Project Report

Def-ny

@se2011_19T3

```
|                   |
<     x      x      >
 \       c        /
   \            /
     \  V~V   /
      \'    '/  _  _ _  _  .  _ _
       '\  /' (_| | | | |_) | | (/_
        ' ' '           |
        ' ' '           '
          '             '
```

#define Blood "Love";

-Blood Vlad…

Joshua King z5165228

Muhammad Al Waliy Ismail z5177103

Mohamed Muqaddis z5143739

Mohammad Faiz Ather z5170340

Executive Summary

Handling of blood has been something crucial in every Medical Facility. Hospitals, Clinics and Pathology has been using paper-based system for ages.The inefficiency of this system has led to various kinds of problems such as blood going out-of date before it can be used, running out of blood, poor record keeping and poor donor service.

We came out of a computerized software solution that will handle all blood supplies of all Medical Facility registered with us. Our system will ensure that all facilities will never runs out of blood, minimal blood wastage and an efficient record keeping. All facilities will have their blood stored in their own storage but we will ensure that whenever the blood falls below the minimum storage quantity, blood from other facilities with a higher blood quantity is transferred to low blood facilities.

We will also make sure batmobile is sent out to collect and deliver blood if the majority of facilities have small blood quantities. The system tries to maintain an equal balance of blood among all the facilities. Furthermore, we will make sure that all facilities will be using the older blood first before using the newly added blood, this will reduce the blood wastage. Each facility will only need to send a request to the system to use and add blood. The rest will be handled by the system.

Our blood management system has undergoes robust verification to ensure that all of our functionality works correctly. For example, we will verify the sorting algorithm for the bloods so that every expired blood is removed and older blood will be used first. Next, we will verify all requests that are made upon us such as addition of blood, removal of blood and so on.

Requirements

1. Vampire

a. Knowledge of blood in all facilities (High)
  i. Query blood storage details (Use case 7)
    1. Specify parameters
      a. Blood type
      b. Location
      c. Donor available in blood types
  ii. Add Medical Facility (Hospital/Clinic/Pathology) (Use case 9)
    1. Specify details.
      a. Address
      b. Usage
        i. Must provide High, Medium or Low usage.
        ii. Set Minimum amounts of Supplies (Each type of Blood)
      c. Storage
        i. Capacity - Amount of blood it can contain

b. Notify expired blood removal (High) (Use Case 1) (#Verification)
  i. The system must generate a notification for the facility to remove expired blood.
  ii. The system must update the record (on confirmation).

c. Handle Request for Blood (High)
  i. Use Blood (Use Case 2)
    1. Vampire must specify whereabouts of Blood.
    2. Vampire must use old Blood first
    3. Vampire must make sure expired Blood is not being used.
    4. Vampire must Remove Blood from Record.
  ii. Add Blood (Use Case 3)
    1. Vampire must specify which storage the blood should be stored
      a. Blood
        i. Must provide Quantity.
        ii. Must provide Type.
        iii. Must provide Collection Date.
        iv. Vampire must assign Blood ID
        v. Vampire must update the Record of Non-Verified Blood
  iii. Verify Blood - Pathology (Medium)
    1. Vampire must specify where to store **Tested** Blood.

      2. Vampire must remove the Blood from **Non-Tested** and update the **Tested** Blood
   iv. As Vampire keeps track of all blood at a facility, organizing them by type and bag size, the facility will request to use their own blood supply to the system, the system will provide the details (i.e. according to types, sizes and expiry date) of storage room number, fridge, shelf, section, pick from front as new blood added to the back of section.
d. Managing a balance (High) (Use case 5, 6) (#Verification)
   i. Manage High Scores for Hospital/Clinic
      1. Calculate score of each facility with respect to amount of blood about to expire of each type left as well as incorporating scores of facilities according to proximity to other facilities. E.g a facility with low blood supply close to a facility with high supply but low usage will have a higher score compared to a facility with low supply far away from any facility with high supplies (supply and usage).'
      2. The Vampire will pool up all resources and dividing among the most needy and how far by dividing every facility into seeders and leechers.
   ii. Pathology (Low)
      1. The score calculation for pathology will be different, in that we would want pathology to have low levels of supply so that if supply exceeds say for example 50% of storage capacity or blood starts getting old so score would decrease triggering a system alert to order distribution of blood according to the score of medical facilities.
      2. The system will notify the pathology that the batmobile is on its way to suck 🧛‍♀️ all blood from pathology, then it will move the blood to health facilities.
   iii. Blood Supplies go below threshold (High) (Use case 10)
      1. The Vampire continuously monitors blood supplies, if a Facility's supplies fall below a set threshold then,
         a. The Score will drop down drastically putting the facility with dangerously low supplies on the top list for recipient. (High)
            i. The system will take from the Facility with High Scores and distribute to facility with low score(Lowest first). (High)

ii. If there is no facility who can be classified as Provider (Low)
    1. System will generate alert for Batmobile to collect donations
    2. Donors on the System will get notified for short supply and be asked to donate to the nearest Collection unit.

2. The vampire will make sure no facility runs out of any blood type so that it will able to cater to any patient with even special blood type such as O-neg who can only accept O neg blood.

e. Handle Donor Request (Low)
    i. Registration(Medium)
        1. The system should be able to register public personnel who are willing to donate. The contact details of the donors are acquired.
    ii. Notification
        1. The system should notify the donors when there is a batmobile closer to their location.
    iii. Query
        1. Schedule
            a. Vampire must provide Schedule of Donation Drives/Batmobile Drives location and times
        2. Location
            a. Vampire must provide information about nearest Donation Drive according to Donor location
        3. Donation Conditions
            a. Vampire must provide list of health requirements the Donor must fulfil to be able.

f. Generate notification batmobile (Low)
    i. The system should Generate notification for batmobile to collect blood from the public when there is less quantity of blood available in the system.
    ii. The system should Generate notification the batmobile the pickup and dropoff of blood according to demands(i.e. Score).

g. Handle Batmobile Request (Low)
    i. The Vampire must generate ID Tags for Blood
        1. Generate according to,
            a. Registered Donor
                i. Vampire must generate Tag from supplied Donor ID

            - b. Non Registered Donor
                - i. Vampire must register Donor from information from Batmobile
    - ii. The Vampire must handle delivery request
        - 1. The Vampire must provide Pathology information according to batmobile location

