MAKERERE UNIVERSITY

Virtual Portfolio Manager

By

BSE 22-26

Name	Registration No.	Student No.
Bartile Kapkusum Emmanuel	**17/U/3508/PS**	**217003654**
Bwayo Joel	**17/U/3777/EVE**	**217010218**
Khalid Luwaki	**18/U/23400/EVE**	**1800723400**

Software Design Document

DEPARTMENT OF NETWORKS SCHOOL OF COMPUTING AND INFORMATICS TECHNOLOGY

A Software Design Document submitted to the School of Computing and Informatics Technology

For the Study Leading to a Project Proposal in Partial Fulfillment of the

Requirements for the Award of the Degree of Bachelor Of Makerere University.

Supervisor: Dr Joab Agaba

Department of Networks School of Computing and Informatics Technology, Makerere University

blessedjagaba@gmail.com, +256-41-540628, Fax: +256-41-540620

Table of Contents

List of Tables

List of Figures

1. INTRODUCTION

1.1 Purpose

The purpose of the software design document is to effectively describe the architecture and system design in order to aid the system developers and testers to understand the system architecture, system design, modules, intended functionality of the system and the technologies used to develop and implement the system. This document aims to give the development team a better understanding of how the system is to be developed.

This project collected data which can be used by other developers and researchers developing projects or researching the same topic. The target audience for this document includes Project designers, project developers, testers, project supervisors and researchers.

1.2 Scope

The Virtual Portfolio Manager (VPM) is a combination of a web and a mobile phone App system that integrates start-ups and investors. The VPM automatically links the investor's interested field of business with potential start-ups in the same field. The system helps startups to keep track of their financial progress from their profile and captures all daily and authentic activities of a startup and it can Rank all startups for investors to identify the best startups their issues out annual Certificates.

The VPM shall contain blockchain technology and manage and execute smart contracts between a startup and an investor. The electronic notary shall be used to affix the authorized seal and signatures to certified documents. The notary startup certificate issuing activity shall use cryptography and a secured public key to manage, create, store and distribute the digital certificate. The documents shall be saved on blockchain and any modifications on such documents are detected and easily traced through the timestamps placed on the documents. This technology shall ease investment in startups in a secured and authentic time-saving process.

The VPM system will be programmed in blockchain languages which include Solidity, NodeJs, and Flutter. The objective of the VPM system is to help improve the current state of investment protocols in the Business incubators in Uganda.

Benefits of the VPM system

- The system shall reduce the risks of investing in non-potential startups who fake data and submits an authentic document to the investors leading to loss of money. The system will have the ability to certify existence and carry out the proof of existence function effortlessly by saving the documents in the blockchain which ensures privacy and proof of ownership on document release.

- The VPM shall improve the effectiveness of managing startups in the business incubators through real-time smart contracts between startups and investors which reduces time in processing investment deals.

- The VPM shall enable startups, investors and business incubators to improve their transparency and accountability. This is because no one will make any alteration in the logged data within a blockchain until he gets permission for the same.

- The VPM shall have an easier hiring process for business incubators. Better and faster hiring will be done if the data of all startups and employees are uploaded to a database. This is because at the moment hiring is a challenging task for higher managers as they need to call a previous company of the employee to find the history of an employee or startup they want to hire or invest in. This technology will save countless human hours and resources which are used in verifying the resume of a job applicant or a startup.

Goals of the VPM system

- To reduce risks of investing in non-potential startups who fake data and submit an authentic document to the investors leading to loss of money

- To enable startups, investors and business incubators to improve their transparency and accountability.

- To improve the current state of managing startups in the business incubators through real-time smart contracts between startups and investors which reduces time in processing investment deals.

1.3 Overview

This document consists of 8 sections; Section 1 is the introduction that contains the purpose which explains the goals, objectives and benefits of the Virtual Portfolio Manager, the scope, and the overview sub-section which explains all the different sections of the document, the definition of acronyms and abbreviations, reference material which lists all the references and citations. This section gives a high-level explanation of what the software design document it's all about.

Section 2 of this document is the system overview which gives a general description of the functionality, context and design of the project and provides any background information if necessary.

Section 3 is the system architecture which contains the architectural design, decomposition description and the design rationale. This section breaks down the project into smaller modules illustrating them in terms of DFDs, interface specifications, and hierarchical diagrams and more.

Section 4 is the data design that contains the data description which explains how the information domain of the VPM is transformed into data structures and the data dictionary which contains the major data along with their types and descriptions.

Section 5 is the component design which contains a closer look at what each component does in a more systematic way.

Section 6 is the human interface design which contains the overview of the user interface, screen images, and screen objects and actions. The overview of the user interface describes the functionality of the system from the users' perspective. The screen images display screenshots showing the interface from the users' perspective. The screen objects and actions detail a discussion of screen objects and actions associated with those objects.

Section 7 is the requirements matrix which provides a cross-reference that traces components and data structures to the requirements in the SRS document.

Section 8 is the Appendices that provide supporting details that could aid the understanding of the software design document.

1.4 Reference Material

Kane, Gerald C. "A Three-Step Process for Scaling Digital Innovation." MIT Sloan Management Review 10 (2018): 1–5. MIT Sloan Management Review. Web.

USAID. "Policy Brief Information and Communications Technology (ICT) Solutions for Inclusive Agricultural Value Chains." Feed the Future (2018): n. pag. Feed the Future. Web.

