UNIVERSITY OF NAIROBI
SCHOOL OF COMPUTING AND INFORMATICS

SECURE COMPUTERISED FINANCIAL ACCOUNTING

BY

SIMEON KENGERE OSIEMO

THIS PROJECT HAS BEEN SUBMITTED IN PARTIAL FULFILMENT FOR THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF SCIENCE IN COMPUTER SCIENCE

APRIL 2019

DECLARATION

I hereby declare that this project is entirely my own work. All work as shown and documented in this report are mine except where indicated otherwise. I also affirm that this project has not been presented in any other University award.

Signature: ………………………………….. Date: ……………………………………….

Name: Osiemo Simeon Kengere

Admission Number: P15/100005/2017

This project has been submitted as partial fulfilment of the requirements of the Bachelor of Science in Computer Science of the University of Nairobi. It has been done with full support of my supervisor.

Signature: ………………………………………….. Date: ……………………………………………

Name: Dr Elisha Tonye Opiyo

DEDICATION

This project is dedicated to my family who have seen me through my undertaking of this project. Their motivation, love and support have been a great stepping stone towards the completion and achievement of the objectives and goals of this project. Many thanks to my parents who have supported me in all aspects and especially my education at the University of Nairobi. Furthermore, I thank the staff and lecturers of the School of Computing and Informatics for their dedication in inculcating knowledge in me which has greatly facilitated and conception, construction and completion of this project. Not to forget my fellow colleagues who have criticized and helped me correct the project until its completion. I could not have made it on time without their help. That is why I wish to thank them greatly. Finally, I thank God for the energy, strength, knowledge and resources that have enabled me to achieve all the objectives of the project. Forever I will remain grateful to Him for He has always seen my through good and tough times.

ABSTRACT

Despite Financial Accounting being a critical part of a business, most businesses suffer from unreliable accounting reports most of the time due to unethical behaviours practised by accountants thus rendering huge losses for companies and businesses. These behaviours may cause stakeholders to lack faith and trust in the companies therefore rendering a poor relationship between the two parties. This can lead to potential losses for a company since external support may be limited due to lack of trust in the companies affected by fraud. One method in which fraud is committed is by disobeying the matching concept rule of accounting, which states that “Matching concept is an accounting practice whereby firms recognize revenues and their related expenses in the same accounting period. Firms report "revenues," that is, along with the "expenses" that brought them.” This project is intended to cover some of the loopholes that accountants use to manipulate financial data in committing fraud. The project was intended for all companies that handle their financial transactions.

ACKNOWLEDGEMENT

I greatly thank my supervisor, Dr. Elisha Opiyo for his support, supervision, guidance and advice as I undertook the project. His knowledge and suggestions greatly played a role in the effective achievement of the system objectives as well as providing possible objectives of the system.

Contents

TABLE OF FIGURES

LIST OF TABLES:

CHAPTER 1: INTRODUCTION

1.1 Background

Financial accounting is a specialized branch of accounting that keeps track of a company's financial transactions. Using standardized guidelines, the transactions are recorded, summarized, and presented in a financial report or financial statement such as an income statement or a balance sheet. It is the process in which the company records and reports all the financial data that go in and out of its business operations on a routine schedule

When it comes to computers in accounting, they can perform different accounting functions, for example a billing machine. This machine is capable of computing discount, adding net total and posting the requisite data to the relevant accounts. With substantial increase in the number of transactions, computers have been developed to store and process accounting data with greater speed and accuracy. The use of computers has been significantly influenced by advancements in technology and its acceptability by people.

In addition, computerised Financial Accounting is needed for the following reasons:

1. **Labour Saving:**

Labour saving is the main aim of introduction of computers in accounting. It refers to annual savings in labour cost or increase in the volume of work handled by the existing staff.

2. Time Saving/ Speed:

Using accounting software, the entire process of preparing accounts becomes faster. Furthermore, statements and reports can be generated instantly at the click of a button, especially when important jobs should be completed in a specified time frame such as preparation of statement of accounts. Managers do not have to wait for hours, even days, to lay their hands on an important report.

3. Accuracy

The accounting system is designed to be accurate to the minutest detail. Once the data is entered into the system, all the calculations, including additions and subtractions, are done automatically by software.

4. Cost Effective

Since using computerized accounting is more efficient than paper-based accounting, which costs more in purchase of paper and stationery, it’s a better option to choose a computerised means whenever considering a cost-effective measure in financial accounting.

5. Data access

By using accounting software, it becomes much easier for different individuals to access accounting data outside of the office, securely. This is particularly true if an online accounting solution is being used.

1.2 Problem Definition

Accountants report incomplete information to the extent that shareholders and potential investors will never know how their firms is performing. As much as the current existing systems try to solve specific problems that are experienced in the field of accounting, whereas some other systems try to cover the loopholes that unethical accountants use to commit fraud, they are developed on the basis of the experiences of the team of developers. As such, it may not be able to deal with a specific problem that may arise.

1.3 Objectives

The main aim of this project is to create a computerised accounting information system that is secure.
In addition, the following are the system development techniques that will be used together with some of their objectives:

1. **Requirements Specification**
- Create a requirements report for the secure computerised financial accounting system

2. **System Design**
- Design a secure computerised financial accounting system

3. **System Construction**
- Construct a secure computerised financial accounting system

4. **Testing phase**
- Test the secure computerised financial accounting system

5. **Maintenance phase**
- Evaluate the secure computerised accounting system

1.4 Justification

Financial Accounting is an important aspect for the survival of any organisation or business, because finance is what revolves around every existing business today. It helps to ascertain the results of the financial transactions of a business concern.

Every business concern is interested to know its operating results at the end of a particular period. In this case, the amount of profit or loss can be ascertained by preparing an income statement with the help of the proposed system.

For running a business successfully a businessman is to acquire various assets like land, building, machinery etc. He is to face various debts and liabilities like accounts payable, notes payable, loan, bank overdraft etc. side by side with acquisition of assets. The actual position of these debts-liabilities, property, and assets can be ascertained through the proper keeping of accounts in a computerised format. A businessman can take the right steps for controlling the quantity of assets decrease and liability increase, hopefully, with the help of the capabilities from the proposed system.

CHAPTER 2: LITERATURE REVIEW

Financial Accounting is the field of accounting concerned with the summary, analysis and reporting of financial transactions pertaining to a business. This involves the preparation of financial statements available for public consumption. Stockholders, suppliers, banks, employees, government agencies, business owners, and other stakeholders are examples of people interested in receiving such information for decision making purposes. Financial accounting is governed by both local and international accounting standards.

Financial Accounting is a way for businesses to keep track of their operations, but also to provide a clear view of their financial health. By providing data through a variety of statements including the balance sheet and income statement, a company can give investors and lenders more power in their decision making.

If Financial Accounting is going to be useful, a company's reports need to be credible, easy to understand and comparable to those of other companies. To this end, Financial Accounting follows a common set of rules called accounting standards, such as International Accounting Standards (IASs) and International Financial Reporting Standards (IFRSs). In Kenya, a body called Kenya Accountants and Secretaries National Examination Board (KASNEB) is responsible for training people to become professional accountants locally, and within the East African region.

As stated earlier, financial Accounting generates quite a number of general purpose, external financial statements, some of which include:

1. Income Statement / Profit and loss statement
2. Balance Sheet / Statement of financial position
3. Statement of Cash Flows

Income Statement reports a company's profitability during a specified period of time, usually referred to as a financial year and it could range from one year, to one month, to one week, or any other time interval chosen by the company. The main components of the income statement are revenues, expenses, gains, and losses. Revenues include: sales and interest revenue. Expenses include the cost of goods sold, operating expenses (such as salaries, rent, utilities, advertising), and no operating expenses (such as interest expense).

Balance Sheet is organised into two parts: (i) Assets, and the second part is further organised into: (ii) Equity/ Capital (iii) Liabilities.
It follows a basic principle, that states: **ASSETS= CAPITAL+LIABILITIES**

Statement of Cash Flows that explains the change in a company's cash (and cash equivalents) during the time interval indicated in the heading of the statement. The change is divided into three parts: (1) operating activities, (2) investing activities, and (3) financing activities.
The operating activities section explains how a company's cash (and cash equivalents) have changed due to operations. Investing activities refer to amounts spent or received in transactions involving long-term assets. The financing activities section reports such things as cash received through the issuance of long-term debt, the issuance of stock, or money spent to retire long-term liabilities.

Most of the popular accounting systems developed have been tailored to specific Ministries, Departments and Agencies. Accounting information systems are already in use by most organisations in order to support decision systems, communication, knowledge management, as well as many others.

Existing Accounting Software

1. Quick Books

Manages the finances of a small business. QuickBooks allows you to have instant access to customers, employees as well as vendor information. Through this software, accountants can create invoices, track debtors and sales as well as company expenses. It is one of the most commonly used accounting software in Kenya. Other key future include, online banking, payroll assistance, e- payment and reconciliation of banking transactions.

2. Sage Pastel

Sage pastel is one of the most ideal software for medium sized businesses. It provides for innovation, flexibility and power. It has a multi-user and multi-currency capabilities. The software offers the ability to add modules which means good news as your business grows. This software is affordable and flexible.

3. Microsoft Navision

The software has different functionality modules that include, general ledger, accounts payables, receivables, inventory and management of purchase orders.

4. Oracle Financials

This financial management software includes aspects like accounts payable, general ledger, accounts receivable, cash management and fixed assets.

The figure below shows a high level diagram of the proposed system:

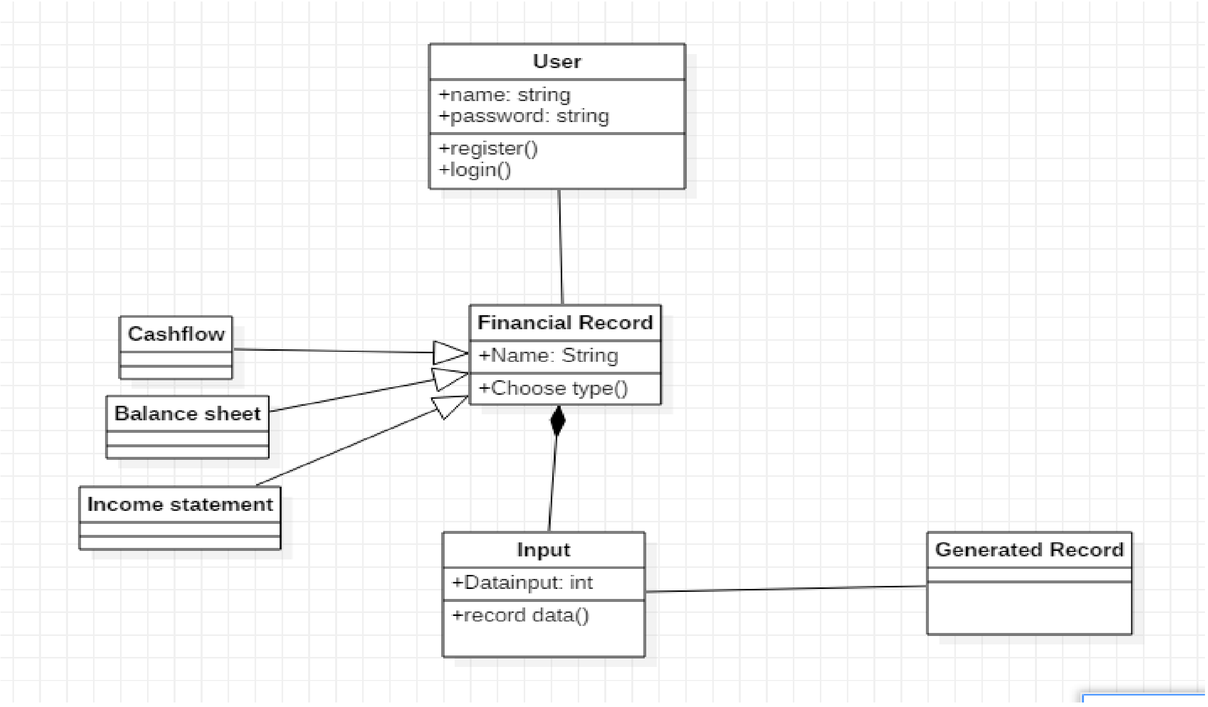


Figure 1: High level diagram of the proposed system

CHAPTER 3: METHODOLOGY

System Development Methodology

During the development of this system, the modified waterfall model will be used. The traditional waterfall is a methodology that follows a sequential process in which the phases: problem definition, requirement analysis, system design, implementation, testing and maintenance are executed.

Moreover, the modified waterfall model is very flexible. More than one stage can be done at a time and allows for review and repetition of previous faulty stages.

This model will be used because of the following reasons: Requirements are very well documented, clear and fixed, Technology is understood and is not dynamic, There are no ambiguous requirements, Ample resources with required expertise are available to support this project, Clearly defined stages, Easy to arrange tasks, Phases are processed and completed one at a time, and it's Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process

During the development of the project, the following activities will be carried out at each stage:

1. Requirement analysis

The types of activity that will be conducted include:

a) Eliciting requirements will be conducted in the following ways: using some of the information gathering techniques which are: interviewing, issuing questionnaires, and researching the internet. Interviewing and issuing questionnaires will involve employees of some companies in Kenya. This will help determine what the requirements will consist of in detail.

b) Analysing requirements after conducting requirements gathering to determine whether the stated requirements are unclear, incomplete, ambiguous, or contradictory.

c) Evaluate system for feasibility - specifically technical feasibility and operational feasibility.

d) Requirements modelling - requirements will be documented here.

2. Design

The proposed system will require user input, in a desktop or laptop environment, in order to process the requested information of type integer, character, or both. This is the data that will be used for processing and output. In addition, a graphical user interface will be used for interaction between the user and the program. The proposed program will run on a desktop environment.

3. Implementation

In order to achieve this, some of the resources needed will include: **KASNEB CPA part 1 section 1 Financial Accounting notes,** in order to create a system that follows the accounting standards that Kenya already follows, **Library resources,** where other books related to accounting will be used to create the proposed system, **Internet resources,** in cases where some data is unclear and further clarification is needed, and a computer for coding and testing the program, **a computer** for coding together with an I.D.E, preferably Code Blocks for writing the code in C++, and a G.U.I

During coding, with help of inputs from the system design, the system will first be developed in small programs called units, which will be integrated in the next phase. Each unit will be developed and tested for its functionality, which is referred to as Unit Testing.

4. Testing

All the units developed in the implementation phase will be integrated into a system after testing of each unit. Post integration the entire system will be tested for any faults and failures.

5. Maintenance

Once the system is operational, it will be assessed. Aspects of this phase include: implementation of changes that the software might undergo over a period of time, or implementation of new requirements after the software is deployed at the customer location. It will also include handling the residual errors that may exist in the software even after the testing phase. This phase will also monitors system performance, rectifies bugs and requested changes are made.

Summary

The proposed system was expected to generate financial records upon request by the accountant, which works under a set of well-defined concepts and principles.

In addition, it was expected to encourage the user to appreciate computerised accounting and discourage the manual method of recording in manual spreadsheets, due to unfavourable reasons that the proposed system will find favourable, and also to ensure that other users of financial statements get sufficient information necessary to enable them make informed decisions for the business.

In conclusion, the proposed system will help in recording, manipulating and generating financial records. It will provide a secure platform to perform these actions.

CHAPTER 4: SYSTEM ANALYSIS

The existing system

Most Accounting software products are ideal for micro-enterprises and sole proprietorship with a few employees. Entry-level accounting provides standard accounting functions like accounts payables and accounts receivables and basic financial reporting.

The existing accounting software, such as quickbooks online, caters to individual accountants and small businesses. The solution helps with bank reconciliation, tracking expenses, drafting invoices and monitoring financial reports. It enables users to download and reconcile credit card transactions. It allows businesses to organize weekly timesheets and track bank deposits. Users can take photos of bills and submit them as expense proofs or share with the concerned stakeholders.

