

# Valley View University

(<https://vvu.edu.gh/>)



FACULTY OF SCIENCE

DEPARTMENT OF INFORMATION TECHNOLOGY

A PROJECT SUBMITTED IN PARTIAL FULFILMENT OF THE  
REQUIREMENTS FOR THE BACHELOR OF SCIENCE (BSC.) IN  
INFORMATION TECHNOLOGY

TOPIC:

A SOCIAL PROTECTION MANAGEMENT INFORMATION SYSTEM  
(CASE STUDY INTO SOCIAL SECURITY AND NATIONAL INSURANCE  
TRUST)

BY

SOLOMON YAW ADEKLO

(<https://solomonadeklo.me/>)

(<https://github.com/solomonyaw>)

ID: 213DS01000274

DATE:

15<sup>TH</sup> MAY, 2020

## Contents

ACKNOWLEDGEMENT.....	4
CHAPTER 1.....	6
CHAPTER TWO.....	14
C2MS INSURANCE ERP SOFTWARE .....	30
3.2 Background of the Existing System.....	33
3.3 Review of Existing System.....	34
3.4 Components and Features.....	34
3.5 Weakness of The Existing System.....	34
3.6 Strength of The Existing System.....	35
CHAPTER FOUR.....	36
4.2.2 Non- Functional Requirements.....	37
The system will remotely register all informal sector workers (Since they constitute 80% of the working population in Ghana) on to the scheme to enable them to have better retirement packages in future. Also the system will reduce the cycle whereby they need to fill out forms in order to receive their retirement benefits. Also an online payments system will facilitate cashless payments of employee financial contributions and benefits.....	39
4.5.1 CONTEXT DIAGRAM OF THE PROPOSED SYSTEM.....	39
6.3 Testing the System.....	61
6.3.1 Description of the Testing Strategy .....	62
6.3.2 Statement of Test Case for the Proposed System .....	63
WEBREADER API INTEGRATION.....	65
CHAPTER SEVEN.....	67
USER MANUAL OF SYSTEM ADMINISTRATOR.....	67
7.1 System Home Page.....	67
7.1.1 System Login Interface.....	68
7.1.2 Admin Portal .....	68

7.1.3 Add Policy .....	69
7.1.4 Add New User .....	70
7.1.5 User Signup .....	70
This shows the signup portal where the user signs-up in order to be registered to have access to the system. ....	70
7.1.6 User(Employee or Customer) Portal .....	71
7.1.7 Customer Profile .....	71
This view allows the user to complete a registration form so that they can be allowed by the system to enroll in social security policies. ....	71
7.1.8 Enroll Policy .....	72
This view allows the user to enroll in social security policies. ....	72
7.1.9 Make Financial Contribution .....	73
This view allows the user to make financial contributions into social security policies. ....	73
7.1.10 Request Benefits .....	74
This view allows the user to request financial benefits from due to their financial previous obligations to social security policies. ....	74
7.1.11 View Report .....	75
This view allows the user to generate financial statement of benefits and contributions .....	75
CHAPTER EIGHT.....	75
CONCLUSION AND RECOMMENDATION.....	75
8.0 Conclusion.....	75
References .....	78

## **ACKNOWLEDGEMENT**

With great respect, I would like to express my most profound gratitude to God Almighty for giving me the strength and will power to complete this project, without Him I wouldn't have had enough intellect and strength to complete this project.

I am highly indebted to my parents, especially my educational sponsor, Mr. Gibson Kodjo Kumah (C.E.O of Gibson Construction Company Limited) who supported me throughout this research work. Not to be forgotten to the entire staff of the Department of Information technology(Faculty of science) who were always there for me and gave me tremendous support throughout the project development. Thank you so much for your support.

Special thanks to Mr Prince Yaw Amoako, the lecturer and my project supervisor at the Institute of Computer Science who provided me with a vast information I needed to successfully complete this project. With no exception, a lot of thanks to my friends and individuals that I failed to mention here, who directly or indirectly contributed to this project.

### **DECLARATION**

This is to declare that, the research work underlying this thesis has been carried out by the under mentioned student under the supervisor. Both student and the supervisor certify that the work documented in this thesis is the output of the research conducted by the student as part of his final year project work in partial fulfillment of the requirement of the Bachelor of Science in Information Technology degree.

STUDENT:

SOLOMON YAW ADEKLO

-----

SIGNATURE

DATE:

SUPERVISOR:

MR. PRINCE OWUSU AMOAKO

-----

SIGNATURE

DATE:

## **CHAPTER 1**

### **GENERAL INTRODUCTION**

#### **1.0 Introduction**

Social protection is now recognized as an essential public service in developing countries, with many having established social security schemes, such as old age pensions, child benefit, disability benefits and workfare schemes. It is essential that social security schemes are well managed and to help minimize fiduciary risk. Indeed, the credibility of cash transfer schemes depends on the right money reaching the right people at the right time. Also, there is the need to integrate the most marginalized groups in society into social protection schemes since they constitute the most vulnerable beneficiaries of the program. Women and especially children count to this groups. The protection of unattended people is and will still be a challenge. Underpinning the effective implementation of schemes are Management Information Systems. Reliable information management is of crucial importance for successfully implementing social protection programs. All core processes of social protection programs, starting from registration, targeting, enrolment and payment or delivery of cash transfers, require information that needs to be captured, stored, analyzed and made accessible. Apart from the identification of applicants and beneficiaries through registration and targeting, Social Protection Management Information Systems have an important role for programs administration, rights and grievance management and monitoring and evaluation purposes. The information captured for one program may provide the basis for the selection of beneficiaries for other social programs. They also provide an important evidence base for policy making processes. Their crucial role has also been recognized as a key area of reform in social

protection in developing countries. This paper proposed an integrated social protection management information system to register potential members to make contributions into social security schemes and also provide beneficiaries with online social assistance via a secure web portal.

## **SUBJECT AND FIELD OF STUDY**

The subject of this study is to develop a Social Protection Management Information System whereby potential members can be registered to make contributions into social security schemes and beneficiaries can apply online for social assistance via a secure web portal.

The field of study is Information technology, specifically web development.

### **1.1 STUDY OBJECTIVES**

The objectives of this study comprise both of general as well as specific objectives.

#### **1.1.1 Global (General) objectives**

To develop a social protection management information system to register potential members online to make contributions into social security schemes and to provide social assistance to beneficiaries of the scheme via a secure web portal.

### 1.1.2 Specific objectives

- Provide increased harmonization and consolidation of fragmented social protection schemes through comprehensive oversight of all social protection initiatives
- For better communication to the general population and target audiences
- To provide a single platform where common and essential information across social protection programs are stored, analyzed and reported for the benefit of the stakeholders
- To provide checks against one beneficiary receiving multiple benefits within and across program.
- Helps in the elimination of fraud by ensuring that beneficiary details are verified electronically against the database. This ensures that the programs pay only genuine beneficiaries.
- To help integrate marginalized informal workers or employees into social protection schemes and to encourage them to participate in social protection programs

## 1.2 BACKGROUND TO THE STUDY

Most countries implement anti-poverty or risk management policies and programs that are usually classified as a “social protection system.” Institutional arrangements vary widely across countries and regions but similar programs are often observed: cash or in-kind transfers to alleviate poverty and build human capital; mandatory social insurance programs to protect workers and their families against the risk of unemployment, disease,



disability, or the inability to work during old-age; regulations that protect workers from exploitation, abuse, or accidents at work; and programs that aim to help individuals build skills, find jobs, and, in general, improve their earnings opportunities. These programs usually fall under the purview of the Ministries of Labor, Social Affairs, and/or Health and are delivered through public or private providers, including social assistance and insurance institutions, employment service offices, training centers, or private insurance companies. Although the term "system" gives the idea of interconnected programs achieving interrelated functions, the reality in most cases is that of an amalgam of programs operating with little or no coordination. For instance, most insurance programs (pensions, unemployment benefits, health insurance) operate independently despite often being part of the same institution. Even within each of type of insurance program there is considerable fragmentation. Most countries, for example, have different pension programs for civil servants, formal workers in the private sector, and agricultural workers. The health insurance program is also often divided in two: social insurance for formal sector workers and national health services for the rest. There is even less coordination between social insurance and social assistance programs, or between these two and active or passive labor market programs. Research suggests that reducing fragmentation and promoting harmonization/integration can enhance both the performance of individual programs and the overall provision of social protection. With an integrated management information system people are less likely to fall through the cracks and lack coverage for a given risk or fail to benefit from assistance if poor or vulnerable. In addition, social protection programs can complement each other within an integrated system. This paper proposed social protection management information system to register potential members to make contributions into social security schemes and also provide beneficiaries with online social assistance via a secure web portal.

### **1.3 SCOPE OF STUDY**

The study is focused on a social protection information management system. The system is built using php to run on cross platform operating systems such as windows, IOS and Linux.

### **1.4 SIGNIFICANCE OF THE STUDY**

Integrated data management of social protection programs can lead to more equitable distribution of resources; provide oversight of multiple schemes; establish links with other services; and increase efficiency through economies of scale. It is generally agreed that MIS enable better organization of program services, help identify differences in coverage and detect anomalous situations. The social protection MIS will improve accountability, auditing, and transparency processes. More broadly, social protection MIS will contribute to better coordination of services and linking beneficiaries with other government programs. Information in the MIS can be used by other programs, for example to target different beneficiaries. Barca and Chirchir (2014) show that integrated MIS have achieved the following impacts, within limitations:

- ♣ More equitable approach to the distribution of resources based on objective and comparable information.

- ♣ Increased responsiveness and inclusiveness of interventions.
- ♣ Able to serve both the chronic poor and those structurally vulnerable to poverty, as well as respond to individual shocks or large crises.
- ♣ Increased transparency and accountability as program information can be more easily shared and compared.
- ♣ Facilitated oversight of multiple schemes and reporting to policy-makers.
- ♣ Avoidance of duplication of effort, for example in data collection activities for program targeting.
- ♣ Establishment of a common payment system across all schemes, increasing efficiencies and saving costs.
- ♣ Avoidance of fraud or simply keep track of who is receiving what. They also note a series of impacts which might be expected from MIS, but where there is little evidence of this type of experience:
- ♣ Increased linkage to the complementary institutional framework and wider social and economic policies in place.
- ♣ Increased knowledge on issues around poverty and vulnerability.
- ♣ Establishment of more effective emergency responses

## 1.5 METHODOLOGY

The methodology that would be used to design and construct this Social Protection Management Information System would be incremental build software model.

#### **INCREMENTAL BUILD SOFTWARE ENGINEERING MODEL**

The incremental build model is a method of software development where the model is designed, implemented and tested incrementally (a little more is added each time) until the product is finished. It involves both development and maintenance. The product is defined as finished when it satisfies all of its requirements. This model combines the elements of the waterfall model with the iterative philosophy of prototyping. The product is decomposed into a number of components, each of which are designed and built separately (termed as builds). Each component is delivered to the client when it is complete. This allows partial utilization of product and avoids a long development time. It also creates a large initial capital outlay with the subsequent long wait avoided. This model of development also helps ease the traumatic effect of introducing completely new system all at once.

#### **1.6 EXPECTED RESULT OF THE STUDY AND POSSIBLE USE**

Social protection programs which provide cash transfers and assistance directly to beneficiaries are an essential

aspect of the poverty alleviation strategies of many countries. However, administering social safety nets can be complex and burdensome, generating large volumes of data. Social protection agencies must therefore leverage on social protection information management systems as tools to deliver social benefits to poor and vulnerable people with greater efficiency and maximum impact. This project therefore provides social protection program managers and staff with a flexible, scalable and sustainable software solution to implement and track all phases of social safety net programs on a single platform.

## **PRESENTATION OF THESIS**

CHAPTER 1. General introduction (research proposal)

CHAPTER 2. Literature review

CHAPTER 3. Crystallization of the Research Problem

CHAPTER 4. Analysis of the Proposed System

CHAPTER 5. Detailed design of the Proposed System

CHAPTER 6. System Implementation and Testing

CHAPTER 7. System Documentation

CHAPTER 8. Conclusion and Recommendations

## **CHAPTER TWO**

### **LITERATURE REVIEW**

#### **2. INTRODUCTION**

This section presents a review of the existing literature on the useful role of software systems in the management of social insurance (protection) policies or programs and a review of how the social security system works in a developing country such as Ghana

##### **2.1.0 REVIEW OF RELATED WORK**

##### **2.2.0 GHANA SOCIAL SECURITY AND NATIONAL INSURANCE TRUST (CASE STUDY)**

###### **2.2.1 INTRODUCTION**

The Social Security and National Insurance Trust (SSNIT) is a statutory public Trust charged under the National Pensions Act 2008 Act 766 with the administration of Ghana's Basic National Social Security Pension Scheme and to cater for the First Tier of the contributory three-tier scheme. The Trust is currently the largest non-bank financial institution in the country. The primary responsibility is to replace part of lost income of workers in Ghana due to Old Age, Invalidity and permanent migration of an expatriate contributor from Ghana. The Trust is also responsible to pay survivors Lump sum to nominated dependants of contributors in the event of death. The Pension Scheme as administered by SSNIT has a registered membership of over **1,307,882 million** as at **August 2017** with over 184,761 pensioners who regularly receive their monthly pensions from SSNIT. The annual absolute growth of pensioners is over 12,000.

###### **WHAT DOES SSNIT DO?**

SSNIT has a primary duty to collect contributions to pay pensions and other benefits as they fall due. In carrying out this primary responsibility the Trust undertakes the following specific duties:

- Register all employers and issue them with unique Establishment Registration (ER) numbers.
- Register all members and issue them with individual unique Social Security number and a Smart Card.
- Collect contributions of members and compile relevant data related to the contributions.
- Maintain and update personal and financial records on members.
- Process benefits for members as they fall due.
- Manage and invest the social security funds.

## **THE TRUST**

The Trust was established in 1972 under NRCD 127 to administer the National Social Security Scheme. Prior to 1972, the Scheme was administered jointly by the then Department of Pensions and the State Insurance Corporation. The Trust administered the Social Security Scheme as a Provident Fund Scheme until 1991 when it was converted to a Social Insurance Pension Scheme then governed by the PNDC law 247. The scheme in Ghana was reformed by an Act of Parliament, Act 766 of 2008 and was implemented in January 2010 to replace all pension schemes in Ghana including Cap 30. In 2014, the National Pensions (Amendment) Act 883 was passed to amend portions of Act 766.

## **THE CORE FUNCTIONS OF SSNIT**

- o Register employers and workers
- o Collect contributions
- o Manage records on members
- o Invest the funds of the Scheme
- o Process and pay benefits to eligible members and nominated dependants.

## **ADMINISTRATION OF SSNIT**

The day-to-day administration of the Trust is headed by the

Director General assisted by three (3) Deputy Director Generals (DDGs).

There are seven (7) **General Managers** in charge of **Investment and Development (IDD)**, **General Counsel (GC)**, **Finance, Operations, Administration and Human Resources (ADMIN/HR)**, **Management Information Systems (MIS)** and **Benefits**. Other responsibilities under the Directorate include that of the **Chief Actuary**, the **Chief Internal Auditor**, the **Company Secretary** and the **Corporate Affairs Manager**.

#### **BENEFITS UNDER THE SSNIT SCHEME**

There are four (4) types of benefit under the SSNIT scheme that members can enjoy depending on which contingency has occurred.

- Superannuation Pension/ Old age Pension
- Invalidity Pension
- Survivor's Lump sum.
- Emigration benefit

#### **SUPERANNUATION PENSION/ OLD AGE PENSION**

To qualify for old age pension, the member must be at least 60 years and must have contributed a minimum of 180 months (15 yrs.) under act 766 and 240 months (20 yrs.) under PNDCL 247. The member who is 55 years but below 60 years receives reduced pension whilst the 60 year old receives full pension.

#### **OLD AGE/RETIREMENT PENSION**

##### **Qualifying Conditions**

##### **FULL PENSION**

To qualify for Full Pension,

- o You must be at least 60 years and
- o You must have made a minimum contribution of 180 months (15yrs.) under act 766 and 240 months (20 yrs.) under PNDCL 247.

#### **BASIS FOR CALCULATION OF OLD AGE PENSION**

- ☐ Age
- ☐ Average of Best 36 months / 3 years' Salary
- ☐ Earned Pension Right - Rating for the number of months you have contributed to the Scheme.

