

PROJECT REPORT  
(Review III report)

On  
**WORDSMITH**

Submitted by

Aaron K Mathew 20BCE0788  
Athul A 20BCT0288  
Abhinav Bijith 20BCE2149  
Anshuman Gupta 20BCE2119  
Karthik Remesh 20BCE2361

For

**Software Engineering**

**CSE3001**

**Slot: B1, J Component**

**B.Tech. in Computer Science and Engineering**



**VIT<sup>®</sup>**  
**Vellore Institute of Technology**  
(Deemed to be University under section 3 of UGC Act, 1956)

**November 2022**

# **Table of Contents**

## Chapter 1 Introduction to Project

- 1.1 Introduction
- 1.2 Problem Definition
- 1.3 Project Scope
  - 1.3.1 Scope of the project
- 1.4 Motivation
- 1.5 Background Study/Literature Survey
- 1.6 SDLC approach used to develop project

## Chapter 2 Project Planning

- 2.1 Project Schedule
- 2.2 Effort and Resource Estimation

## Chapter 3 Requirement Gathering and Analysis

- 3.1 SRS
  - 3.1.1 Introduction
  - 3.1.2 External Interface Requirements
  - 3.1.3 System Features
  - 3.1.4 Other Nonfunctional Requirements
  - 3.1.5 Other Requirements
- 3.2 Data Modelling
- 3.3 Structural Analysis

## Chapter 4 Designs

- 4.1 UML Designs
  - 4.1.1 Structural Diagrams
    - 4.1.1.1 Class Diagram
    - 4.1.1.2 Object Diagram
    - 4.1.1.3 Component Diagram
    - 4.1.1.4 Deployment Diagram
  - 4.1.2 Behavioural Diagrams
    - 4.1.2.1 Use case diagram
    - 4.1.2.2 Sequence diagram
    - 4.1.2.3 Collaboration diagram

4.1.2.4 Statechart diagram

4.1.2.5 Activity diagram

## 4.2 Algorithms and FlowCharts

## Chapter 5 Development

5.1 Tools Description and Development approach used

5.1.1 Figma

5.1.2 Node.js

5.1.3 Electron

5.1.4 Discord

5.2 Pseudocodes of Important Modules.

5.2.1 Login module

5.2.2 Register module

5.2.3 Store topic module

5.2.4 Get topics Module

5.2.5 Store words module

5.2.6 Get words module

## Chapter 6 Testing

6.1 Test cases for Important Modules

6.1.1 Login module

6.1.2 Registration module

6.1.3 Input validation module

## Chapter 7 Screenshots of Developed Product

## Annexures

1. Review I Presentation

2. Review II Presentation

3. Review III Presentation

## **List of Figures**

- fig 1.1 : Work breakdown structure
- fig 1.2 : SDLC Model (Waterfall model)
- fig 2.1 : Project scheduling
- fig 2.2 : Task Usage
- fig 2.3 : List of resources
- fig 2.4 : Resource Distribution
- fig 2.5 : Estimation Report
- fig 3.1.1 : Level 0 DFD
- fig 3.1.2 : Level 0 DFD (with server)
- fig 3.2 : Level 1 DFD
- fig 3.3 : Level 2 DFD
- fig 3.4 : ER diagram
- fig 4.1 : Class diagram
- fig 4.2 : Object diagram
- fig 4.3 : Component diagram
- fig 4.4 : Deployment diagram
- fig 4.5 : Use case diagram
- fig 4.6 : Sequence diagram
- fig 4.7 : Collaboration diagram
- fig 4.8 : State Chart diagram
- fig 4.9 : Activity diagram
- fig 4.10 : Flowchart
- fig 7.1 : Registration page
- fig 7.2 : Login page
- fig 7.3 : home page
- fig 7.4 : words page

## **List of Tables**

table 1.1 : Project Scope

table 1.2 : Literature Review

table 2.1 : Cocomo Values

table 6.1 : Test case for Login Module

table 6.2 : Test case for Registration Module

table 6.3 : Test case for Input Validation

# **Chapter 1**

## **INTRODUCTION TO PROJECT**

### **1.1 INTRODUCTION**

In today's world, good speaking skills in English are sought after by many, be it in the education sector, in job interviews or just for general conversations. An extensive vocabulary can help pave the path to success for many by enhancing one's speaking skills and enriching their grasp on the language. Therefore, many seek to learn vocabulary and improve their existing knowledge, leading to potential capitalizing on this customer need.

Our project, 'WordSmith' aims to do just that with its goal to be a standalone application that serves as a learning platform for those seeking to build up their vocabulary. WordSmith will provide them with the option to select key topics in which they want to learn terms and improve their vocabulary on, including technical topics such as 'Artificial Intelligence' or 'Cryptocurrency'. Our project will make use of APIs to help filter relevant words sought after by the user. The user can gauge their progress and select words and topics accordingly. Our project utilises the Waterfall Model in which the outcome of one phase acts as the input for the next phase sequentially. Hence, our project has the societal impact of being a vocabulary builder helping people become fluent in their English and helping in preparing them for important interviews and exams where their English proficiency is checked.

### **1.2 Problem Definition**

Many people struggle with their vocabulary in the English language. This problem may cause them issues in their professional career in exams and interviews. We with "Wordsmith" aim to tackle this issue and make our user's vocabulary one step at a time, by helping them improve their vocabulary in their selected topics.

## **1.3 Project Scope**

Project scope is the part of project planning that involves determining and documenting a list of specific project goals, deliverables, tasks, costs and deadlines. The documentation of a project's scope is called a scope statement or terms of reference. It explains the boundaries of the project, establishes responsibilities for each team member and sets up procedures for how completed work will be verified and approved.

### **1.3.1 Scope of the project:**

- A standalone software that will help users improve their vocabulary.
- It helps users prepare for major exams like GRE.
- It introduces users to words that are related to his/her interest.
- It also keeps repeating words which he/she had missed earlier, so that they learn it thoroughly.

### **The software will be able to:**

- Suggest words.
- Repeatedly display difficult or missed words. The software will be deployed as a standalone application.

<b>Project Name</b>	WordSmith
---------------------	-----------

<b>Project Sponsor</b>	-	Project Manager	Group 14
<b>Date of Project Approval</b>	08-August-2022	Last Revision Date	-
<b>Scope Description</b>	<p><b>IN SCOPE:</b></p> <ul style="list-style-type: none"> <li>1) WordSmith is a standalone application that helps build one's vocabulary.</li> <li>2) We would require 1-2 UI/UX designers to design user friendly and an easily navigable application.</li> <li>3) We would require 1-2 Frontend and Backend Developers for the development process</li> <li>4) We would require 2 front end and 2 backend developers for the development phase</li> <li>5) We would be required to be proficient in electron js for the development process.</li> <li>6) WordSmith would help people improve in English. It also helps one prepare for exams such as GRE.</li> <li>7) A user manual would be required</li> </ul> <p><b>The software will able to perform :</b></p> <ul style="list-style-type: none"> <li>1) Filtration of words based on the interest of the user.</li> <li>2) A cross compatible standalone desktop app which can deployed in windows/linux/mac. The users would be provided with a .exe file.</li> </ul>		

	<p>3) Software will be managed purely by the client and there will be database(remote) at the backend</p> <p><b>OUT OF SCOPE:</b></p> <ul style="list-style-type: none"> <li>• Financial Investments</li> <li>• Out sourcing</li> <li>• Advanced Functionalities.</li> <li>• Pronunciation of the word(audio)</li> <li>• Supporting different</li> <li>• Having online competition with your friends through the app</li> <li>• A leaderboard</li> <li>• Exam mode – to help you prepare for exams such as GRE</li> <li>• Gamification</li> <li>• Supporting different languages</li> </ul>
<b>Project Deliverables</b>	1) WordSmith 2) Source Code 3) User Manual 4) Supporting Documents 5) Software System
<b>Acceptance Criteria</b>	1) Demo work 2) Availability of required Documents
<b>Constraints</b>	1) Time – Getting the work done in the short span of this semester. 2) lack of availability of open-source code 3) Obtaining the correct API/ Datasets that we would use 4) Expertise – Our team is not well versed with electron JS.
<b>Assumptions</b>	1) Requirements of our systems match the requirements. 2) Correct choice of ML algorithm. 3) People assigned different roles would adhere to their work 4) End users will be available for testing 5) The project scope will not change once the stakeholders sign off on the scope statement 6) Project will follow waterfall methodology throughout execution

Table 1.1 Project Scope

# Work Breakdown Structure : Process driven

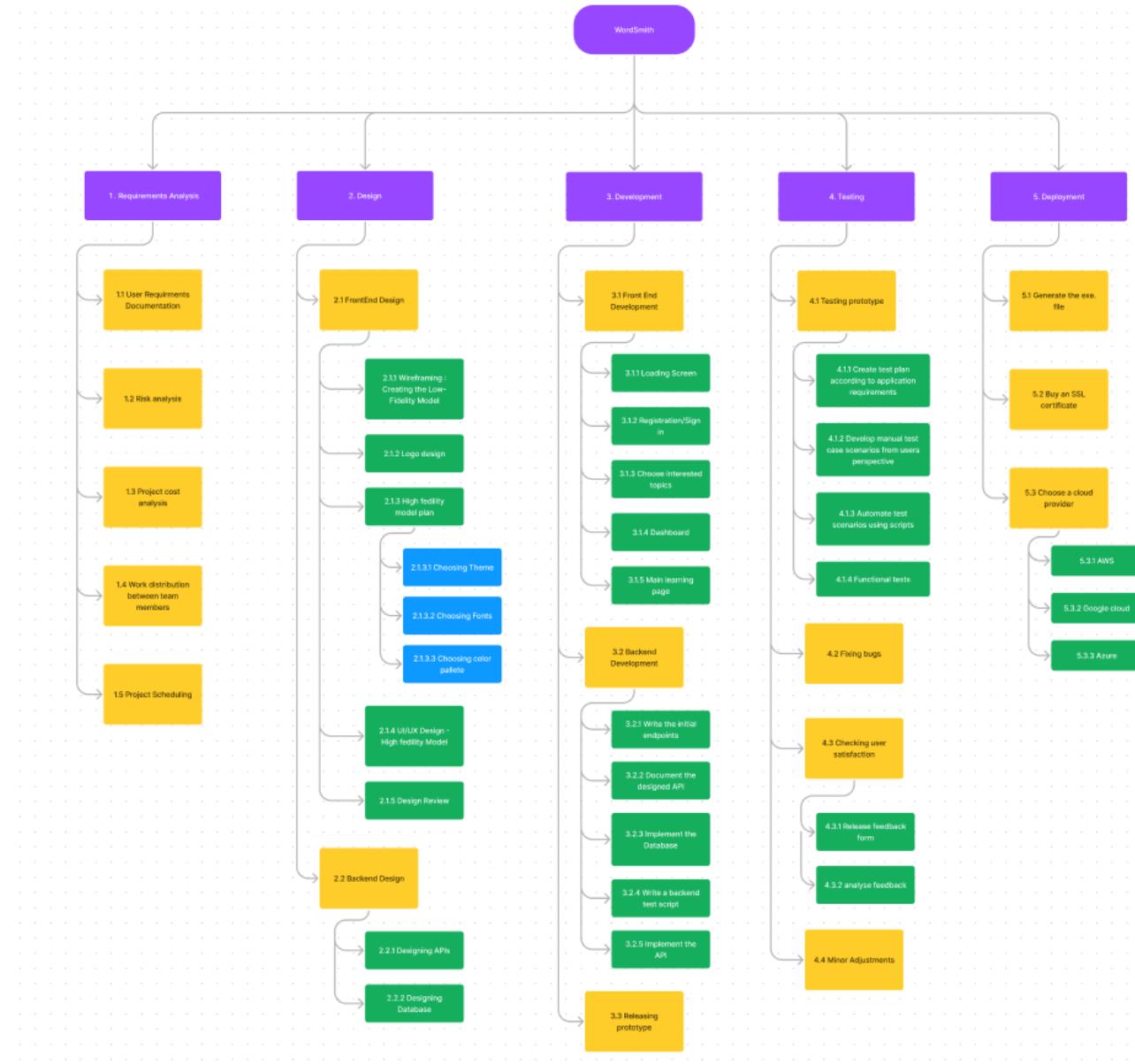


Fig: 1.1 Work Breakdown Structure

For the given project we are using a process based WBS to clearly define the processes involved. A process-oriented WBS defines a project in terms of steps, work phases, or functions. This type of WBS is focused on the steps that need to be taken within individual disciplines to complete a project and typically phrases individual elements in verb form.

## 1.4 Motivation

The motivation behind creating this application is to help the users improve their English vocabulary and in turn make them proficient in English language.

## 1.5 Background Study/Literature Survey

Paper Title Journal Details	Method	Challenges	Observations
Personalized mobile English vocabulary learning system  By: Chih-Ming Chen and Ching – Ju Chung (2007) Graduate Institute of Library, Information and Archival Studies, National Chengchi University, No. 64, Sec. 2, ZhiNan Rd., Wenshan District, Taipei City 116, Taiwan, ROC	The method applied in this was based on item response theory and learning memory cycle, by which appropriate English vocabulary will be suggested based on an individual's vocabulary ability and memory cycle. This aimed to be a flexible way to help polish one's vocabulary skills.	Method while applicable in concept didn't prove to be as effective for a larger audience and more efficient as needed to properly help one grasp the knowledge required. Technology gap detected.	Clearly, there is merit in the idea of building a vocabulary builder suited to an individual's brain patterns and schedule, if there is a sufficient dataset to efficiently provide the results.
The Effect of Blogging on Vocabulary Enhancement and Structural Accuracy in an EFL Context	Weblog was applied in language learning as a tool for vocabulary enhancement and grammatical accuracy of Iranian foreign language learners. The study was conducted over five-week	Improvement in vocabulary was significant, but slow and entirely too dependent on factors and providing external stimuli as motivation to keep	One can observe how peer feedback among other individuals when learning together can help stimulate a healthier learning environment and serve as motivation for more efficient learning. Therefore, the

By: Ramin Rahmany, Bahador Sadeghi and Sajad Faramarzi	<p>period using twenty-five students. The students were instructed to write five articles, from predetermined topics and to post them on their weblog. This study used Chi-square and Wilcoxon Signed ranks tests, the number of errors and suggested words as a way to analyse the results. The results showed that there was “significant increase in the number of words students recommend to each other and the number of grammatical errors decreased dramatically during the peer feedback”.</p>	<p>students continue to improve, without merits of its own that can constitute a learning process. The methods used to analyse the results aren’t as efficient either.</p>	<p>concept of leaderboard and progress tracker in a potential vocabulary builder application has merit.</p>
Exploring the relationship between self the relationship vocabulary learning and web-based collaboration  By: Sarah Hsueh Jui Liu, Yu Ju Lan and Cloudia Ya Yu Ho	<p>Research focused on whether the use of web - based collaboration tools such as a google docs can influence vocabulary improvement of learners.</p> <p>Two hundred and ten students from Taiwan participated. They undertook a vocabulary test to assess their level.</p> <p>After the test students were assigned to small groups to practice vocabulary learning by reading texts</p>	<p>While promising for a small test case of users, not a viable option for a large sample size.</p> <p>Google Docs can help serve as a “classroom” project to help accumulate vocabulary skills rather than a large-scale web application usage.</p> <p>Offers less flexibility to the users to learn</p>	<p>In this study, learners exhibited more equal opportunities when expressing their thoughts in the electronic modality rather than face to face.</p> <p>This suggests more efficiency when teaching through an electronic medium rather than a face-to-face medium.</p>

	<p>on Google Docs. They were instructed to provide answers to the questions and comment on Google docs in English. They were encouraged to collaborate with group members and at the end of an eighteen-week period the students took a post-test. Between the pre-test and the post-test there was significant increase in student's ability. Eighty percent of the students who participated thought that using Google Docs increased their English vocabulary ability. The study showed that through the use of Google Docs student were able to significantly increase their vocabulary.</p>		
An Efficient Minimum Vocabulary Construction Algorithm for Language Modeling  By: Sina Lin, Zengchang Qin, Zehua Huang & Tao Wan	<p>In this paper, an algorithm is proposed to construct the Minimum Vocabulary by using word frequency information. Minimum vocabulary can be defined as a certain set of basic terms and words frequently used in a language that directly corresponds to it.</p>	<ul style="list-style-type: none"> <li>- Computational complexities exist, making it a bit too complex to compute</li> <li>- Dataset won't include every term, therefore access to</li> </ul>	<p>The minimum vocabulary dataset concept can be applied to certain topics we broadly hope to cover in our application such as Artificial Intelligence or Cryptocurrency, so can have a basic set of terms related to those topics ready to learn, helping us broadly cover many smaller topics.</p>

	<p>The algorithm proposed plans to find or define a vocabulary set that can recursively define the entire vocabulary in one dictionary. By transforming the whole dictionary into a functional space, one can model word-definition relations as a functional dependency of graphs. The problem is then simplified to seek the minimum closure of a set that shares the common strategy with calculating candidate key of a relational database.</p> <p>Proposed therefore is a fast approximation by combining each word with the prior knowledge of the word frequency and iteratively calculating the basic word set to construct the minimum Vocabulary.</p>	<p>specific terms for specific topics will not be available limiting the options one has.</p>	
<p>Modeling Vocabulary for Big Code Machine Learning By: Hlib Babii, Andrea Janes and Romain Robbes</p>	<p>The language model used in this study estimates the probabilities of sequences of words based on a training corpus. In Natural Language Processing, these models are used in</p>	<p>Too complex to carry out Vocabulary must be known in advance and built out using a training corpus</p>	<p>The study provides insights into how effective machine learning algorithms can be to specifically handle large vocabularies, and despite they being complex to carry out can pave the way for an</p>

Free University of Bozen-Bolzano Bozner-Bolzano, Italy	<p>tasks such as speech recognition or machine translation. An N-gram language model is used traditionally, this study introduces as part of pre-processing, conversion of words to vector representations via one-hot-encoding, producing (sparse) vectors of length equal to the vocabulary. These NLMs convert these to word embeddings, dense word vectors of much smaller dimensions (usually in the hundreds), in their first layer. Given enough training data, words that are close in this vector space are semantically similar, and some arithmetic operations are semantically meaningful</p>		API to be used instead to carry out a similar function
---	---	--	--

Table 1.2 Literature Review

## 1.6 SDLC Approach used to develop project

The SDLC Approach we have used here is the Waterfall model. The waterfall model is suitable here for our project because the requirements are specified and the project is not too big and complex.

It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

The Waterfall model is the earliest SDLC approach that was used for software development.

The waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. In this waterfall model, the phases do not overlap.

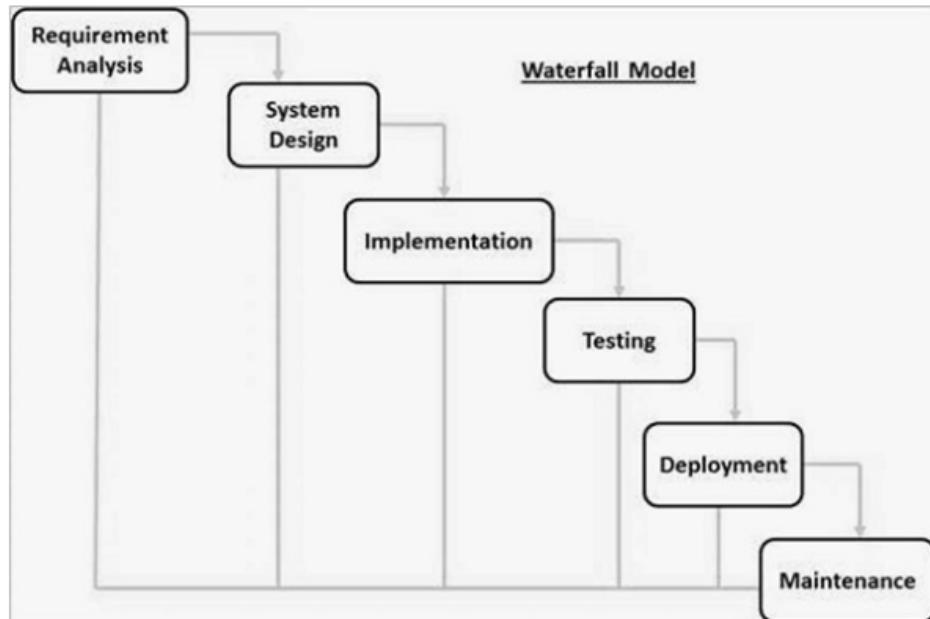


Fig 1.2 Waterfall Model

### Why did we use this model ?

- Application is not complicated and big
- Project is short
- Requirement is clear
- Resources are available and trained

### Advantages

- Before the next phase of development, each phase must be completed
- Suited for smaller projects where requirements are well defined
- They should perform quality assurance test (Verification and Validation) before completing each stage
- Elaborate documentation is done at every phase of the software's development cycle

### Disadvantages

- Error can be fixed only during the phase
- Testing period comes quite late in the developmental process

# CHAPTER 2

## PROJECT PLANNING

### 2.1 Project Schedule

Project Gantt Chart showing tasks and dependencies from August 2022 to March 2023.

**Legend:**

- Green checkmark: Task completed.
- Yellow warning sign: Task in progress.
- Red error icon: Task pending or failed.
- Blue info icon: Task information.
- Green info icon: Task calendar.
- Green info icon: Task notes.

