

**THE OPEN UNIVERSITY OF SRI LANKA**

**DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING**

Fund Management System

Project Report

**EEX3417 – Software Development for Engineers**

Submitted by:

Shahani Farwin Intikab - 616960907

Date of submission- 17th February 2020

# Table of content

[**Introduction**](#_heading=h.wpavgkwx51eb)2

[Club Background](#_heading=h.v2pqc9mswrz) 2

[System overview](#_heading=h.ob56mzj4l479) 2

[**Requirement Analysis**](#_heading=h.cns821g0xidd)4

[**High-level design**](#_heading=h.8wkit9d13brk)5

[**Data Modelling**](#_heading=h.a6e5gxwtkvtl)6

[Normalization](#_heading=h.looueufqr0w1) 8

[**Detail design**](#_heading=h.9uc8q8vs5bbd)9

[**Testing Plan and Results**](#_heading=h.76ntxq82v0qv)10

[Test Plan](#_heading=h.r7dgs9lgclse) 11

[**Conclusion or Discussion**](#_heading=h.5u0u3mxg7sia)18

[Future Works](#_heading=h.wsvs24ny77ix) 18

[**References**](#_heading=h.nibr41fhu89n)19

[**Appendix A - Letter of Certification**](#_heading=h.cpf84pf8d2li)20

# Introduction

A fund management system is responsible for executing a fund's financing strategy and managing its collection trading activities. Fund management is associated with managing the cash flows of a financial institution. There are many types of fund management.

* Mutual Funds
* Trust Funds
* Pension Funds
* Hedge Fund
* Equity fund management

The fund can be managed by one person, by two people as co-managers, or by a team of three or more people as well as by a system. Nowadays people prefer to work with software systems as they make one person's life easy as well as it makes the task or the process reliable. The fund management system can be used by both admin and the members of the fund collection co-operation. A system to manage the process of fund collection does not waste the time of the people. they can stay at any place and could be connected via the system. It is possible to know about the upcoming find programmes and it allows you to get all the details they need to know through the system.

## Club Background

Benzer's social service club, Wattala is a non-political service organization established originally in 2014 in Wattala, Sri Lanka. As of January 2020, it had over 5 local clubs and more than 1000 members. The stated purpose of the Benzers club is to help the poor schools and to ensure the quality of education for the poor students. Moreover, the social service club has more than 100 sponsor bases.

## System overview

As per the client's requirement, the proposed fund management system should contain the following requirements.

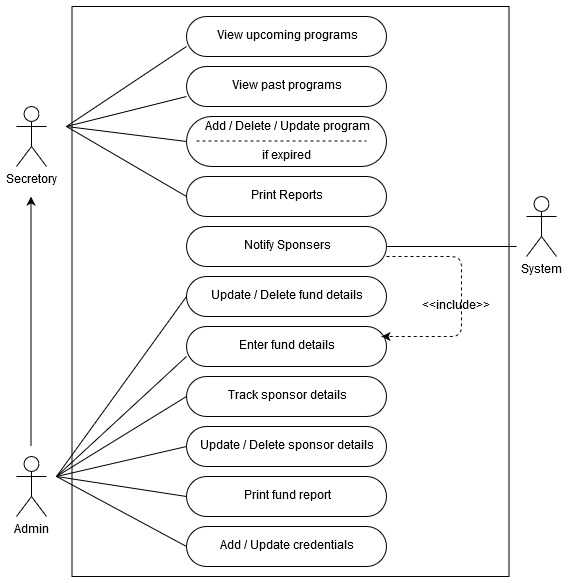
* Ability to manage the funds which were given by the sponsors.
* The system should have a proposer solution to manage the charity program schedules.
* Ability to keep the sponsor details.
* Ability to notify the sponsor whenever the sponsor details are added to the system.

Currently, the Charity organization uses a documentation process to manage funds. The management of the organization keeps all the records of fundraising activities, contributors, and beneficiaries written in books and files. Also, a summarised report of monthly activities should be prepared by the management of the organization every month.

The proposed system solves all these problems. It allows entering data of contributors and receivers. Details about the upcoming and past fundraising activities display in the system as well. Also, the system generates a report every month with a summary of monthly activities.