You can earn a "pension right" between 37.5% and 60% depending on the number of months contributed at the time of retirement.



E.g. the minimum contributions of 180 months gives a "pension right" of 37.5%. Every additional month over the 180 months attracts an additional percentage of 0.09375% or 1.125% for One (1) year respectively.

### **TO CALCULATE YOUR PENSION**

Multiply your best 36 months (3 years) average salary by your "pension right".

### **EARNED PENSION RIGHT UNDER THE NATIONAL PENSION ACT, 2008 ACT 766**

The Pension Right is 2.5% for each year of contribution for the first 15 years and 1.125% for every additional year up to a maximum of 60.0%.

### **INVALIDITY PENSION**

To qualify for invalidity pension, the member must have contributed for 12 months in aggregate within the last 36 months preceding the incidence of the invalidity. He/ She must have also been certified by a Medical Board as being incapable of any normal gainful employment due to a permanent physical or mental disability.

The invalidity pension is paid monthly to such a person who has been confirmed and certified incapable of earning an income.

### **INVALIDITY PENSION**

#### **Qualifying Conditions**

To qualify for invalidity pension:

- You must have made a minimum of 12 months/within the last 36 months with the date of termination of appointment due to your invalidity as the reference point.
- You must have been declared permanently invalid and incapable of any normal gainful employment:  
By a qualified and recognized medical officer and  
Certified by a Regional Medical Board on which a SSNIT
- Medical Officer is represented.  
MEDICAL EXAMINATION
- Report to the nearest SSNIT Branch in person or through your representative with a Medical Report from a recognized Medical Practitioner certifying you are invalid.
- You will be required to appear before a Medical Board for

examination where your invalidity is certified by the Medical Board.

#### **DEATH AND SURVIVORS LUMP SUM**

Death and Survivors Lump sum is benefit paid to nominated or eligible beneficiaries of a deceased contributor or pensioner. It can also be paid to persons upon an order by a court of competent jurisdiction. This benefit is paid when a member dies in active service or during retirement and have not attained the age of **72 years for those under PNDCL 247 and 75 years for those under the National Pensions Law, Act 766 respectively**. Deceased member's beneficiaries will be identified and all benefit due paid to the beneficiaries identified.

#### **SURVIVOR'S LUMP SUM**

This benefit is paid to dependants of members under the following conditions:

- When the member dies before retirement; or
- When a pensioner dies before attaining age 75.

#### **CALCULATION OF SURVIVORS' LUMP SUM**

The benefit is computed as follows:

- Where a member dies having made at least twelve months contributions within the last 36 months prior to the death of the member, a lump sum payment of the earned pension of the deceased member for a period of 15 years will be paid.
- The amount will be the present value discounted and paid to the member's nominated dependants.
- Where a member dies before making at least twelve months contribution within the last 36 months, a lump sum equal to the total contributions and interest shall be paid to the nominated dependants.
- Where a pensioner dies before attaining age 75, a lump sum payment based on the present value of the unexpired pension of the member will be made to the beneficiaries.

#### **EMIGRATION BENEFIT**

Emigration benefit is a lump sum payment of benefit to non-Ghanaian members of the Social Security Scheme under Act 766

whose services are ended and are leaving Ghana permanently. Whether the member has reached the retiring age or not, whatever benefit is due him/her will be paid as lump sum in Ghanaian currency to the member.

To qualify for emigration benefits:

- You must be a non-Ghanaian contributor.
- You should be leaving Ghana Permanently.
- 

### **2.1.1 REVIEW OF EXISTING SYSTEMS**

#### **ISI ENTERPRISE**

ISI Enterprise is a fully-integrated browser-based insurance policy administration software that helps you reengineer the way insurance carriers do business. Most legacy systems were built to be policy-centric, but ISI believes in putting the clients at the heart of system, providing an end-to-end policy administration system with a "client-centric" approach. Its advanced architecture runs from a single database and code base with real time processing. The solution was designed to help insurers achieve maximum benefit from their IT systems.

#### **CORE MODULES OF ISI ENTERPRISE**

##### **1. Policy Administration System :-**

Imagine having a top-down, complete view of every client portfolio. ISI Enterprise Policy Administration makes this a reality by providing users a client-centric interface in order to manage the full policy administration lifecycle.

ISI takes a broader view of what Policy Administration means to insurers, and thus leverages the fully-integrated nature of the ISI Enterprise solution to incorporate claims, risk control, reinsurance, billing and financial information into Policy Administration workflows.

##### *Key Features*

- Automated quote and policy life cycles
- Full transactional versioning
- Out-of-sequence transactions

- Central repository for client communication and documentation
- Flexible, configurable hierarchical policy structure
- Policy levels provide attachment points for associating rating rules, documents and underwriting data elements
- Expandable areas for gathering risk details
- Automated and manual renewal features
- Two-tiered transaction locking
- Unique user interface to facilitate multi-tasking
- Bulk import feature for large policies

#### *Key Benefits*

- Empowers users by providing integrated client information and at their fingertips
- Rules-based underwriting system increases efficiency
- User-friendly interface that support multi-tasking for optimal productivity

## **2. Billing**

ISI Enterprise offers robust billing functionality for both direct and broker/agency bill for all lines of business. Tight integration with the Policy Administration module allows for a quick view of billing status at both the client/account and policy level, with the flexibility of modifying billing independent of the policy. The Billing module streamlines the billing process through tools such as automated statement creation and delivery, Excel-based copy-and-paste account reconciliations and batch receipt functionality.

#### *Key Features*

- Supports both direct bill and broker/agency bill for all lines of business
- Configurable and flexible installment/payment plans
- Configurable taxes, fees and charges including complex rules based calculations
- Integrated installment schedules with invoicing, credit card transactions and ACH/Automated Withdrawal
- Rules-based cancellation for non-payment functionality

#### *Key Benefits*

Flexible billing options catering to any distribution model

- Flexible payment plans and installment schedules
- Accurate financial data promotes effective and timely decision making
- Allows for billing changes independent of the policy
- Multi-currency facilitates operations in multiple geographical regions
- Quick view of billing status at any level

### **3. Reinsurance**

Most insurers think of reinsurance processing as an afterthought. In fact, in a recent survey, more than 80% of property and casualty insurers handle their reinsurance processing external to their systems using manual processes or spreadsheets. There is a better way. ISI Enterprise allows insurers to configure their reinsurance treaties and facultative arrangements and automatically integrate reinsurance processing within the policy administration and claims lifecycles. ISI Enterprise was architected to handle even the most complex reinsurance requirements, including retentions, layering, complex attachment criteria and pooling.

#### *Key Features*

- Simple user interface to set-up and maintain treaties
- Visual presentation of reinsurance according to treaty layering
- Automatic assignment of reinsurance during policy and claims transaction
- Ability to trace reinsurance calculations
- Ability to associate reinsurance at coverage and policy levels
- A centralized module to manage all interactions with reinsurance partners
- Bordereau report generation
- Grouping mechanism to manage reinsurance partners
- Limits for setting up reinsurance are tied to user security
- Rules-based attachment criteria

#### *Key Benefits*

- Transactions automatically flow through to the accounts of

reinsurers for ceded and recovery reinsurance transactions as well as receivables and payables

- Flexibility in assigning and validating conditions for treaty set-up
- Ability to trace reinsurance calculations when unexpected results occur due to treaty set-up

#### **4. Accounting**

The integrated accounting module automatically captures all insurance and general business transactions processed anywhere within ISI Enterprise in real-time. Your financials are always up to date and you can manage your business in real time. Define all financial information in a customized, flexible chart of accounts which supports multiple underwriting companies and an inter-company configuration. Multi-currency capabilities allow you to operate efficiently across borders. Receive financial roll-ups for each business unit, multiple branches, countries, geographical regions, subsidiaries and parent companies. Maintain up-to-date sub-ledgers for all key business partners.

#### ***Key Features***

- Configurable, flexible Chart of Accounts
- Setup automatic postings, fiscal year and periods, cheque series and currency rates
- Perform bank reconciliations
- Fully-integrated with all other ISI Enterprise modules
- Transactions posted in real-time
- Supports multi-layered company structures
- Multi-currency capabilities

#### ***Key Benefits***

- Real-time posting of all financial transactions from all modules of the system
- Up-to-date financial information
- Accurate financial data promotes effective and timely decision making
- Collect data in multiple ledgers, accounts payable, accounts receivable and General Ledger

- Multi-currency facilitates operations in multiple geographical regions
- Analysis of financial data with roll-ups for each company,
- multiple branches, countries, geographical regions,
- subsidiaries and parent companies.

## **5. Rating & Rules Engine**

The integrated Rules Engine allows you to load and maintain rates and business rules for all your standard and non-standard lines of business. The intuitive interface is easy to use; rates can be copied from Microsoft Excel, allowing business users to effectively manage rates without programming intervention. The Rules Engine also enables straight through processing of policies you define as standard - based on various parameters or user inputs - and tells the system to ensure that all unusual risks are analyzed. To ensure consistency and improve compliance you can systematically make complex decisions inside the Rule Engine module.

## **6. Configuration Studio**

One of the most common reasons insurers seek to replace their policy administration systems today is the inability to roll-out new products and make ongoing changes quickly to respond to market demands.

Configuration Studio allows insurers to implement company specific requirements for products, claims, billing, accounting and workflow without having to involve programmers or modify source code. Configuration Studio includes a robust rules engine which facilitates the implementation of rates and business rules throughout the core modules in ISI Enterprise.

### *Key Features*

- Product configuration tools to define and manage product definitions based on flexible package or monoline structures
- Productivity tools which streamline the configuration process including copy functions, Excel bulk import and pre-configured templates which can be leveraged to reduce time-to-market
- Full version management for all configuration elements based on dates for new business and renewal

- Screen configuration tools which enable expansion of the base data model
- Robust document configuration tools to produce and manage declaration pages, forms, letters, invoices and other communications
- Integrated ETL tool to facilitate data exchange with 3rd party systems and services

#### *Key Benefits*

- Perform initial implementation and ongoing maintenance changes without modifying source code, minimizing risk and costs
- All company-specific configurations are table driven and forward-compatible as new system versions are released
- Insurers have the option of becoming self-sufficient in configuration without having to rely on vendor support for ongoing changes
- 

## **MUTUAL EXPERT**

Mutual Expert's [dot] Expert is a Web-based insurance platform built using the latest technologies and informed by an extensive industry experience. With [dot] Expert, you gain a powerful yet elegant solution that integrates all facets of your insurance operation. Mutual Expert is the insurance ecosystem created for mutual insurance companies that elegantly integrates all aspects of a carrier's business.

### ***But what exactly is an ecosystem?***

An ecosystem is a group of interconnected elements found within a given community. When it comes to insurance carriers, Mutual Expert successfully unifies the different elements of a carrier's business. For example, within the company itself, Mutual Expert seamlessly integrates the underwriting department with the claims department and the accounting department.

## **MODULES OF THE MUTUAL [DOT] EXPERT**

### **1. POLICY**

The [dot] Expert policy module gives you complete control over policy rating, underwriting and management. The powerful rating engine was engineered to handle both personal and commercial lines of business. It also gives you the flexibility you need to maintain your own insurance products.



## *Features*

- Configurable underwriting rules, rates, and formulas.
- Customizable user inbox for company messages and assignments.
- Complete inspection tracking, from initial request to compliance issues.
- Third party integrations to simplify processing.
- Automated processing based on carrier-defined rules.

## **2. BILLING**

With [dot] Expert, you can reduce your carrier's operational cost through automation and batch processing. The billing module streamlines the routine process, enabling your billing department to focus on the situations that demand their attention.

## *Features*

- Company-defined payment plans and fee structures.
- Transfer money between different accounts and policy periods.
- Sweep agency and/or policyholder accounts for electronic payments.
- Integrate with online payment processors.
- Email billing reminders based on company rules.

## **3. CLAIMS**

The [dot] Expert claims management module enables your company to track multiple claimants per accident, each with their own reserves and payments. By allowing both agents and policyholders to submit first notices of loss, your company can process claims quickly and, in turn, improve customer service.

## *Features*

- Built-in diary and reminder system.
- Complete litigation tracker.
- Record salvage and subrogation.
- Track individual coverage and peril items.
- Issue Loss and LAE payments via check or EFT.

## *Features*

- Custom report designer for ad-hoc reporting.
- Schedule automatic report runs and deliveries.
- Report data grouping and filtering capabilities.
- Premium and claim reinsurance reports.
- Schedule P and loss ratio analysis reports.

## **PARTNER XE**

Partner XE is a cloud-based insurance agency management system for independent agencies.

Partner XE offers accounting, claims management and policy management. Using the system's integrated accounting functionality, users can manage direct billing, insurance agency billing, bank reconciliation and more. A document management feature allows users to digitize and organize files and information. Users can also customize naming conventions, data entry preferences and dashboard views.

The software integrates with Microsoft Outlook to capture email communications within the Partner XE system. It also integrates with WebLink, UpLink and IVANS Transformation Station to pass information directly to carriers.

## **CAPABILITIES OF PARTNER XE**

### **1. Integrated Accounting**

Accounting for your insurance agency should be easy and accurate. With Partner XE's fully integrated insurance agency accounting your client information, policy information, and financial information is all organized in one system. Eliminate third party insurance software and dual entry giving your agency real time insights into key information to better serve your clients and more effectively grow your business.

## *Features:*

- Daily processing dashboard
- Multi-level insurance producer commission splits
- Sweep transactions
- Carrier direct bill commission download
- Direct billing and reconciliation

- Insurance agency billing and reconciliation
- Bank reconciliation
- Producer reconciliation
- Vendor payables
- Branch, division, and department accounting
- Financial and production reports
- Producer reports
- Balance sheet
- Income statement
- Expense report
- Aged accounts receivable report
- Branch reports

## **2. Document Management**

Paperless document management for your insurance agency is critical, and your agency management system is the best place for these capabilities.

Partner XE's document management enables you and your staff to quickly and easily attach information as well as retrieve key documents. You can auto attach, drag and drop, and customize how you organize document folders to make it simple to find what you need for rapid client response time.

### ***Features:***

- easily attach information
- quickly find documents
- organize your way
- reduce e&o risk

## **3. Email Integration**

Capturing email communications in your insurance agency management system is essential for E&O and strong client service.

Partner XE is designed to work seamlessly with Outlook, enabling significant productivity improvements without any change in how you already work. Agencies leverage the auto attach capability to ensure all client inbound and outbound email communications are captured in the system.

This unique innovation and one of a kind email integration saves insurance agencies hours of work while ensuring better E&O protection.

## **4. Personalized Workflows**

Your independent insurance agency is unique. You have your own way of doing business. You have your own processes, naming

systems, and priorities. Instead of making you adapt all processes to your agency management system, Partner XE is designed to adapt to your agency. Experience freedom with custom naming conventions and multiple options for data entry, without losing data consistency. This allows for strong client service and standardized reporting that helps you run your agency. Run your business the way you want. Stay independent. Your Partner XE agency management system will make it easy to work the way you want to.

*Features:*

- create custom naming conventions and use your agency's
- vocabulary.
- customize your dashboard and tracking views.
- take the paths you prefer to enter and access data.

## **5. Reporting**

Produce reliable insights with the click of a button to manage your staff, your business, and your clients. Gain valuable, real time information on agency finances, client retention rates, new policy business growth, and more without specialized knowledge. For custom agency reports, SIS provides expertise and experience service team to help you learn how to access the information or create the reports for you.

*Features:*

- simple to use
- key insights
- accurate information
- standard and custom reports

## **INSURACCESS**

Insuraccess is a simple, secure solution for international insurance programs reporting. The solution is a reporting platform designed to assist insurance brokers in managing the international insurance programs of their clients. Insuraccess can be used as a private portal to give insured clients access to all their local policies in one centralized and secure location.

Insuraccess provides an intuitive user interface that makes it easy for any broker from any country to easily report their insurance programs. The solution substantially streamlines and

improves the reporting of International Insurance Programs with local brokers around the world.

Insuraccess offers via the producing broker website, a private portal to multinational clients with secure, 24/7 access to all their local insurance policies and related documents.

## FEATURES OF INSURACCESS

### **1.Easy to Use**

With a simple, intuitive design there is a low learning curve and no advanced technical skills needed.