**Resource View:**

Resources	RBS	Zoom In	Zoom Out	Copy	Cut	Insert	Delete	Indent	Outdent	Information	Find
Clipboard	Paste										
Views											
Resource											

**Task Details:**

Task ID	Name	Duration	Start	Finish	Predecessors
1	WordSmith	71 days	10/8/22 8:00 AM	14/11/22 5:00 PM	
2	1. Requirements and Planning	22 days	8/8/22 8:00 AM	6/9/22 5:00 PM	
3	1.1 User Requirements documentation and GUI proposal	1 day	8/8/22 8:00 AM	8/8/22 5:00 PM	
4	1.2 Project plan and risk analysis	2 days	10/8/22 8:00 AM	12/8/22 5:00 PM	3
5	1.3 Project cost analysis	1 day	10/8/22 8:00 AM	10/8/22 5:00 PM	3
6	1.4 Work distribution between team members	1 day	11/8/22 8:00 AM	11/8/22 5:00 PM	5
7	1.5 Project Scheduling	2 days	12/8/22 8:00 AM	15/8/22 5:00 PM	6
8	2. Design	11 days	16/8/22 8:00 AM	30/8/22 5:00 PM	
9	2.1 Wireframing - Low Fidelity model	2 days	16/8/22 8:00 AM	17/8/22 5:00 PM	7
10	2.2 Logo design and Theme selection	1 day	18/8/22 8:00 AM	18/8/22 5:00 PM	9
11	2.3 choosing fonts and color palette	1 day	19/8/22 8:00 AM	19/8/22 5:00 PM	10
12	2.4 UI/UX design - High fidelity	5 days	20/8/22 8:00 AM	26/8/22 5:00 PM	11
13	2.5 Design review	2 days	27/8/22 8:00 AM	30/8/22 5:00 PM	12
14	3. Development	40 days	29/8/22 8:00 AM	29/10/22 5:00 PM	
15	3.1 Frontend	13 days	31/8/22 8:00 AM	14/10/22 5:00 PM	
16	3.1.1 Landing Screen	1 day	31/8/22 8:00 AM	31/8/22 5:00 PM	13
17	3.1.2 Registration/ sign in	3 days	1/9/22 8:00 AM	5/9/22 5:00 PM	16
18	3.1.3 Choose interested topics	4 days	6/9/22 8:00 AM	9/9/22 5:00 PM	17
19	3.1.4 Database	4 days	9/9/22 8:00 AM	14/9/22 5:00 PM	18
20	3.1.5 Main learning page	3 days	14/9/22 8:00 AM	16/9/22 5:00 PM	19
21	3.2 Backend	25 days	17/9/22 8:00 AM	21/10/22 5:00 PM	
22	3.2.1 Write initial endpoints	4 days	18/9/22 8:00 AM	22/9/22 5:00 PM	20
23	3.2.2 Design and document the API	4 days	19/9/22 8:00 AM	28/9/22 5:00 PM	22
24	3.2.3 Design and implement the database	6 days	20/9/22 8:00 AM	6/10/22 5:00 PM	23
25	3.2.4 Backend test script	4 days	21/9/22 8:00 AM	12/10/22 5:00 PM	24
26	3.2.5 Implement the API	7 days	13/10/22 8:00 AM	20/10/22 5:00 PM	25
27	3.2.6 Test Suite	2 days	24/10/22 8:00 AM	25/10/22 5:00 PM	26

Fig 2.1 Project Schedule in project libre

## 2.2 Effort and resource estimation

Cocomo (Constructive Cost Model) is a regression model based on LOC, i.e number of Lines of Code. It is a procedural cost estimate model for software projects and is often used as a process of reliably predicting the various parameters associated with making a project such as size, effort, cost, time, and quality. It was proposed by Barry Boehm in 1981 and is based on the study of 63 projects, which makes it one of the best-documented models. The key parameters which define the quality of any software products, which are also an outcome of the Cocomo are primarily Effort & Schedule:

**Effort:** Amount of labor that will be required to complete a task. It is measured in person-months units.

**Schedule:** Simply means the amount of time required for the completion of the job, which is, of course,

proportional to the effort put in. It is measured in the units of time such as weeks, months. The model we will be using is Semi Detached as our team size, experience, knowledge of the various programming environments lie in between that of organic and Embedded. The team size consists of 5 members (some experienced and others learning),

Software Projects	a	b	c	d
Organic	2.4	1.05	2.5	0.38
Semi Detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

Table 2.1 Values for a,b,c and d for different project types

From the above table we have the values for semi detached projects as :

a= 3.0

b= 1.12

c= 2.5

d= 0.35

Therefore,

$$\text{Effort} = 3 \times (2) \times 1.12 = 6.520$$

$$\text{Time} = 2.5 \times (6.52) \times 0.35 = 4.81$$

$$\text{Person required} = \text{Effort}/\text{time} = 6.520/4.8 = 1$$

It is observed that the estimated time and the time planned in project libre are within limits of estimated error.

Task Usage:

	<i>Athul A</i>	8 hours	1 day 14/11/22 8:00 AM	14/11/22 5:00 PM	Work				
36	5.3 Choose a cloud provi	8 hours	1 day 14/11/22 8:00 AM	14/11/22 5:00 PM	Work				
	<i>Athul A</i>	8 hours	1 day 14/11/22 8:00 AM	14/11/22 5:00 PM	Work				
					Work				
					Work				
					Work				

Fig 2.2 Task Usage

## Resources :

Resources		RBS	Zoom Out	Copy	Insert	Indent	Information	Find				
	Views	Paste	Cut	Delete	Outdent	Calendar	Notes					
	Clipboard	Resource										
1	Abhinav Bijth		Work				Ab			100%	Rs. 10.00/hour	
2	Aaron K Mathew		Work				A			100%	Rs. 10.00/hour	
3	Athul A		Work				A			100%	Rs. 10.00/hour	
4	Karthik Remesh		Work				K			100%	Rs. 10.00/hour	
5	Anshuman Gupta		Work				A			100%	Rs. 10.00/hour	
6	Discord		Material				D				Rs. 0.00	
7	VS code		Material				V				Rs. 0.00	
8	Figma		Material				F				Rs. 0.00	
9	Illustrator		Material				I				Rs. 0.00	
10	Postman		Material				P				Rs. 0.00	
11	Project Libre		Material				Pr				Rs. 0.00	
12	Google Forms		Work				G			100%	Rs. 0.00/hour	

Fig 2.3: List of resources

## Resource Distribution :

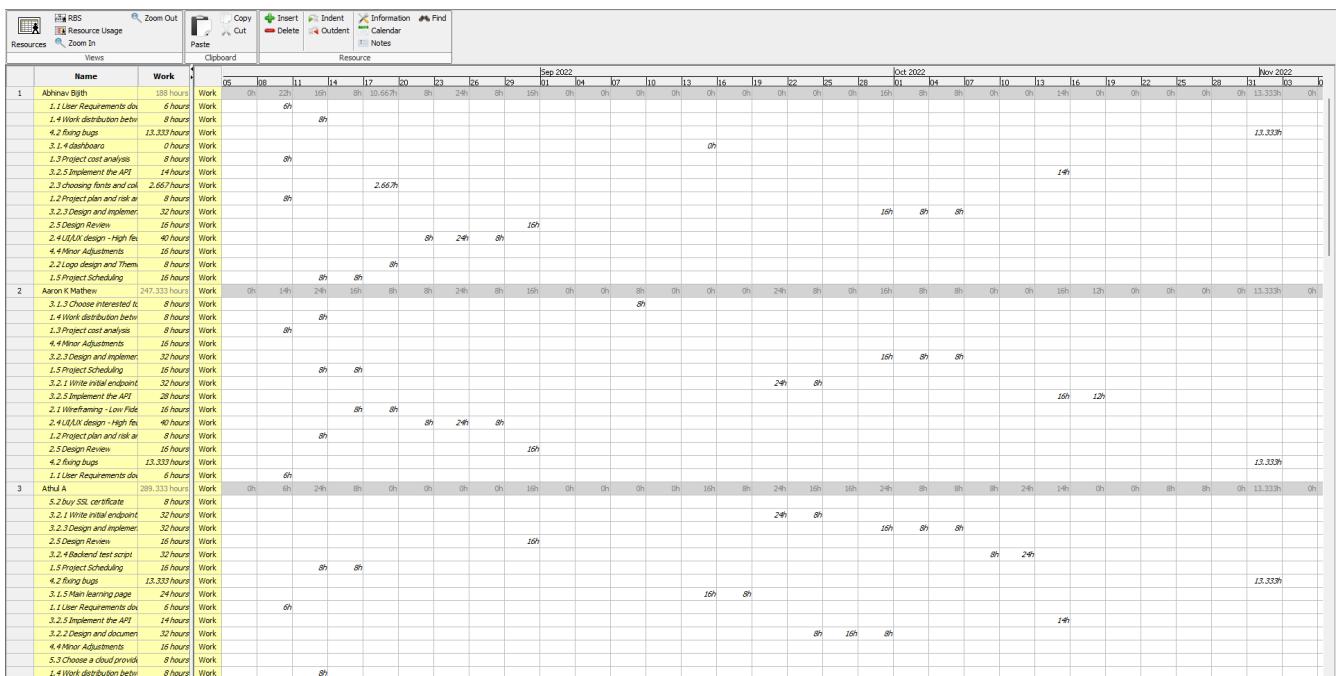
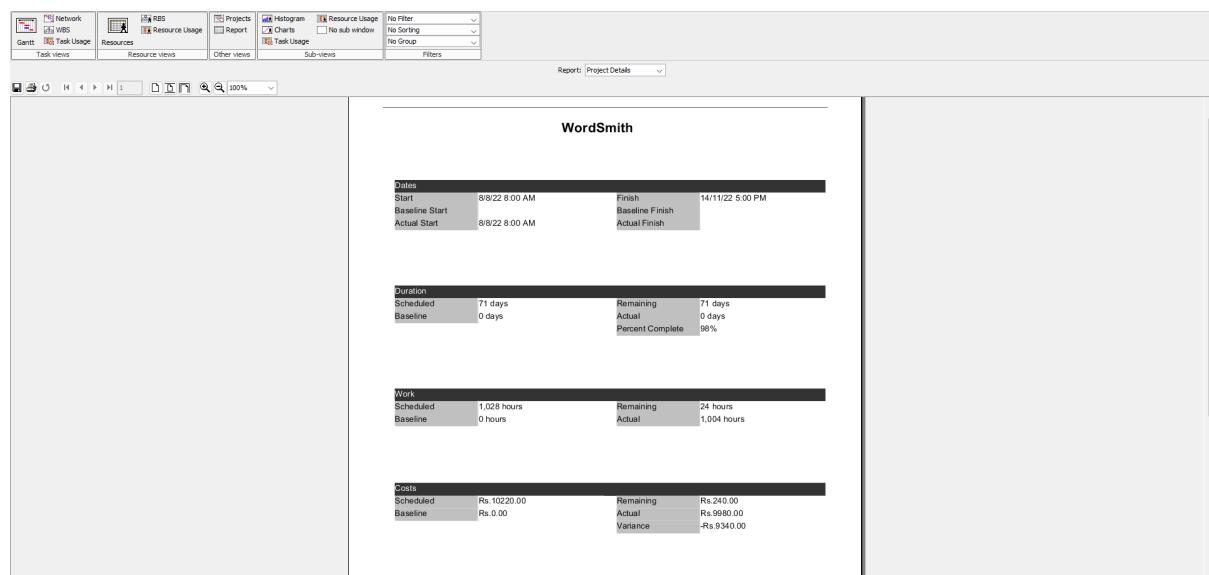


Fig 2.4: Resource Distribution

## Estimation Report :



## Fig 2.5 Estimation Report

Estimated time : 71 days

# **CHAPTER 3**

## **REQUIREMENT GATHERING AND ANALYSIS**

### **3.1 Software Requirement Specification**

#### **3.1.1 Introduction**

##### **3.1.1.1 Purpose**

The purpose of this document is to specify the software requirements For WordSmith which is a standalone application based on building vocabulary that would help one to become more fluent in said language. WordSmith would provide filtration and recommendation of words based on topics chosen by the user. This is the first release of the SRS for the same.

##### **3.1.1.2 Document Conventions**

- 1) Highlighted/Bolded format meant to convey the levity/importance of their impact Also meant to highlight an object or a point
- 2) Points divided using bullets and dashes
- 3) Font format is Times New Roman, size of content is 12. Headings are Bolded, format is Times New Roman and size is 14.

##### **3.1.1.3 Intended Audience and Reading Suggestions**

This document is intended for the developers, project managers and testers. This has been implemented under the guidance of my college professor.

##### **3.1.1.4 Product Scope**

The software will be able to perform:

- 1) Filtration of words based on the interest of the user
- 2) A Cross Compatible standalone desktop App which can be deployed in Windows/Linux/Mac. The user would be provided with an .EXE file.
- 3) Software will be managed purely by the client and there will be Database (Remote) at the Backend.

### **3.1.1.5 References**

- 1) <https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database>
- 2) [https://web.cs.dal.ca/~hawkey/3130/srs\\_template-ieee.doc](https://web.cs.dal.ca/~hawkey/3130/srs_template-ieee.doc) Overall Description

### **3.1.1.6 Product Perspective**

This is a new, self-contained project that aims to rival the likes of existing Vocabulary builders such as Duolingo in the field of topic recommendations and word search related to the topic chosen by the user to improve their vocabulary in.

### **3.1.1.7 Product Functions**

Major functions of WordSmith -

User side :

- 1) Choose Topics
- 2) Filter Through words based on difficulty
- 3) Search Topics (Based on the vocabulary one wants to pull from)

Admin side :

- 1) Insert Topic databases
- 2) Update Topic Databases
- 3) Delete Topic Databases

### **3.1.1.8 User Classes and Characteristics**

Users should be able to view , filter out and search for topics they want to expand their vocabulary in. Filtering can be based on level of difficulty and time taken to master more difficult and longer words. Searching of topics can be done according to specific terminology the user needs to master.

On searching the user would get to view a list of possible search results.

The users would be divided into 2 broad classes : Signed in and Not signed in

These would have the following functionalities :

User (Not Signed in) :

- 1) View Topics
- 2) Search for Topics
- 3) Filter Words in topics based on time and difficulty

User ( Signed in) :

- 1) All functionalities of not signed in users
- 2) Like/unlike Various topics
- 3) Enter suggestions for more words
- 4) Get access to a wider range of words ranging from easy to hard difficulty
- 5) Access to Leaderboards

Admin :

- 1) Update Topic Databases
- 2) Insert more Topics
- 3) Delete

### **3.1.1.9 Operating Environment**

Operating Environment of the Vocabulary Builder is as follows:

- Frontend: HTML, CSS and React JS

- Backend: Node Js , Express
- Database: mysql
- Operating system: Windows 11, Linux, Mac
- Client/server system

The client-side components of the software system must operate within common web browser environments using Secure Sockets Layer (SSL) / Transport Layer Security (TLS) cryptographic protocols

The minimum set of browsers that must be supported is :

- Apple Safari 7+
- Google Chrome 44+
- Microsoft Internet Explorer 10+
- Mozilla Firefox 40+

### **3.1.1.10 Design and Implementation Constraints**

- 1) The information of all users i.e., liked topics and scores must be stored in a database that is accessible by the website.
- 2) My SQL server will be used as SQL engine and database.
- 3) WordSmith is running 24 hours a day.
- 4) Users may access the application from any computer that has Internet browsing capabilities and an Internet connection.
- 5) Users must have their correct usernames and passwords to enter into their online accounts and do actions.

### **3.1.1.11 User Documentation**

A user guide will be available to help the user get familiar with the different functionalities of the website. This would help the user at each step (Regardless of whether it is his first time using the application). This will help the user understand the application easily.

### 1) Description Document

Gives a detailed overview of the product with all the services offered by the product. End-users read this document and decide if this is the product, he/she is looking for or not.

### 2) User Manual

It gives the normal functions of the product with illustrated examples. It has all the ‘How-to’ information regarding the regular use of the product.

Any user of the software system is the target audience for user documentation generated about the software system. A range of short document types (e.g., guidelines, tutorials, frequently asked questions) in HyperText Markup Language (HTML) and/or Portable Document Format (PDF) format must describe the use of the software system.

### **3.1.1.12 Assumptions and Dependencies**

- 1) Requirements of our systems matches the requirements.
- 2) Correct choice of API to be used.
- 3) People assigned different roles would adhere to their work
- 4) End users will be available for testing
- 5) The project scope will not change once the stakeholders sign off on the scope statement
- 6) Project will follow waterfall methodology throughout execution

### **3.1.2. External Interface Requirements**

#### **3.1.2.1 User Interfaces**

First off there will be a sign in page that allows the user to sign up, sign in or continue without signing in. Next, will be the Recommendations Page similar to Spotify and Twitter, where users choose their preferred topics for which they want to master vocabulary in. Next, we will have the home page where they will receive words based on the topic they have chosen and a search bar at the top would be available where users can search for other specific topics. Once the user enters the name of a topic a list of related search results is displayed from which the users can select the topic whose word database they want to follow. Apart from this, the user can also search for words specifically.

- Front-end software: React Js (Html, CSS and Js)
- Back-end software: SQL+

### **Login Interface :**

In case the user is not registered yet, he can enter the details and register. Which asks the user to type his username and password. If the user entered either his username or password incorrectly then an error message occurs.

### **Admin Side:**

This interface would allow admins to add, update or delete topic specific word databases.

### **User Side:**

Search – The user can enter the name of the topic he/she is looking for or specific keywords

Results View – View all the search results and get the topics needed.

### **3.1.2.2 Hardware Interfaces**

All server-side components must execute on server-class computers. All client-side components must execute on workstation-class and personal-class computers.

The input devices will be a mouse and a keyboard.

The hardware interfaces will include :

- Windows devices (32-bit or 64-bit)
- MacOS devices
- A browser which supports CGI, HTML & JavaScript.

### **3.1.2.3 Software Interfaces**

The interface must be able to connect to a database to store XML schema defined using XSD and data streams. Source and destination formats for data must include XML and may also include Extensible Stylesheet Language Transformation (XSLT), JavaScript Object Notation (JSON), Comma Separated Value (CSV), and American Standard Code for information interchange. The application should be supported by

Chrome, Mozilla Firefox, and Microsoft Edge.

### **3.1.2.4 Communications Interfaces**

Web Browser that it must support :

Chrome, Mozilla Firefox, and Microsoft Edge.

Communication standards and network server communication protocols :

The communication architecture must follow the client-server model. Communication between the client and server should utilize a REST-compliant web service and must be served over HTTP Secure (HTTPS). The client-server communication must be stateless. A uniform interface must separate the client roles from the server roles.

### **3.1.3 System Features**

This section of the SRS describes the requirements for the system's features. Specifically, requirements for component functionality, data curation, and software system administration are defined.

#### **3.1.3.1 Component Functional Requirements**

##### **Admin :**

###### **1) Insert Topics :**

This action is done to add new topics and its relevant words/keywords to the database.

###### **2) Delete Topics :**

This event is to delete an existing Topic and its word database from the collection.

###### **3) Modify Topics :**

This is to update the content of an existing topic with even more words/keywords in the database.

##### **Normal User :**

###### **1) Filter search results :**

Filter for Words based on time, difficulty etc.

- 2) View search results :
- 3) View the Topics that match with the search query

### **Common functions :**

- 1) Login :

Both admin and normal users must log in.

- 2) Search for a Topic :

Both admin and normal users can search for topics to either view or edit them.

### **Terminology Finder**

The following API is used to give us the related terms and words for the topic chosen.

#### **3.1.3.2 Data curation**

This subsection of the SRS describes the requirements for data curation within the software system.

- 1) The software system must provide a RESTful API for curating data in the database.
- 2) The software system must provide the data curator with the ability to add data to the database in accordance with a stored data-structure template.
- 3) The software system may provide all curation functionality via a web-based user interface.

#### **3.1.3.3 Software System Administration**

This subsection of the SRS describes the requirements for administering the software system.

#### **Description and Priority**

The software system must provide a function to manage user accounts and data-structure templates. The minimum user account types must be: (1) data explorer, (2) data curator, and (3) administrator. The data-explorer account type will grant access to end-users. The data-curator account type will grant access to a user for the purpose of storing and managing data in the database. The administrator

account type will grant “super user” access to a user for the purpose of managing the software system. The administration function of the software system is a **medium priority**.

### **3.1.4 Other Nonfunctional Requirements**

#### **3.1.4.1 Performance Requirements**

##### **A) Normalization :**

The basic objective of normalization is to reduce redundancy which means that information is to be stored only once. Storing information several times leads to wastage of storage space and increase in the total size of the data stored. If a database is not properly designed it can give rise to modification anomalies. Modification anomalies arise when data is added to, changed, or deleted from a database table. Similarly, in traditional databases as well as improperly designed relational databases, data redundancy can be a problem. These can be eliminated by normalizing a database. Normalization is the process of breaking down a table into smaller tables. So that each table deals with a single theme.

**B)** All other performance related to storage, memory, and processing should follow industry recommended practices to ensure resource requirements are minimized.

##### **3.1.4.2 Safety Requirements**

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed-up log, up to the time of failure.

##### **3.1.4.3 Security Requirements**

The software system specified in this SRS shall adhere to generally accepted industry standards for secure software development. For determining the access-level requirements of the software system and its associated services, the software development process must at the very least adhere to the principle of least privilege. An automated dynamic application security testing tool must successfully pass the software system's production release version.

##### **3.1.4.4 Software Quality Attributes**

Availability : The app should be available as per the users' requirements

**Management** : The administrators should be able to maintain the database of words going through the application

**Usability** : The application should be easy to use and satisfy a maximum number of customer needs.

**Testability** : The system should be easy to test and find defects. If required it should be easy to divide into different models for testing.

#### **3.1.4.5 Business Rules**

For accessing and interacting with the system, at least a Super-User role and a User role must be defined. As long as the business rules for the administrator and user roles are met, the system may define additional roles. The Super-User role must at the very least consider the requirements for the administrator, data curator, and data explorer account types

#### **3.1.5 Other Requirements**

Apart from the above-mentioned requirements there are a few other requirements that lie out of the scope covered before. These may include the following:

- 1) Easy to use - The system should be easy to use, to ensure that even inexperienced users do not have any issues while understanding and interacting with the system.
- 2) Clean interface - The interface that is developed for the project shall be intuitive and clean, allowing it to be easy to navigate and the flow should be easy to understand.
- 3) Accessible to multiple disabilities - The interface must follow the guidelines of international agencies including WWC, to ensure that the entire system is accessible to people with different disabilities.
- 4) Minimalistic UI/UX - The User Interface / User Experience should be designed with a careful minimalistic flow, to prevent the confusion caused by a cluttered interface. There should be no information fatigue.
- 5) Smooth and functional experience - The user experience should be smooth without hiccups and the functions should be clear and easy to use, without a lot of intermediate steps that act as a buffer, providing its main functionality in an instant.

6) Scalable development - The development of the project shall incorporate a set of scalable guidelines, such that as the user base of the system grows, it should be able to handle multiple users utilizing the system at the same time without significant performance issues occurring.

7) Secure user experience - The user's data and other private information should be secured by the latest web technologies, and should conform to the best data privacy standards, to ensure there are no data leaks.

8) Portability - The system should be designed with a portable usability approach in mind, allowing users to be helped even in environments where they may not have full access to their devices.

