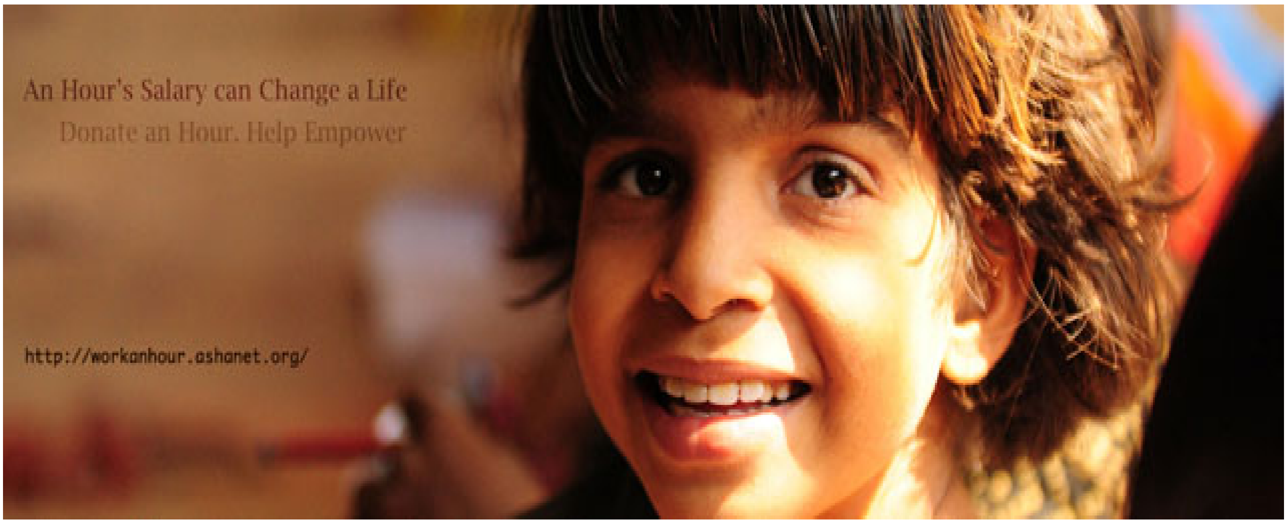
NMS

(NGO Management Software)


An Hour's Salary can Change a Life
Donate an Hour. Help Empower
http://workanhour.ashanet.org/

System Analysis and System Description

❿ Aishwarya Hakande (13CS10004)

❿ Sohan De Sarkar (13CS30033)

Table of Contents

1.0 Introduction

1.1 Purpose

❿ This Software Requirements Specification provides a complete description of all the functions and specifications of the NGO Management System (NMS) developed for the management of the NGO which works for providing free school education for poor children.

1.2 Scope

- The NGO Management Software (NMS) is designed to provide NGO, which works for providing free school education to children, a system for its management.
- This software allows the *volunteers* to register poor students who approach them for help and maintain track record of their performance. Also, volunteers contact donors when asked to by the President.
- This software allows the *donors* to register and enter the amount of money or other donations they intend to contribute, the NGO allows anonymous donations also.
- This software also allows the *President* to check the various records. It allows him to check the amount of money needed to continue its operations, check for insufficiency of funds in case of which it alerts the volunteers to contact the pledged donors and maintain an account of expenditure made by the NGO in the present year. The President can keep track of all the volunteers also, he is responsible to add/remove volunteers.
- This software has a very user-friendly interface resulting

in knowing each and every usability feature of the system.

1.3 Glossary

Term	Definition
NMS	NGO Management Software
Volunteers	Poor students approach volunteers for help and they in turn register them and maintain a record of their performance. Also contact donors when required.
Donors	The people who pledge to help NGO with a certain amount of money or other donations such as books, dress, shoes, bags.
President	The person who manages funds and supervises the NGO.
Database	Collection of all information monitored by the system.
DFD	Data Flow Diagram
GUI	Graphical User Interface
I/O	Input/Output
OOA	Object Oriented Architecture
OOD	Object Oriented Design
ERD	Entity Relationship Diagram

1.4 References

1. [IEEE] the applicable IEEE standards are published in "IEEE Standards

Collection”, 2001 Edition.
2. Class slides.