# Requirement Analysis

As a module of a system designing the use case diagram was designed to represent the functions of the system. It serves to define the desired form of the proposed system and classify interaction between the system and the external entities. Because of the use case diagram, it was easy to understand client requirements and also the priorities of the system as to the user role. The use case diagram of the system will be attached below for the reference.

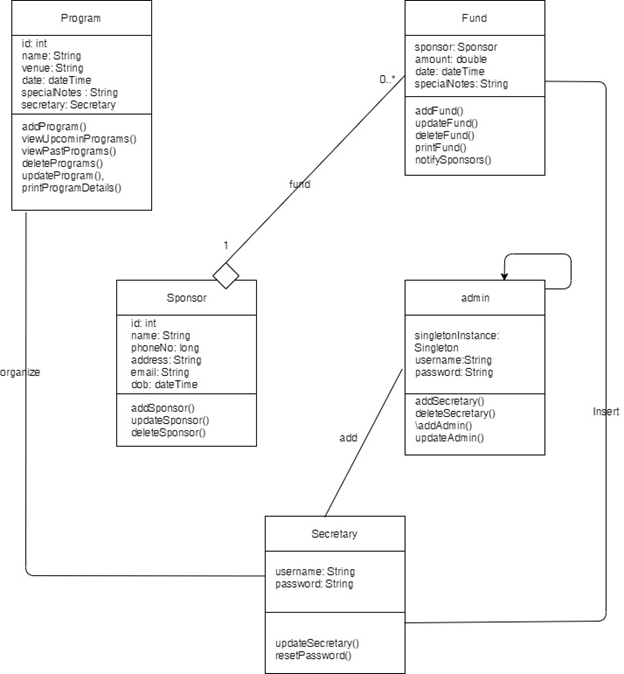


# High-level design

To identify the high-level design for the proposed system Class diagram has been designed. The class diagram is static and because of the class diagram, it enables the ability to easily identify the proposed fund management system. The class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

The prepared class diagram describes the attributes and operations of a class and also the constraints imposed on the system. As well as with the help of the class diagrams it is easy to map the classes directly with object-oriented languages.

Since the class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

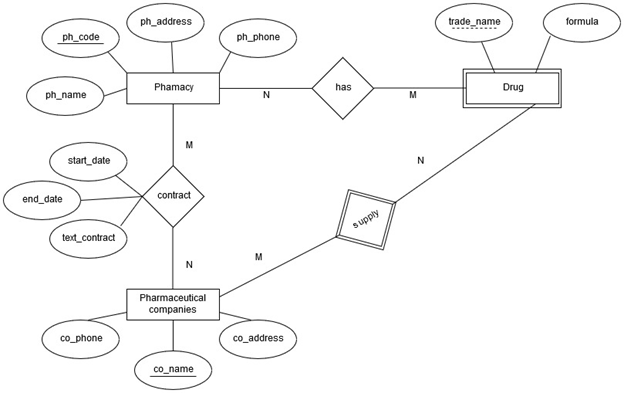


# Data Modeling

"Database structure that will be utilized as the plan to store and handle data", is described as Database design. The reasons for maintaining a decent database design is that Solid application programs can't overcome awful database plans as well as the end-user and the database planner choose what data will be put in the database. In this way, keeping up a proper database design is significant. In advance, ER Diagram was made after collecting data requirements. And SQL lite used as the database system for the proposed system. The reasons for selecting the SQLite as the database system because;

|  |  |
| --- | --- |
| **Advantage** | **Description** |
| **Better performance** | * The application only has to load the data it needs, rather than reading the entire file and holding a complete parse in memory. * Small edits only overwrite the parts of the file that change, reducing write time and wear on SSD drives. |
| **Reduced application cost and complexity** | * No application file I/O code to write and debug. * Content can be accessed and updated using concise SQL queries instead of lengthy and error-prone procedural routines. |
| **Portability** | * The application file is portable across all operating systems, 32-bit and 64-bit and big- and little-endian architectures. * A federation of programs, perhaps all written in different programming languages, can access the same application file with no compatibility concerns. |
| **Reliability** | * Content can be updated continuously and atomically so that little or no work is lost in a power failure or crash. * Bugs are far less likely in SQLite than in custom-written file I/O code. |
| **Accessibility** | * SQLite database content can be viewed using a wide variety of third-party tools. * Content stored in an SQLite database is more likely to be recoverable decades in the future, long after all traces of the original application have been lost. Data lives longer than code. |

The ER diagram of the proposed system will attach below for the reference.



