



WORD MASTER 3000

Software Requirements Specification



Submitted by

Syed Yusuf Ahmed (Exam Roll **1016**)

Radowan Mahmud Redoy (Exam Roll **1001**)

Supervised By:

Dr. Kazi Muheymin-Us-Sakib
Professor

AUGUST 28, 2021

Table of Contents

1 INTRODUCTION	1
1.1 Purpose.....	1
1.2 INTENDED AUDIENCE.....	1
1.3 CONCLUSION	2
2. WORD MASTER 3000 INCEPTION	2
2.1 INTRODUCTION.....	2
2.2.1 Identifying the client of our project.....	3
2.2.2 ICEBREAKING	3
2.2.3 IDENTIFYING THE STAKEHOLDERS OF A COURSE.....	3
2.2.4 IDENTIFYING THE MULTIPLE VIEWPOINTS OF THE	4
3 ELICITATION OF WORD MASTER 3000	5
3.1 INTRODUCTION.....	5
3.2 ELICITING REQUIREMENTS.....	5
3.2.1 COLLABORATIVE SOLUTION	5
3.2.2 QUALITY FUNCTION DEPLOYMENT	6
3.2.2.1 NORMAL REQUIREMENTS	6
3.2.2.2 EXPECTED REQUIREMENTS.....	6
3.2.2.3 EXCITING REQUIREMENTS.....	7
3.2.3 USAGE SCENARIO	7
4. SCENARIO BASED MODELING OFWORD MASTER 3000	10
4.1 INTRODUCTION.....	10
4.2 USE CASE Diagram	10
Level 0	11
Level 1	11
Level 1.1.....	12
Level 1.1.1.....	13
Level 1.1.2.....	14
Level 1.1.3.....	14
Level 1.2.....	14
Level 1.2.1.....	15
Level 1.2.1.1.....	16
Level 1.2.1.2.....	16
Level 1.2.3.....	16
Level 1.2.2.....	17

Level 1.2.2.1.....	17
Level 1.2.2.2.....	18
Level 1.3.....	18
Level 1.3.1.....	19
Level 1.3.1.1.....	19
Level 1.3.1.2.....	20
Level 1.3.1.3.....	20
Level 1.3.1.4.....	20
Level 1.3.1.5.....	20
Level 1.3.2.....	21
Level 1.3.2.1.....	21
Level 1.3.2.2.....	22
Level 1.3.2.3.....	22
Level 1.3.3.....	22
Level 1.4.....	22
Level 1.4.1.....	23
Level 1.4.1.1.....	24
Level 1.4.1.2.....	24
Level 1.4.1.3.....	25
Level 1.5.....	25
Level 1.5.1.....	26
Level 1.5.2.....	26
Level 1.5.2.1.....	27
Level 1.5.2.2.....	27
Level 1.5.2.2.....	27
Level 1.5.2.4.....	28
Level 1.6.....	28
4.3 Activity Diagram.....	29
4.3.1 Application.....	29
4.3.2 Authentication	30
4.3.3 Dictionary Words	31
4.3.3.1 Search	32
4.3.4 Memorization	33
4.3.4.1 Quiz.....	33
4.3.4.2 Memorization.....	35
4.3.4.3 Daily Training	36

4.3.5 Spelling Master	37
4.3.6 Market Place	38
4.3.6.1 Forum.....	39
4.3.6.2 Share wordlist	40
4.4 Swimlane Diagram of WM 3000.....	41
4.4.1. Authentication	41
4.4.2 Dictionary Words	42
4.4.3 Quiz.....	43
4.4.4 Daily Training	44
4.4.5 Memorize.....	45
4.4.6 Spelling Master	46
4.4.7 Forum	47
4.4.8 Share Wordlist	48
5. Data Based Modeling for Word Master 3000.....	49
5.1 ER DIAGRAM (Dictionary database)	49
5.2 Schema Diagram (Dictionary database)	50
5.3 Data objects (Cloud storage)	51
6. Class-Based Modeling for Word Master 3000	52
6.1 CLASS BASED MODELING CONCEPT	52
6.2 GENERAL CLASSIFICATION	52
6.3 General Classification of Word Master 3000.....	53
6.4 SELECTION CRITERIA	54
6.5 Associate Noun and Verb identification	55
6.6 Identify Attributes.....	57
6.7 Identify Methods	58
6.8 Analysis.....	59
6.9 Class cards	61
6.10 Class Modeling Diagram.....	67
7. BEHAVIORAL MODELING OF WORD MASTER 3000	68
7.1 STATE TRANSITION.....	68
7.1.1 Events after analysis:.....	70
7.1.2 State Transition Diagram:.....	72
7.2 Sequence Diagram	81

WORD MASTER 3000

Mastery Over English Words

1 INTRODUCTION

This chapter is a part of our software requirement specification for the project “Word Master 3000”. In this chapter, we have focused on the intended audience for our project “Word Master 3000”

1.1 Purpose

This document describes the Software Requirement Analysis of Word Master 3000 application. It contains all the functional, non-functional, supporting requirements and expected requirements. At the same time it establishes a requirement’s baseline for the development of the project. The requirements contained in the SRS are independent, uniquely numbered and organized by topics. But as time goes on, our SRS document is expected to evolve as users and developers work together to validate, clarify and expand its contents. The SRS serves as an official medium of expressing users’ requirements to the developer and provides a common reference point for both the developer team and the stakeholder community.

1.2 INTENDED AUDIENCE

This SRS report is intended for several audiences which includes the end users as well as the project managers, designers, developers, and testers.

- The work of the developer team that means the product which they have created can easily be verified by the customer whether it is acceptable to them or not by using this SRS document of Word Master 3000.
- Project Managers will be able to plan milestones, delivery date and at the same time ensure whether the developing team is on track or not during the development of the application by using this SRS document.
- The designers will use this SRS as a basis for creating the application’s design. The designers will continually refer back to this SRS to ensure that the system they are designing will fulfill the customer’s needs.

- For developing the system's functionality, this SRS document will be used by the developers. The developers will link the requirements defined in this SRS to the software they create to ensure that they have created a software that will fulfill all of the customers' documented requirements.
- The testers will use this SRS in various phases of testing. They will use this SRS to derive test plans and test cases for each documented requirement. When portions of the software are complete, the testers will run their tests on that software to ensure that the software fulfills the requirements documented in this SRS. The testers will again run their tests on the entire system when it is complete and ensure that all requirements documented in this SRS have been fulfilled.

1.3 CONCLUSION

This analysis of the audience helped us to focus on the users who will be using our software requirements analysis. This overall document will help each and every person related to this project that includes users, project managers, designers, developers, testers, stakeholders to have a better idea about the project

2. WORD MASTER 3000 INCEPTION

The inception part of our SRS is briefly discussed in this part:

2.1 INTRODUCTION

Word Master 3000 is a mobile based application which aims to help/guide the user achieve mastery over English words. The app helps users learn spelling, word meaning, multiple meanings of words, use of words in sentences, synonyms and antonyms. This can be achieved using standard memorization techniques, short training session, games and quizzes.

2.2 INCEPTION OF The Project

At first, we have entered into the inception stage. This stage includes, how our project will start and what are the scopes and limitations. The main goal of this phase is to identify the requirements & demands and then establish some sort of mutual understanding between the software team and the customers. They are the intended users for our software. In order to make this phase effective we took the following steps:

- Identifying the client of our project
- Icebreaking
- Identifying the stakeholders of a course
- Identifying the multiple viewpoints of the stakeholders

2.2.1 Identifying the client of our project

At first, we have identified the location from where we will start our expedition. Normally teachers and students from various organizations along with any potential user looking to improve English skills will act as a stakeholder. But there are other things related with this as well. So, we have to go through a systematic approach in order to identify all stakeholders. But over all people from all ages are expected to be our clients. We have analyzed our requirements with the consent of all types of users.

2.2.2 ICEBREAKING

Icebreaking refers to the fact that to diminish the communication barrier between you and the other person. It is a crucial part since it denotes the acceptance of our proposal. We started this phase by talking with various students from universities and colleges with context free languages. The students informally expressed their expectations. We also carried out informal meeting with a few teachers from different institutions. The response from the potential users were positive and looks forward to use the applications.

