

Information Management

REPORT

The Red Cross

Group 13

Niall Hunt
Jakub Slowinski

Michael McKay
Aaron Duggan

Contents

Background Research	1
The Red Cross	1
Changes based on feedback	2
Overall design	2
UML Changes	2
How we went about designing our system	2
UML Use Case Diagram	4
Design	5
Scenario Descriptions	5
UML Class Diagram	11
Design Decisions	12
UML Activity Diagrams	13
Pay	13
Appeal and amount	14
Design	15
XML	16
Employee DTD	16
Employee Database	17
Donor DTD	18
Donor Database	19
Appeal DTD	20
Appeal Database	21
Donation DTD	22
Donation Database	23
Member Area DTD	24
Member Area Database	25
Member DTD	27
Member Database	28

Table of Contents

Volunteer Area DTD	30
Volunteer Area Database	31
Volunteer DTD	33
Volunteer Database	34
XQueries	36
Who did what	39
Niall Hunt	39
Jakub Slowinski	39
Michael McKay	39
Aaron Duggan	39

*Here we talk
about our
research and
our changes
since the last
report*

Background Research

The Red Cross

The topic we were given for this project was the Irish Red Cross. The Irish Red Cross Society was established in 1949. It is organized on a voluntary basis to provide humanitarian support and community services to the most vulnerable at home and abroad.

Due to the scale at which the Irish Red Cross performs, donations are required to keep the society running at full capacity and are needed to provide care and support. In our original research we thought we would focus on the online donation system of the Red Cross.

However, we now realize that this was a somewhat naïve view and was not broad to model correctly.

In our new design we took a broader view and tried to model their website, as a whole. This includes donations, member area, volunteers and employees.

Changes based on feedback

Overall design

From the feedback given to us on the first report we knew we had a lot of work and changes to make. Our system was far too focused and too narrow-minded to fully design and model. We have pulled right back and tried to focus on how the Red Cross could use technology to help run their organization.

We boiled it down to volunteer databases, employee databases, donation databases and online member areas for all types of users. We also continue to look at the donations system as this is a vital piece of the Red Cross.

This broader view definitely made it a lot easier when it came to modeling and designing our system. It allowed us to really see the whole picture.

UML Changes

These changes required us to update and change our Use Case Diagram and our Class Diagram. We feel that these changes greatly enhance our design and are a lot more realistic than our previous design.

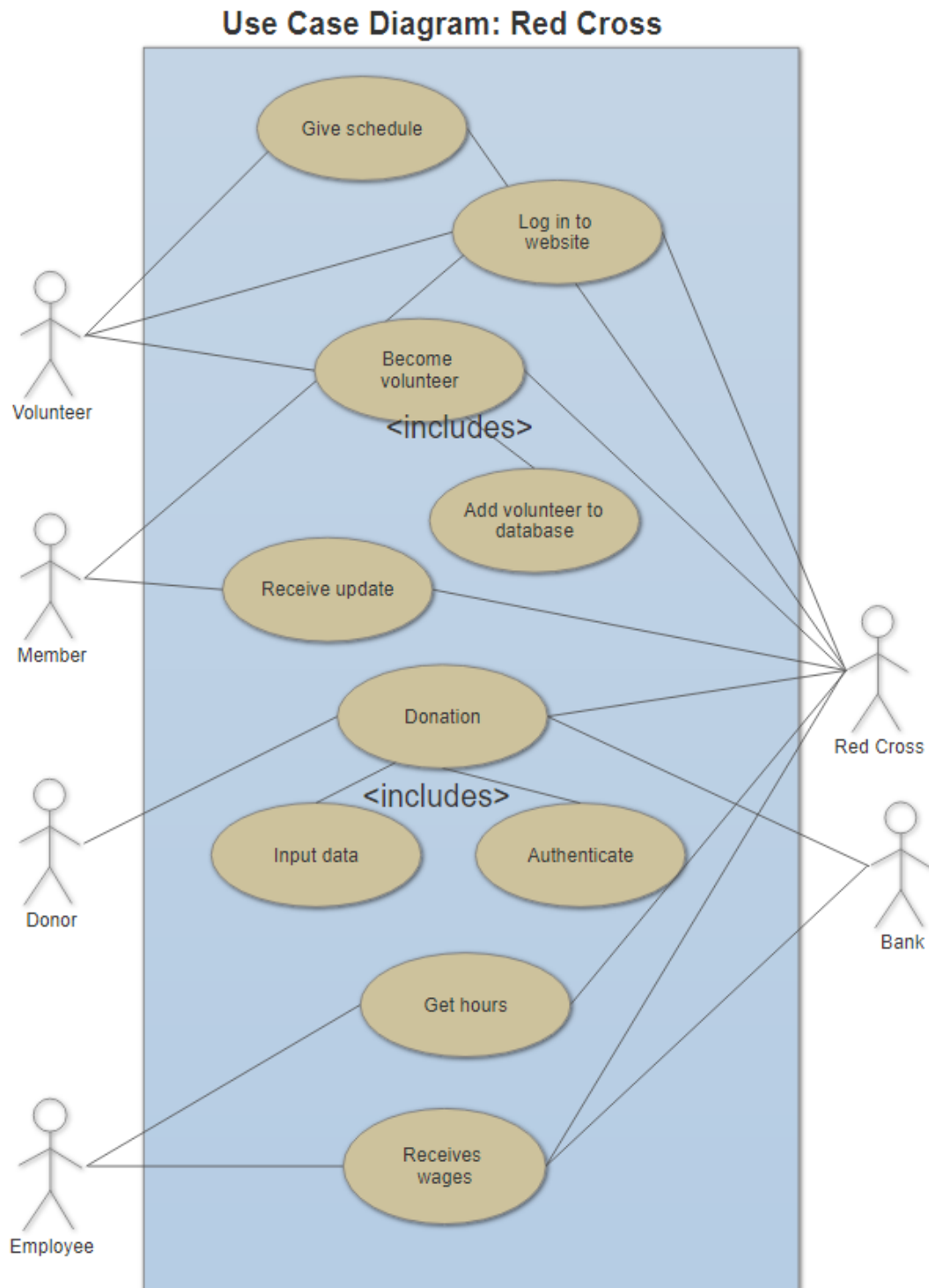
In our case diagram we added new actors (Volunteer, Employee, Member) and the cases they are a part of. Some of these include logging into the website or informing the Red Cross of the hours you can volunteer.

How we went about designing our system

We followed the guidelines and deadlines given to us and this steered our design process. We first designed our UML Use Case. This allowed us to see an overview of how our system was working and allowed us to write the scenario descriptions. We then wrote our 10 scenario descriptions based on our use cases. From there we moved onto Class Diagram and the Activity diagrams. The final part of our design was the ethics canvas. This we could only do once we knew the full scope of our design.