2. Medical Facility (Hospital/Clinic)

- a. Request Vampire to use facility blood supply (High) (Use case 2)
    - i. Medical Facility can make request for blood
        - 1. Medical Facility must specify type.
        - 2. Medical Facility must specify quantity required.
- b. Request Vampire to add Storage Units (Medium) (Use case 8)
    - i. Medical Facility can make request to add Storage Units.
        - 1. Storage Unit Details (Low)
            - a. Room Capacity
            - b. Section Capacity
            - c. Fridge Capacity
            - d. Shelf Capacity
- c. Add Blood to facility (High)
    - i. Must provide Quantity.
    - ii. Must provide Type.
    - iii. Must provide Whereabouts(Room, Section, Fridge and Shelf) of blood.
    - iv. Must provide Collection Date.
    - v. Vampire must assign Blood ID
    - vi. Vampire must update the Record
- d. Query about blood available (Medium)
    - i. Medical facility can query the system based on the blood type
    - ii. Medical facility can query about the number of bags available in each blood type
    - iii. Medical facility can query about the total quantity of blood and total quantity of each blood type in the system
- e. Register donor to the system (Low)
    - i. The medical facility should be able to register public personnel who are willing to donate.
    - ii. The contact details of the donors are acquired.
- f. Can query about donor (Low)
    - i. Query all the donors registered with the system
    - ii. Query donors based on blood group

3. Medical Facility (Pathology)

a. Request Vampire to add/update blood to facility supply of **Non-Tested** (Low)

   i. Pathology can request Vampire to add Blood

      1. Pathology must specify Type.
      2. Pathology must specify Quantity.
      3. Pathology must specify Collection Date.

   ii. Request Vampire to Update Blood to facility supply of **Tested** (Low)

      1. Pathology must specify Blood ID.

4. Donors

a. Register with the system (Medium) (Use Case 4)

   i. The donors should be able to register with the system by providing their basic details in order to donate blood.

b. Notified about batmobile (Low)

   i. The donor should get notifications when there is a batmobile next to their area.

c. Donate blood in medical facilities or batmobile (Low)

   i. Donor can donate blood

      1. Donor must register with system through batmobile
      2. The donors must provide Donor ID if already registered with system

d. Enquire about donating blood (Low)

   i. The donors can enquire the system about when and where they can donate blood.

e. Enquire about health requirements (Low)

   i. The public should be able to enquire the system about the health condition requirements to donate blood.

5. Batmobile

a. Deliver blood (Medium)

   i. The Batmobile can see of notifications to visit a list of medical facilities for collection of blood, and view list of recipient facilities determined by Vampire.

b. Go out to collect blood (Low)

   i. The Batmobile must view emergency notification for drives to collect blood from donors.

c. Register donor (Low)

   i. The Batmobile must request Vampire to generate ID Tag for blood bag.

      1. Registered Donor

        a. Batmobile provides Donor ID to Vampire
    2. Non-registered Donor
        a. The batmobile should allow users to register with the system.
d. Send the blood to pathology (Low)
    i. Batmobile requests blood delivery address.
        1. Batmobile must provide Location.

Use-Cases

Use Case 1 (1.b)	Remove Expired Blood
Actor	Hospital
Precondition	The system has notified the hospital that they have expired blood
Basic Flow	The hospital is given a list of IDs of blood bags that have expired. The bags are removed and safely thrown away. The hospital confirms with the system that the correct blood has been removed.
Postcondition	The expired blood is no longer in the hospital

Use Case 2 (1.c.i)	Using blood
Actor	Hospital
Precondition	Blood is available for the hospital to use
Basic Flow	The hospital requests blood by type and amount from the system. The system finds the best-scoring blood (by location, type, amounts), and reserves it for the hospital. If the blood is from a different location, it delivers the blood to the hospital. The system returns the ID(s) of the blood that the hospital can use. The hospital then locates and uses the blood. The hospital confirms with the system what blood it used. The system removes those blood bags from the system.
Postcondition	The blood has been removed from the system

Use Case 3 (1.c.ii)	Receiving donated blood
Actor	Hospital
Precondition	The blood has been tested and all the details of the blood are recorded
Basic Flow	If the donor is not registered, the hospital registers the blood type and other details of the donor. The hospital submits the blood details to the system.
Postcondition	New blood has entered the system

Use Case 4 (4.a)	Register Donor
Actor	Hospital/Batmobile
Precondition	A donor is donating blood and has not been registered before, nor has blood testing pending.
Basic Flow	The basic details (name, age, address etc.) are recorded by the hospital/clinic. The blood is sent off for testing. After testing the details of the blood (type, diseases) are recorded on the profile. The donor is also asked if they would like to register for notifications for future blood drives in their area.
Postcondition	The donor is registered

Use Case 5 (1.d)	Take Blood From Location
Actor	Hospital
Precondition	The hospital has enough of the required blood
Basic Flow	Blood is taken from the hospital. The blood's location is marked as in transit and taken to the new location.
Postcondition	The blood is marked as in transit

Use Case 6 (1.d)	Deliver Blood To Location
Actor	Hospital
Precondition	The hospital has enough storage for the new blood and the blood is marked as in transit
Basic Flow	Blood is delivered to the hospital. The hospital accepts the blood and notifies the system that the transaction has taken place. The system updates the details of the blood to include the new location.
Postcondition	The blood is assigned to the new location

Use Case 7 (1.a.i)	Queries about blood
Actors	Hospital
Precondition	NA
Basic Flow	The hospital sends a query to the system regarding blood. The system executes the query and returns the information requested.
Postcondition	NA