2.2.3 IDENTIFYING THE STAKEHOLDERS OF A COURSE

Stakeholder refers to any person or group who will be affected directly or indirectly by the system. Stakeholders include end-users who interact with the system and everyone else in an organization who may be affected by its

installation. The institutions that we have visited and teachers we have met indicates that Word Master 3000 has a few stakeholders only. Identification of the stakeholders were done from the information provided by the potential users from various institutions. The stakeholders of our system are given below:

- Learner
- Instructor

2.2.4 IDENTIFYING THE MULTIPLE VIEWPOINTS OF THE STAKEHOLDER

Different stakeholders expect different benefits from the system as every person has his own point of view. So, we have to recognize the requirements from multiple viewpoints. Different viewpoints of the stakeholders about the expected software are given below:

- Learner's View Point
 - Smartphone based system
 - Easy access to words and meanings.
 - Various techniques to improve skill
 - Open platform for discussion.
 - Getting the chance to practice and learn at the same time
 - Friendly user interface
- Instructor's View Point
 - Simple Course Management.
 - Easy user interface to interact.
 - Online performance evaluate.
 - Storing result against student's profile.
 - Being able to share opinion.
 - Making sure student can find appropriate course

2.2.3 Conclusion

Our primary goal is to design a software which will make learning English easier and more efficient, not only for general students but also for people of all ages. At the same time, where students can get what information, they need or raise questions in any fact related to the learning process. Can explore different memorization techniques in order to memorize the word. And all of this through an easy way. For these reasons, the software will be designed in such a way that it won't be complicated for both the learners and instructors who will use it. The

software will be so simple that a instructor who does not have any idea about software, can be able to maintain it without any annoyance. Otherwise, it will not meet its goal. That is to make the learning process easy and efficient.

3 ELICITATION OF WORD MASTER 3000

So far, we have discussed the Inception phase of our project. Now we need to focus on the Elicitation phase. So this chapter specifies the Elicitation phase.

3.1 INTRODUCTION

Elicitation is a part of requirements engineering that is. We have faced many difficulties, like understanding the problems, making questions for stock holders, getting appointments from stockholders & instructors in spite of their busy schedule, making them understand the power of such innovative applications in this sector. Despite not being easy to gather requirements within a very short time, we have overcome all problems in a systematic manner. We have done several meetings and finalized the requirements for our software.

3.2 ELICITING REQUIREMENTS

Elicitation phase is mainly combining the elements for problem solving, elaboration, negotiation and specification. Without the collaboration of the stakeholder eliciting would have been really hard. We have finished the following tasks for eliciting requirements-

- Collaborative Solution
- Quality Function Deployment
- Usage Scenarios
- Elicitation work products

3.2.1 COLLABORATIVE SOLUTION

We have met with many instructors and learner in the Inception phase. These meetings created an indecisive state and we could identify what needs to be done in order to make learning, practicing and memorizing English easier. To solve this problem, we have met with the stakeholders several times and came up with the solution which eventually helped us to elicit the requirements.

3.2.2 QUALITY FUNCTION DEPLOYMENT

QFD, Quality Function Deployment, is a technique that translates the needs of the customer into technical requirements for software. QFD mainly translates subjective quality criteria into objective ones that can be quantified and measured and which can then be used to design and manufacture the product. It is a methodology concentrating on maximizing customer satisfaction from the software engineering process. So, we have followed this methodology to identify the requirements for the project. The requirements identified successfully by the QFD are given below

3.2.2.1 NORMAL REQUIREMENTS

The requirements which are normally stated by the customer in various meetings are the normal requirements of a system, In general sense the objectives and goals that are stated for a product or system during meetings with the customer are normal requirements. The presence of these requirements fulfills customers' satisfaction. The normal requirements for our project are given below

- Simple and user-friendly interface
- Smartphone based application
- Easily accessible for all
- Word meaning with proper example
- Synonyms and various usage of words
- Quiz system for evaluation
- Open discussion
- Bookmark words
- Create course (Special wordlist)
- Share course (Special wordlist)
- Enroll in course (Special wordlist)

3.2.2.2 EXPECTED REQUIREMENTS

These requirements are intrinsic to the system and may be so elementary that the customer does not explicitly state them. But the absence of this requirements would result in dissatisfaction. The expected requirements are given below

- Proper dictionary database
- Being able to search words offline
- Authentication
- Scalability
- Maintain user records

3.2.2.3 EXCITING REQUIREMENTS

These requirements are for features that go beyond the customer's expectations and prove to be very satisfying. These are the exciting features of our project

- Games that make learning fun
- Leaderboard
- Daily Training session
- Memorization techniques
- Storing wordlist
- Smart Suggestions
- Automated quiz generation
- Search on copy

3.2.3 USAGE SCENARIO

Word Master 3000 is a mobile based application which aims to help/guide the user achieve mastery over English words. The app shall help users learn spelling, word meaning, multiple meanings of words, use of words in sentences, synonyms and antonyms; using standard memorization techniques, short training session, games and quizzes. The app will have the following features:

Authentication:

Sign up: To access all features, the user must create an account. The user needs to provide his/her name, email, unique username, password.

Log-in: The user can log-in using his username (or email) and password.

Recover account: If the user forgets his password, it can be recovered by sending a recovery link to his email.

Dictionary:

There will be an offline dictionary for the users. It will show word meaning, parts of speech, pronunciation, and example sentences. There will be a check box with

every meaning the word has. The user can select the specific meanings that interest him. He / She can also select all the meanings. Words with n different meanings will be treated as n different words with one meaning. But all the meanings of a word will be grouped together.

Search

The user will be able to search for different words. To make the process easy and efficient there will be two types of ways for giving input

Input in app: The user can directly type or paste any word in the word to search for any word. It is basically manual input direct in the app.

Search on copy: The user copies the word in his clipboard from any app in his phone. A pop up will show him word meaning.

Storing:

The user can search for words and store them (in database). Words can be stored in lists. Each user can create multiple lists. Each list must be named by the user. Lists can be shared with other users. When the user saves a list, it is saved as private but when the user shares his/her word to the market place, the list then it will no longer be private. It will become publicly downloadable.

Memorization:

Memorization techniques discussed below will be applied on user's lists.

Memorization and review word meaning by repetition: The user will be shown a word in a sentence. He will have to recall the meaning and check it with the answer. If he gets it right, he will tick mark it or else cross it. This will be repeated but only with the words that he has crossed.

Quiz: Users can take part in quizzes in order to judge their skills. Quizzes composed of MCQ.

Smart suggestion: Questions for daily training will be set based on the user's previous quiz result. Suggestions will be generated by the server.

Daily training: A short training session will be automatically generated everyday based on users' previous quiz and memorization results.

Easy access: Word meanings can easily be accessed from any application.

Gamification:

The app will allow users to learn through playing games.

Spelling master: The player will be shown meaning and pronunciation. He will have to guess the spelling of the word. This will help users prepare for international spelling bee competitions.

Leaderboard:

Players will find themselves in the global leaderboard based on the points that they score in games and quizzes.

Market Place

There will be a marketplace for the user. The user will be able to access the market place via login. The user can share his own word list and add tags to it. All shared word lists will be stored in the server. These word lists will appear by the name “courses”. When the user shares his word list in the marketplace then it is visible to all users. The users will be able to view and download the wordlist.

Forum

Users can have open discussions in these sections. They can share their own thoughts and problems and other users can comment on it sharing their opinion and solutions. Forum is open to all users.

4. SCENARIO BASED MODELING OF WORD MASTER 3000

This chapter describes the Scenario Based Model for “Word Master 3000”

4.1 INTRODUCTION

Although the success of a computer-based system or product is measured in many ways, user satisfaction resides at the top of the list. If we understand how end users (and other actors) want to interact with a system, our software team will be better able to properly characterize requirements and build meaningful analysis and design models. Hence, requirements modeling begins with the creation of scenarios in the form of Use Cases, activity diagrams and swim lane diagrams.

4.2 USE CASE Diagram

Use case defines the stylized story about how an end user interacts with the system under a specific set of circumstances. A Use Case diagram simply describes a story using corresponding actors who perform important roles in the story and makes the story understandable for the users. The first step in writing a Use Case is to define that set of “actors” that will be involved in the story. Actors are the different people that use the system or product within the context of the function and behavior that is to be described. Actors represent the roles that people play as the system operators.