After all the planning we created 8 XML classes each with their own DTD. We created 8 XQueries using these classes. Our XML covers employee, volunteer and donor records and records of donations and appeals. Our queries are then used to narrow these databases down to specific nodes.

UML Use Case Diagram



Design

Here we can see our Actors on the right who will be interacting through our use cases with our system provider Actors on the right (Red Cross and Bank).

In our design each actor is given the opportunity to log into the website and get to their designated area. For example, volunteers can log into the volunteer area.

Donors are given the opportunity to sign up to be a member.

Members and donors are given the opportunity to become volunteers.

Our donation system is still here but it is greatly simplified. Donors input their data and then authenticate their payment.

Scenario Descriptions

Title: Become volunteer

Participating actors:

Volunteer, Red Cross

Entry condition:

Volunteer wants to assist the Red Cross

Red Cross are accepting volunteers

Exit condition:

Volunteer becomes volunteer

Normal scenario:

- Volunteer applies to become volunteer
- Red Cross vets the volunteer to see if they are eligible
- Volunteer gets accepted as a new volunteer

Error scenario:

Volunteer is ineligible to join Red Cross

Title: Add volunteer to database

Participating actors:

Red Cross

Entry condition:

Volunteer successfully applies

Volunteer is accepted by Red Cross

Exit condition:

Volunteer has been added to the database

Normal scenario:

- Red Cross receives volunteer's information
- Red Cross adds the new volunteer to their database.

Error scenario:

System malfunction

Incorrect data supplied by volunteer

Title: Give schedule

Participating actors:

Volunteer, Red Cross

Entry condition:

Volunteer has free time to commit to the Red Cross.

Red Cross needs volunteers.

Exit condition:

Red Cross recognizes and files the volunteer's availability

Normal scenario:

- Volunteer gives availability to the Red Cross
- Volunteer commits to specific role
- Red Cross accepts and responds to volunteer's request

Error scenario:

Volunteer has made an error in the process.

Title: Log into website

Participating actors:

Volunteer, Member, Red Cross

Entry condition:

Volunteer or member has a profile on the Red Cross system

Exit condition:

Volunteer or member can see his profile.

Normal scenario:

- User puts in his email/username.
- User puts in his password.

Error scenario:

User has no account but tries to log in

User puts in wrong password

Title: Donation

Participating actors:

Donor, Red Cross, Bank

Entry condition:

Donor wants to donate.

Donor has money to donate

Exit condition:

Donor successfully donated the amount desired

Normal scenario:

- Donor performs sub-cases.
- Red Cross accepts donation.
- Donor receives confirmation of successful donation by email.

Error scenario:

Server times out during donation.

Red Crosses system fails to take donation.

Title: Input data

Participating actors:

Donor, Red Cross, Bank

Entry condition:

Donor has information to input

Exit condition:

Red Cross has recorded the donor's information.

Normal scenario:

- Donor inputs their information into the system
- System records their information

Error scenario:

Donor did not input a desired field.

Donor made an error with input of information.

Title: Authenticate

Participating actors:

Donor, Red Cross, Bank

Entry condition:

Donor has attempted payment of donation

Exit condition:

Donor successfully donates to the chosen cause.

Normal scenario:

- Donor attempts to donate to chosen cause
- Red Cross and bank authenticate the information provided by the donor
- Donation is approved

Error scenario:

Problem discovered when authenticating the transaction

Title: Receive update

Participating actors:

Member, Red Cross

Entry condition:

Member has opted into mailing list.

Exit condition:

member receives recent update.

Normal scenario:

- Member agrees to receive updates from Red Cross
- Red Cross sends out an update to all members that have opted in
- Member receives the update via email
-

Error scenario:

Member does not receive update from Red Cross

Member's email address is incorrect

Title: Get hours

Participating actors:

Employee, Red Cross

Entry condition:

Red Cross has hours they need to be worked

Red Cross has employees to work these hours

Exit condition:

The employee has received their hours (roster) from Red Cross

Normal scenario:

- Red Cross decides what hours they need worked over a certain period
- Red Cross sends the hours to the specific employee via email, text etc. for this period
- Employee receives these hours

Error scenario:

Employee does not successfully receive hours

Red Cross does not have hours they need to be worked

Title: Receives wages

Participating actors:

Employee, Red Cross, Bank

Entry condition:

Employee has worked in the time period of which they are to be paid

Exit condition:

Employee receives payment

Normal scenario:

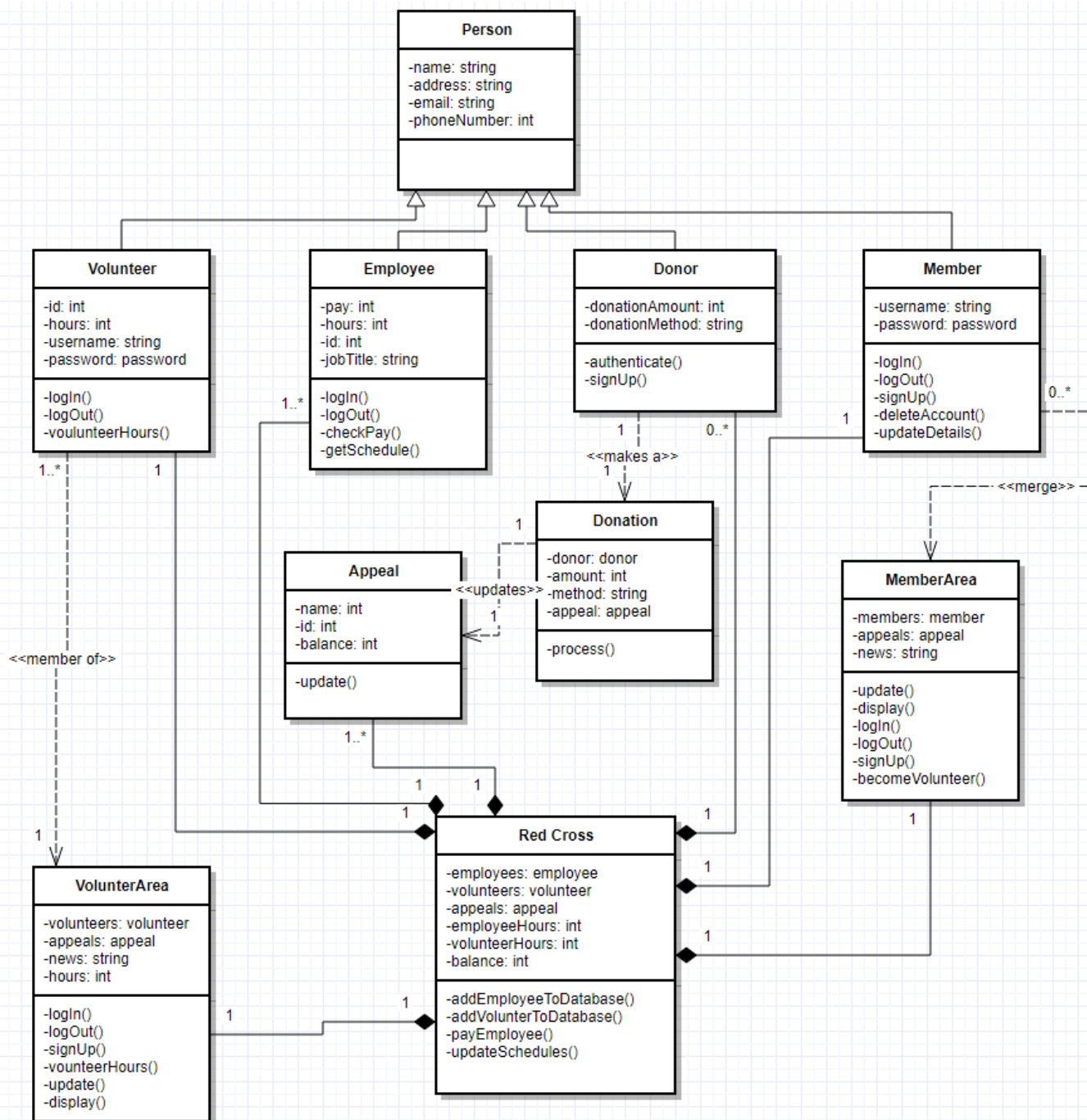
- Employee works
- Red Cross records the time worked by the employee
- Red Cross calculates the amount to be paid
- Red Cross brings all this information to the bank
- Bank pays the employee

Error scenario:

Red Cross does not have the funds to pay this employee

Bank Account details of employee are incorrect

UML Class Diagram



Design Decisions

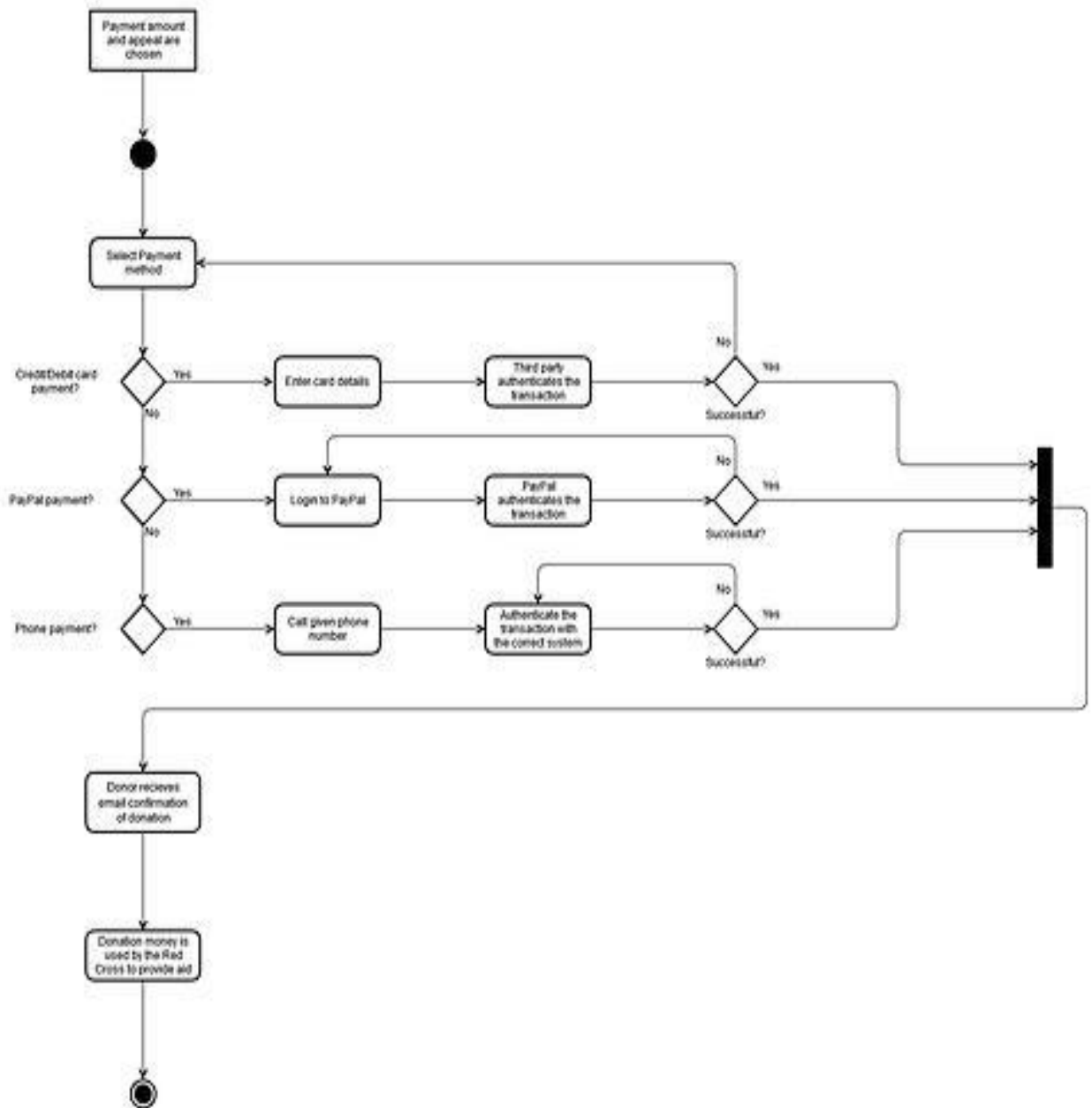
Our main class is the Red Cross class. Without this no other classes could exist. Here we see it has a composition relation with the different people areas, the different people, appeals and donations. The Red Cross class is made up of these other classes.

At the top of our diagram we see the Person class which is used as a parent class for our different types of people. It stores some basic information that we need about each type of person, like name and address. We have a member class, employee class, volunteer class and a donor class implementing our Person class. These all store more specific information about the person.