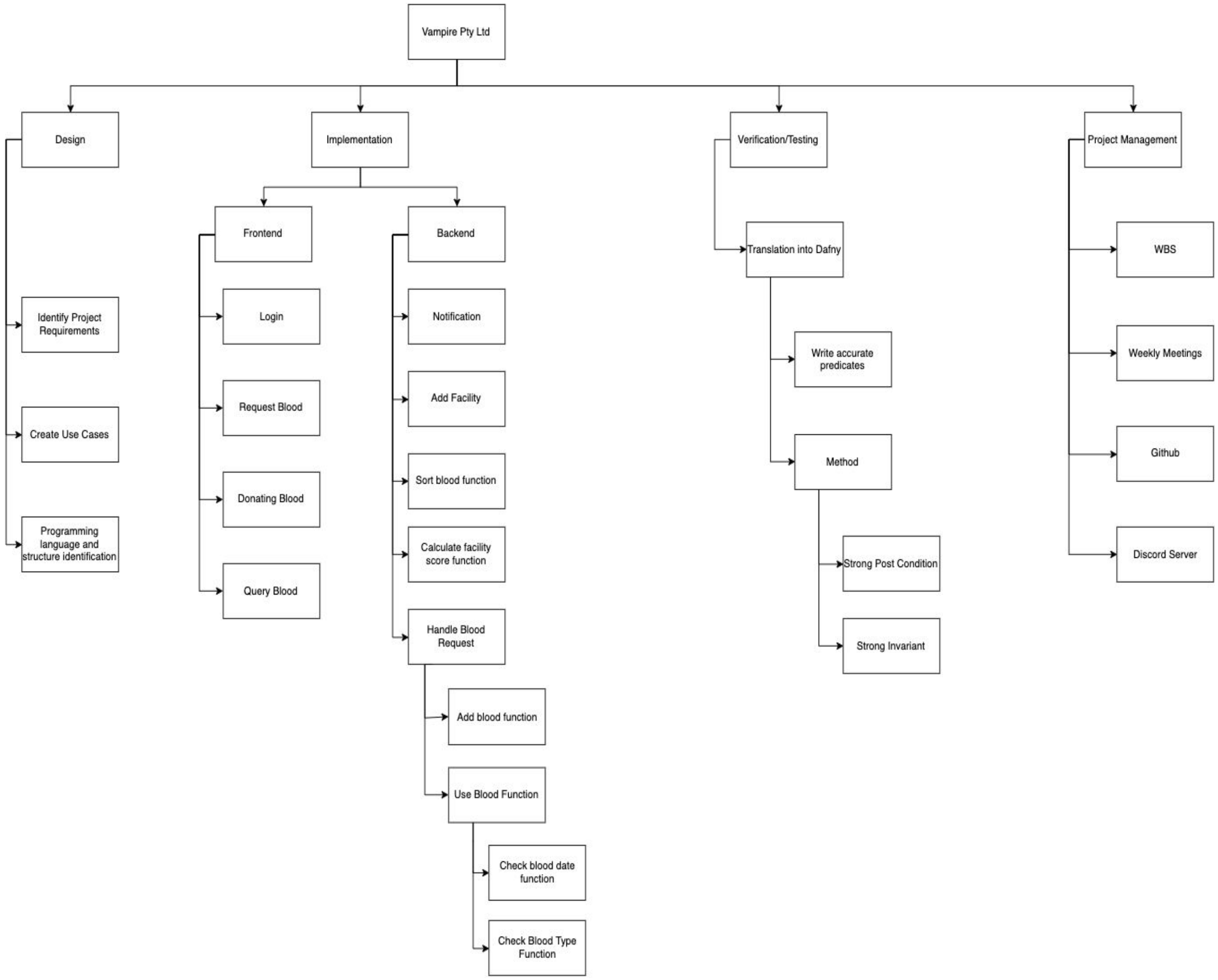
Use Case 8 (2.b)	Update Storage Info
Actor	Hospital
Precondition	The new update does not reduce the storage below the amount of blood the hospital is currently storing
Basic Flow	The hospital notifies the system that the amount of blood it can store has changed. The system updates the storage capabilities of the hospital to the new storage amount.
Postcondition	NA

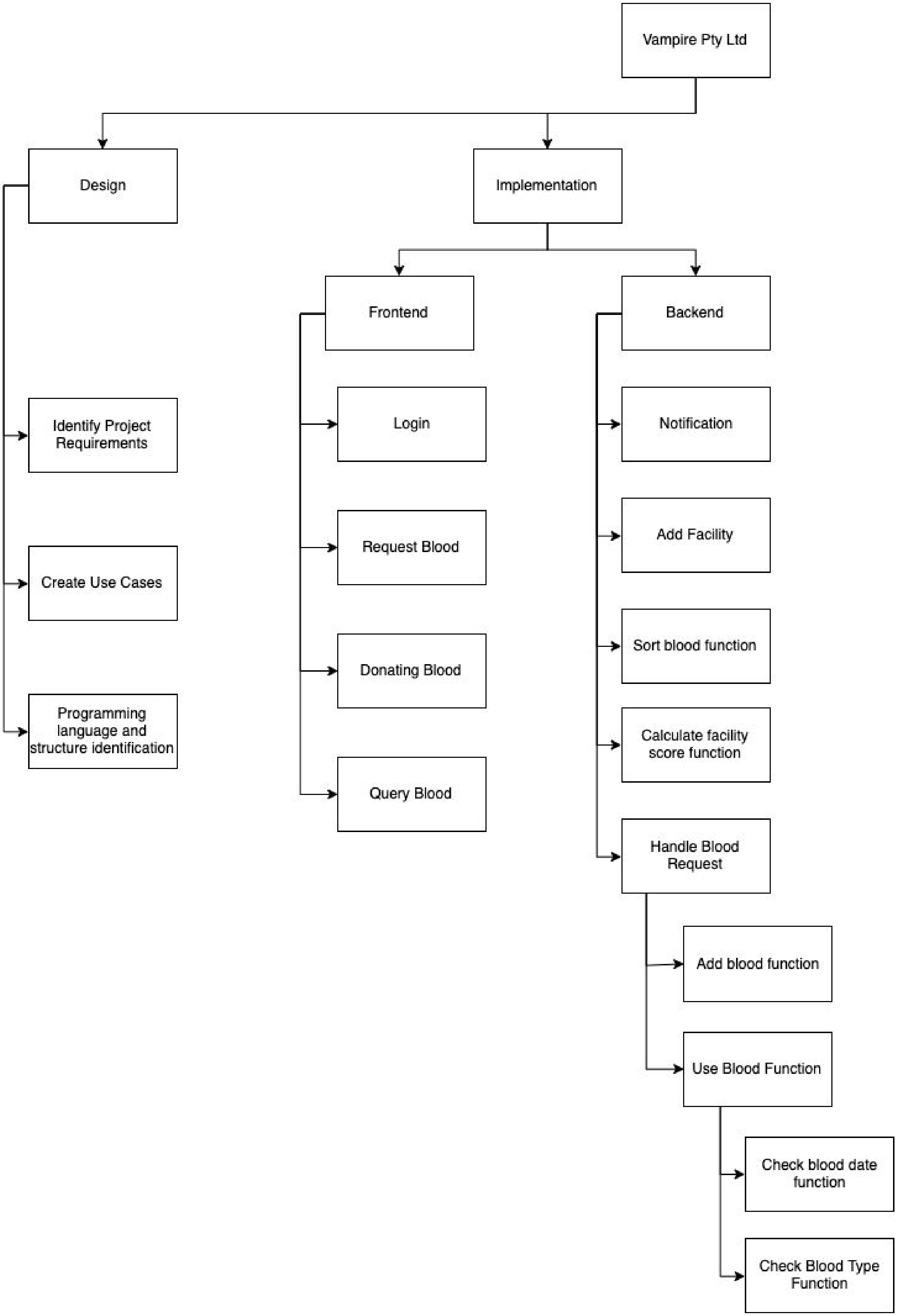
Use Case 9 (1.a.ii)	New Hospital
Actor	Vampire
Precondition	The new location does not conflict with any of the exist ing ones
Basic Flow	The details of a new hospital are sent to the system. The new details are entered into the system and a new location is created.
Postcondition	A new location is in the system