Bogner, Andreas, Mathieu Chanson, and Arne Meeuw. "A Decentralised Sharing App Running a Smart Contract on the Ethereum Blockchain." ACM International Conference Proceeding Series. Vol. 07-09-November-2016. Association for Computing Machinery, 2016. 177–178. ACM International Conference Proceeding Series. Web.

Tomlinson, Bill et al. "Analyzing the Sustainability of 28 'Blockchain for Good' Projects via Affordances and Constraints." Information Technology for Development 27.3 (2021): 439–469. Information Technology for Development. Web.

Khatib, Mounir El et al. "The Impact of Blockchain on Project Management." ICIC Express Letters 15.5 (2021): 467–474. ICIC Express Letters. Web.

Guustaaf, Edward et al. "Blockchain-Based Education Project." Aptisi Transactions on Management (ATM) 5.1 (2021): 46–61. Aptisi Transactions on Management (ATM). Web.

Baharmand, Hossein et al. "Developing a Framework for Designing Humanitarian Blockchain Projects." Computers in Industry 131 (2021): n. pag. Computers in Industry. Web.

1.5 Definitions and Acronyms

Abbreviation	Definition
VPM	Virtual Portfolio Manager
SDD	Software Design Document
GUI	Graphical User Interface
ETH	Ethereum

2. SYSTEM OVERVIEW

2.1 Background information

At present, Crowdfunding source of raising funds typically for startups or projects has gained popularity with most startups resorting to the use of Crowdfunding platforms to raise funds in exchange for equity because it is relatively inexpensive and uncomplicated in nature. In the existing model, a Pool of people contribute small amounts of money towards a project or cause and expect some financial returns. The call for a solution to issues related to security, investor abuse and, illegal transactions that could plague crowdfunding has led me to investigate the implications of blockchain in Crowdfunding.

2.2 System description

VPM is an online equity crowdfunding platform managed by smart contracts on Ethereum. The platform enables start-ups and projects to raise funding in return for equity. An individual can invest a relatively small amount of money in order to receive a stake in a company at an early stage hoping to get good returns in the long term as the startup/project grows. Along with funding, VPM also enables startups/projects to manage their business through the multiple functionalities that the system will provide such as financial management, statistical analysis, and risk management. The system shall allow business hubs to monitor the health of startups through a web portal.

What makes VPM truly unique is its decentralized and autonomous approach to crowdfunding using smart contracts that are deployed on the Ethereum blockchain and a mobile app created using Google's Flutter which can be used on both Android and IOS.

The platform can create camps in which normal users can purchase equity by investing in VPM (Virtual Portfolio Manager token) an ERC20 fungible token exclusive only to the VPM platform. This platform's exclusive ERC20 token along with smart contracts enables VPM to tackle the issue of trust and security that plagues all the existing crowdfunding platforms.

The web portal shall be used by the business hubs and startups/projects to manage their information.

Product Perspective

The VPM shall be a group of system components separately built. These components include a MongoDB database, a system API, a blockchain application, a web interface application and a mobile interface application.

The MongoDB database shall store mutable content while the blockchain network shall store immutable content.
The system API shall be an intermediary between the blockchain application and the user to process and format the data to be stored. The user interface shall be built to facilitate both mobile and web.

