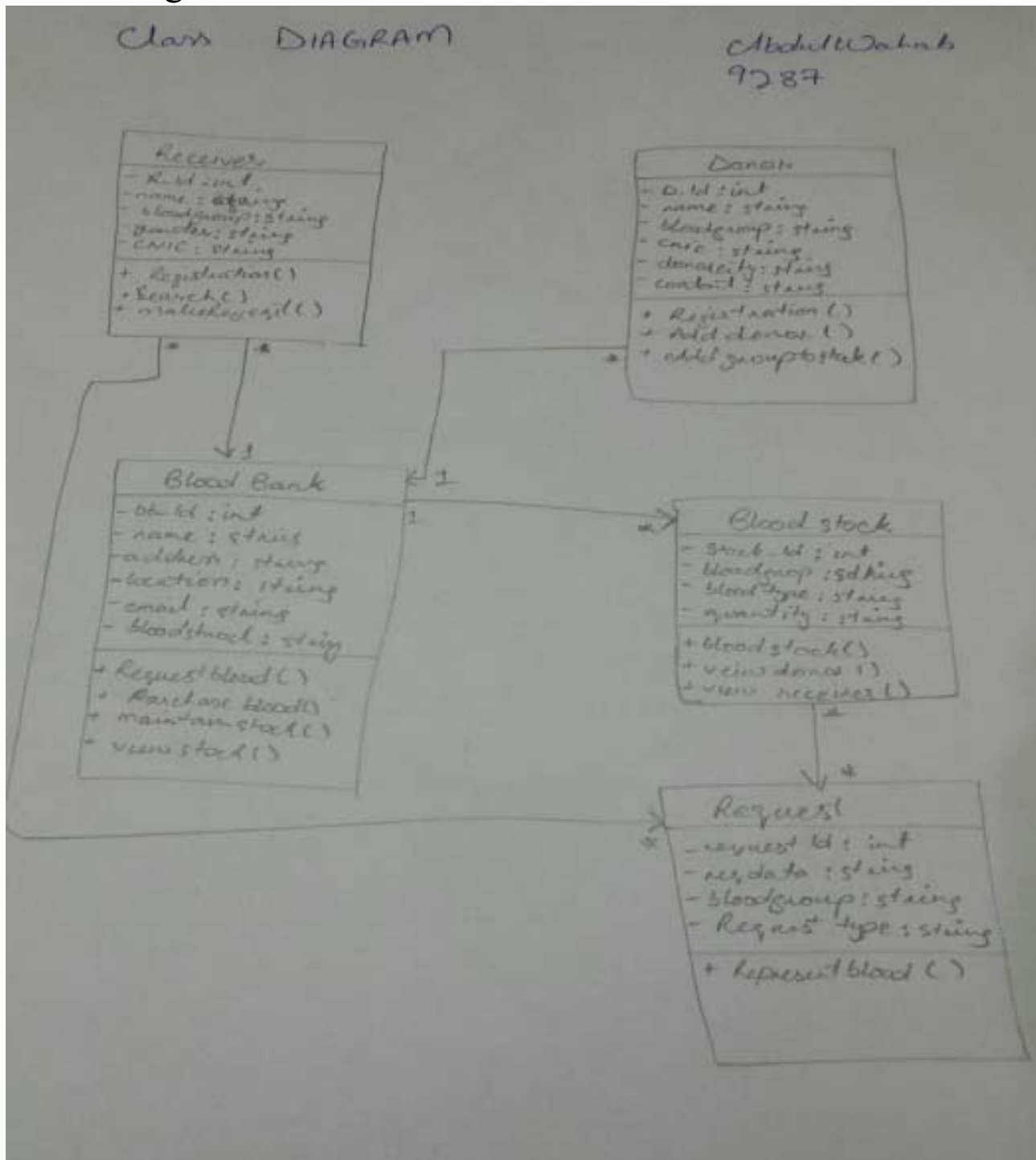
Activity Diagram:

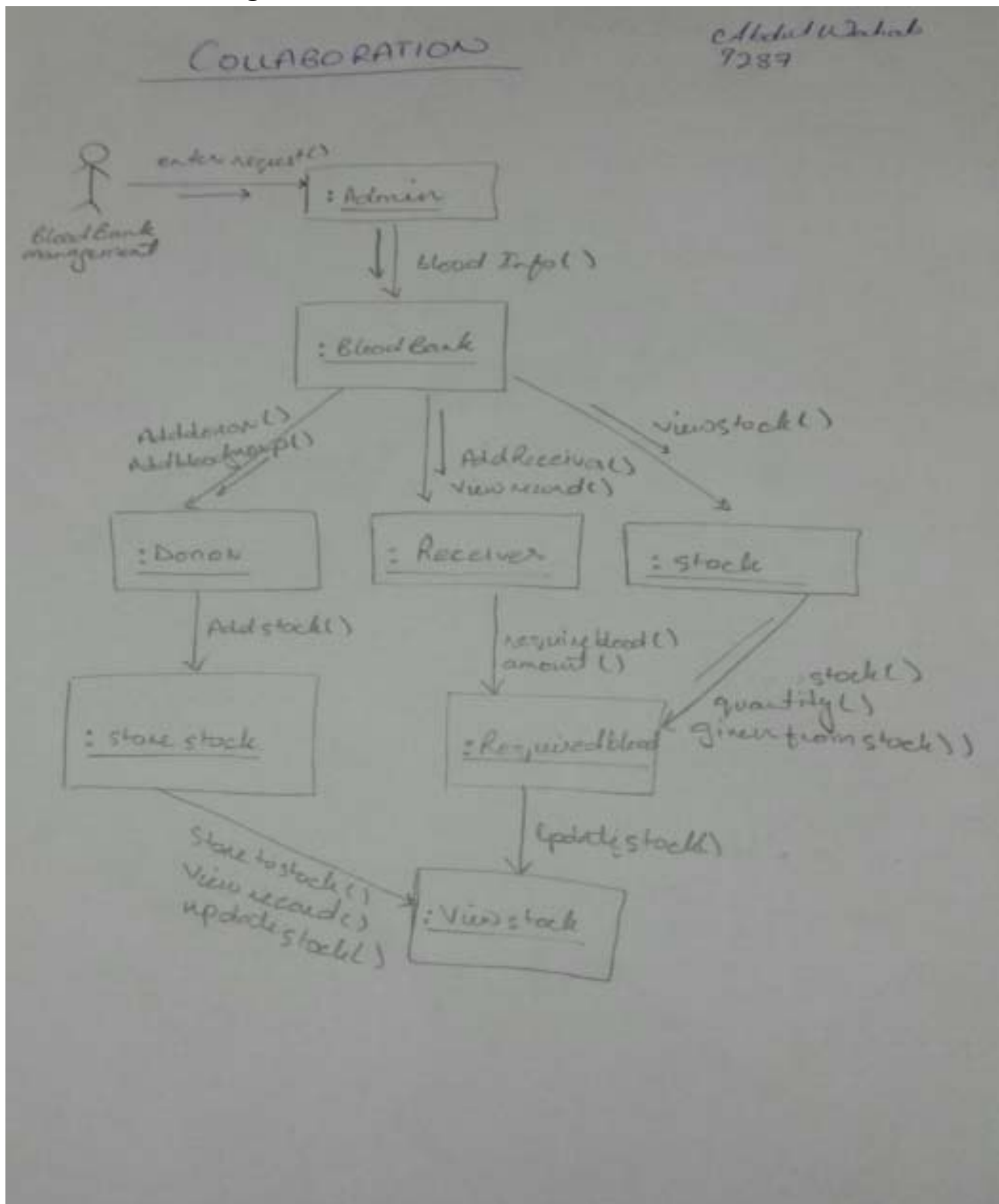


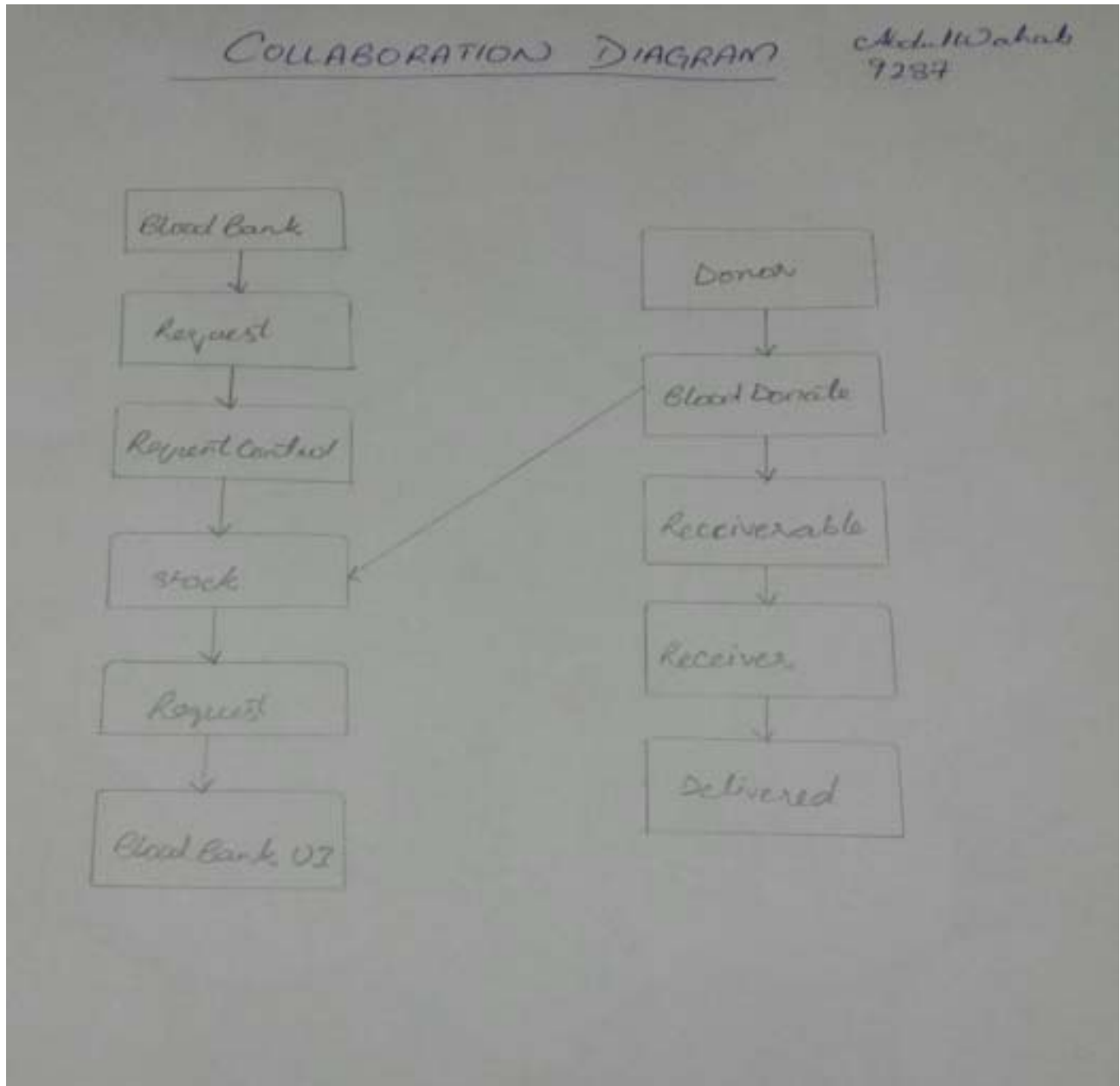