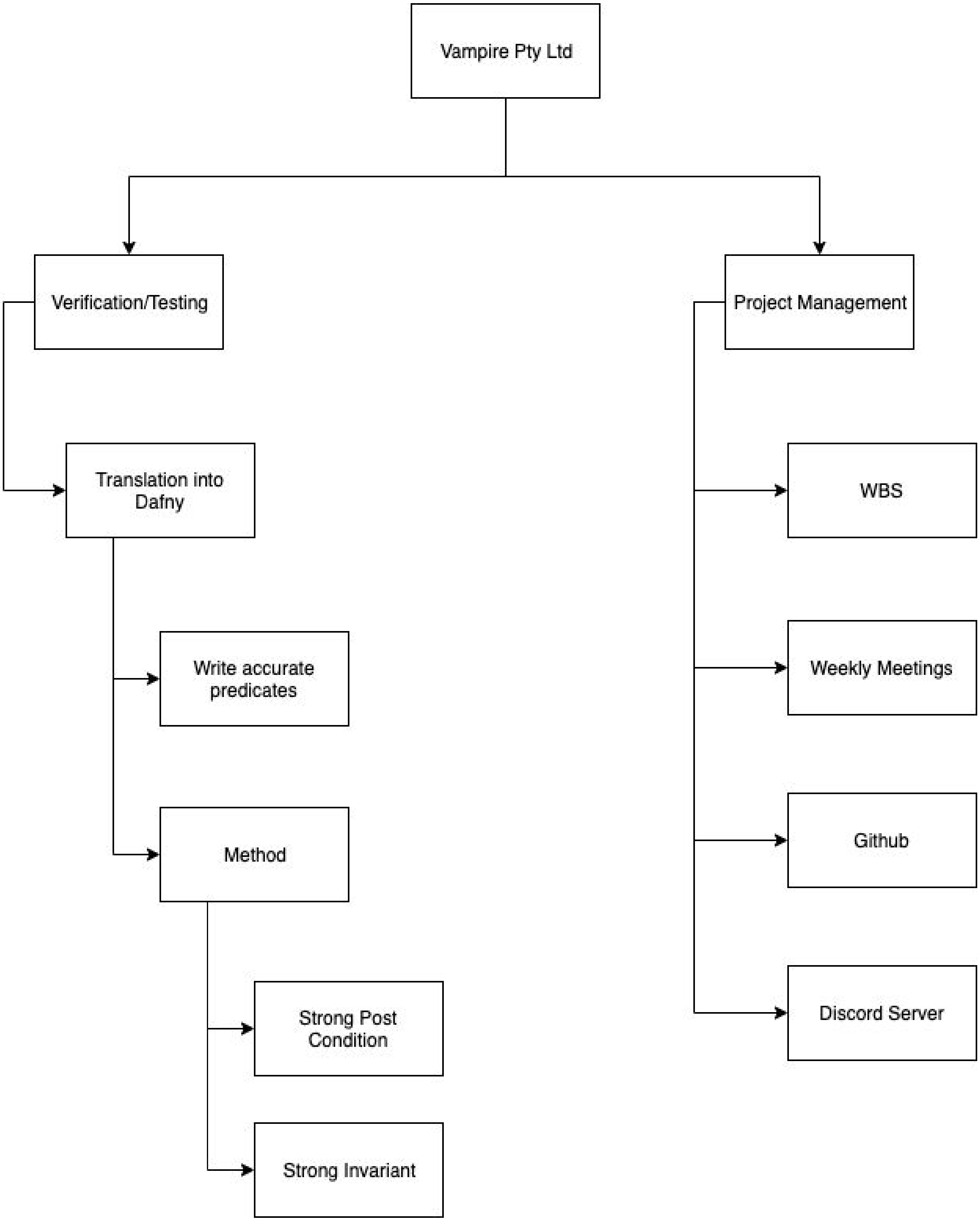
Use Case 10 (1.d.iii)	Blood falls below threshold
Actor	Hospital
Precondition	The amount of blood at a location has fallen below a set threshold
Basic Flow	If nearby hospitals are still above the threshold then blood is delivered from the hospitals with the most blood to this hospital until it is a set amount or more above the threshold. This is done without lowering other hospitals below the threshold. If there are no nearby hospitals that are valid, then the Batmobile is sent out on an emergency blood drive to collect more blood to be delivered to that hospital. Until it rises above the set amount above the threshold, the hospital is marked as priority for delivery of blood.
Postcondition	The hospital is either prioritised or has risen above the set amount above the threshold.

Work Breakdown Structure

Draw the Work Breakdown Structure (WBS): a graph showing how the project can be broken down into component builds. The coding is planned and managed using the WBS.