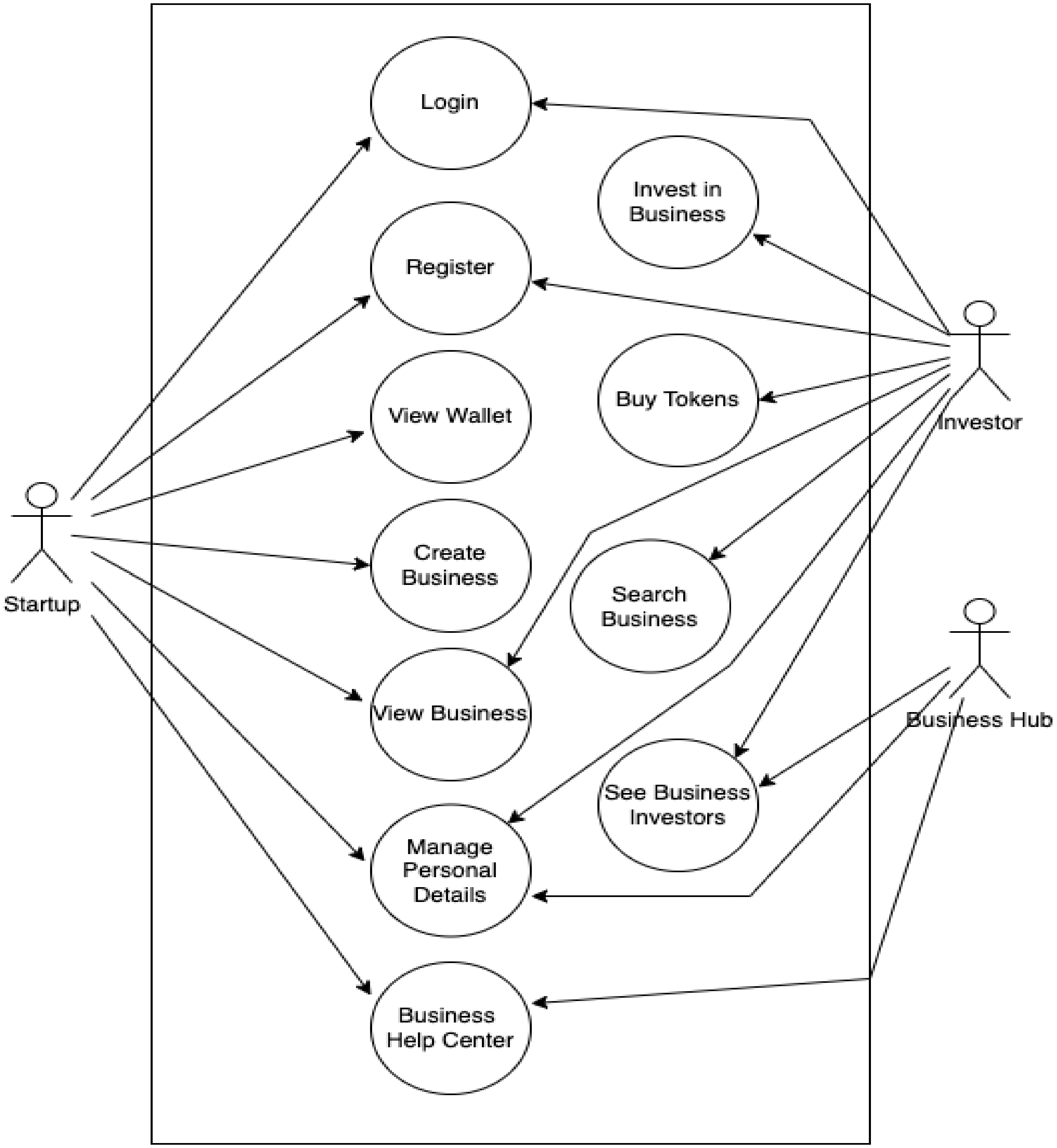


Figure 1 Use case Diagram

3. System Architecture

3.1 Architectural Design

Here's the basic architecture, the remote database is only accessible through the WEB API. -technically through the webserver hosting the WEB API- The mobile application consumes the WEB API, interacts with the Ethereum blockchain through the Nodejs Servers which are part of the Ethereum Network and save some data locally.

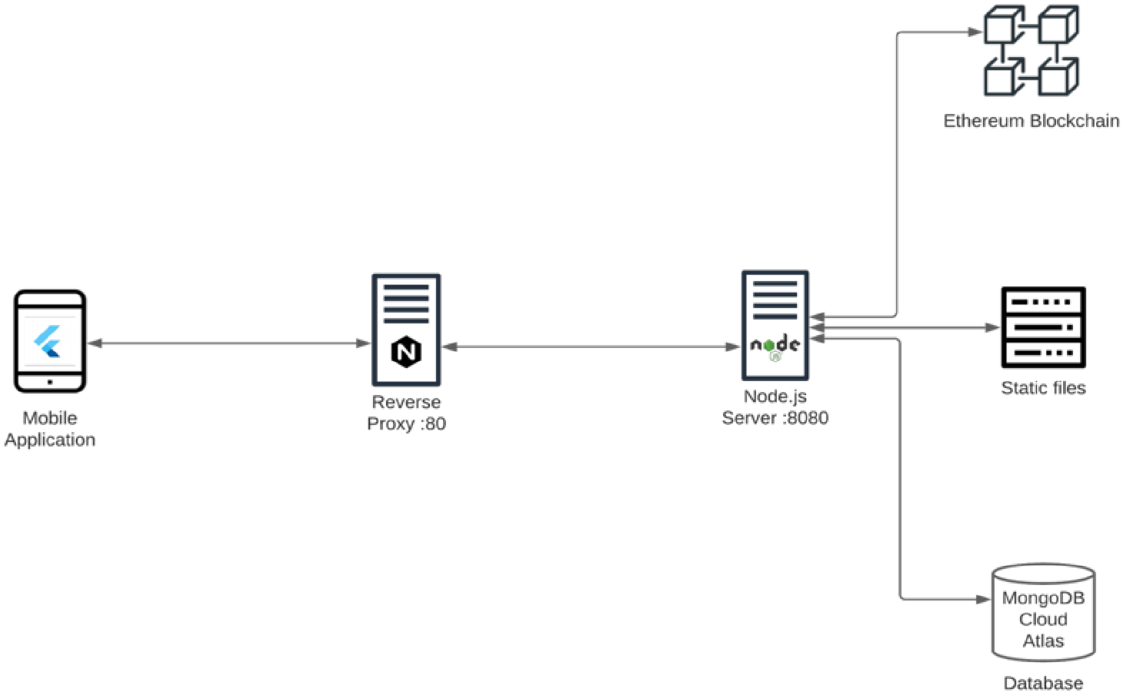


Figure 2 Block diagram of the VPM system

3.2 Mobile App sub system Architecture

The VPM Mobile App sub system shall comprise 3 layers thus following the layered architecture. The Model-View-ViewModel (MVVM) helps to separate the logic of an application from its user interface (UI). This separation makes the application testing and maintenance a lot easier. Also, it greatly improves code re-use. Besides, it allows developers and UI designers to work simultaneously and collaborate more often.

There are 3 main components of the MVVM pattern: the View, the View Model, and the Model.