Primary Actors:

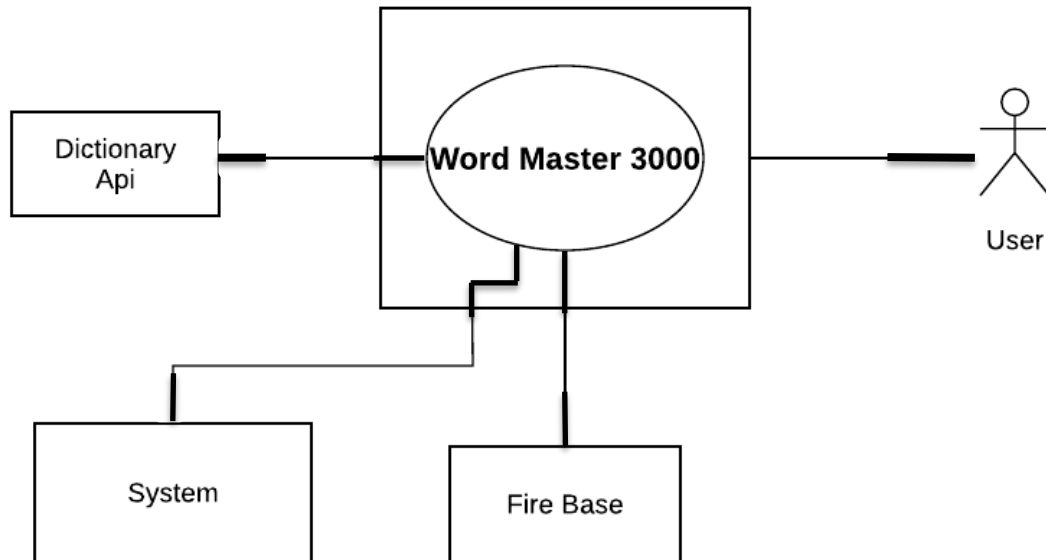
As word master 3000 incorporates only instructors and learners and the in most case have same functionality so we can define them as user. So, the system has one primary actor: user.

Secondary Actor:

The system also acts as an internal actor in some cases. Therefore, it is the secondary actor of our system. Outside system: Dictionary, Firebase.

Level 0

Use case Name: Word Master 3000



Use case Id: 0

Fig 1: Level 0(use case diagram)

Primary Actor: User

Secondary Actor: Dictionary Api, Fire Base, System

Description: This use case shows the low-level interaction between system and actors.

Level 1

Use case Name: WORD MASTER 3000 SUB SYSTEM

Use case Id: 1

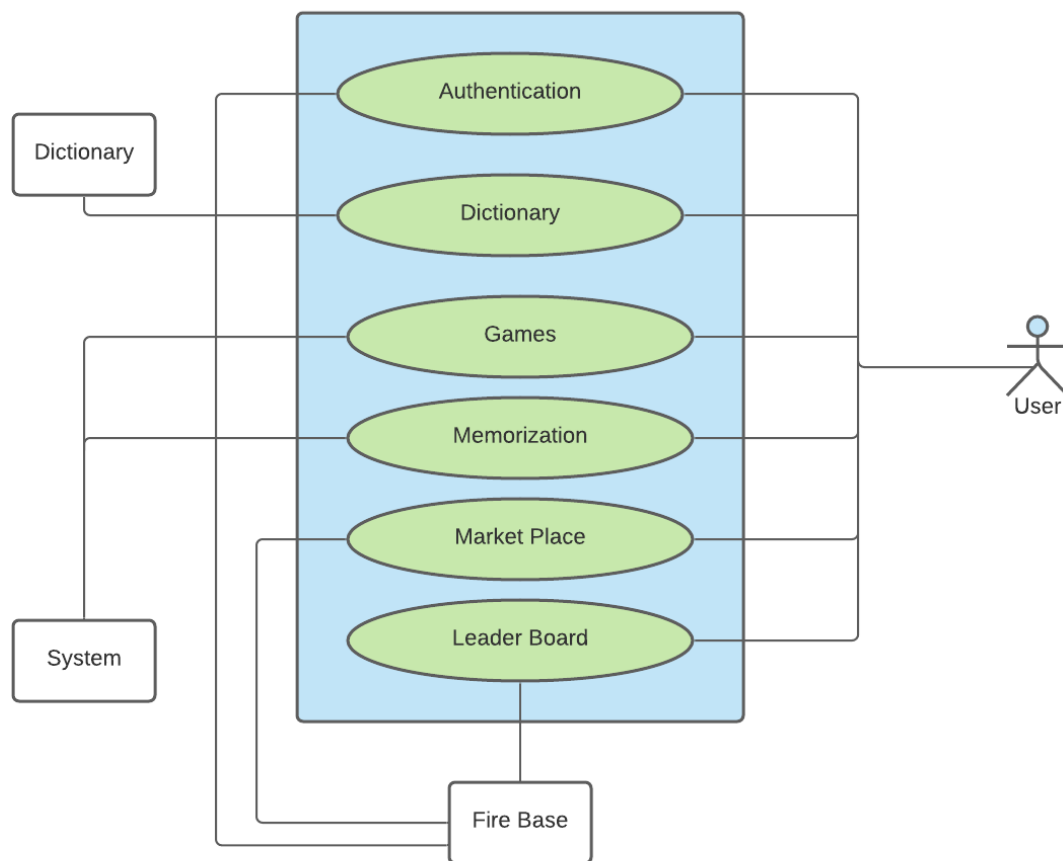


Fig: Level 1 (Use case diagram)

Primary Actor: User

Secondary Actor: Dictionary Api, Fire Base, System

Description: From this level all the subsystem of the proposed main system and connectivity of those subsystem through actors has been explicit. From this level interaction between actors and subsystem will be clearer. Here, the whole system is divided into six subsystem and Fire base is the outside system in this proposed system

Level 1.1

Use case Name: Authentication

Use case Id: 1.1

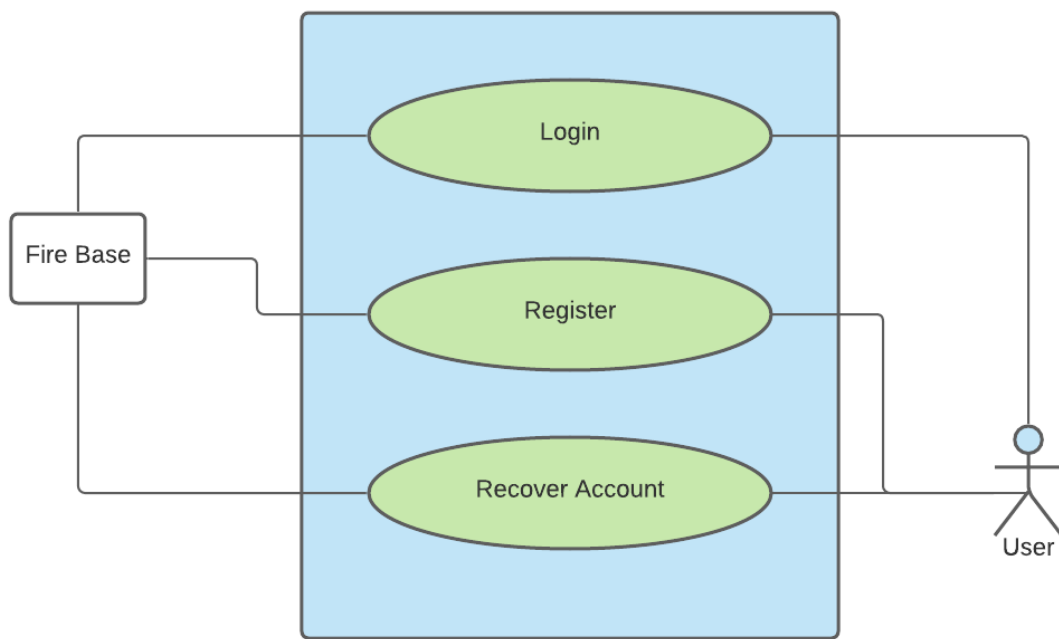


Fig: Level 1.1 (Use case diagram)

Primary Actor: User

Secondary Actor: Fire Base

Description: The whole authentication process has been explicit in this level in four more sub systems. They are Registration, log in & recover info.

Level 1.1.1

Use case Name: Login

Primary Actor: User

Secondary Actor: Fire Base

Description:

Action: The user provides user name and password or asks for anonymous login.

Reply: User is logged in.

Level 1.1.2

Use case Name: Register

Primary Actor: User

Secondary Actor: Fire Base

Description:

Action: The user provides name , email, user name ,password .

Reply: Data is saved in firebase and user is logged in.

Level 1.1.3

Use case Name: Recover Account

Primary Actor: User

Secondary Actor: Fire Base

Description:

The user asks to recover account if password or user name is forgotten.

Firebase uses its verification process to verify user and recovers account.