1.5 Overview of document

The remaining chapters and their contents are listed below.

❿ Section 2 is Feasibility study which helps us understand the problem in depth. In this section, we analyse the various stakeholders and discuss the alternatives.

❿ Section 3 is the Requirements Analysis in which we first review the feasibility study and look into it for deeper information. Also in this section we gather functional and non-functional requirements.

❿ Section 4 is the final section in which we define the global system architecture, select platform comprising hardware, software and networking. Also we propose software architecture for the new system and do database design.

2.0 Feasibility Study

2.1 Understanding the Problem

In this section, we try to understand the purpose of the software. The purpose of the NGO Management Software (NMS) is to provide NGO, which works for providing free school education to children, a system for its management. This software allows the volunteers to register poor students who approach them for help and maintain track record of their performance. Also, volunteers contact donors when asked to by the President. This software allows the donors to register for the NGO and enter the amount of money or other donations they intend to contribute. This software also allows the President to check the various records. It allows him to check the amount of money needed to continue its operations, check for insufficiency of funds and alert the volunteers to contact the donor. It allows the President to add/remove volunteers. It also allows to maintain an account of expenditure made by NGO in the present year.

2.2 Scope the Problem

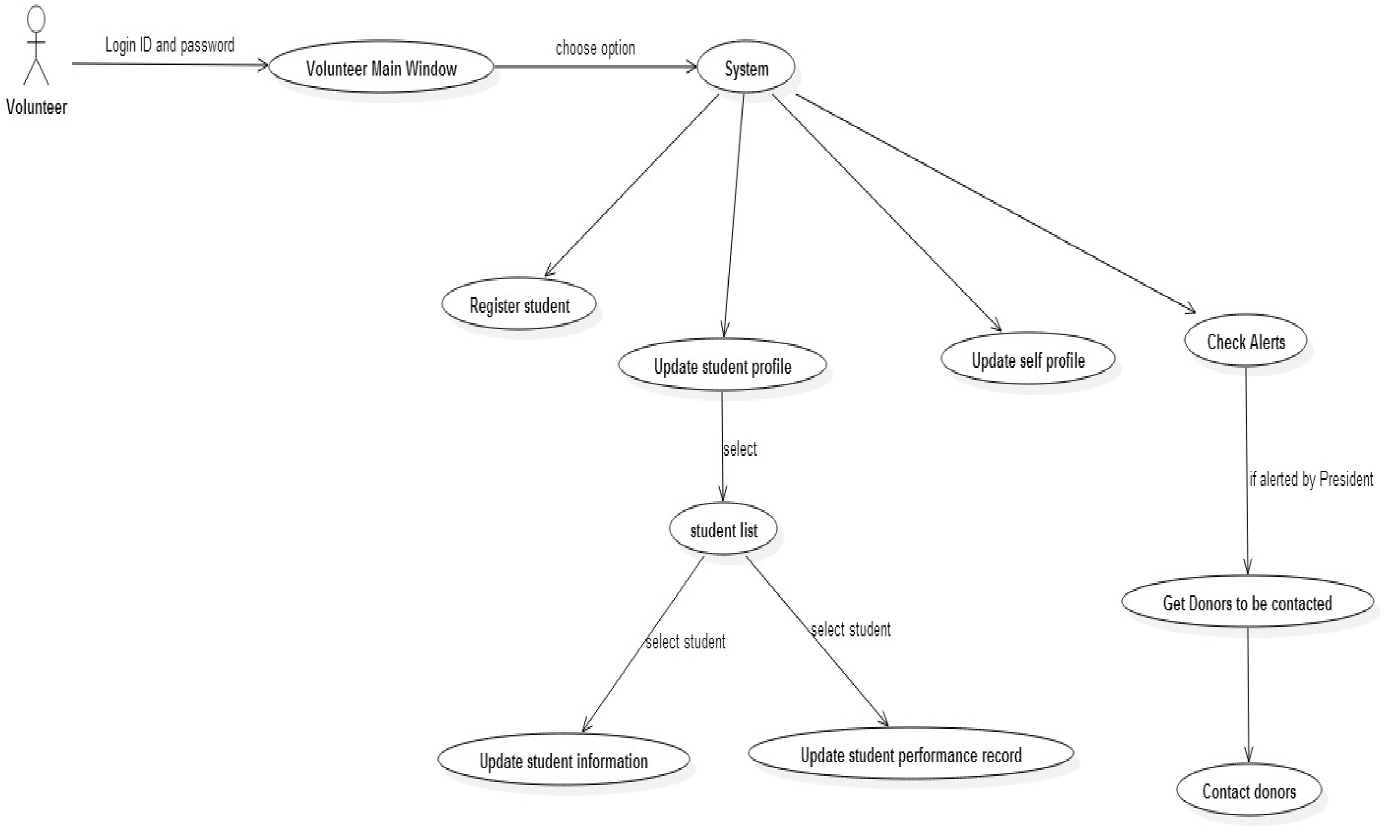
❿ Volunteer register students also updates their performance . They also contact donor for funds.
❿ Donors register and donate money or other items
❿ President maintains funds and keeps record of students, donors, volunteers and the annual expenditure