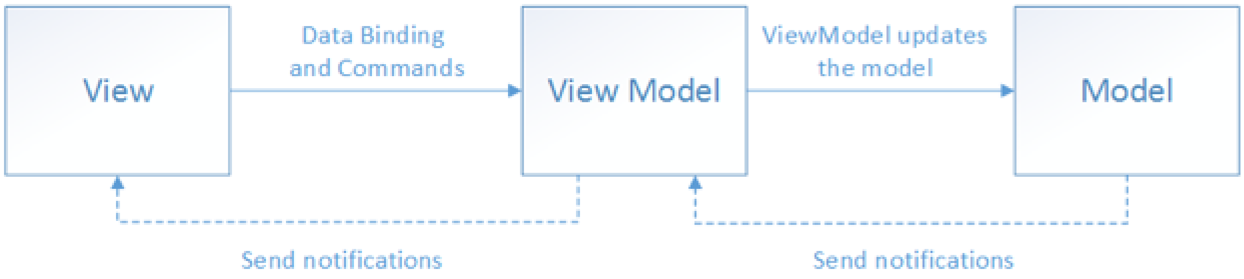


Figure 3 VPM Mobile App subsystem Architecture

As you can see the view is aware of the View Model and the View Model is aware of the Model. Therefore the View Model isolates the View from the Model.

- **The View:** is related to the user interface, whether it's a Content Page or just a Content View. It contains all categories or components related to the presentation of data. In this system this layer will be responsible for presenting the data rendered by the system. In this system, the presentation layer will be through a mobile application developed for both Android and IOS devices using flutter, an open-source user interface software development kit created by google.

- **The View Model:** contains the business logic related to the View or references to it, the actions of each Command, and a reference to the Model. This layer deals with all the business rules and constraints and it dictates how data will be handled or processed. At this layer sits the Web Server (Backend API). It is responsible for the core functionalities of the system like user management including creation, deletion, update and retrieval of user data; it will be developed using NodeJs Framework.

- **The Model :** are classes that are schemas to the app's data with some validation logic here and there. The View Model performs CRUD operation on the Model. This layer is responsible for storing all the data in the system. MongoDB will be used to store data of this system into a database which is part of this layer.

3.3 Decomposition Description

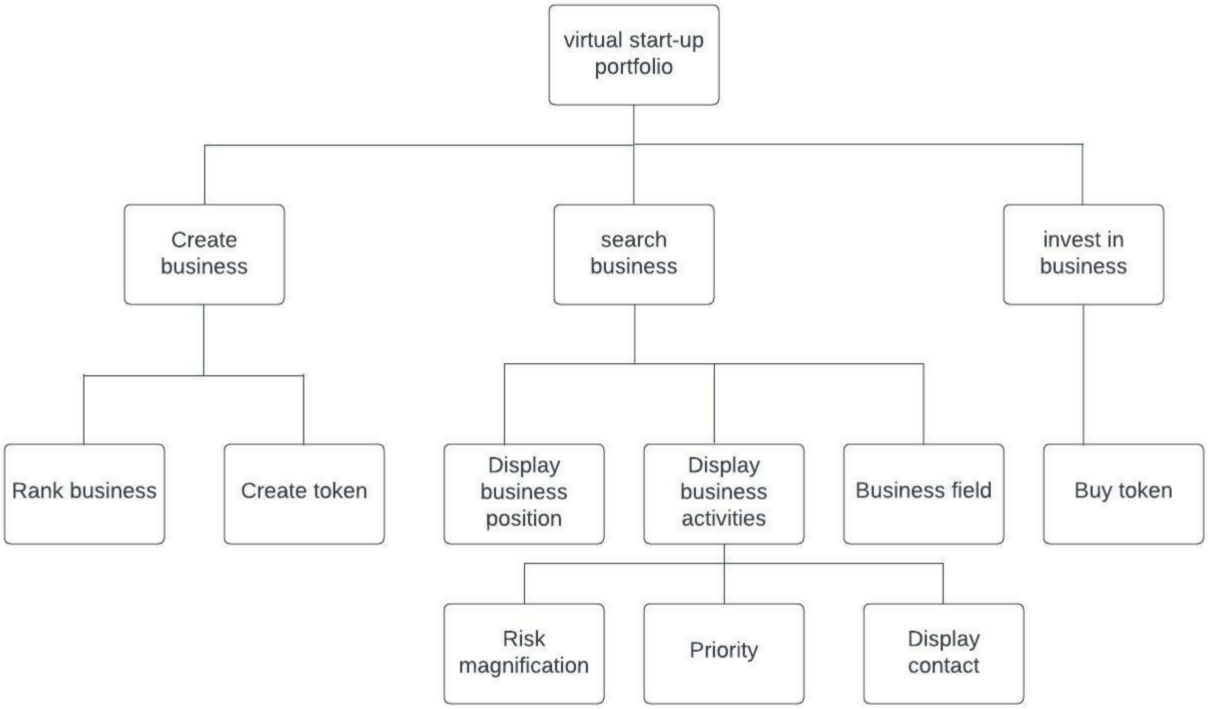


Figure 4 VPM system decomposition diagram

3.4 Design Rational

The VPM is an advanced application that interacts with the Ethereum Block and consume a Web API using chains. The VPM have multiple clones of startups each clone is a full node . The VPM keeps records of all transactions qnd some transactions are send, receive and deposit. The VPM digitally excuse these transactions and many more. Each set of transactions once executed and mined is assembled into a blockchain and a blockchain is all these blocks linked together. The blockchain acts as a ledger record of transactions. The blocks are added to the blockchain (mined) following a set of rules and instructions in the Ethereum smart contract code and documents. The VSP contracts shall be compiled in low level bytecode to run on the Ethereum virtual machine on every Ethereum node.