The figure below shows a high level diagram of the existing system:

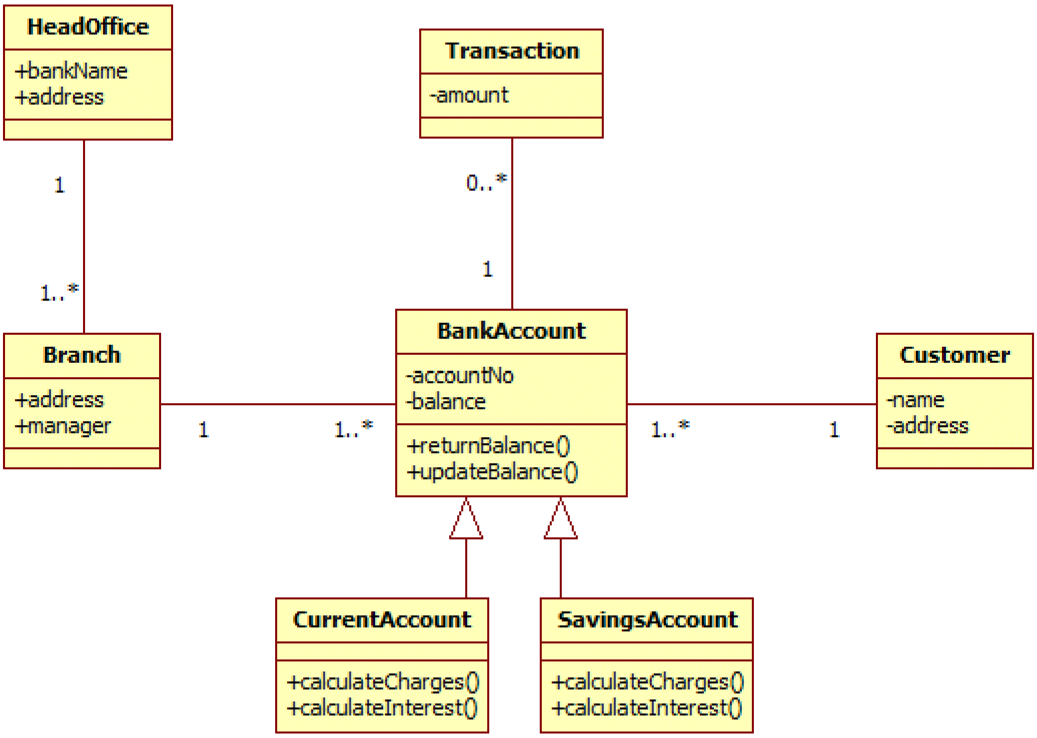


Figure 2: High level diagram of the existing system

Introduction

The proposed system was examined to understand its constraints, procedures and functions. Information was gathered to figure out how the system can be built to suit user needs and also defining the system requirements.

Feasibility Study

Feasibility was conducted to measure how the development of the proposed system will be beneficial or practical to the target group of leaders.

Operational Feasibility

The proposed system will be created in accordance to the accounting standards to ensure security of financial records. It will also be able to reflect the financial health of a company in terms of Assets and Liabilities, and necessary controls will be in place to accurately record and process data. It will also generate financial statements, which will help in decision making.

This will ensure that a company's financial records are safe and accurate.

Technical Feasibility

The proposed system is to be a desktop application whose users are expected to be accountants to facilitate proper reporting. Expertise required to build the system include a knowledge in C++ and the resources that will be used to build the system are QT application, and a windows based 64 bit computer.

Schedule feasibility

Four weeks are enough to build the system to completion. This is as there exists the technical expertise required for building the system.

Economic Feasibility

No purchases will be needed in development. The free QT application which is available online will be used. During its use, however, a monthly fee of around Ksh. 10,000 per month. This is because QT charges commercial based applications.

Minimal training is needed to master the system so training cost will be reasonably low.

Requirements Analysis

Existing systems were inspected to determine if there was a need for a new system. Interviews were conducted to compare existing systems with the proposed system. Other mechanisms such as observation and research were conducted to further understand the need for a new system.

The parties interviewed were employees from various companies such as: CBA bank and Remsons Limited.

The following were the results of the interviews:

a.) Remsons Limited:

They use Quickbooks application which has the following features:

Multi monitor support, Payroll liability Payment reminder, Cash/Accrual Toggle on reports, Mobile inventory scanning, Sales order fulfilment workflow.

From their point of view, they experience the following drawbacks from the Quickbooks system:

Has a lot of unnecessary capabilities, One has to navigate through many areas to get to where they want, Have to pay for subscriptions at costly prices. Prices range from Ksh. 115500 to 184800 per year, Needs intensive training to understand how to operate the system, Instability/ System Crashes, Limited reporting and transparency of their business health, Problematic when addressing robust inventory management concerns.*(Source: Maurine Kemunto, Office number: 0208600415)*

b.) Commercial Bank of Africa, two rivers branch
They use Oracle Financials application which has the following features:
Supports most Accounting modules such as Accounts payables, Accounts receivables, Customer relationship management, Human resources and payroll.

From their point of view, they experience the following drawbacks from the Oracle Financials system:
Poor customer service, Most modules, e.g.: Oracle iProcurement which touches on end users is not user friendly, In case of issues with the system, it becomes difficult to debug and resolve the issue, Has issues connecting to the internet in addition to the software being incompatible with many web browsers, This software is expensive compare to their margin cost for this software is impossible are not user friendly, It is can be confusing at times to retrieve data, Customization is difficult and requires hiring experienced personnel to customize, The system is incompatible with most browsing software, The system is costly. The monthly subscription fee ranges from Ksh. 15,000 to 17,500, Requires intensive user training for the user to be comfortable using the software, Upgraded versions e.g: version 11, from version 10 don't resolve problems it's supposed to, Support requests take long to respond, Transactional entry forms are more time consuming.
(Source: Sarah Mokeira, Branch relationship manager. Mobile: 0725892585)

Weak areas of the existing accounting systems:

They include: instability and system crashes, Lack of direct professional support, limitations on the number of users, lack of key reports in accounting, lack of business specific features, double entry and keying errors across systems and departments.

REQUIREMENTS SPECIFICATION FOR THE PROPOSED SYSTEM

Functional Requirements for the proposed secure financial accounting system:

1. The proposed system will enable the user to be able to record the available Assets of the company, and its liabilities.
2. The proposed system should help the user generate three fundamental financial records: statement of cash flows, income statement and statement of financial position.
3. The proposed system should be able to reflect a company's financial health in form of a graph.
4. The proposed system should give financial analysis based on current financial status
5. The proposed system will record transactions with the help of the user.
6. The proposed system will enable the user to update any transactions that might have occurred.
7. The proposed system will provide a means for a user to be registered through the administrator
8. The proposed system will provide a way of removing users from the system.

Non Functional Requirements for the proposed system:

1. The proposed system will require user login to validate the user before any data can be manipulated.
2. The proposed system will be reliable- this is because some functions will be automated.
3. The proposed system will generate accurate records.
4. The proposed system will be user friendly.

Use Case Model

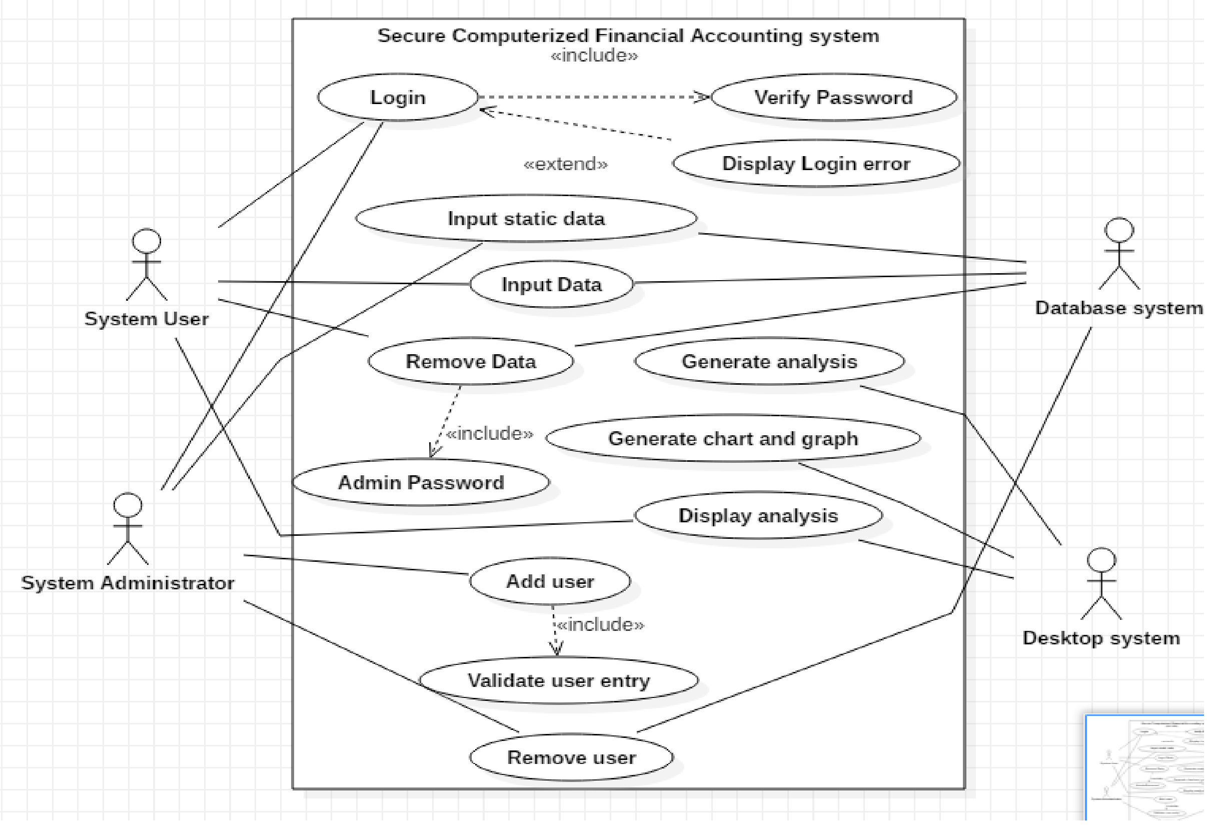


Figure 3: Use case model

From the use case model, the following use cases are derived:

Login

The following diagram shows a use case for login

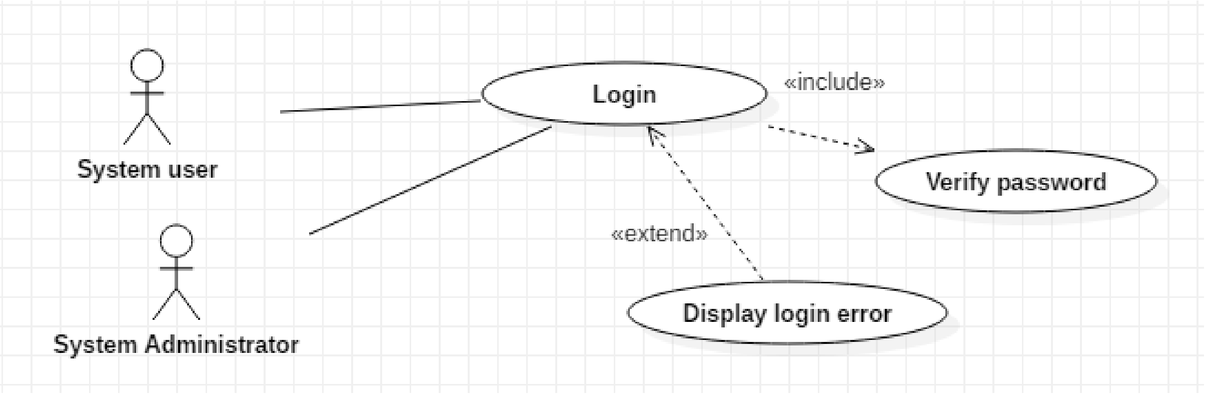


Figure 4: Login Use case

The following table describes the login use case

Use case section	**Description**
Use case name	Login
Participating actors	System administrator, System user
Brief description	The use case starts when either the user or administrator attempts to login to the system
Pre-condition	Both the administrator and user need to have accounts with the system
Flow of events	-Administrator or user clicks on the login button -Administrator or user enters the username and password -If success, a message of success is shown on the screen -If failure, a message of failure is shown on the screen

Table 1: Login use case table

Activity	Basic flow	Alternate flow
Scenario		
Admin or user tries to login	Basic flow of events take place	No alternate event
Admin or user tries to login with invalid password	Basic flow of events after alternate events occurs	Display error message

Table 2: Scenarios in the user and administrator data manipulation

Register user

-Similar to add user

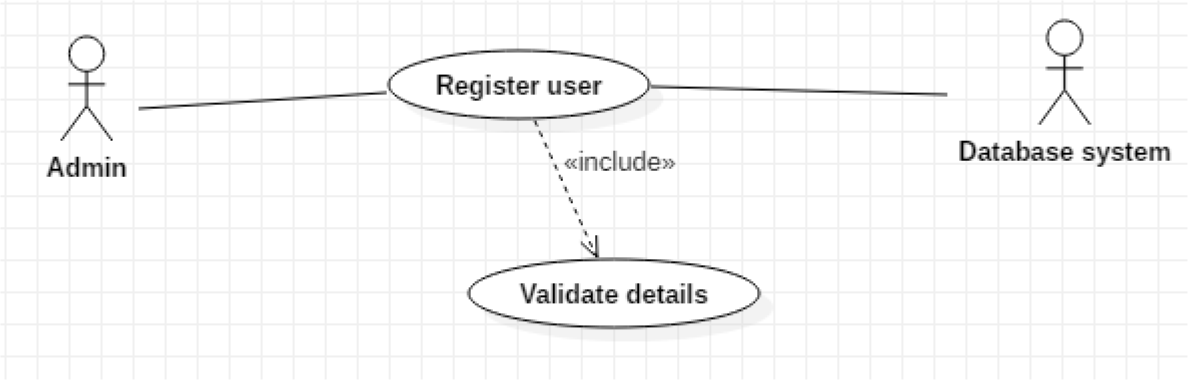


Figure 3: Register use case

The following table describes the register use case:

Use case section	Description
Use case name	Register user
Participating actors	Admin, Database system
Brief description	The use case starts when the admin clicks on the registration section of the window to create an account
Pre-condition	The admin should have his own existing account in the system in order to register a user. The admin account details will already be encoded in the system
Flow of events	-The admin clicks on the add user section of the system -He/She is presented with a registration form to fill -After filling the registration form, the admin is required to submit it -If successful, the details are stored in the user database system and user is already registered -The admin can verify the user details by clicking on the verify button -If failed, a failure message is displayed on the screen
Post-condition	The admin can remove a user upon his/her successful registration

Table 3: Description of the Register use case

Use case scenarios of register user:

Activity	Basic flow	Alternate flow
Scenario		
Admin enters all correct details	Basic flow of events take place	No alternate event
Admin fails to fill in a mandatory field	Basic flow of event after alternate event occurs	-Display error message -Next section becomes unavailable until mandatory field is supplied
Admin Does not enter the details as per the constraints	Basic flow of events after alternate event occurs	-Display error message -Next section becomes unavailable until mandatory field is supplied

Table 4: Scenarios in the register use case

c) Remove user

The following shows the use case of when the user is removed from the system

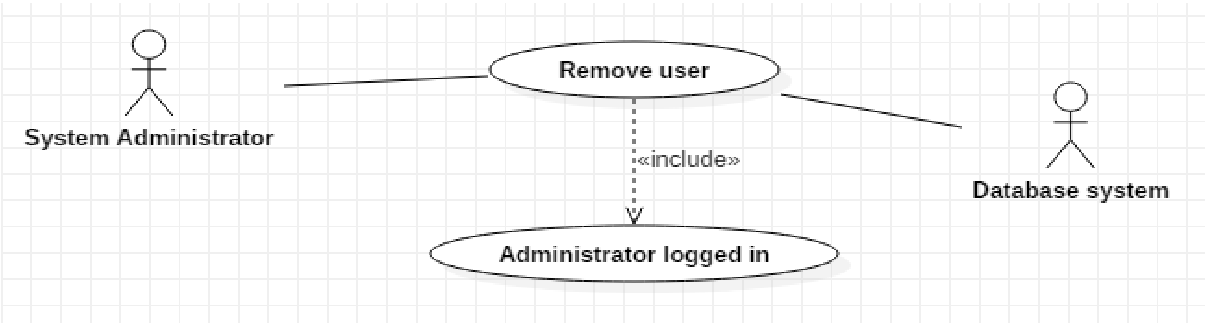


Figure 5: System user remove use case

Use case section	**Description**
Use case name	Remove user
Participating actors	System administrator, Database system
Brief description	The use case starts when the administrator logs in to the register user window
Pre-condition	There has to be an already registered user
Flow of events	-Administrator logs in -Upon successful login, he/she is presented with a window from which he/she can remove the user -If success, a message of success is shown on the screen -If failure, a message of failure is shown on the screen
Post-condition	-The old user needs to be registered again in order to access the data in the system