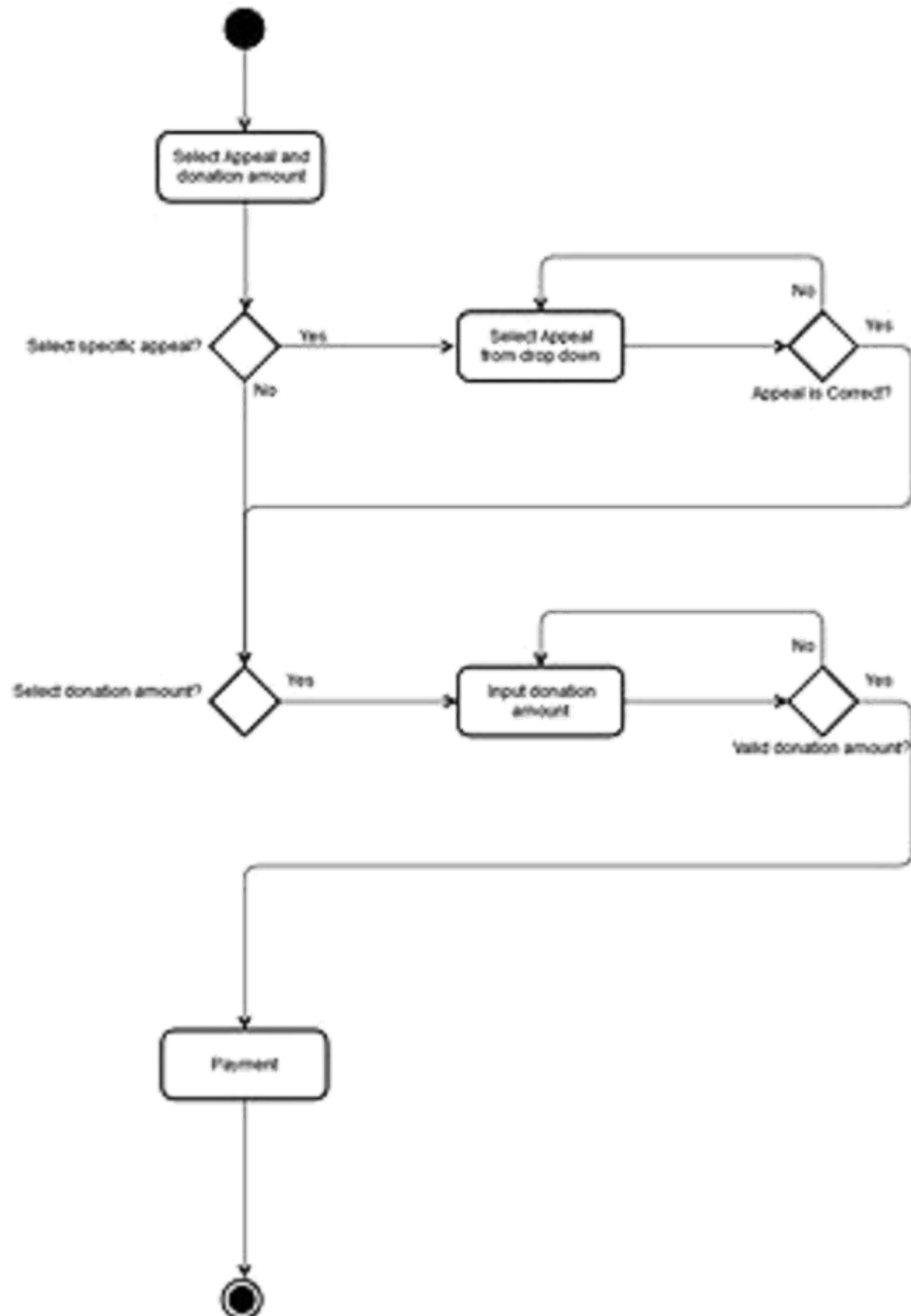
Our donor class is associated with the donation class. Donors make a donation. The Donation class stores the specific information about the donation and uses it to update the appeal class. The appeal class stores information about the specific appeals and the money they have gotten from donations.

UML Activity Diagrams

Pay



Appeal and amount



Design

In the activity diagram the flow of the payment method is easy to follow. The prerequisites for the method can be seen at the top and they are the donation method and the donation amount.

With this information we go into the choose payment method function. Here the donor decides which way they want to pay. In each different payment option, the validation and authentication are left to the appropriate third party.

After the payment is complete the system can then confirm the donation with the donor via email, mail or phone. The personal data the donor inputted is used for this. The confirmation can also include a thank you and links to other appeals that the Red Cross are involved in.

XML

Employee DTD

```
<?xml version="1.0" encoding="UTF-8"?>

<!ELEMENT employeeDatabase (employee*)>

<!ELEMENT employee (name, address, contactInfo, jobInfo)>
<!-- The Database has a list of employees -->
<!ATTLIST employee employeeID ID #REQUIRED>
<!-- Each employee has a unique ID -->

<!ELEMENT name (firstname+, surname+)>
<!-- The first node is name. This consists of at least one firstname and surname -->
<!ELEMENT firstname (#PCDATA)>
<!ELEMENT surname (#PCDATA)>

<!ELEMENT address (addressLine1, addressLine2?, city, county, country?)>
<!-- Each employee also has an address node which is used to help pay them -->
<!ELEMENT addressLine1 (#PCDATA)>
<!ELEMENT addressLine2 (#PCDATA)>
<!ELEMENT city (#PCDATA)>
<!ELEMENT county (#PCDATA)>
<!ELEMENT country (#PCDATA)>

<!ELEMENT contactInfo (email, phoneNumber)>
<!-- Contact info is needed for each employee -->
<!ELEMENT email (#PCDATA)>
<!ELEMENT phoneNumber (#PCDATA)>

<!ELEMENT jobInfo (payPerHour, hoursWorkedPerWeek, jobTitle)>
<!-- Here we have the information specifically regrading their jobs -->
<!ELEMENT payPerHour (#PCDATA)>
<!ATTLIST payPerHour currency CDATA #REQUIRED>
<!ELEMENT hoursWorkedPerWeek (#PCDATA)>
<!ELEMENT jobTitle (#PCDATA)>
```

Employee Database

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE employeeDatabase SYSTEM "employeeDatabase.dtd">

<employeeDatabase>
  <employee employeeID = "id_001">
    <name>
      <firstname> Aoife </firstname>
      <surname> O'Connel </surname>
    </name>

    <address>
      <addressLine1> 3 Beach Road </addressLine1>
      <addressLine2> Beach Village </addressLine2>
      <city> Dublin </city>
      <county> Dublin </county>
    </address>

    <contactInfo>
      <email> JohnDoe@gmail.com </email>
      <phoneNumber> 0871234524 </phoneNumber>
    </contactInfo>