What's a blockchain?

A blockchain is all these blocks linked together. The blockchain acts as a ledger (record of transactions). Since all of the full nodes are equal, they all have this ledger which gives it the attribute: Distributed.

The blocks are added to the blockchain (mined) following a set of rules and instructions called consensus algorithms. There are many algorithms like Proof of authority (POA), Proof of Work (POW), Proof of Stake(POS), and Delegated Byzantine fault tolerance(dBFT). These algorithms are very theoretical and beautiful, but I won't cover them for now because it is out of the scope of this project.

What's a smart contract?

A “smart contract” is simply a piece of code that is running on Ethereum. It’s called a “contract” because code that runs on Ethereum can control valuable things like ETH or other digital assets. - Ethereum docs. However, the word contract has no legal meaning under any circumstance. The main features of a smart contract are: deterministic and immutable. Immutable means that once deployed the code cannot be edited. Deterministic is getting the same outcome for the same given parameters. There isn't anything random. Recently smart contracts are developed using the Solidity programming language. Smart contracts run on the Ethereum Virtual Machine and the EVM runs as a local instance on every Ethereum node. But, for a smart contract to run on the EVM it is compiled to low-level bytecode. Once compiled the smart contract is deployed on the Ethereum network. To deploy a smart contract, the contract developer sends a transaction without specifying a recipient.

What's GAS?

Let's say you have a car and you want to go from home to work, you're going to need a definitive amount of oil and you pay for it using USD. This amount will never change unless you change your car. Remember from the last paragraph, that a smart contract is compiled to bytecode. Under the hood, the EVM uses a set of instructions (called opcodes) to execute specific tasks. The bytecode previously mentioned is a set of these opcodes and each opcode execution costs a different amount of gas. So analogically speaking, the car is the EVM, Gas is the oil and Ether is USD. To execute a transaction like calling a function of a smart contract you're gonna need gas priced in Ether. Note that once created a block has a property called Gas limit and if your transaction requires more than this limit it will not be executed.

Why do we need MongoDB ?

A fully decentralized application has its data only stored on the Ethereum blockchain. Using the Ethereum Blockchain as the single data storage location has numerous limitations. Modification requests cost ether(money) and are slow to process (15 secs) which causes a bad user experience. In Addition each smart contract has a maximum capacity of around ~1.2 GB. Our Application is semi-decentralized which means we use a web api and a database to process large data in addition to the Ethereum blockchain.

What is an VPM Ethereum Wallet ?

Wallets are applications that make it easy to hold and send VPM, as well as interact with applications built on Ethereum. The most important properties of an VPM Wallet are the address and the private key , and we won't use more than these keys for interacting with our smart contracts.

Private keys are generated as random 256 bits, which is 64 (hex) characters or 32 bytes. You must make sure that the random key generated is unique enough, do not generate it using numbers only because it will be prone to collision attacks. In this project, we will be using both ethereum and Metamask for demonstration purposes. Also, Metmask is essential to deploy our smart contracts later.

After the private key generation, VPM Ethereum public keys of length 128 characters are created using an algorithm called Elliptic Curve Digital Signature Algorithm (ECDSA). Ethereum uses secp256k1 to generate public keys. A public key is a point in this Elliptic curve algorithm. To create an Ethereum Address, the keccak256 algorithm is applied to the coordinates of the public key.

After developing our smart contracts, we will deploy them to the blockchain for further testing and we will use a Testnet because it's free. For smart contracts development we will be using Remix, an online Ethereum IDE.

4. DATA DESIGN

4.1 Data Description.

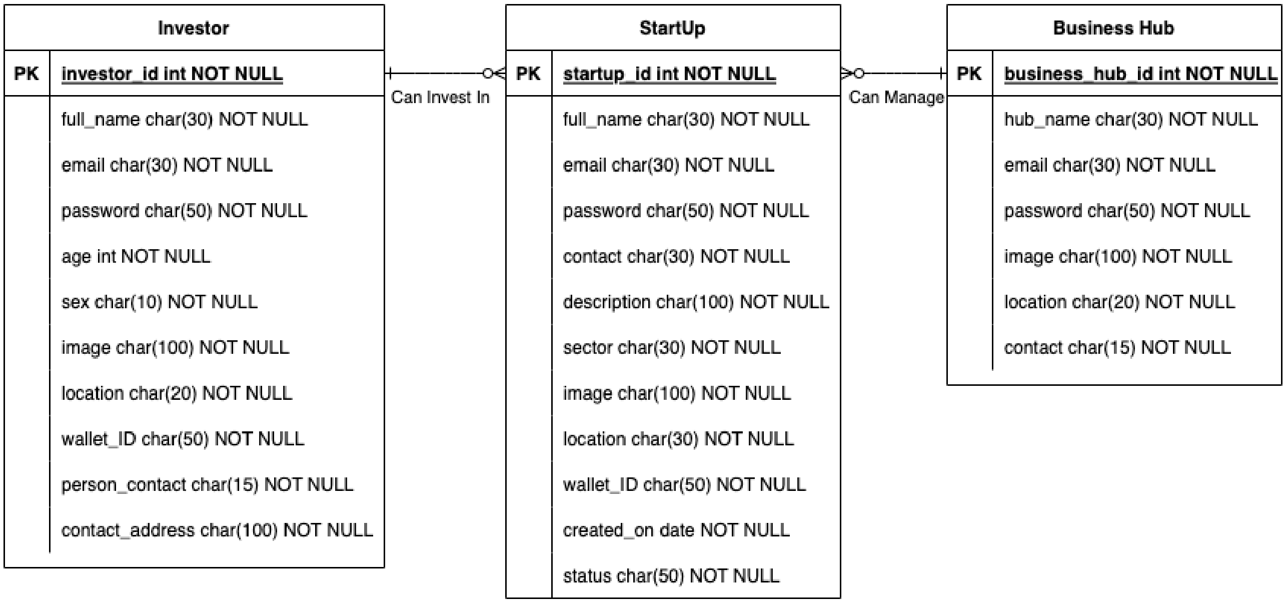
Since the VPM will be used to store data about various entities like startup, investor, business hub etc., the users of the mobile application system are expected to create, manipulate, update and delete data. This section therefore gives a description of the data structures of the system.