## Normalization

The tables of the above ER diagram is in 3NF, because;

Verify 1NF

* All the tables are holding atomic values

Verify 2NF

* The table is in 1NF (First normal form)
* No non-prime attribute is dependent on the proper subset of any candidate key of the table.

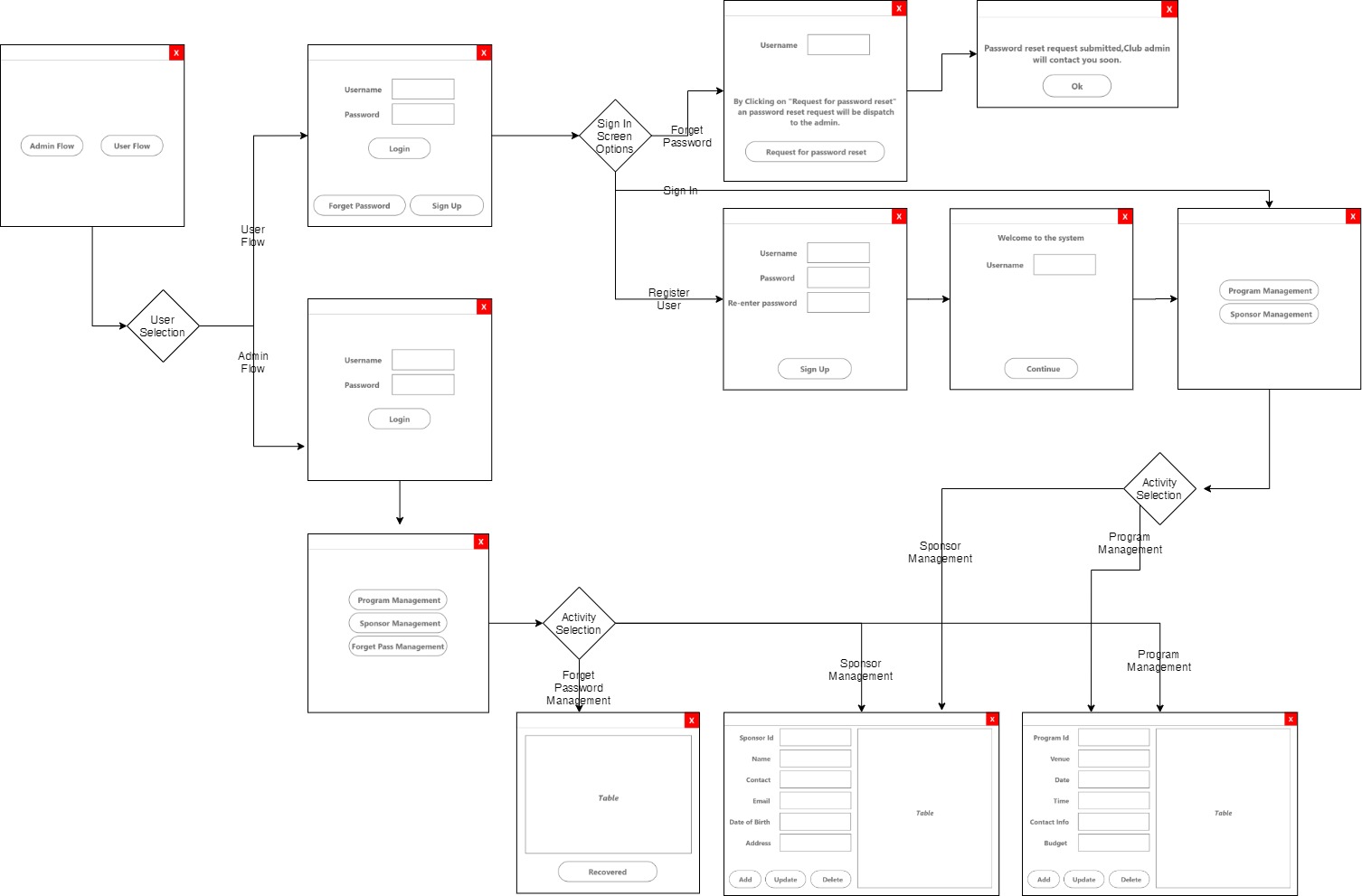
Verify 3NF

* The table must be in 2NF
* [The transitive functional dependency](https://beginnersbook.com/2015/04/transitive-dependency-in-dbms/) of the non-prime attribute on any super key should be removed.

