Problem Statement

Implement python and web development concepts to manage client’s multiple products within Sanlam.

Group No. 3

**Software Requirement Specification (SRS)**

1. INTRODUCTION:

A web application that will provide clients with an overview and make it possible for them to manage all their different insurance plans from one place.

* 1. PURPOSE

**Educational purposes:** The purpose of this project is to solve for clients that have multiple products within one provider, this will enable the user to load all their products in one place. Once loaded the user does not need to login into different platforms to access or modify their products. The functionality of this platform enables the user to make changes to their policies, make a single deposit where each product can then debit their monthly premium and the user can track their spending/ get a visual representation of their contribution.

1.2 FUNCTIONALITIES OF THE SYSTEM:

1)    Client:

Will login via a website. Access all the insurance policies. Manage them all and make payments. Also get to analyse their spending.

2)    System:

System will check credentials. Retrieve user data from a database and retrieve all the policies belonging to that user and keep track of the users spending.

* 1. OPERATING ENVIRONMENT:
* Devices that have access to internet i.e. Desktop PC , Laptop, Tablet, Smartphone

1. SOFTWARE REQUIREMENTS:

*SR1*: Client login screen

Login credentials: Correct username and password is required to login and have authentication for user

*SR2*: Failure cases for logging in

Prompt user to enter valid credentials

*SR3*: Loading active products (how will we retrieve data)

Display all active insurance products the user currently has.

*SR4*: Manage

User can manage view each insurance policy, ability to modify the products to their liking i.e.

* View Each Insurance Policy: User can click on insurance policy to view the inner details.
* Removing a Policy: Allow a user to get out of a policy.
* Edit Beneficiary: Allow the user to edit who will benefit from the policy.
* Pay Policy: User can pay each policy manually.

*SR5*: Failure cases:

* Failure Cases for Removing a Policy: Check if the user is allowed to remove it.
* Failure Cases for Editing the Policy: Documents of the new users are required.
* Failure Cases for Payment a Policy: Is a valid amount inserted.

*SR6*: Pay All Policies:

User can let the system decide which policies to pay based on importance or show any lapsed/ pending payments

*SR3*: Display Overall Payments.

Show the user a chart of how much they have paid, or tracking their payment rate(display the dates they missed).

**UML DIAGRAM**

Graphical user interface

Description automatically generated

CONCLUSION:

PLEASE ADD CONCLUSION HERE IF YOU HAVE ANY.