Table 5: Description of a system user removal

Activity	Basic flow	Alternate flow
Scenario		
Admin removes a user	Basic flow of events take place	No alternate event
Admin tries to remove a user without successful login	Basic flow of events after alternate events occurs	Display error message

Table 6: Scenarios in the user and administrator data manipulation

d) Data manipulation

The following shows a use case on how the administrator and user can manipulate data in the database system

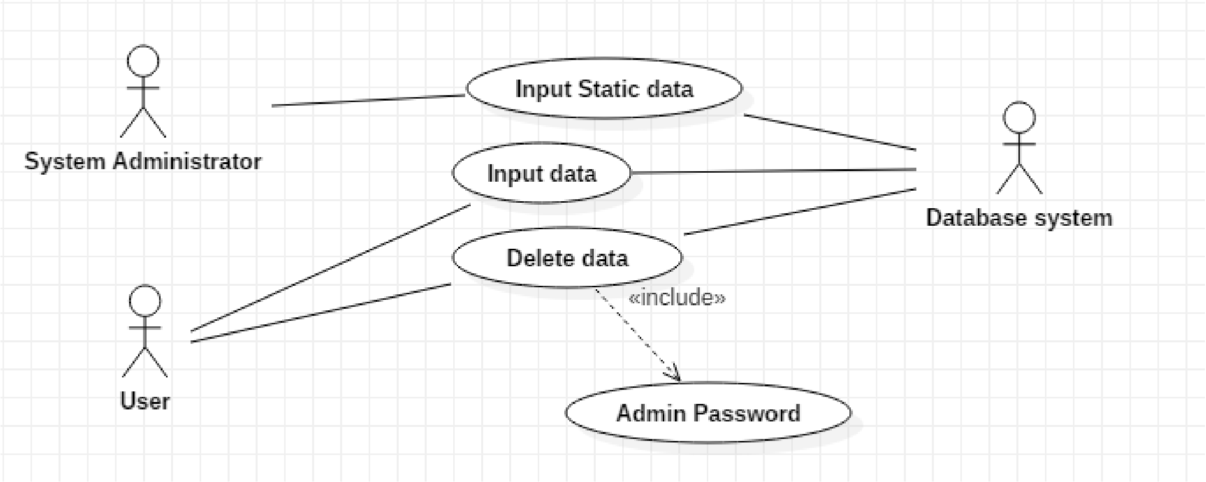


Figure 6: Administrator and user data manipulation use case

Use case section	**Description**
Use case name	Input static data, Input data, Delete data
Participating Actors	System administrator, user, database system
Brief Description	-Use case starts when either the user or the administrator clicks on either one of the three menus which are: Income statement, Balance sheet or cash flows -The administrator inputs static data, which means the data that must exist for the system to operate. The user on the other hands inputs the dynamic data which can vary, and in turn can delete data only with the administrator's password.
Pre-Condition	User or administrator must be logged in before they can perform any operation on data
Flow of events	-Either the customer or Admin clicks on the desired push button to work with the desired menu. -If either the user or admin are logged in, they have the liberty to add data -When the user or admin wish to delete a record, they are prompted to provide the administrator's password
Post conditions	-The user or Admin can view the data manipulated on a window by clicking on a button

Table 7: Description for the user and administrator data manipulation

Use case scenario for user and administrator data manipulation

Activity	Basic flow	Alternate flow
Scenario		
User or admin tries to add data while logged in	Basic flow of events take place	No alternate event
User or admin tries to delete data without administrator's password	Basic flow of events after alternate events occurs	Display error message

Table 8: Scenarios in the user and administrator data manipulation

e) Statistical data generation and display use case

The following shows a use case on system statistical data generation and display of the generated statistical data

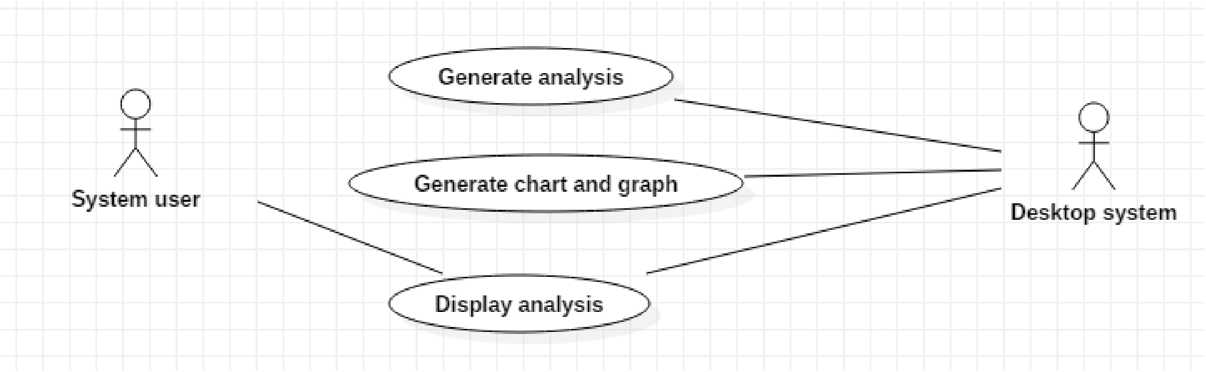


Figure 7: Statistical data generation and display use case

The following table describes the statistical data generation and display use case:

Use case section	Description
Use case name	Generate analysis, Generate chart and graph and display analysis
Participating actors	System user, Desktop system
Brief description	The use case starts after the user has input data into the system and prompted the system to compute graphical and chart data and display them
Pre-condition	The system user should have logged into the system
Flow of events	-The user logs into the system and selects an option of accounts to work with -He/She is presented with a workspace to work with -After filling the data, the user submits the data -If successful, the details are stored in the database system and the user is shown a success message -The admin can verify the user details by clicking on the verify button -If failed, a failure message is displayed on the screen -Upon success of data entry, the system automatically generated the statistical data upon prompt by the user -The user can view this message if he/she wishes to as long as the use is prompted

Table 9: Description of the statistical data generation and display use case

Activity Scenario	Basic flow	Alternate flow
User enters data	Basic flow of events take place	No alternate event
User attempts to prompt the system to display statistical data without data entry	Basic flow of events after alternate events occurs	Display error message

Table 10: Scenarios in the statistical data generation and display use case

CHAPTER 5: DETAILED/ ARCHITECTURAL DESIGN

The detailed design involves the detailed description of the compositions and functionalities of the system. Each component is widely and sufficiently described to allow for its construction.

SmartAccountant design has been handled using two design approaches:

a) Structured Design. b) Object-Oriented Design.

Structured Design- This refers to the design of how data flows in the system. Here, data flow diagrams have been created and refined enough until data flow is visible enough.

Context Diagram

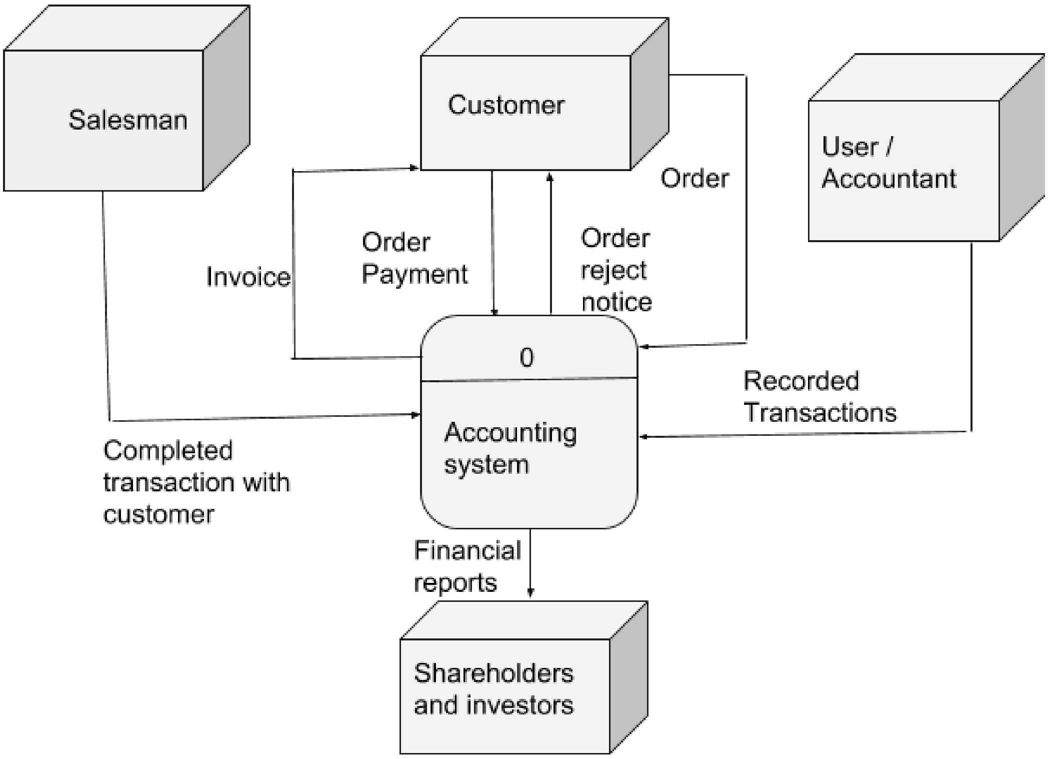


Figure 8: Context diagram for the secured computerised accounting system

Object oriented design-Here, the identification of various components in terms of objects and group them to view how data flows between them will be shown.

Class diagram

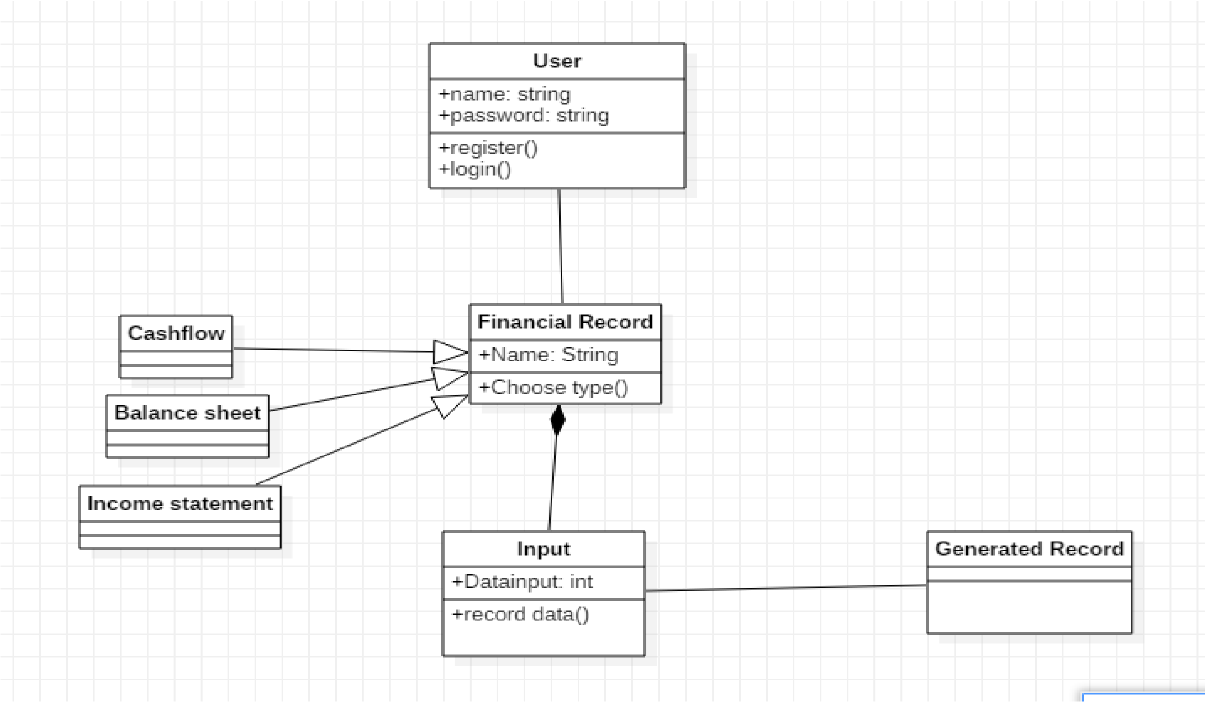


Figure 9: Class diagram

Collaboration/ Communication diagram

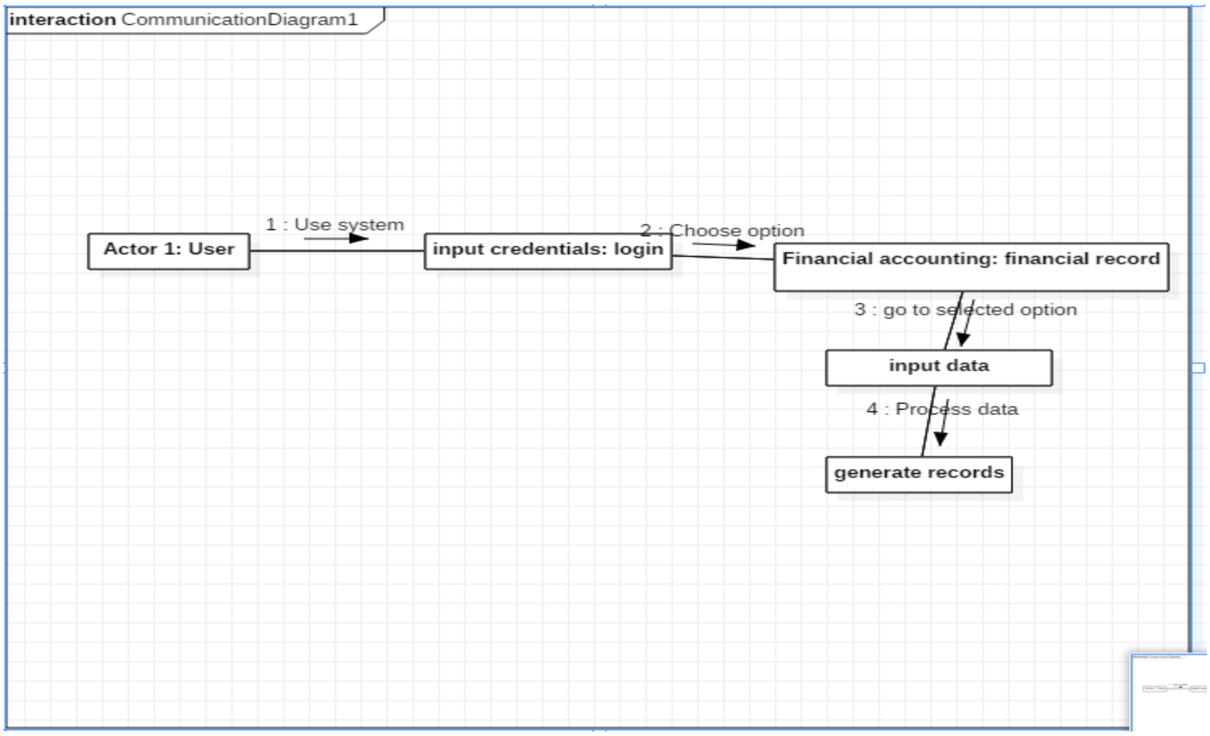


Figure 10: Collaboration diagram

Sequence diagram

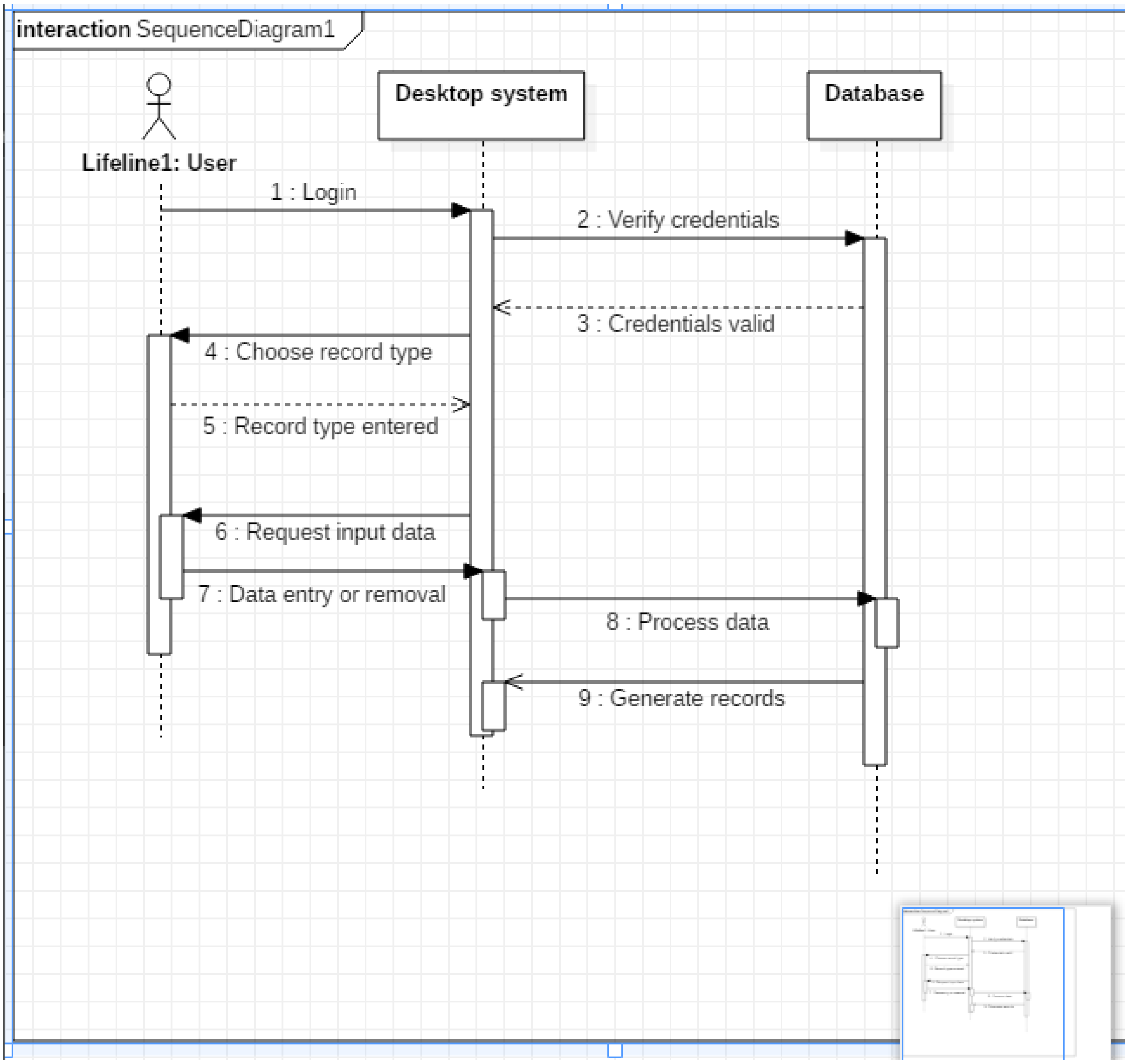


Figure 11: Sequence diagram

General system flowchart:

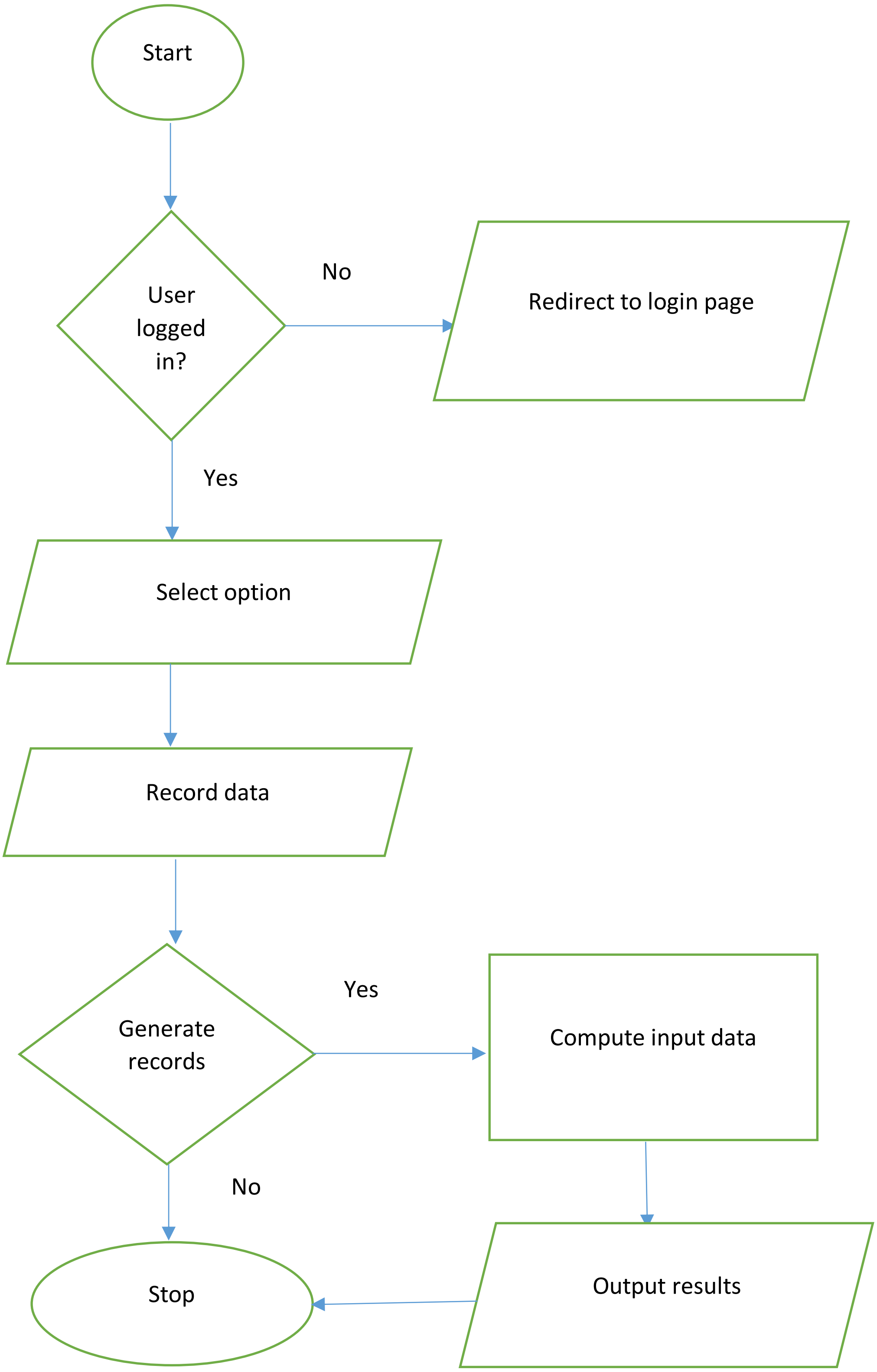


Figure 12: System flowchart

User Interface design

Here, the making of the interfaces that the user will interact to access the system will be discussed.

The following are simple UI designs proposed for the SmartAccountant system:

Login page:

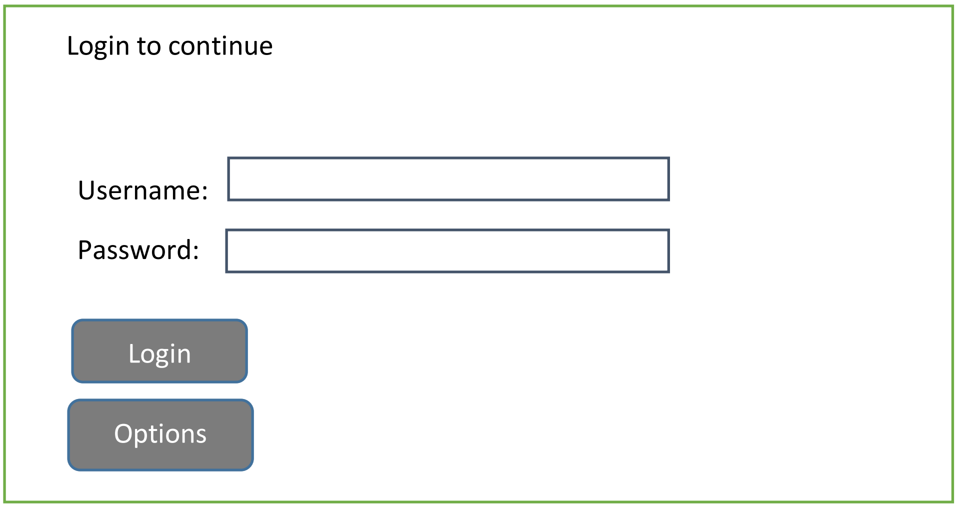


Figure 13: Login page user interface design

Registration page:

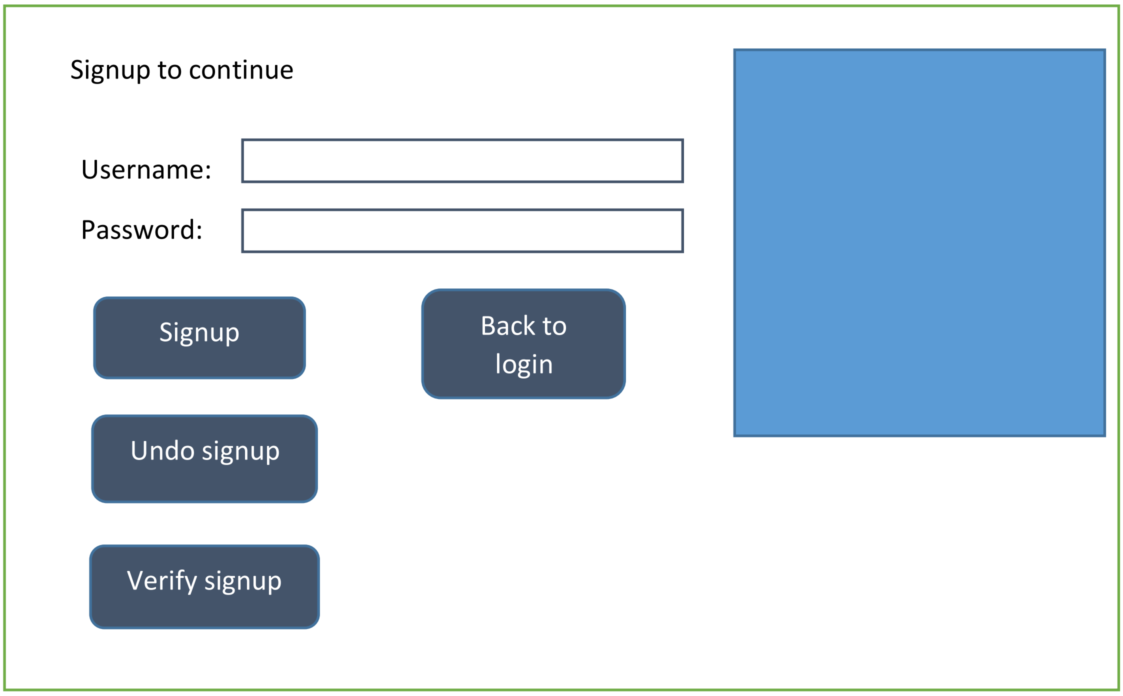
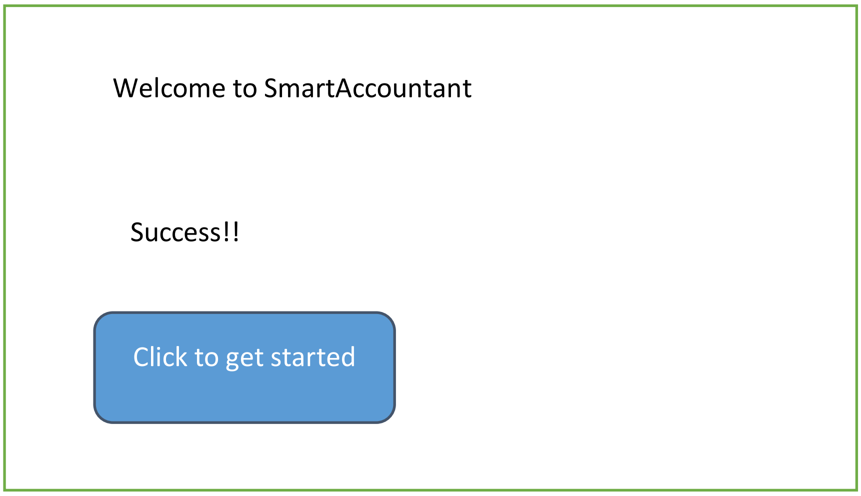


Figure 14: Registration page user interface design

Homepage

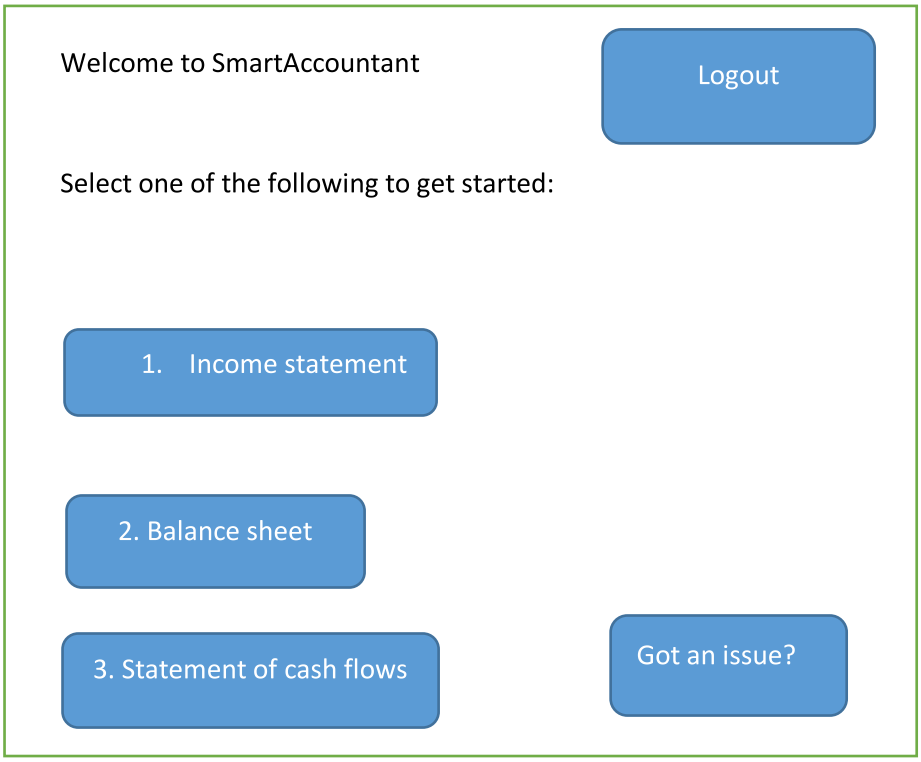


Welcome to SmartAccountant

Success!!

Figure 15: Homepage user interface design

Main menu:



Welcome to SmartAccountant

Select one of the following to get started:

Figure 16: Main menu user interface design

CHAPTER 6: SYSTEM IMPLEMENTATION AND TESTING

4.1 SYSTEM CODING

Here, the system designs were converted into to actual code. This involved the use of a programming language while following its standards in terms of handling of functions, variables, comments, file organization and general code layout.

For SmartAccountant, the tools for implementation were:

1) Hardware Resources

A computer with the following specifications:

a) 250GB free disk space.

b) 4GB RAM.

c) 2GHz processor.

2) Software Resources

The following software resources were used:

a) Windows 10x64 Operating System.

b) QT GUI platform

c) GCC compiler for c++

d) SQLite database management software.

e) Q Custom plot

Programming Languages

SQL

Standard Query Language (SQL) is the language used for creation and querying of databases. The queries are executed by the database management system engine (MySQL in this case)

HTML and CSS

Hypertext Markup Language is the language used to create the actual web pages and Cascading

Style Sheets (CSS) is used to style the content of these web pages.

C++

C++ is a general-purpose object-oriented programming (OOP) language, which is an extension of the C language. It is therefore possible to code C++ in a "C style" or "object-oriented style." In certain scenarios, it can be coded in either way and is thus an effective example of a hybrid language.

C++ is considered to be an intermediate-level language, as it encapsulates both high- and low-level language features. Initially, the language was called "C with classes" as it had all the properties of the C language with an additional concept of "classes."

SYSTEM TESTING

Here, the system's functionality was checked using test values to discover and eradicate errors. The SmartAccountant system underwent the following testing:

UNIT TESTING

Each function and module was tested individually to see if they were working as expected. Drivers were used to test for modules that depended on other modules for data. The functions were made to be highly cohesive, i.e. have only one function and loosely coupled. An example is shown in the table below:

Test case 1	Test case 1	Type: Module testing Module: NumPressed()	Type: Module testing Module: NumPressed()	Type: Module testing Module: NumPressed()
Test No	Field Name	Data	Expected outcome	Actual outcome
1	Key space	0-9,-/+,*,/,=	The function enables any button, from the range of 0-9 and any sign buttons to be displayed upon click operation. In addition, the signs can only be written in the form (-/+)x(*//)(+/-)y, where x and y are variables	A string of desired length, from 0 to 9 was generated. In addition, the signs could only be written in a prefix format to a variable integer, hence determining the sign of the variable and the relationship between that variable and other variables

Table 11: Demonstration for unit testing

INTEGRATION TESTING

The functions were all integrated together to form grouped modules which were then tested to determine whether they function as expected when linked together. All occurring errors were corrected.