Level 1.2

Use case Name: Dictionary

Use case Id: 1.2

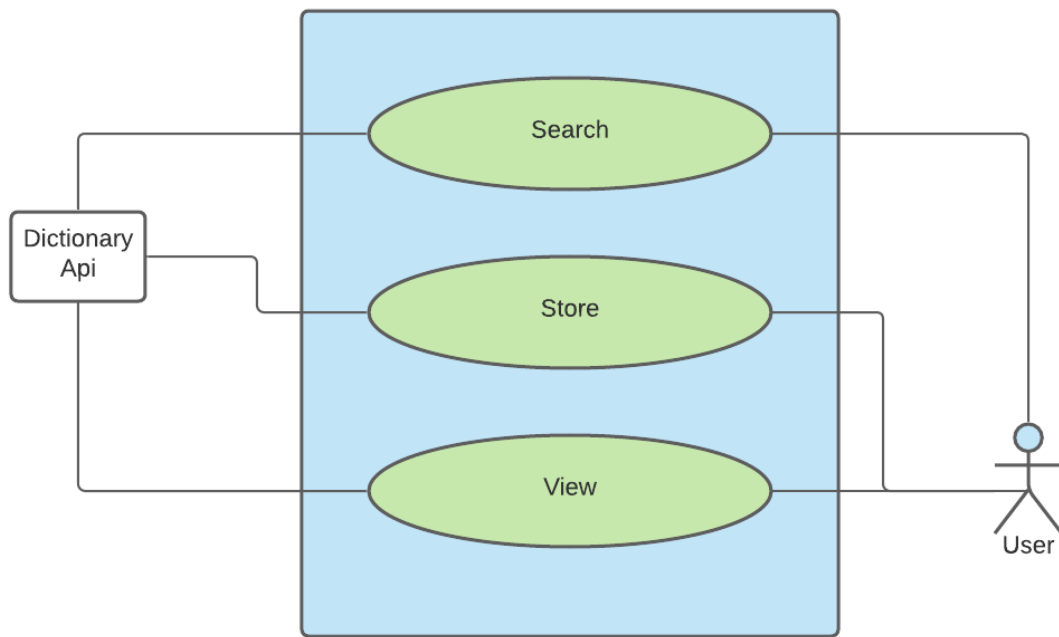


Fig: Level 1.2 (Use case diagram)

Primary Actor: User

Secondary Actor: Dictionary Api

Description:

The sub system is divided into three parts. Search , store and view.

Level 1.2.1

Use case Name: Search

Use case Id: 1.2.1

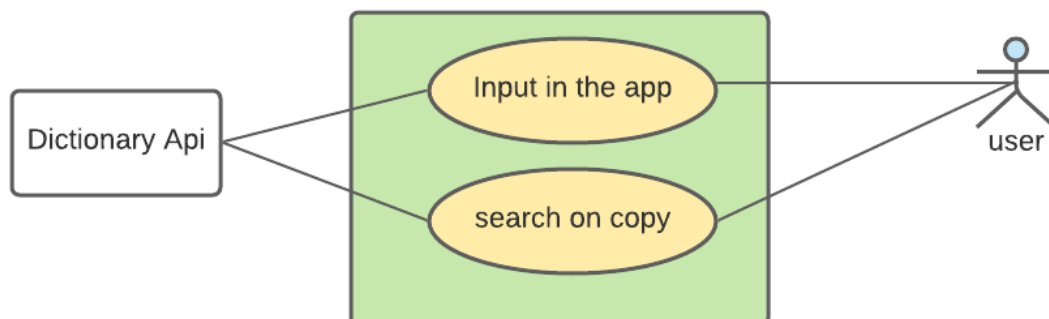


Fig: Level 1.2.1 (Use case diagram)

Primary Actor: User

Secondary Actor: Dictionary Api

Description:

The sub system is divided into three parts. User can search for a word in three different ways.

Level 1.2.1.1

Use case Name: Input in app

Primary Actor: User

Secondary Actor: Dictionary Api.

Description:

Action: The user provides the desired word directly in the app.

Reply: The word is sent to view and the word is viewed.

Level 1.2.1.2

Use case Name: Search on copy

Primary Actor: User

Secondary Actor: Dictionary Api.

Description:

Action: The user copy's a word to clipboard.

Reply: The word is sent to view and the word is viewed.

Level 1.2.3

Use case Name: View

Primary Actor: User

Secondary Actor: Dictionary Api.

Description:

Action: searched word is found.

Reply: The word is sent to viewed to the user.

Level 1.2.2

Use case Name: Store

Use case Id: 1.2.2

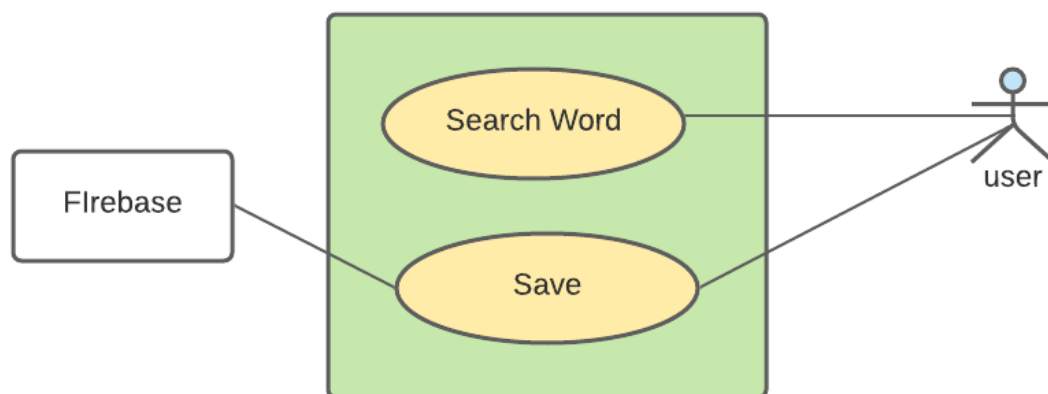


Fig: Level 1.2 .2(Use case diagram)

Primary Actor: User

Secondary Actor: Fire base

Description:

The sub system is divided into two parts. User can select a word from the view and store it.

Level 1.2.2.1

Use case Name: Select word

Primary Actor: User

Description:

The user selects a word from view.

Level 1.2.2.2

Use case Name: Save

Primary Actor: User

Secondary Actor: Fire base.

Description:

The selected word is sent to fire base to save it against the users word list.

Level 1.3

Use case Name: Memorization

Use case Id: 1.3

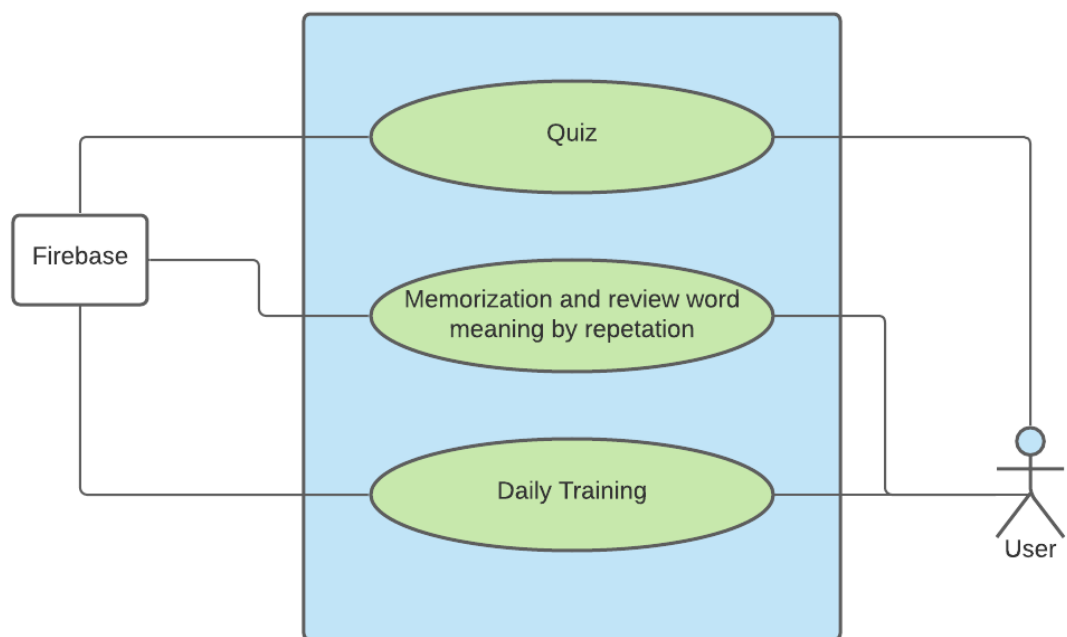


Fig: Level 1.3(Use case diagram)

Primary Actor: User

Secondary Actor: Fire base

Description:

The sub system is divided into three parts. User can take part in a quiz. Besides there is Memorization and review word meaning by repetition and daily training.