    <jobInfo>
      <payPerHour currency = "Euro">10.00</payPerHour>
      <hoursWorkedPerWeek> 30 </hoursWorkedPerWeek>
      <jobTitle> Secratery </jobTitle>
    </jobInfo>
  </employee>

  <employee employeeID = "id_002">

    <name>
      <firstname> John </firstname>
      <firstname> Paul </firstname>
      <surname> Johnston </surname>
    </name>

    <address>
      <addressLine1> 16 Broad Street </addressLine1>
      <city> Dublin </city>
      <county> Dublin </county>
      <country> Ireland </country>
    </address>

    <contactInfo>
      <email> janejohnston@gmail.com </email>
      <phoneNumber> 0878463858 </phoneNumber>
    </contactInfo>

    <jobInfo>
      <payPerHour currency = "Euro">15.00</payPerHour>
      <hoursWorkedPerWeek> 40 </hoursWorkedPerWeek>
      <jobTitle> Accountant </jobTitle>
    </jobInfo>
  </employee>
</employeeDatabase>
```

Figure 1. The effect of the number of nodes on the number of nodes in the network.

Donor Database

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE donorDatabase SYSTEM "donorDatabase.dtd">

<donorDatabase>
  <donor>
    <name>
      <firstname> John </firstname>
      <surname> Doe </surname>
    </name>

    <address>
      <addressLine1> 123 Beach Road </addressLine1>
      <addressLine2> Beach Village </addressLine2>
      <city> Dublin </city>
      <county> Dublin </county>
    </address>

    <contactInfo>
      <email> JohnDoe@gmail.com </email>
      <phoneNumber> 0871234524 </phoneNumber>
    </contactInfo>

    <donationInfo>
      <donationAmount currency = "Euro">50.00</donationAmount>
      <donationMethod> Debit Card </donationMethod>
      <appeal> Where needed most Appeal </appeal>
    </donationInfo>
  </donor>

  <donor>
    <name>
      <firstname> Jane </firstname>
      <surname> Johnston </surname>
    </name>

    <address>
      <addressLine1> 16 Broad Street </addressLine1>
      <city> Dublin </city>
      <county> Dublin </county>
      <country> Ireland </country>
    </address>

    <contactInfo>
      <email> janejohnston@gmail.com </email>
      <phoneNumber> 0878463858 </phoneNumber>
    </contactInfo>

    <donationInfo>
      <donationAmount currency = "Euro">25.50</donationAmount>
      <donationMethod> PayPal </donationMethod>
      <appeal> Myanmar Emergency Appeal </appeal>
    </donationInfo>
  </donor>
</donorDatabase>
```

Appeal DTD

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!ELEMENT appealDatabase (appeal*)>
```

```
<!-- This database is for storing data on the appeals in the IRS -->
```

```
<!ELEMENT balance (#PCDATA)>
```

```
<!-- The balance shows the amount of money currently raised for the appeal -->
```

```
<!ELEMENT appealNumber (#PCDATA)>
```

Appeal Database

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE appealDatabase SYSTEM "appealDatabase.dtd">

<appealDatabase>
  <appeal id="123456789">
    <name> Christmas Appeal </name>
    <balance> €100000.00 </balance>
    <appealNumber> 1 </appealNumber>
  </appeal>

  <appeal id="987654321">
    <name> Myanmar Crisis Appeal </name>

    <balance> €200000.00 </balance>
    <appealNumber> 2 </appealNumber>
  </appeal>

  <appeal id="135792468">
    <name> Yemen Crisis Appeal </name>
    <balance> €500000.00 </balance>
    <appealNumber> 3 </appealNumber>
  </appeal>

  <appeal id="246813579">
    <name> Migration Crisis Appeal </name>
    <balance> €10000.00 </balance>
    <appealNumber> 4 </appealNumber>
  </appeal>
</appealDatabase>
```


Donation DTD

```
<?xml version="1.0" encoding="UTF-8"?>

<!ELEMENT donationDatabase (donor*)>

<!-- This database stores information on donations made to an appeal -->

<!ELEMENT donor (id)>

<!ELEMENT donationInfo (donationAmount, donationMethod, appeal)>
<!-- Each donation has an amount donated, a method of donation and which
appeal it's being made to -->

<!ELEMENT donationAmount (#PCDATA)>
<!ELEMENT donationMethod (#PCDATA)>
<!ELEMENT appeal (#PCDATA)>
```

Donation Database

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE appealDatabase SYSTEM "appealDatabase.dtd">

<appealDatabase>
  <appeal id="123456789">
    <name> Christmas Appeal </name>
    <balance> €100000.00 </balance>
    <appealNumber> 1 </appealNumber>
  </appeal>

  <appeal id="987654321">
    <name> Mynamar Crisis Appeal </name>
    <balance> €200000.00 </balance>
    <appealNumber> 2 </appealNumber>
  </appeal>

  <appeal id="135792468">
    <name> Yemen Crisis Appeal </name>
    <balance> €500000.00 </balance>
    <appealNumber> 3 </appealNumber>
  </appeal>

  <appeal id="246813579">
    <name> Migration Crisis Appeal </name>
    <balance> €10000.00 </balance>
    <appealNumber> 4 </appealNumber>
  </appeal>
</appealDatabase>
```

Member Area DTD

```
<?xml version="1.0" encoding="UTF-8"?>

<!ELEMENT memberArea (memberSpace*)>
<!ELEMENT memberSpace (ownInfo, appeals*, news*)>

<!-- general information for a member -->
<!ELEMENT ownInfo (name, address, contactInfo, accountAccess)>
<!ELEMENT name (firstname+, surname+)>
<!-- can have one or more firstnames and last names -->
<!ELEMENT firstname (#PCDATA)>
<!ELEMENT surname (#PCDATA)>

<!ELEMENT address (addressLine1, addressLine2?, city, county, country?)>
<!-- NEED addressLine1, city and county but addressLine2 and country are
not necessary -->
<!ELEMENT addressLine1 (#PCDATA)>
<!ELEMENT addressLine2 (#PCDATA)>
<!ELEMENT city (#PCDATA)>
<!ELEMENT county (#PCDATA)>
<!ELEMENT country (#PCDATA)>

<!-- members contact information -->
<!ELEMENT contactInfo (email, phoneNumber)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT phoneNumber (#PCDATA)>

<!ELEMENT accountAccess ( accountUsername, accountPassword)>
<!ELEMENT accountUsername (#PCDATA)>
<!ELEMENT accountPassword (#PCDATA)>