Therefore, as mentioned before, the above-attached ER diagram is in 3NF.

# Detail design

User interface prototypes are an excellent means of exploring your user interface, but unfortunately, it is easy to quickly become bogged down in the details of the user interface and not see the bigger picture. Consequently, you often miss high-level relationships and interactions within your system's UI. User interface-flow diagrams - also called storyboards, interface-flow diagrams, windows navigation diagrams, and context-navigation maps - enable you to model the high-level relationships between major user interface elements and thereby ask fundamental usability questions. Adobe XD has been utilized to create the user interfaces for the proposed system.



# Testing Plan and Results

Testing was done by various types of testing methods. Functional testing and white box testing took priority. White box testing can be defined as a testing method in which the internal structure has been tested while the tester knows it. The main purpose of this white box testing method is to find errors and problems easily.

This was done by creating test cases according to a test plan. Some of the test cases were shown below as samples. This is basically needed to determine if the expected requirements were fulfilled or not. Through this testing, it will be tested all the functions as well as the code segment of the system and whether they work properly as expected or not. Testing was done separately for each and every function because to avoid the difficulty of finding the errors which occur when testing functions together.

Not only the availability of the functions required but also the interfaces and the procedure of the application should be understandable. So the interfaces also tested parallel with the other functions and features.

And then Acceptance testing is also done for the application. Acceptance testing is a kind of testing method which used to test the acceptability of a system. And also this testing was done to test whether the system matches the requirements of the customer. Acceptance testing was done for all the functions including buttons and text fields from start to the end of the Fund Management System. Test cases were written for each and every function. Negative testing was done in priority under the acceptance testing. This was done by giving incorrect values for the text fields as well as the user name, password, mobile number etc. Tested the functions work for incorrect values or not.

During the testing, it helped to find and fix the errors and it gave an idea about how to develop the application more successfully as well as about the future requirements.

