



## EDU-KIDS

By :

1. Mahmoud Saad Farhat
2. Omar Mohamed Ibrahim
3. Abdulrahman Muohamed Nedal Baiasy

## Declaration

With gratitude, we wish to inform you that the project we are now submitting for evaluation is the result of our own efforts entirely. We have taken the necessary precautions to ensure the authenticity of our work and its compliance with copyright laws. Any external sources utilized have been appropriately cited and acknowledged within our text.

## Table of Contents

Cover Page .....	Error! Bookmark not defined.
Declaration .....	2
<b>1 INTRODUCTION AND Background.....</b>	<b>4</b>
1.1 Problem Definition .....	4
1.2 Motivation .....	4
1.3 Our Objective .....	5
<b>2 REQUIRMENT AND ANALYSIS.....</b>	<b>6</b>
2.1 Functional and Non-Functional Requirment .....	6
2.1.1 Functional Requitment:.....	6
2.1.2 Non-Functional Requitment:.....	6
2.2 Analysis and Design.....	8
2.2.1 Use Case: .....	8
2.2.2 Context Diagram: .....	9
2.2.3 Data Flow Diagram (Level-1):.....	10
2.2.4 Class Diagram:.....	12
2.2.5 Sequence Diagram: .....	13
2.2.6 Activity Diagram: .....	14
2.2.7 State Diagram: .....	15
2.2.8 ER-Diagram: .....	16
2.2.9 Mapping: .....	17
2.3 Design .....	18
2.3.1 Web Design:.....	18
2.4 Tools:.....	21

## 1.1 Problem Definition

In many Arab countries, financial literacy is not a standard part of the educational curriculum, leading to a lack of understanding of basic financial concepts among children. This gap in knowledge can result in poor financial decisions in adulthood.

### Solution Overview

Our project EDU-KIDS , introduces an interactive, web-based platform designed to fill this educational void by teaching children about money management, saving, investing, and budgeting through engaging activities and games.

## 1.2 Motivation

The idea for EDU-KIDS was inspired by the pressing need to equip the next generation in Arab societies with the tools and knowledge to navigate the financial challenges of modern life. Given the unique socio-economic landscape of the region, it's crucial to foster financial literacy from a young age.

## 1.3 Our Objective

The overarching objective of this project is to confront and alleviate the pervasive challenge inherent in both spoken and written communication the heightened anxiety induced by potential linguistic errors. Through a systematic approach involving meticulous analysis and targeted interventions, the project endeavors to identify and rectify these errors comprehensively.

The primary aim is to foster enhanced linguistic clarity and instill confidence among individuals. By deploying a user-centric website equipped with advanced language learning features, the project seeks to transcend the barriers posed by grammatical and structural issues in speech and writing. This initiative aspires to empower users by offering a supportive environment where they can easily identify, comprehend, and learn from their language-related errors. Additionally, the project strives to mitigate the common fear associated with verbal communication, providing efficient speech and grammar correction tools that analyze and rectify language errors extracted from various inputs. Ultimately, the project's goal is to create a comprehensive online resource that not only facilitates language learning but also makes the process more accessible, enjoyable, and effective for users.

## 2 PLANNING AND REQUIREMENTS.

This chapter serves as an illuminating guide, meticulously elucidating the intricate requirements that underpin the system's architecture. With a thorough exploration of User and System Requirements, coupled with a detailed examination of System Design intricacies, the chapter acts as a beacon, providing clarity on the foundational elements essential for the system's seamless functionality. It offers a comprehensive and detailed clarification of the requisites, ensuring a robust understanding of the system's operational framework and laying the groundwork for successful implementation.

### 2.1 Functional and Non-Functional Requirements:

#### 2.1.1 Functional requirements:

This section illustrates the user requirements of the system. The system developers suggest the following requirements themselves to be verified later by the client.

1. User Registration:

Users should be able to create accounts with unique usernames and passwords to access the platform.

2. Virtual Store:

The platform should feature a virtual store where users can browse, buy, and sell virtual goods.

3. Parental Controls:

The platform should include parental controls to allow parents to monitor and manage their children's activities and transactions.

#### 2.1.2 Non-Functional requirements:

This section presents other non-functional requirements to be taken into consideration when implementing the required application. In order to simplify the design, facilitate delivery of a demonstration application the following requirements and assumptions have been defined.

1. Availability:

The system should maintain high availability, with minimal downtime, to ensure users have consistent access to Edu-kids services.

**2. Usability:**

The user interface should be intuitive and user-friendly, promoting ease of use and ensuring a seamless experience for users interacting with the Edu-kids features

**3. Maintainability:**

The system should be designed for ease of maintenance, facilitating updates, bug fixes, and enhancements to ensure ongoing reliability and performance.