### **2.Export**

By exporting policy schedule of values to an Excel spreadsheet for further analysis, Insuraccess acts as an effective reporting & analysis tool for international clients.

### **3.Client Portal**

Clients are immediately connected to their policy information with virtually no extra time & energy from the broker.

### **4.Currency Converter**

Once invited, local brokers upload policies and enter the requested information using their local currency. The system converts all values to the producing broker pre-determined currency at the rate of the day.

### **5.Custom Permissions**

Custom permissions allow the producing broker to choose what the client can or cannot see.

### **6.Monitoring**

The system identifies which brokers who have not uploaded a requested document and allows the producing broker to send them a reminder.

## **C2MS INSURANCE ERP SOFTWARE**

C2MS is a proven and innovative InsurTech Cloud Insurance Software sales, claims and administration platform, designed for Commercial and Personal Lines Insurance. C2MS Cloud Insurance Software is specifically engineered to make it easier and more profitable for Insurers, MGAs and Scheme Managers to do business via the Internet and add value to traditional Insurance channels.

It is evident that significant business advantage can be gained by introducing innovative transaction and self-service automation to Commercial Lines business, this has been witnessed by the increased automation of Personal Lines over recent years, and the access to new markets that this has enabled.

By harnessing the power of Cloud technology, C2MS can scale from a small efficient setup, to a large autoscaled environment, capable of processing millions of policies, reliably, with predictable costs.

Key to the success of C2MS Cloud Insurance Software, is its ability to simplify commercial and personal lines transactions and policy administration for all users, while increasing productivity and reducing transaction costs.

This is achieved by providing enhanced levels of transaction automation and self-service facilities, which speeds up the trading process and enhances communications.

The increases in efficiency brought by C2MS can translate to an increase in capacity, by improved productivity. This provides the opportunity to grow revenue by utilizing the excess capacity to explore new channels or bring new products to market quickly.

*Features:*

### **Website Portals**

Bring new and existing products to market through various sales channels; self-service portals, mobile apps, broker networks, aggregator websites, White Label schemes or direct to market.

## **Broker, Agent and Consumer CRM**

Powerful CRM tools to manage Brokers, Agents and Consumers. Automate renewal invitations, provide web logins and self-service facilities. Manage relationships between accounts and assign account managers.

## **Aggregator Support**

Seamlessly integrate with leading aggregators using C2MS APIs and surface rating engine services which provide real time quotes. Connected to MoneySuperMarket, GoCompare, Compare the Market, Confused.com and more.

## **Powerful Product Configurator**

Using powerful GUI tools, create and configure Products from the data modelling steps through to validation, logic and mapping for web services and reporting.

## **Policy Life-Cycle & Administration**

Handle business from the initial quote through to renewal, supporting all types of BTE and ATE Insurance. Automation tools assist with administration of renewals and lapsed policies.

## **Securely Process Payments**

Securely process various forms of payments directly within C2MS, both within quote and buy portals and backoffice, including credit and debit cards, BACS, direct debit and premium finance.

## **Powerful Commission Engine**

Fully configurable commission rules, based on product, insurer, channel and more. Automation of commission schedules and even support for splitting commission payments in various ways, over a set period.

## **Underwriting Tools**

A full suite of tools to manage ratings, matrices, equations, referrals and business logic. Change rules and rates in real time and setup individualised rates per channel and/or customer.

## **Accounting**

A fully integrated, double-entry transactional accounting system, designed to streamline and automate complex accounting workflows. Additional workflows can be added to partially or fully automate many accounting functions.

## **Reporting**

Pre-created real-time and customisable reports for common tasks (Accounting, Marketing, Sales, Debtors, Claims, Renewals, Logs, Agents & Brokers and more...). Support for 3rd party tools such as AWS Quicksight and Microsoft PowerBI.

## **Claims Management**

Complete claim life-cycle, from FNOL through to reimbursement. Manage claims using a variety of tools, including Mobile App technology for submission, dedicated claims portals for agents, brokers and claims administrators.

## **Document Management**

Creation of documents and emails from predefined templates, based on workflows and events. Self-manage all document and email content.

## **Single Tenancy Cloud Technology**

C2MS is Cloud Ready, supporting all major hosting providers, but can also be run On-Premise. C2MS operates within a virtualised environment, with dedicated resources and separate databases. No customer data is shared.

## **API-Centric**

Providing real-time connections and feeds to and from third parties; whether aggregators, postcode lookup, car registration lookup, flood and risk databases or partners. Both SOAP and REST protocols are supported.

## **Advanced Security for Data Protection**

Single Sign-On and 2-Factor Authentication services, provided by AuthStack and MFAStack, enables the latest in security and data protection technologies for your users and customers.



## CHAPTER THREE

### CRYSTALLIZATION OF THE PROBLEM

#### 3.1 Introduction

This chapter will take a concrete look at how the current existing system works and also try to unlock the inefficiencies that characterizes the system so as to ensure that an efficient system is developed.

A complete and thorough explanation of how the existing system works is represented with the aid of a context level diagram together with their advantages and disadvantages.

#### 3.2 Background of the Existing System

Social protection management system are useful for the integrated data management of social protection programs which can inadvertently lead to more equitable distribution of state resources. Unfortunately, the social protection system employed in the **Ghana Social Security and National Trust (SNNIT)** is a manual system which does not:

- provide superior services to SNNIT customers or contributors
- reduce member enrolment cycle through forms,
- provide effective reporting solution,

- achieve real time processing of contribution reports
- reduce benefit processing time.

### **3.3 Review of Existing System**

### **3.4 Components and Features**

The existing system at **SSNIT** requires that members visit the regional offices to register for the scheme using a national identification card and other recommended cards. An Automated Fingerprint Identification System (AFIS) will be used to verify the identity of members and pensioners. This system stores the biometric information of all members and pensioners in a database for the verification of the identity of any member or pensioner at any SSNIT branch throughout Ghana. A thirteen(13) alphanumeric Social Security Numbers (SSNO's) is generated for existing members with 8 Digit Social Security (SS) numbers. To apply for benefits, members must contact the nearest **SSNIT** Branch with their Smart Card or Biometric Card and a letter of retirement from their employer—not mandatory. The **SSNIT** Branch will then provide the member with a Pension Application Form for completion. The completed Form is then submitted to the **SSNIT** Branch.

### **3.5 Weakness of The Existing System**

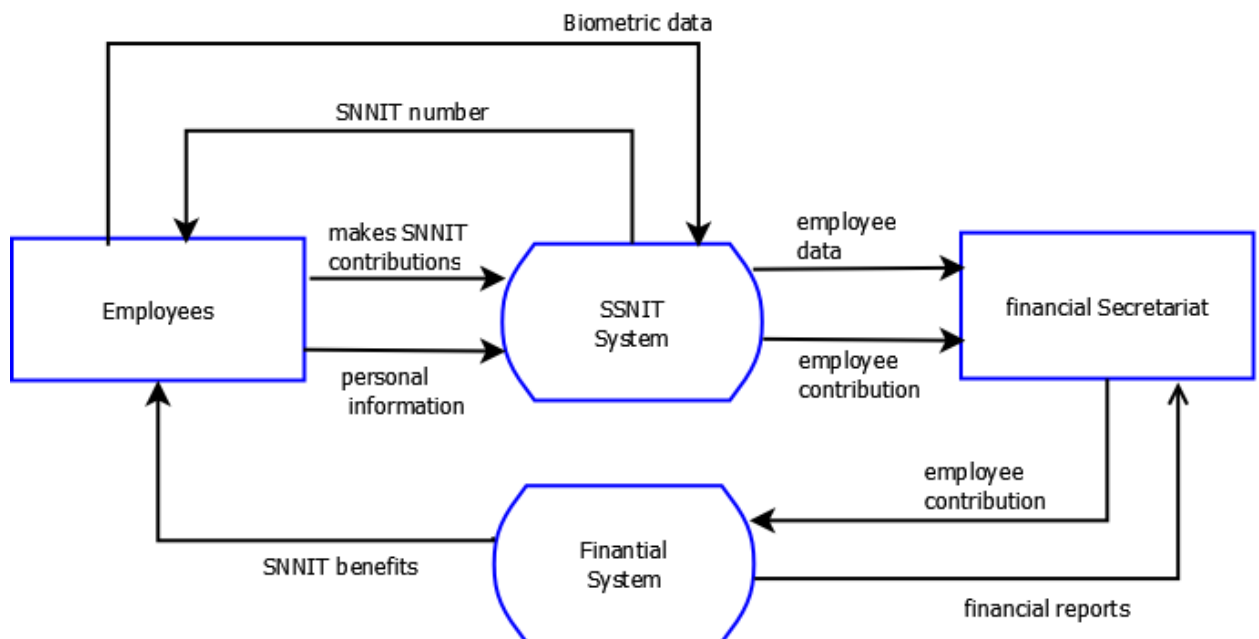
This manual process of registering and providing benefits to members is inefficient because the repetitive cycle whereby members need to fill out application forms for every benefit scheme that they apply for is redundant.

### 3.6 Strength of The Existing System

The use of Automated Fingerprint Identification System (AFIS) enables SSNIT to collect biometric information of all existing SSNIT members and pensioners for easy verification. This therefore prevents multiple registration of members.

### 3.7 Context Diagram

The context level diagram of the system drawn below:



## **CHAPTER FOUR**

### **ANALYSIS OF THE PROPOSED SYSTEM**

#### **3.0 Overview of the System**

In this section, a comprehensive description of the proposed system is stated, the potential requirements and procedures of the system are stated. Systems requirements present services provided by the new system. Data modeling tools like the context level diagram are used. The context level diagrams point out the main features of the system. Also is the external entities and processes of the system proposed.

#### **3.1 System Requirements**

##### **3.1.1 Functional Requirements**

The functional requirement(s) of the proposed system describes what the system offers to meet the user's needs.

Their requirements include:

- ✓ The system will register users and issue them with unique Establishment Registration (ER) numbers.
- ✓ The system will Register users and generate a unique Social Security number for individual users
- ✓ The system will collect contributions of users and compile relevant data related to the contributions.
- ✓ The system will maintain and update personal and financial records on users.

- ✓ The system will process benefits for users as they fall due.

#### **4.2.2 Non- Functional Requirements**

Non-functional requirements are requirements which impose constraints on the design or implementation such as performance requirements, security, and reliability: These includes:

##### **RELIABILITY**

Some of the reliability features are:

- ✓ The system should provide an update of all contributions made by the user/employee.
- ✓ The user should be able to rely on the system

##### **PERFORMANCE**

The performance requirements include:

- Database to keep records of users and their contributions
- The system should perform all its function

##### **USABILITY**

The usability requirement of the system includes:

- Users should use the system without difficulty.
- There should be logical interface to speed up the process.

##### **SECURITY**

The system has specific features that will prevent breaches and are as follows:

- ✓ Data integrity check
- ✓ The pension scheme system shall ensure that the benefit data accrued to the beneficiaries of the scheme can be accessed only by authorized users.

#### **ACCESIBILITY**

- ✓ The system shall be accessible to people with disabilities in accordance with the Ghanaian with disability Act, 2006 Act 715

#### **PORTABILITY**

- The system should efficiently be accessible on every web browser
- The system is providing accurate and reliable data.
- The system should be user-friendly.

### **4.3 Major Features/Components of the Proposed System**

The main features of the system will be:

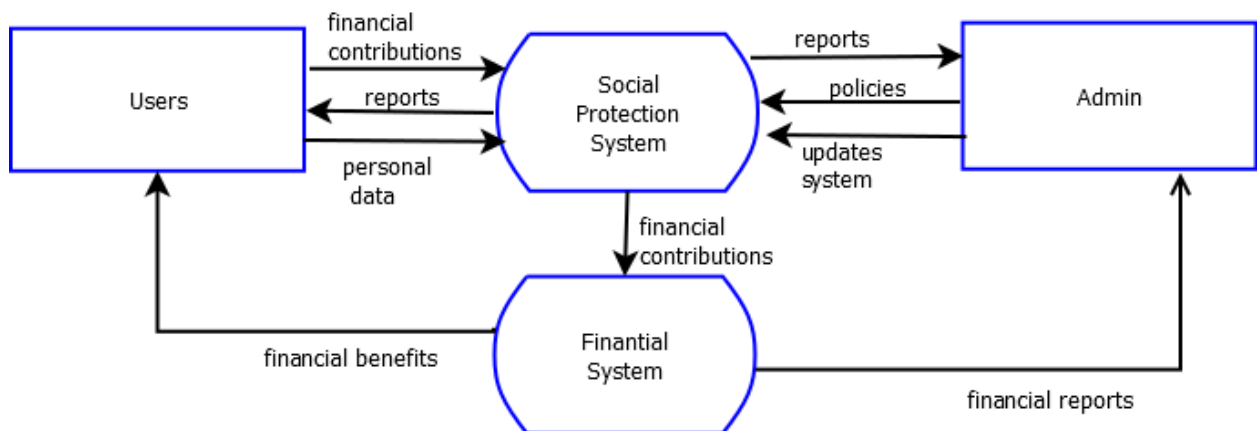
- a. Employees register on the website and login to view their account page.
- b. They can complete their profile information on the site.
- c. Employees can enroll on Social Security scheme.
- d. After enrolling employees will make contributions to the scheme. An online payment system will be integrated
- e. System will also display the benefits information that the user receives after retirement

- f. Employees can receive financial benefits from the social security scheme.

#### 4.4 Benefits of the Proposed System

The system will remotely register all informal sector workers (Since they constitute 80% of the working population in Ghana) on to the scheme to enable them to have better retirement packages in future. Also the system will reduce the cycle whereby they need to fill out forms in order to receive their retirement benefits. Also an online payments system will facilitate cashless payments of employee financial contributions and benefits.

##### 4.5.1 CONTEXT DIAGRAM OF THE PROPOSED SYSTEM



## **CHAPTER FIVE**

### **DETAILED DESIGN OF THE SYSTEM**

#### **5.1 Functional Purpose of the Proposed System**

This chapter elaborates all the significant modules of the proposed system that would be outlined, followed by the individual functional processes within the modules or components of the system.

The functional process diagram of both the Administrator and user is also stated in this section. The task that the administrator/user can perform under the administrative module.