Entity	**Description**	**Attributes**
Startup	The startup entity stores data about startups or businesses who can also enter data about the business they are taking care of.	● Startup_ID (Primary Key) ● FullName ● Email ● Password ● Contact ● Description ● Sector ● Image ● Location ● Wallet_ID ● Created_on ● Status
Investor	Investor stores data about startups, he/she is interested in. All investors are	● Investor_ID (Primary Key) ● FullName

	required to register and buy tokens in order to invest in startups.	● Email ● Password ● Age ● Sex ● Image ● Location ● Wallet_ID ● Person_Contact ● Contact_Address
Business Hub	This entity stores data about the startup colaboration that can be made between startups and business hubs. The also manage support center for investors and hubs to inquire.	● BusinessHub_ID (Primary Key) ● HubName ● Email ● Password ● Image ● Location ● Contact

Table 1 Data Description

The diagram below shows an entity relationship diagram which clearly defines the relationship between the above entities. It is a blueprint of a database that shall later be implemented as a database. The main components of E-R model are: entity set and relationship set.



Investor	
PK	investor_id int NOT NULL
	full_name char(30) NOT NULL
	email char(30) NOT NULL
	password char(50) NOT NULL
	age int NOT NULL
	sex char(10) NOT NULL
	image char(100) NOT NULL
	location char(20) NOT NULL
	wallet_ID char(50) NOT NULL
	person_contact char(15) NOT NULL
	contact_address char(100) NOT NULL

StartUp	
PK	startup_id int NOT NULL
	full_name char(30) NOT NULL
	email char(30) NOT NULL
	password char(50) NOT NULL
	contact char(30) NOT NULL
	description char(100) NOT NULL
	sector char(30) NOT NULL
	image char(100) NOT NULL
	location char(30) NOT NULL
	wallet_ID char(50) NOT NULL
	created_on date NOT NULL
	status char(50) NOT NULL

Business Hub	
PK	business_hub_id int NOT NULL
	hub_name char(30) NOT NULL
	email char(30) NOT NULL
	password char(50) NOT NULL
	image char(100) NOT NULL
	location char(20) NOT NULL
	contact char(15) NOT NULL

Figure 5 Entity Relationship Diagram

4.2 Data Dictionary.

Table 2 Investor Data Dictionary

Field Name	Data type	Field Size for Display	Description
Investor_ID (Primary Key)	Char	20	Unique identifier of the investor
FullName	Char	30	Investor’s full name
Email	Char	30	Investor’s email
Password	Char	50	Investor’s Password
Age	Int	3	Investor’s Age
Sex	Char	10	Investor’s Sex
Image	Char	100	Investor’s Image
Location	Char	20	Investor’s Location
Wallet_ID	Char	50	Unique identifier of the investor’s VPM token etheruem blockchain wallet
Personal Contact	Char	15	Investor’s Contact
Contact Address	Char	100	Investor’s Address

Table 3 Business Hub Data Dictionary

Field Name	Data type	Field Size for Display	Description
Business_Hub_ID (Primary Key)	Char	20	Unique identifier of the business hub
Hub Name	Char	30	Business hub name
Email	Char	30	Business hub email
Password	Char	50	Business hub Password

Image	Char	100	Business hub Image
Location	Char	20	Business hub Location
Contact	Char	15	Business hub Contact

Table 4 Startup Data Dictionary

Field Name	Data type	Field Size for Display	Description
Startup_ID (Primary Key)	Char	20	Unique identifier of the startup
FullName	Char	30	Startup full name
Email	Char	30	Startup email
Password	Char	50	Startup Password
Contact	Char	30	Startup Contact
Description	Char	100	Startup Description
Sector	Char	30	Startup Sector
Image	Char	100	Startup Image
Location	Char	20	Startup Location
Wallet_ID	Char	50	Unique identifier of the Startup VPM token etheruem blockchain wallet
Created On	Date	20	When the startup was created
Status	Char	50	Business Investment Status

Alphabetically list the system entities or major data along with their types and descriptions. If you provided a functional description in Section 3.2, list all the functions and function parameters. If you provided an OO description, list the objects and its attributes, methods and method parameters.