## Test Plan

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TC no** | **Section** | **Title** | **Precondition** | **Steps** | **Expected Result** | **Actual Result** | **Severity** | **Priority** | **Comments** |
| 1 | Welcome Screen | Verifications on Welcome screen |  | 1. Navigate to the system.  2. Check whether the page contain two buttons as 'Admin Flow' and 'User Flow' | 1. Should navigate to the system successfully  2. The page should contain the 'Admin Flow' button and 'User Flow' buttons |  | Medium | Medium |  |
| 2 | User Login Screen | Verification on User Login screen | 1. Navigate to the system | 1. Check whether the UI contains a username field and a password field.  2. Check whether the UI contains a login button  3. Verify whether a sign in button is available  4. Verify whether the forget password button is available | 1. The UI should contain a username field and a password field.  2. The UI should contain a login button  3. A sign in button should be available  4. The forget password link should be available | Worked as per expected | Medium | Medium |  |
| 3 | User Login functionality | Verify whether the user is able to log in with valid credentials | 1. Should have valid credentials  2. Navigate to the system  3. Select user flow | 1. Enter valid credentials in username and password fields  2. Click on login button | 1. Should allow to enter credentials  2. Should login to the system successfully  Should navigate to the Activity Selection screen |  | High | High | Users should not be able to login with invalid credentials. Should display an error. |
| 4 | User Sign Up screen | Verification on User Sign up screen | 1. Navigate to the system.  2. Select User flow | 1. Click on the sign up button  2. Check whether the UI contains a username field, a password field and a Re-enter password field  3. Check whether the page contain a sign up button | 1. Should navigate to the Register screen  2. The UI should contain a username field, a password field and a Re-enter password field  3. The page should contain a sign up button |  | Medium | Medium |  |
| 5 | User Sign Up Functionality | Verify whether a user can create a new account | 1. Navigate to the system.  2. Select User flow  3. Navigate to the Register screen | 1. Add valid details to username field, a password field and a Re-enter password field  2. Click on Sign Up button  3. Check whether the page contains the heading as ''Welcome to the system" and the page contain with User name field and Continue button  4. Click on continue button | 1. Should allow to add details  2. Should navigate to the ''Welcome to the system" page. Page should contain the user name field. User name field should automatically generate the registered user name. Should contain the continue button  3. Should navigate to the activity selection screen |  | High | High |  |
| 6 | User Activity Selection Screen | Verification on User Activity Selection screen | 1. Should have a user account  2. Login to the system in user flow  3. Go to activity selection screen | 1. Check whether the page includes two buttons as below  - Programmed management  - Sponsor Management | 1. The user activity selection screen should contain the Programmed Management button and Sponsor management button |  | High | Medium |  |
| 7 | Admin Login Screen | Verification on Admin Login screen | 1. Navigate to the system | 1. Check whether the UI contains a username field and a password field.  2. Check whether the UI contains a login button | 1. The UI should contain a username field and a password field.  2. The UI should contain a login button |  | Medium | Medium |  |
| 8 | Admin Login functionality | Verify whether the Admin is able to log in with valid credentials | 1. Should have valid credentials  2. Navigate to the system  3. Select admin flow | 1. Enter valid credentials in username and password fields  2. Click on login button | 1. Should allow to enter credentials  2. Should login to the system successfully  Should navigate to the Admin Activity Selection screen |  | High | High |  |
| 9 | Admin Activity Selection Screen | Verification on Admin Activity Selection screen | 1. Should have a Admin account  2. Login to the system in Admin flow  3. Go to Admin activity selection screen | 1. Check whether the page includes buttons as below  - Programme management  - Sponsor Management  - Forget Pass Management | 1. The user activity selection screen should contain the Programme Management button, Sponsor management button and Forget Pass Management button |  | High | Medium |  |
| 10 | User forgot Password screen | Verification on User forgot Password screen | 1. User should be already registered  2. Navigate to the system | 1. Click on Forget Password button  2.Check whether a field contains to enter the user name  3. Type the user name  4. Check whether message displays as 'By Clicking on ''Request for password reset'' and password reset request will be dispatch to the admin'  5. Check whether the ''Request for password reset'' button is available  6. Click on the ''Request for password reset'' button | 1. Should navigate to forget password screen  2. Should contain a field to enter the user name  3. Should allow to type the user name  4. A message should display as 'By Clicking on ''Request for password reset'' and password reset request will be dispatch to the admin'  5. The ''Request for password reset'' button should be available  6. Should display a pop up with the below message  “Password reset request submitted. Club admin will contact you soon”. |  | High | Low |  |

# Conclusion or Discussion

In this report, the background of the Benzers social club along with the introduction of the proposed fund management system is provided.

Under the second chapter of the report, System requirements were described by including use case diagram of the proposed fund management system. As to the Use case diagram, two main characters who are having the ability to use the system were identified along with their activities.

Thereafter high-level design for the proposed system was described using the built class diagram of the proposed system. Moreover, the data modelling chapter is describing the database design of the system and the database management system used for the proposed desktop application. And also the reasons to select the DBMS were provided.

In the Detail design chapter, the screen flow for the system is provided with the necessary system screenshots. As well as under the Test results, the test case documents and the test results will be provided for the reference.

## Future Works

As part of future work, A web portal will be implemented which opens the ability of online registration with the club, realtime sponsor notifying, event live streaming.

# References

Investopedia. (2020). *Funds Management*. [online] Available at: https://www.investopedia.com/terms/f/funds-management.asp [Accessed 03 Feb. 2020].

Smartdraw.com. (2020). *Use Case Diagrams - Use Case Diagrams Online, Examples, and Tools*. [online] Available at: https://www.smartdraw.com/use-case-diagram/ [Accessed 24 Jan. 2020].

GeeksforGeeks. (2020). *Types of Software Testing - GeeksforGeeks*. [online] Available at: https://www.geeksforgeeks.org/types-software-testing/ [Accessed 10 Feb. 2020].

Sqlite.org. (2020). *Benefits of SQLite As A File Format*. [online] Available at: https://www.sqlite.org/aff\_short.html [Accessed 10 Feb. 2020].

Guru99.com. (2020). *What is WHITE Box Testing? Techniques, Example, Types & Tools*. [online] Available at: https://www.guru99.com/white-box-testing.html [Accessed 12 Feb. 2020].

Guru99.com. (2020). UML Class Diagram Tutorial with Examples. [online] Available at: https://www.guru99.com/uml-class-diagram.html [Accessed 12 Feb. 2020].

# Appendix A - Letter of Certification