Level 1.3.1

Use case Name: Quiz

Use case Id: 1.3.1

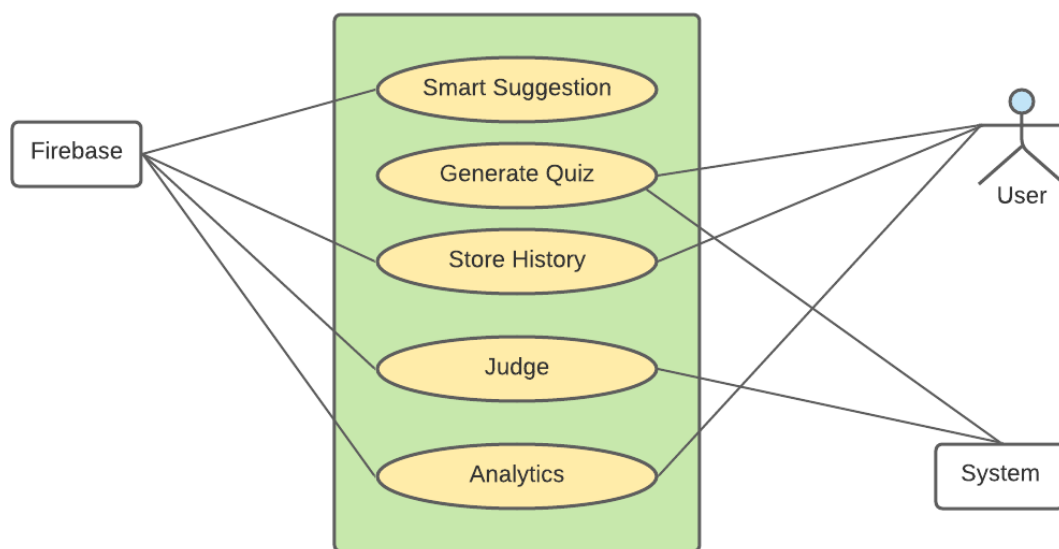


Fig: Level 1.3.1(Use case diagram)

Primary Actor: User

Secondary Actor: Fire base

Description:

The sub system quiz is divided into five parts. Smart Suggestion, generate quiz, store history, judge, analysis.

Level 1.3.1.1

Use case Name: Smart Suggestion

Secondary Actor: Fire Base

Description: Questions for quiz will be set based on the user's previous quiz result. Suggestions will be generated by the server

Level 1.3.1.2

Use case Name: Generate quiz

Primary Actor: User

Secondary Actor: Fire Base, System

Description:

Users can take part in quizzes in order to judge their skills. Quizzes composed of MCQ, active recalling and fill-in-the-blanks for both, word meanings and spelling. Users can participate in new quizzes or continue incomplete quizzes. Questions for quiz will be automatically generated by the server with help of smart suggestion.

Level 1.3.1.3

Use case Name: Store history

Primary Actor: User

Secondary Actor: Fire Base

Description:

If the user quits in the middle of a quiz than it will be saved. The user can again start that incomplete quiz or generate a new one if he wants.

Level 1.3.1.4

Use case Name: judge

Secondary Actor: Fire Base, System

Description:

System will judge the answer given by the user whether the answer is right or wrong. And thus, generate scores.

Level 1.3.1.5

Use case Name: Analysis

Primary Actor: User

Secondary Actor: Fire Base, System

Description:

The system will analysis the result from the quiz with previous quizzes and show improvement.

Level 1.3.2

Use case Name: Memorize by repetition

Use case Id: 1.3.2

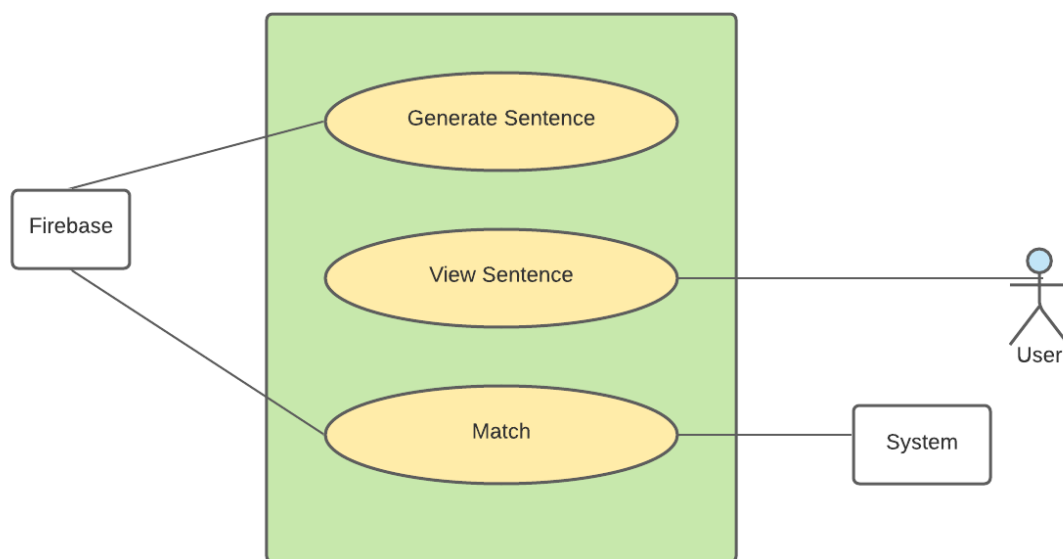


Fig: Level 1.3.2(Use case diagram)

Primary Actor: User

Secondary Actor: Fire base, system

Description:

The sub system quiz is divided into thre parts. Generate sentence for the user, let user review the sentence and finally match the answer

Level 1.3.2.1

Use case Name: Generate sentence

Secondary Actor: Fire Base

Description:

A sentence will be generated from the wordlist of the user kept in fire base.

Level 1.3.2.2

Use case Name: view

Secondary Actor: Fire Base

Description:

The created sentence from the previous subsystem(1.3.2.1) will be viewed to the user.

Level 1.3.2.3

Use case Name: match

Secondary Actor: Fire Base, System

Description:

Action: the user will submit his/her answer.

Reply: the user submitted answer will be matched to the exact answer by the system.

Level 1.3.3

Use case Name: Daily training

Primary Actor: User

Secondary Actor: Fire Base

Description:

A short training session will be automatically generated everyday based on users' previous quiz and memorization results.

Level 1.4

Use case Name: games

Use case Id: 1.4

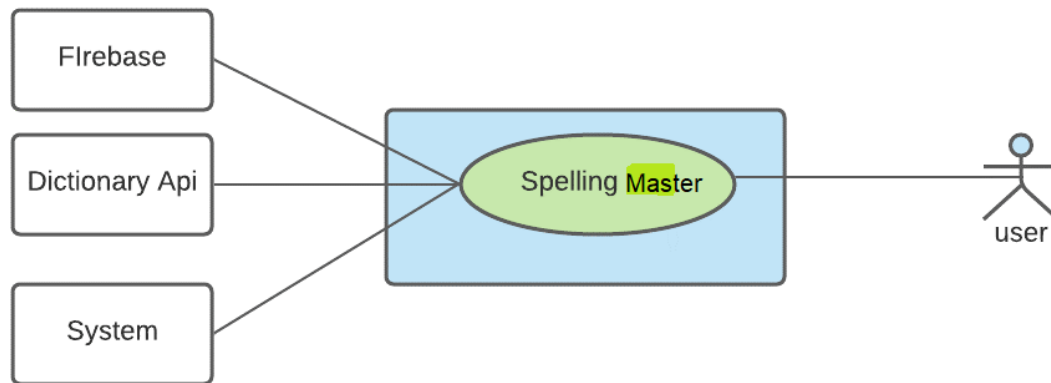


Fig: Level 1.4(Use case diagram)

Primary Actor: User

Secondary Actor: Fire base, Dictionary api, System

Description:

There are three games for the users currently. Spelling bee, Word battle and Word Copter.

Level 1.4.1

Use case Name: Spelling bee

Use case Id: 1.4.1

Primary Actor: User

Secondary Actor: Dictionary api, System

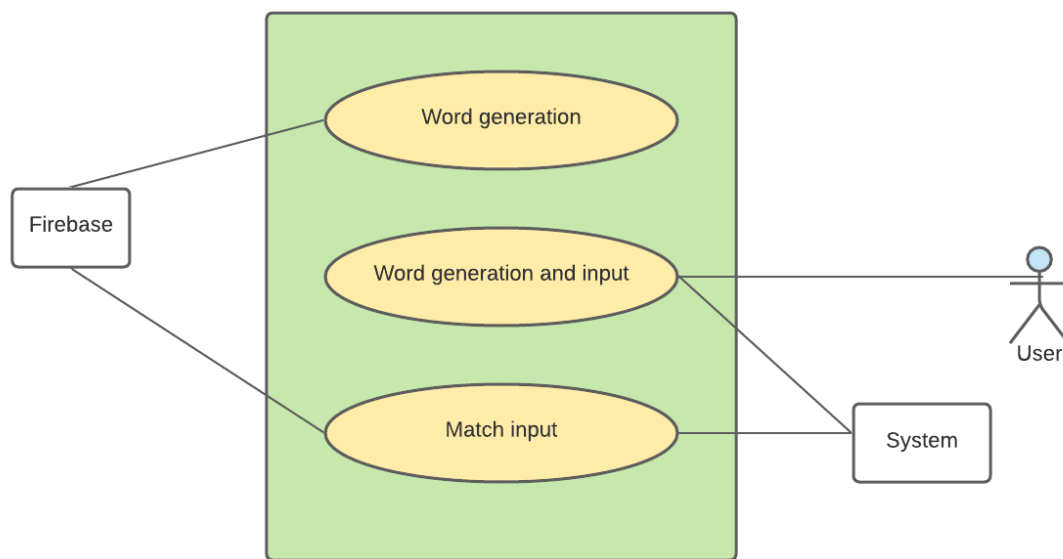


Fig: Level 1.4.1(Use case diagram)

Description:

There are three parts in the game spelling bee. They are mainly generate a word , display it and then take user input.

Level 1.4.1.1

Use case Name: word generation

Primary Actor: User

Secondary Actor: Dictionary api, System

Description:

Select a word from the dictionary to start the game.

Level 1.4.1.2

Use case Name: display

Primary Actor: User

Secondary Actor: Dictionary api, System

Description:

The player will be shown meaning and pronunciation. He will have to guess the spelling of the word and give it as input.

Level 1.4.1.3

Use case Name: display

Secondary Actor: Dictionary api, System

Description:

Match the real answer and match the user given answer. If matches issue point.

Level 1.5

Use case Name: Market Place

Use case Id: 1.5

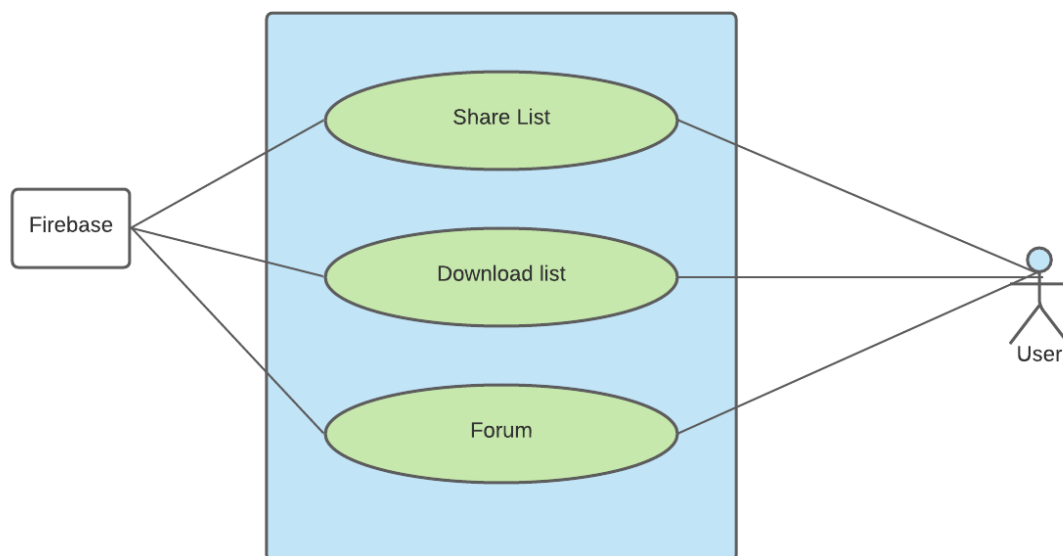


Fig: Level 1.5(Use case diagram)

Primary Actor: User

Secondary Actor: Fire Base

Description:

The whole subsystem has been divided into three parts. Share list, share meme and download list.

Level 1.5.1

Use case Name: Share list

Use case Id: 1.5.1

Primary Actor: User

Secondary Actor: Fire Base

Description:

Sharing list has two parts. Set list type as public and add tags.

Level 1.5.2

Use case Name: Forum

Use case Id: 1.5.2

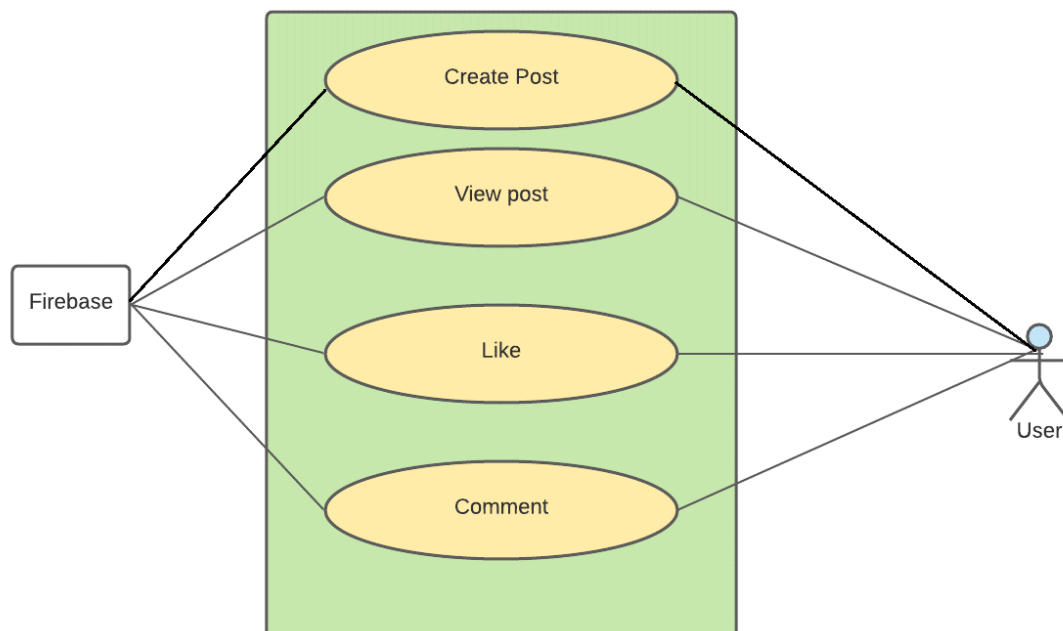


Fig: Level 1.5.2(Use case diagram)

Primary Actor: User

Secondary Actor: Fire Base

Description:

The whole share meme process has been divided into four parts

Level 1.5.2.1

Use case Name: create post

Primary Actor: User

Secondary Actor: fire base

Description:

Action: User create post.

Reply: The post is saved against the user.

Level 1.5.2.2

Use case Name: view post

Primary Actor: User

Secondary Actor: fire base

Description:

Post is viewed in the newsfeed

Level 1.5.2.2

Use case Name: like

Primary Actor: User

Secondary Actor: fire base

Description:

Action: User likes post.

Reply: Like count of the post is increased.

Level 1.5.2.4

Use case Name: comment

Primary Actor: User

Secondary Actor: fire base

Description:

Action: User comments on a post .

Reply: The comment is saved.

Level 1.6

Use case Name: Leader board

Primary Actor: User

Secondary Actor: fire base

Description:

Players will find themselves in the global leaderboard based on the points that they score in games and quizzes.

4.3 Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency.

