**WCOMSATS University Islamabad, Abbottabad Campus**

**Department of Computer Science**

**Project Proposal**

**Blood Donation Management System**

**CSC392 Object Oriented Software Engineering**

Submitted on: <5-2-23>

Group Members:

Waleed Khan (SP21-BSE-026)

…

Table of Contents

[CHAPTER 1 PROJECT PROPOSAL 3](#_Toc126935057)

[Introduction 3](#_Toc126935058)

[Vision and Business Case 3](#_Toc126935059)

[the high-level goals and constraints 3](#_Toc126935060)

[Business case 4](#_Toc126935061)

[Executive Summary 4](#_Toc126935062)

[Use-Case Model 4](#_Toc126935063)

[Supplementary Specification 4](#_Toc126935064)

[Logging and Error Handling: 4](#_Toc126935065)

[Security: 4](#_Toc126935066)

[Usability 4](#_Toc126935067)

[Human Factors: 4](#_Toc126935068)

[Reliability 5](#_Toc126935069)

[Recoverability 5](#_Toc126935070)

[Supportability 5](#_Toc126935071)

[Performance 5](#_Toc126935072)

[Glossary 5](#_Toc126935073)

[Risk List & Risk Management Plan 5](#_Toc126935074)

[CHAPTER 2 USE CASES 5](#_Toc126935075)

[Use Case Diagram 5](#_Toc126935076)

[Use Cases Distribution 5](#_Toc126935077)

[Brief Level Use Cases 6](#_Toc126935078)

[Student Name 1 (Registration Number 1) 6](#_Toc126935079)

[Student Name 1 (Registration Number 1) 7](#_Toc126935080)

# CHAPTER 1 PROJECT PROPOSAL

## Introduction

BDMS (Blood donation management system) is a computerized application which is used to Manage the blood donation i.e.it handles blood donated to seeker, blood stored in blood bank and reward to the donor for donating the blood. It includes hardware components such as computer along with the software on which the system will run.

## Vision and Business Case

We envision a high quality fault-tolerant blood donation application, BDMS, with the flexibility to enable the donor to donate blood and the seeker to get blood, multiple terminal and user interface mechanisms, and integration with multiple third-party supporting system.

### the high-level goals and constraints

|  |  |  |  |
| --- | --- | --- | --- |
| High-level Goals | Priority | Problems and Concerns | Current Solutions |
| Secure and reliable blood donation point | High | Donor feels that the blood may not get to the right person hence he feels insecure and do not trust the blood donation point | BDMS will make the donation point secure and reliable by showing proof to the donor where and to whom his blood is being donated. |
| Giving blood to the deserving seeker | High | Usually the person who is in need of blood does not get the blood as the blood that is required by the seeker is in high demand and there is a high possibility that the blood given to the seeker may be less needy, | BDMS will have mechanism through which blood will be given to the most needy seeker. |
| Preserving blood in a cool and safe environment | High | Blood preserving is a quite hectic job. A cool place is needed to store the blood | Blood will be stored in blood banks, keeping the blood in a cold place |
| Keeping the blood being genuine and reliable | High | Donor donates blood before four months which is not reliable.  Sometimes blood is not genuine due to genetic and other problems | Donor will be alerted if he apply to donated blood before 4 months. Blood after donation will be check for reliability |

### Business case

Existing BDMS are not adaptable in terms of varying business rules and varying network designs. In addition, they do not scale well with the large amount of data I.e., increase in number of bloods donated, donor and seeker. Which becomes hectic when it comes to managing the data. also, there is a chance of getting blood from an unreliable source. there is also trouble getting the blood because some bloods are very rare and hence cannot be get for free. There is marketplace dissatisfaction with this inflexible state of affairs, and demand for a reliable BDMS that rectifies this.

### Executive Summary

BDMS deals and manage the blood that is being donated along with the data of donor and seeker and the blood bank in which it will be stored. The vision behind the system is the unsatisfaction of the users of the existing BDMS like getting an unreliable blood or getting blood from an unreliable source. Rare bloods are hard to get mostly the person who is willing to pay gets it which should not be the case. All these issues will be tackled by our visionary BDMS.

## Use-Case Model

**Login:** login will allow the user the access the BDMS and all its available services

**Registration:** The new user should be registered first to access all BDMS available services

**Donate Blood:** Donor will be able to donate blood along with some constraints

**View Donors:** Admin will be able to see top donors on his UI

**Give Reward:** seeker after receiving blood will be able to give reward to the donor. (reward can be of different types. Reward can also be given through third party)

## Supplementary Specification

# Logging and Error Handling:

Logging and Error Handling

# Security:

All the usage will require user authentication

# Usability

# Human Factors:

The user will be able to see a large-monitor display of the BDMS. Therefore:

* Text should be easily visible from 1 meter.
* Avoid colors associated with common forms of color blindness.

# Reliability

# Recoverability

Data should be recoverable in case of system failure

# Supportability

# Performance

authorization should take no longer than 30 seconds

## Glossary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Term | Definition and Information | Format | Validation Rules | Aliases |
| BDMS | Blood Donation Management System |  |  |  |

## Risk List & Risk Management Plan

* **Getting Doner and seeker in range**: getting the matching donor i.e. the blood required by the seeker in the closest range as possible
* **Getting the right blood type to the right person:**  making sure that the person getting the blood is the person the most deserving
* **Getting authentic blood:** blood that is being donated should be authentic
* **Verifying donor:** Every donor should be verified so the reliability of blood can be measure
* **Managing blood:** blood should be preserve in a most suitable environment.

# CHAPTER 2 USE CASES

## Use Case Diagram

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

## Use Cases Distribution

|  |  |  |
| --- | --- | --- |
| S#. | Group Member | Assigned Use Cases |
| 1 | <Waleed Khan>  <SP21-BSE-026> | UC 1  UC 2 |
| … | … | … |

## Brief Level Use Cases

### Student Name 1 (Registration Number 1)

#### Use Case: Process Sale

A customer arrives at a checkout with items to purchase. The cashier uses the POS system to record each purchased item. The system presents a running total and line-item details. The customer enters payment information, which the system validates and records. The system updates inventory. The customer receives a receipt from the system and then leaves with the items.

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>