2.3 Analyzing Stakeholders

The stakeholders are:

- Volunteer
- Donor
- President

2.3.1 User class: Access to Volunteer



<u>Brief description</u>: Volunteer registers a new student and updates the performance.

<u>Step-by-step description:</u>

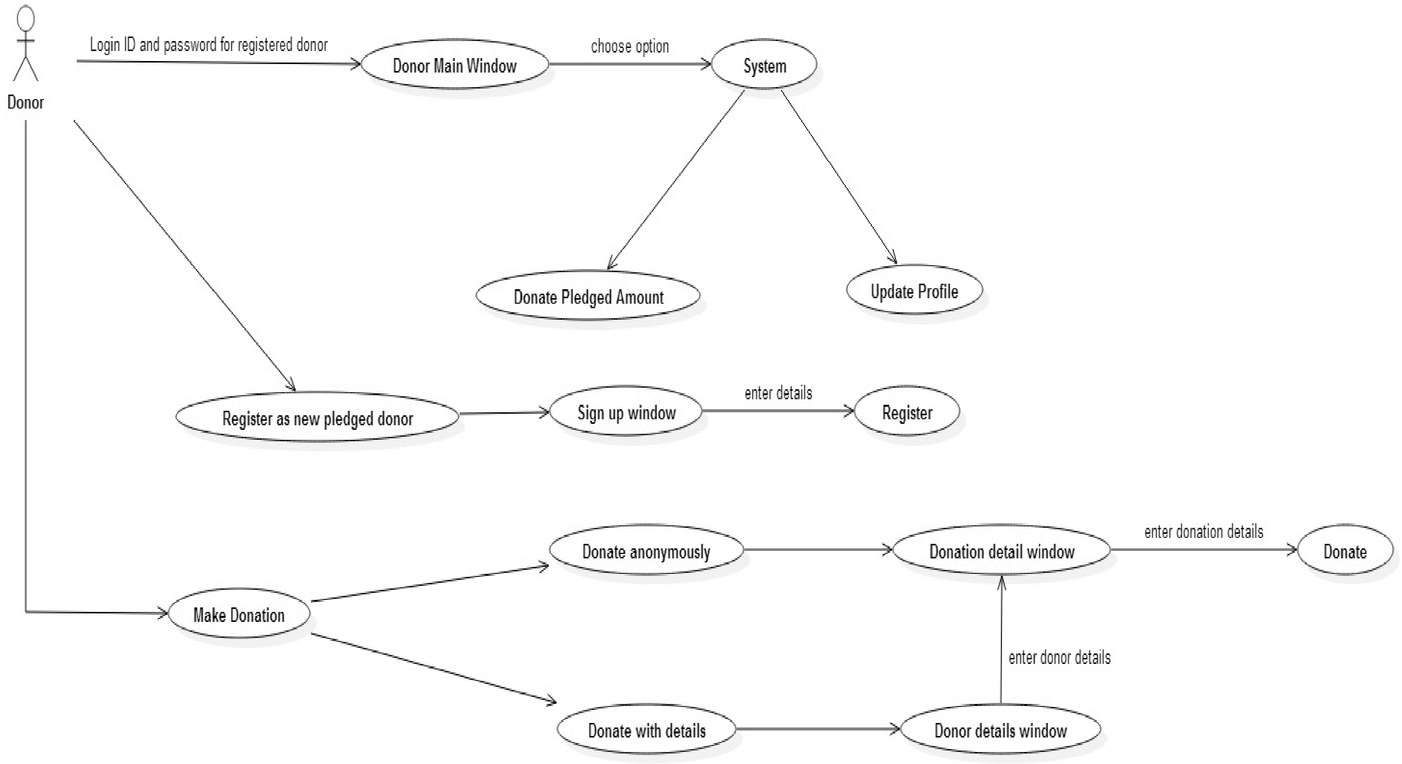
1. Volunteer enters login ID and password
2. Volunteer main window opens
3. Volunteer selects an option among

- Register new student
- Update student profile
- Update self profile

- Check alerts

The volunteer is then re-directed to the particular window.

2.3.2 User class: Access to Donor



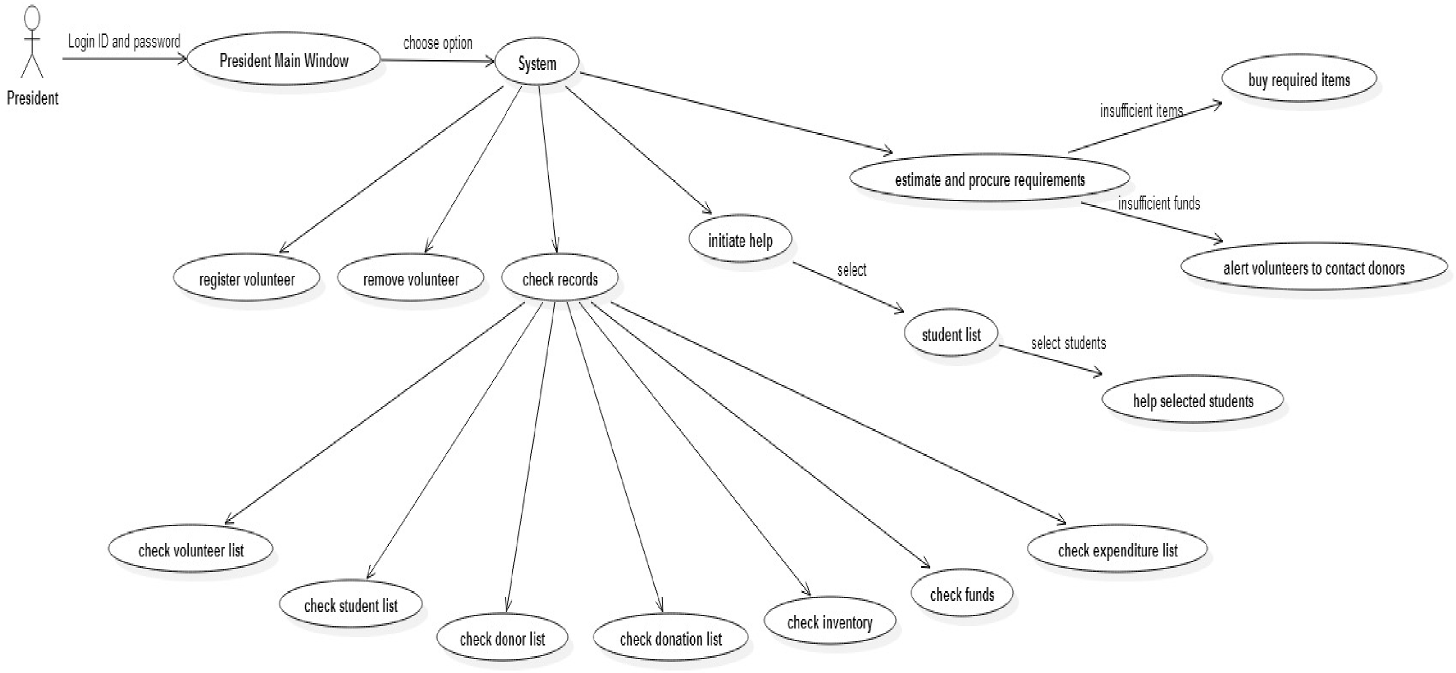
Brief description: Donor can make donation either via money or in-kind or both. Users can register as a new donor.

Step-by-step description:

1. Existing donor will login and main donor window will open. He can also update his profile.
2. Donor can choose what kind of donation he wants to make.
3. User can register as a new donor by selecting "register as a new donor" button and sign-up window opens. On entering details, he is registered. User could also make an

anonymous donation selecting “donate anonymously” . The Donor is re-directed to the Donor main window.

2.3.3 User class: Access to President



<u>Brief description:</u> President can check the records of expenditures, students, volunteers, donors, donations, funds and inventory and allocate help or ask volunteer to contact donors, depending on the sufficiency of the funds, add or remove volunteers, initiate help etc.

<u>Step-by-step description:</u>

1. President will login using his login ID and password and President main window will open.
2. President can check records of students, donors etc by selecting get student list, get donor list and so on.
3. He can estimate funds and check inventory and if funds are insufficient, he will alert the volunteers to contact

donors and if funds are insufficient . If funds are sufficient he will initiate the help to students.
4. He can also maintain an account of annual expenditure.
5. He maintains the record of volunteers, he can add/remove them.

2.4 Defining alternatives

2.4.1 Connection between students and volunteers

- Students can contact volunteers in person by coming to the NGO office.
- Volunteer might visit the students for help.

2.4.2 Connection between volunteers and donors

- Volunteers can contact donors via email, telephonically or personally visiting them.
- Volunteers use NMS to notify the donors.

2.4.3 Hardware Infrastructure

- Although the software is designed to run on Linux and Windows Operating systems, it can also be alternatively designed to run compatibly with MAC OS X systems.
- It can be designed in such a way that instead of requiring a RAM of 512 MB, it requires only 256 MB of RAM.
- Instead of using the internal memory of the system on which NMS is being run, we can use external hard disk of any size given that it satisfies the minimum hardware

requirements as mentioned in SRS.

- NMS can be alternatively designed to run on a 32-bit machine instead of a 64-bit machine.

2.4.4 Software Infrastructure

❿ The database of the system is stored in the files in the form of objects, the passwords stored are stored using encryption. We can use MySQL etc also for this purpose.

❿ Depending on our requirements, cross platform compatibility, customized controls, ease of accessibility and speed, we can use other programming languages which offer a Graphical User Interface (GUI) such as PHP.2.4.5 Security .

❿ Instead of using password protection for login into the system, we can also incorporate Face recognition, a Bar Code scanning system, or a thumb impression recognition system for login of the important stakeholders.

2.5 Defining Criteria to Evaluate

The primary criteria which are to be kept in mind while evaluating the alternatives are:

- Cost of technology
- Cost of infrastructure
- Lifetime of technology
- Stability of technology

2.6 Assessment of Unusual Circumstances

The design should take care of the fact that the data is not lost in any case, be it software or a hardware failure, system going down or any unusual circumstances that might intervene in between the smooth functioning of the NMS. For this, we can design a MASTER system which stores the backup of all the data which is fed into the NMS time to time. This master system would allow us to retrieve data at any point of time and restore the database to its original state.

2.7 Evaluation of Alternatives

2.7.1 Connection between students and volunteers

This alternative will not affect the software.

2.7.2 Connection between volunteers and donors

When volunteers contact donors via notification through software, the software will have to display a message to the donor as soon as the donor enters the login details to notify him.

2.7.3 Hardware Infrastructure

Using an external hard disk to save the database will indirectly imply a backup of the database along with the master system which can be retrieve the data at any point of time desired. Thus, this is better than using the system hard disk to store the data. This also ensures that the software does not any space on the system hard disk. The NMS can be designed for a 32-bit system as well as a 64-bit system. Designing the NMS for a 64-bit system should be

preferred for the graphical interface
concerns.

2.7.4 Software Infrastructure

Any database system can be used for the managing the database of NMS and all have equal priority.

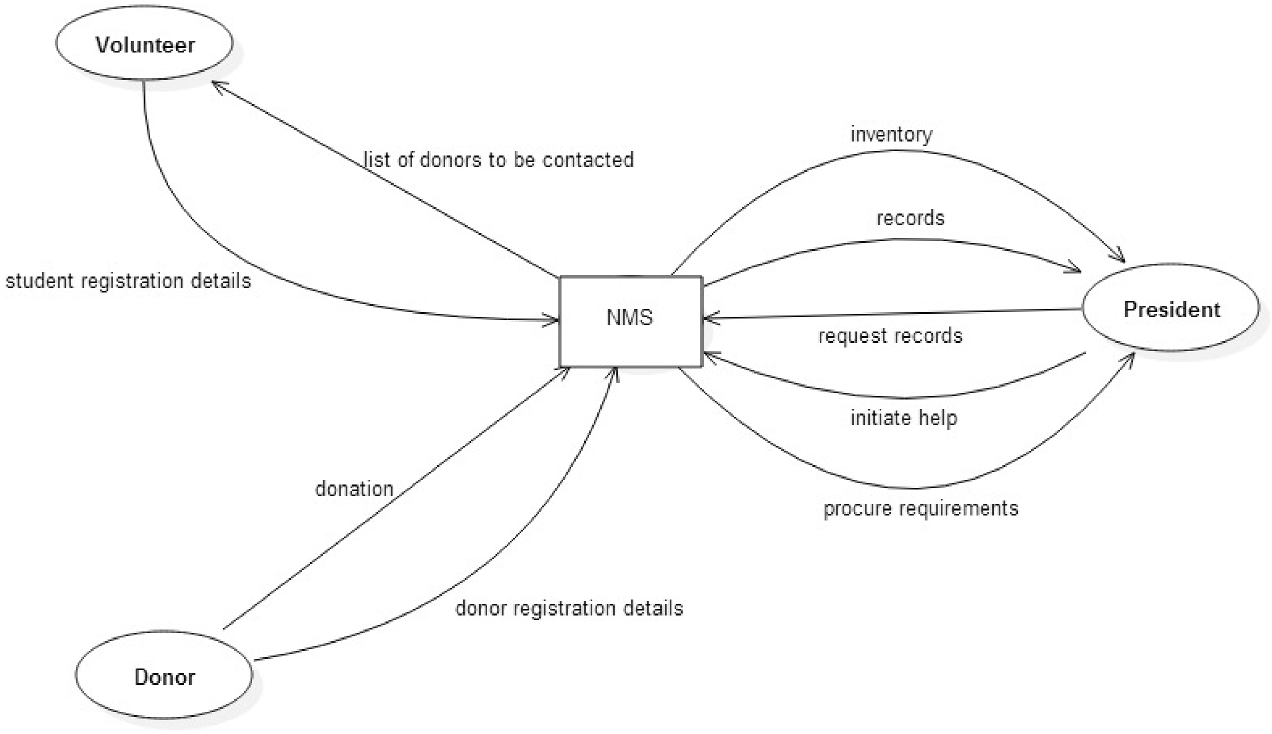
2.8 Report

In the feasibility study, we went through the complete details of the problem. The objectives of the NMS have been laid out and the various scopes have been discussed in detail. Firstly we understood the complete problem and found the various functions that the software performs such as registering students, updating their performance, registering donors, accepting donations, estimation of funds, allocation of funds and maintaining expenditure. Then, various cases were analysed with the help of use case diagrams along with the brief description followed by a step-by- step description of each use case. Each of the use cases were supported by the use case diagrams for ease of understanding. Then, the various alternatives were developed keeping in mind the cost and the lifetime of the components the alternative brings with it and hence the advantages and disadvantages were highlighted. These alternatives included the hardware, software, technology, security and many other aspects which form an integral part of the software and which could be incorporated in the NMS, if desired. The primary criteria for evaluation were expected lifetime, cost, stability, and instability of the technology. The unusual circumstances like loss of data due to hardware or software failure or hacking were taken care of by certain concepts of data backup, cryptography etc.

At last, all the alternatives proposed earlier were analysed in depth and their advantages and boon to the NMS were clearly mentioned. A very vivid comparison was made between the NMS development without the alternatives and the BAS as it would function with the alternatives if incorporated in the software.

3.0 Functional Requirements

3.1 Data Flow : Context Diagram



In this diagram volunteer, donor and president are shown as external entities and the whole NMS is shown as a process.

Inputs:

1. Student registration details
2. Donated item
3. Donor registration details
4. Request records
5. Initiate help

Process

NMS (NGO Management Software)

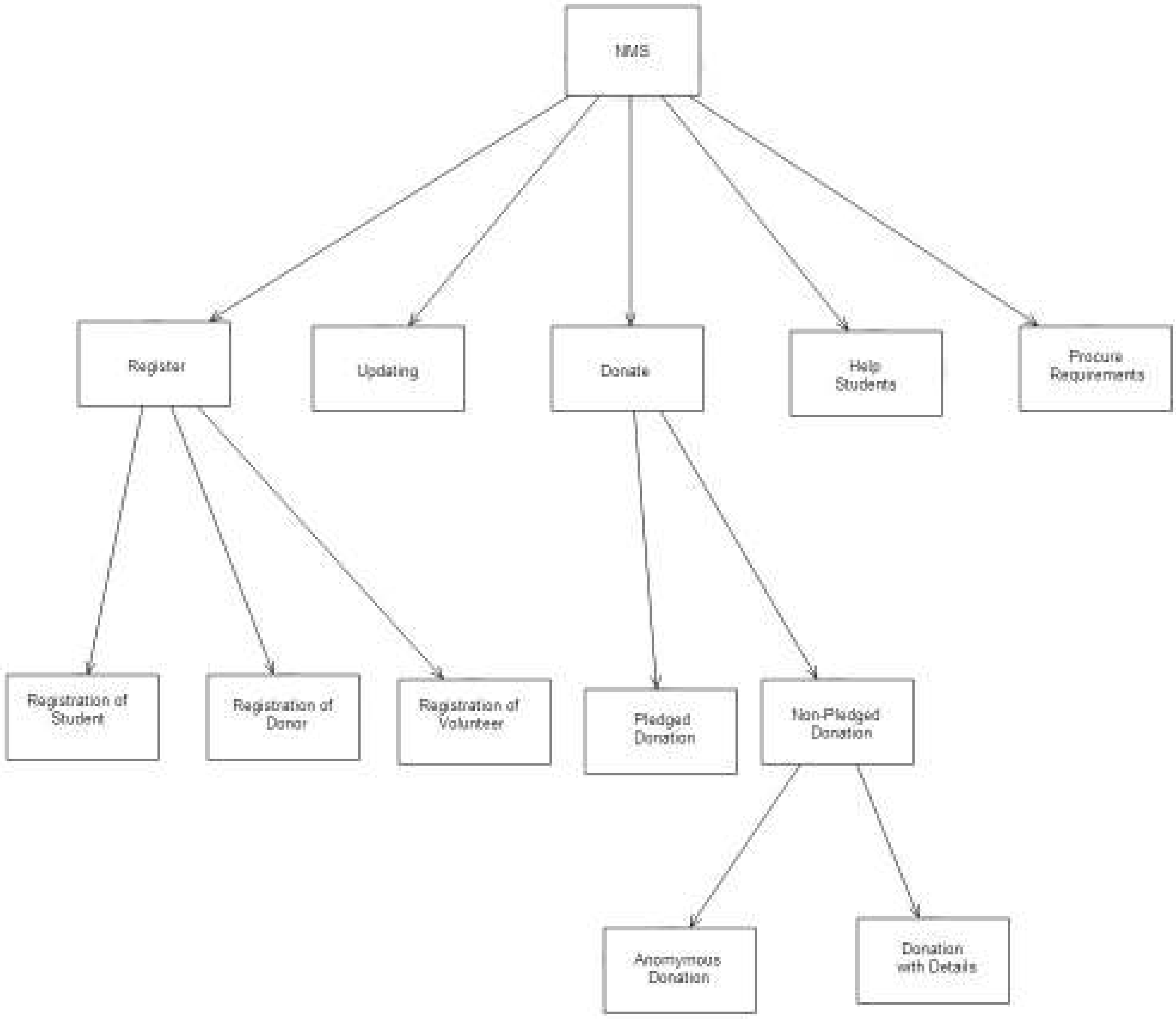
Outputs

1. List of donors to be contacted
2. Inventory
3. Procure Requirements
4. Records

3.2 Data Dictionary

Record : ArrayList<Record>
Inventory : Inventory
Funds : Double
Student List : ArrayList<Student>
Volunteer List : ArrayList<Volunteer>
Donor List : ArrayList<PledgedDonor>
Donation List : ArrayList<Donation>
Expenditure list : ArrayList<Expenditure>

3.3 Structure Chart



3.4 Non-Functional Requirements

• Database Requirements: As mentioned earlier, the NMS stores all its database in files (which are created or are already present). So, if the no. of students or number of donors are more, size of database will increase proportionally and hence there must be space in the hard disk to accommodate the increased size i.e. the size on disk when in use might be greater than what was mentioned in hardware specifications section.

- Legal Requirements: The NMS has a software license agreement according to which any user of the software will be severely punished by law if he is found guilty of distributing copies of NMS to any source.
- Availability of NMS: NMS is available always, as it is offline software, it does not have much issues with maintenance.
- Physical security: If any stakeholder closes the system for any reason, then he is automatically logged out of the system which requires his password to login again. Thus any intruder will not be able to tamper with the system in his/her absence.

3.5 Report

In the requirements analysis section, we started with the functional requirements of the NMS and explained detail it using data flow diagram, Structure chart and data dictionary. The data flow diagram graphically represent the "flow" of data through the information system, modelling the process aspects of the NMS. They are a preliminary step to create an overview of the NMS. Thus the DFDs along with the structure chart and the data dictionary have been used for the visualization of data processing in the NMS, i.e. the structural design of the software being developed. The second part of the requirements analysis deals with the non- functional requirements of the NMS. These include the database requirements, the legal requirements, and the availability of the NMS over a day and the physical security of the software being developed. The various non-functional requirements ensure the delivery of an operable and manageable system which provides the required functionality in a reliable fashion, uninterrupted and with minimal time of interruption even under the unusual circumstances

4.0. Detailed design

4.1. Global System Architecture

The overall system architecture is a 2-tier architecture which includes client at one end and the database at the other. There is no server based middle tier in the software being designed.

4.2. Platform

Minimum Systems Requirements:

Hardware Requirements : Operating System Windows XP/98 or later versions , Linux Processor Pentium II processor or equivalent .

Hard Disk Space: RAM 512 MB

Software Requirements :
JDK 7 or above

4.3 Software Architecture

Object-oriented architecture forms the basis of the RRTS. In this style data representations and their associated

primitive
operations are encapsulated in an abstract data type or object. The components of this style are the objects—or instances of the abstract data types. Objects interact through function and procedure invocations. Two important aspects of this style are :
(a) That an object is responsible for preserving the integrity of its representation (usually by maintaining some invariant over it), and
(b) That the representation is hidden from other objects.
Thus the aspects of OOA mentioned justify our choice.

4.4 Report

Under the detailed design section of the software design, the global system architecture was discussed. The NMS has a 2-tier architecture comprising of the client and the database with no server. Then the platform requirements for the NMS was discussed in terms of the operating system, the processor required, the minimum and recommended hard disk space and RAM requirements, etc.
The software architecture of the NMS was later stated to be of the object-oriented type using JAVA as the core technology. The important aspects of OOD used for the NMS are data abstraction and the preservation of integrity of the software.