4.3.1 Application

Activity name: Application

Activity id: 3.1

Version 1.0

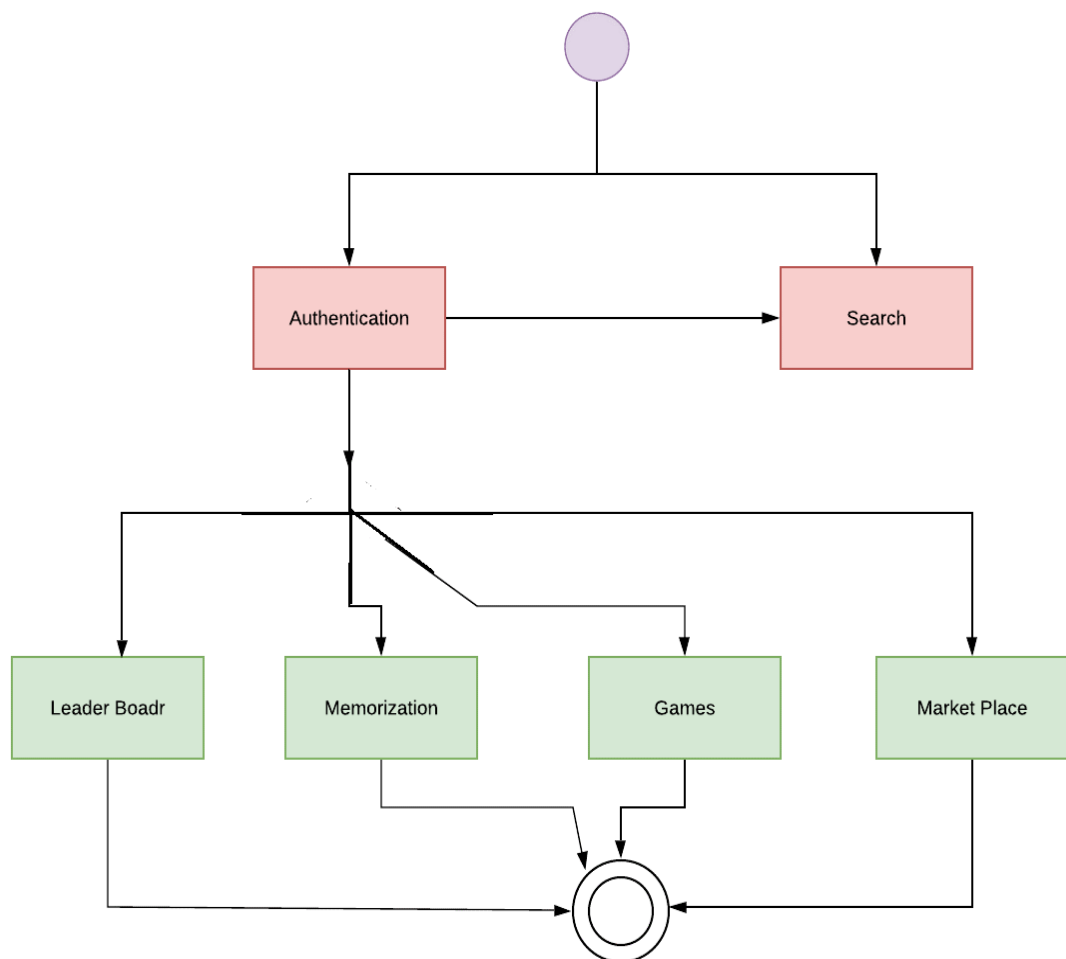


Fig: Application Activity Diagram

4.3.2 Authentication

Activity name: Authentication

Activity id: 3.1

Version 1.0

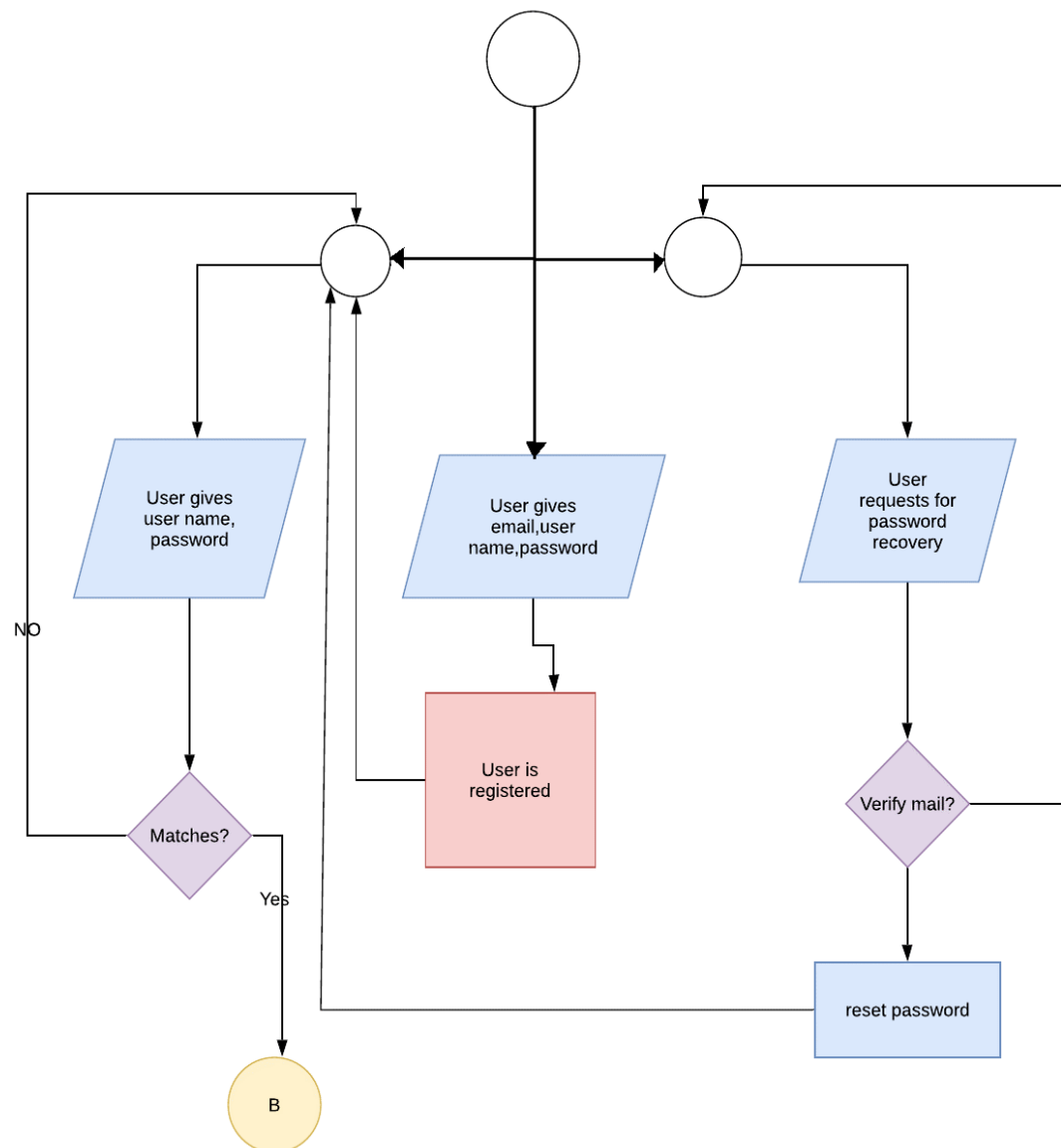


Fig: Authentication Activity Diagram

4.3.3 Dictionary Words

Activity name: Dictionary Words

Activity id: 3.3

Version 1.0

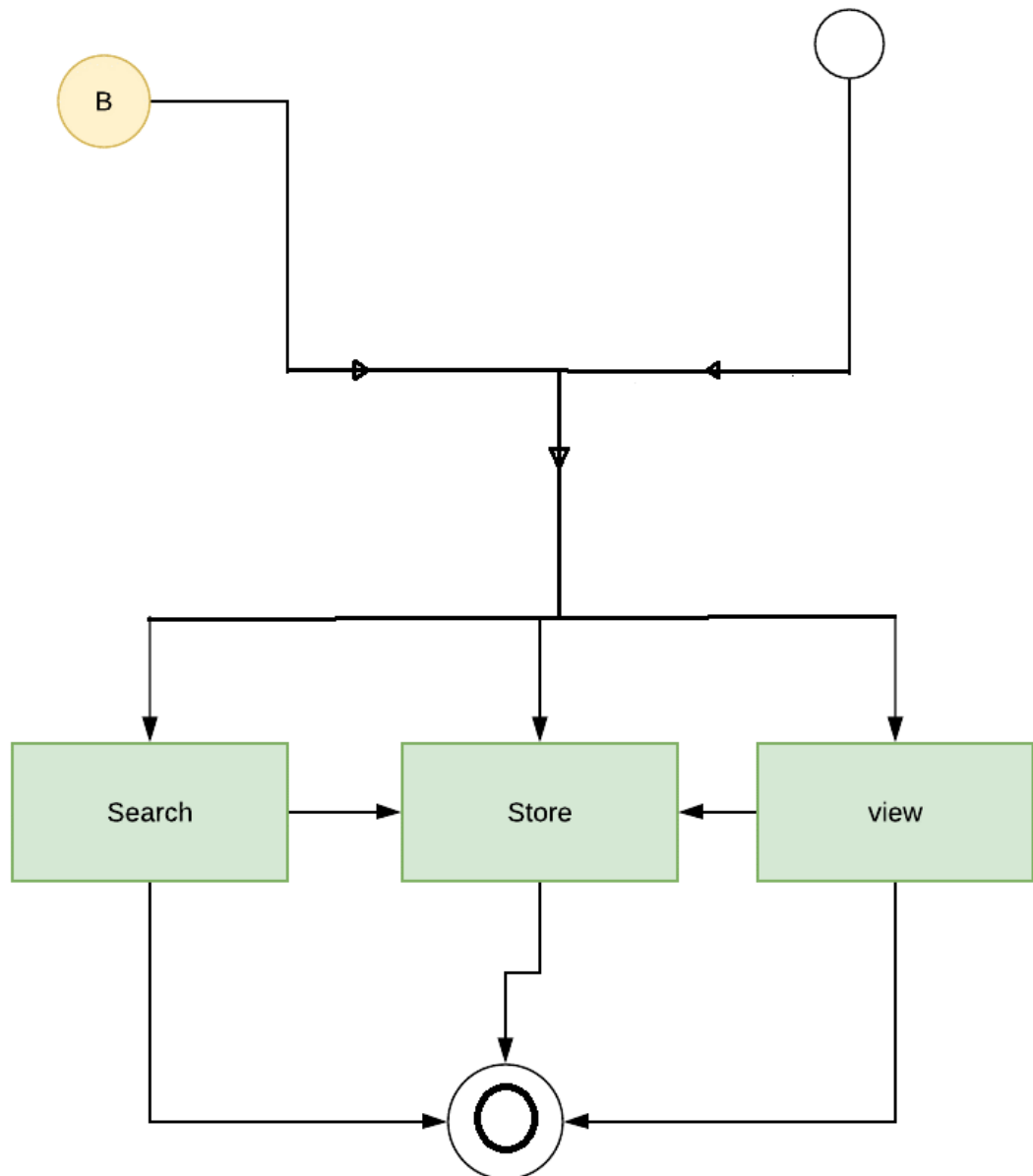