5.2 Algorithm and flowchart of the proposed system.

##### **5.2.1 Algorithm for Administrator**

STEP 1: START

STEP 2: Administrator login

STEP 3: Login authentication

STEP 4: Add social protection policy

STEP 5: Create users

STEP 6: View users

STEP 7: View reports

STEP 8: Approve and process financial benefits to users



STEP 9: Logout

STEP 10: Return to Step 1

STEP 11: Stop

### **5.2.2 Algorithm for employees**

STEP 1: START

STEP 2: User login or go to register

STEP 3: Complete registration form or user profile

STEP 4: Enroll social protection policy

STEP 5: Make contributions or payments

STEP 6: Request Benefits

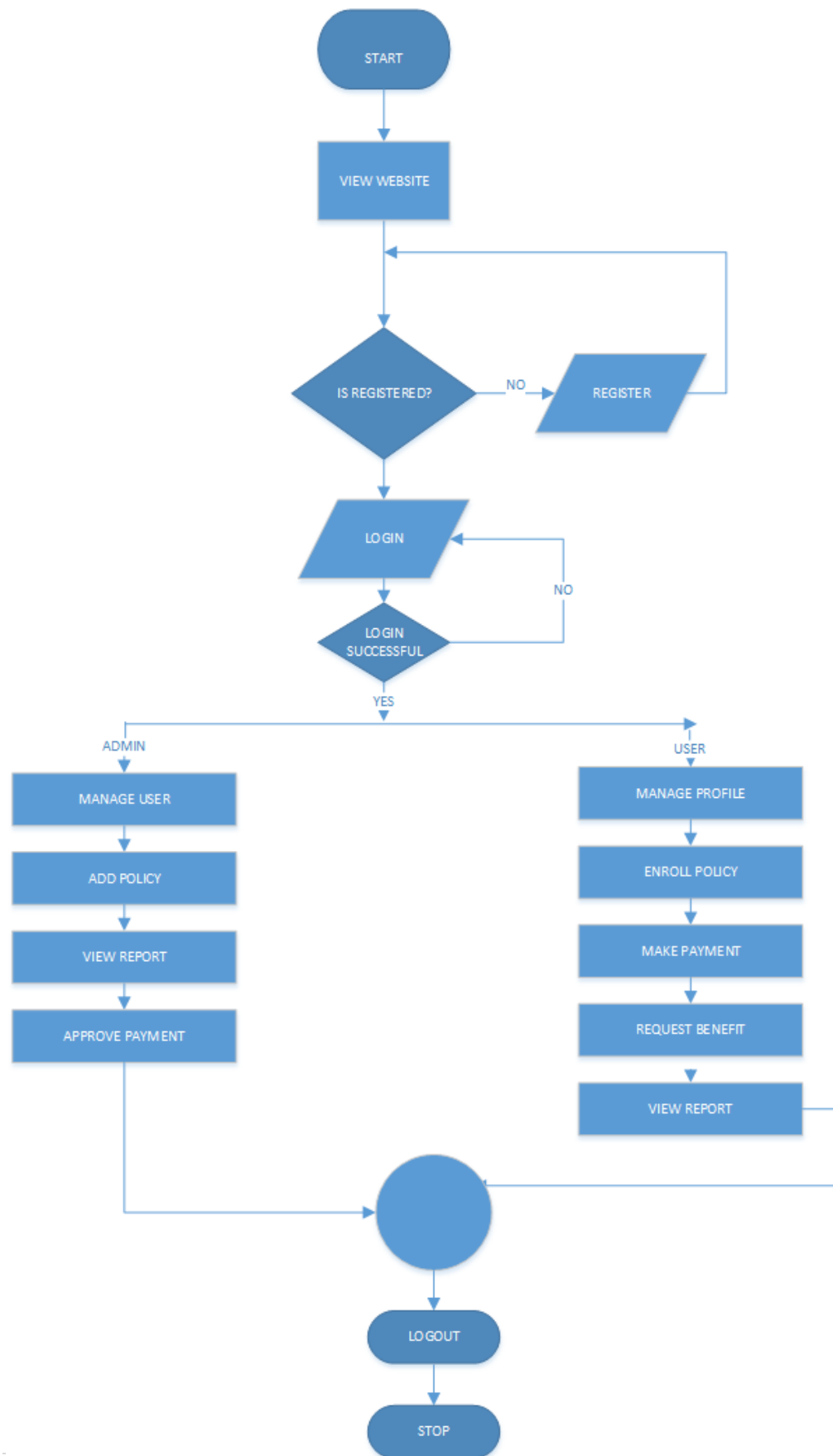
STEP 7: View financial reports

STEP 8: Logout

STEP 9: Return to Step 1

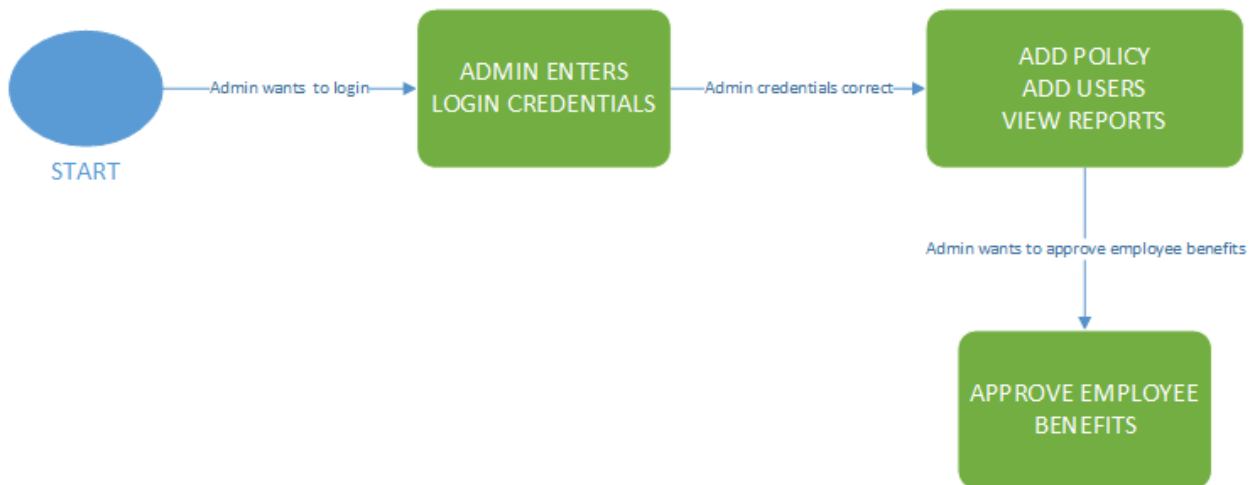
STEP 10: Stop

BELOW IS THE FLOWCHART OF THE SYSTEM

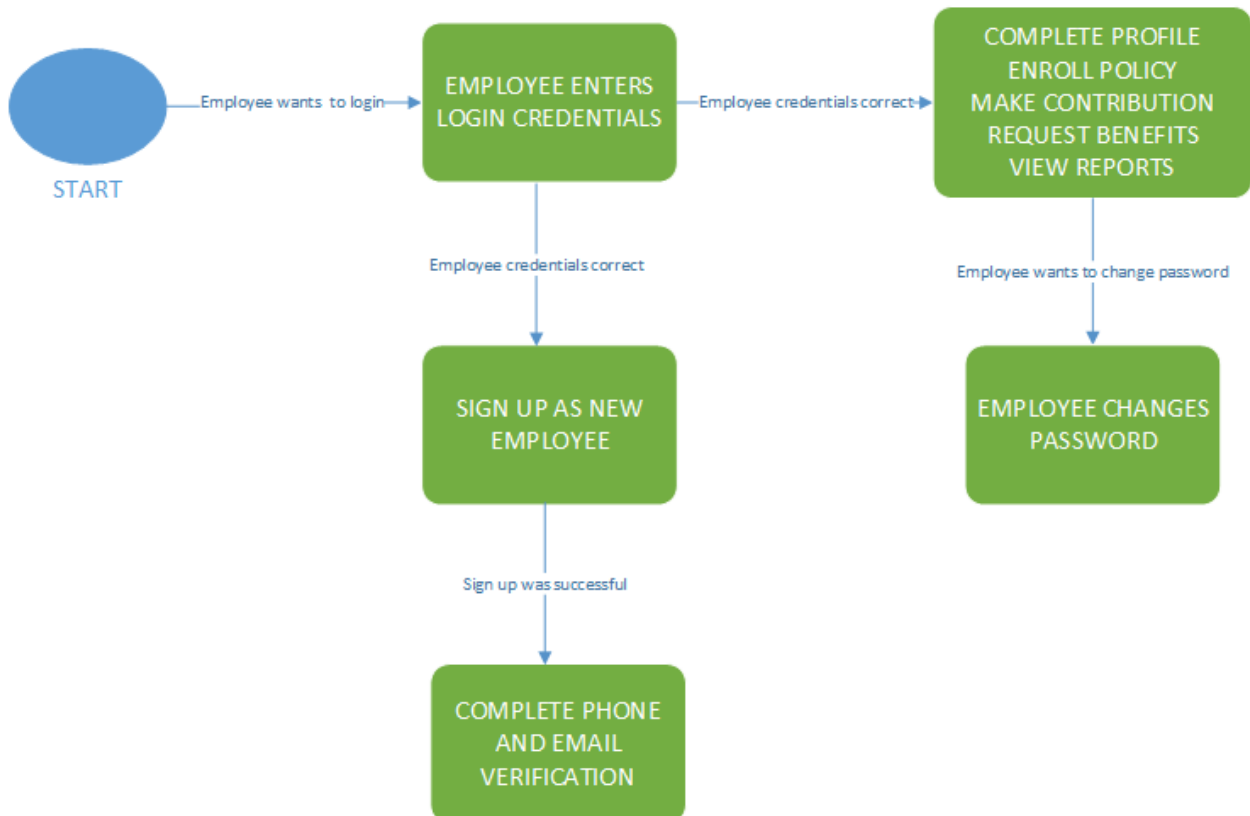


### 5.3 Modelling the proposed system with an activity diagram

#### ACTIVITY DIAGRAM OF ADMIN

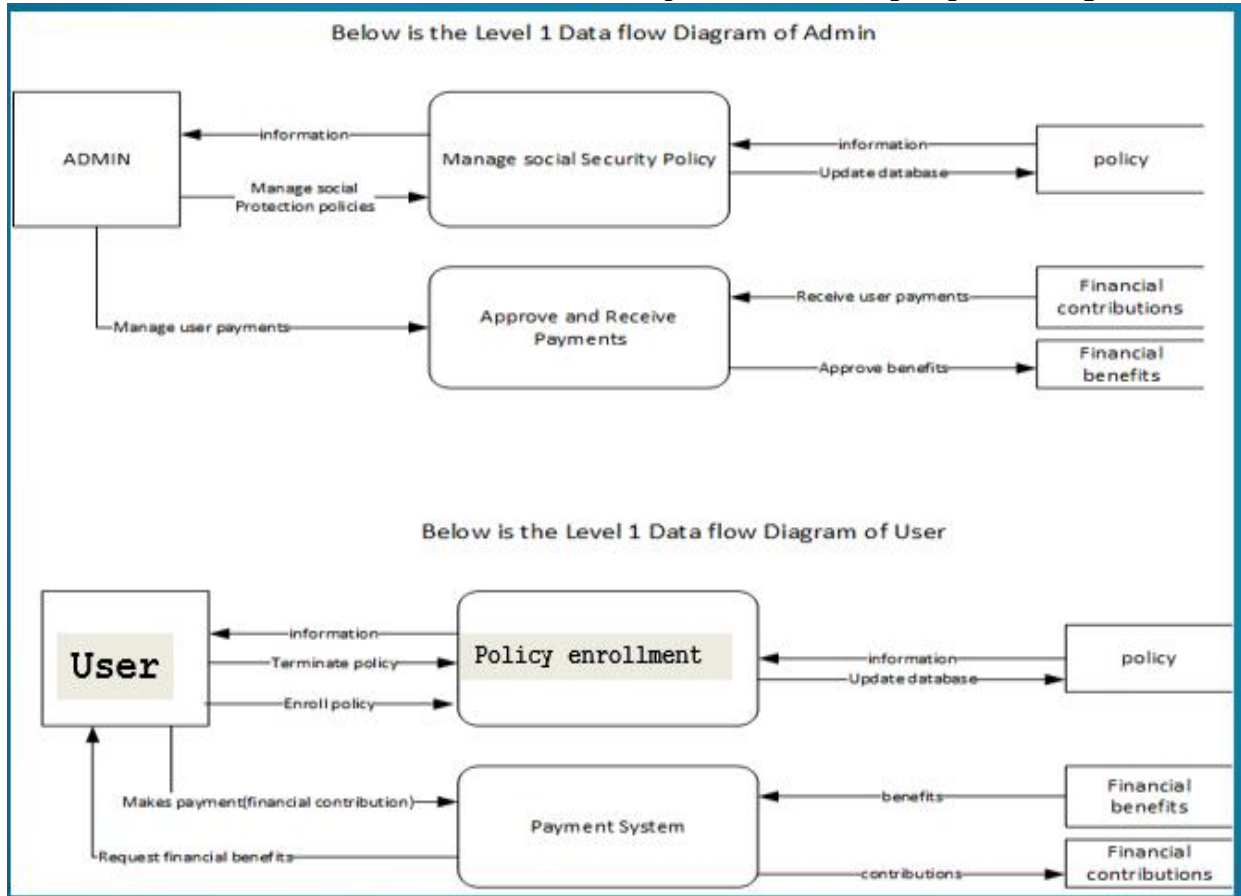


#### ACTIVITY DIAGRAM OF EMPLOYEE



### 5.3.1 Data Flow Diagram

Below is the level 1 data flow diagram of the proposed system

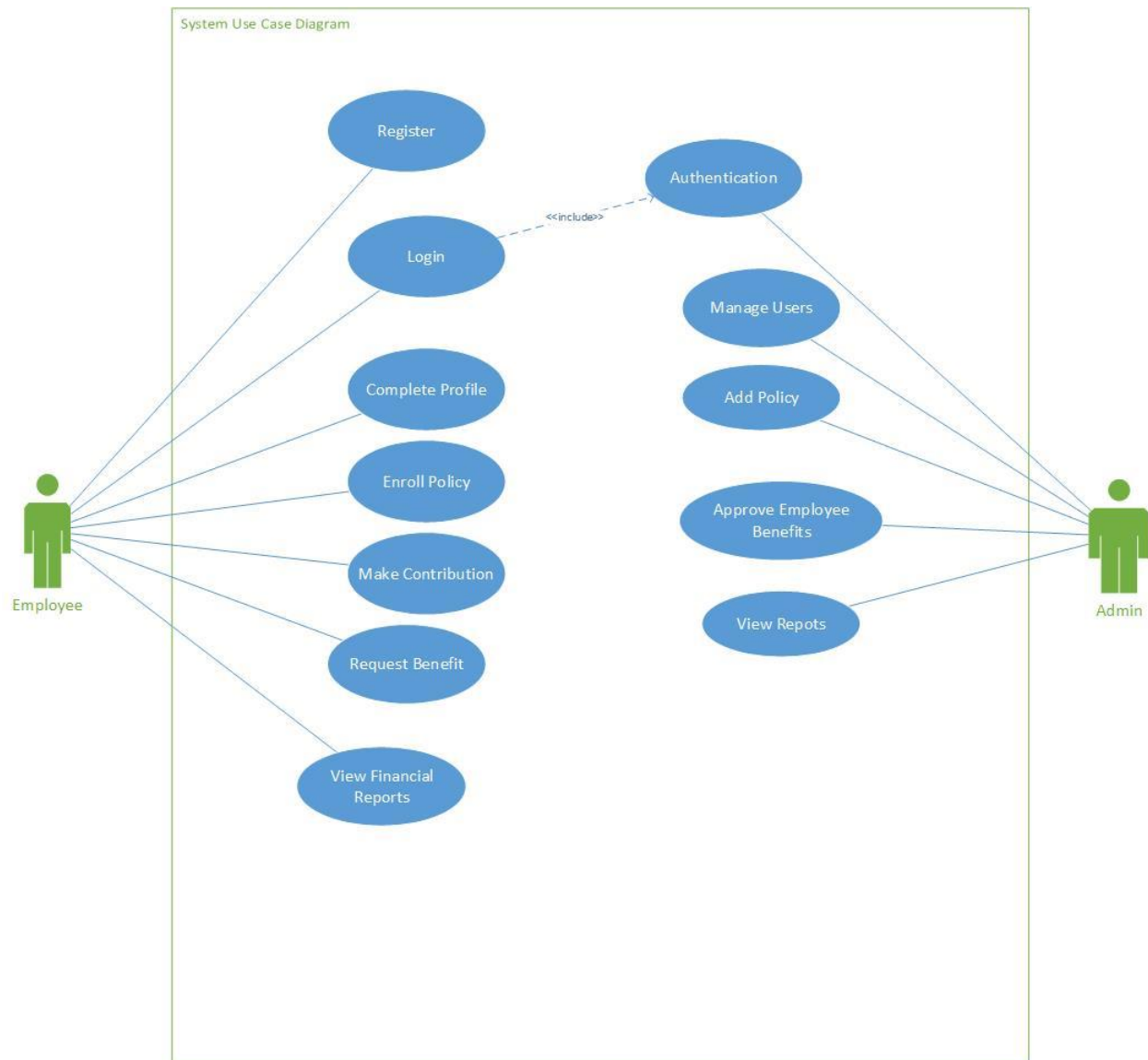


### 5.3.2 Use Case Diagram

Use case diagram is a representation of a user's interaction with the system that shows the relationship between the user and different use cases in which the user is involved. It is used to describe the actual condition or to depict the functionality of the system.

The figure below shows the use case diagram of the proposed system and illustrates the main actors, use cases, and their relationships.