9) Flexibility - The system should be flexible enough to provide the users with a customized experience helping different people in the best way they need assistance

10) Financial Investments – To further boost the platform

11) Outsourcing – To add more features

12) Pronunciation of the word (audio) – To have audio files available for the user's use

13) Supporting different Languages – Currently English is to be used

14) Having online competition with your friends through the app

15) Exam mode – to help you prepare for exams such as GRE

## 3.2 Data modelling

Level 0 DFD without server

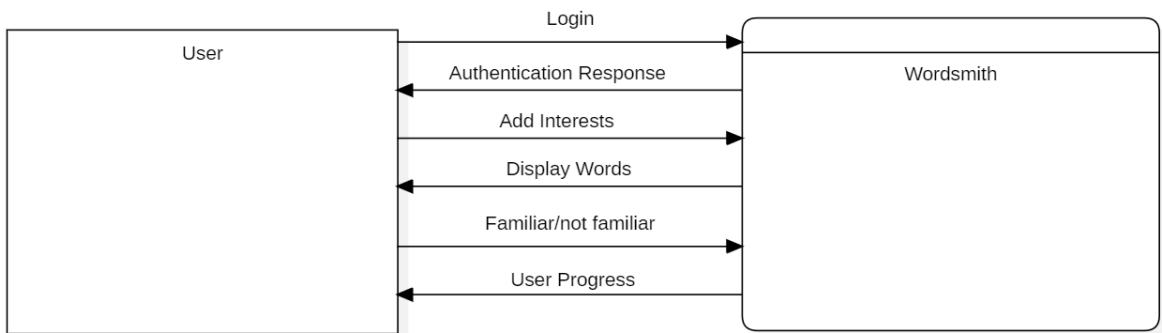


Fig 3.1.1 Level 0 DFD without server

Level 0 DFD with server

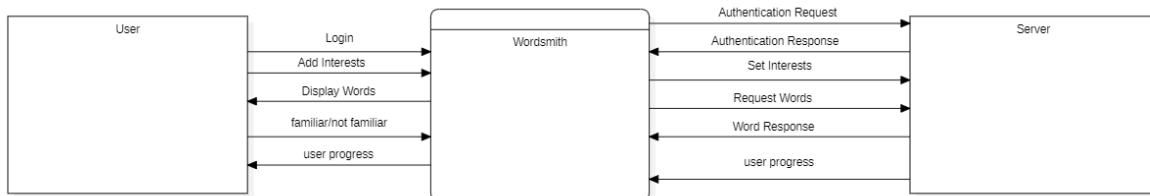


Fig 3.1.2 Level 0 DFD with server

Level 1 DFD :

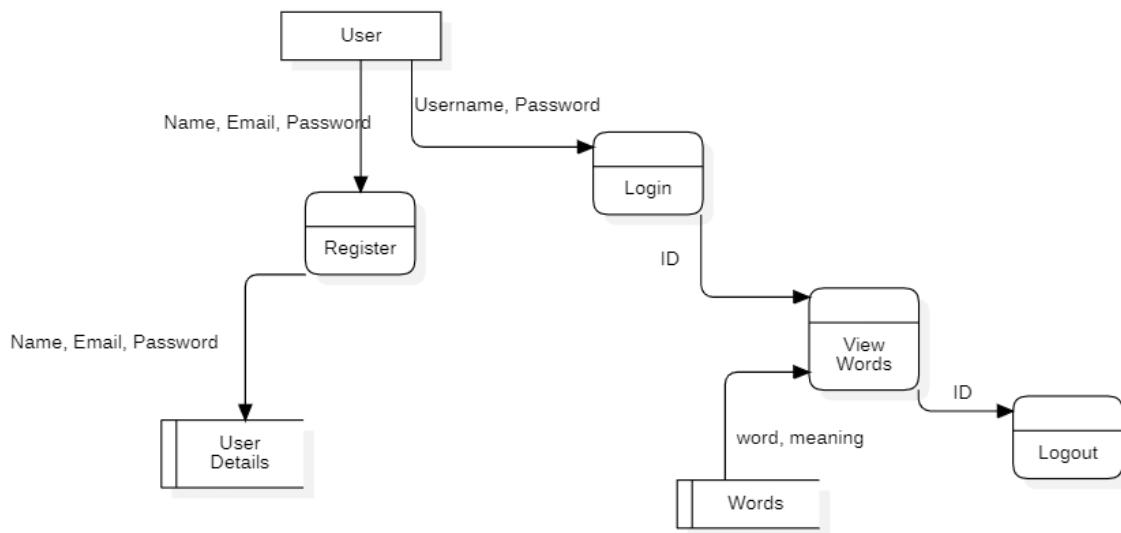


Fig 3.2 Level 1 DFD

Level 2 DFD :

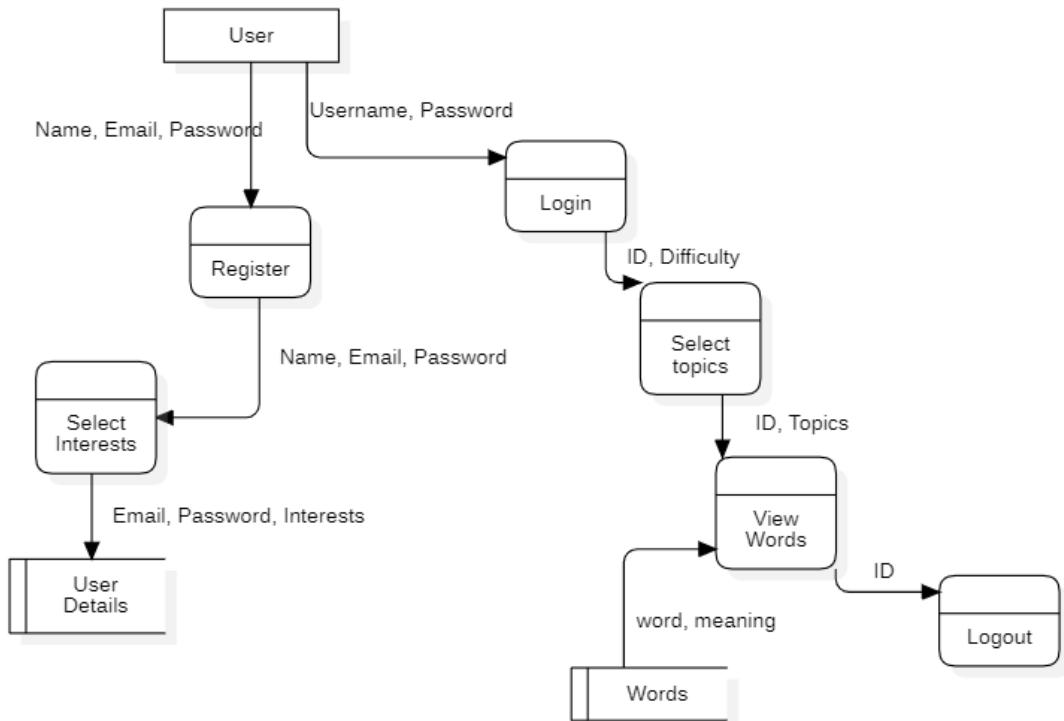


Fig 3.3 Level 2 DFD

### 3.3 Structural Analysis

ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships.

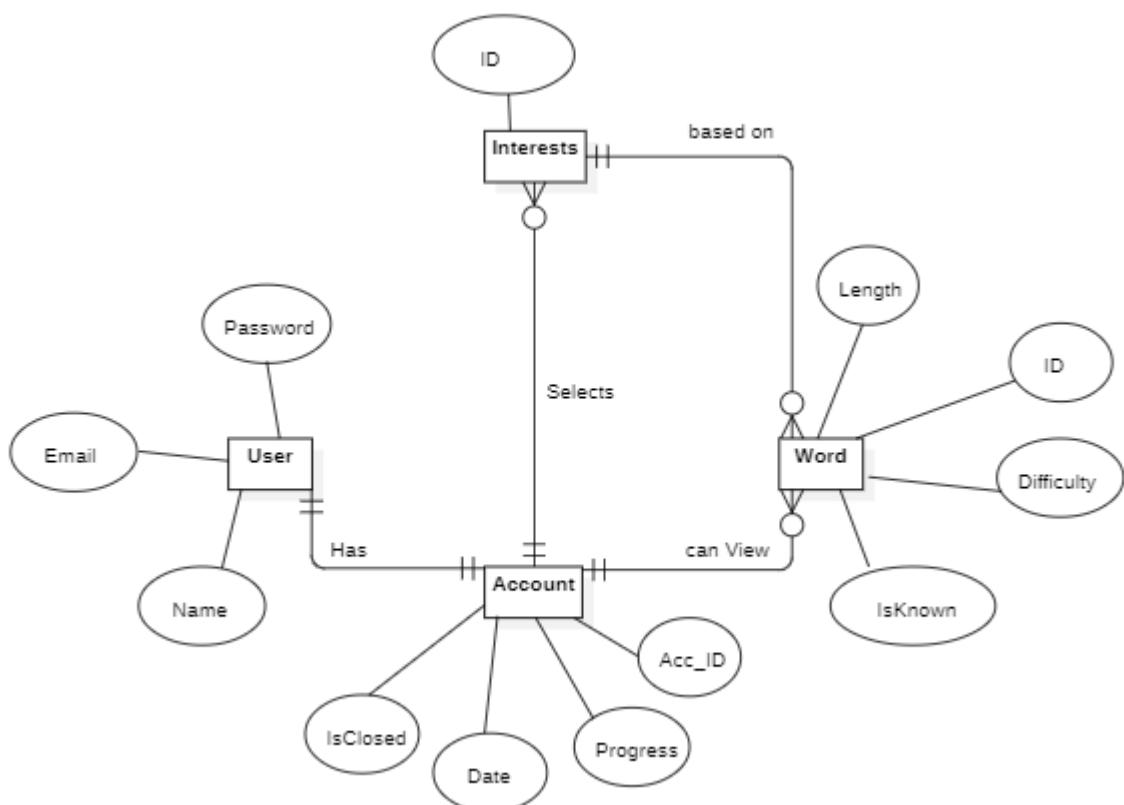


Fig 3.4 ER diagram

## CHAPTER 4

## DESIGNS

### 4.1 UML Designs

#### 4.1.1 Structural Diagrams :

The structural diagrams represent the static aspect of the system.

These static aspects represent those parts of a diagram, which forms the main structure and are therefore stable.

These static parts are represented by classes, interfaces, objects, components, and nodes. The four structural diagrams are –

1. Class diagram
2. Object diagram
3. Component diagram
4. Deployment diagram

#### 4.1.1.1 Class Diagram :

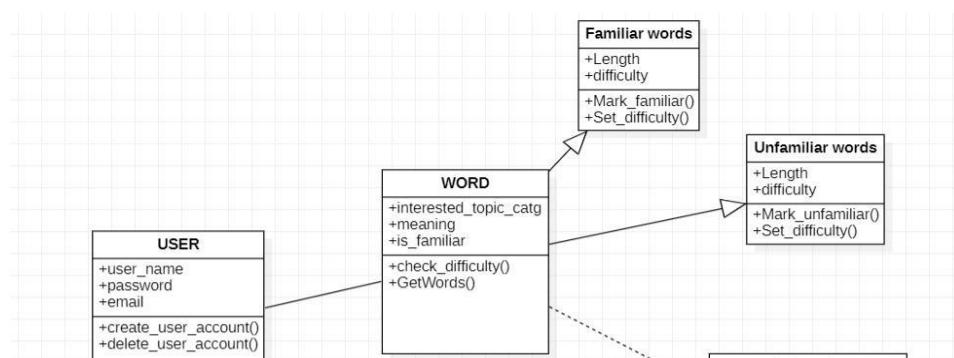


Fig 4.1 Class Diagram

#### 4.1.1.2 Object Diagram

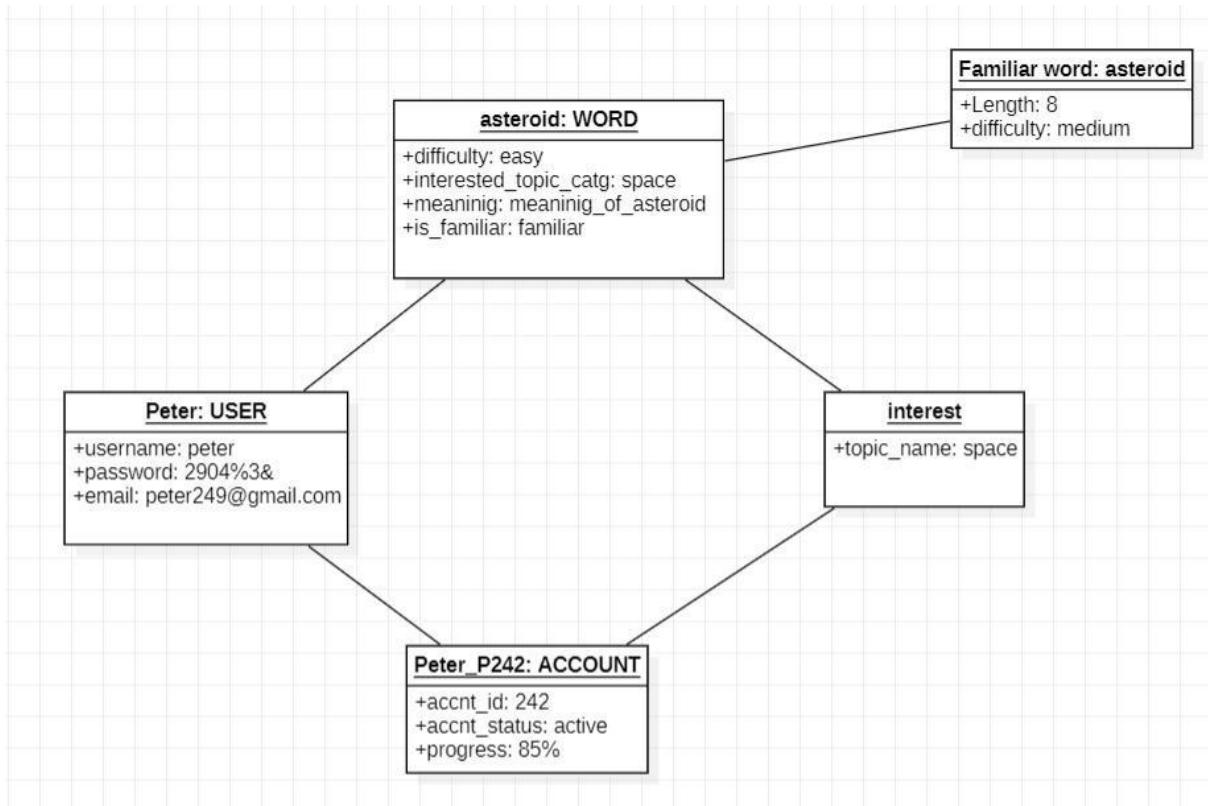


Fig 4.2 Object Diagram

#### 4.1.1.3 Component Diagram

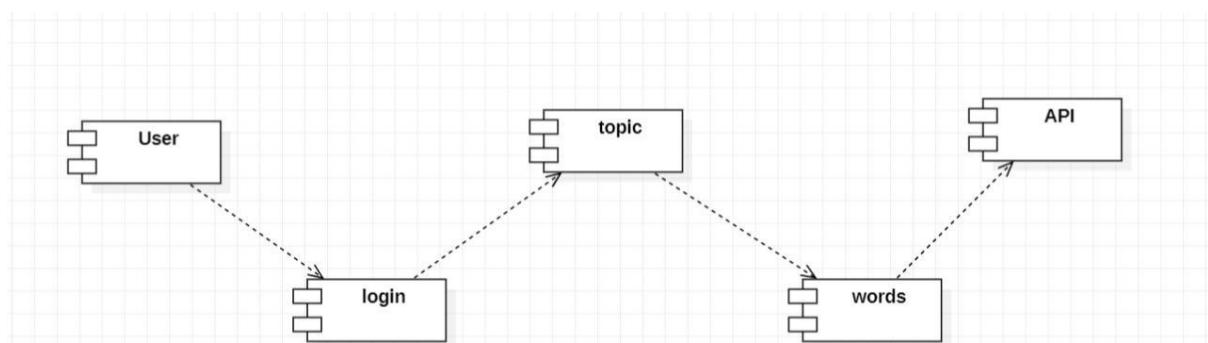


Fig 4.3 Component Diagram

#### 4.1.1.4 Deployment Diagram

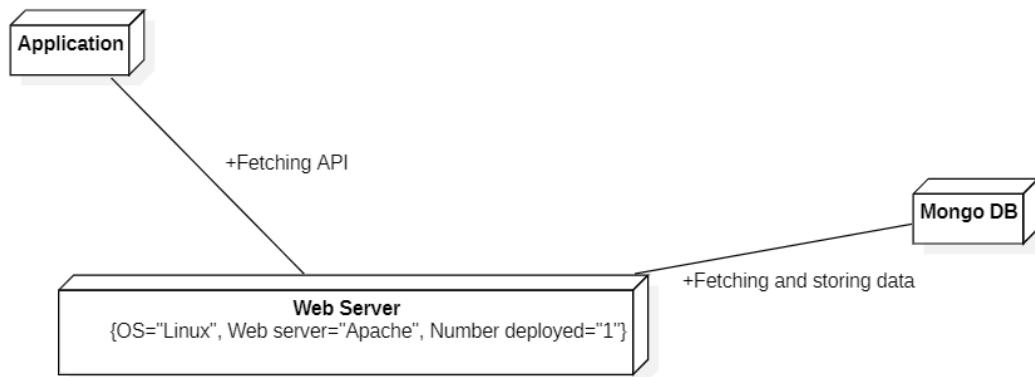


Fig 4.4 Deployment diagram

#### 4.1.2 Behavioural Diagrams :

Any system can have two aspects, static and dynamic. So, a model is considered as complete when both the aspects are fully covered.

Behavioral diagrams basically capture the dynamic aspect of a system. Dynamic aspects can be further described as the changing/moving parts of a system.

UML has the following five types of behavioral diagrams –

1. Use case diagram
2. Sequence diagram
3. Collaboration diagram
4. Statechart diagram
5. Activity diagram

#### 4.1.2.1 Use case diagram :

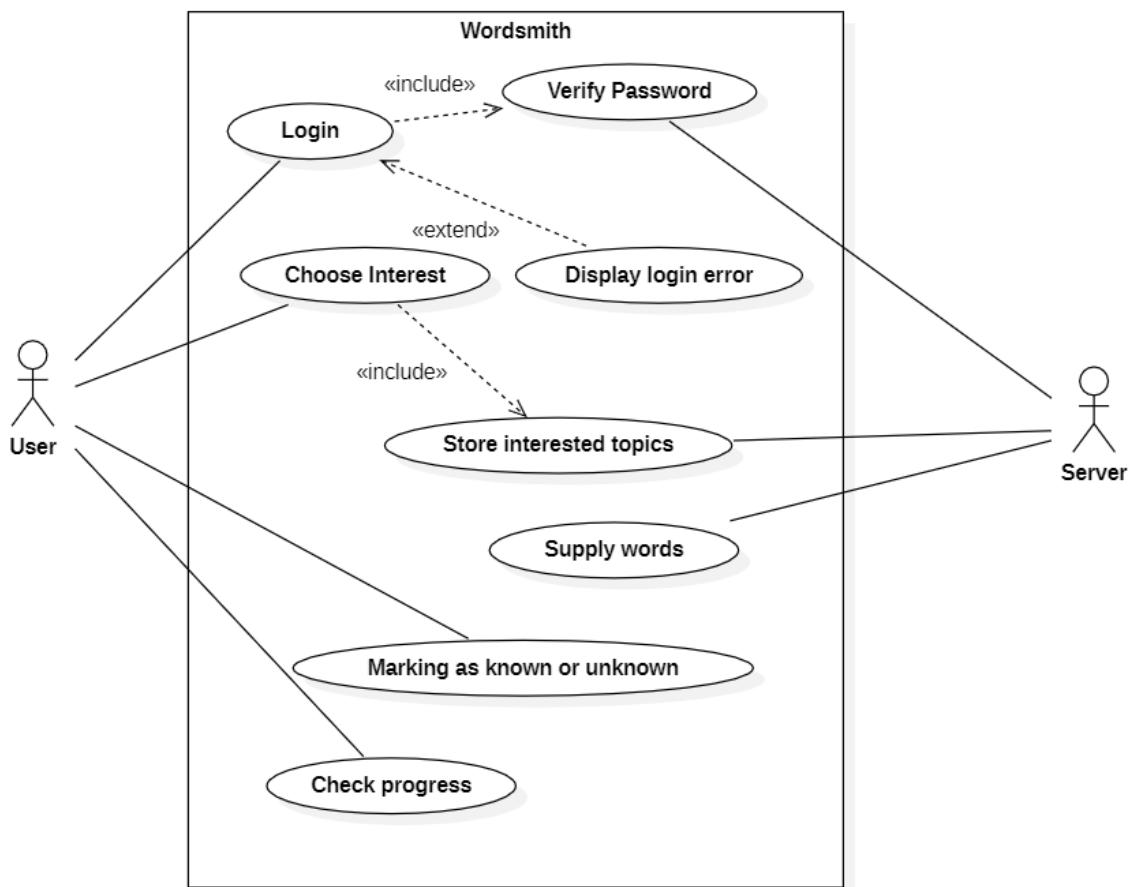


Fig 4.5 Use case diagram

#### 4.1.2.2 Sequence diagram :

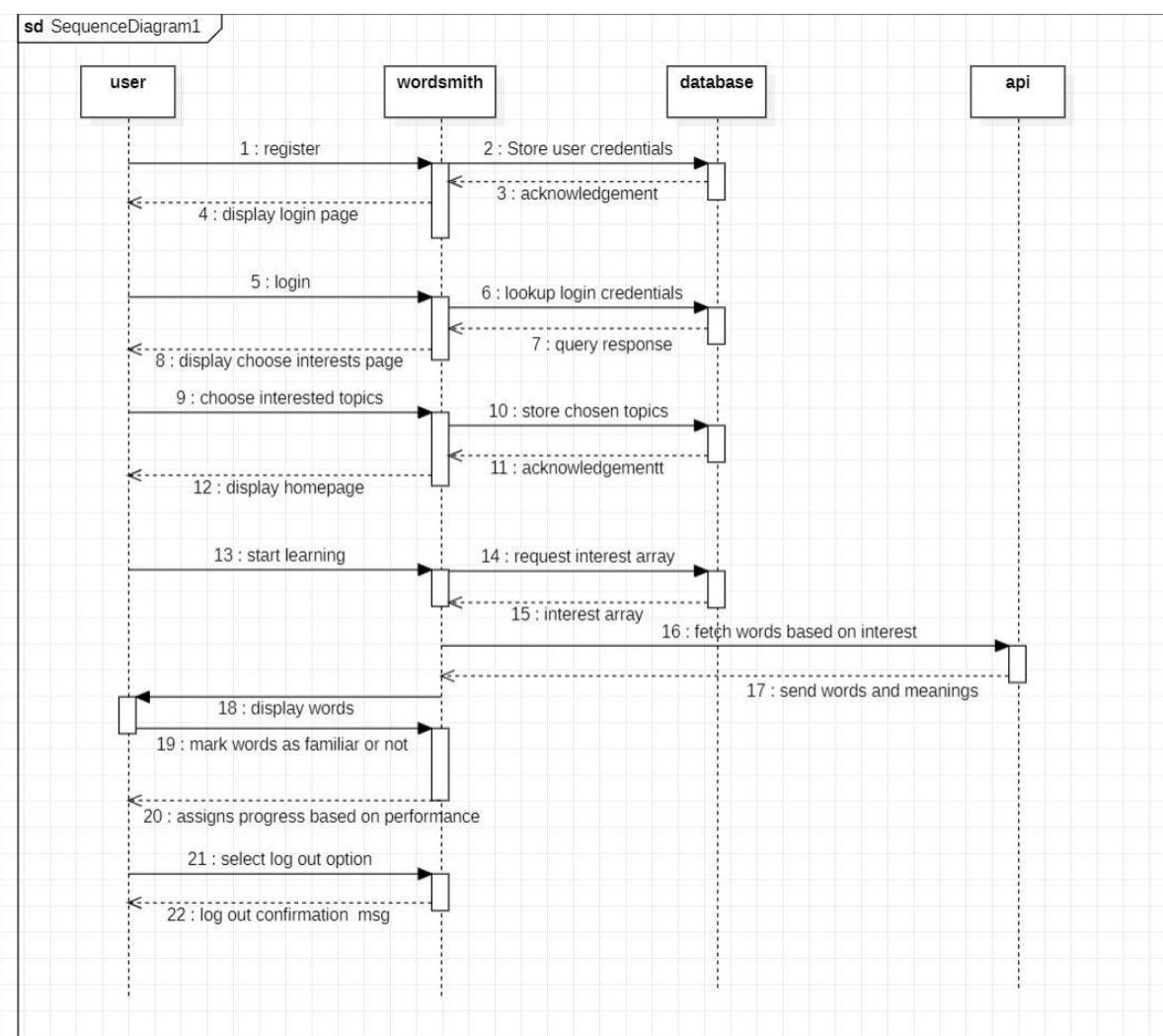


Fig 4.6 Sequence Diagram

#### 4.1.2.3 Collaboration diagram :

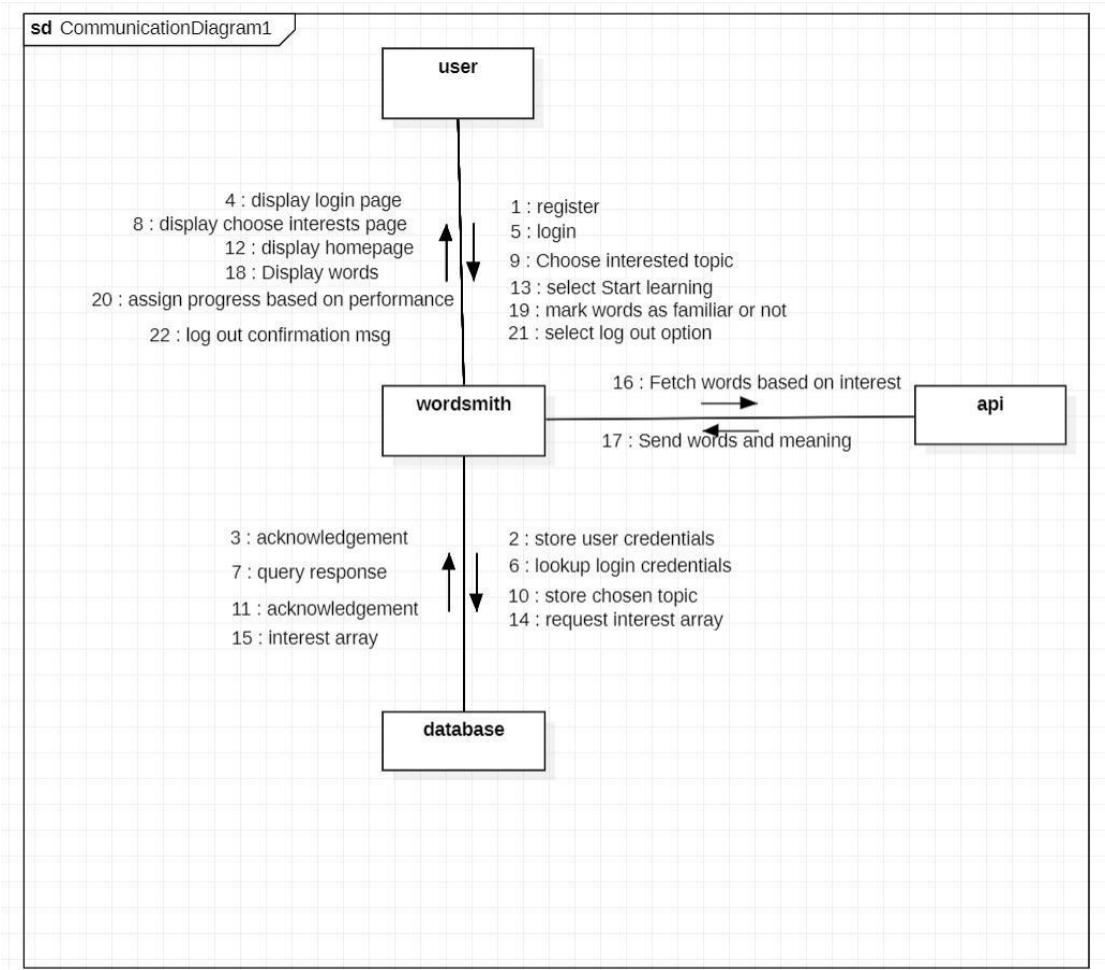


Fig 4.7 Collaboration diagram

#### 4.1.2.4 Statechart diagram :

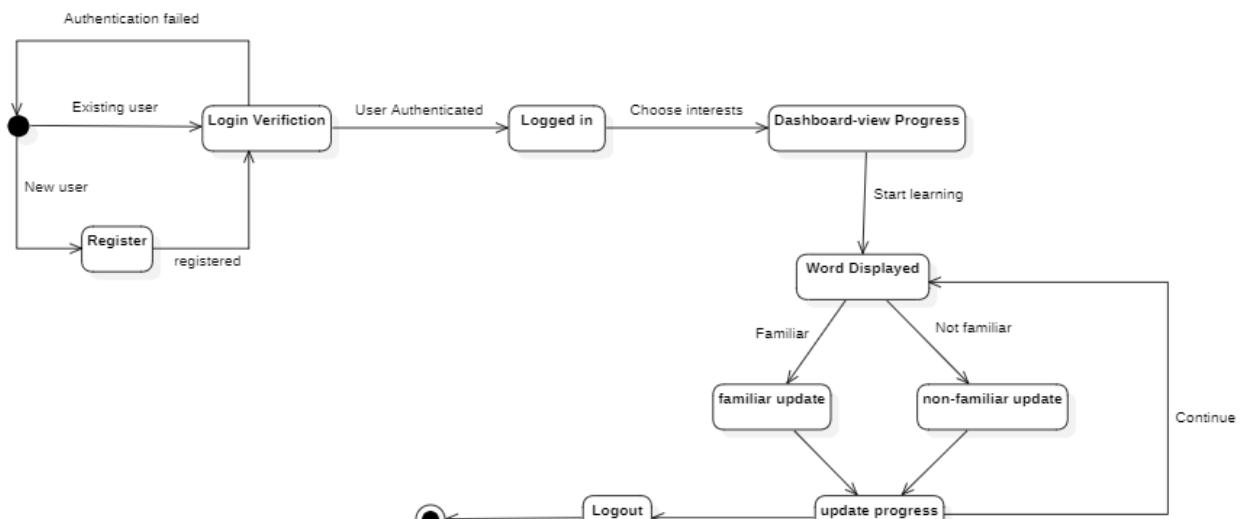


Fig 4.8 Statechart Diagram

#### 4.1.2.5 Activity diagram

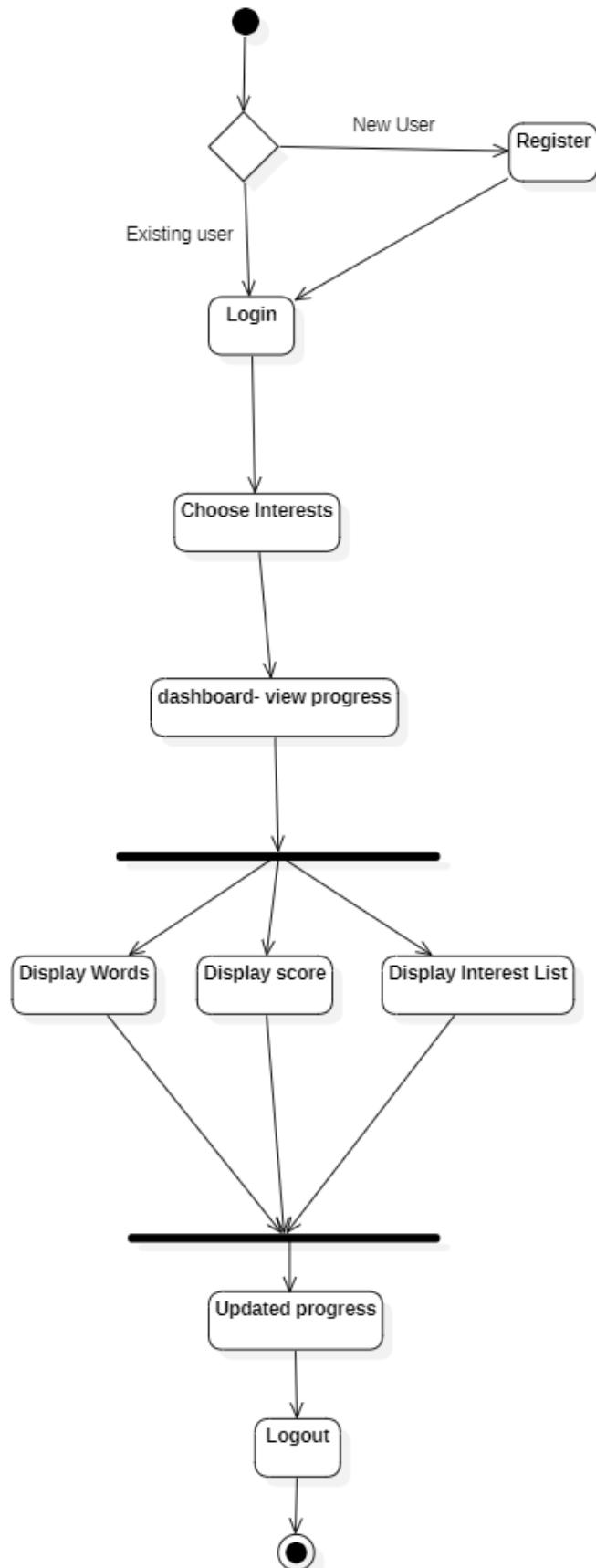


Fig 4.9 Activity Diagram

## 4.2 Algorithms and flowcharts

### ALGORITHM

1. Start
2. Check if JWT token is valid and available
3. If yes;then go to home page
4. else go to register page;
5. Enter credentials for registering;
6. If user exist;
7. then print user already exist;
8. else go to login page;
9. Enter login details
10. If valid log the user in and store the jwt token in localStorage;
11. else print invalid user;
12. After user logged in
13. Check if user already selected interested topics
14. If yes,redirect to home page;
15. else ask user to select topics ;
16. redirect to home page
17. Click proceed;
18. Show the words for All topics
19. If user click any particular topic;
20. show the words for that topic only;
21. Show the progress while user clicking familiar or not familiar button
22. User logout
23. Go to step 6

## Flowchart :

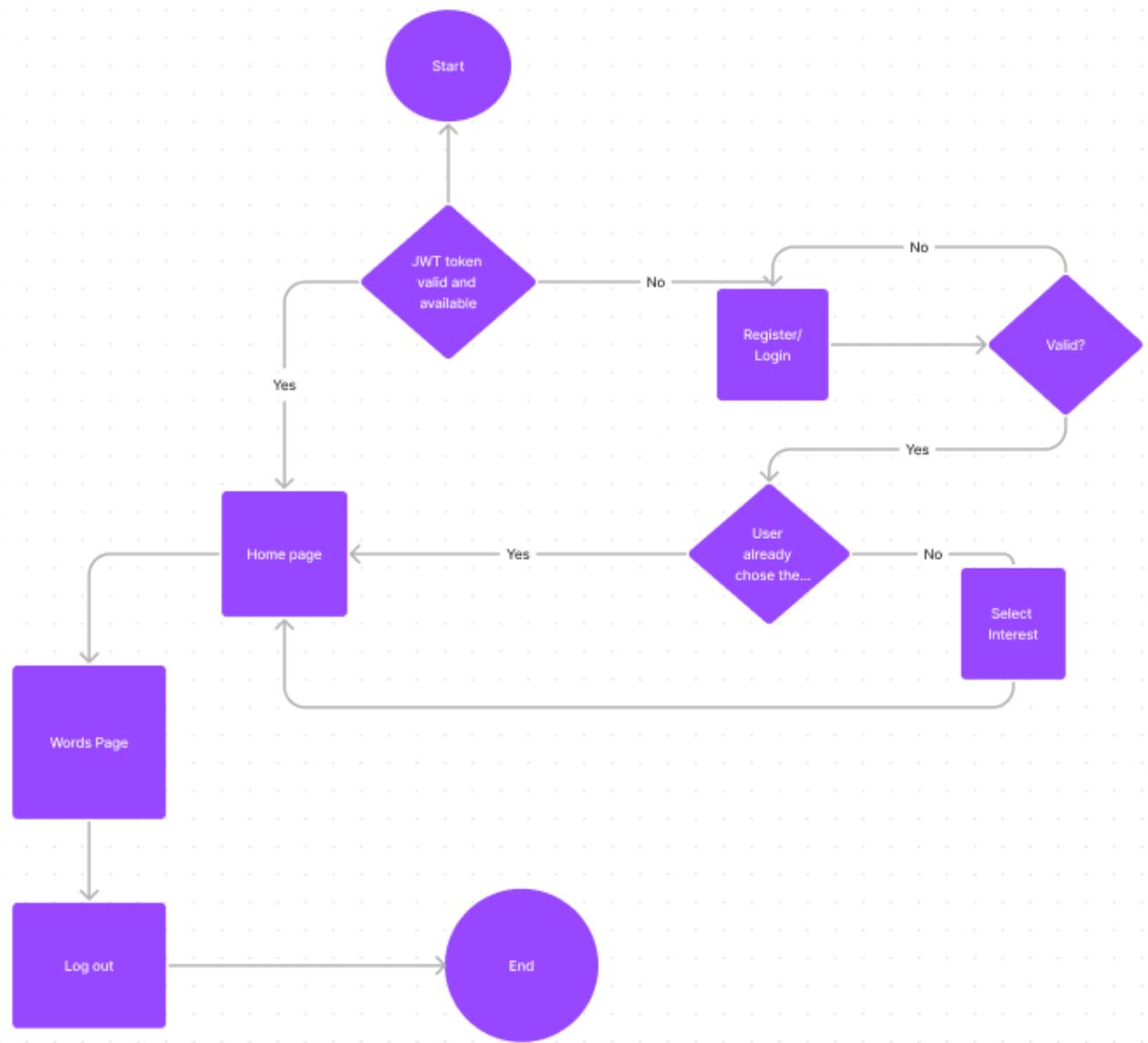


Fig 4.10 Flowchart

# **CHAPTER 5**

## **DEVELOPMENT**

### **5.1 Tools Description and Development approach used**

#### **5.1.1 Figma**

Figma is a collaborative web application for interface design, with additional offline features enabled by desktop applications for macOS and Windows. The feature set of Figma focuses on user interface and user experience design, with an emphasis on real-time collaboration,[1] utilising a variety of vector graphics editor and prototyping tools.

#### **5.1.2 Node.js**

Node.js is an open -source server environment which is free and runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.). It uses JavaScript on the server.

##### **What Can Node.js Do?**

Node.js can generate dynamic page content

Node.js can create, open, read, write, delete, and close files on the server

Node.js can collect form data

Node.js can add, delete, modify data in your database

#### **5.1.3 Electron**

Electron is a free and open-source software framework developed and maintained by GitHub.[6] The framework is designed to create desktop applications using web technologies (mainly HTML, CSS, and JavaScript, though other technologies such as frontend frameworks and Web Assembly are possible) which are rendered using a flavour of the Chromium browser engine, and a backend using the Node.js runtime environment.[7] Additionally, it also uses various APIs to allow things such as native integration with Node services, and an Inter-process communication module.

#### **5.1.4 Discord**

Discord is a voice, video, and text chat app that's used by tens of millions of people ages 13+ to talk and hang out with their communities and friends.

People use Discord daily to talk about many things, ranging from art projects and family trips to homework and mental health support. It's a home for communities of any size, but it's most widely used by small and active groups of people who talk regularly.

Development approach used is waterfall model

## 5.2 Pseudocodes of Important Modules

### 5.2.1 Login module

1. Start
2. Get password and email from user
3. Validate password and email
4. If valid;create a jwt token by signing it with user email id
5. Send the token to user
6. Else; send invalid error message
7. Stop

### 5.2.2 Register module

1. Start
2. Get password,name,email from user
3. Check whether if the user exist already;
4. If yes; send user already available error message
5. else;store it in database and send registered message
6. Stop

### 5.2.3 Store topic module

1. Start
2. Get the jwt token and topics to store from user
3. Check if jwt token is valid
4. If yes; store the topic in database of respective user

5. else; send invalid request message

6. Stop

#### 5.2.4 Get topics Module

1. Start

2. Get the jwt token from the user

3. Check whether the the token is valid

4. If valid; then send the topics of that respective user

5. else; send invalid message

6. Stop

#### 5.2.5 Store words module

1. Start

2. Get the jwt token and verify it

3. If valid; then store the words in database

4. Check the length of the words

5. If length<=7; increment easy variable

6. Else if length > 8 and <10;increment medium variable

7. Else; increment hard variable

8. store these variables in database of respective users

9. If not valid;send ERROR message

#### 5.2.6 Get words module

1. Start

2. Get the jwt token from user and verify

3. If valid;get all the words related to the topics chosen of that user

4. send that words and meaning to user in an array of object

5. Stop

# CHAPTER 6

## TESTING

### 6.1 Test cases of important modules

#### 6.1.1 Login module

<b>Project Name:</b>	Wordsmith					 <a href="http://www.SoftwareTestingMaterial.com">www.SoftwareTestingMaterial.com</a>			
<b>Module Name:</b>	Login								
<b>Created by:</b>	Aaron Mathew								
<b>Date of creation:</b>	09-11-2022								
<b>Date of review:</b>	10-11-2022								
TEST CASE ID	TEST SCENARIO	TEST CASE	PRE-CONDITION	TEST STEPS	TEST DATA	EXPECTED RESULT	POST CONDITION	ACTUAL RESULT	STATUS (PASS/FAIL)
TC_LOGIN_001	Verify the login	Enter valid User Name and valid Password	1. Need a valid Account to do login	1. Enter User Name 2. Enter Password 3. Click "Login" button	<Valid User Name> <Valid Password>	Successful login	Home Page is Shown	Successful login while maintaining user session	PASS
TC_LOGIN_002	Verify the login	Enter valid User Name and invalid Password	1. Need a valid Account to do login	1. Enter User Name 2. Enter Password 3. Click "Login" button	<Valid User Name> <Invalid Password>				
TC_LOGIN_003	Verify the login	Enter invalid User Name and valid Password	1. Need a valid Account to do login	1. Enter User Name 2. Enter Password 3. Click "Login" button	<Invalid User Name> <Valid Password>	A message "Invalid Credentials" is shown	Error Message is displayed	Unsuccessful Login	PASS
TC_LOGIN_004	Verify the login	Enter invalid User Name and invalid Password	1. Need a valid Account to do login	1. Enter User Name 2. Enter Password 3. Click "Login" button	<Invalid User Name> <Invalid Password>				

Fig 6.1 Test case for login module

#### 6.1.2 Registration module

<b>Project Name:</b>	Wordsmith					 <a href="http://www.SoftwareTestingMaterial.com">www.SoftwareTestingMaterial.com</a>			
<b>Module Name:</b>	Registration								
<b>Created by:</b>	Aaron Mathew								
<b>Date of creation:</b>	09-11-2022								
<b>Date of review:</b>	10-11-2022								
TEST CASE ID	TEST SCENARIO	TEST CASE	PRE-CONDITION	TEST STEPS	TEST DATA	EXPECTED RESULT	POST CONDITION	ACTUAL RESULT	STATUS (PASS/FAIL)
TC_REGISTER_001	Verify the registration	Enter Invalid Email , Valid Username and Password	1. Need a valid email	1. Enter Username 2. Enter Email 3. Enter password	Valid Username InValid Email Valid Password	A message "The email you entered is invalid" is shown	Message "Registration successful is shown" User details stored in database	Unsuccessful Registration	PASS
TC_REGISTER_002	Verify the registration	Enter Invalid Username Valid Email And Valid Password	1. Need a valid email	1. Enter Username 2. Enter Email 3. Enter password	Invalid Username Valid Email Valid Password				
TC_REGISTER_003	Verify the registration	Enter Valid Username , Valid Email and Invalid Password	1. Need a valid email	1. Enter Username 2. Enter Email 3. Enter password	Valid Username Valid Email Invalid Password	A message "The password you entered does not match the criteria" is shown	Error Message is displayed	Unsuccessful Registration	PASS
TC_REGISTER_004	Verify the registration	Enter Valid Username , Valid Email and Valid Password	1. Need a valid email	1. Enter Username 2. Enter Email 3. Enter password	Valid Username Valid Email Valid Password				

Fig 6.2 Test case for registration module

### 6.1.3 Input validation module

Project Name:	Wordsmith					<a href="http://www.SoftwareTestingMaterial.com">www.SoftwareTestingMaterial.com</a>			
Module Name:	Input validation and data sanitization								
Created by:	Aaron Mathew								
Date of creation:	09-11-2022								
Date of review:	10-11-2022								
TEST CASE ID	TEST SCENARIO	TEST CASE	PRE-CONDITION	TEST STEPS	TEST DATA	EXPECTED RESULT	POST CONDITION	ACTUAL RESULT	STATUS (PASS/FAIL)
TC_INPTOPIC_001	To add an interest by typing it out	Valid Input	1. Need to be registered and validated	1. Tap on a Search field 2. Enter data 3. Add interest	Input text (Interest)	The text being searched is validated. Data sanitization takes place	Interest is added to the array	Valid input is added to the array	PASS
TC_INPTOPIC_002	To add an interest by typing it out	Valid Input	1. Need to be registered and validated	1. Tap on a Search field 2. Enter data 3. Add interest	Input text (Interest)	The text being searched is validated. Data sanitization takes place	Interest is added to the array	Invalid input is added to the array	FAIL

Fig 6.3 Test case for input validation

# CHAPTER 7

## SCREENSHOTS OF DEVELOPED PRODUCT

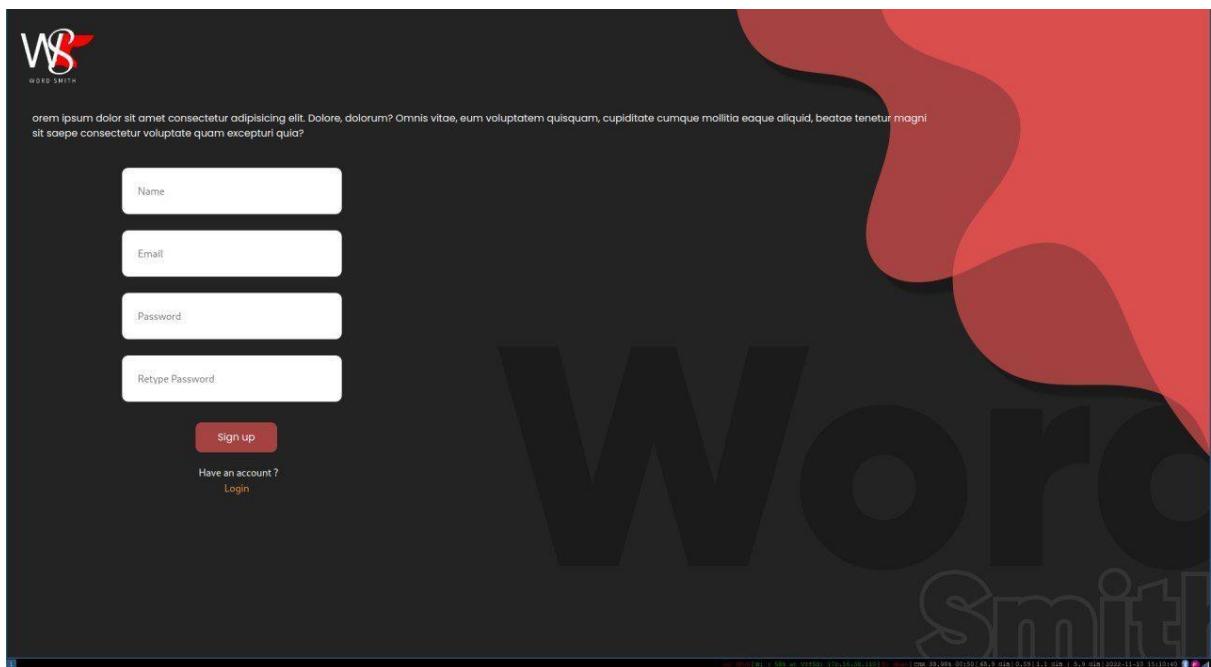


Fig 7.1 Registration page

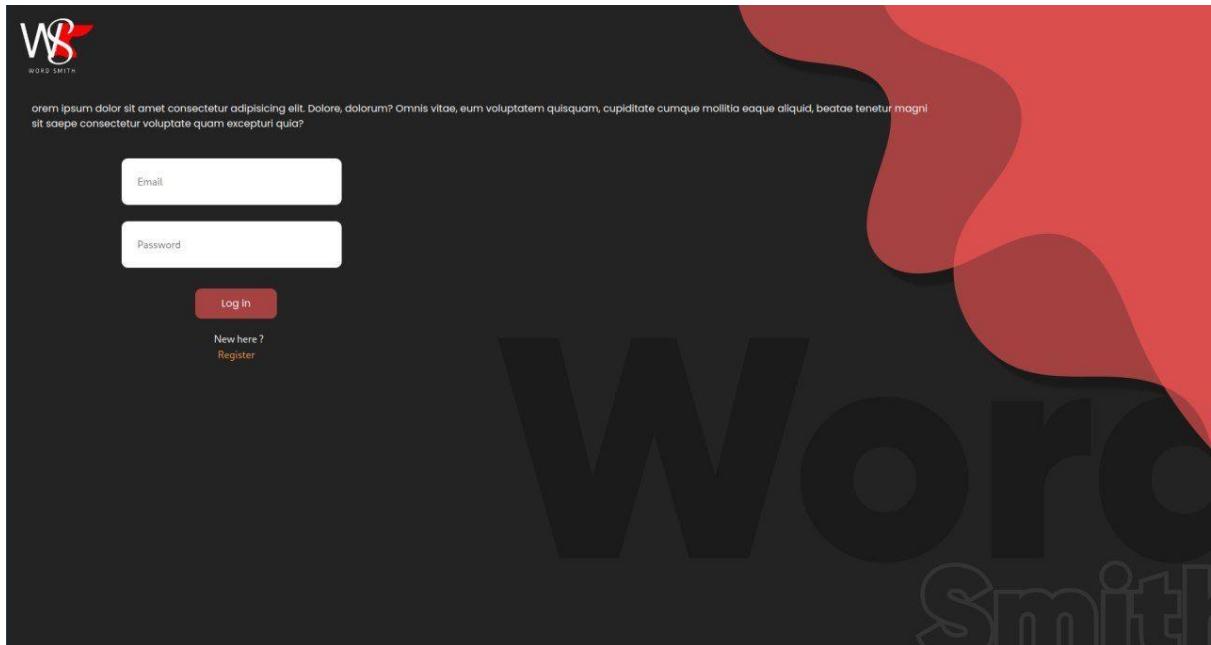


Fig 7.2 login page

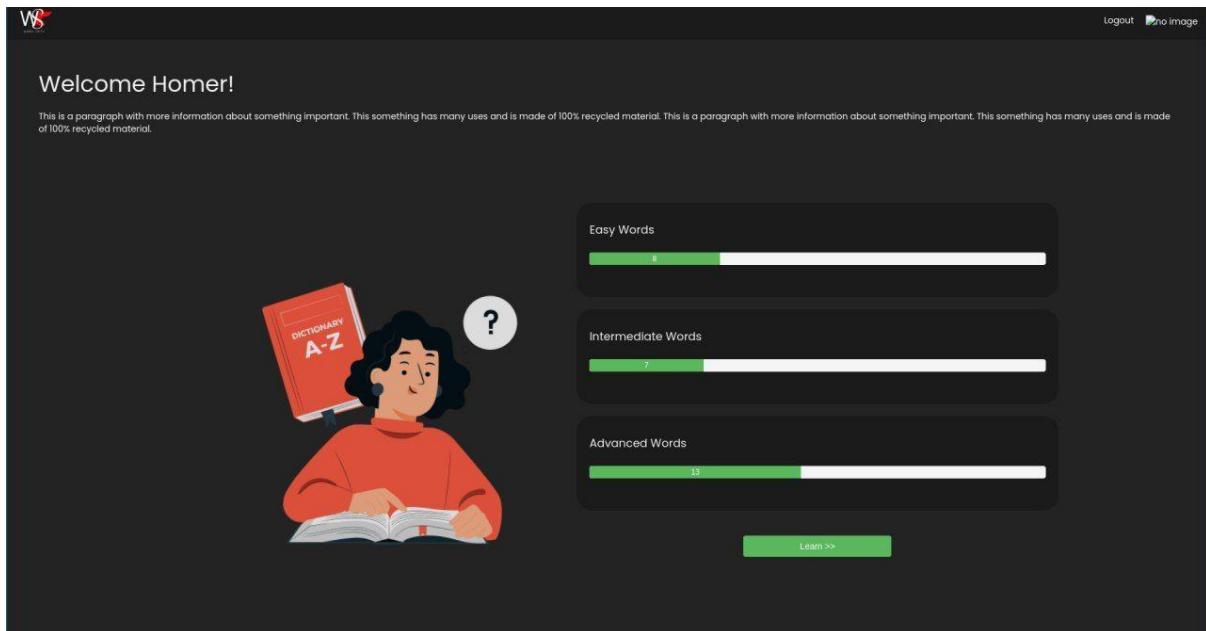


Fig 7.3 home page

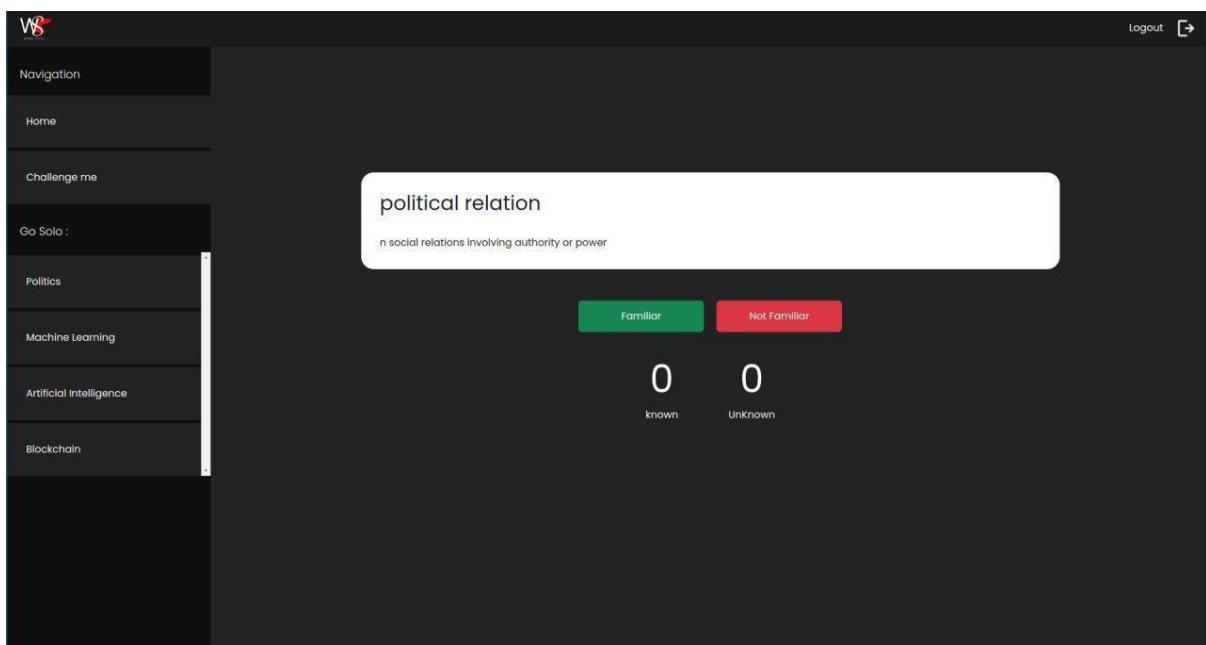
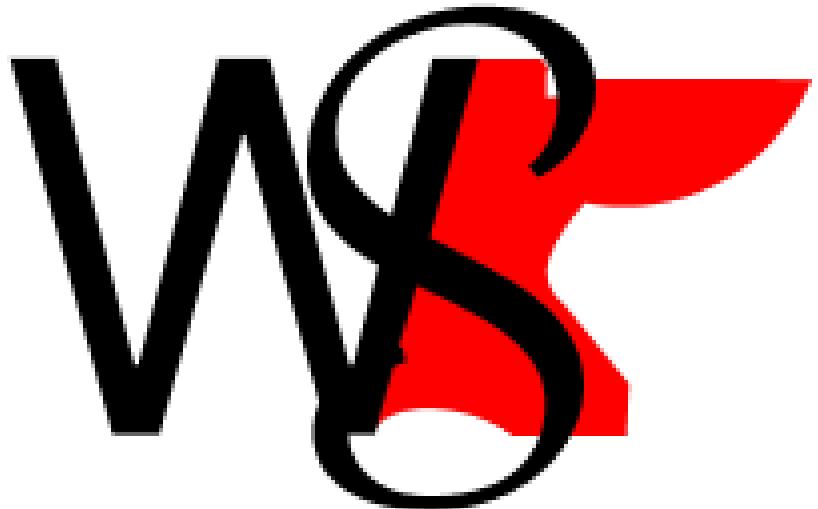


Fig 7.4 Words page





WORD SMITH

## User Manual

---

**Version 1.0**

**10/11/2022**

# **Table of Contents**

<b>1. Introduction.....</b>	<b>3</b>
1.1    Overview .....	3
<b>2. Getting Started .....</b>	<b>4</b>
2.1    Cautions & Warnings.....	4
2.2    Set-up Considerations .....	4
2.3    User Access Considerations.....	4
2.4    Accessing the System .....	4
2.5    System Organization & Navigation.....	5
2.6    Exiting the System .....	5
<b>3. Using the System .....</b>	<b>6</b>
<b>4. Troubleshooting &amp; Support.....</b>	<b>7</b>
4.1    Support.....	7

---

# **1. Introduction**

---

WordSmith is a standalone application aimed at improving your proficiency when it comes to the English language, encouraging you to work on your vocabulary and build upon it based on your interests and desires.

## **1.1 Overview**

The application offers the following features:

- Ability to register your own account
- Choose your interested topics
- Ability to filter words based on your interests
- Search for topics and words
- Classify words as easy, intermediate, or advanced
- Mark a particular word as ‘familiar’ or ‘unfamiliar’ based on your knowledge
- Get the meaning of any word you are unfamiliar with.

---

## **2. Getting Started**

---

- Once application is downloaded, you are prompted to create an account and register
- After registering your account, you can sign in using your login credentials
- Upon first login, you are prompted to select your topics of interest, for which you would like to expand your vocabulary in
- After choosing your topics of interest, you will be guided to your user dashboard account, where you can access the features

### **2.1 Cautions & Warnings**

Minimum age required for registration is 13 years old, as per government guidelines.

### **2.2 Set-up Considerations**

- Recommended OS is Windows 10 or above
- Internet connection required to access the platform
- Firewall must allow access to download EXE file for the application

### **2.3 User Access Considerations**

- Any user can access the site once they have registered an account
- They can sign in and access the content at any time

### **2.4 Accessing the System**

- If accessing the system for the first time, one must sign up and are prompted to register.
- Fill in an appropriate username and password
- Fill in your email address and other relevant details asked
- Keep the password at a recommended length of 8 characters, using Uppercase and lowercase letters, numbers, and symbols
- Register your account
- Sign in using your credentials after registering by clicking the login button.
- If credentials need to be recovered, click the forgot password option. You can use this to recover your credentials via email

---

## **2.5 System Organization & Navigation**

- First page will be a registration/login page
- For first time users, after first logging in they will be navigated to a “Choose Interests” section where they may choose their interests
- They are then navigated to their user dashboard. For every subsequent login they will directly be redirected to the dashboard

## **2.6 Exiting the System**

- Click on the ‘Logout’ button on the top right corner of your user dashboard once you wish to sign out of the application
- You can also simply close the application window as well

---

### **3. Using the System**

---

- You can access all the features of the system through your user dashboard
- Click on the Search Icon to search for other topics of interest or words that may interest you
- Click on “Learn” button to begin word recommendations along with their meanings
- Mark each word as ‘familiar’ or ‘unfamiliar’ based on your proficiency
- Words will be suggested based on the topics of interest chosen
- Words will be filtered into three categories: Beginner, Intermediate and Advanced
- A green sidebar will show the progress in each category
- Click on either category to view the respective terminologies already covered
- To view words from one specific topic only, click on the left sidebar in the user dashboard and click on the specific topic you wish to learn from
- Click on ‘Challenge Me’ in the left-side of the sidebar in your user dashboard if you wish to be quizzed on the words you’ve already gone through to check your proficiency
- Logout once you’re finished.

---

## **4. Troubleshooting & Support**

---

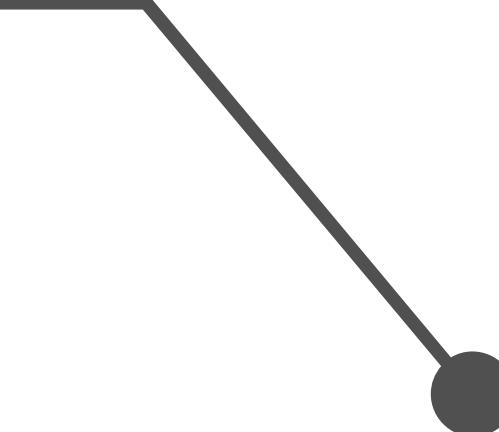
- Check your internet connection in case of slow connectivity
- Upgrade to the latest Window Software

### **4.1 Support**

In case of further assistance, contact support using the below details:

Contact	Organization	Phone	Email
Supervisor	WordSmith Inc.	+91 9650599044	WordSmith@gmail.com

---



# WORDSMITH

A Software Engineering project by:-

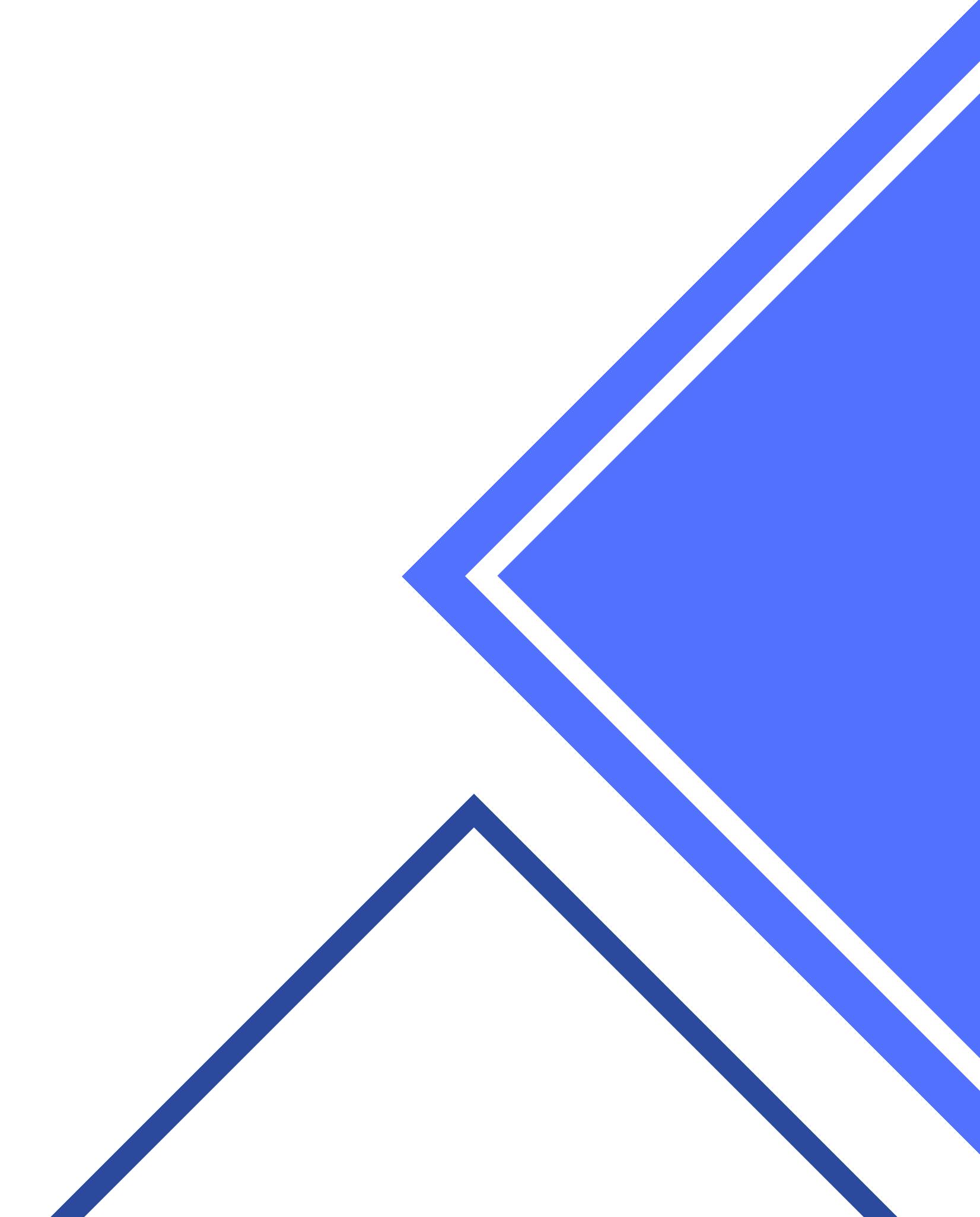
Anshuman Gupta

Aaron K Mathew

Abhinav Bijith

Karthik Remesh

Athul A



## IN SCOPE

- 1) WordSmith is a standalone application that helps build ones vocabulary.
- 2) We would require 1-2 UI/UX designers to design a user friendly and easily navigable application.
- 3) We would require 2 front end and 2 backend developers for the development phase
- 4) We would be required to be proficient in electron js for the development process.
- 5) WordSmith would help people improve in English. It also helps one prepare for exams such as GRE.
- 6) A user manual would be required

# SCOPE

THE SOFTWARE WILL ABLE TO PERFORM :

- 1) FILTRATION OF WORDS BASED ON THE INTEREST OF THE USER.
- 2) A CROSS COMPATIBLE STANDALONE DESKTOP APP WHICH CAN BE DEPLOYED IN WINDOWS/LINUX/MAC. THE USERS WOULD BE PROVIDED WITH A .EXE FILE.
- 3) SOFTWARE WILL BE MANAGED PURELY BY THE CLIENT AND THERE WILL BE DATABASE(REMOTE) AT THE BACKEND

# OUT OF SCOPE

- 1) FINANCIAL INVESTMENTS
- 2) OUT SOURCING
- 3) PRONUNCIATION OF THE WORD(AUDIO)
- 4) SUPPORTING DIFFERENT LANGUAGES
- 5) HAVING ONLINE COMPETITION WITH YOUR FRIENDS THROUGH THE APP
- 6) A LEADERBOARD
- 7) EXAM MODE – TO HELP YOU PREPARE FOR EXAMS SUCH AS GRE
- 8) GAMIFICATION

## ASSUMPTIONS

- 1) REQUIREMENTS OF OUR SYSTEMS MATCHES THE REQUIREMENTS.
- 2) CORRECT CHOICE OF ML ALGORITHM.
- 3) PEOPLE ASSIGNED DIFFERENT ROLES WOULD ADHERE TO THEIR WORK
- 4) END USERS WILL BE AVAILABLE FOR TESTING
- 5) THE PROJECT SCOPE WILL NOT CHANGE ONCE THE STAKEHOLDERS SIGN OFF ON THE SCOPE STATEMENT
- 6) PROJECT WILL FOLLOW WATERFALL METHODOLOGY THROUGHOUT EXECUTION

## DELIVERABLES

- 1) WORDSMITH
- 2) SOURCE CODE
- 3) USER MANUAL
- 4) SUPPORTING DOCUMENTS  
SOFTWARE SYSTEM

## CONSTRAINTS

- 1) TIME - GETTING THE WORK DONE IN THE SHORT SPAN OF THIS SEMESTER.
- 2) LACK OF AVAILABILITY OF OPEN-SOURCE CODE
- 3) OBTAINING THE CORRECT API/ DATASETS THAT WE WOULD USE EXPERTISE - OUR TEAM IS NOT WELL VERSED WITH ELECTRONJS.

# WORK BREAKDOWN STRUCTURE



# SURVEY RESULTS

Questions Responses 51 Settings

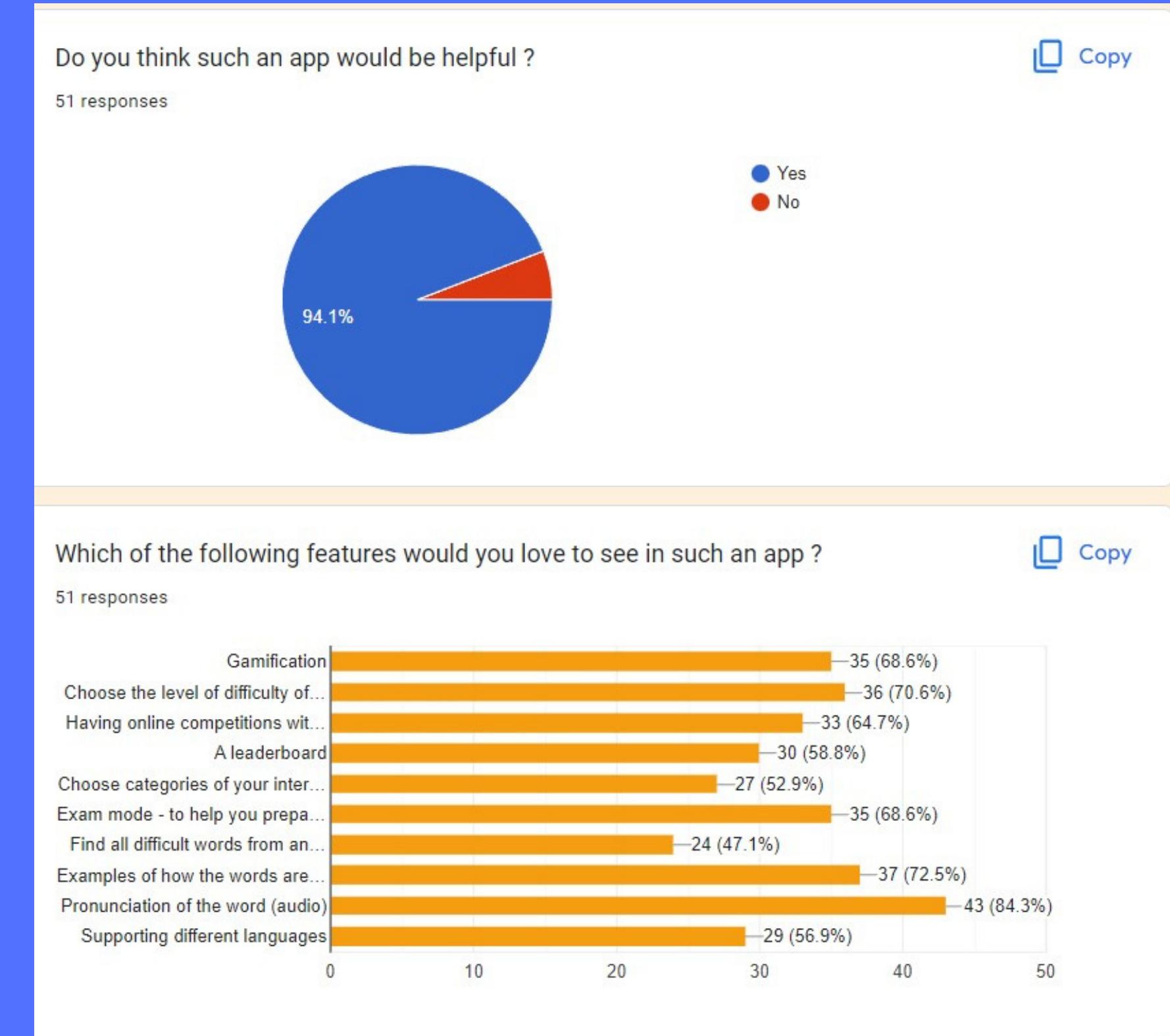
51 responses

Accepting responses 

Summary Question Individual

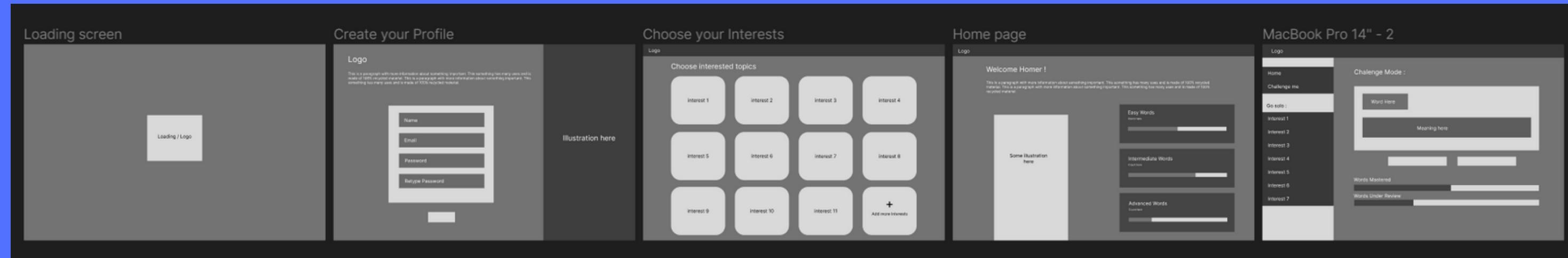
Who has responded?

- nesamanit@gmail.com
- dhwanil.dubal@vitstudent.ac.in
- aniruddh.pandey1601@gmail.com
- abhinavbijith@gmail.com
- srividymurugadoss@gmail.com
- poddar.srisha@gmail.com
- pranavram063@gmail.com
- varun.v2021@vitstudent.ac.in
- tanban1905@gmail.com

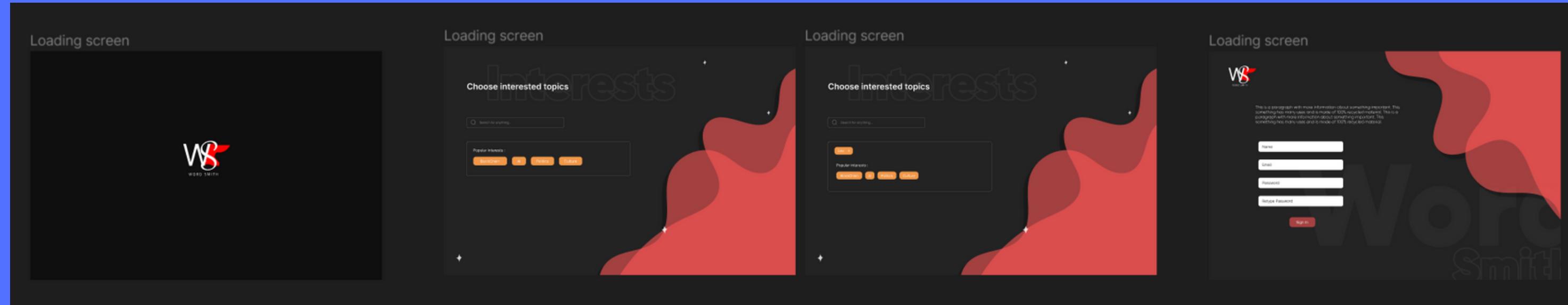


# DESIGN PROCESS

## WIREFRAMING

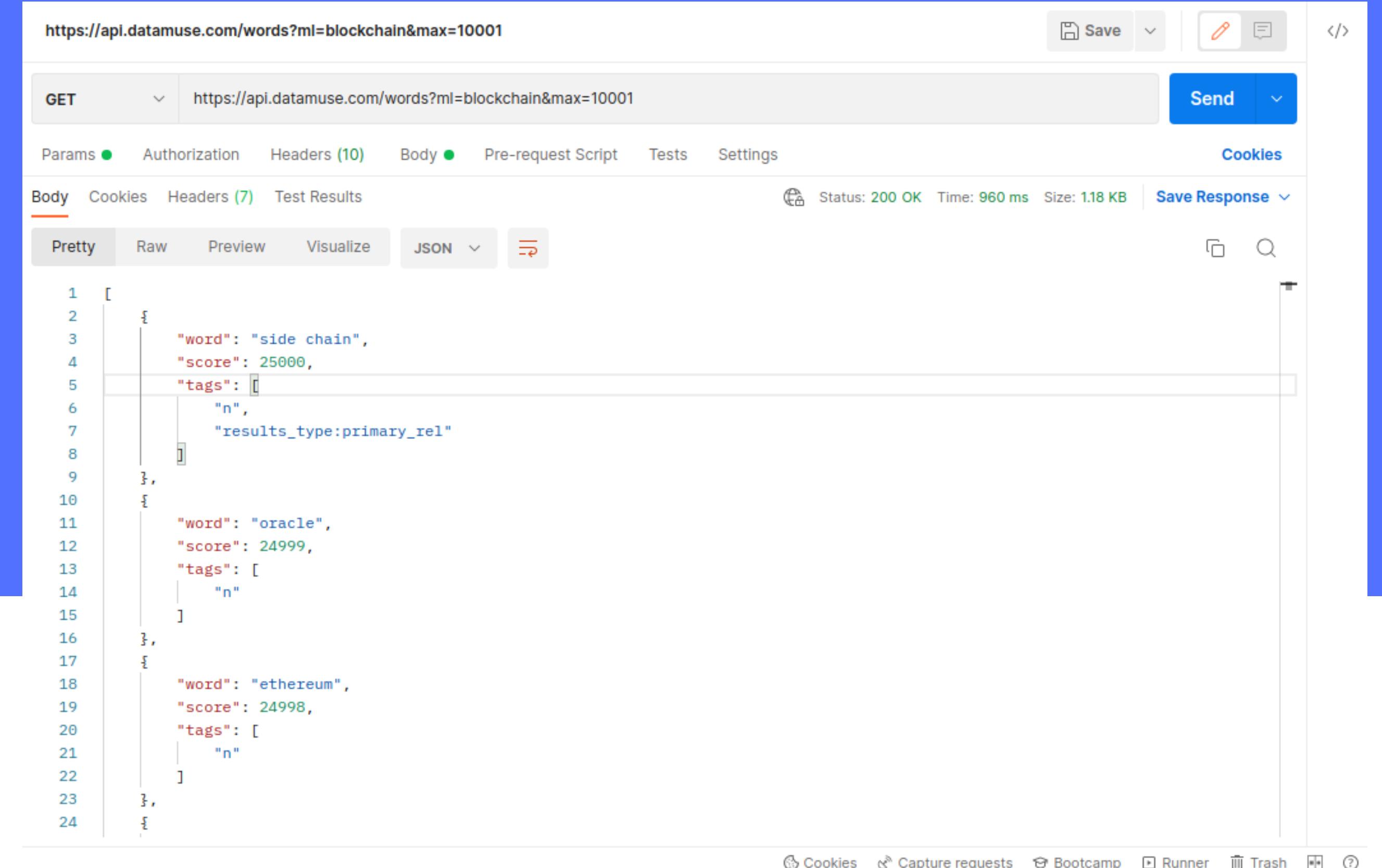


## HIGH FIDELITY DESIGN



# API BEING USED

GIVES US THE  
RELATED TERMS  
AND WORDS



The screenshot shows the Postman API client interface. The URL in the header is `https://api.datamuse.com/words?ml=blockchain&max=10001`. The method is set to `GET`. The response status is `200 OK` with a time of `960 ms` and a size of `1.18 KB`. The response body is displayed in a pretty-printed JSON format, showing three results. Each result includes a word, its score, and a list of tags.

```
1 [  
2 {  
3     "word": "side chain",  
4     "score": 25000,  
5     "tags": [  
6         "n",  
7         "results_type:primary_rel"  
8     ]  
9 },  
10 {  
11     "word": "oracle",  
12     "score": 24999,  
13     "tags": [  
14         "n"  
15     ]  
16 },  
17 {  
18     "word": "ethereum",  
19     "score": 24998,  
20     "tags": [  
21         "n"  
22     ]  
23 },  
24 {
```

1

### WHAT IS THE PRODUCT AND WHAT PROBLEM DOES IT SOLVE ?

*Wordsmith is a vocabulary building app that helps improve ones vocabulary. this can be used by people that aren't fluent in the language, people preparing for exams such as GRE and people wanting to improve themselves*

2

### WHAT WERE THE INITIAL FEATURES DECIDED FOR THE PRODUCT ?

- 1) Learning terms and meanings of words based on one's interest.
- 2) Choosing the level of difficulty
- 3) Scoreboard
- 4) Registration / Sign-in

3

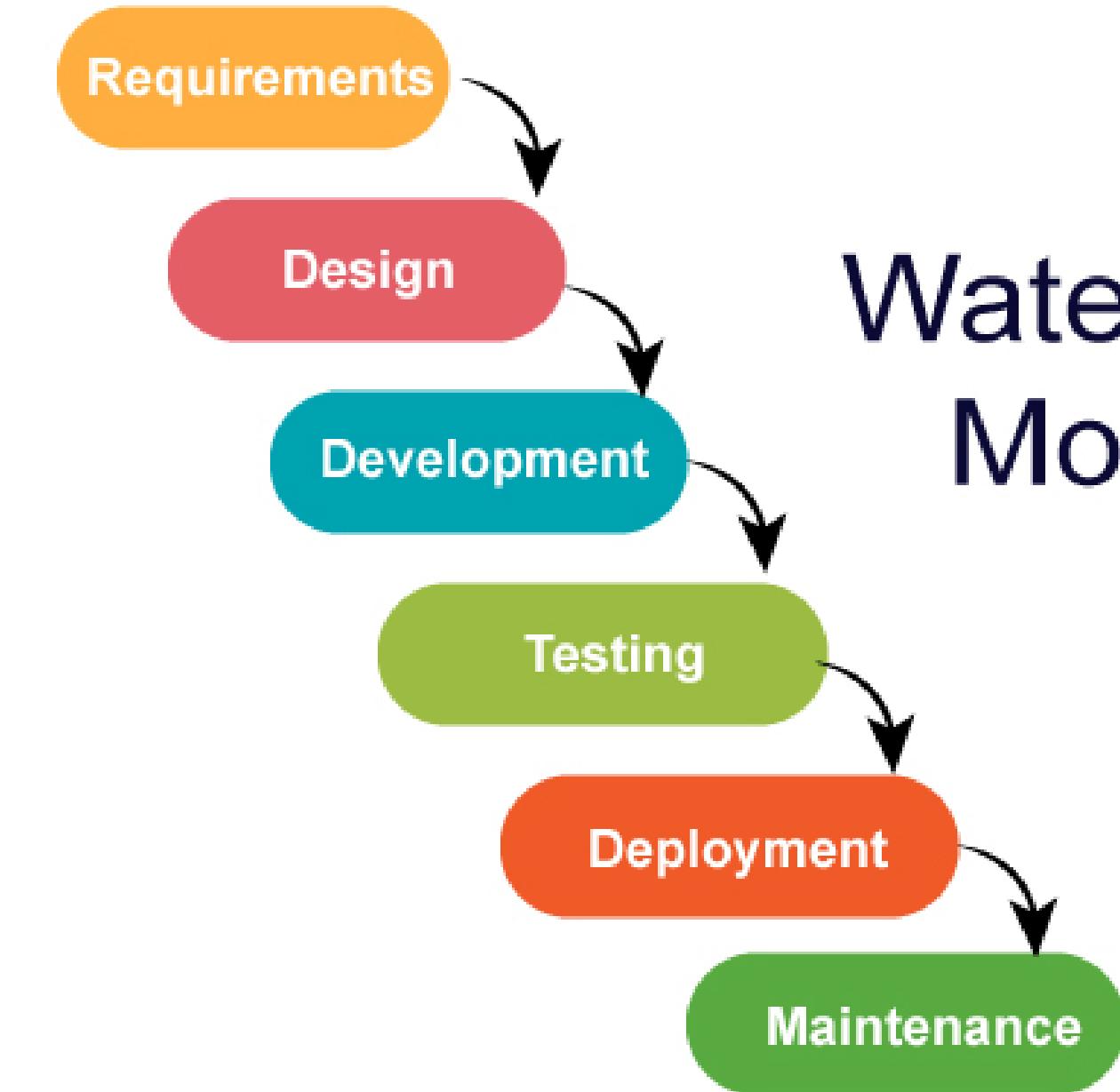
### WHAT ARE THE REMAINING SET OF FEATURES THAT YOU DECIDED WOULD MAKE YOUR MVP AND WHY?

- 1)Pronunciation of the word(audio)
- 2)Supporting different languages
- 3)Having online competition with your friends through the app
- 4)A leaderboard
- 5)Exam mode - to help you prepare for exams such as GRE
- 6)Gamification



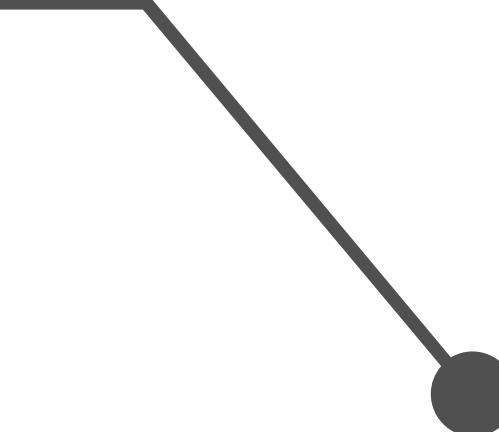
# MODEL WE CHOSE

WE HAVE DECIDED TO GO WITH THE WATERFALL MODEL FOR OUR PROJECT IN THIS WATERFALL MODEL, TYPICALLY, THE OUTCOME OF ONE PHASE ACTS AS THE INPUT FOR THE NEXT PHASE SEQUENTIALLY. THE FOLLOWING ILLUSTRATION IS A REPRESENTATION OF THE DIFFERENT PHASES OF THE WATERFALL MODEL.



## Waterfall Model

---



# WORDSMITH

A Software Engineering project by:-

Anshuman Gupta

Aaron K Mathew

Abhinav Bijith

Karthik Remesh

Athul A



## IN SCOPE

- 1) WordSmith is a standalone application that helps build ones vocabulary.
- 2) We would require 1-2 UI/UX designers to design a user friendly and easily navigable application.
- 3) We would require 2 front end and 2 backend developers for the development phase
- 4) We would be required to be proficient in electron js for the development process.
- 5) WordSmith would help people improve in English. It also helps one prepare for exams such as GRE.
- 6) A user manual would be required

# SCOPE

THE SOFTWARE WILL ABLE TO PERFORM :

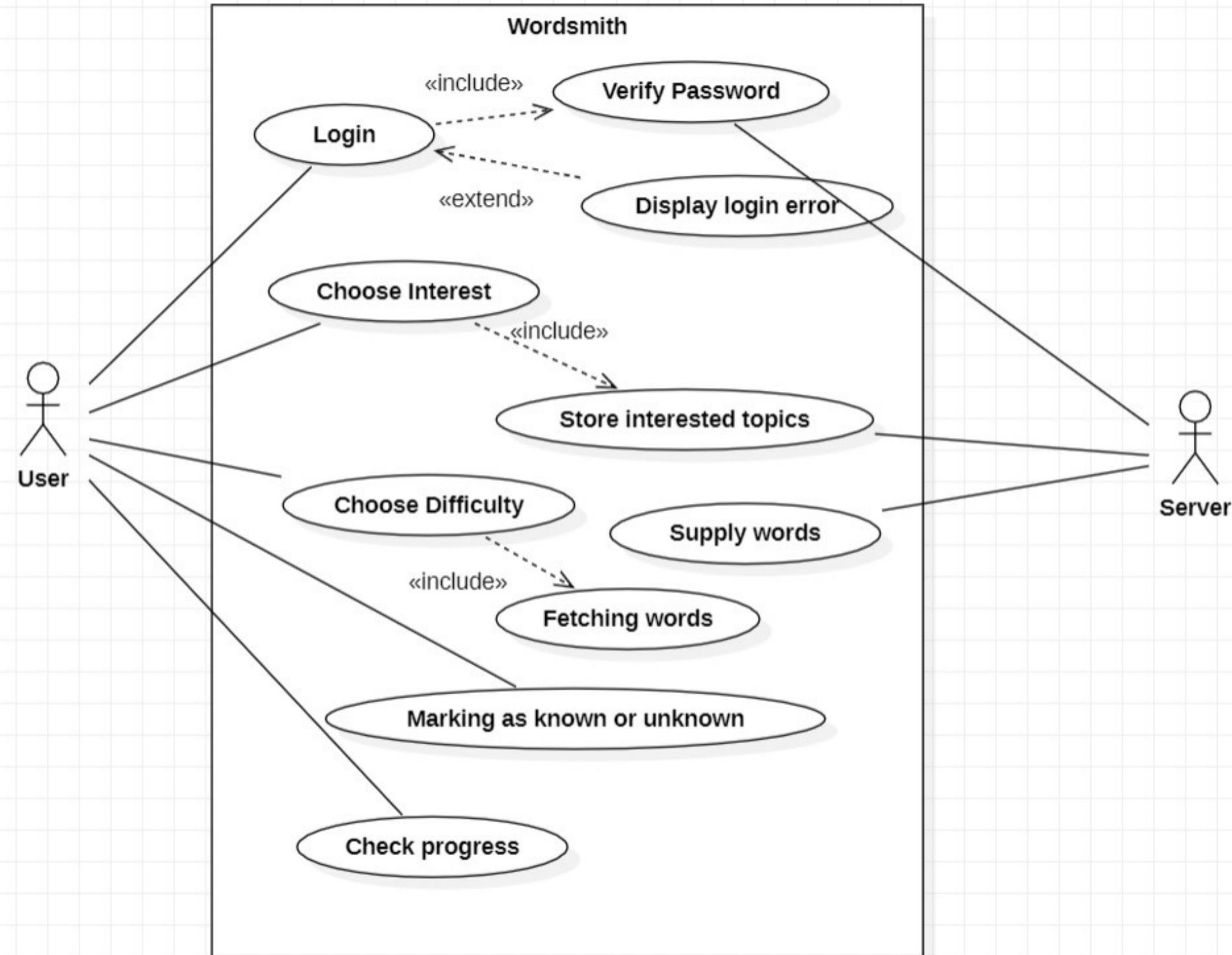
- 1) FILTRATION OF WORDS BASED ON THE INTEREST OF THE USER.
- 2) A CROSS COMPATIBLE STANDALONE DESKTOP APP WHICH CAN BE DEPLOYED IN WINDOWS/LINUX/MAC. THE USERS WOULD BE PROVIDED WITH A .EXE FILE.
- 3) SOFTWARE WILL BE MANAGED PURELY BY THE CLIENT AND THERE WILL BE DATABASE(REMOTE) AT THE BACKEND

# OUT OF SCOPE

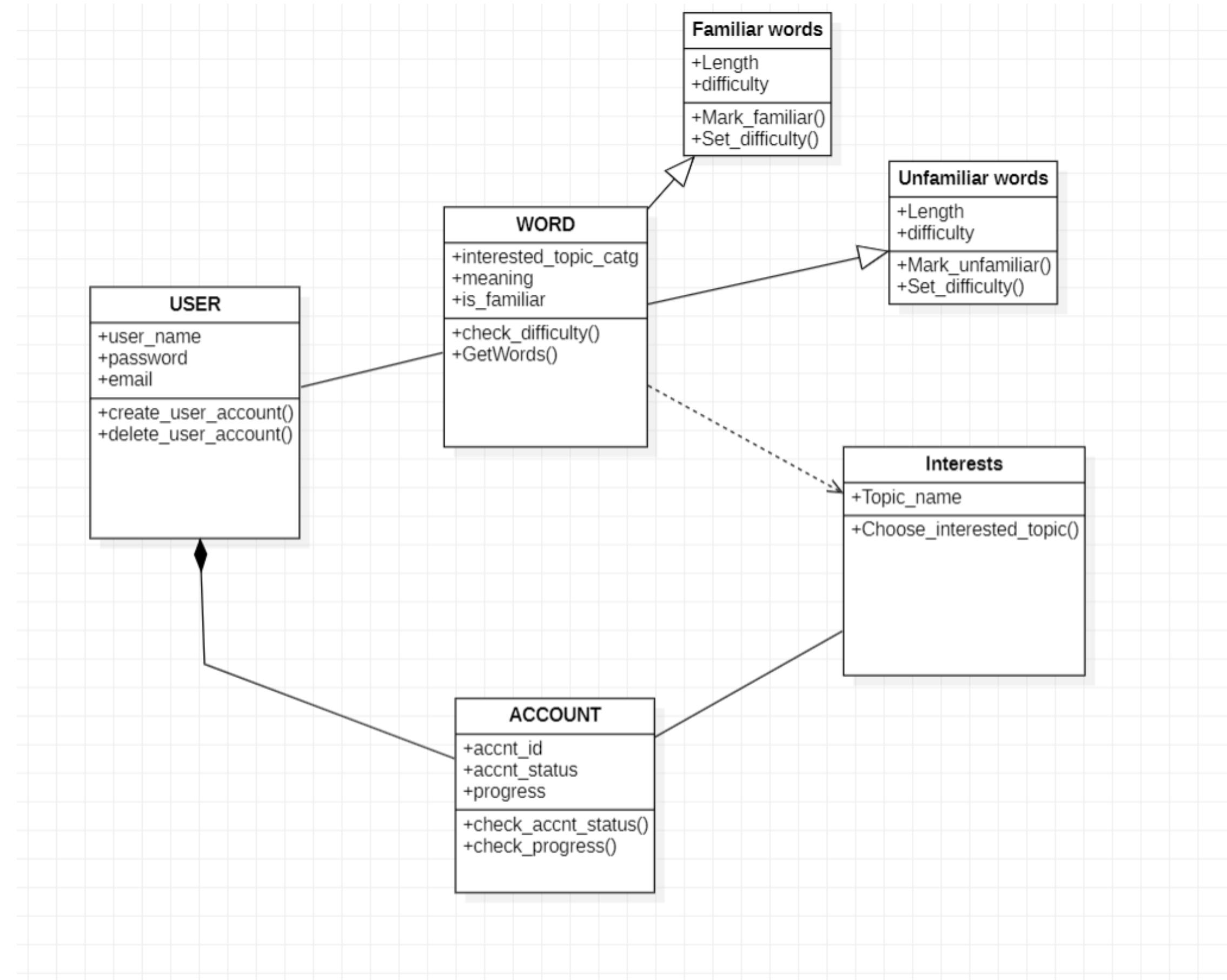
- 1) FINANCIAL INVESTMENTS
- 2) OUT SOURCING
- 3) PRONUNCIATION OF THE WORD(AUDIO)
- 4) SUPPORTING DIFFERENT LANGUAGES
- 5) HAVING ONLINE COMPETITION WITH YOUR FRIENDS THROUGH THE APP
- 6) A LEADERBOARD
- 7) EXAM MODE – TO HELP YOU PREPARE FOR EXAMS SUCH AS GRE
- 8) GAMIFICATION

# **UML DIAGRAMS**

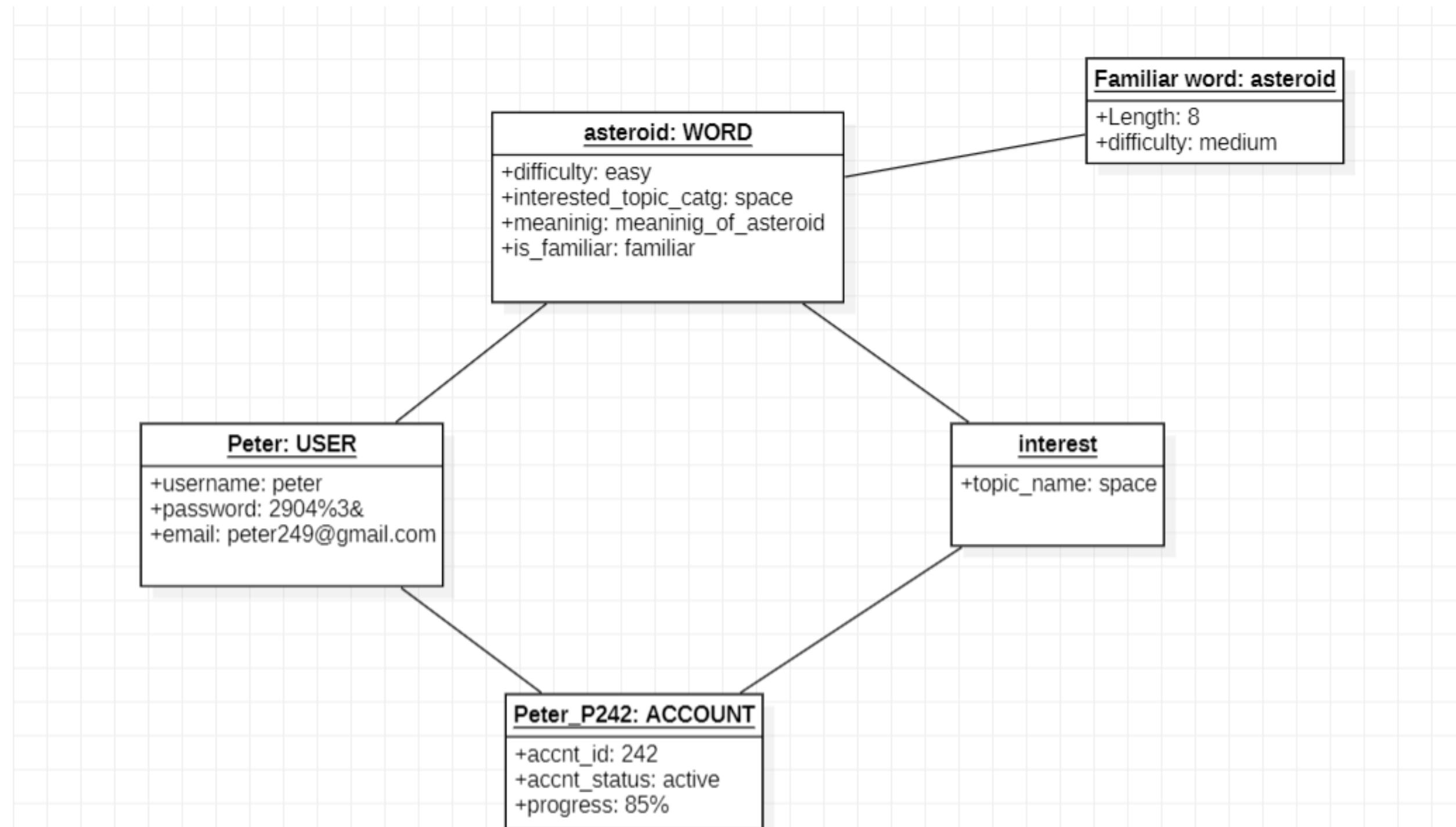
# USE CASE



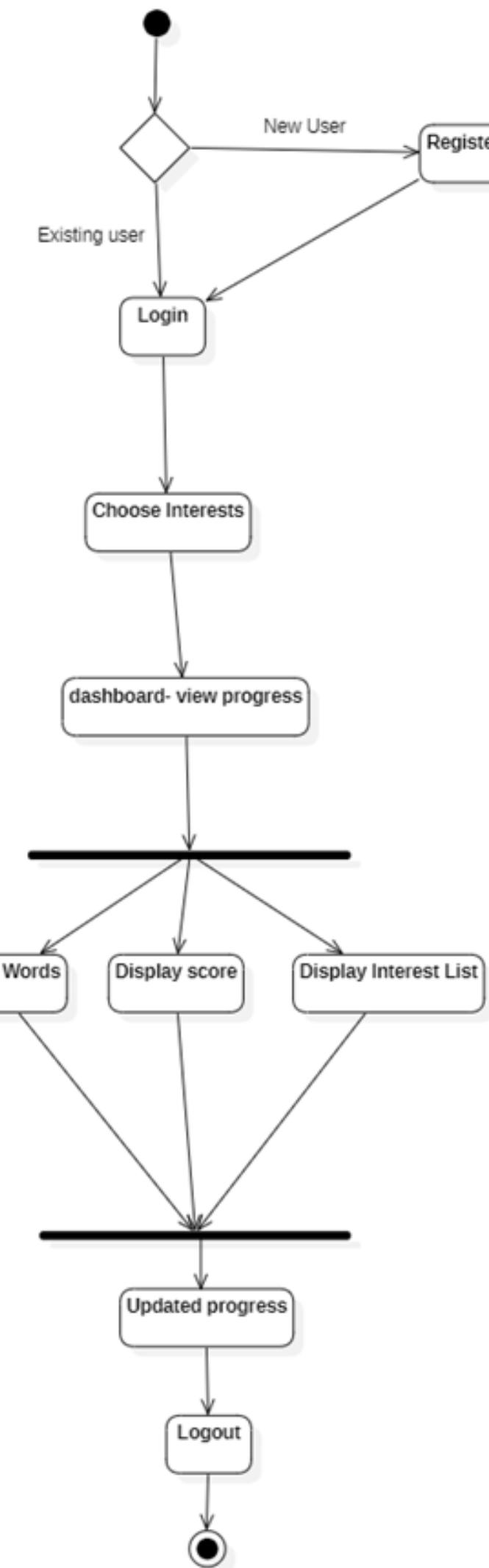
# CLASS DIAGRAM



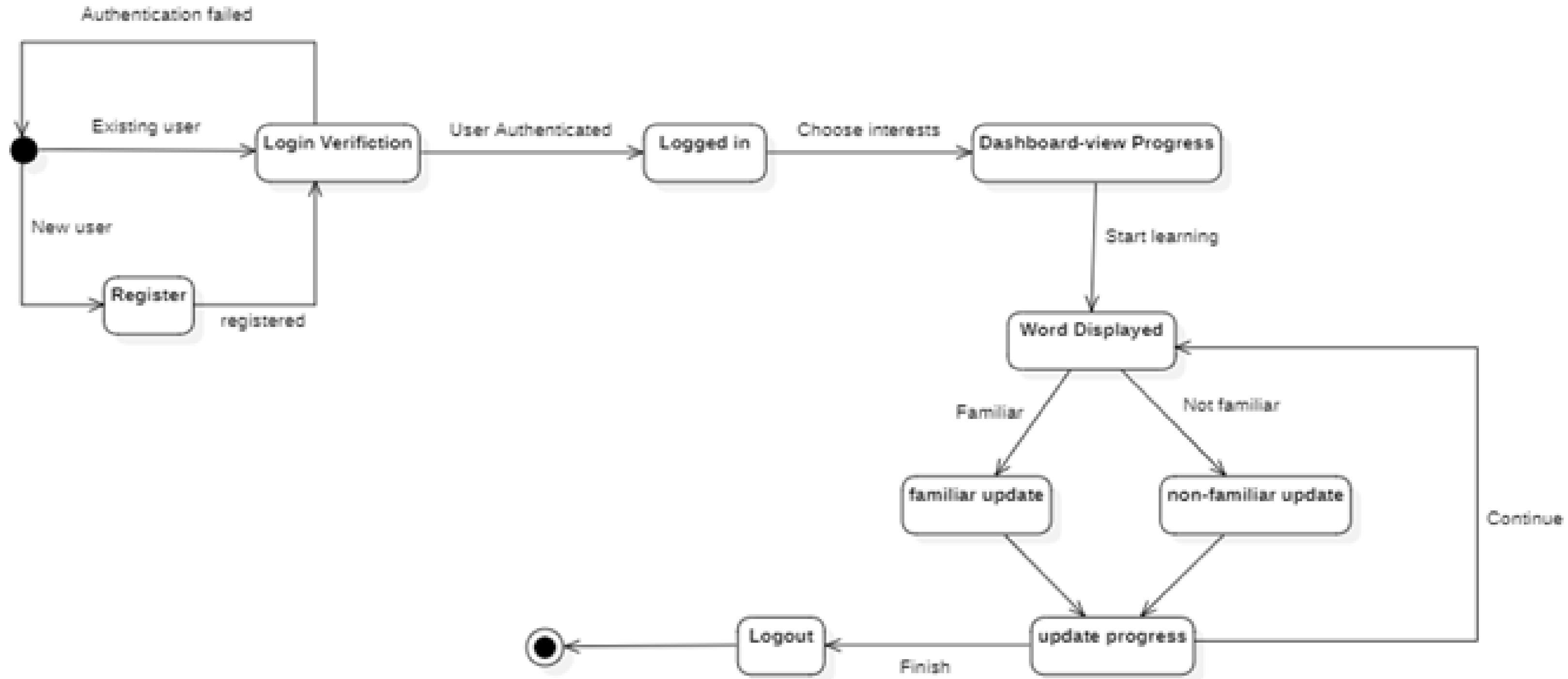
# OBJECT DIAGRAM



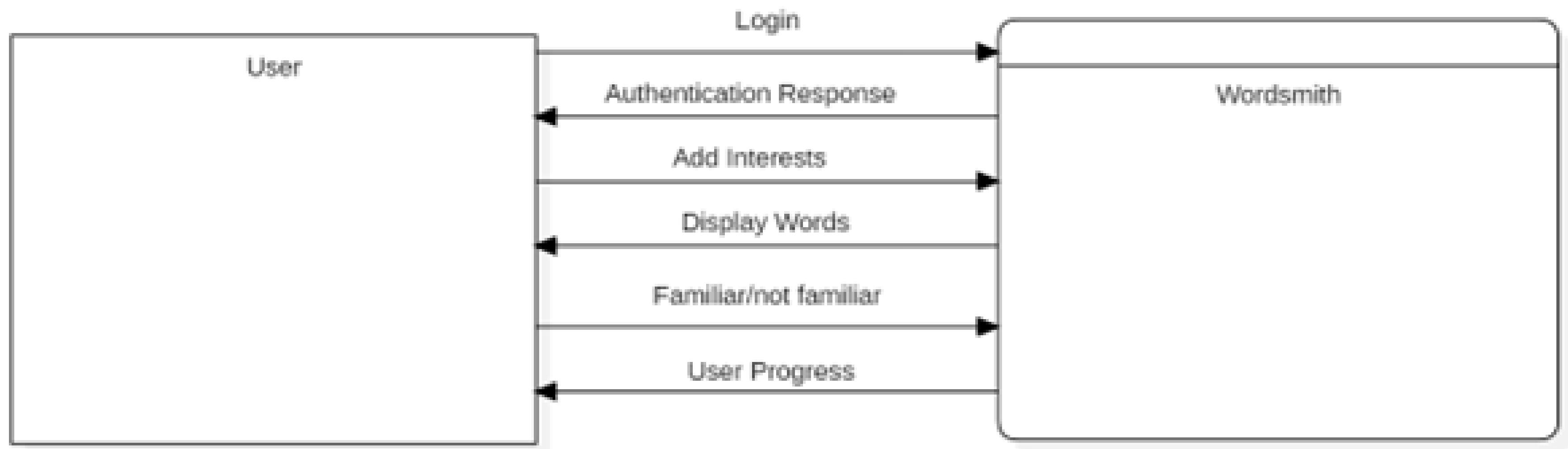
# ACTIVITY DIAGRAM



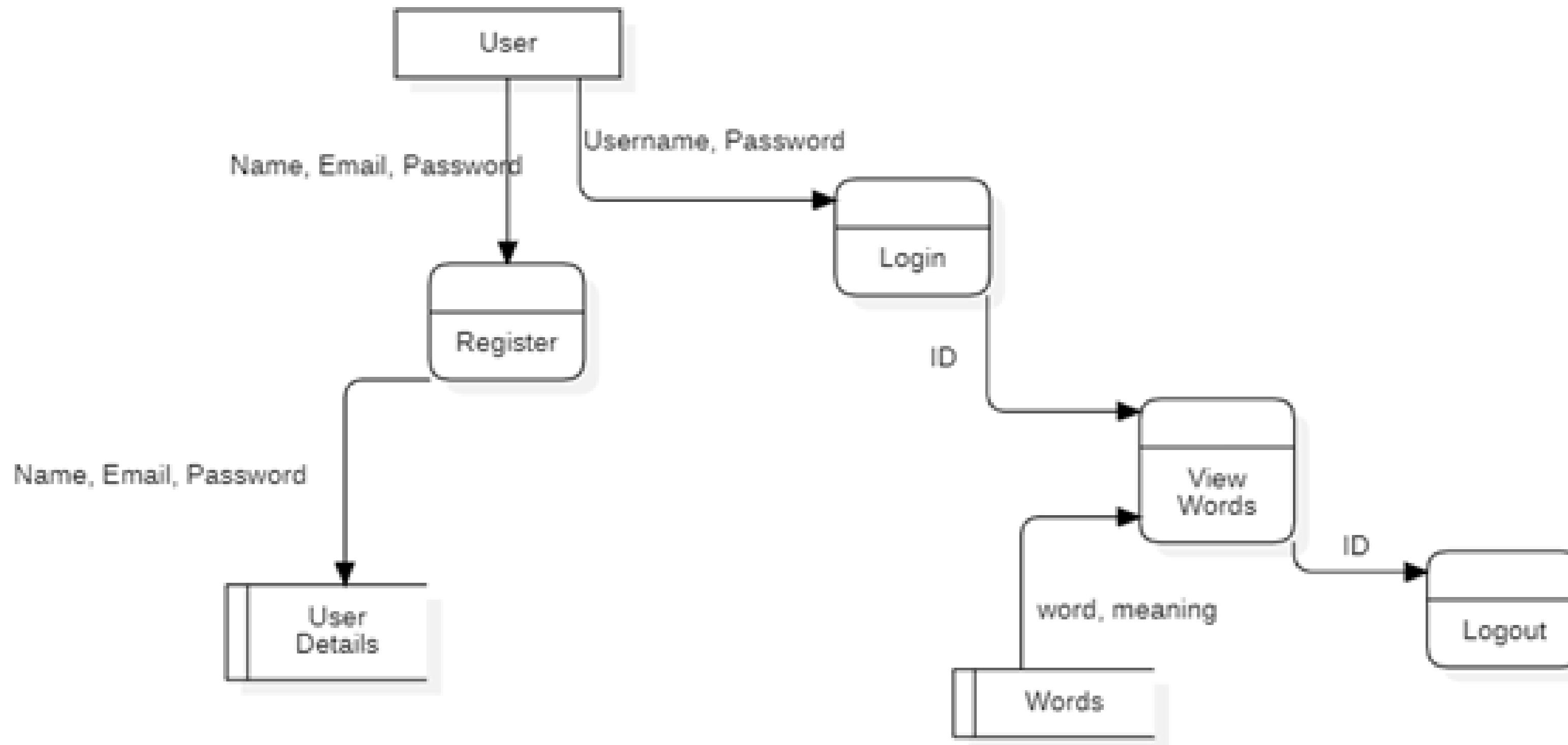
# STATE DIAGRAM



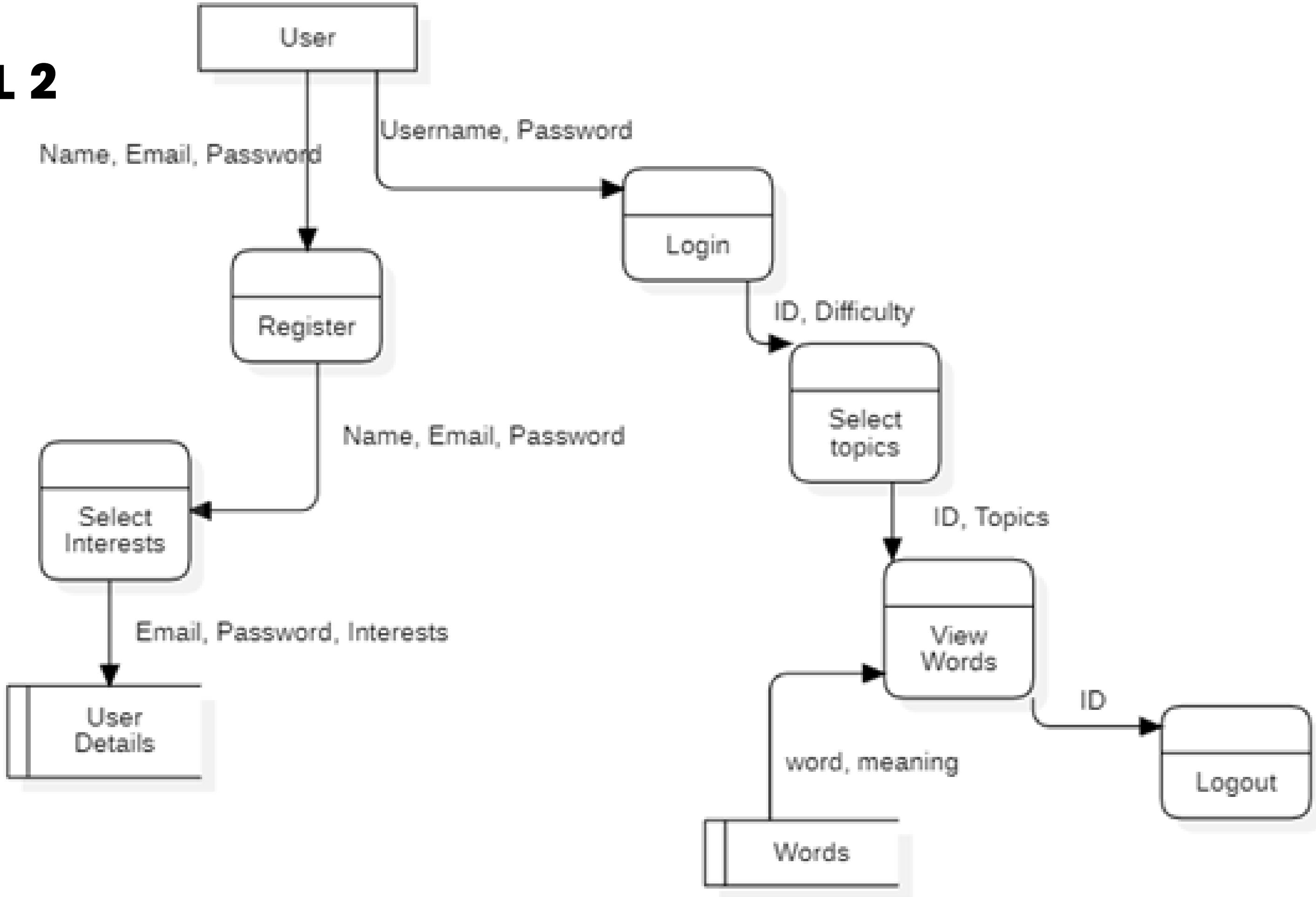
# DFD LEVEL 0



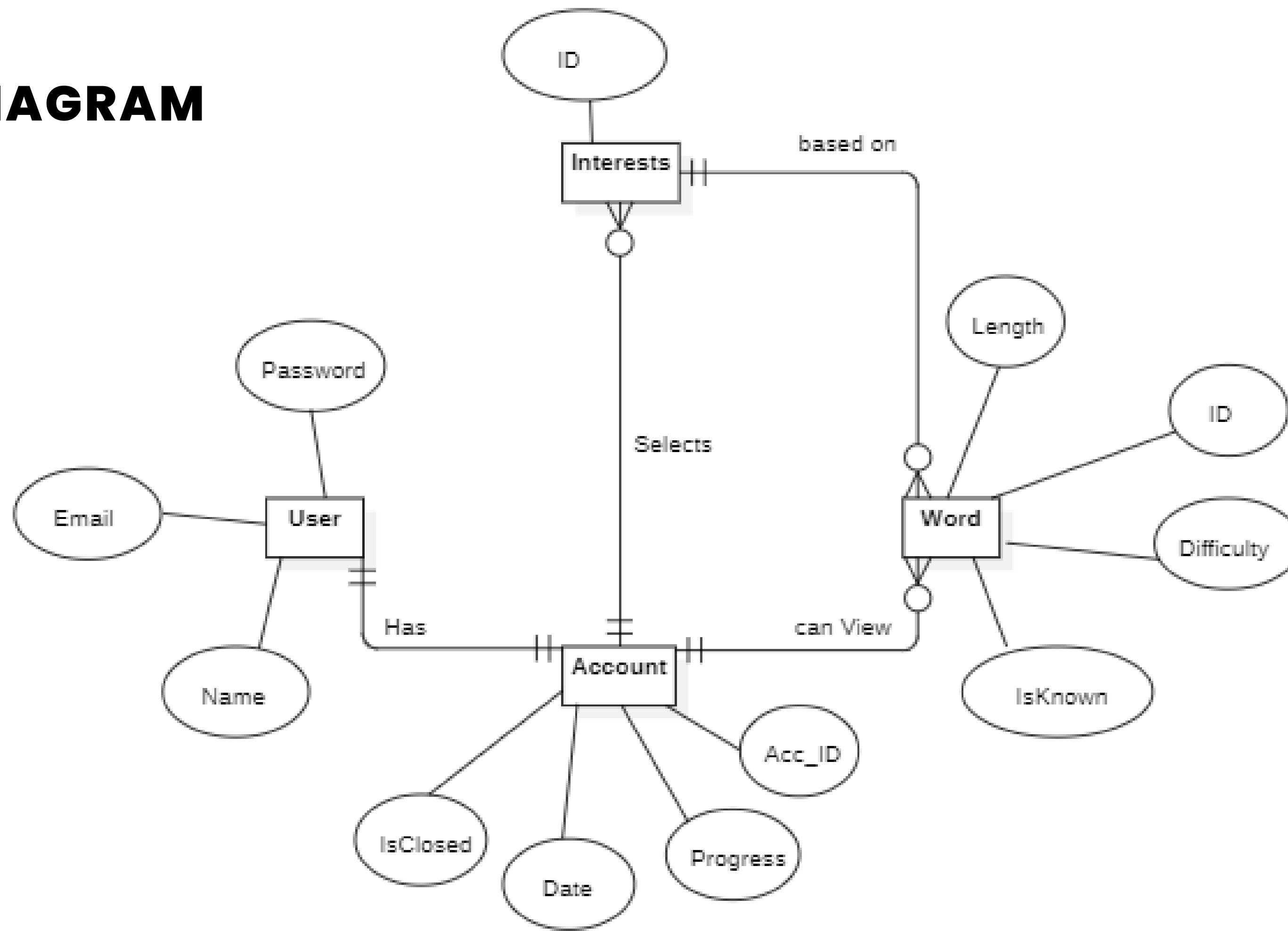
# DFD LEVEL 1



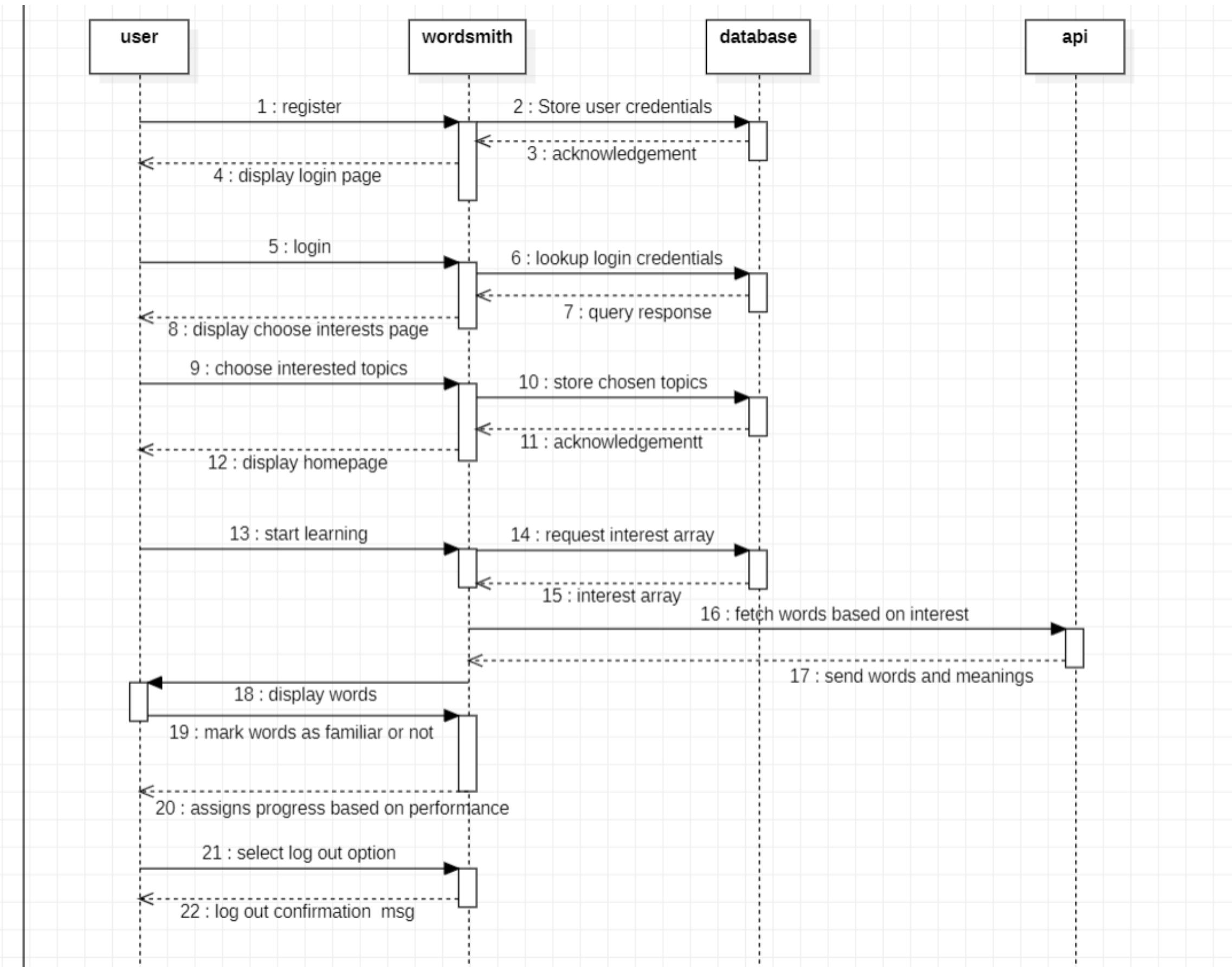
## DFD LEVEL 2



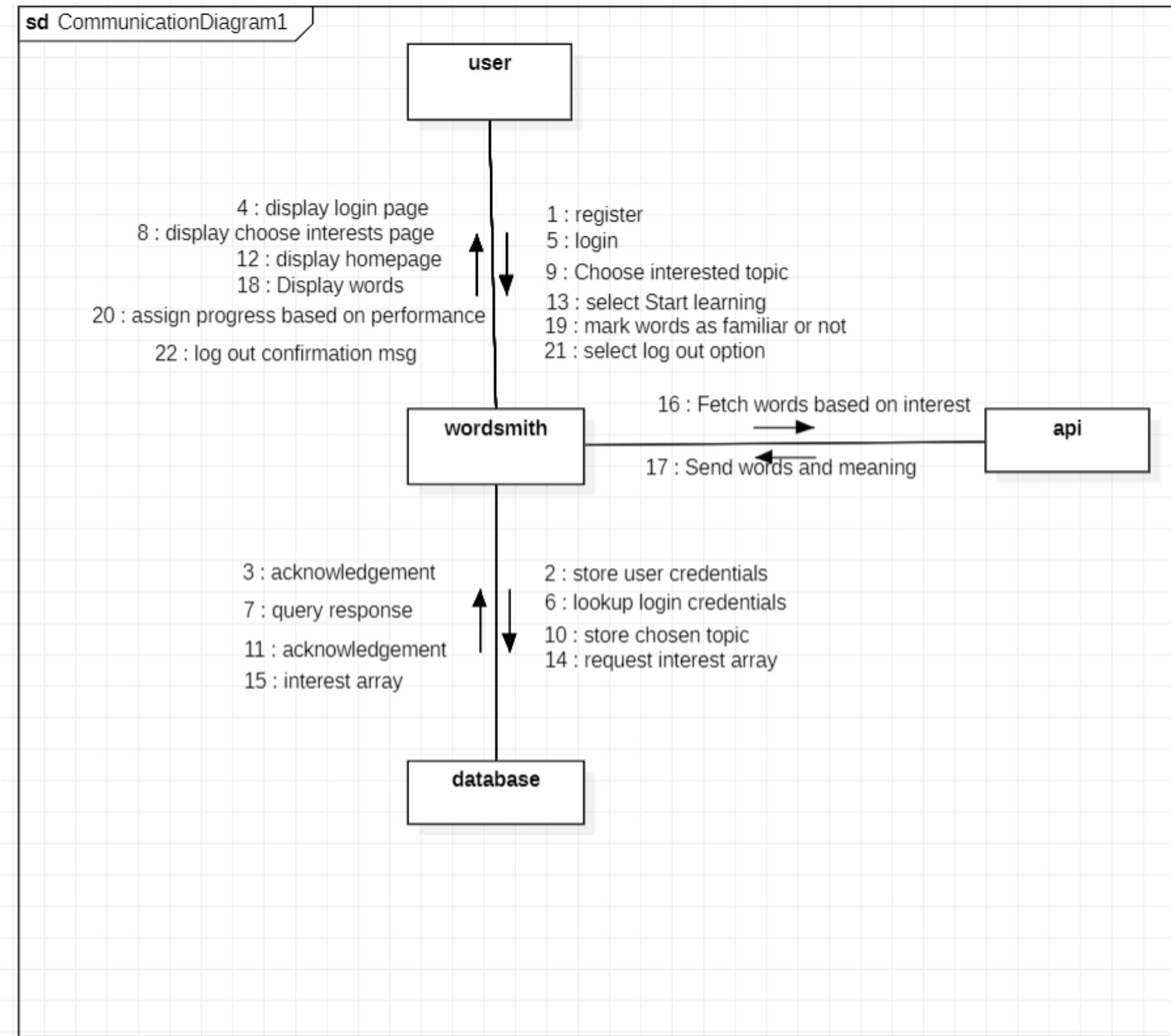
# ER DIAGRAM



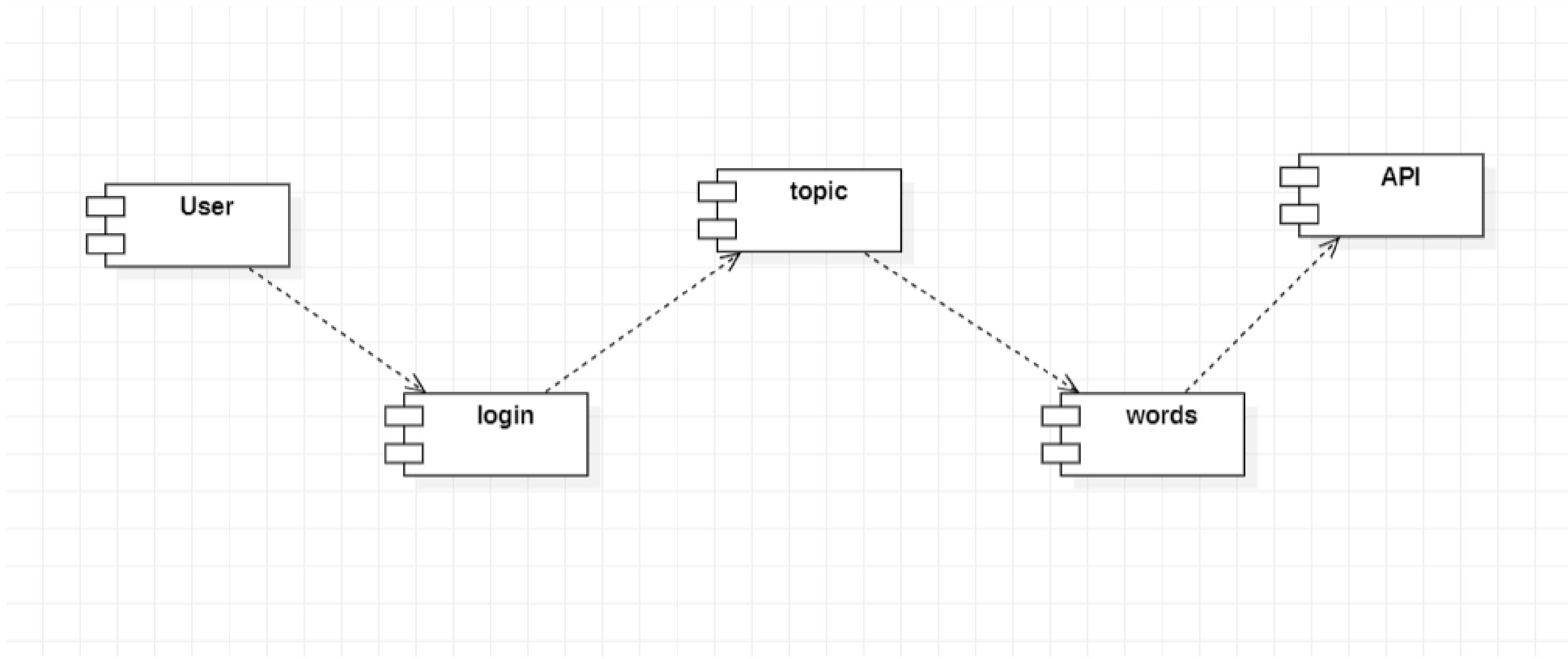
# SEQUENCE DIAGRAM



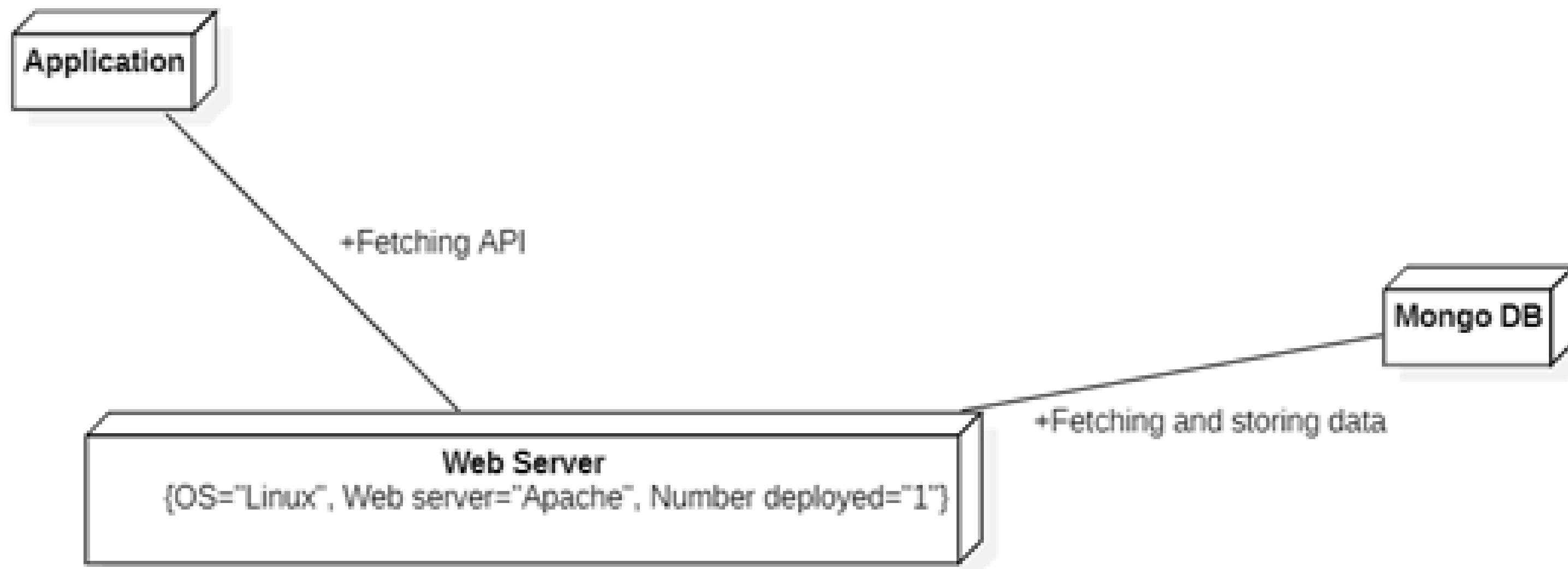
# COLABORATION DIAGRAM



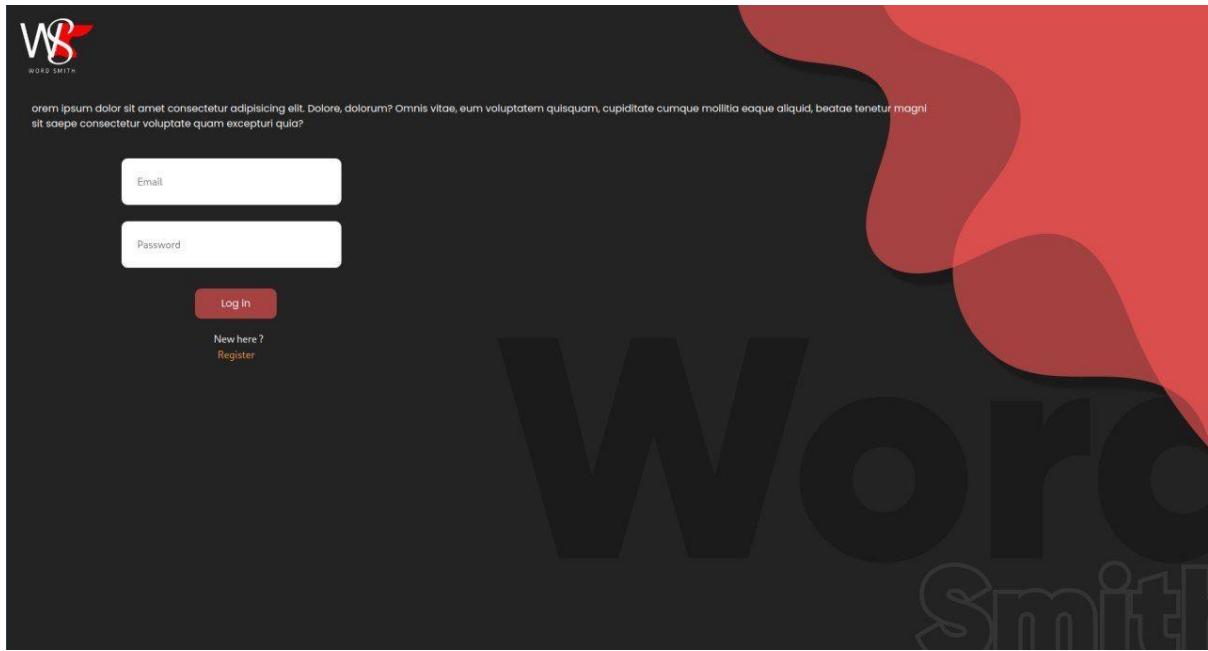
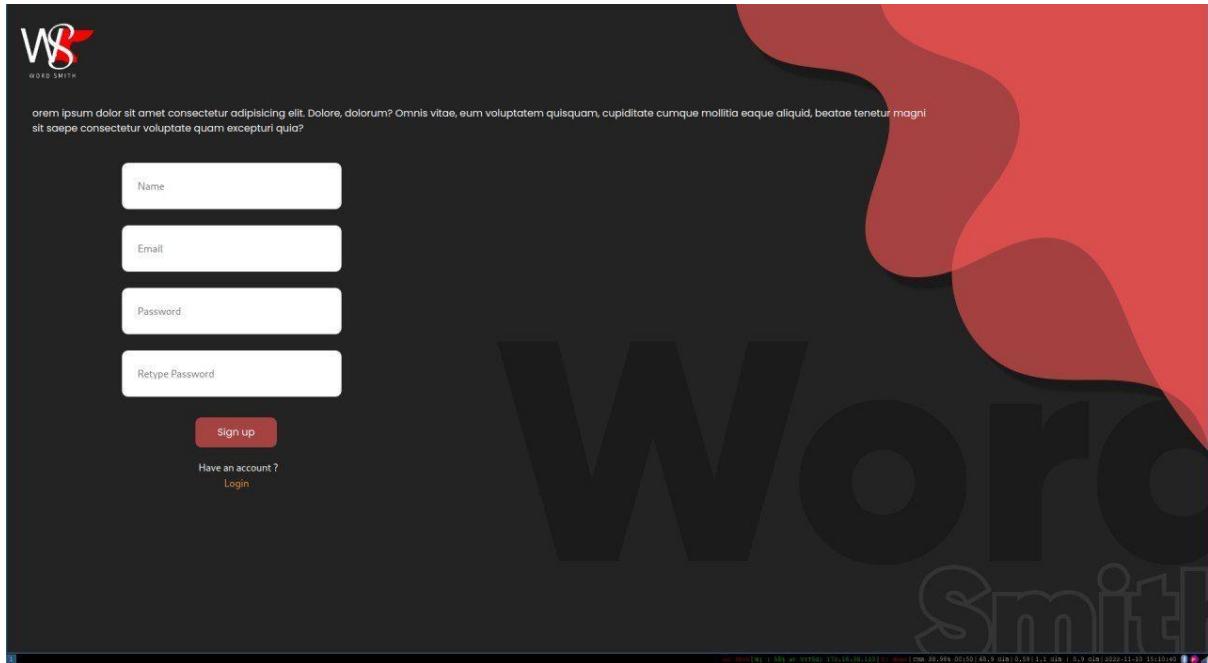
# COMPONENT DIAGRAM

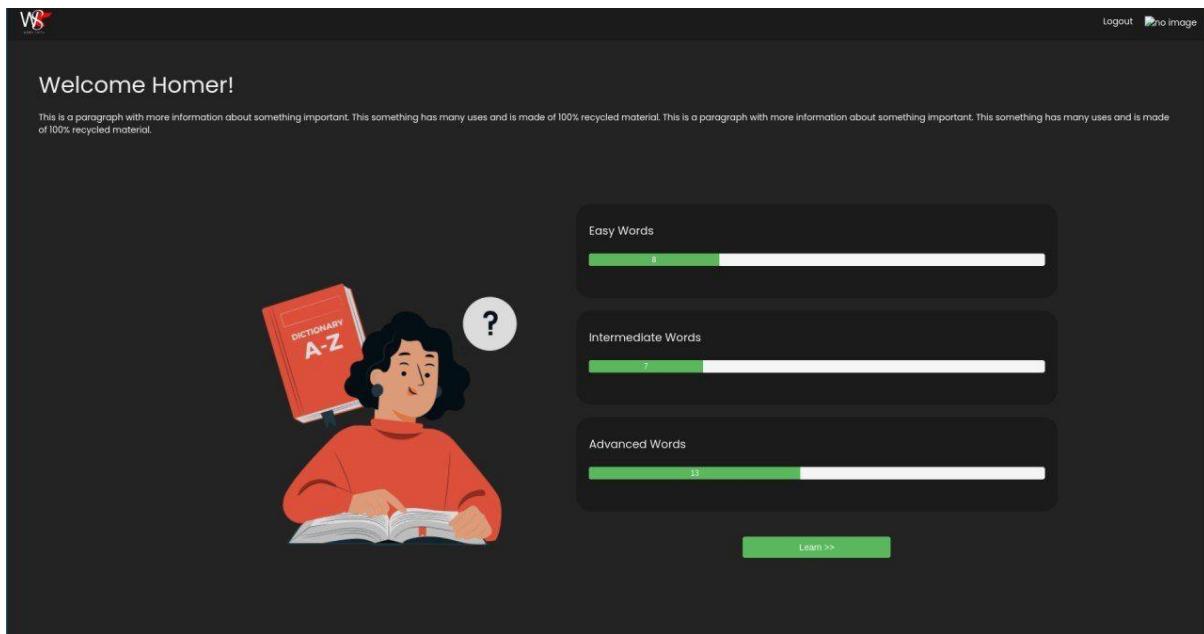


# DEPLOYMENT DIAGRAM



# Wordsmith : Project Screenshots





WS

Logout

Navigation

Home

Challenge me

Go Solo :

- Politics
- Machine Learning
- Artificial Intelligence
- Blockchain

political relation

n social relations involving authority or power

Familiar Not Familiar

0 0

known Unknown

**Github Link :**

<https://github.com/AaronMathew11/Wordsmith>