SYSTEM TESTING

All other components of the system were tested e.g. user interface and the storage system. Inputs and matching outputs were tested.

BLACKBOX TESTING

This refers to the testing of the systems in terms of using actual values as test inputs and checking if they produce expected outputs.

The following test cases were generated for the system:

Test case for submitting data on the balance sheet menu

Scenario	Particulars	Balance c/f	Balance b/f	Time	Expected results
User enters all details correctly	Receivables	12000	15000	15:00	Display success message
User fails to enter a mandatory detail	N/A	12000	15000	15:00	Displays error message, followed by possible cause of error.
User fails to enter data as per the field constraints	15000	Receivables	15:00	12000	Displays error message, followed by possible cause of error.
User fails to input all fields	N/A	N/A	N/A	N/A	Displays error message, followed by possible cause of error.

Table 12: **Test case for submitting data on the balance sheet menu**

Chapter 7: CONCLUSION AND RECOMMENDATIONS

ACHIEVEMENTS

The project was completed successfully after its implementation according to the models and designs outlined. This enabled it to achieve its objectives and requirements in terms of:

a) System users can register an account with the system through the admin.

b) System users can view and choose from among several options of accounts to work with.

c) System users can manipulate data in the system.

d) System users can prompt the system to produce statistical data and analysis

e) System users can extract the data in the system at will

f) System administrators can add new system users.

g) There are some actions which system users need system administrators in executing, such as data deletion, where a user will require the admin's password in order to delete data.

CONSTRAINTS

Despite all the achievements, a few challenges were encountered during system development:

- Direct generation of reports was extremely difficult. Consequently, the user, through the administrator, will have to manually export data from the database.

- Automatic backup of data was difficult to accomplish. Consequently, the user will need third party applications such as window's one drive or google drive backup services.

CONCLUSION

The system performs its intended work as stated in the requirements and in addition, it offers additional features such as extracting files to a document of any type. With the features the system has, it is expected to fully meet its expectations which are data entry, manipulation, extraction and security of the financial records.

RECOMMENDATIONS

The system will only meet its goals if ethical system administrators are entrusted to run the system. It is through this that ethical standards in accounting will be met, through the system's user input data. In addition, the system will need system users who have some knowledge in accounting and computing to fully guarantee the optimal running of the system in producing accurate and precise results.

APPENDIX A: REFERENCES AND BIBLIOGRAPHY

1. Accounting Coach (2019). Financial Accounting. URL: https://www.accountingcoach.com/financial-accounting/explanation last accessed February 2019
2. Panvar Kumar (2019). Role of computers in Accounting. URL: http://www.yourarticlelibrary.com/accounting/computerized-accounting/role-of-computers-in-accounting/62752 last accessed February 2019
3. Carmelo Romano (2013). Advantages of Computerized Accounting. URL:http://www.cleveraccounting.com/9-advantages-computerized-accounting/ last accessed February 2019
4. Otieno Odhiambo (2017). Why Financial Reporting is important for investors. URL:https://www.standardmedia.co.ke/article/2001262229/why-financial-reporting-is-important-for-investors last accessed February 2019
5. Wikipedia (2019). Financial Accounting. URL:https://en.wikipedia.org/wiki/Financial_accounting last accessed February 2019
6. Visual Paradigm (2019). Requirement Analysis Techniques. URL: https://www.visual-paradigm.com/guide/requirements-gathering/requirement-analysis-techniques/ last accessed February 2019
7. Trends in I.S: Executive Support Systems (2019). Waterfall model, Implementation. URL: https://eternalsunshineoftheismind.wordpress.com/2013/03/10/sdlc-phase-5-maintenance/ last accessed February 2019.
8. Cleverism (2019). How to conduct a feasibility study the right way. URL: https://www.cleverism.com/conduct-feasibility-study-right-way/ last accessed February 2019.

9. Bizfluent (2017). Disadvantages of accounting software. URL: https://bizfluent.com/about-5106557-disadvantages-accounting-software.html Last accessed February 2019
10. Chron (2019). Disadvantages of using accounting software. URL: https://smallbusiness.chron.com/disadvantages-using-accounting-software-3823.html Last accessed February 2019
11. Solution matrix, business encyclopaedia (2019). Retrieved from: https://www.business-case-analysis.com/matching-concept.html last accesses February 2019

APPENDIX B: GNATT CHART AND RESOURCES

Gantt chart

	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12
Requirements analysis	■						
Design		■					
Implementation			■	■	■		
Testing				■	■	■	
Maintenance							■

Table 13: Gantt chart

List of resources required

1. Text materials: Frank Wood's Business Accounting 1
2. Internet resources
3. Computer
4. I.D.E for C++, such as QT
5. G.U.I for C++, such as QT
6. K.A.S.N.E.B CPA section 1 part 1 notes for Financial Accounting
7. Library resources

Budget

Resource	**Cost(Ksh)**
Books	6,500
Internet	500
Information gathering e.g: Interviews	4,000
Computer software	450
Additional notes	200
Total	**11,650**

Table 14: Budget

APPENDIX C: USER MANUAL

The following is the user manual for SmartAccounatant system:

Account creation:

These are the steps to be followed during account creation:

a) On the login menu, click on the Options button

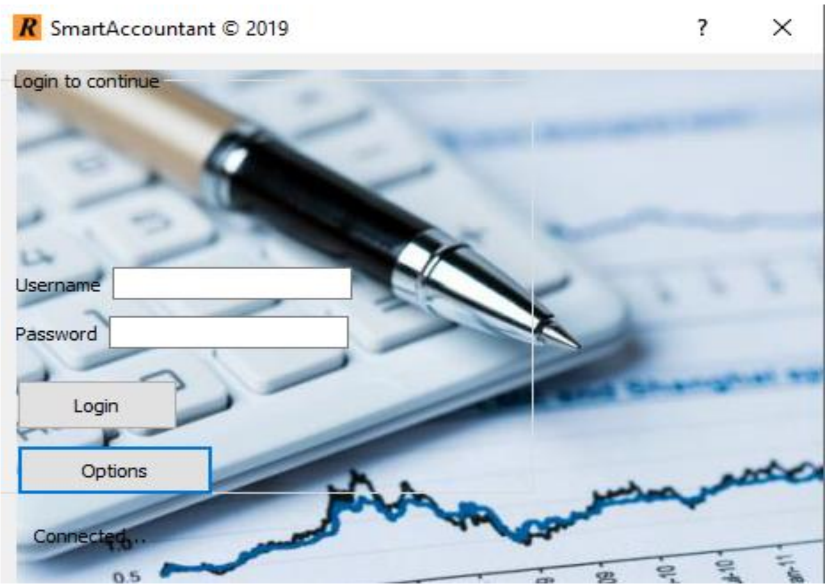


Figure 17: User login window

b) An administration login is required. The admin has to enter the administrator Username and password

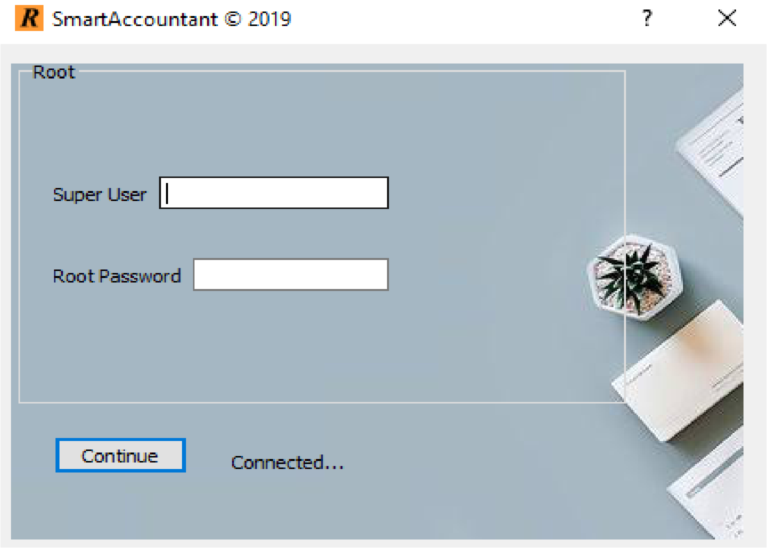


Figure 18: Admin login window

c) Enter details on the username and password line edits then click on the signup pushbutton. The Admin can undo signup by entering data on the username field only, and he/she can verify signup by clicking on the 'verify signup pushbutton' to verify the user credentials. Upon successful user registration, click on the back to login pushbutton

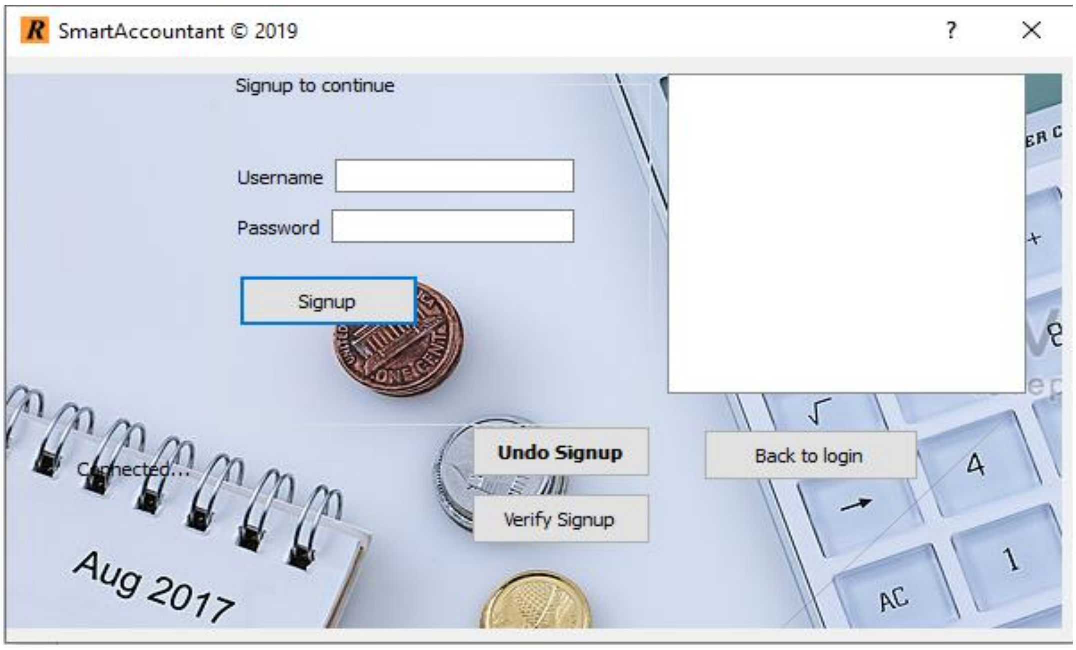


Figure 19: User signup window

Choosing type of accounts to work with

Please note: Only users that are logged in can view this window.

The following are the steps to be followed when choosing an account type

a) After successful login, the following window will pop up. Click on the get started pushbutton to continue

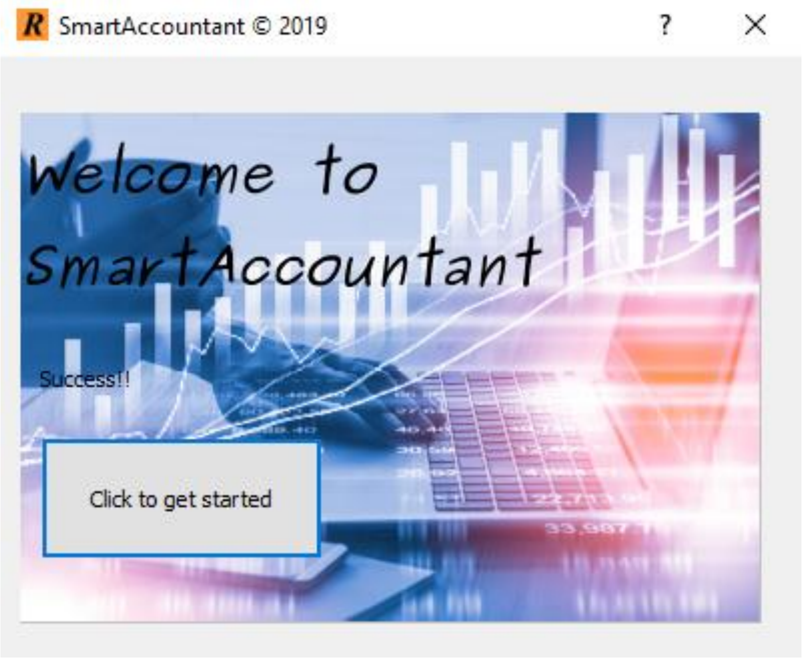


Figure 20: Welcome window

b) The following window pop up, from which the user can choose an account type to deal with

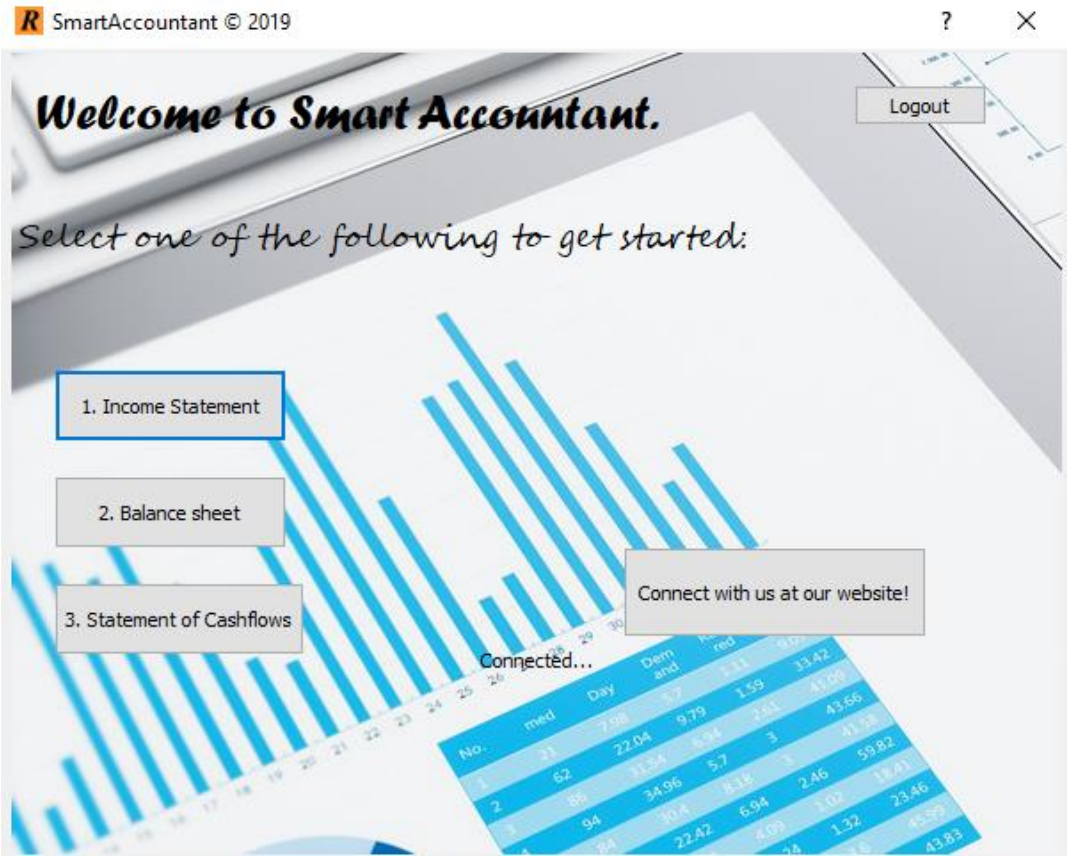

SmartAccountant © 2019
?
Welcome to Smart Accountant.
Logout
Select one of the following to get started:
1. Income Statement
2. Balance sheet
3. Statement of Cashflows
Connect with us at our website!
Connected...