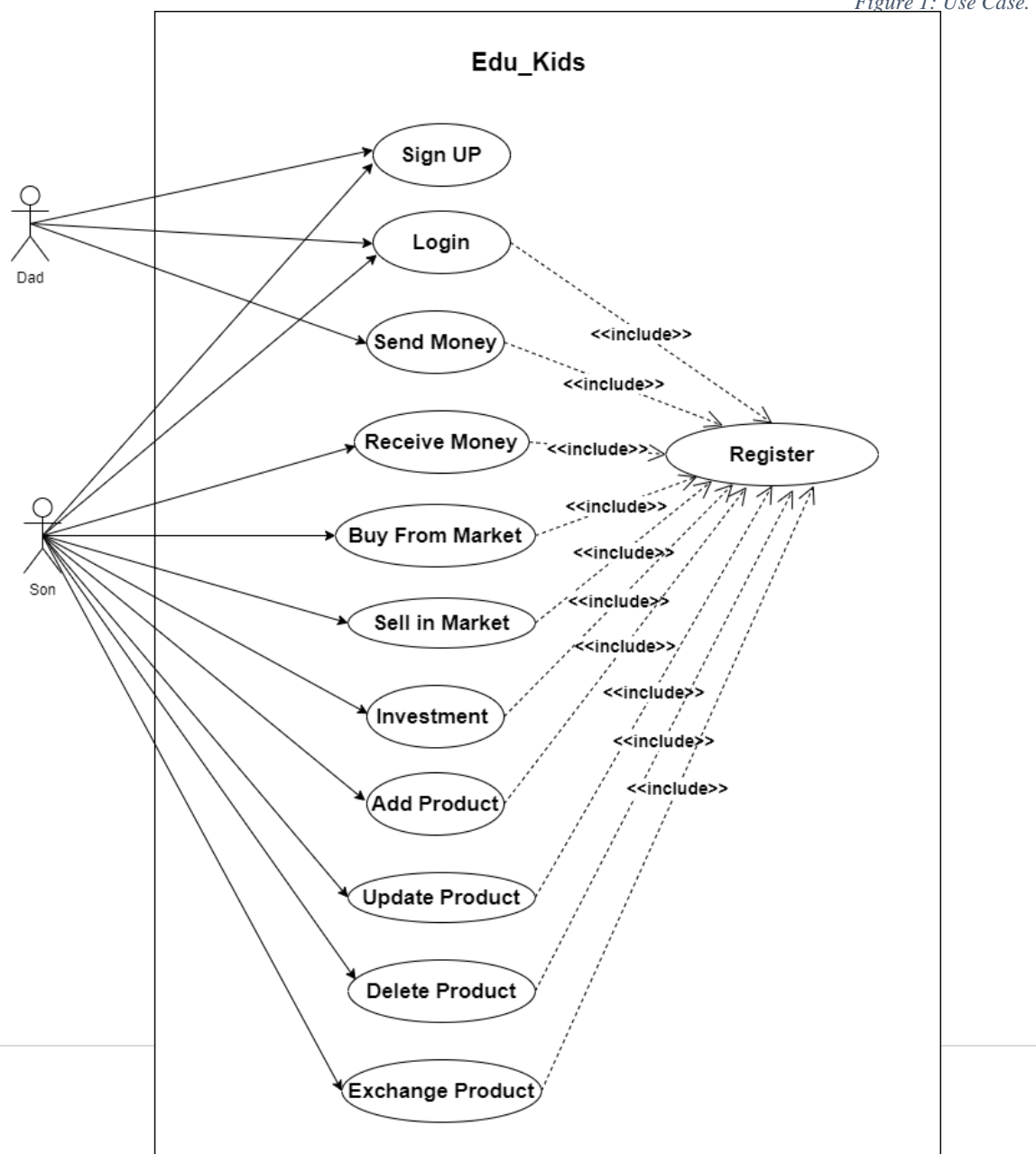
## 2.2 ANALYSIS AND DESIGN

### 2.2 Analysis

#### 2.2.1 Use Case:

In this diagram, a graphical depiction of a user's possible interactions with our system is presented. It shows various use cases and different types of users in our system and will often be accompanied by other types of diagrams as well. In this diagram the users are represented by ellipses.

##### 2.2.1.1 Use Case:



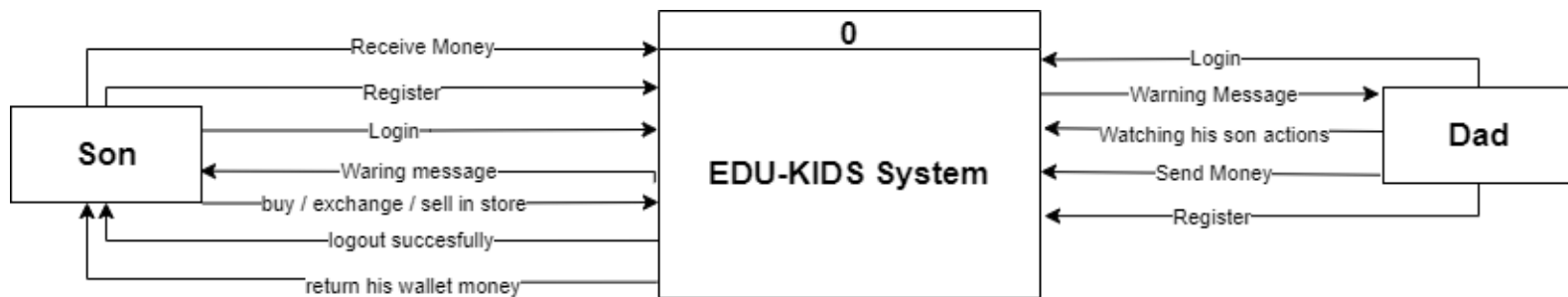


## 2.2.2 Context Diagram:

Context diagrams outlines how external entities interact with our internal software system. It also shows the interactions between a system and other actors (external factors). In addition to clarifying the context which the system will be part of and shows the entire system as a single process.