<!ELEMENT appeals ( appealName, appealCountry*, appealUrgency)>
<!ELEMENT appealName (#PCDATA)>
<!ELEMENT appealUrgency (#PCDATA)>

<!-- news, byline not necessart -->
<!ELEMENT news ( article, headline, byline?,author)>
<!ELEMENT article (#PCDATA)>
<!ELEMENT headline (#PCDATA)>
<!ELEMENT byline (#PCDATA)>
<!ELEMENT author (#PCDATA)>
```

Member Area Database

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE memberArea SYSTEM "member_area.dtd">

<memberArea>
  <memberSpace>
    <ownInfo>
      <name>
        <firstName> Cristiano </firstName>
        <surname> Ronaldo </surname>
      </name>

      <address>
        <addressLine1> 7 Real Road </addressLine1>
        <city> Madrid </city>
        <county> Madrid </county>
        <country> Spain </country>
      </address>

      <contactInfo>
        <email> ronaldoc@tcd.ie </email>
        <phoneNumber> 0871234567 </phoneNumber>
      </contactInfo>

      <accountAccess>
        <accountUsername> CR7 </accountUsername>
        <accountPassword> thebest </accountPassword>
      </accountAccess>
    </ownInfo>

    <appeals>
      <appealName> Childrens Hospital Appeal </appealName>
      <appealUrgency> 1 </appealUrgency>
    </appeals>

    <appeals>
      <appealName> Homelessness Appeal </appealName>
      <appealUrgency> 2 </appealUrgency>
    </appeals>

    <appeals>
      <appealName> Homelessness Appeal </appealName>
      <appealUrgency> 2 </appealUrgency>
    </appeals>

    <appeals>
      <appealName> New Schools Appeal </appealName>
      <appealUrgency> 3 </appealUrgency>
    </appeals>

    <news>
      <article> Your kind donate to the New Schools Appeal you can
help
to build new rooms and provide new state of the art
machinery
and technology for many childrens hospitals around the
world

      </article>
      <headline> Improved Hospitals Around the World </headline>
      <author> James Murphy </author>
    </news>
```

```

    </volunteerSpace>

    <volunteerSpace>
      <ownInfo>
        <name>
          <firstName> Kyrie </firstName>
          <surname> Irving </surname>e>
        </name>

        <address>
          <addressLine1> 12 Greenough Street </addressLine1>
          <addressLine2> Brookline </addressLine2>
          <city> Boston </city>
          <county> Massachusetts </county>
          <country> USA </country>
        </address>

        <contactInfo>
          <email> irvingk@tcd.ie </email>
          <phoneNumber> 6172429980 </phoneNumber>
        </contactInfo>

        <accountAccess>
          <accountUsername> kyriell </accountUsername>
          <accountPassword> celtics </accountPassword>
        </accountAccess>
      </ownInfo>

      <appeals>
        <appealName> Childrens Hospital Appeal </appealName>
        <appealUrgency> 1 </appealUrgency>
      </appeals>

      <appeals>
        <appealName> Homelessness Appeal </appealName>
        <appealUrgency> 2 </appealUrgency>
      </appeals>

      <appeals>
        <appealName> New Schools </appealName>
        <appealUrgency> 3 </appealUrgency>
      </appeals>

      <news>
        <article> Your kind donate to the New Schools Appeal you can
help
to build new rooms and provide new state of the art
machinery
and technology for many childrens hospitals around the
world
        </article>
        <headline> Improved Hospitals Around the World </headline>
        <author> James Murphy </author>
      </news>
    </volunteerSpace>
  </memberArea>

```

Member DTD

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT memberDatabase (member*)>

<!ELEMENT member (name, address, contactInfo, accountAccess)>

<!-- can have one or more firstname and surname -->
<!ELEMENT name (firstname+, surname+)>
<!ELEMENT firstname (#PCDATA)>
<!ELEMENT surname (#PCDATA)>

<!-- addressLine1, city and county are necessary.. addressLine2 and country
are not -->
<!ELEMENT address (addressLine1, addressLine2?, city, county, country?)>
<!ELEMENT addressLine1 (#PCDATA)>
<!ELEMENT addressLine2 (#PCDATA)>
<!ELEMENT city (#PCDATA)>
<!ELEMENT county (#PCDATA)>
<!ELEMENT country (#PCDATA)>

<!ELEMENT contactInfo (email, phoneNumber)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT phoneNumber (#PCDATA)>

<!-- members account information -->
<!ELEMENT accountAccess ( accountUsername, accountPassword)>
<!ELEMENT accountUsername (#PCDATA)>
<!ELEMENT accountPassword (#PCDATA)>
```

Member Database

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE memberDatabase SYSTEM "member_database.dtd">

<memberDatabase>
  <member>
    <name>
      <firstName> Cristiano </firstName>
      <surname> Ronaldo </surname>
    </name>

    <address>
      <addressLine1> 7 Real Road </addressLine1>
      <city> Madrid </city>
      <county> Madrid </county>
      <country> Spain </country>
    </address>

    <contactInfo>
      <email> ronaldoc@tcd.ie </email>
      <phoneNumber> 0871234567 </phoneNumber>
    </contactInfo>

    <accountAccess>
      <accountUsername> CR7 </accountUsername>
      <accountPassword> thebest </accountPassword>
    </accountAccess>

    <name>
      <firstName> Kyrie </firstName>
      <surname> Irving </surname>
    </name>

    <address>
      <addressLine1> 12 Greenough Street </addressLine1>
      <addressLine2> Brookline </addressLine2>
      <city> Boston </city>
      <county> Massachusetts </county>
      <country> USA </country>
    </address>

    <contactInfo>
      <email> irvingk@tcd.ie </email>
      <phoneNumber> 6172429980 </phoneNumber>
    </contactInfo>

    <accountAccess>
      <accountUsername> kyriell </accountUsername>
      <accountPassword> celtics </accountPassword>
    </accountAccess>

    <name>
      <firstName> Rob </firstName>
      <surname> Lipsett </surname>
    </name>

    <address>
      <addressLine1> 2 Wellington Road </addressLine1>
      <addressLine2> Blackrock </addressLine2>
      <city> Dublin </city>
      <county> Dublin </county>
```

```
</address>

<contactInfo>
  <email> lipsettr@tcd.ie </email>
  <phoneNumber> 0831234567 </phoneNumber>
</contactInfo>

<accountAccess>
  <accountUsername> thelipsett </accountUsername>
  <accountPassword> lippo </accountPassword>
</accountAccess>
</member>
</memberDatabase>
```


Volunteer Area DTD

```

<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT volunteerArea (volunteerSpace*)>
<!ELEMENT volunteerSpace (ownInfo, appeals*, news*, hours)>

<!ELEMENT ownInfo (name, address, contactInfo, accountAccess)>
<!ELEMENT name (firstname+, surname+)>
<!-- one or more firstnames and surnames -->
<!ELEMENT firstname (#PCDATA)>
<!ELEMENT surname (#PCDATA)>

<!ELEMENT address (addressLine1, addressLine2?, city, county, country?)>
<!-- addressline1 always needed, addressline2 one or nothing -->
<!ELEMENT addressLine1 (#PCDATA)>
<!ELEMENT addressLine2 (#PCDATA)>
<!ELEMENT city (#PCDATA)>
<!ELEMENT county (#PCDATA)>
<!ELEMENT country (#PCDATA)>

<!ELEMENT contactInfo (email, phoneNumber)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT phoneNumber (#PCDATA)>

<!ELEMENT accountAccess ( accountUsername, accountPassword, idNumber,
hoursAvailable)>
<!-- every volunteer has a username, account, id number, and hours
available -->
<!ELEMENT accountUsername (#PCDATA)>
<!ELEMENT accountPassword (#PCDATA)>
<!ELEMENT idNumber (#PCDATA)>
<!ELEMENT hoursAvailable (#PCDATA)>
<!-- all this above is your volunteer info when you log into the volunteer
area -->

<!ELEMENT appeals ( appealName, appealCountry*, appealUrgency)>
<!-- appeal urgency determines which donations get donations when the
donation is sent to where needed most, an urgency of 1 is highest/ 10
lowest
, not every appeal has a set country so it can be none or one or more
-->
<!ELEMENT appealName (#PCDATA)>
<!ELEMENT appealCountry (#PCDATA)>
<!ELEMENT appealUrgency (#PCDATA)>

<!ELEMENT news ( article, headline, byline?,author)>
<!-- not every article has byline -->
<!ELEMENT article (#PCDATA)>
<!ELEMENT headline (#PCDATA)>
<!ELEMENT byline (#PCDATA)>
<!ELEMENT author (#PCDATA)>

<!ELEMENT hours ( hoursEligible, hoursGiven)>
<!-- every volunteer has a username, account, id number, and hours
available -->
<!ELEMENT hoursEligible (#PCDATA)>
<!ELEMENT hoursGiven (#PCDATA)>

```

Volunteer Area Database

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE volunteerArea SYSTEM "volunteer_area.dtd">

<volunteerArea>
  <volunteerSpace>
    <ownInfo>
      <name>
        <firstname> Dylan</firstname>
        <surname> Long </surname>
      </name>

      <address>
        <addressLine1> 123 Beach Road </addressLine1>
        <addressLine2> Willow Heights </addressLine2>
        <city> Rolando </city>
        <county> Clare </county>
      </address>

      <contactInfo>
        <email> longy@gmail.com </email>
        <phoneNumber> 0872234524 </phoneNumber>
      </contactInfo>

      <accountAccess>
        <accountUsername> longy </accountUsername>
        <accountPassword> longerson </accountPassword>
        <idNumber> 52 </idNumber>
        <hoursAvailable> 13 </hoursAvailable>
      </accountAccess>

    </ownInfo>
    <appeals>
      <appealName> Christmas Appeal </appealName>
      <appealUrgency> 1 </appealUrgency>
    </appeals>

    <appeals>
      <appealName> Yemen Crisis Appeal </appealName>
      <appealCountry> Yemen </appealCountry>
      <appealUrgency> 3 </appealUrgency>
    </appeals>

    <news>
      <article> Your donations has been put to amazing use as we help
a village of 200
in Yemen with the building of a new water well and pump
</article>
      <headline> Yemen relief arrives just in time </headline>
      <author> Damien McCarthy </author>
    </news>

    <hours>
      <hoursEligible> 13 </hoursEligible>
      <hoursGiven> 20 </hoursGiven>
    </hours>
  </volunteerSpace>
</volunteerArea>
```

```
</hours>

</volunteerSpace>

<volunteerSpace>
  <ownInfo>
    <name>
      <firstname> Darren </firstname>
      <surname> Madden </surname>
    </name>

    <address>
      <addressLine1> 47 Oak Rise </addressLine1>
      <city> Clondalkin </city>
      <county> Dublin 22 </county>
      <country> Ireland </country>
    </address>

    <contactInfo>
      <email> darbro@gmail.com </email>
      <phoneNumber> 0871263858 </phoneNumber>
    </contactInfo>

    <accountAccess>
      <accountUsername> dar_mad </accountUsername>
      <accountPassword> 111 </accountPassword>
      <idNumber> 53 </idNumber>
      <hoursAvailable> 6 </hoursAvailable>
    </accountAccess>
  </ownInfo>

  <appeals>
    <appealName> Christmas Appeal </appealName>
    <appealUrgency> 1 </appealUrgency>
  </appeals>

  <appeals>
    <appealName> Yemen Crisis Appeal </appealName>
    <appealCountry> Yemen </appealCountry>
    <appealUrgency> 3 </appealUrgency>
  </appeals>

  <news>
    <article> Your donations has been put to amazing use as we help
a village of 200
in Yemen with the building of a new water well and pump
</article>
    <headline> Yemen relief arrives just in time </headline>
    <author> Damien McCarthy </author>
  </news>

  <hours>
    <hoursEligible> 6 </hoursEligible>
    <hoursGiven> 5 </hoursGiven>

  </hours>

</volunteerSpace>
</volunteerArea>
```

Volunteer DTD

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT volunteerDatabase (volunteer*)>

<!ELEMENT volunteer (name, address, contactInfo, accountAccess)>

<!ELEMENT name (firstname+, surname+)>
<!-- one or more firstnames and surnames -->
<!ELEMENT firstname (#PCDATA)>
<!ELEMENT surname (#PCDATA)>

<!ELEMENT address (addressLine1, addressLine2?, city, county, country?)>
<!-- addressline1 always needed, addressline2 one or nothing -->
<!ELEMENT addressLine1 (#PCDATA)>
<!ELEMENT addressLine2 (#PCDATA)>
<!ELEMENT city (#PCDATA)>
<!ELEMENT county (#PCDATA)>
<!ELEMENT country (#PCDATA)>

<!ELEMENT contactInfo (email, phoneNumber)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT phoneNumber (#PCDATA)>

<!ELEMENT accountAccess ( accountUsername, accountPassword, idNumber,
hoursAvailable)>
<!-- every volunteer has a username, account, id number, and hours
available -->
<!ELEMENT accountUsername (#PCDATA)>
<!ELEMENT accountPassword (#PCDATA)>
<!ELEMENT idNumber (#PCDATA)>
<!ELEMENT hoursAvailable (#PCDATA)>
```