Figure 21: Main menu window

c) Data entry

-For instance, upon clicking on the Income statement pushbutton, the user is presented with a window as shown below:

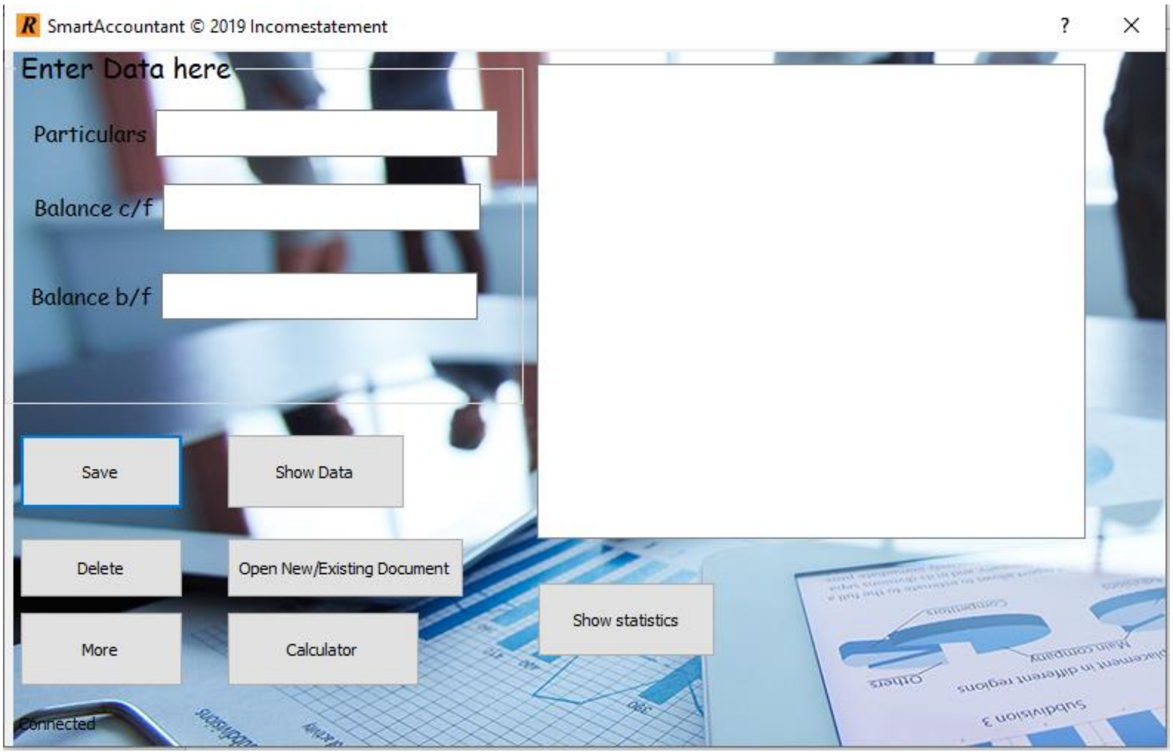


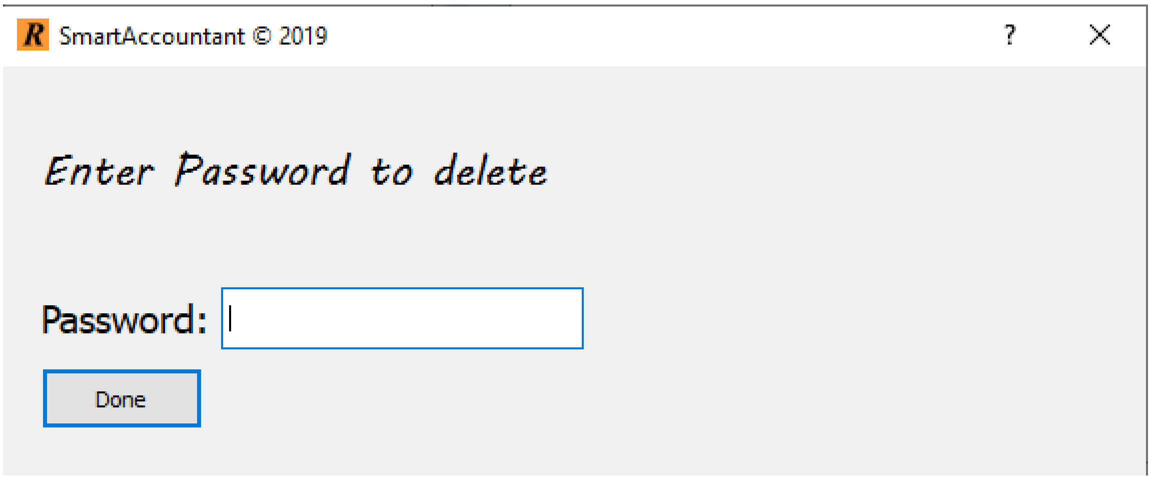
Figure 22: Income statement window

-In order to view the already existing data in the system, click on the 'show data' pushbutton

-In order to save data into the system, insert data into the 'Particulars', 'Balance b/f' and 'Balance c/f' line edits and click on save.

-In order to delete data, follow the following steps:

> -> Insert the type of particulars on the particulars line edit and click on the delete push button. If the user is unsure of the type of particular to insert, click on the show data pushbutton. The following window will pop up upon clicking the delete pushbutton:



SmartAccountant © 2019

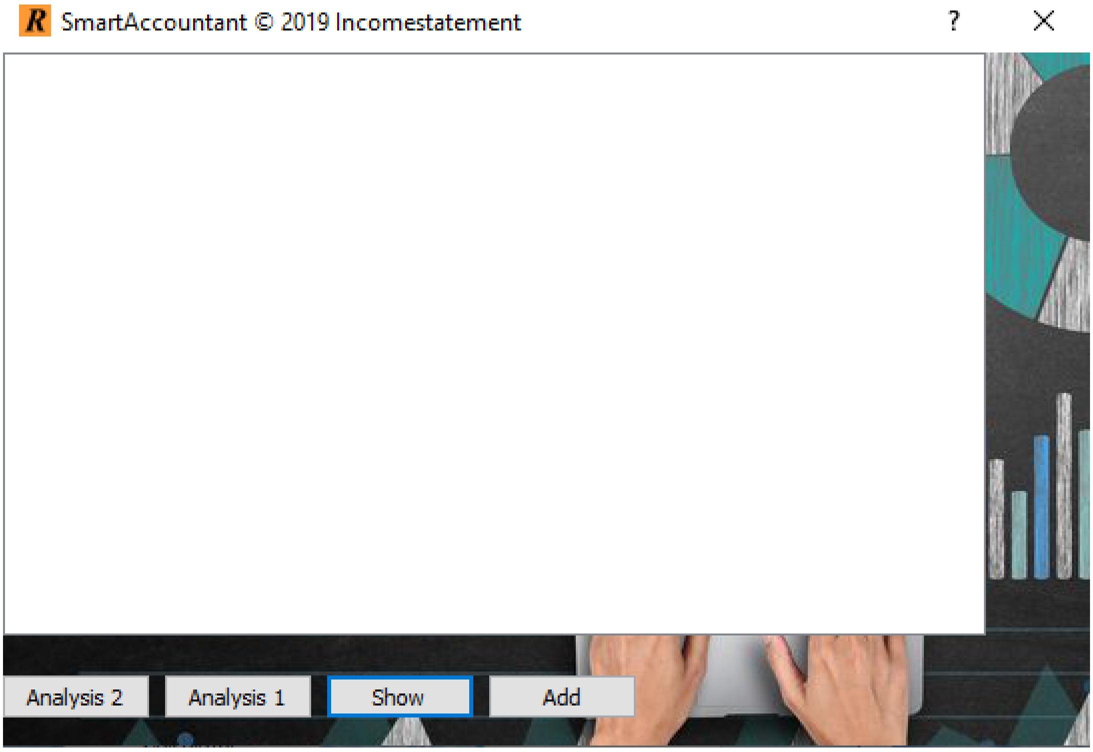
Enter Password to delete

Password:

Figure 23: Enter password window

- ➔ Enter the administrative password and click done
- ➔ A popup window showing success will pop up on the screen else a failure popup will show on the screen. In case of a pop up failure, make sure that only the particulars line edit is filled with data.

- If in case the user wishes to view a more detailed view of the data, click on the more push button and the following window will be displayed:



SmartAccountant © 2019 Incomestatement

Figure 24: Income statement window

-Click on the show pushbutton to view data in a more detailed view, and click on add to perform an add operation on all the input data hence click on the show pushbutton to see the results of the computation.

-In case the user wants an analysis based on the input data and computation, click on the analysis 1 and analysis 2 pushbuttons and click on the show pushbutton to see the results.

-In addition, if the user wishes to view a statistical approach of the data, he/she should click on the 'show statistics' pushbutton. The following shows a sample statistical pie chart:

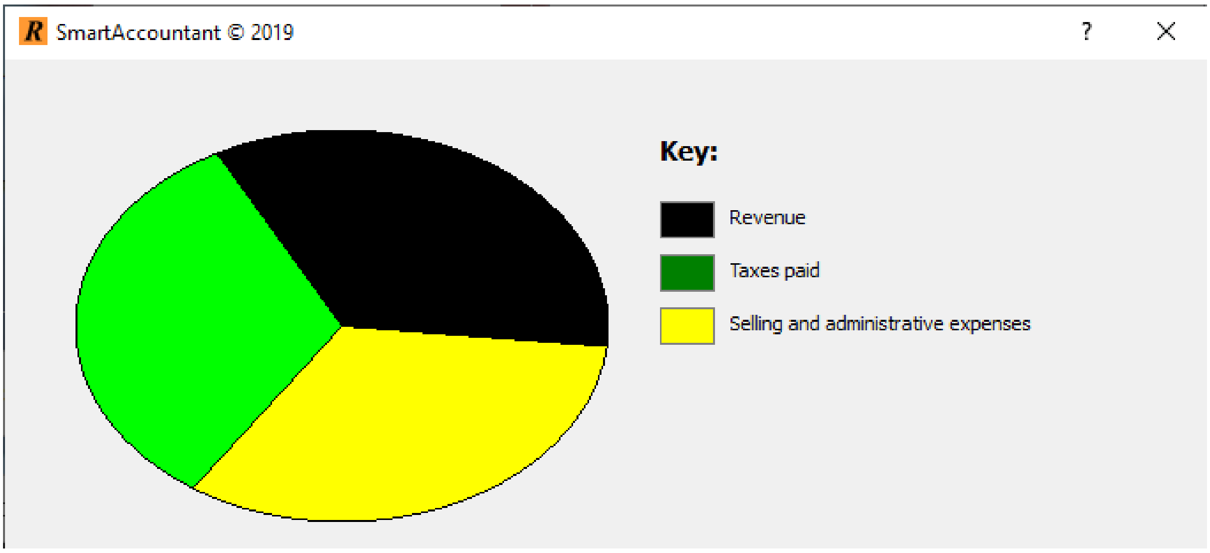
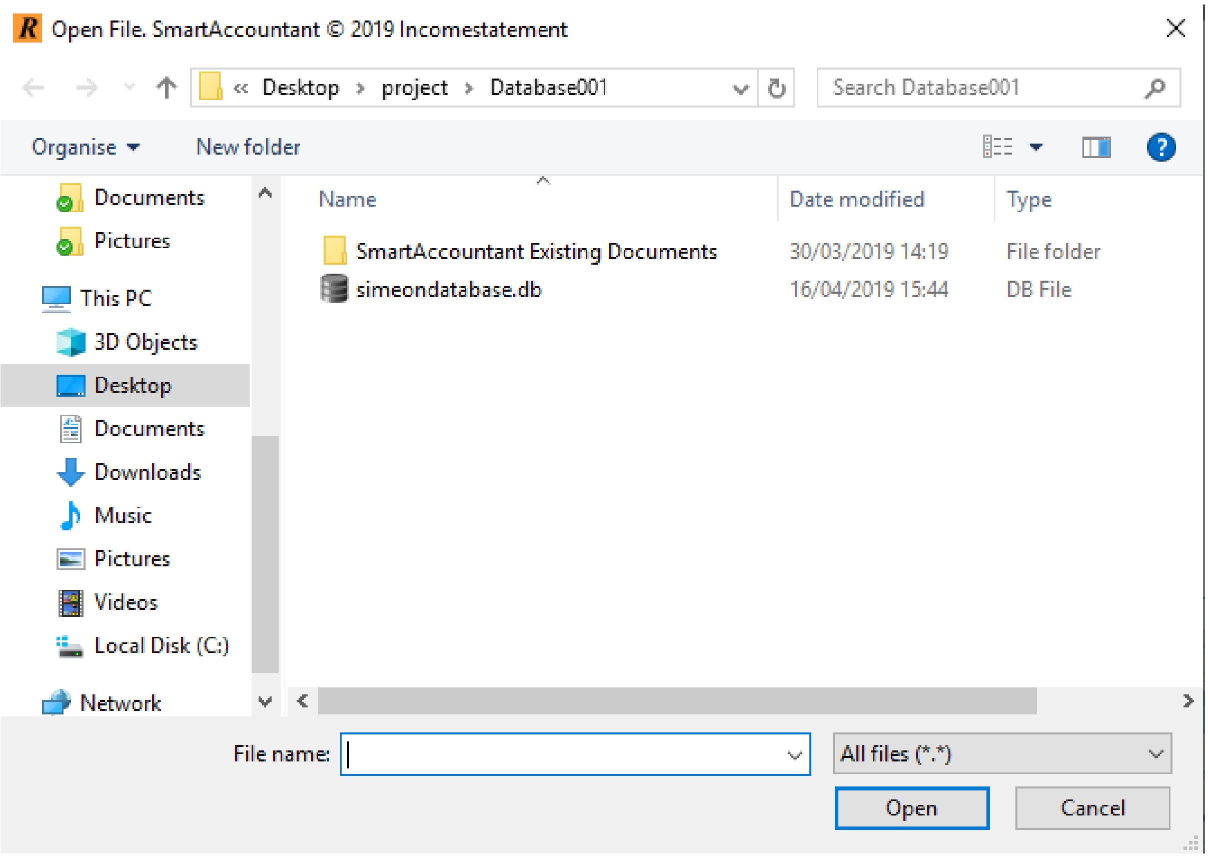


Figure 25: Sample pie chart

-In addition, if the user wishes to export data in the system to another form, say in excel form, follow the following steps:

-> Click on the 'Open new/existing document' pushbutton. The following window will be displayed:



Open File. SmartAccountant © 2019 Incomestatement

« Desktop › project › Database001

Search Database001

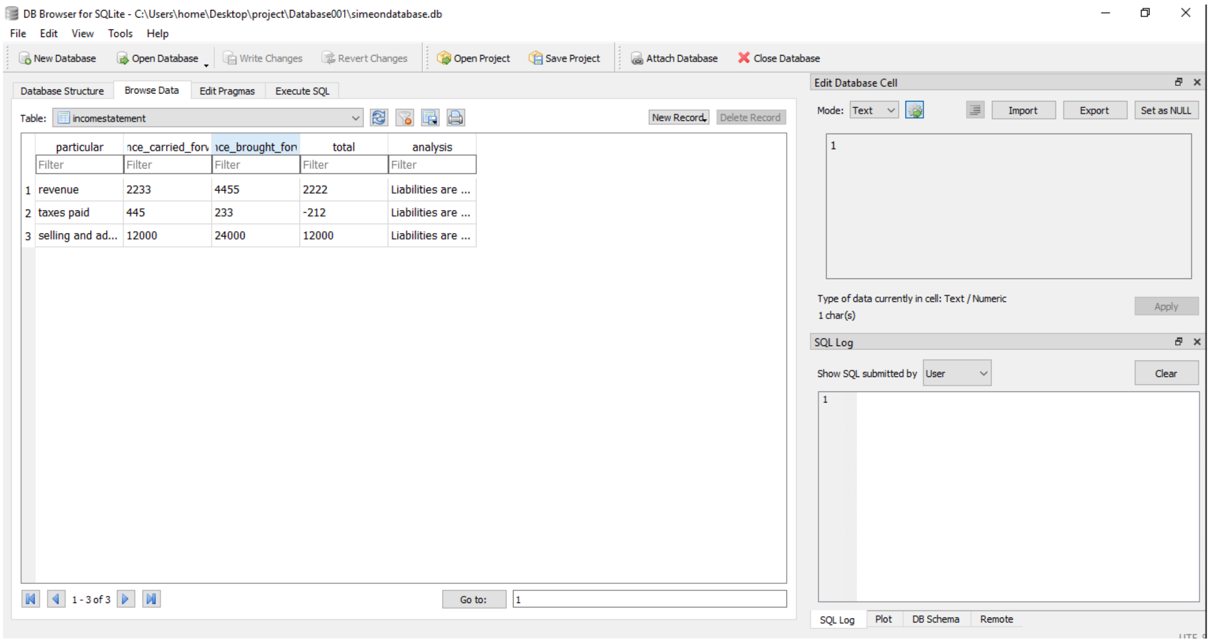
Name	Date modified	Type
SmartAccountant Existing Documents	30/03/2019 14:19	File folder
simeondatabase.db	16/04/2019 15:44	DB File

File name:

All files (*.*)

Figure 26: Directory window

➔ Click on the simeondatabase.db file and open it.