5. COMPONENT DESIGN

5.1 The VPM Investment Processes.

As shown in the flowchart, the investor goes to the investor section and creates an account to be able to invest in the startups. After that the investor enters their wallet address and the amount of tokens that they would like to buy. Then the investor will pay the equal price in relation to 1 token in fiat currency. Then the investor is verified and the tokens are sent to the startup wallet. The investments details are then saved in the blockchain database.

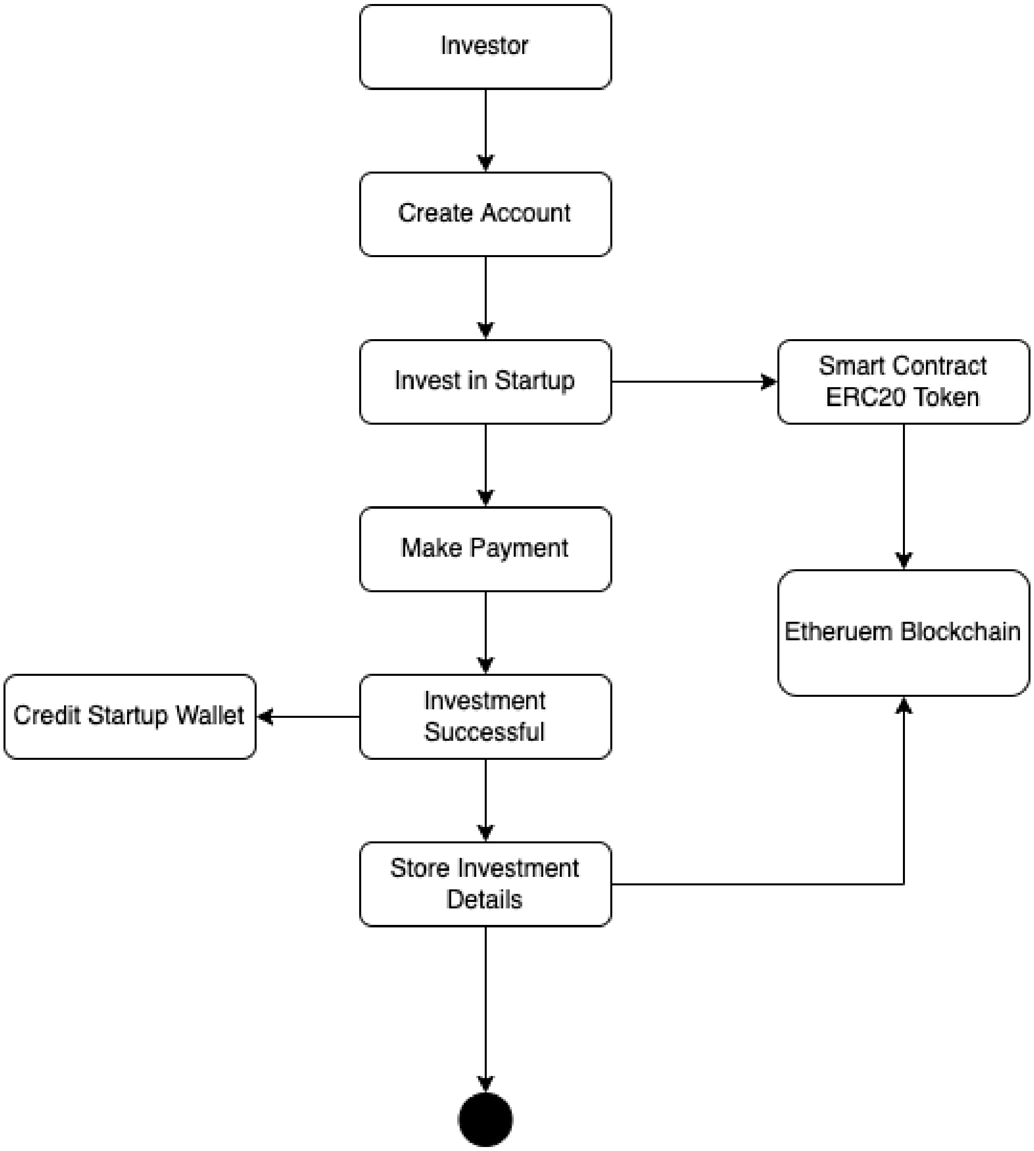


Figure 6 VPM Investment Flow chart

6. HUMAN INTERFACE DESIGN

6.1 Overview of User Interface.

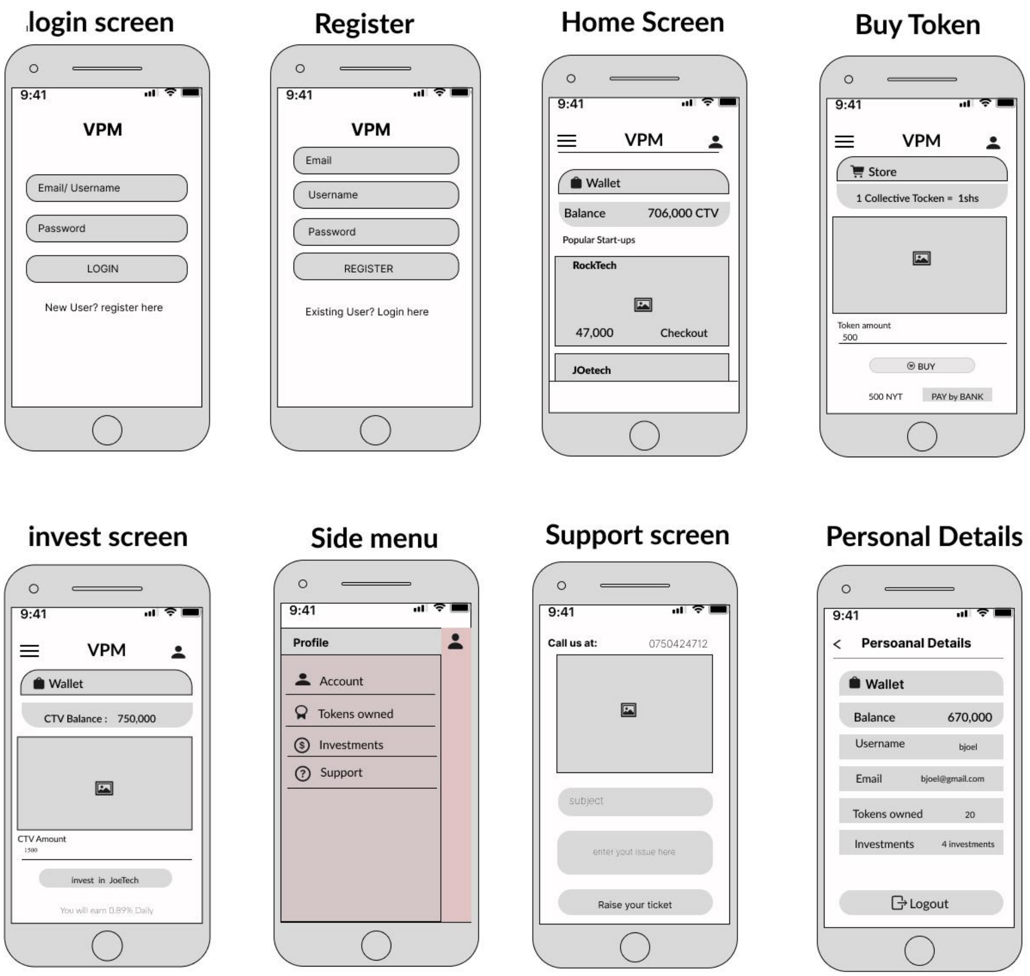


Figure 7 Mobile application screen mockups

Describe the functionality of the system from the user's perspective. Explain how the user will be able to use your system to complete all the expected features and the feedback information that will be displayed for the user.

Virtual portfolio manager requires its users to have an account before they can use it. If the user has an account, he/she logs into the account otherwise the user has to first register to use the system. And registering requires the user to enter his username, email, and password.

When the users log into the system, the home screen is opened. On the home screen, the user is able to see a list of startups that are currently trending so he/she can scroll through the listed start-ups.

The home screen also shows the balance of the investor.

If the user selects a startup, the invest screen is opened where the user can see details about the business including how much the token for that particular start-up or business costs.

The user can also keep track of his activities. By selecting the side menu, the user is able to view his account, the number of tokens he/she owns, investments made, and also the help center.

The support screen contains contacts. In case of any problem, the user can reach out to the help center for help or even rise a flag by leaving a comment in the provided space for submission.

Users also have their personal details stored in the account as viewed on the personal details screen.

7. REQUIREMENTS MATRIX

The sections of the matrix marked with an indicate a particular component being used to achieve the requirement in question