### 2.2.2.1 Context Diagram:

*Figure 2: Context Diagram*

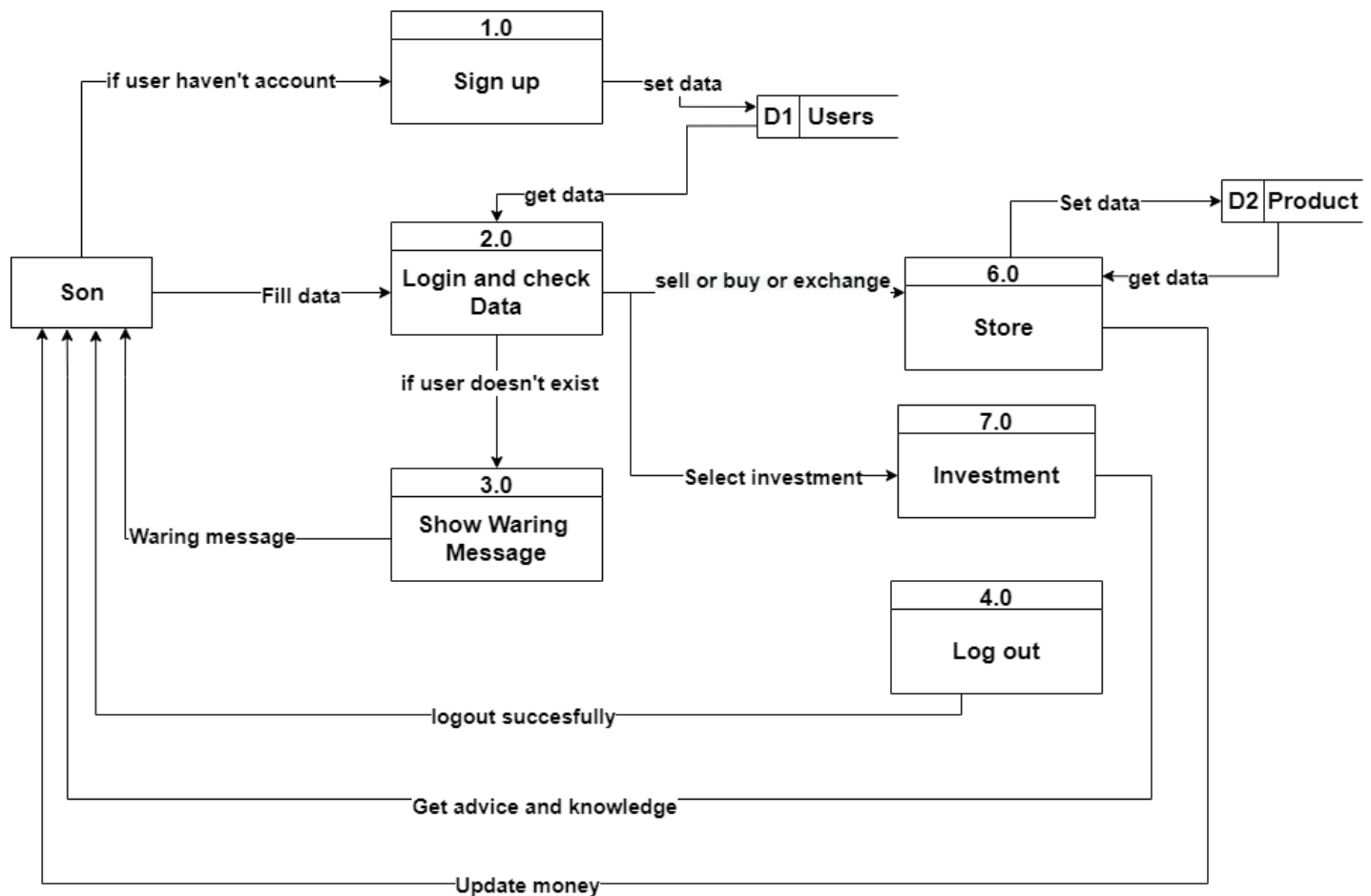


### 2.2.3 Data Flow Diagram (Level-1):

The data flow diagram shows the way information flows through our system. It includes data inputs and outputs, data stores, and the various sub-processes the data moves through.

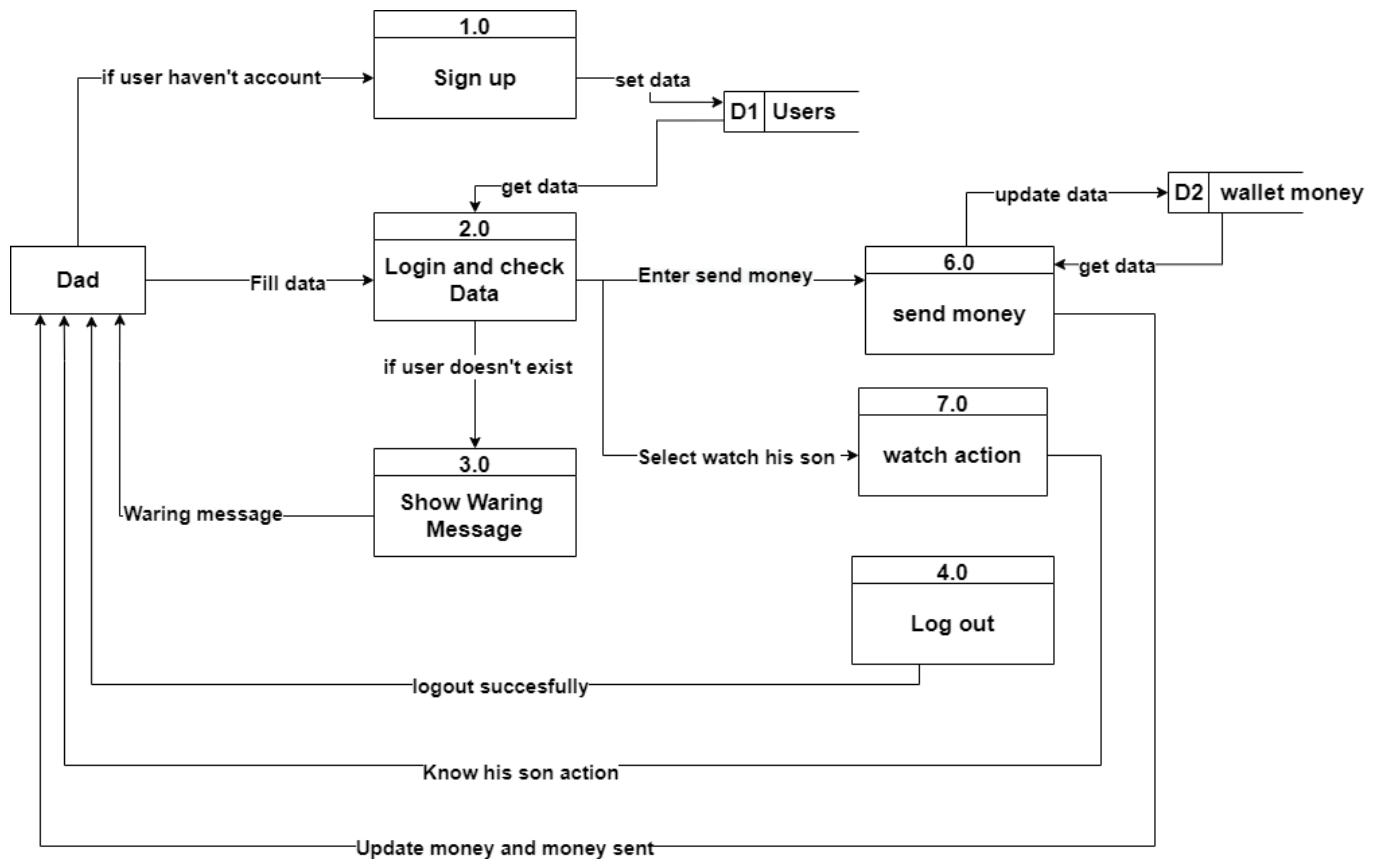
#### 2.2.3.1 DFD Diagram:

Figure 3: DFD-Diagram for son



2.2.3.2 DFD Diagram:

Figure 4: DFD-Diagram for dad

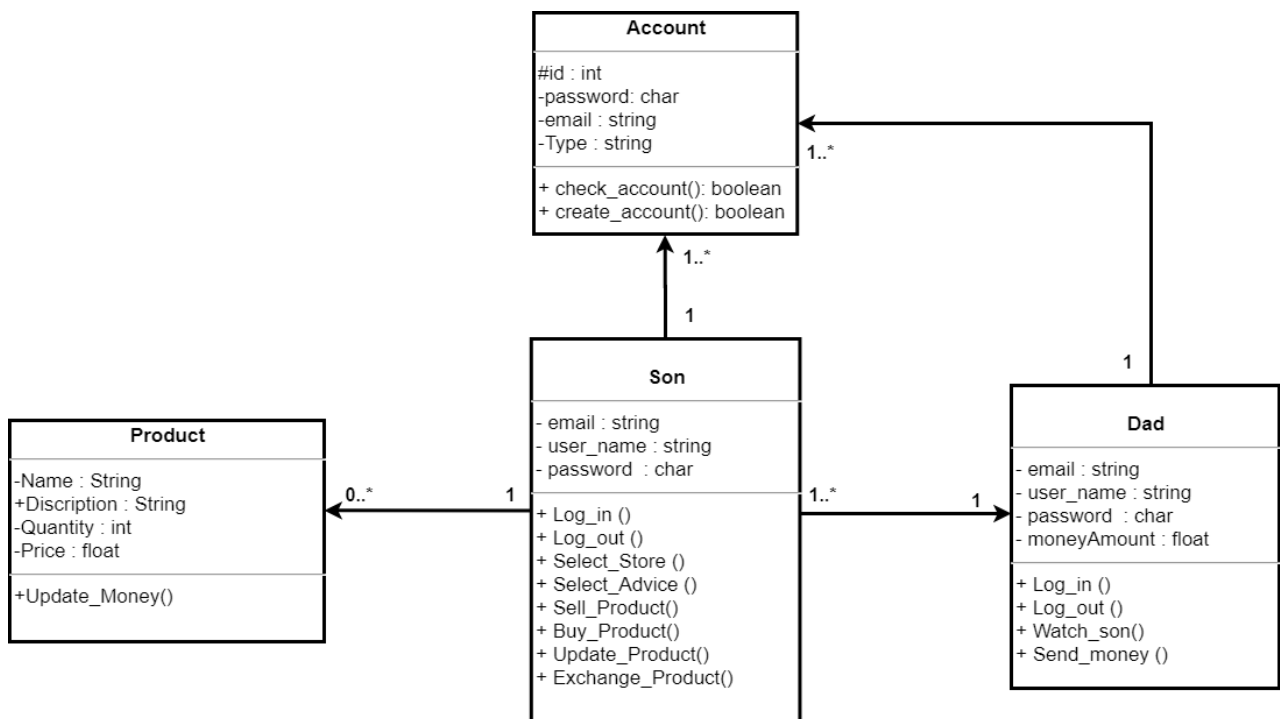


## 2.2.4 Class Diagram:

Class diagram is an important static UML diagram that describes the structure of a system by showing its classes, their properties, operations, methods, and the relationships between objects. It's the most abstract concept we use in system design. So, a basic understanding of the UML representation of class diagrams can help us understand the design of existing systems better.

### 2.2.4.1 Class Diagram:

Figure 5: Class Diagram

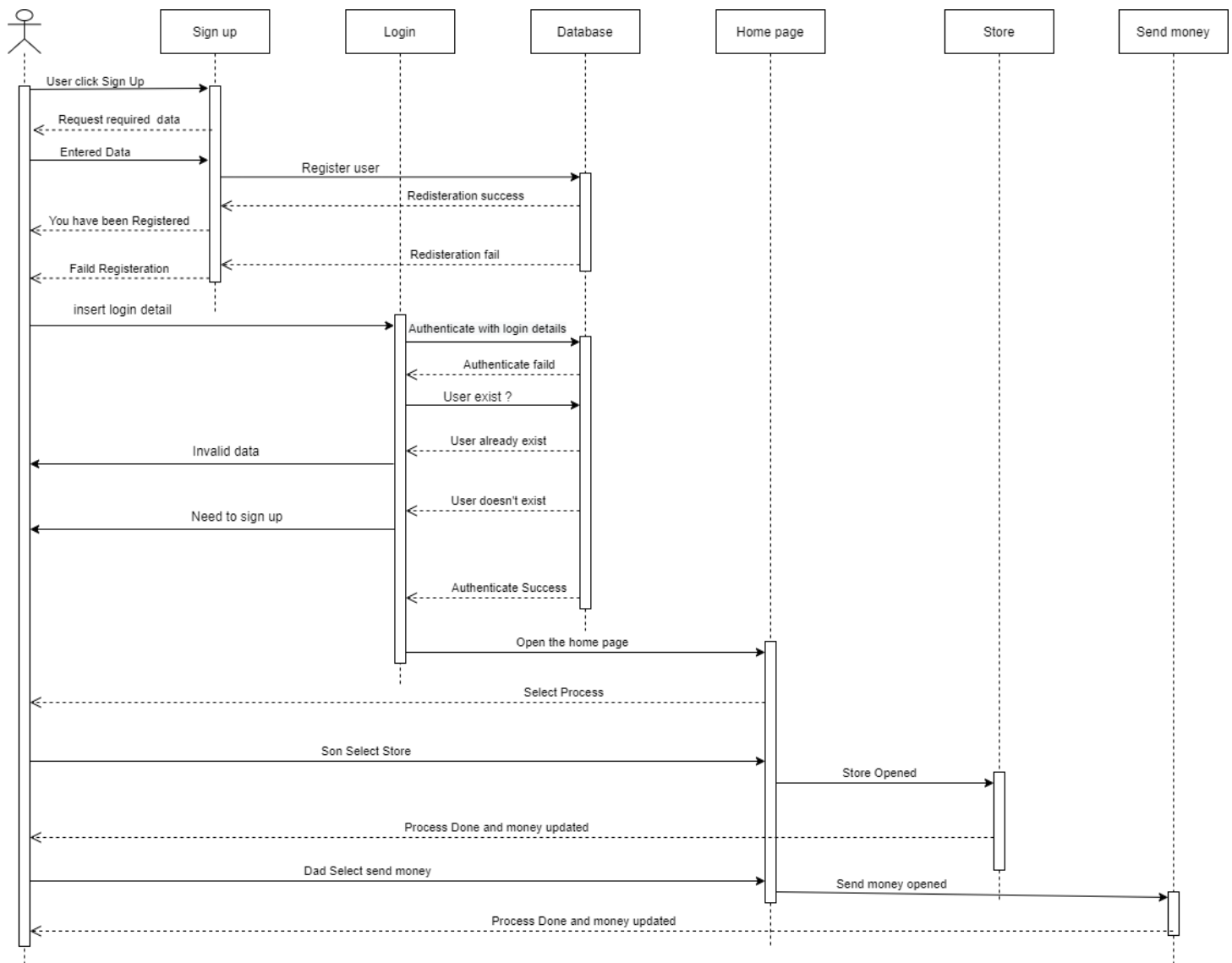


## 2.2.5 Sequence Diagram:

A sequence diagram is also a Unified Modelling Language diagram that illustrates the sequence of messages between objects in an interaction. A sequence diagram consists of a group of objects that are represented by lifelines, and the messages that they exchange over time during the interaction. It also can show the control structures between objects.