Entity Relationship Diagram:



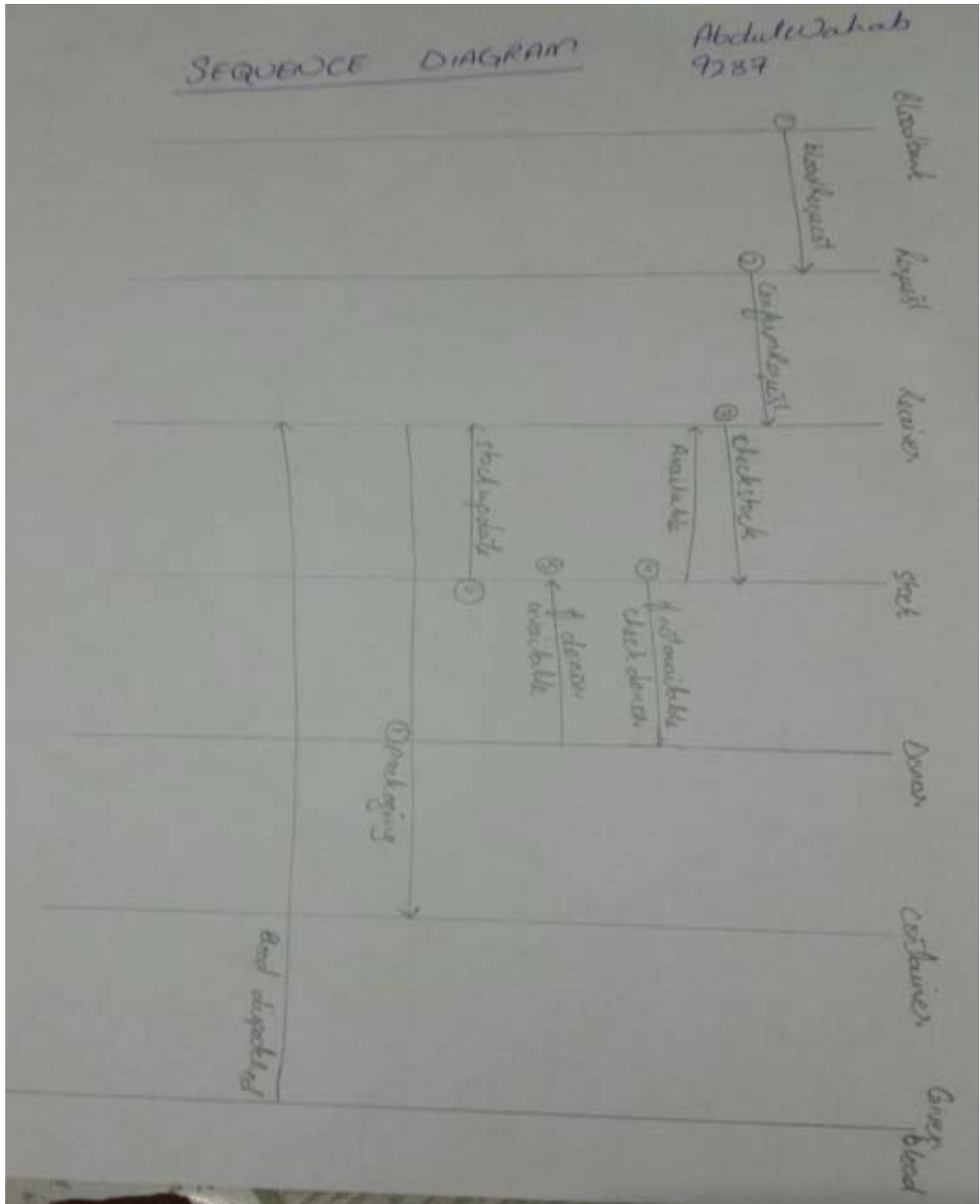
Class Diagram:



Collaboration Diagram:



Sequence Diagram:



State Chart Diagram:

