**1. Introduction**

Church donations are a significant part of a church, hence there is a need for a good financial records keeping software. The software not only helps in keeping track of the funds, but also makes the church’s secretary work easier. An existing system is used by the secretary to keep records of all accounts and donations. However, the new application needed should to be similar to the existing one, but with additional features to support certain functionalities, especially report generation. The secretary will use the application to enter funds, donations, member data, and generate various reports for both the donors and the church.

**1.1 Project Scope**

The scope of the application is to address the problems faced by the customer using the present application. The application currently being used lacks functionalities and features that are of important to the management of the church and the user. The current graphical user interface is also not friendly to the user of the application. The main function is to develop a user friendly application. The user can record the donations and generate reports.

**1.1.1 Main Objective**

The main objective of the application is to provide a user friendly interface to the church secretary who is responsible for management of the finances of the church. The aim is to develop and provide the user with a good GUI and better options for record keeping. The application is to help the user with generation of various reports like monthly donations per donor and yearly donations among others. These reports are not only needed by the user but by the donors too for the purpose for tax filing. The system is to make capturing of these records easier, efficient and secure.

**1.1.2 Overview of document**

Section 2 describes the user of the system. It describes the various use cases that will be there as well as the sequence diagrams. The rest of the document gives a description of system requirements for both the developers and the client. Other deliverables are also mentioned**.**

**2. Users**

This section describes the user of the system being developed.

**2.1 System User**

The church financial application is used by the church secretary, who is responsible for financial management of church funds. He/she uses the application to keep records of all the donations and generate reports for the donors weekly, monthly and yearly depending on the needs of both the donors and the secretary.

**2.2 Use Cases and Use Case Diagrams**

This section describes the different use cases and scenarios when the system is in use.

**Use cases:**

Use case 1: The secretary account login.

The secretary logs in to the system using the credentials.

* Secretary starts the system for the usage
* Secretary clicks on the login or enter button.
* Secretary logs in to the system.

Extensions:

* System login fails, displays the error message and asks user to enter correct login credentials.
* Contact system admin for credentials if system is locked.

Use case 2. The secretary makes record of the funds.

The secretary uses the system to record the type of funds for the church.

* Secretary selects the fund button.
* Secretary selects the function button to add, update or delete.
* Secretary fills the required fields for recording the fund.
* Secretary submits the details.

Extensions:

* System failure to record, display error message.
* Component failure, display error message with options to correct it.

Use case 3. The secretary makes record of the donations.

The secretary uses the system to record the donations received.

* Secretary selects the donation button.
* Secretary selects the function button to add, update or delete.
* Secretary fills the required fields for recording donations.
* Secretary saves the details.

Extensions:

* System failure to record, display error message.
* Component failure, display error message with options to correct it.

Use case 4. The secretary records member information.

The secretary maintains member information in the system.

* Secretary selects the member button.
* Secretary selects the function button to add, update or delete a member.
* Secretary fills the required fields for recording member.
* Secretary saves the details.

Extensions:

* System failure to save, displays error message.
* Display error details with options to correct it.

Use case 5. The secretary generates reports.

The system is used to generate report for different purposes.

* Secretary selects the report button.
* Secretary fills the required fields and needed report period.
* Secretary selects submit.
* The requested report is generated and displayed.

Extensions:

* Display error message because no record exist.
* Display error message because incorrect period selected.

Use case 6. The secretary logs out from the system.

The secretary exits the system.

* Secretary selects the logout button.
* Secretary successfully logout the system
* System displays login screen.

The figure below is a use case diagram. The main actor in the system is the church secretary/ Registered Client.

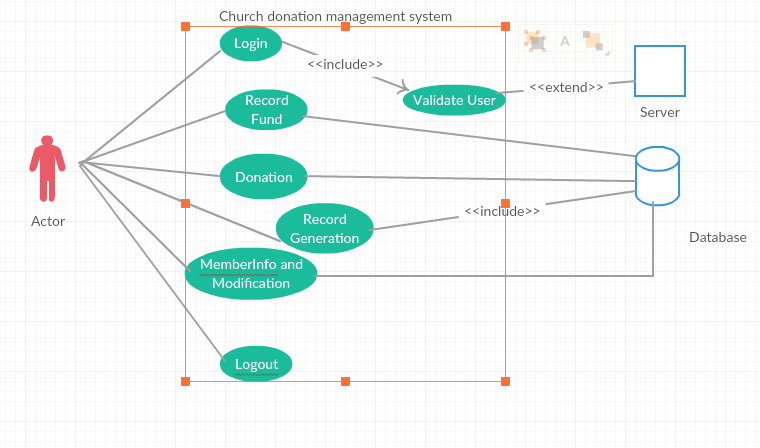


Fig 1: Use case diagram.

The diagram below is a sequence diagram. It shows a typical interaction between the user and system.

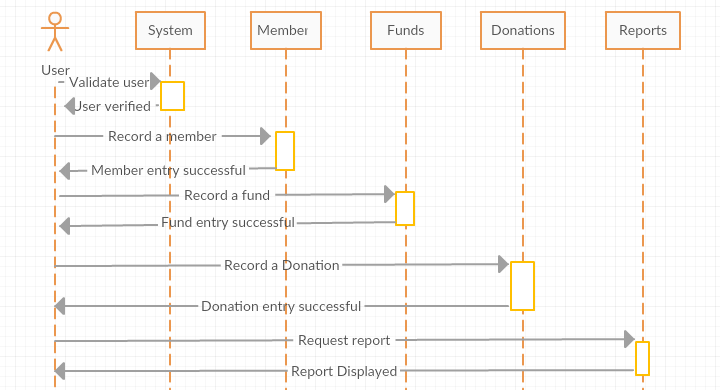


Fig 2: Sequence diagram.

**3. System**

This section describes the system and the tools that are used to develop the application, as well as the system and applications the user is required to have.

The computer system used to develop the project is a computer with the latest available windows operating system, because of its easy availability, is simple to use and offers easily available software.

The application developed is a web based application that is compatible with the main browsers like internet Explorer, Safari, Opera, Mozilla and Chrome.

**Hardware:**

Minimum requirements

Processor : Dual core

Memory Ram : 1 GB

Hard Disk : at least 40 GB

**Software:**

Development : Eclipse

Applications : MySQL Server.

Operating System: Windows 7/8/8.1/10

Web browser : Internet explorer, Chrome, Mozilla Firefox, Opera.

**3.1 Functional Requirements**

**3.1.1 Issues**

This application is going to be a single user software application with good security features. This is because the application is going to handle financial matters of the church and have some confidential about members.

**3.1.2 Major Subsystems or Functions**

The system has several functions. These include handling church member’s information, keeping proper records of donations, and generating customized reports.

The diagram below is a class diagram; it shows the classes, relationships and methods and to be implemented.

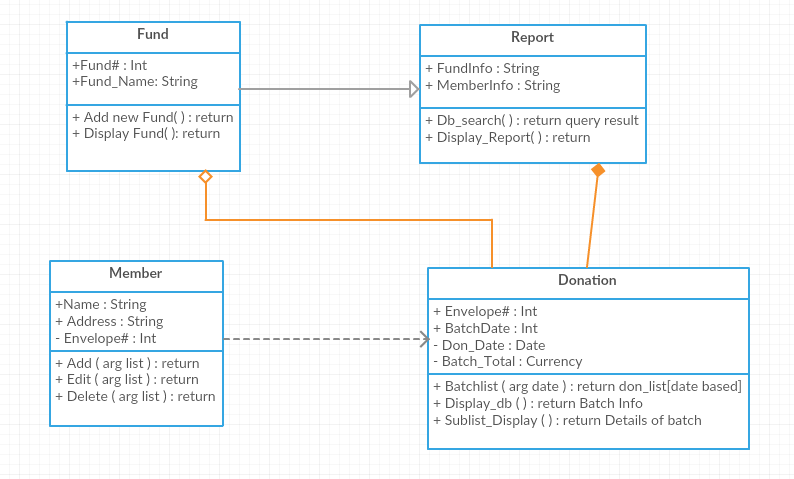


Fig 3: Class diagram

**3.2 User interface specifications**

The application will have a graphical user interface. The user interface is to be determined soon.

**3.3 Non-Functional Requirements: Management**

Proper usage of resources is needed. The work force for developing the project is very important, with less time and limited people to work on the project, these are the tight constraints to handle and finish the project.

The management of the project is a very difficult job to be done. It involves the proper planning of resources, management and co-ordinations between stake holders.

**3.3.1 Technical**

Eclipse is a very powerful development tool used for fast and robust application development.

MySQL is used at the backend of the application for data storage, retrieval and updating.

**3.3.2 Performance**

The software application to be developed is a web based application. The end system should have good hardware and software configuration for smooth functioning of the application.

**3.3.3 Security**

The system should be very secure since financial records are handled by the application which is being developed. For that reason, there shall be strong authentication and validation of users.

**3.4 System Evolution/Maintenance**

The software will require maintenance and updates from time to time. Reason being the user might experience some errors that were not captured during testing. The user might also need some functionality that was not specified during initial requirements specifications. The application will be scalable enough to adapt to changing user needs and requirements.

**4. Other Deliverables Required**

The deliverables for this project are the software code, user manual, requirements document, final plan/report and UML design diagrams.

**5. Glossary**

GUI: Graphical User Interface.

SQL: Structured Query Language.

UML: Unified Modeling Language.

**6. References**

[1]. Larman, C., Applying UML and Patterns: Introduction to Object-Oriented

Analysis and Design and Iterative Development, 3rd edition (October 2004).

[2]. Sommerville, I., Software Engineering 9th edition book, NY, Pearson, 2011.