### 2.2.5.1 Sequence Diagram- User Side:

Figure 6: Sequence Diagram- User Side

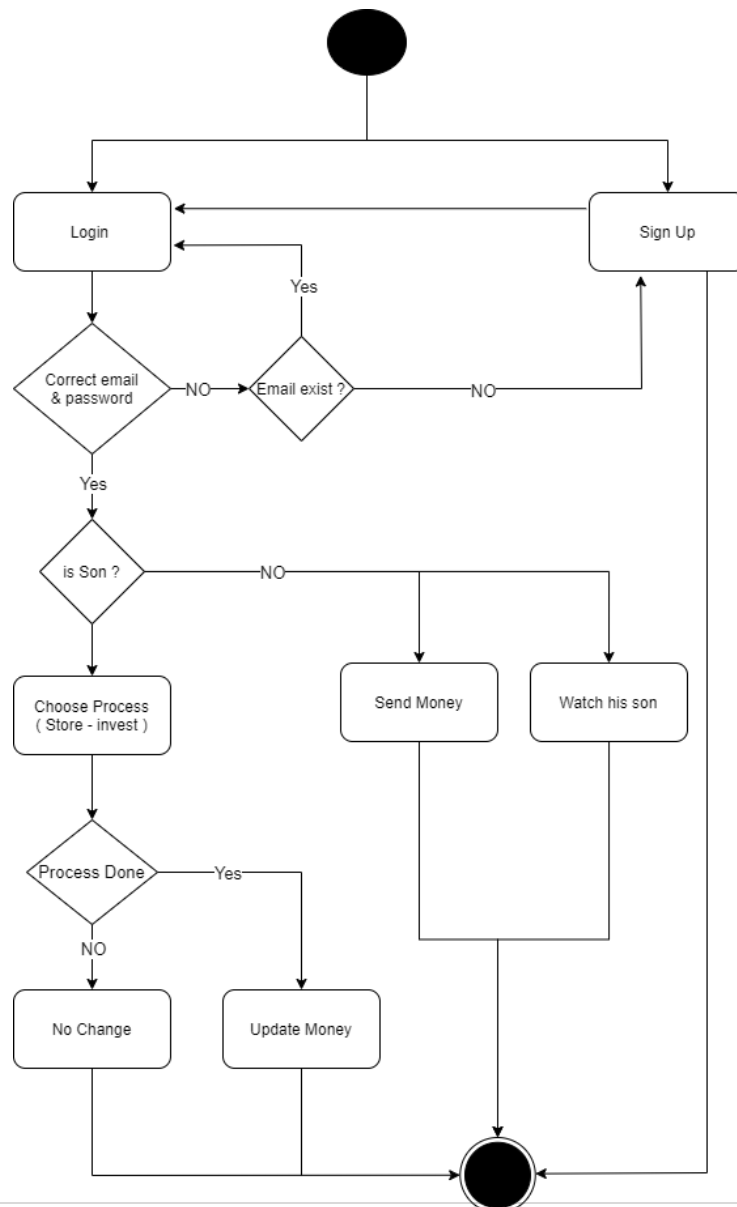


## 2.2.6 Activity Diagram:

Activity diagram provides a view of the behaviour of our system by describing the sequence of actions in the processes. Activity diagram is essentially an advanced version of flow chart because they show the flow between the actions in an activity; however, activity diagrams can also show parallel or concurrent flows and alternate flows.

### 2.2.6.1 Activity Diagram- All System:

Figure 7: Activity Diagram - All System

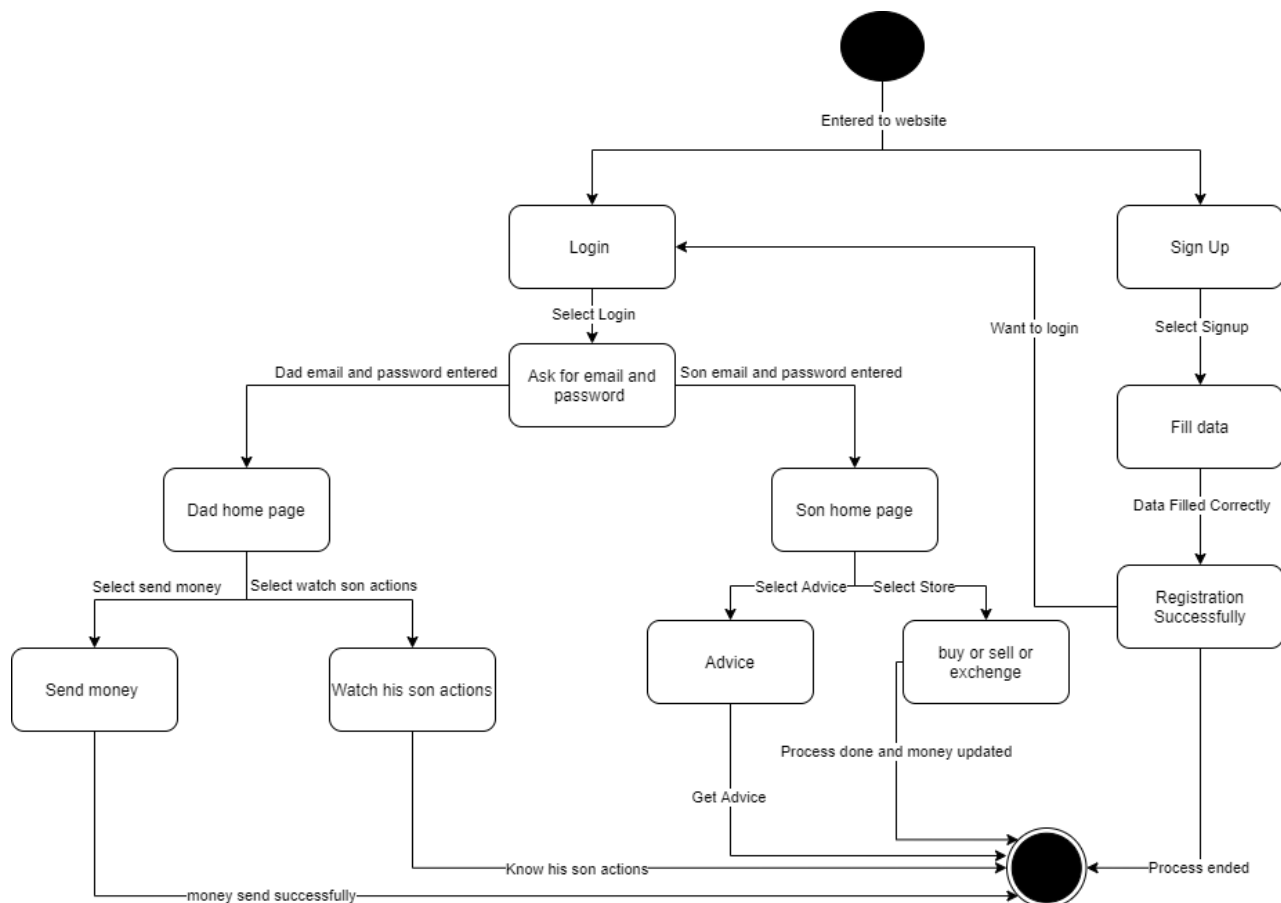


## 2.2.7 State Diagram:

State diagram is a diagram used in computer science to represent the condition of the system or part of the system and to give an abstract description of the behaviour of a system. This behaviour is analysed and represented by a series of events that can occur in one or more possible states. Each diagram represents objects and tracks the various states of these objects throughout the system.

### 2.2.7.1 State Diagram- All System:

Figure 8: State Diagram- All System

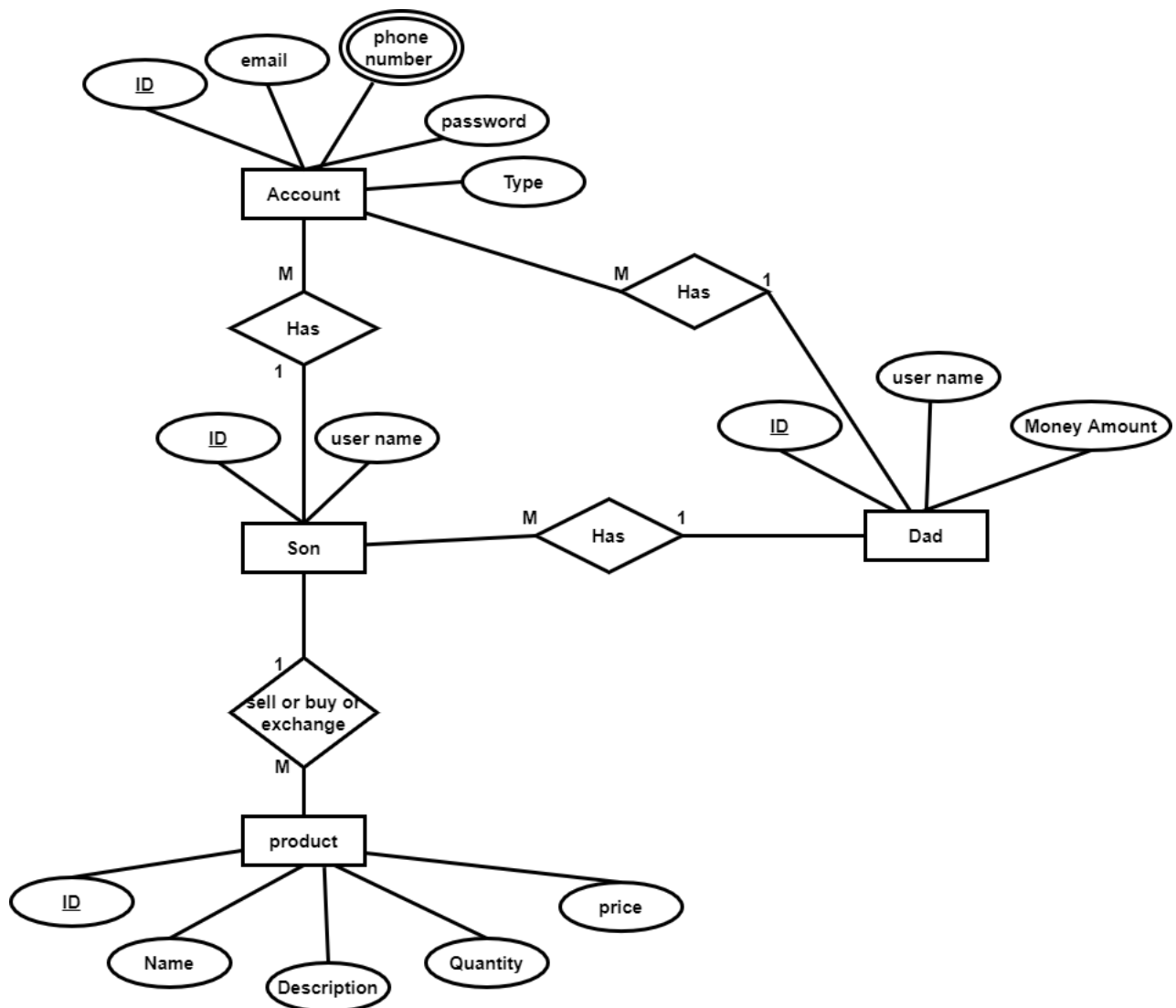


## 2.2.8 ER-Diagram:

ERD visualizes the relationships between entities like people, things, or concepts in a database. It also visualizes the attributes of these entities. By defining the entities, their attributes, and showing the relationships between them, an ER diagram can illustrate the logical structure of databases.

### 2.2.8.1 ER-Diagram:

Figure 9: ER-Diagram



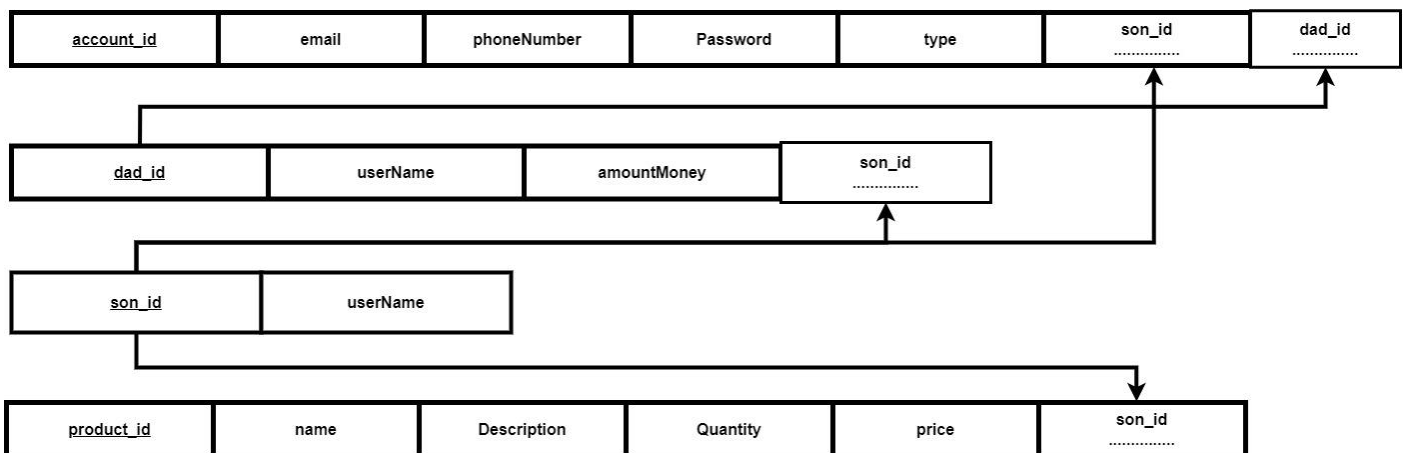


## 2.2.9 Mapping:

Data mapping is the process of matching fields from one database to another. It facilitates data migration, data integration, and other data management tasks. Before data can be analyzed for business insights, it must be homogenized in a way that makes it accessible to decision makers.