DB Browser for SQLite - C:\Users\home\Desktop\project\Database001\simeondatabase.db

Table: incomestatement

	particular	ıce_carried_forv	ıce_brought_forı	total	analysis
1	revenue	2233	4455	2222	Liabilities are ...
2	taxes paid	445	233	-212	Liabilities are ...
3	selling and ad...	12000	24000	12000	Liabilities are ...

1 - 3 of 3

Mode: Text

Type of data currently in cell: Text / Numeric
1 char(s)

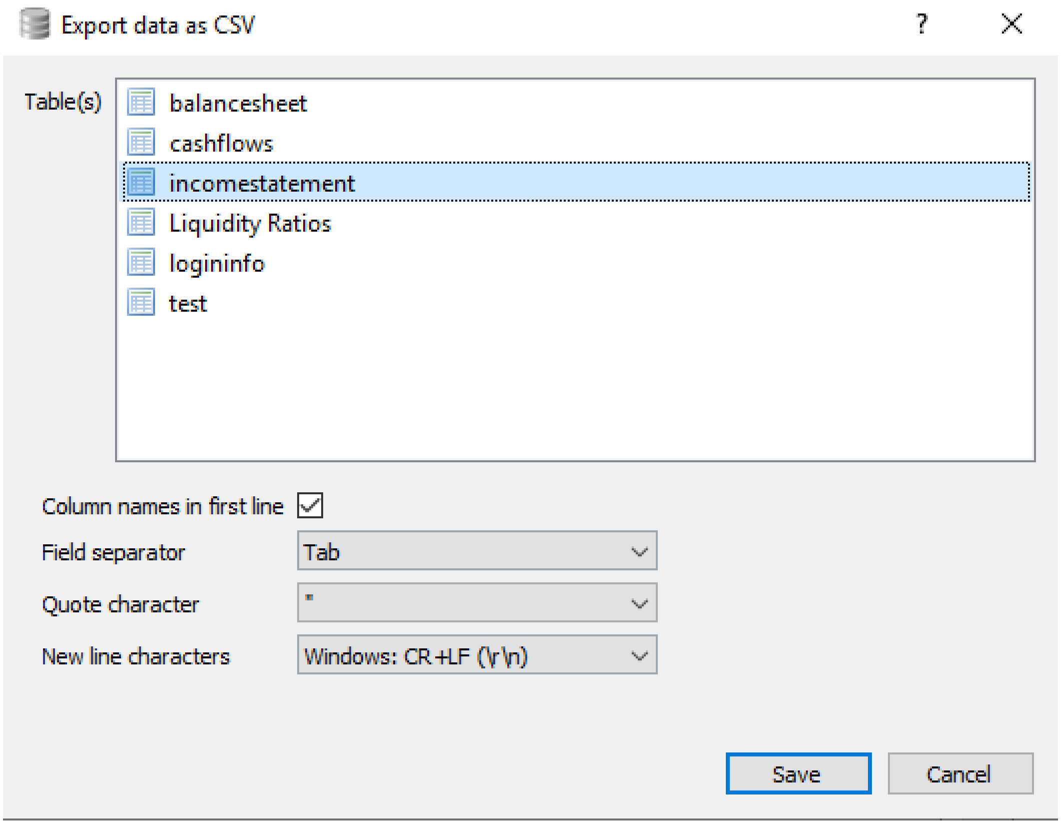
Show SQL submitted by User

Figure 27: Database table

➔ Click on the browse data bar and on the table pushdown table, select the type of table you wish to export. **NOTE: Only the**

administrator can perform this task, to prevent possibility of data manipulation

- ➔ Click on the file menu on the top left of the window. Go to the export side and select 'table(s) as csv file'.
- ➔ The following table will pop up. Click on save with the credentials as shown on the diagram below:



Export data as CSV

Table(s)
- balancesheet
- cashflows
- incomestatement
- Liquidity Ratios
- logininfo
- test

Column names in first line ☑

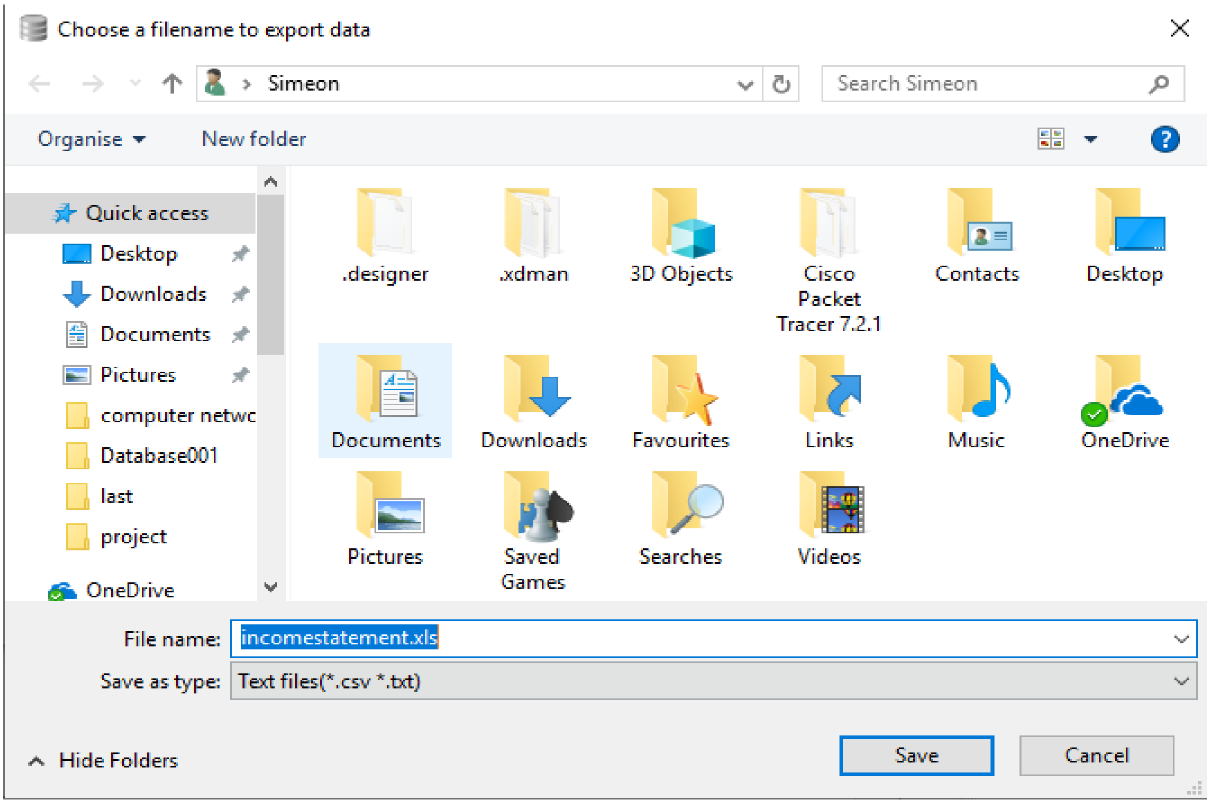
Field separator: Tab

Quote character: "

New line characters: Windows: CR+LF (\r\n)

Figure 28: Save-option window

- ➔ The following window will pop up:

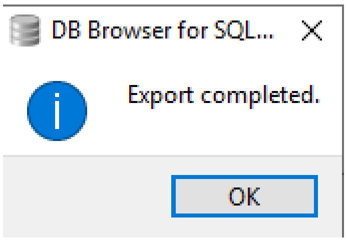

Choose a filename to export data

File name: incomestatement.xls

Save as type: Text files(*.csv *.txt)

Figure 29: Directory window

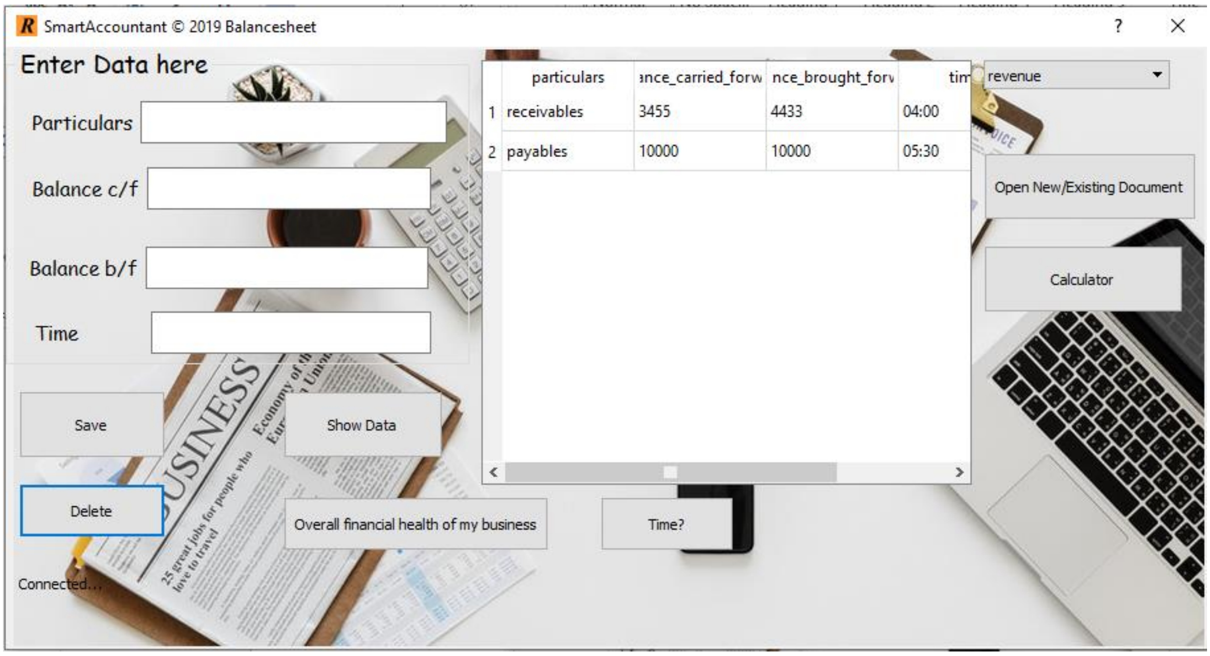
- ➔ Save the file as the administrator wishes to save it, with the extension '.xls', for instance if an excel format is desired, or '.docx', if a word format is needed.
- ➔ Click on save and a popup window showing successful operation of the task has been completed.


DB Browser for SQL...

Export completed.

Figure 30: Success message

- On the other hand, on the balance sheet, on clicking pushbutton showdata, the following slide down window will show up on the top left:



SmartAccountant © 2019 Balancesheet

Enter Data here

Particulars

Balance c/f

Balance b/f

Time

	particulars	ance_carried_forw	nce_brought_forv	tim
1	receivables	3455	4433	04:00
2	payables	10000	10000	05:30

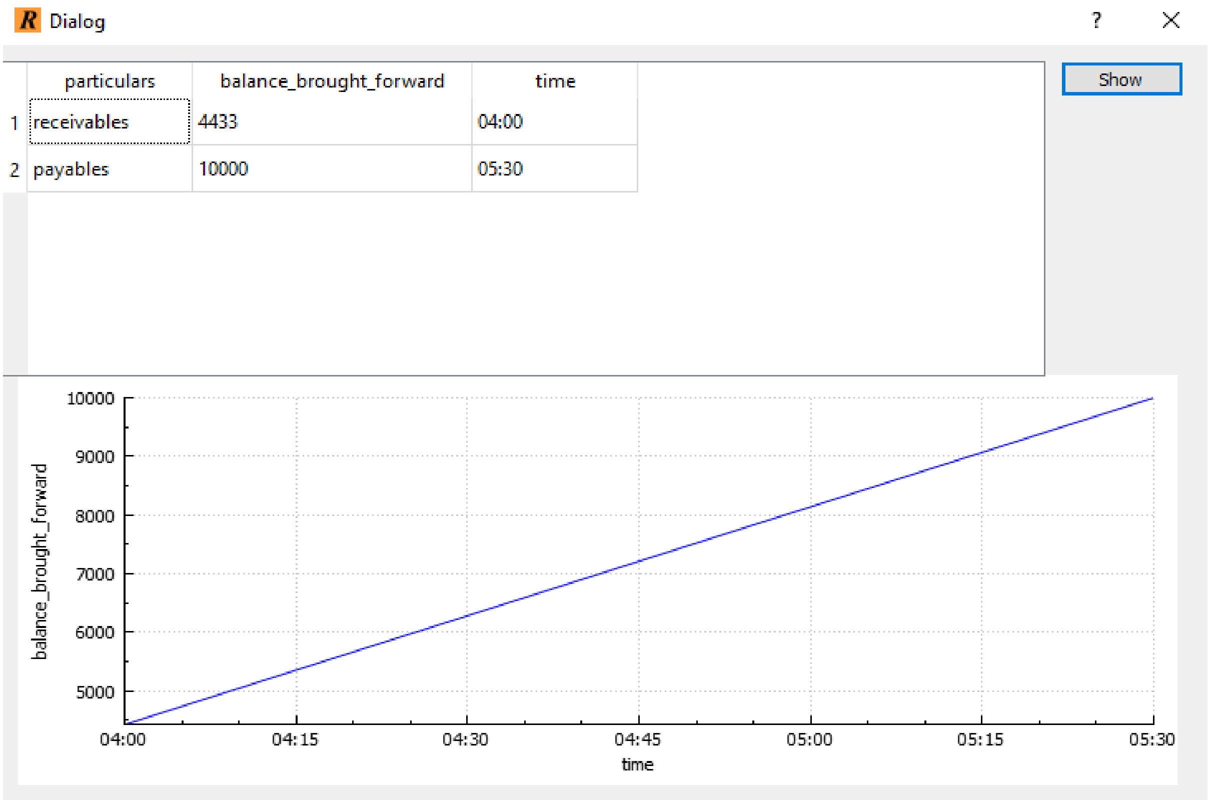
Connected...

?
revenue
Open New/Existing Document
Calculator
Save
Show Data
Delete
Overall financial health of my business
Time?

Figure 31: Balance sheet window

-The purpose of this is to show data input in the previous account, i.e. the income statement account. This is because the data from the income statement and balance sheet are is related.

-Upon clicking the 'overall financial health of my company' pushbutton, a graphical representation of the data is displayed, as shown below for example:



Dialog

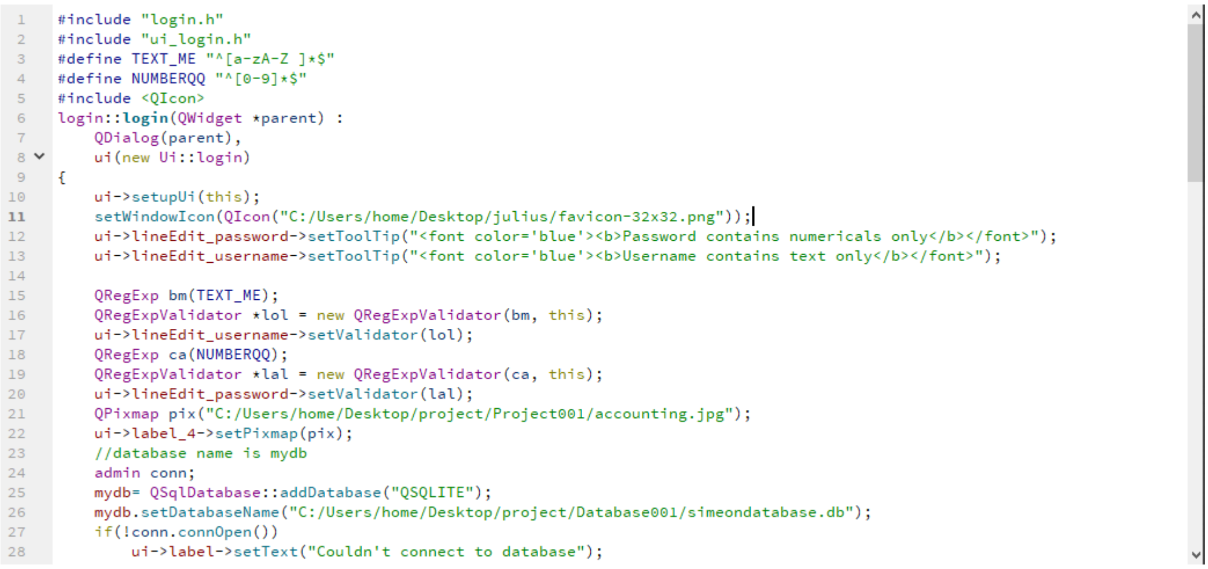
	particulars	balance_brought_forward	time
1	receivables	4433	04:00
2	payables	10000	05:30

Figure 32: Graph Diagram based on user input

-Click on the show pushbutton to see the type of data that the user is possibly dealing with.