Volunteer Database

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE volunteerDatabase SYSTEM "volunteer_database.dtd">

<volunteerDatabase>
  <volunteer>
    <name>
      <firstname> Dylan</firstname>
      <surname> Long </surname>
    </name>

    <address>
      <addressLine1> 123 Beach Road </addressLine1>
      <addressLine2> Willow Heights </addressLine2>
      <city> Rolando </city>
      <county> Clare </county>
    </address>

    <contactInfo>
      <email> longy@gmail.com </email>
      <phoneNumber> 0872234524 </phoneNumber>
    </contactInfo>

    <accountAccess>
      <accountUsername> longy </accountUsername>
      <accountPassword> longerson </accountPassword>
      <idNumber> 52 </idNumber>
      <hoursAvailable> 13 </hoursAvailable>
    </accountAccess>
  </volunteer>

  <volunteer>

    <name>
      <firstname> Darren </firstname>
      <surname> Madden </surname>
    </name>

    <address>
      <addressLine1> 47 Oak Rise </addressLine1>
      <city> Clondalkin </city>
      <county> Dublin 22 </county>
      <country> Ireland </country>
    </address>

    <contactInfo>
      <email> darbro@gmail.com </email>
      <phoneNumber> 0871263858 </phoneNumber>
    </contactInfo>

    <accountAccess>
      <accountUsername> dar_mad </accountUsername>
      <accountPassword> 111 </accountPassword>
      <idNumber> 53 </idNumber>
      <hoursAvailable> 6 </hoursAvailable>
    </accountAccess>
  </volunteer>
```

```
<volunteer>

    <name>
        <firstname> Jack </firstname>
        <surname> Rovet </surname>
    </name>

    <address>
        <addressLine1> 73 Big Street </addressLine1>
        <city> Coolock </city>
        <county> Dublin </county>
        <country> Ireland </country>
    </address>

    <contactInfo>
        <email> Jack1337@gmail.com </email>
        <phoneNumber> 0852742195 </phoneNumber>
    </contactInfo>

    <accountAccess>
        <accountUsername> parojack </accountUsername>
        <accountPassword> 12345r </accountPassword>
        <idNumber> 54 </idNumber>
        <hoursAvailable> 10 </hoursAvailable>
    </accountAccess>
</volunteer>
</volunteerDatabase>
```

XQueries

Number 1. Works on volunteerDatabase.xml

It returns the number 29 which is all the hours total volunteers can work this week. Has a user defined function and for function

```
declare function local:all_hours()
{
  for $s in
    doc('volunteerDatabase.xml')
  return
    sum($s//volunteer/accountAccess/hoursAvailable)
};
local:all_hours()
```

Number 2. Works on appealDatabase.xml

It returns the number 202500. Has a user defined function and for function

```
Declare function local:avg_balance()
{
  for $s in
    doc('appealDatabase.xml')/appealDataBase/appeal/balance
  return
    <Avg_Balance>{avg(string($s))}</Avg_Balance>
};
local:avg_balance()
```

Number 3. Works on appealDatabase.xml

Retrieves the names of all the appeals. Uses for and order by.

```
for $x in
  doc("appealDatabase.xml")/appealDatabase/appeal
order by $x/appealNumber ascending
return
  {$x/name}
```

Number 4. Works on employeeDatabase.xml

Retrieves the names of all employees earning €15.00 or more. Uses for and where.

```
for $x in
doc("employeeDatabase.xml") /employeeDatabase/employee
where $x/jobInfo/payPerHour >= 15.00
return
<HighRankingEmployee>
{$x/name}
</HighRankingEmployee>
```

Number 5

Works on volunteerDatabase.xml

Returns any volunteer living in rolando result = Dylan Long

```
for $so in
doc('volunteerDatabase.xml')//volunteer

let $s := (substring($so/address/city, 1))
let $n :=(substring($so/name, 1))
let $z :=(substring($so/contactInfo/phoneNumber, 1))
where ($s= "Rolando")
return ($n, $z)
```

Number 6

Works on donorDatabase

Returns the name of any donor with the donation method Debit Card

```
for $x in doc("donorDatabase.xml") //donor
where $x/donationInfo/string(donationMethod) = "Debit Card"

return fn:string-join($x//name[firstname,surname])
```

Number 7

Works on donorDatabase

Returns the name of any donor that has donated more than €100

```
for $x in doc("donorDatabase.xml") //donor
where $x//donationAmount >= 100
```

```
return $x/name
```

Number 8

Works on employeeDatabase

Returns whether there are any entry level jobs (payPerHour < €15) available

```
for $t in doc("employeeDatabase.xml") //employee
where $t//payPerHour < 15.00
```

```
return
<entryLevelJobs>
{
  if (every $x in
    doc("employeeDatabase.xml") //employee satisfies $x/jobInfo/payPerHour >= 15)
  then "All entry level jobs are open"
  else if (count($t) <= 2)
  then "Some entry level jobs are open"
  else "No entry level jobs are open"
}
</entryLevelJobs>
```

Who did what

Niall Hunt

- Report part 1 and 2
- Employee database
- Donor database
- Queries number 4, 6, 7, 8
- Redid UML Class diagram

Jakub Slowinski

- Report part 2
- Volunteer area database
- Volunteer database
- Scenario descriptions
- Queries
- UML Use Case Diagram

Michael McKay

- Member Area database
- Member database
- Scenario Descriptions
- Use Case Diagram
- Queries
- Ethics Canvas

Aaron Duggan

- Appeal Database
- Donation Database
- Queries

Individuals Affected

- Employees
- Volunteers
- Donors
- Members

Groups Affected

- Red Cross employees and the Red Cross organisation are involved in the production of our system
- Donors, members, volunteers and employees would all use our system

Behaviour

- Employees receiving their hours/roster will have their schedule affected

Relations

- Those the Red Cross aim to help would be affected due to volunteers committing their time
- They would also be affected by donors kindly donating
- This would strengthen the relationship between those in need and the average donor/volunteer

Worldviews

- Our system helps to make Red Cross more efficient so people would have good views towards it
- When it comes to Red Cross itself, it has a very positive worldwide view

Group Conflicts

- No conflict would arise because all users of the system are there to help/complete a goal

Product or Service Failure

- Users personal information could be leaked due to security fault

Problematic Use of Resources

- Information inputted incorrectly to the system

What can we do ?

- It is important to make it as easy as possible to successfully accept donations/volunteers hours
- Users need to be able to use the system easily and efficiently

Uncategorised Ideas