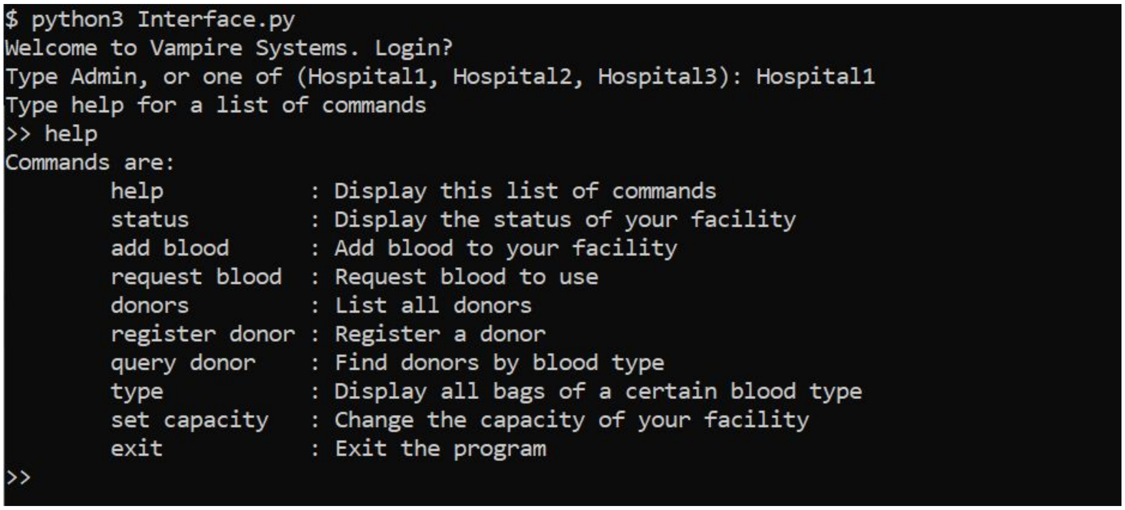



Vampire Pty Ltd
Design
Implementation
Frontend
Backend
Identify Project Requirements
Create Use Cases
Programming language and structure identification
Login
Request Blood
Donating Blood
Query Blood
Notification
Add Facility
Sort blood function
Calculate facility score function
Handle Blood Request
Add blood function
Use Blood Function
Check blood date function
Check Blood Type Function


Vampire Pty Ltd
Verification/Testing
Project Management
Translation into Dafny
Write accurate predicates
Method
Strong Post Condition
Strong Invariant
WBS
Weekly Meetings
Github
Discord Server

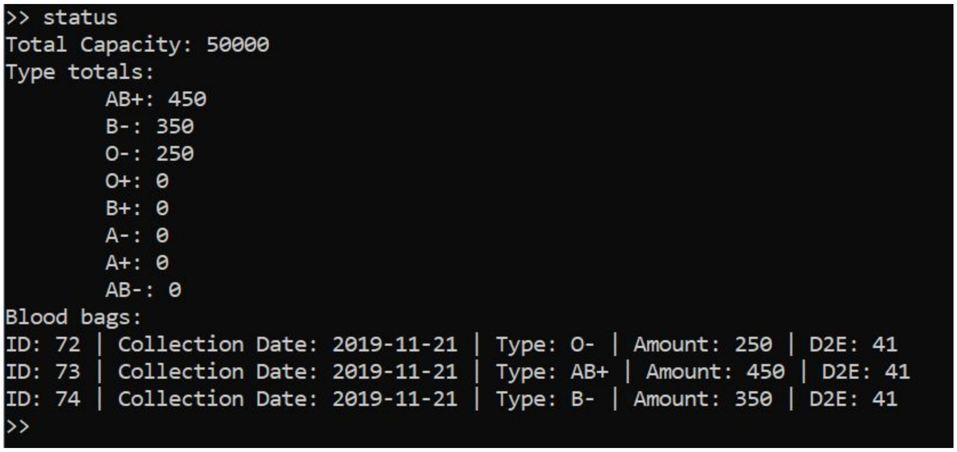
Front-End Implementation

The front-end interaction of our project takes place through a simple command line interface, with the ability to log in as either an Admin or as one of the Medical facilities. Medical facilities are able to view and interact with their blood supplies through several commands:



```
$ python3 Interface.py
Welcome to Vampire Systems. Login?
Type Admin, or one of (Hospital1, Hospital2, Hospital3): Hospital1
Type help for a list of commands
>> help
Commands are:
        help           : Display this list of commands
        status         : Display the status of your facility
        add blood      : Add blood to your facility
        request blood  : Request blood to use
        donors         : List all donors
        register donor : Register a donor
        query donor    : Find donors by blood type
        type           : Display all bags of a certain blood type
        set capacity   : Change the capacity of your facility
        exit           : Exit the program
>>
```

Facilities can view their blood supplies with the status of their supplies, as well as the individual blood bags and their details, such as expiry date or blood type.

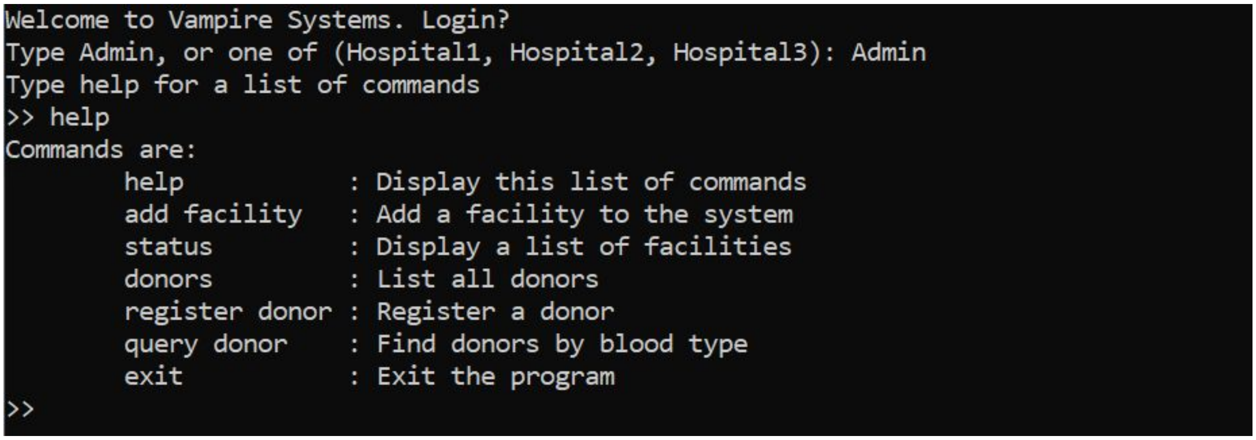


```
>> status
Total Capacity: 50000
Type totals:
        AB+: 450
        B-: 350
        O-: 250
        O+: 0
        B+: 0
        A-: 0
        A+: 0
        AB-: 0
Blood bags:
ID: 72 | Collection Date: 2019-11-21 | Type: O- | Amount: 250 | D2E: 41
ID: 73 | Collection Date: 2019-11-21 | Type: AB+ | Amount: 450 | D2E: 41
ID: 74 | Collection Date: 2019-11-21 | Type: B- | Amount: 350 | D2E: 41
>>
```

Users can add blood to their capacity by giving the type and amount. Users can also then request blood by giving a blood type and an amount, and will return an available blood bag and remove it from the existing facility. If there is no available blood bag, a message saying so will be displayed. Users can also search for blood bags by type or update their own capacity. They can also view and register donors.