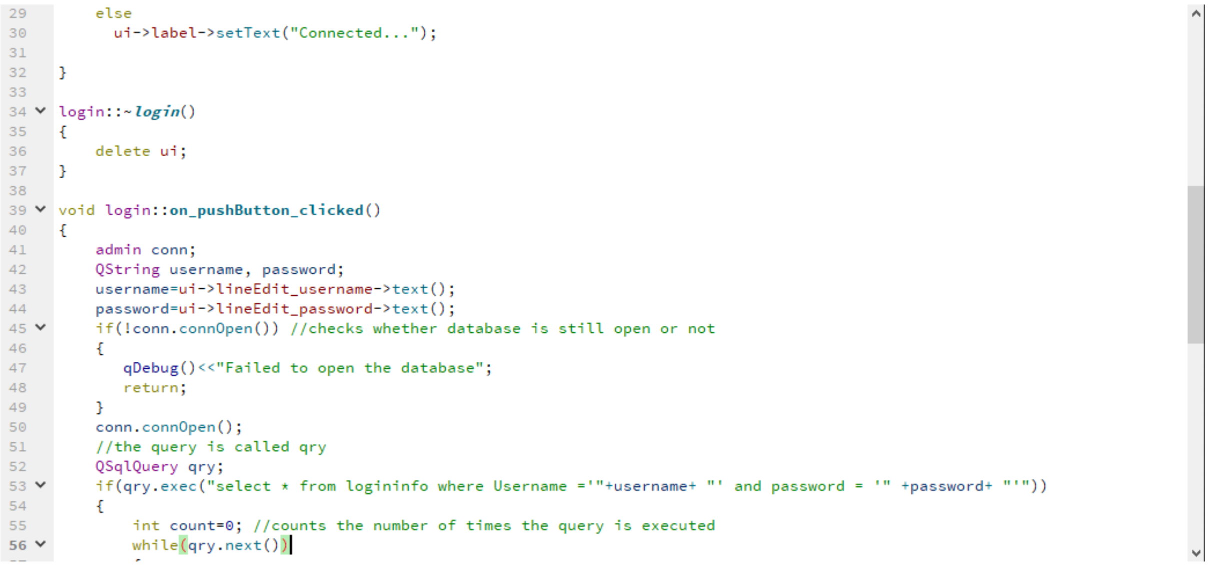
-On the cash flows window, the same operations as stated above are experienced.

APPENDIX C: SAMPLE PROGRAMS



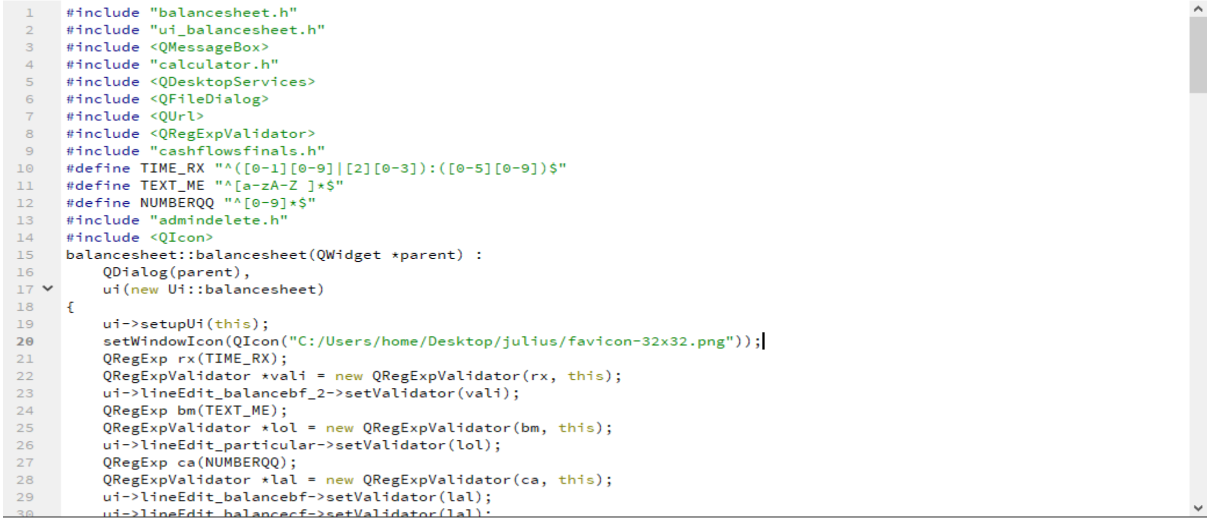
```cpp
#include "login.h"
#include "ui_login.h"
#define TEXT_ME "^[a-zA-Z ]*$"
#define NUMBERQQ "^[0-9]*$"
#include <QIcon>
login::login(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::login)
{
    ui->setupUi(this);
    setWindowIcon(QIcon("C:/Users/home/Desktop/julius/favicon-32x32.png"));
    ui->lineEdit_password->setToolTip("<font color='blue'><b>Password contains numericals only</b></font>");
    ui->lineEdit_username->setToolTip("<font color='blue'><b>Username contains text only</b></font>");

    QRegExp bm(TEXT_ME);
    QRegExpValidator *lol = new QRegExpValidator(bm, this);
    ui->lineEdit_username->setValidator(lol);
    QRegExp ca(NUMBERQQ);
    QRegExpValidator *lal = new QRegExpValidator(ca, this);
    ui->lineEdit_password->setValidator(lal);
    QPixmap pix("C:/Users/home/Desktop/project/Project001/accounting.jpg");
    ui->label_4->setPixmap(pix);
    //database name is mydb
    admin conn;
    mydb= QSqlDatabase::addDatabase("QSQLITE");
    mydb.setDatabaseName("C:/Users/home/Desktop/project/Database001/simeondatabase.db");
    if(!conn.connOpen())
        ui->label->setText("Couldn't connect to database");
```

Figure 33: Sample c++ code



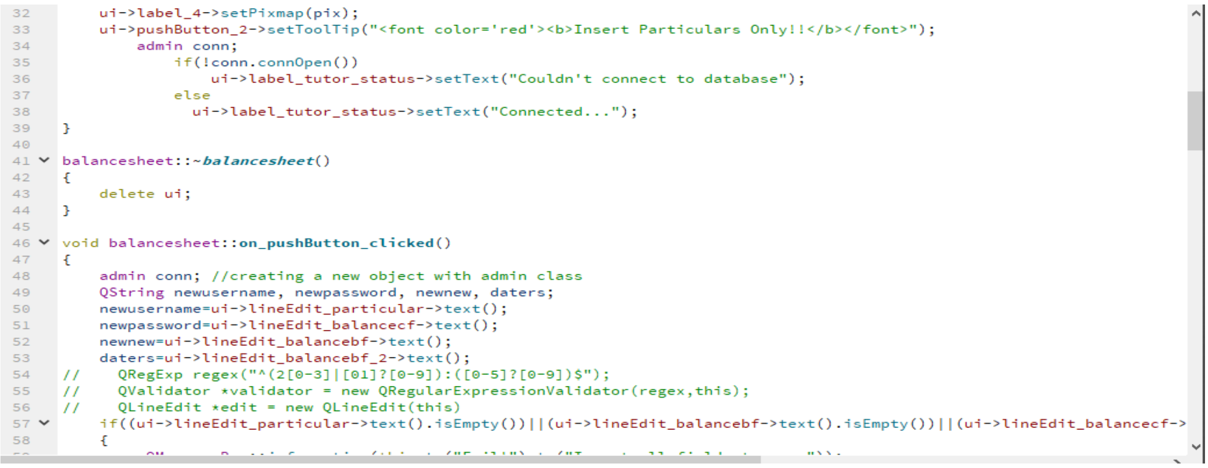
```cpp
    else
      ui->label->setText("Connected...");

}

login::~login()
{
    delete ui;
}

void login::on_pushButton_clicked()
{
    admin conn;
    QString username, password;
    username=ui->lineEdit_username->text();
    password=ui->lineEdit_password->text();
    if(!conn.connOpen()) //checks whether database is still open or not
    {
       qDebug()<<"Failed to open the database";
       return;
    }
    conn.connOpen();
    //the query is called qry
    QSqlQuery qry;
    if(qry.exec("select * from logininfo where Username ='"+username+ "' and password = '" +password+ "'"))
    {
        int count=0; //counts the number of times the query is executed
        while(qry.next())
```

Figure 34: Sample c++ code



```cpp
#include "balancesheet.h"
#include "ui_balancesheet.h"
#include <QMessageBox>
#include "calculator.h"
#include <QDesktopServices>
#include <QFileDialog>
#include <QUrl>
#include <QRegExpValidator>
#include "cashflowsfinals.h"
#define TIME_RX "^([0-1][0-9]|[2][0-3]):([0-5][0-9])$"
#define TEXT_ME "^[a-zA-Z ]*$"
#define NUMBERQQ "^[0-9]*$"
#include "admindelete.h"
#include <QIcon>
balancesheet::balancesheet(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::balancesheet)
{
    ui->setupUi(this);
    setWindowIcon(QIcon("C:/Users/home/Desktop/julius/favicon-32x32.png"));
    QRegExp rx(TIME_RX);
    QRegExpValidator *vali = new QRegExpValidator(rx, this);
    ui->lineEdit_balancebf_2->setValidator(vali);
    QRegExp bm(TEXT_ME);
    QRegExpValidator *lol = new QRegExpValidator(bm, this);
    ui->lineEdit_particular->setValidator(lol);
    QRegExp ca(NUMBERQQ);
    QRegExpValidator *lal = new QRegExpValidator(ca, this);
    ui->lineEdit_balancebf->setValidator(lal);
    ui->lineEdit_balancecf->setValidator(lal);
```

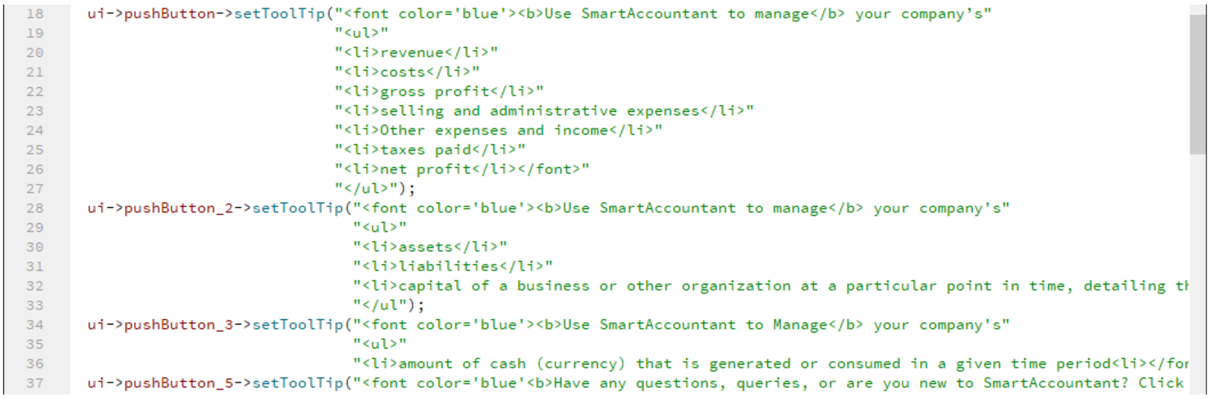
Figure 35: Sample c++ code



```cpp
    ui->label_4->setPixmap(pix);
    ui->pushButton_2->setToolTip("<font color='red'><b>Insert Particulars Only!!</b></font>");
        admin conn;
            if(!conn.connOpen())
                ui->label_tutor_status->setText("Couldn't connect to database");
            else
              ui->label_tutor_status->setText("Connected...");
}

balancesheet::~balancesheet()
{
    delete ui;
}

void balancesheet::on_pushButton_clicked()
{
    admin conn; //creating a new object with admin class
    QString newusername, newpassword, newnew, daters;
    newusername=ui->lineEdit_particular->text();
    newpassword=ui->lineEdit_balancecf->text();
    newnew=ui->lineEdit_balancebf->text();
    daters=ui->lineEdit_balancebf_2->text();
//    QRegExp regex("^(2[0-3]|[01]?[0-9]):([0-5]?[0-9])$");
//    QValidator *validator = new QRegularExpressionValidator(regex,this);
//    QLineEdit *edit = new QLineEdit(this)
    if((ui->lineEdit_particular->text().isEmpty())||(ui->lineEdit_balancebf->text().isEmpty())||(ui->lineEdit_balancecf->
    {
```

Figure 36: Sample c++ code

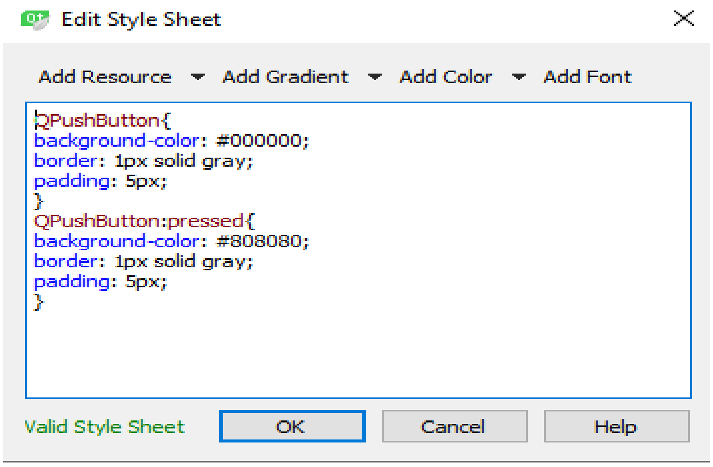
Sample html code:



```cpp
  ui->pushButton->setToolTip("<font color='blue'><b>Use SmartAccountant to manage</b> your company's"
                             "<ul>"
                             "<li>revenue</li>"
                             "<li>costs</li>"
                             "<li>gross profit</li>"
                             "<li>selling and administrative expenses</li>"
                             "<li>Other expenses and income</li>"
                             "<li>taxes paid</li>"
                             "<li>net profit</li></font>"
                             "</ul>");
  ui->pushButton_2->setToolTip("<font color='blue'><b>Use SmartAccountant to manage</b> your company's"
                               "<ul>"
                               "<li>assets</li>"
                               "<li>liabilities</li>"
                               "<li>capital of a business or other organization at a particular point in time, detailing th
                               "</ul");
  ui->pushButton_3->setToolTip("<font color='blue'><b>Use SmartAccountant to Manage</b> your company's"
                               "<ul>"
                               "<li>amount of cash (currency) that is generated or consumed in a given time period<li></for
  ui->pushButton_5->setToolTip("<font color='blue'<b>Have any questions, queries, or are you new to SmartAccountant? Click
```

Figure 37: Sample html code

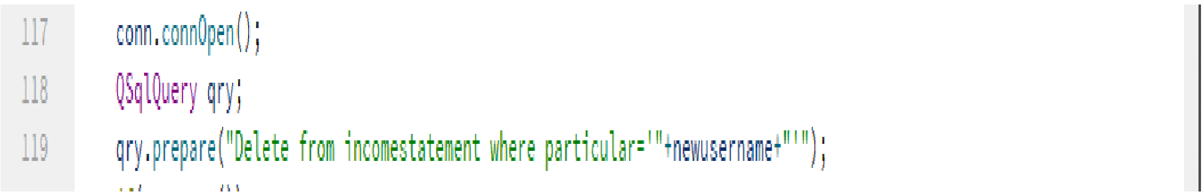
Sample css code:



```
QPushButton{
background-color: #000000;
border: 1px solid gray;
padding: 5px;
}
QPushButton:pressed{
background-color: #808080;
border: 1px solid gray;
padding: 5px;
}
```

Figure 38: Sample CSS code

Sample SQL code:



```
conn.connOpen();
QSqlQuery qry;
qry.prepare("Delete from incomestatement where particular='"+newusername+"'");
```

Figure 39: Sample Sql code