Below is the use case diagram of the proposed system



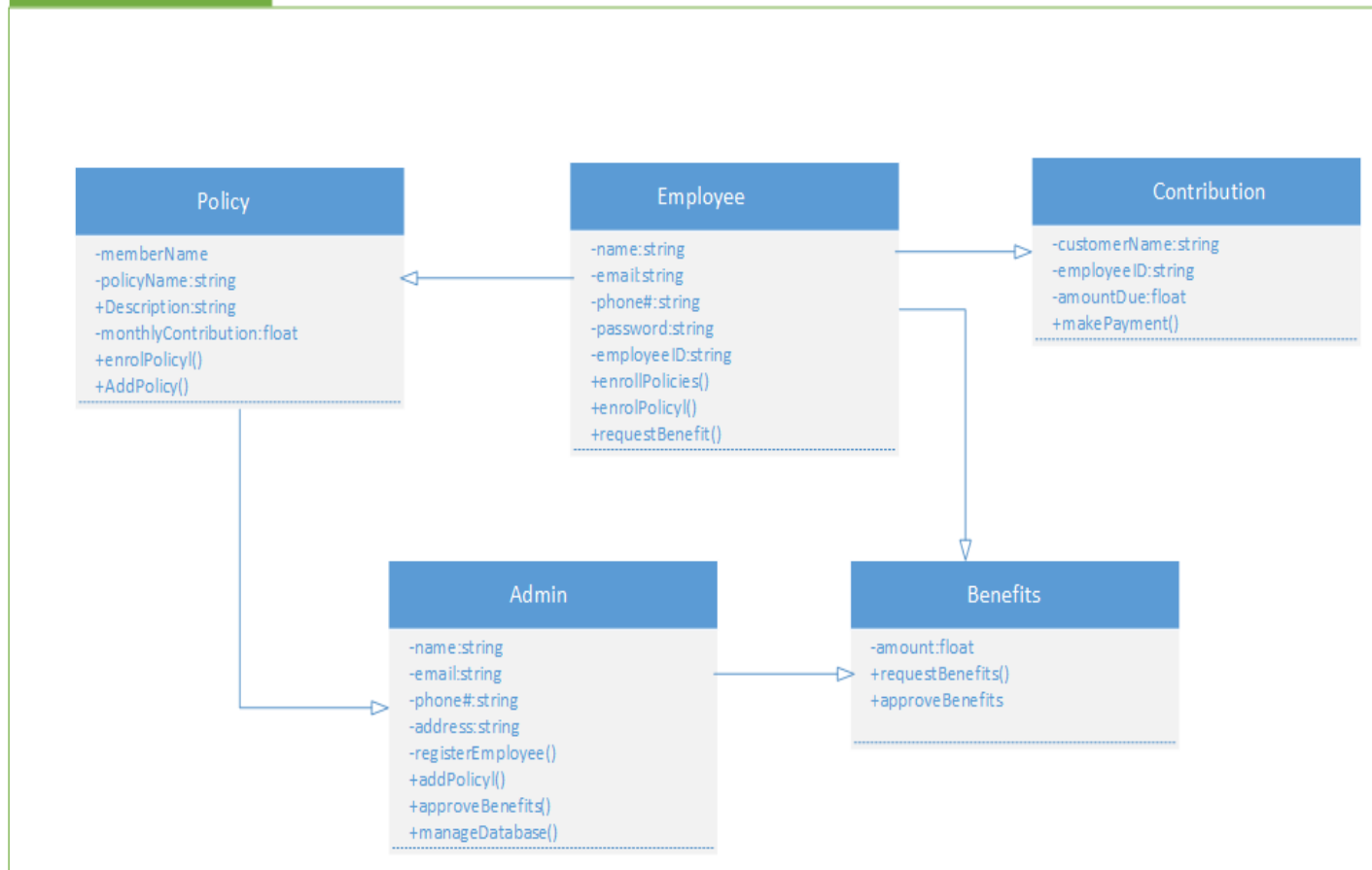
### 5.3.3 Use case actors and their description

ACTORS	DESCRIPTION
ADMINISTRATOR	A system user charged with the responsibility of adding social protection policies and also approving and processing financial benefits to employees
Clients	A system user who makes financial contribution into social protection schemes and also receives financial benefits when the time is due.
USE CASE	DESCRIPTION

LOGIN	This use case allows the user to access entry into the system
MANAGE USERS	This use case allows the system administrator to update and delete users
REGISTRATION	This use case allows the system administrator to add new users to the system
ADD POLICY	This use case allows the system administrator to add multiple social protection schemes or policies to the system
MAKE CONTRIBUTIONS	This use case allows the employee make financial contributions into social protections schemes or policies
ENROLL POLICY	This use case allows the employee to enroll into multiple social protection schemes or policies
REQUEST BENEFITS	This use case allows the employee to request for funds or benefit when the right time is due
APPROVE BENEFITS	This use case allows the system administrator to approve financial payments to employees
LOGOUT	This allows users exit from the system

### 5.3.4 UML Class Diagram

SPMIS UML CLASS DIAGRAM



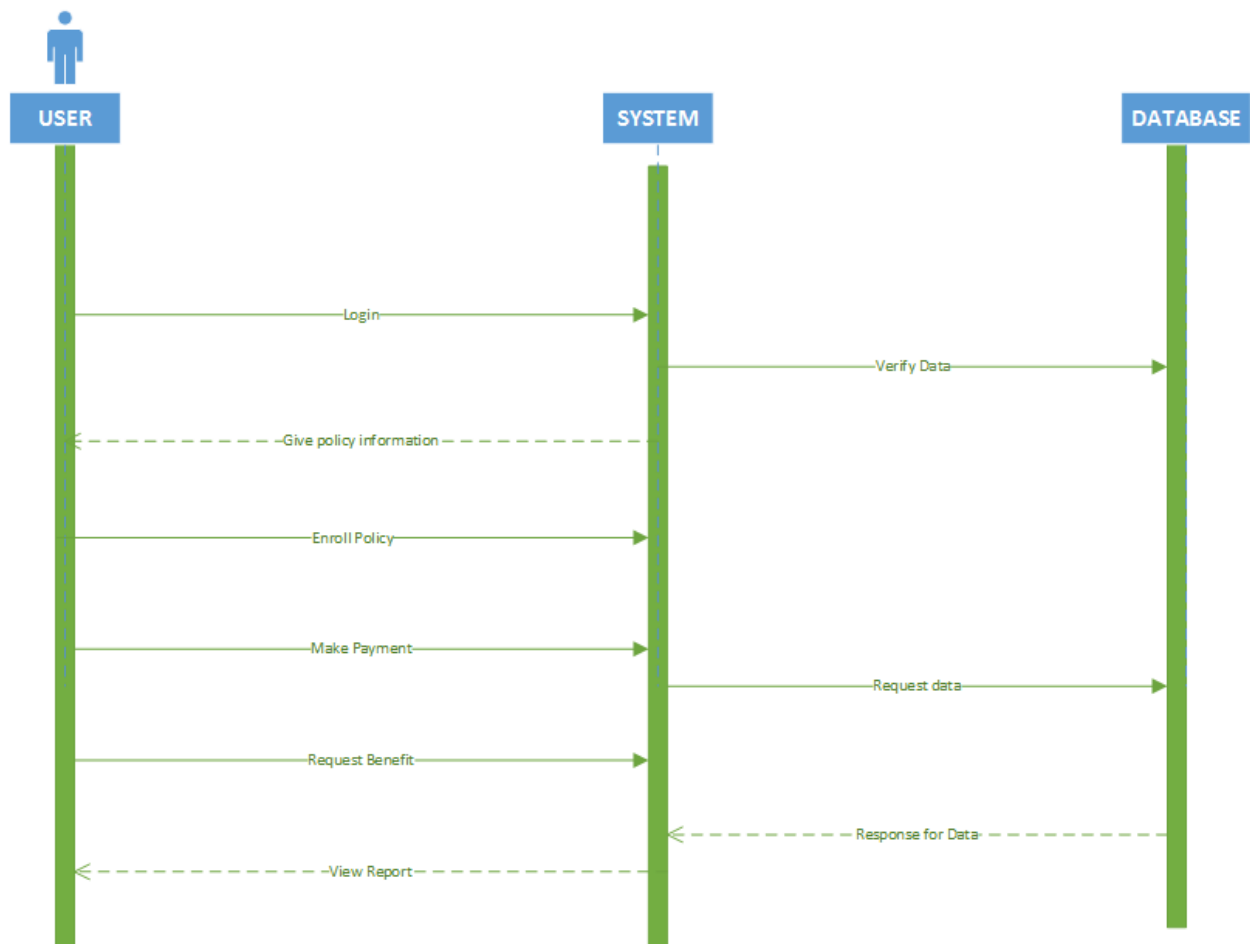
### 5.3.5 Entity Relationship Diagram Of The Proposed System



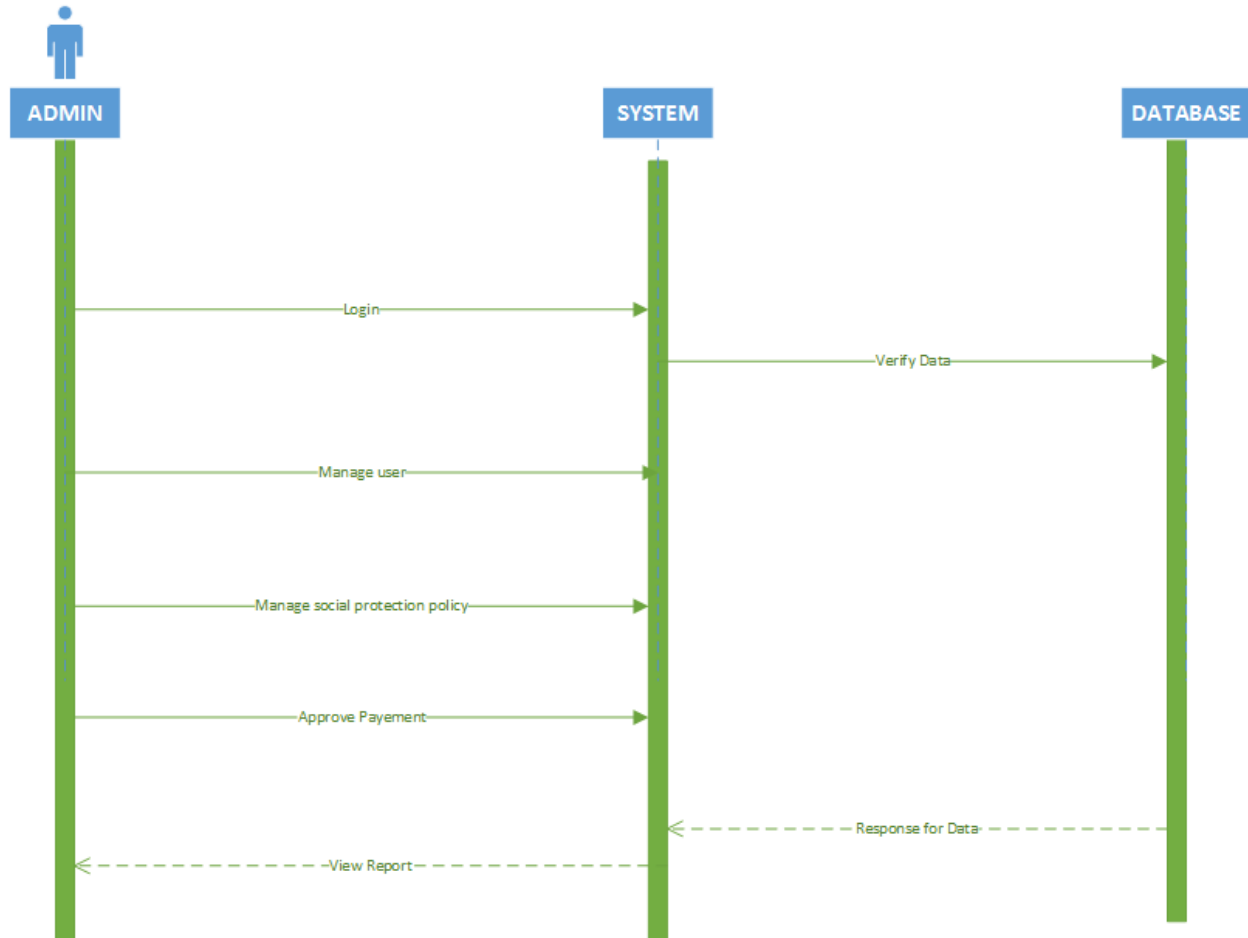


### 5.3.5 Sequence Diagram Of The Proposed System

Below is a sequence diagram of the user's side



Below is a sequence diagram of the Admin's side



### 5.3.7 Database Tables

#### User table

FIELD	TYPE	NULL	DEFAULT	COMMENTS
Id	Bigint (20)	No	None	Id of user
Name	varchar (255)	No	None	Name of user
Email	varchar (255)	No	None	Email of the user
Password	varchar (255)	No	None	Password of user
Roles	varchar (255)	No	None	Determines whether the user is a customer
Phone	varchar (255)	No	None	Phone number of user
Smstoken	Varchar (255)	No	None	SMS activation token

**Admin table**

FIELD	TYPE	NULL	DEFAULT	COMMENTS
Id	Bigint(20)	No	None	Id of admin
Name	varchar(255)	No	None	Name of admin
Email	varchar(255)	No	None	Email of the admin
Password	varchar(255)	No	None	Password of admin
Roles	varchar(255)	No	None	Determines whether the user is an admin
Phone	varchar(255)	No	None	Phone number of admin
Smstoken	Varchar(255)	No	None	SMS activation token

**Customer or Employee Registration table**

FIELD	TYPE	NULL	DEFAULT	COMMENTS
Id	Bigint(20)	No	None	Id of customer
firstname	Varchar(255)	No	None	First name of customer
Middlename	Varchar(255)	Yes	Null	Middle name of customer
Lastname	Varchar(255)	No	None	Last name of customer
Gender	Varchar(255)	No	None	Gender of customer
DOB	Varchar(255)	No	None	Date of birth of customer
Occupation	Varchar(255)	No	None	Occupation of customer
SSFNumber	Varchar(255)	No	None	Social security number of customer
Gross salary	Varchar(255)	No	None	Gross salary of customer
Net Salary	Varchar(255)	No	None	Net Salary of customer
UserID	Varchar(255)	No	None	Id of user
IDnumber	Varchar(255)	No	None	ID card number of the customer

**Enroll\_policymodels table**

FIELD	TYPE	NULL	DEFAULT	COMMENTS
PolicyID	Bigint(20)	No	None	ID of policy
Ssfnumber	Varchar(255)	No	None	Social security number of employee
userID	Bigint(20)	No	None	User id of customer

**Claims or Policy table**

FIELD	TYPE	NULL	DEFAULT	COMMENTS
Id	Bigint(20)	No	None	Id of the policy
Claims	Varchar(255)	No	None	Title of policy
Podes	Varchar(255)	No	None	Description of policy
mded	Double	No	None	Monthly deduction
bcconst	Double	No	None	Total Number of Monthly Contribution a Client Customer should make
age	int(11)	No	None	The age that will determine whether the user qualifies to receive his/her benefit
nat	Varchar(255)	No	None	Nationalities who can enroll in the policy

**Payments table**

FIELD	TYPE	NULL	DEFAULT	COMMENTS
Id	Bigint(20)	No	None	Id of payment
UserID	Int(10)	No	None	Id of user
Ddate	Varchar(255)	No	None	Date of payment
Amt	Double	No	None	Amount contributed
Mode of Payment	Varchar(255)	No	None	The mode of payment
WalletNumber	Varchar(255)	No	None	The wallet number of employee

**Benefits table**

FIELD	TYPE	NULL	DEFAULT	COMMENTS
Id	Bigint(20)	No	None	Id of benefit
UserID	Int(10)	No	None	Id of user
Ddate	Varchar(255)	No	None	Date of benefit payment
Amt	Double	No	None	Amount Received
Mode of Payment	Varchar(255)	No	None	The mode of payment
WalletNumber	Varchar(255)	No	None	The wallet number of employee

## CHAPTER SIX

### SYSTEM IMPLEMENTATION AND TESTING

#### 6.1 Implementation

In the implementation phase, we will look into the minimum requirements that will facilitate the implementation process.

The Social Protection Management Information System (SPMIS) is a web-based management system used to enroll both formal and informal workers to make contributions into social security (protection) schemes or policies and to receive their financial benefits when the time falls due.

#### 6.2 Hardware and Software Requirements

The software requires a computer system to run the application and the machine can run because all required software system are installed.

The minimum requirements for the system to run successfully are as follows:

##### Recommended Operating Systems

- **Windows:** 7 or newer
- **MAC:** OSX v10.7 or higher
- **Linux:** Ubuntu

##### Hardware Requirements

- **Processor:** Minimum 1GHz; Recommended 2GHz or more
- **Ethernet connection:** (LAN) or wireless adapter (Wi-Fi)
- **Hard Drive:** Minimum 32GB; Recommended 64 GB or more
- **Memory (RAM):** Minimum 1GB; Recommended 4GB or above
- **Sound card w/speakers:** Required to listen to screen reader.

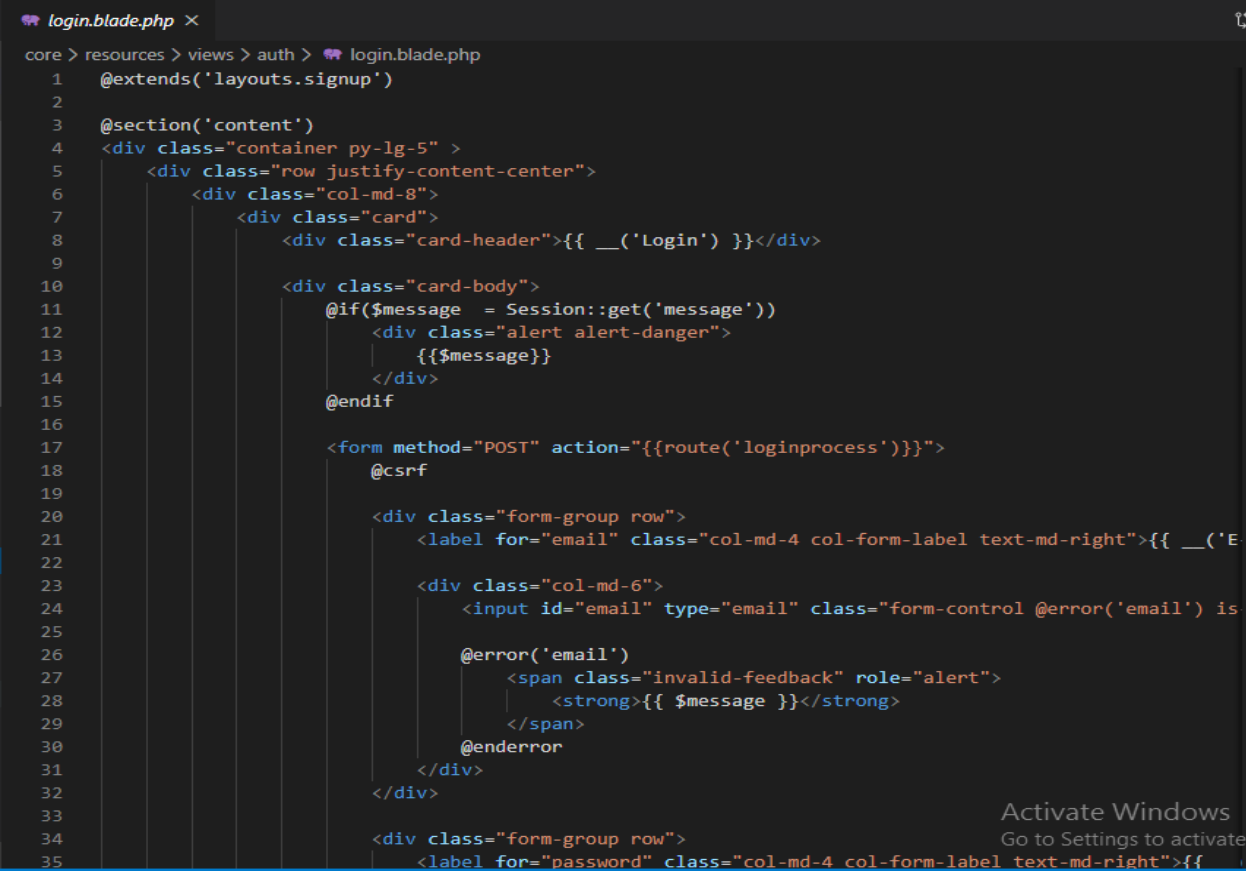
##### Supported Browsers

- Firefox
- Chrome

## 6.2 Code Description of the system

### 6.2.1 LOGIN

The Login portal is to ensure that only legitimate users has access to the system components. Access is granted to the user by entering their email and password. Below is the code for the login code.



```
login.blade.php ×
core > resources > views > auth > login.blade.php
1  @extends('layouts.signup')
2
3  @section('content')
4  <div class="container py-lg-5" >
5      <div class="row justify-content-center">
6          <div class="col-md-8">
7              <div class="card">
8                  <div class="card-header">{{ __('Login') }}</div>
9
10                 <div class="card-body">
11                     @if($message = Session::get('message'))
12                         <div class="alert alert-danger">
13                             {{ $message }}
14                         </div>
15                     @endif
16
17                     <form method="POST" action="{{route('loginprocess')}}">
18                         @csrf
19
20                         <div class="form-group row">
21                             <label for="email" class="col-md-4 col-form-label text-md-right">{{ __('E
22
23                             <div class="col-md-6">
24                                 <input id="email" type="email" class="form-control @error('email') is
25
26                                 @error('email')
27                                     <span class="invalid-feedback" role="alert">
28                                         <strong>{{ $message }}</strong>
29                                     </span>
30                                 @enderror
31                             </div>
32                         </div>
33
34                         <div class="form-group row">
35                             <label for="password" class="col-md-4 col-form-label text-md-right">{{
```

Figure 6.2.1 Login Code

### 6.2.2 SIGNUP

The signup portal is to facilitate the registration of users into the system. The user registers by entering their name, phone number, email and password. Below is the code for the signup code.

```

register.blade.php ×
core > resources > views > auth > register.blade.php
1  @extends('layouts.app')
2
3  @section('content')
4  <div class="container">
5      <div class="row justify-content-center">
6          <div class="col-md-8">
7              <div class="card">
8                  <div class="card-header">{{ __('Register') }}</div>
9
10                 <div class="card-body">
11                     <form method="POST" action="{{ route('register') }}">
12                         @csrf
13
14                         <div class="form-group row">
15                             <label for="name" class="col-md-4 col-form-label text-md-right">{{ __('Name') }}</label>
16
17                             <div class="col-md-6">
18                                 <input id="name" type="text" class="form-control @error('name') is-invalid">
19
20                                 @error('name')
21                                     <span class="invalid-feedback" role="alert">
22                                         <strong>{{ $message }}</strong>
23                                     </span>
24                                 @enderror
25                             </div>
26                         </div>
27
28                         <div class="form-group row">
29                             <label for="email" class="col-md-4 col-form-label text-md-right">{{ __('Email') }}</label>
30
31                             <div class="col-md-6">
32                                 <input id="email" type="email" class="form-control @error('email') is-invalid">
33
34                                 @error('email')
35                                     <span class="invalid-feedback" role="alert">

```

Figure 6.2.2 signup Code

### 6.2.3 Admin Add Policies Page

This is where the Admin adds a social protection policy. Below are the codes for adding policies.

```
addclaims.blade.php X
core > resources > views > admin > addclaims.blade.php
1  @extends("layouts.layout")
2  @section('content')
3      <div class="dashboard-wrapper">
4          <div class="dashboard-ecommerce">
5              <div class="container-fluid dashboard-content">
6                  <div class="row">
7                      <div class="col-5">
8                          <div class="card-header" style="background-color:#0E0C28;color: white">
9                              <h4 style="color: white">Add Policies</h4>
10                         </div>
11                         @if($message = Session::get('message'))
12                             <div class="alert alert-success">
13                                 {{$message}}
14                             </div>
15                         @endif
16                         @if(count($errors)>0)
17                             <div class="alert alert-danger">
18                                 <ul style="list-style: none">
19                                     @foreach($errors->all() as $error)
20                                         <li>{{$error}}</li>
21                                     @endforeach
22                                 </ul>
23                             </div>
24                         @endif
25                         @if(! $flag)
26                             <form action="{{route('claim.store')}}" method="POST">
27                                 @csrf
28                                 @else
29                                     <form action="{{ $action }}" method="post">
30                                         {{csrf_field()}}
31                                         <input type="hidden" name="_method" value="POST">
32                                         @endif
33                                     <div class="form-group">
34                                         <label for="">Policy</label>
35                                     </div>
36                                     @if(! $flag)
37                                         <form action="{{route('claim.store')}}" method="POST">
38                                             @csrf
39                                             @else
40                                                 <form action="{{ $action }}" method="post">
41                                                     {{csrf_field()}}
42                                                     <input type="hidden" name="_method" value="POST">
43                                                     @endif
44                                                     <div class="form-group">
45                                                         <label for="">Policy</label>
46                                                         <input tabindex="1" autofocus type="text" name="claim" class="form-control">
47                                                     </div>
48                                                     <div class="form-group">
49                                                         <label for="">Description</label>
50                                                         <textarea tabindex="2" resize="none" name="podes" class="form-control">
51                                                     </div>
52                                                     <div class="form-group">
53                                                         <label for="">Montly Deduction (%) <a href="javascript:mo">
54                                                         <input tabindex="3" type="text" name="mded" class="form-control">
55                                                     </div>
56                                                     <div class="form-group">
57                                                         <label for="">Benefits Constriant (Montly Contribution) <a href="javascript:mo">
58                                                         <input tabindex="4" type="text" name="bconst" class="form-control">
59                                                     </div>
60                                                     <div class="form-group">
61                                                         <label for="">Benefits Constriant (Age) <a href="javascript:mo">
62                                                         <input tabindex="5" type="text" name="age" class="form-control">
63                                                     </div>
64                                                     <div class="form-group">
65                                                         <label for="">Benefits Constriant (Nationality) <a href="javascript:mo">
66                                                         <select tabindex="6" type="text" name="nat" class="form-control">
67                                                             <option value="">--Select--</option>
68                                                             @isset($cl) @if($cl->nat == "GHANAIAN")selected
69                                                             @endif @endisset
70                                                         </select>
71                                                     </div>
72                             </form>
73                         @endif
74                     </div>
75                 </div>
76             </div>
77         </div>
78     </div>
79 </section>
80 </pre>
```



Figure 6.2.3 Add policy(claims) code

#### 6.2.4 Admin Create users

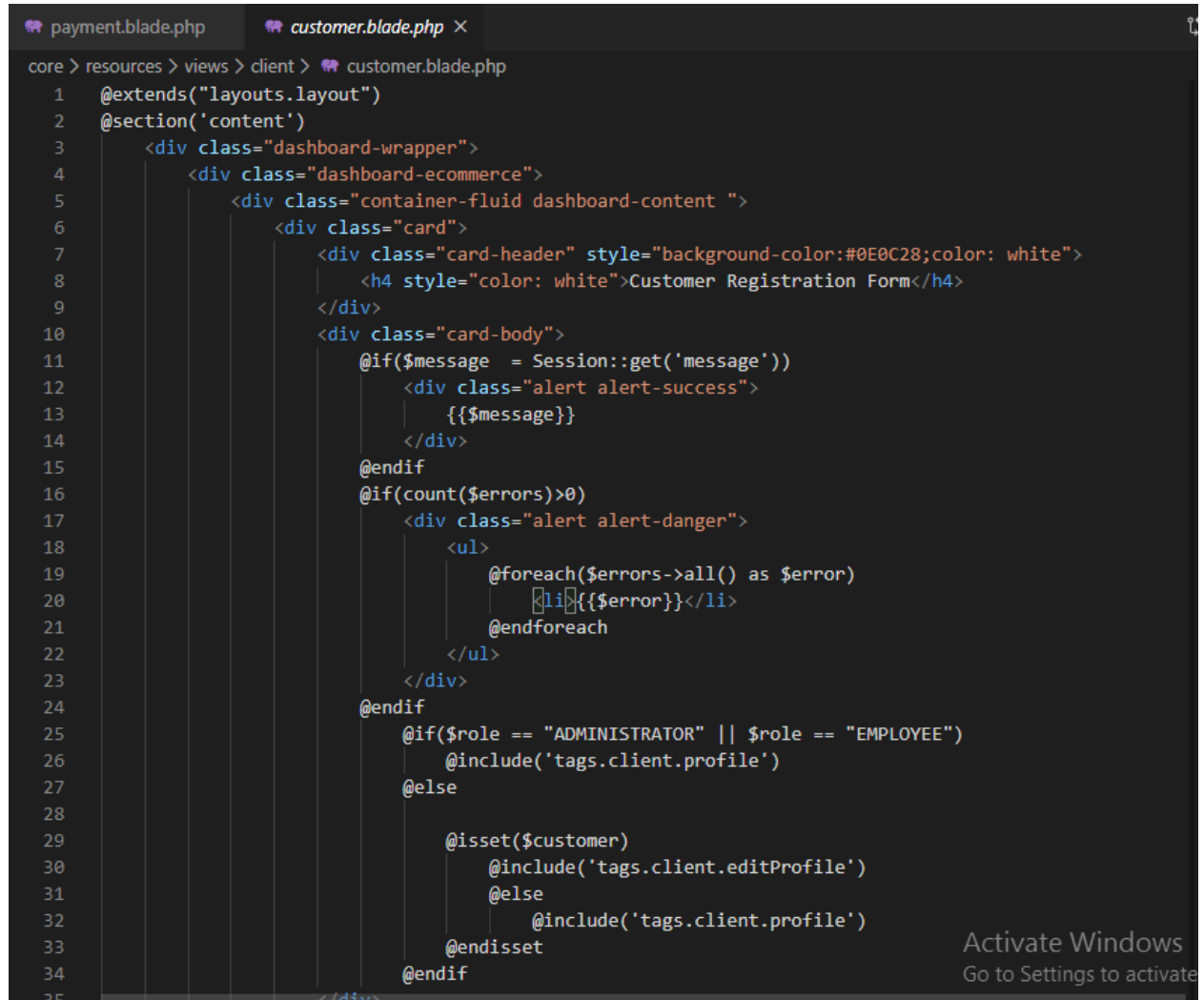
This is where the Admin adds or create users on the system. Below is the code for adding users to the system.

```
registerStaff.blade.php ×
core > resources > views > admin > registerStaff.blade.php
1  @extends("layouts.layout")
2  @section('content')
3      <div class="dashboard-wrapper">
4          <div class="dashboard-ecommerce">
5              <div class="container-fluid dashboard-content ">
6                  <div class="row">
7                      <div class="col-md-8">
8                          <div class="card">
9                              <div class="card-header">{{ __('Register') }}</div>
10                             <div class="card-body">
11                                 @if($message = Session::get('message'))
12                                     <div class="alert alert-success">
13                                         {{ $message }}
14                                     </div>
15                                 @endif
16                                 @if(count($errors)>0)
17                                     <div class="alert alert-danger">
18                                         <ul>
19                                             @foreach($errors->all() as $error)
20                                                 <li>{{ $error }}</li>
21                                             @endforeach
22                                         </ul>
23                                     </div>
24                                 @endif
25                                 <form method="POST" action="{{ route('store-employee-account') }}">
26                                     @csrf
27
28                                     <div class="form-group row">
29                                         <label for="name" class="col-md-4 col-form-label text-md-right">{{ __('Name') }}</label>
30
31                                         <div class="col-md-6">
32                                             <input id="name" type="text" class="form-control @error('name') is-invalid">
33                                             @error('name')
34                                             <span class="invalid-feedback" role="alert">
35
```

Figure 6.2.4 Add users code

### 6.2.5 User fills Customer registration form (User profile)

This is where the user fills his customer registration form in order to enroll in a social protection policy or scheme. Below is the code for filling the registration form.



```
payment.blade.php customer.blade.php x
core > resources > views > client > customer.blade.php
1 @extends("layouts.layout")
2 @section('content')
3     <div class="dashboard-wrapper">
4         <div class="dashboard-ecommerce">
5             <div class="container-fluid dashboard-content ">
6                 <div class="card">
7                     <div class="card-header" style="background-color:#0E0C28;color: white">
8                         <h4 style="color: white">Customer Registration Form</h4>
9                     </div>
10                    <div class="card-body">
11                        @if($message = Session::get('message'))
12                            <div class="alert alert-success">
13                                {{$message}}
14                            </div>
15                        @endif
16                        @if(count($errors)>0)
17                            <div class="alert alert-danger">
18                                <ul>
19                                    @foreach($errors->all() as $error)
20                                        <li>{{$error}}</li>
21                                    @endforeach
22                                </ul>
23                            </div>
24                        @endif
25                        @if($role == "ADMINISTRATOR" || $role == "EMPLOYEE")
26                            @include('tags.client.profile')
27                        @else
28
29                            @isset($customer)
30                                @include('tags.client.editProfile')
31                            @else
32                                @include('tags.client.profile')
33                            @endisset
34                        @endif
35                    </div>
36                </div>
37            </div>
38        </div>
39    </div>
```

Figure 6.2.5 User profile code

### 6.2.6 Enroll Policy

This is where the user enrolls in a social protection policy or scheme. Below is the code for the enroll policy.

```

payment.blade.php  policyenrolment.blade.php X
core > resources > views > client > policyenrolment.blade.php
1  @extends("layouts.layout")
2  @section('content')
3      <div class="dashboard-wrapper">
4          <div class="dashboard-ecommerce">
5              <div class="container-fluid dashboard-content ">
6                  <div class="row">
7                      <div class="col-12" style="border-bottom: 2px solid #cccccc">
8                          <h3>Self Policy Enrolment</h3>
9                      </div>
10                 </div>
11                 <div class="row py-3">
12                     <div class="col-12">
13                         <div class="card">
14                             <div class="card-header">Policy Enrolment</div>
15                             <div class="card-body">
16                                 @if($message = Session::get('message'))
17                                     <div class="alert alert-success">
18                                         {{$message}}
19                                     </div>
20                                 @endif
21                                 @if($sect > 0)
22                                     <form id="unenrollme" action="{{route('policy-unenrol')}}" me
23                                 @else
24                                     <form action="{{route('store-policy')}}" method="POST">
25                                         @endif
26                                 @csrf
27                                     <input value="{{$customer->ssfnumber}}" type="hidden" class="
28                                     <input id="cname" value="{{$customer->lastname.' '.$customer->
29                                     <div class="form-group">
30                                     <table class="table">
31                                     <thead>
32                                     <tr>

```

Figure 6.2.5 Enroll policy code

### 6.2.6 Make Contribution

This is where the user makes financial contributions to social protection policies or schemes. Below is the code for making financial contributions.

```

payment.blade.php x policyenrolment.blade.php
core > resources > views > client > payment.blade.php
1  @extends("layouts.layout")
2  @section('content')
3      <div class="dashboard-wrapper">
4          <div class="dashboard-ecommerce">
5              <div class="container-fluid dashboard-content ">
6                  <div class="row">
7                      <div class="col-12" style="border-bottom: 2px solid #cccccc">
8                          <h3>Payment</h3>
9                      </div>
10                 </div>
11                 <div class="row py-3">
12                     <div class="col-6">
13                         <div class="card">
14                             <div class="card-header">Payment Notice</div>
15                             <div class="card-body">
16                                 <form action="">
17                                     <div class="form-group">
18                                         <label for="SPMIS Number">SPMIS Number</label>
19                                         <input value="{{ $customer->ssfnumber }}" type="text" class="form-control">
20                                     </div>
21                                     <div class="form-group">
22                                         <label for="SPMIS Number">Customer Name</label>
23                                         <input value="{{ $customer->lastname.' '.$customer->firstname }}" type="text" class="form-control">
24                                     </div>
25                                     <div class="form-group">
26                                         <label for="SPMIS Number">Amount Due
27                                         @foreach($policy as $po)
28                                             <span>({{ $po->policy->claims }})</span>
29                                         @endforeach
30                                         </label>
31                                         <input value="{{ number_format($ded,2) }}" type="text" class="form-control">
32                                     </div>
33                                     <div class="form-group">
34                                         <label for="SPMIS Number">Mode of Payment</label>
35                                     </div>

```

Figure 6.2.6 Financial contribution code

### 6.2.7 View Financial reports

This is where the user views and prints his or her financial reports and statements. Below is the code for viewing reports

```

report.blade.php ×
core > resources > views > report.blade.php
1  @extends("layouts.layout")
2  @section('content')
3      <div class="dashboard-wrapper">
4          <div class="dashboard-ecommerce">
5              <div class="container-fluid dashboard-content ">
6                  <div class="row" style="border-bottom: 2px solid #cccccc">
7                      <div class="col-4" >
8                          <h3>Report(s)</h3>
9                      </div>
10                     <div class="col-8 text-right">
11                         @if(isset($rpt))
12                             <a href="javascript:printReport()" ><i class="fa fa-print"></i> Print</a>
13                         @endif
14                     </div>
15                 </div>
16                 <div class="row py-1">
17                     <div class="col-3">
18                         <form action="{{route('report-show')}}" method="GET">
19                             <div class="card">
20                                 <div class="card-header">
21                                     Report Options
22                                 </div>
23                                 <div class="card-body">
24                                     <div class="form-group">
25                                         <label for="">Report Type</label>
26                                         <select name="rptOption" class="form-control" id="rptOption">
27                                             {{--<option value="">-Select-</option>--}}
28                                             <option value="">Contribution</option>
29                                             {{--<option value="">Benefit Report</option>--}}
30                                         </select>
31                                     </div>
32                                     {{--<div style="display: none">--}}
33                                     <div class="form-group">
34                                         <label for="">Beginning Date</label>
35                                         <input type="date" id="bgdate" name="bgdate" class="form-conti

```

Figure 6.2.7 Financial reports code

### 6.3 Testing the System

System testing should be performed to know if the system is working correctly or not. This checks if the intended purpose of the program has been achieved.

### 6.3.1 Description of the Testing Strategy

Software testing is an activity to check whether the actual results match the expected results and to ensure that the software system is defect free. It involves the execution of a software component or system component to evaluate one or more properties of interest. Software testing also helps to identify errors, gaps, or missing requirements in contrary to actual requirements. It can be done either manually or using automated tools. Some prefer saying software testing as a White Box and Black Box Testing.

Testing is important because software bugs could be expensive or even dangerous. Software bugs can potentially cause monetary and human loss, and history is full of examples.

Testing is needed at every given stage of a software's life cycle.

Testing done at the various levels during the development of software is done for different purposes/objectives.

- **Unit Testing:** Unit testing is a means of testing at the lowest level. This is usually done to test for the basic units of the software application. A unit is a basic module, i.e. the least possible set of lines of code which can be tested. This is done for continual testing and revision. Developers usually write these codes until all units' tests are passed.
- **System Testing:** This is the stage where the whole system or application is tested as a whole. At the level, system quality is tested for and also functional, and non-functional attributes of the system are checked for (i.e. reliability, maintainability, security and requirement specifications)

Also, there are other forms of testing strategies which can be employed for this test case.

- **Black-Box Testing:** Is a method of testing that test the functionality and requirement of an application as opposed to its internal structures or working.
- **White-Box Testing:** Is an internal perspective of the system as well as the programming skills.

#### 6.3.2 Statement of Test Case for the Proposed System

TEST CASE	EXPECTED RESULTS	TESTING STRATEGY	RESULTS
User login	Deny access to invalid username and password and allow access to valid username and password	UNIT TESTING	Successful
Logout/Exit/Close		UNIT TESTING	Successful
Add Policies	Adds social protection policies and schemes	UNIT TESTING	successful
Enroll Policies	Enroll a user into social protection policies	UNIT TESTING	Successful

Performance	The number of times it takes for the system to complete processes	SYSTEM TESTING	successful
User Registration	Deny entry with empty name, phone number and email address	UNIT TESTING	Successful
Database data types	The right and appropriate data types are needed	SYSTEM TESTING	Successful
Database connection	All opened connection should be closed	SYSTEM TESTING	successful
Manage Users	The Admin manages all the Users within the system	UNIT TESTING	Successful
Reports	Users should have access to the report of their financial	UNIT TESTING	Successful



	statements on contribution and benefits		
--	---	--	--

### 6.3.1 Screen Reader API Integration (Disability friendly)

The main objective of integrating a screen reader API is to build systems that are disability friendly. The system was built in order to enable the blind to better interact with web applications. It's a social protection management information system with a Screen Reader JavaScript API being integrated.

#### WEBREADER API INTEGRATION

WebReader is a free and open source JavaScript library developed for modern web browsers. Built using emerging functionality, WebReader can be integrated into any web page. Underneath, WebReader relies on the Web Speech API to offer speech recognition and speech synthesis functionality. It integrates a subset of the features provided by classic screen readers.

A major advantage of WebReader is that it does not require any additional software to be installed on a user's machine, apart from a supported web browser.

It is built with the hope to help users with disabilities better interact with web pages. The intention is not to replace screen readers, but with basic features embedded into a web page, any user with an internet connection and a headset will be able to browse a website. It frees people from constantly looking at the screen which is useful for anyone which is on the move.

Below is the code for the screen reader

```
JS bundle.js x
assets > screen-reader > JS bundle.js > ...

125 });
126 System.register("speak/LabelledSpeaker", ["speak/SpeakerInterface"], function (exports_7, context_7)
127 var __moduleName = context_7 && context_7.id;
128 var SpeakerInterface_1, LabelledSpeaker;
129 return {
130   setters: [
131     function (SpeakerInterface_1_1) {
132       SpeakerInterface_1 = SpeakerInterface_1_1;
133     }
134   ],
135   execute: function () {
136     LabelledSpeaker = class LabelledSpeaker extends SpeakerInterface_1.AbstractSpeaker {
137       constructor(elementToTextMediator, getterByIds) {
138         super();
139         this.elementToTextMediator = elementToTextMediator;
140         this.getterByIds = getterByIds;
141       }
142       getText(node, config) {
143         let text = this.getRefText(node, config);
144         if (!text) {
145           text = node.getAttribute("aria-label");
146           text = text ? `${text}.` : ``;
147         }
148         return text;
149       }
150       getRefText(node, config) {
151         let text = "";
152         if (!config.isRef) {
153           let refLabel = (node.getAttribute("aria-labelledby") || node.getAttribute("aria-label"));
154           if (refLabel) {
155             let ids = refLabel.split(/\s+/);
156             let elements = this.getterByIds.getElements(ids);
157             let newConfig = Object.assign({}, config, { isRef: true });
158             // do not check ref for further elements, to avoid infinite label checks
159             for (let element of elements) {

```

## CHAPTER SEVEN

### USER MANUAL OF SYSTEM ADMINISTRATOR AND USER

The system is a web-based social protection management information system that registers potential members online to make contributions into social security schemes(policies) and to provide social assistance to beneficiaries of the scheme via a secure web portal.

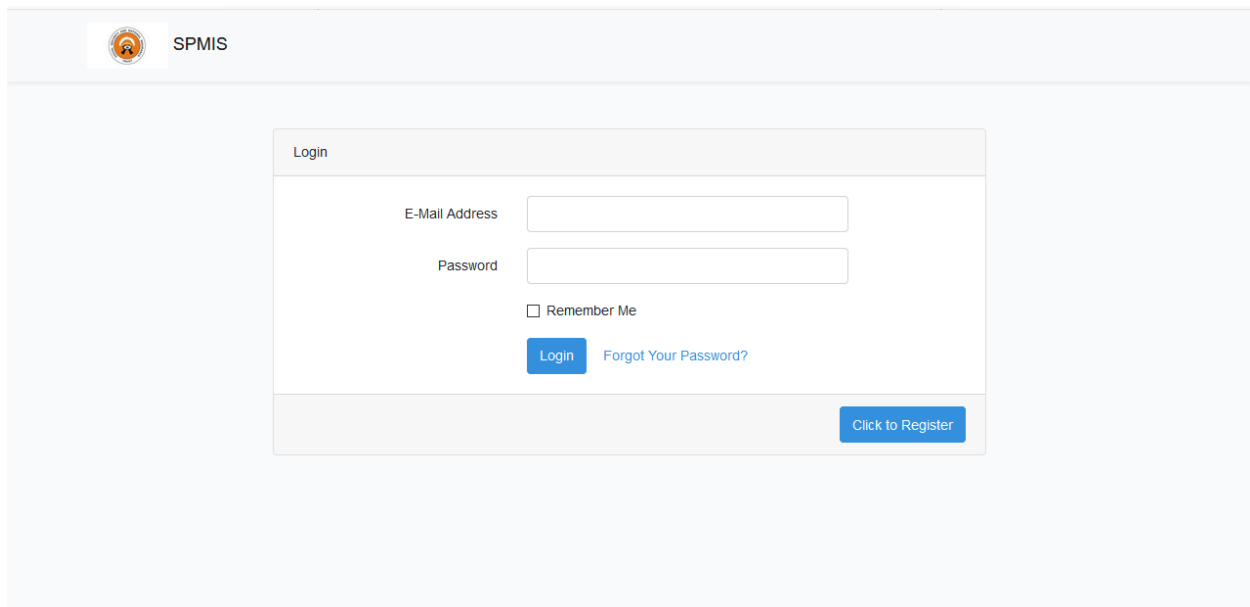
#### 7.1 System Home Page

This shows the homepage of the system



### 7.1.1 System Login Interface

This shows the login portal where the user logs in as an Admin or User before they can have access to the program. This portal can be viewed by both the Admin and the User.



The screenshot displays the SPMIS System Login Interface. At the top left, there is a logo and the text "SPMIS". The main content area features a "Login" form. The form includes two input fields: "E-Mail Address" and "Password". Below these fields is a checkbox labeled "Remember Me". A blue "Login" button is positioned below the "Password" field, and a link "Forgot Your Password?" is located to its right. At the bottom right of the form, there is a blue button labeled "Click to Register".

### 7.1.2 Admin Portal

This view the administrators and their privileges they can perform which includes Adding Policies, Creating users and Viewing reports.

SPMIS

trinarycompanylimited@gmail.com

MENU

Dashboard

Add Policy

Create Users

Report

View Users

Social Protection Management Information System Dashboard Overview

Dashboard > Dashboah for SPMIS

Client Count

0

SMS Credit Balance

0

Claim Requisition Count

0

Daily Total Payment

GH¢0.0

Client Benefit Request list

#	SSF Number	Name	Policy	Date Enrolled	# Month Contibuted
---	------------	------	--------	---------------	--------------------

Copyright © 2020 SPMIS. All rights reserved. Dashboard by Spmis.

### 7.1.3 Add Policy

This allows the Admin to add social protection policies to the system

trinarycompanylimited@gmail.com

MENU

Dashboard

Add Policy

Create Users

Report

View Users

Add Policies

Policy

Description

Montly Deduction (%)

Benefits Constriant (Montly Contribution)

Benefits Constriant (Age)

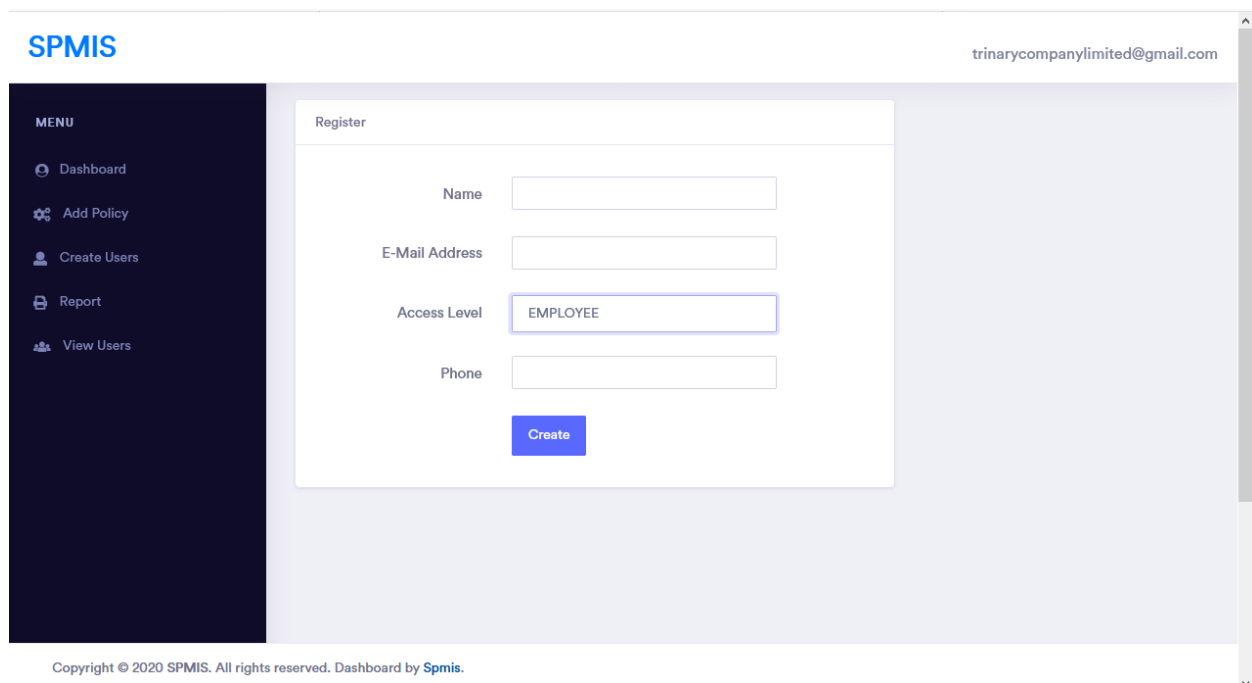
Policies

SN	Policy	Description	Deduction	ACTION
1	SUPERANNUATION PENSION/ OLD AGE PENSION	To qualify for old age pension, the member must be at least 60 years and must have contributed a minimum of 180 months (15 yrs.) under act 766 and 240 months (20 yrs.) under PNDCL 247. The member who is	2	

Copyright © 2022 SPMIS. All rights reserved. Dashboard by Spmis. Designed by Solomon Yaw Adeklo

#### 7.1.4 Add New User

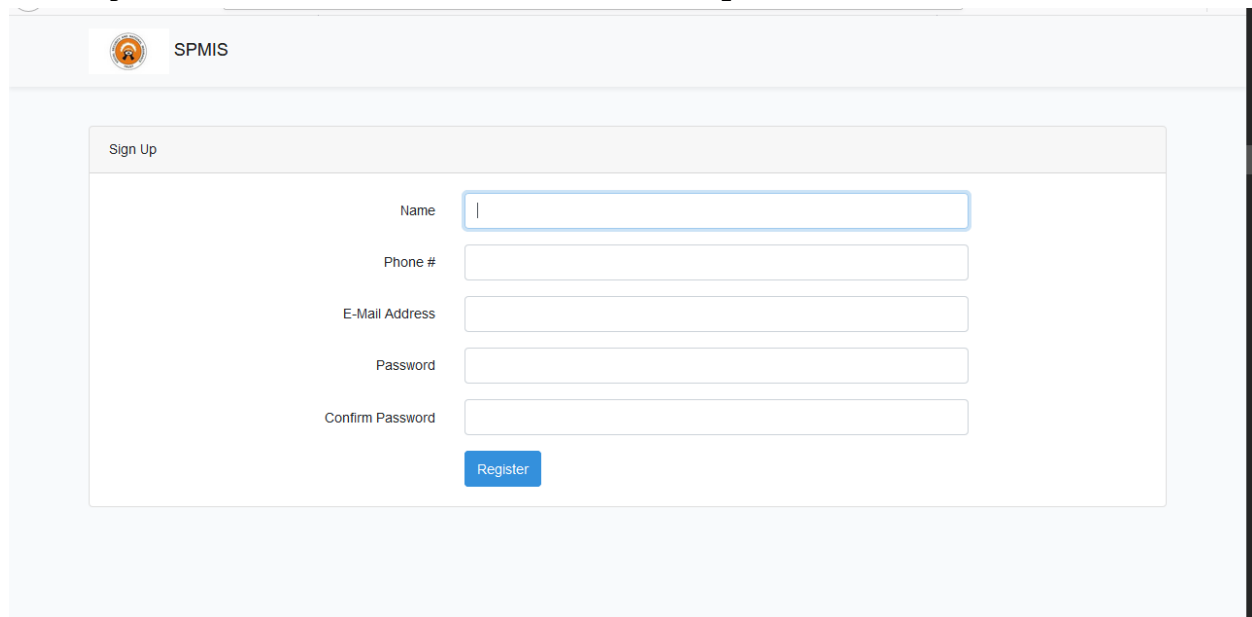
This allows the Admin to add new Users to the system and manage their accounts



The screenshot shows the SPMIS web application interface. On the left is a dark blue sidebar with a 'MENU' section containing links: Dashboard, Add Policy, Create Users, Report, and View Users. The main content area is titled 'Register' and contains a form with the following fields: Name, E-Mail Address, Access Level (a dropdown menu currently showing 'EMPLOYEE'), and Phone. A blue 'Create' button is positioned below the Phone field. The top right of the page displays the email address 'trinarycompanylimited@gmail.com'. At the bottom, a copyright notice reads: 'Copyright © 2020 SPMIS. All rights reserved. Dashboard by Spmis.'

#### 7.1.5 User Signup

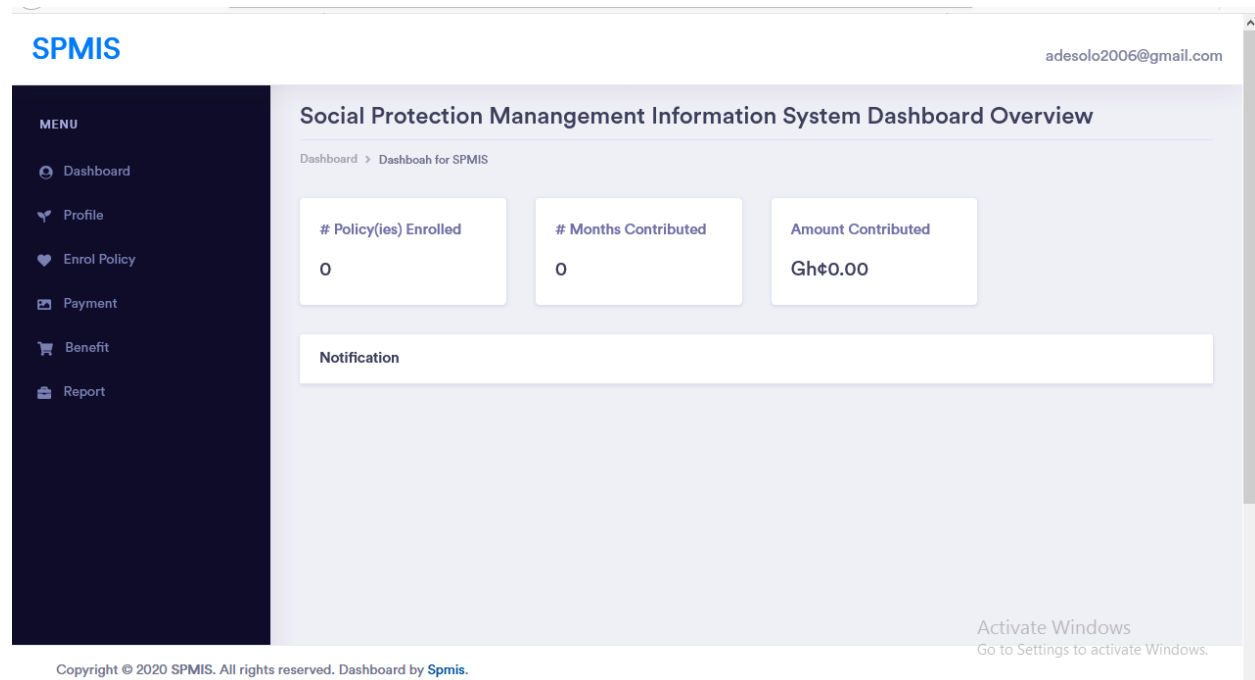
This shows the signup portal where the user signs-up in order to be registered to have access to the system.



The screenshot shows the SPMIS web application's 'Sign Up' portal. The header includes the SPMIS logo and the text 'SPMIS'. The main content area is titled 'Sign Up' and contains a form with the following fields: Name, Phone #, E-Mail Address, Password, and Confirm Password. A blue 'Register' button is located at the bottom of the form.

### 7.1.6 User(Employee or Customer) Portal

This views the users and their privileges they can perform which includes completing their profile, enrolling policies, making their contribution, requesting for benefits and viewing reports.



### 7.1.7 Customer Profile

This view allows the user to complete a registration form so that they can be allowed by the system to enroll in social security policies.

SPMIS

adesolo2006@gmail.com

MENU

Dashboard

Profile

Enrol Policy

Payment

Benefit

Report

Customer Registration Form

1. Name

Last Name

First Name

Middle Name

Solomon

Adeklo

Yaw

2. Gender

☒ Male

☐ Female

3. Date / Place of Birth

Birth Date

Birth City

05 / 04 / 1989

Tema

3. Address

Activate Windows

Go to Settings to activate Windows.

Copyright © 2020 SPMIS. All rights reserved. Dashboard by Spmis.

## 7.1.8 Enroll Policy

This view allows the user to enroll in social security policies.

← → ↺ 🏠

🔒 ⓘ 127.0.0.1/client/policy-enrol

⋮ 📄 ⌂ 🌐 🏠 ☰

SPMIS

adesolo2006@gmail.com

MENU

Dashboard

Profile

Enrol Policy

Payment

Benefit

Report

Self Policy Enrolment

Policy Enrolment

SN	Policy(ies)	Description	Deduction	
1	SUPERANNUATION PENSION/ OLD AGE PENSION	To qualify for old age pension, the member must be at least 60 years and must have contributed a minimum of 180 months (15 yrs.) under act 766 and 240 months (20 yrs.) under PNDCL 247.	18	Terminate Policy
2	INVALIDITY PENSION	To qualify for invalidity pension, the member must have contributed for 12 months in aggregate within the last 36 months preceding the incidence of the invalidity. He/ She must have also been certified by a Medical Board as being incapable of any normal gainful employment due to a permanent physical or mental disability. The invalidity pension is paid monthly to such a person who has been confirmed and certified incapable of earning an income. INVALIDITY PENSION	18	Apply

Activate Windows

Go to Settings to activate Windows.

Copyright © 2020 SPMIS. All rights reserved. Dashboard by Spmis.



### 7.1.9 Make Financial Contribution

This view allows the user to make financial contributions into social security policies.

The screenshot displays the SPMIS (Social Pension Management Information System) web application. The browser address bar shows the URL `127.0.0.1/client/payment`. The user is logged in as `adesolo2006@gmail.com`. The interface features a dark blue sidebar menu with the following options: Dashboard, Profile, Enrol Policy, Payment, Benefit, and Report. The main content area is titled "Payment" and is divided into two sections. The left section, "Payment Notice", displays the following information: SPMIS Number (SM19890405541), Customer Name (Solomon Adekio Yaw), Amount Due (3.60), and Mode of Payment (Mobile Money). A green "Confirm" button is located at the bottom of this section. The right section, "Payment (Using Momo)", contains input fields for Network Provider (MTN), Wallet Number (0540834343), and Amount (3.60). Below these fields are two buttons: "back" and "Pay Now". At the bottom of the page, there is a copyright notice: "Copyright © 2020 SPMIS. All rights reserved. Dashboard by Spmis." and a Windows activation watermark.

SPMIS

adesolo2006@gmail.com

MENU

- Dashboard
- Profile
- Enrol Policy
- Payment
- Benefit
- Report

Payment

Payment Notice

SPMIS Number

SM19890405541

Customer Name

Solomon Adekio Yaw

Amount Due (SUPERANNUATION PENSION/ OLD AGE PENSION)

3.60

Mode of Payment

Mobile Money

Confirm

Payment (Using Momo)

Network Provider

MTN

Wallet Number

0540834343

Amount

3.60

back Pay Now

Activate Windows  
Go to Settings to activate Windows.

Copyright © 2020 SPMIS. All rights reserved. Dashboard by Spmis.

SPMIS

adesolo2006@gmail.com

MENU

Dashboard

Profile

Enrol Policy

Payment

Benefit

Report

Payment

Payment Notice

SPMIS Number

SM19890405541

Customer Name

Solomon Adeklo Yaw

Amount Due (SUPERANNUATION PENSION/ OLD AGE PENSION)

3.60

Mode of Payment

Mobile Money

Confirm

Payment (Using Momo)

Please Wait !!! do not Refresh the Page

MTN MOMO

- Dial \*170#
- Choose option: 6) Wallet
- Choose option: 3) My Approvals
- Enter your MOMO pin to retrieve your pending approval list
- Choose a pending transaction
- Choose option 1 to approve
- Tap button to continue
- You will receive an SMS confirmation message for a successful Payment.

Copyright © 2020 SPMIS. All rights reserved. Dashboard by Spmis.

### 7.1.10 Request Benefits

This view allows the user to request financial benefits from due to their financial previous obligations to social security policies.

sadeklo@st.vvu.edu.gh

MENU

Dashboard

Profile

Enrol Policy

Payment

Benefit

Report

Benefit

# Monthly Contribution

2

Amount Contributed

GH¢ 12.18

Monthly Payment

Click to Initiate Monthly payment

GH¢ 8.40

Copyright © 2022 SPMIS. All rights reserved. Dashboard by Spmis. Designed by Solomon Yaw Adeklo

### 7.1.11 View Report

This view allows the user to generate financial statement of benefits and contributions

**SPMIS** adesolo2006@gmail.com

**Report(s)** Print

**Report Options**

**Report Type**  
-Select-

**Beginning Date**  
mm / dd / yyyy

**Ending Date**  
mm / dd / yyyy

**Generate**

**Social Protection Management Information System (SPMIS)**  
Pacomac street 12, New Sebrepor

Name: Solomon AdekioYaw  
SSF #: SM19890405541

Policy: SUPERANNUATION  
PENSION/ OLD AGE PENSION  
Benefit

From: 2020-06-01 To: 2020-06-27

SN	Date	Month	Amount
End of Statement			

## CHAPTER EIGHT

### CONCLUSION AND RECOMMENDATION

#### 8.0 Conclusion

This system, on the whole, covered what was stated in the specific objects and the expected results.

The objective of this project is to develop an efficient system which will reduce the repetitive cycle whereby beneficiaries

need to fill out forms in order to receive their retirement benefits.

The project is to facilitate the enrollment of both formal and informal workers into harmonized social security schemes so they can retrieve retirement benefits when the time falls due.

The development of this involved the review of the similar existing system with the same purpose.

The first chapter of this document was on the project proposal, outlining the basis for embarking on this project which included the specific objectives.

The second chapter focused on the literature review of existing systems on the proposed project topic.

Chapter three focused on the crystallisation of the problem and vivid analysis of the existing system

Chapter four gave a partial description of the proposed system including its functional and non- functional requirement.

Chapter five gave a more detailed description of the system with diagrams depicting use case, flow charts, Entity Relationship, the UML Class and data flow diagrams.

Chapter six showed the hardware and software requirement for system implementation as well as testing strategies and statement of the test case employed to evaluate the system.

Chapter seven provides the user manual that serves as a guide for proper system use.

## **8.1 Recommendations**

Future development of the system should consider the following:

- The system should integrate a biometric registration system so that potential members of SNNIT do not have to go to the SSNIT office to provide their biometric data for verification though that will require the user purchase a biometric device
- My recommendation is that in future, the SPMIS system can integrate blockchain technology(smart contracts) whereby potential members of the program can make monetary contributions using cryptocurrencies like bitcoin or ethereum since technology is veering to Web 3.0 in the 21<sup>st</sup> Century

## References

David A. Robalino, Laura Rawlings (2012) Building Social Protection and Labor Systems

Baldeon C and Arribas-Banos M, *Strengthening the MIS in Social Protection Programs: A Toolkit*, Washington D.C., World Bank, 2007

Social Security and National Trust, <https://www.ssnit.org.gh>  
Retrieved from <https://www.ssnit.org.gh/about-us/> and  
<https://www.ssnit.org.gh/about-us/benefits/>

Joyce Abebrese, Social Protection in Ghana, 2011, retrieved September, 9, 2011 from <http://library.fes.de/pdf-files/bueros/ghana/10497.pdf>

Bishkek, Developing Management Information Systems for Social Protection, Deutsche Gesellschaft für , 6 November 2012

Sam WE, 'Rapid Assessment of Leaps Database: Towards the Design of a Management Information System', unpublished work, UNICEF Ghana, 2011

<http://insurancesystems.com/products/isi-enterprise>

[accessed May 16, 2018]

<http://insuraccess.com/index.php/features/> [accessed May 16, 2018]

<https://www.buckhill.co.uk/products/c2ms-insurance-software>

[accessed June 1, 2018]

<https://www.softwareadvice.com/insurance/partner-xe-profile/>

[accessed June 23, 2018]

<https://www.mutualexpert.com/> [accessed July 5, 2018]