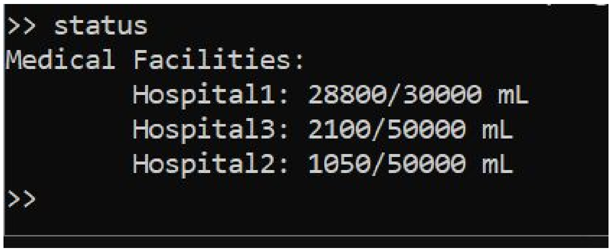
When expired blood is removed from the facility, a corresponding message is shown to the user.

Admins have a slightly different interface, which does not allow them to add or remove blood.



```
Welcome to Vampire Systems. Login?
Type Admin, or one of (Hospital1, Hospital2, Hospital3): Admin
Type help for a list of commands
>> help
Commands are:
        help           : Display this list of commands
        add facility   : Add a facility to the system
        status         : Display a list of facilities
        donors         : List all donors
        register donor : Register a donor
        query donor    : Find donors by blood type
        exit           : Exit the program
>>
```

They are still able to register and query donors, but the status command instead shows the blood levels and capacities of all the medical facilities in the system.



```
>> status
Medical Facilities:
        Hospital1: 28800/30000 mL
        Hospital3: 2100/50000 mL
        Hospital2: 1050/50000 mL
>>
```

Finally, admins are able to register new medical facilities by giving the name and capacity of the new facility.

Back-End Implementation

General Description:

The System is implemented in an Object Oriented fashion in Python 3.

=Class Blood

- CollectionDate, Amount, Type, isVerified, Weight ⇒ Attributes

-- Takes a String of Date converts to DateTime object

-- Calculation of Expiry Date using isExpired() returns the number of days to expiry or TRUE if expired already

=Class Medical_Facility

- Name, Capacity(*), Weight ⇒ Attributes

=Class Capacity

- Maximum_Capacity, Storage ⇒ Attributes

-- Transfer (*refer Allocation, Distribution*)

=Class Storage

- Inventory(**), BloodTypes ⇒ Attributes

-- Add Blood to Storage

-- Remove Blood

(*) Pizza - Radius, (**) Each Slice of The Pizza

=Class Efficiency

- CompatibleType, Contributor , BloodRank ⇒ Attributes

-- Calculate Score for each Blood (*refer Request Blood*)

-- Sorting Blood, init blood weight then calculate weight then sort by weight. Choose best Blood according to Contributors(*)

=Class Tracker

- Classify(**), List<MedicalFacility> ⇒ Attributes

-- Invokes the Efficiency when Medical Facility requests Blood if after using this blood supply falls below Danger Level i.e. 35% of Capacity Find list of Seeder and sort according to Max donatable blood. Keep transferring blood until Medical Facility not above Danger Level

Note: Each Seeder can only donate the % of blood above 50% of Capacity *(refer Allocation, Distribution)*

(*) Contributor : { Expiry, Wastage, Total Quantity, Blood Rank }

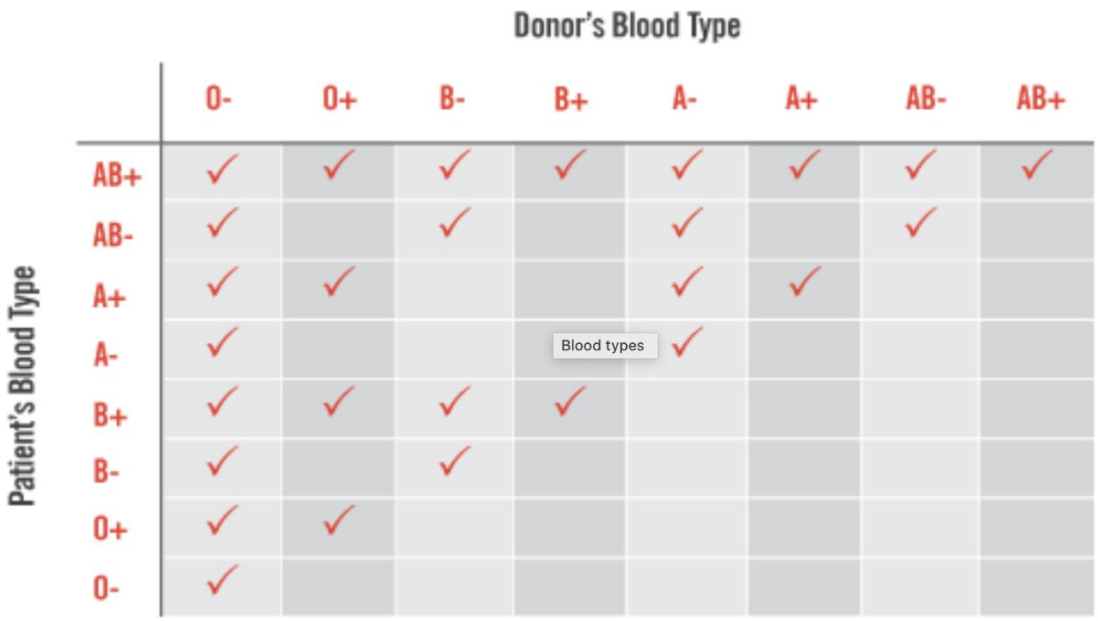
(**) Classify : **Seeder** is someone whose selected blood **type** is above *50% * Capacity_Type*

Request Blood - Facility

The vampire selects the best blood by using the four parameters:

1. Number of days remaining to expire
   - Closest to expiry first
2. Amount of blood wastage
   - Least wastage first
3. Total quantity of blood available in the selected blood type
   - Highest quantity first
4. The rank of the blood.
   - Lowest blood rank first
   - This is chosen by the number of blood types a specified blood type can donate to.

```
BLOOD_RANK = {'AB+' : 1, 'AB-' : 2, 'A+' : 3, 'B+' : 4, 'A-' : 5, 'B-' : 6 ,'O+' : 7, 'O-' : 8}
```



Donor's Blood Type

Patient's Blood Type

	O-	O+	B-	B+	A-	A+	AB-	AB+
AB+	✓	✓	✓	✓	✓	✓	✓	✓
AB-	✓		✓		✓		✓	
A+	✓	✓			✓	✓		
A-	✓				✓			
B+	✓	✓	✓	✓				
B-	✓		✓					
O+	✓	✓						
O-	✓							