### 2.2.9.1 Mapping:

Figure 10: Database Mapping



## 2.3 Design

### 2.3.1 Web Design:

edu**kids**

### sign up

if you're dad your enter your email


user name

email

password

forget password click here

edu kids

home store 

### Features at edu kids



#### advices

you will learn more about business



#### store

in the store you can buy and sell and go business




#### books



you will reading some business books




#### games


you will play edu games

**edu kids**homestore


**Wallet : \$1774**all






**Apple**  
Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
**price : 450\$**  
[buy](#)




**bed**  
Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
**price : 180\$**  
[buy](#)




**physiotherapy**  
Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
**price : 25\$**  
[buy](#)



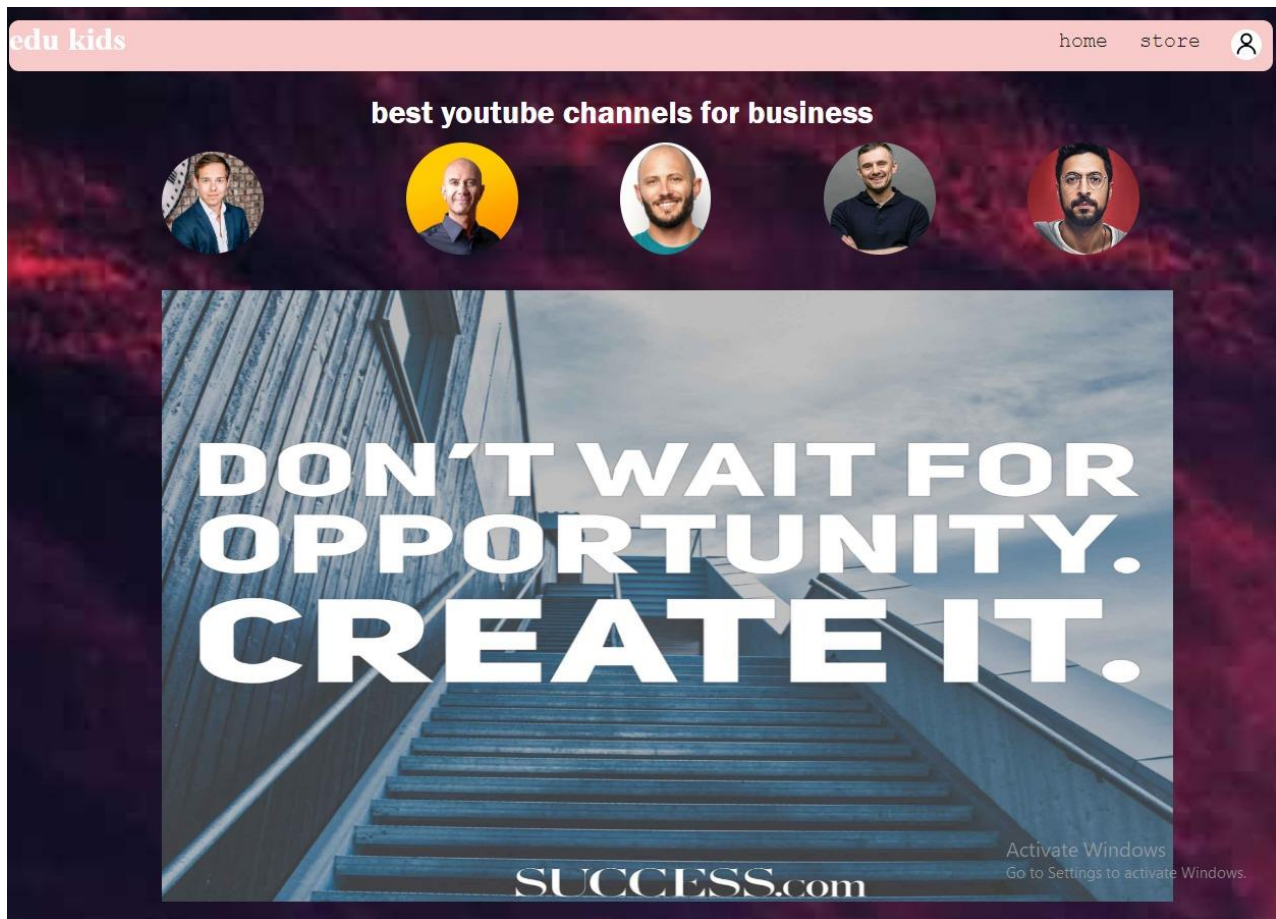
Activate Windows  
Go to Settings to activate Windows.

**edu kids****Wallet : \$1774**homestore

**mac book pro**

**mac labtop pro**  
**storege : 256gb**  
**rams : 12 rams**  
**screen : 2 k**

Activate Windows  
Go to Settings to activate Windows.



## 2.4 Tools:



VS Code:

Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control.



GIT:

Git is a DevOps tool used for source code management. It is a free and open-source version control system used to handle small to very large projects efficiently.



Draw.io:

draw.io is a free diagramming application that allows users to create and share diagrams within a web browser.