**COMSATS University Islamabad, Abbottabad Campus**

**Department of Computer Science**

**Project Proposal**

**Blood Donation Management System**

**CSC392 Object Oriented Software Engineering**

Submitted on: <Date&Time>

Group Members:

Waleed Khan (SP21-BSE-026)

Ahmed Bajwa (FA21-BSE-169)

Tayyab Rafique(SP21-BSE-024)

Table of Contents

[CHAPTER 1 PROJECT PROPOSAL 3](#_Toc101427096)

[Introduction 3](#_Toc101427097)

[Vision and Business Case 3](#_Toc101427098)

[Use-Case Model 3](#_Toc101427099)

[Supplementary Specification 3](#_Toc101427100)

[Glossary 3](#_Toc101427101)

[Risk List & Risk Management Plan 3](#_Toc101427102)

[CHAPTER 2 USE CASES 4](#_Toc101427103)

[Use Case Diagram 4](#_Toc101427104)

[Brief Level Use Cases 4](#_Toc101427105)

[Student Name 1 (Registration Number 1) 4](#_Toc101427106)

[Student Name 1 (Registration Number 1) 5](#_Toc101427107)

# CHAPTER 1 PROJECT PROPOSAL

## Introduction

Blood donation system is a system where different categories of people donate their blood To the needy person across the country. People will store their information on the system i.e. their mobile number, NIC number, blood type, medical history etc. and the system then scan for the appropriate person who fits the description and match the pair. The donor will be informed right after the blood has been donated. Also some verification will also be made on the person who needs blood incase of any misinformation provided by him to the system. The donor can set his data public or private. Incase he doesn’t want to be traced. Blood donation bank will also play an important role as all the blood will be stored in the bank.in case a person requires more blood then he will pay for the blood, the poor and needy person will be given free blood. In the end if the person had successfully recovered from an illness due to the blood donated to him then he can also give something (present)to the donor as a good gesture.

## Vision and Business Case

## Vision

The vision is To become the leading world class blood system in the innovation of new services, technology and research that positively impacts blood product safety and availability and enhances the live of our team members. The system makes the overall project management much easier and flexible.

***executive summary***

It is a blood donation management system which will serve human by automating the process of blood donation .It will consist of a website which will be connect to all blood donation center and a database to add blood, record blood donation history ,doner medical back ground as well as the patent record who are in need of blood in emergency cases or on regular bases(thalassemia patent) in addition to this there will be a reward system which will provide rewards for the doners

***the high-level goals and constraints***

|  |  |  |  |
| --- | --- | --- | --- |
| High-level Goals | Priority | Problems and Concerns | Current Solutions |
| Fast, user friendly, blood management system | High | Website link down in case of high use.  Lack of up to date accurate information from database and other blood donation centers due to non synchronization |  |

***the business case***

* Donor Management: Doner will donate blood which will be managed by blood donation management system
* Seeker Management: Seeker will request blood donation management system for a particular blood type which will be required by them
* Donation Management: this module will contain the list of donors along with the blood type that is donated by them.
* Stock Management: This module will deal with the all types of blood which are currently store in the repository of blood donation management
* Receiver List Management: this module deal with the all the person list that has successfully received the blood that was required by them.
* Reward Management: In this module it will manage the reward system of blood donation management.

## Use-Case Model

* Login: it will perform the local login function
* Registration: it will perform the local login function
* Manage System: it will manage the whole blood donation management system
* Donate Blood: Donate blood will enable the donor to donate blood
* Conduct Test: Conduct test function is a function through which donor will be able to test his blood before donation
* Request Blood: Request blood is a functionality of BDMS where a certain type of blood will be requested
* Accept Request : BDMS have a functionally of accepting blood request from a particular seeker
* Check Emergency: BDMS will gather circumstances through this feature under which blood is requested
* Check Donor List: This feature will enable BDMS to check donor list incase of emergency situation
* Check Stock: Check stock feature will enable BDMS to check the stock of blood bank
* Adding Receiver in list: This feature will be used to maintain the list of all the seeker that have received the blood in the database of BDMS
* Managing reward: this feature will manage the information of rewards that have been received by the donor

## Supplementary Specification

* Free blood must be given to the needy and poor person
* Donor and the seeker must be in range
* Text should be bold and clear
* Text must be visible from 1 meter.
* Result should be displayed within 30 seconds 90% of the time
* Text should be bold and clear

## Glossary

* Blood BDMS

## Risk List & Risk Management Plan

# Getting Doner and seeker in range: doner and seeker will be linked by identifying their current location

# Getting the right blood type to the right person: person will be identified through some verification process

# Getting authentic blood info: blood information will be verified by conduction blood test

# Verifying the donor information: donor information will be identified by the documents submitted by them

# Management of blood bank stock : blood bank stock should be kept up to date and manage by a data base system

# Management of rewards: rewards should be managed by process so that the right donor gets the reward

(Chapter 2 is pending)

# CHAPTER 2 USE CASES

## Use Case Diagram

<Paste your semester project diagram imported from CASE Tool i.e. starUML here>

## Brief Level Use Cases

### Waleed Khan (SP21-BSE-026)

#### Use Case 1: Check Donor list

The seeker ask for a particular blood type. Then the BDMS search for that particular blood the blood bank if it is not available then it will search in the other blood banks at the End the Admin will search in the Data base of the DBMS if no blood is found in the blood bank. The admin will then search for the same blood group type donor. the blood group type donor is found and he is also eligible to donate blood. The donor will be requested to donate blood . the donor will willingly donate blood at the near blood bank which will further be delivered to the seeker

#### Use Case 2: Donate Blood

Registered donor can willingly donate blood in the blood bank whenever they want . Some Donors are requested to donate blood incase of emergency. If the blood bank has not a particular blood and it is needed then the donor will be requested to donate it. The donor will go to the nearest blood bank available and donate blood.