Blood types

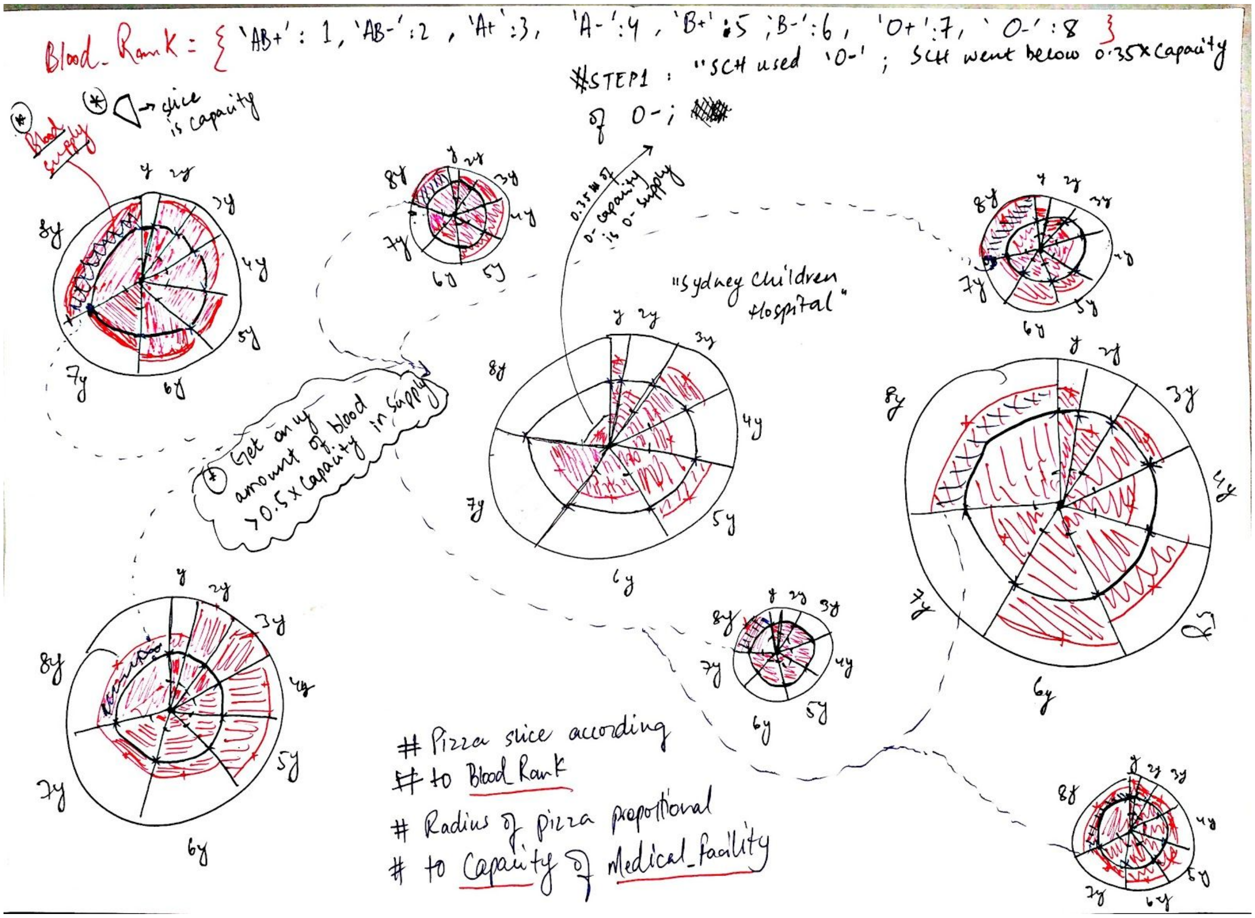
- When a blood is requested by the facility, the vampire goes through the compatibility blood types list and selects all the blood list that can be donated to the requested blood type.
- The best blood of all the compatible blood is chosen and added to a list to calculate the score of the best suitable blood
- The blood from this list is chosen by following criterias
    - Wastage should not be more than 20% of the requested quantity
    - **or**
    - There is within 2 days to expiry
    - **else**
    - If both the above conditions are not satisfied, then the blood with the least bag quantity is selected (**minimal wastage**)
- The best blood from all the compatible blood types are considered to get the best suitable blood.
- The best suitable blood is selected by calculating its score with a derived formula.
- Formula with criterias chosen using ascending order
    - Days to Expiry
    - Blood Wastage
    - Blood Rank
    - **Weight * [1 - [(value - min) / max]]**
- Formula for the criteria with descending order
    - Total quantity
    - **-1 * Weight * [(value - min) / max]**
- Min - minimum value of that criteria (Of the selected list)
- Max - maximum value of that criteria (Of the selected list)
- Value - The value of the current blood
- The weight of each criteria is calculated based on their priority
    - **weight = (1/n) *** $\sum_{m}^{n}$
- n - number of criterias considered
    - In this case there are 4 criterias
- m - Priority level of the criteria
    - Priority 1 - Days to expiry = 0.52
    - Priority 2 - Blood wastage = 0.27
    - Priority 3 - Total quantity = 0.15
    - Priority 4 - Blood Rank = 0.06
- The total score of the blood is attained by the sum of the individual scores of each criteria
    - Blood_Score = Expiery_Score + Wastage_Score + TotalQuantity_Score + Rank_Score

- The blood with the maximum score will be chosen.
- Insertion sort is used to sort the blood score

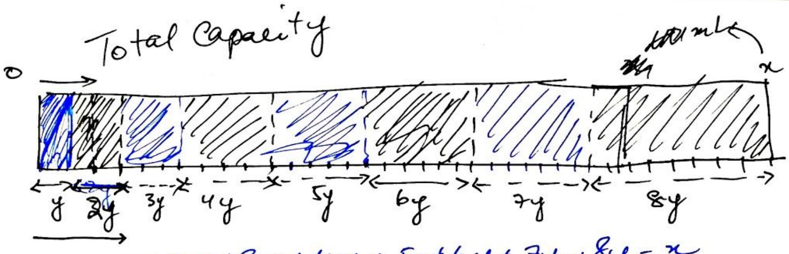
Allocation, Distribution - Facility

Each Circle represents a Medical Facility

The area of each Circle represents the size of the Capacity of the Medical Facility


Blood_Rank = { 'AB+': 1, 'AB-':2 , 'A+':3, 'A-':4 , 'B+':5 ;B-':6 , 'O+':7, 'O-':8 }
#STEP1 : "SCH used 'O-' ; SCH went below 0.35×capacity of O-;
0.35 of O- capacity is O- supply
* Blood supply
* slice is capacity
"Sydney Children Hospital"
* Get any amount of blood >0.5×capacity in supply
y
2y
3y
4y
5y
6y
7y
8y
# Pizza slice according
# to Blood_Rank
# Radius of pizza proportional
# to Capacity of Medical_facility