Requirement	Requirement Type	Mobile application	Web application	Blockchain Application
The system shall provide a mobile app interface for start-ups to register their businesses.	**Functional**	**X**	**X**	**X**
The system shall provide a mobile app interface for investors to register.	**Functional**	**X**	**X**	
The system shall provide an interface for start-ups to log in.		**X**	**X**	
The system shall provide an interface for investors to log in.	**Functional**			
The system shall provide an interface for start-ups to view their wallets which hold all their funds.	**Functional**	**X**	**X**	**X**
The system shall have an interface where they create the businesses which are to be invested in.	**Functional**	**X**	**X**	
The system shall have an interface for start-ups to view their created business.	**Functional**	**X**	**X**	
The system shall an interface for investors			**X**	

to view created businesses	**Functional**	**X**		
The system shall provide an interface for start-ups to manage their personal details	**Functional**	**X**	**X**	
The system shall provide an interface for investors to manage personal details.	**Non-Functional**	X	**X**	
The system shall provide an interface for businesses wishing to invest in the available start-ups to manage their personal details	**Non-Functional**	X	**X**	
The system shall provide an interface to provide a help center for all its users	**Non-Functional**	**X**	**X**	
The system shall provide an interface for investors to perform investment activities.	**Functional**	**X**	**X**	**X**
The system shall provide an interface for the investor to buy tokens	**Functional**	**X**	**X**	**X**
The system shall provide an interface for investors to search for businesses.	**Non-Functional**	**X**	**X**	
The system shall provide an interface for investors to see other business investors investing into the businesses within the system.	**Non-Functional**	**X**	**X**	

8. APPENDICES

Software Requirements Specification (SRS) Describes what the software will do and how it will be expected to perform.

Virtual Portfolio Manager (VPM) is a web and mobile phone App system that integrates start-ups and investors designed and implemented to automatically link the investor's interested field of business with potential and authentic start-ups in the same field.