Student Name 2 (Registration Number 2)

Student Name 3 (Registration Number 3)

Fully Dressed Use Cases

### Student Name 1 (Registration Number 1)

| Use Case UC1: Process Sale |
| --- |
| **Scope**: NextGen POS application  **Level**: user goal  **Primary** **Actor**: Cashier  **Stakeholders and Interests**:  - Cashier: Wants accurate, fast entry, and no payment errors, as cash drawer shortages are deducted from his/her salary.  - Salesperson: Wants sales commissions updated.  - Customer: Wants purchase and fast service with minimal effort. Wants easily visible display of entered items and prices. Wants proof of purchase to support returns.  - Company: Wants to accurately record transactions and satisfy customer interests. Wants to ensure that Payment Authorization Service payment receivables are recorded. Wants some fault tolerance to allow sales capture even if server components (e.g., remote credit validation) are unavailable. Wants automatic and fast update of accounting and inventory.  - Manager: Wants to be able to quickly perform override operations, and easily debug Cashier problems.  - Government Tax Agencies: Want to collect tax from every sale. May be multiple agencies, such as national, state, and county.  - Payment Authorization Service: Wants to receive digital authorization requests in the correct format and protocol. Wants to accurately account for their payables to the store.  **Preconditions**: Cashier is identified and authenticated. |

**Success Guarantee** (or Postconditions): Sale is saved. Tax is correctly calculated. Accounting and Inventory are updated. Commissions recorded. Receipt is generated. Payment authorization approvals are recorded.

**Main Success Scenario (or Basic Flow):**

1. Customer arrives at POS checkout with goods and/or services to purchase.
2. Cashier starts a new sale.
3. Cashier enters item identifier.
4. System records sale line item and presents item description, price, and running total. Price calculated from a set of price rules.

Cashier repeats steps 3-4 until indicates done.

1. System presents total with taxes calculated.
2. Cashier tells Customer the total, and asks for payment.
3. Customer pays and System handles payment.
4. System logs completed sale and sends sale and payment information to the external Accounting system (for accounting and commissions) and Inventory system (to update inventory).
5. System presents receipt.
6. Customer leaves with receipt and goods (if any).

**Extensions (or Alternative Flows):**

\*a. At any time, Manager requests an override operation:

1. System enters Manager-authorized mode.
2. Manager or Cashier performs one Manager-mode operation. e.g., cash balance change, resume a suspended sale on another register, void a sale, etc.
3. System reverts to Cashier-authorized mode.

\*b. At any time, System fails:

To support recovery and correct accounting, ensure all transaction sensitive state and events can be recovered from any step of the scenario.

1. Cashier restarts System, logs in, and requests recovery of prior state.
2. System reconstructs prior state.

2a. System detects anomalies preventing recovery:

* 1. System signals error to the Cashier, records the error, and enters a clean state.
  2. Cashier starts a new sale.

1a. Customer or Manager indicate to resume a suspended sale.

1. Cashier performs resume operation, and enters the ID to retrieve the sale.
2. System displays the state of the resumed sale, with subtotal.

2a. Sale not found.

* 1. System signals error to the Cashier.
  2. Cashier probably starts new sale and re-enters all items.

1. Cashier continues with sale (probably entering more items or handling payment).

**Special Requirements:**

- Touch screen UI on a large flat panel monitor. Text must be visible from 1 meter.

- Credit authorization response within 30 seconds 90% of the time.

- Somehow, we want robust recovery when access to remote services such the inventory system is failing.

- Language internationalization on the text displayed.

- Pluggable business rules to be insertable at steps 3 and 7.

- …

**Technology and Data Variations List**:

\*a. Manager override entered by swiping an override card through a card reader, or entering an authorization code via the keyboard.

3a. Item identifier entered by bar code laser scanner (if bar code is present) or keyboard.

3b. Item identifier may be any UPC, EAN, JAN, or SKU coding scheme.

7a. Credit account information entered by card reader or keyboard.

7b. Credit payment signature captured on paper receipt. But within two years, we predict many customers will want digital signature capture.

Frequency of Occurrence: Could be nearly continuous.

**Open Issues:**

- What are the tax law variations?

- Explore the remote service recovery issue.

- What customization is needed for different businesses?

- Must a cashier take their cash drawer when they log out?

- Can the customer directly use the card reader, or does the cashier have to do it?

**Screen Shots:**

<Paste the images from netbeans here for the relevant screen>