The entire area of the Circle is divided into 8 portions with increasing size according to the rank of Blood i.e. as O-ve blood can be accepted by everyone, it would be beneficial for a Medical Facility to have more allocated storage for that blood.



$$y+2y+3y+4y+5y+6y+7y+8y=x$$

$$36y=x$$

$$y=\frac{x}{36}$$

Efficency - BLOOD_Rank*y => storage allocation of blood.type()

e.g Total Capacity i.e. max = 10,000 (ml), ~~1*36/x = Y*36/10,000~~

AB+ : $1\times\frac{10,000}{36}=277.8$

AB- : $2\times\frac{10,000}{36}=555.6$

⋮

O- : $8\times\frac{10,000}{36}=2222.2$

Each slice of the Circle is divided into three; setting a threshold of Levels:

Danger - **0.35** * *Capacity_Type*

isSeeder - **0.50** * *Capacity_Type* <____

(*) The Portion between 0.50 and 1.00 of Capacity will be the Donatable Blood.

Maximum - **1.00** * *Capacity_Type* ____>

The Tracker will sort the Medical Facilities according to most donatable blood first

The Tracker will keep removing Donatable Blood (*) from Medical Facilities and adding to the Medical Facility that went below the Danger Level until the Medical Facility is not above Danger Level.

Dafny

As the vampire system handles bloods in all facilities that are registered with them , so it is important to verify the system's blood management .

We have used Dafny to verify our core methods in the system that is insertion sort to sort bloods by score, removing of expired blood algorithm and the score calculation for choosing the best blood to be used.

1. Calculation of Blood Scores
    a. We have used score system for choosing the best blood.
    b. The medical facility will make request to use blood by giving amount needed and the patient's blood type.
    c. The system will then use these parameters for calculation of the scores for each blood that is suitable for the patient
        i. Expiry date
        ii. Wastage = amount of blood in the bag - requested amount
        iii. Total quantity of the current blood type in blood banks
        iv. Blood rank (see Appendix 1)
    d. The score calculation is important that will determine which blood is best
    e. The blood then sorted according to scores by using insertion sort
    f. So, we have verified in the Dafny on the formula that we choose for the score calculation
2. Insertion Sort
    a. We have used insertion sort to sort array of bloods so that it is sorted by descending order that is the higher score is in the front of the array
    b. It is important to sort the bloods by score because the blood with the highest score will be used first.
    c. The system then will give the chosen blood details to the facility that has requested blood.
    d. We have verified the insertion sort so the blood is sorted accordingly
3. Removal of Expired Blood
    a. We have made an algorithm for the expired blood removal so that any medical facility would not accidentally use the expired one.
    b. The algorithm will run automatically that it will check the current date and the date of expiry of each blood which is 42 days after the blood collection date.
    c. It will the notifies the associated medical facilities that the blood is expired.
    d. The Medical Facility will remove the blood and the system will update their records.

Evaluation

Requirements Status Table

All of priority 1(HIGH) requirements were completed, and some of priority2(MEDIUM) and priority3(LOW) were implemented.

1. Status
   - Y - Implemented
   - N - Not implemented
   - P - Partially implemented
2. Priority
   - Priority1 - H - High
   - Priority2 - M - Medium
   - Priority3 - L - Low
3. Requirement - Reference to the requirement section
4. Use Case - Reference to the use case section

NO	Requirement	Use Case	Requirement	Priority	Status
1. VAMPIRE					
1	1.a.1	UC7	Query blood storage details	H	Y
2	1.a.2	UC9	Add Medical Facility	H	Y
3	1.a.2.1	UC9	Add medical facility details	H	Y
4	1.b	UC1	Notify facility and remove expired blood	H	Y
5	1.c.1	UC2	Handle request to use blood	H	Y
6	1.c.2	UC3	Handle request to add blood	H	Y
7	1.c.3		Allow pathology to verify blood	M	N
8	1.c.4		Knows details of all the blood collected	H	Y
9	1.d.1	UC5	Calculate score for facilities and maintain blood balance in the storage	H	Y
10	1.d.2		Manage pathology	L	N

11	1.d.3	UC6,10	Transfer blood to facilities with low score	H	Y
12	1.e.1	UC4	Register donor to the system	M	Y
13	1.e.2		Notify donors about batmobile	L	N
14	1.e.3		Allow donors to query	L	N
15	1.f		Generate batmobile notification	L	N
16	1.g		Handle batmobile request	L	N
2. MEDICAL FACILITY (Hospital/Clinic)					
17	2.a		Request vampire to use blood	H	Y
18	2.b	UC8	Request vampire to add storage units	M	Y
19	2.c		Add blood to a medical facility	H	Y
20	2.d		Query about blood in the system	M	Y
21	2.e		Register donor to the system	L	Y
22	2.f		Query about donor	L	Y
3. MEDICAL FACILITY (Pathology)					
21	3.a.1		Request vampire to manage bloods to other facilities	L	N
22	3.a.2		Request vampire update blood	L	N
4. DONORS					
23	4.a	UC4	Register with the system	M	Y
24	4.b		Get notified about batmobile	L	N
25	4.c		Donate blood in the medical facility	L	N
26	4.d		Enquire about donating blood	L	N
27	4.e		Enquire about health requirements	L	N
5. BATMOBILE					
2 8	5.a		Deliver blood to the facilities	M	N

29	5.b		Go out to collect blood	L	N
30	5.c		Register donor with the system	L	N
31	5.d		Send blood to pathology	L	N

Teamwork

Joshua King : Use cases + WBS + Frontend + Dafny

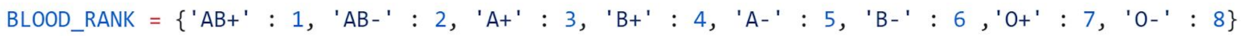
Muhammad Al Waliy Ismail : Executive Summary + WBS + Backend + Dafny

Mohamed Muqaddis : Requirements + WBS + Backend + Dafny

Mohammad Faiz Ather : Requirements + WBS + Backend + Dafny

Appendices

Appendix 1:



```
BLOOD_RANK = {'AB+' : 1, 'AB-' : 2, 'A+' : 3, 'B+' : 4, 'A-' : 5, 'B-' : 6 ,'O+' : 7, 'O-' : 8}
```

Appendix 2:

Project Link:

The submitted project is inside **submission** branch.

- Submission - https://github.com/FaizAther/def-ny_se2011/tree/Submission
- Master - https://github.com/FaizAther/def-ny_se2011