Fig: Dictionary Words Activity Diagram

4.3.3.1 Search

Activity name: Search

Activity id: 3.3.1

Version: 1.0

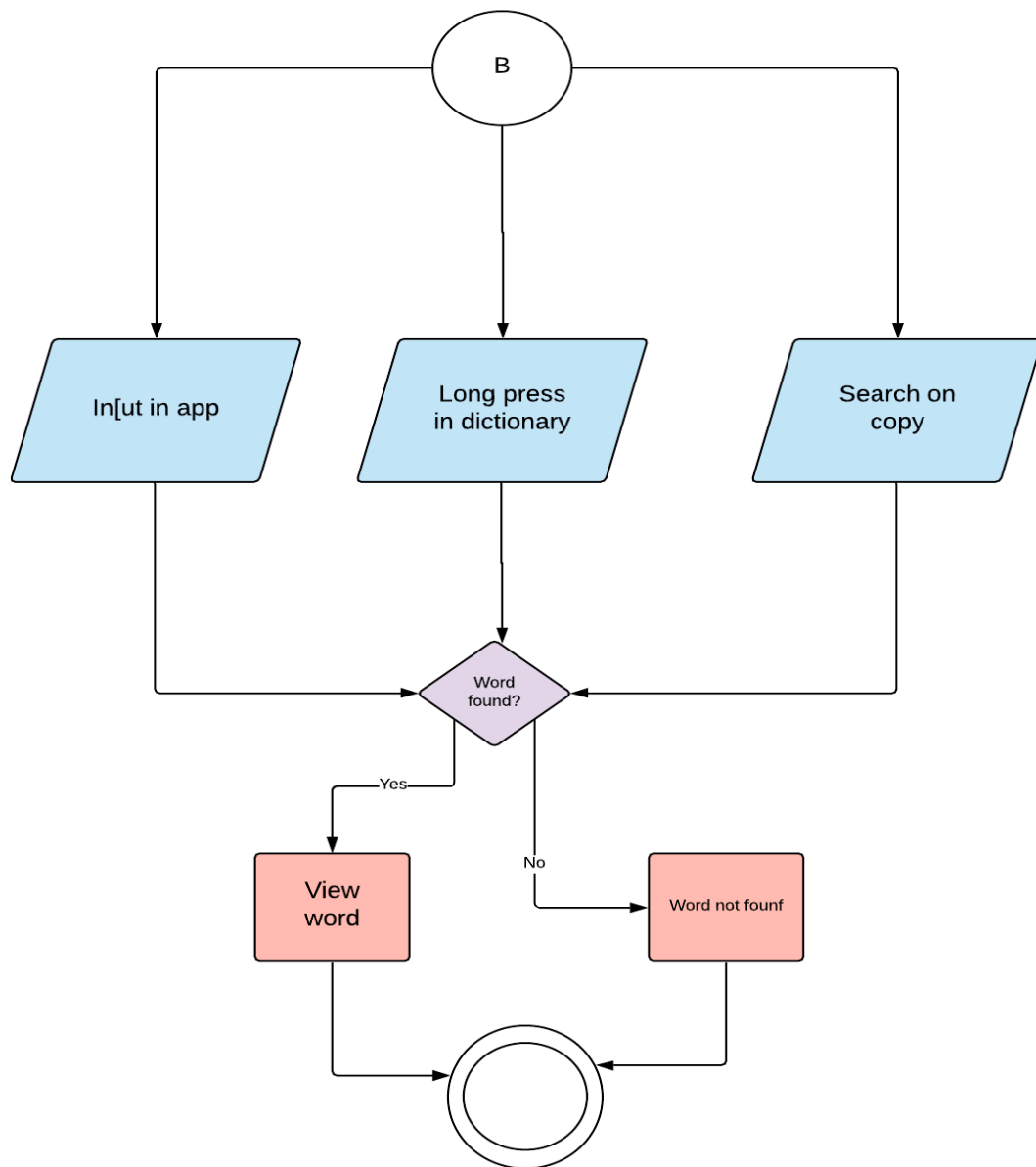


Fig: Search Activity Diagram

4.3.4 Memorization

Activity name: Memorization

Activity id: 3.4

Version: 1.0

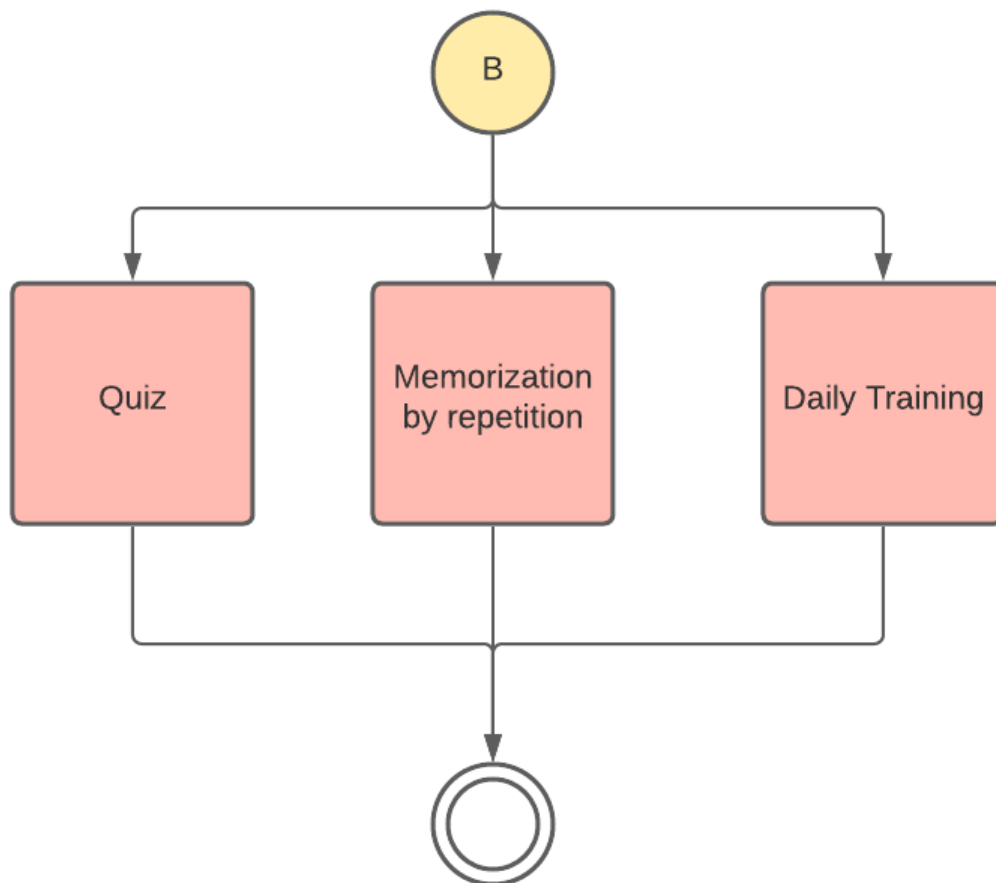


Fig: Memorization Activity Diagram

4.3.4.1 Quiz

Activity name: Quiz

Activity id: 3.4.1

Version: 1.1

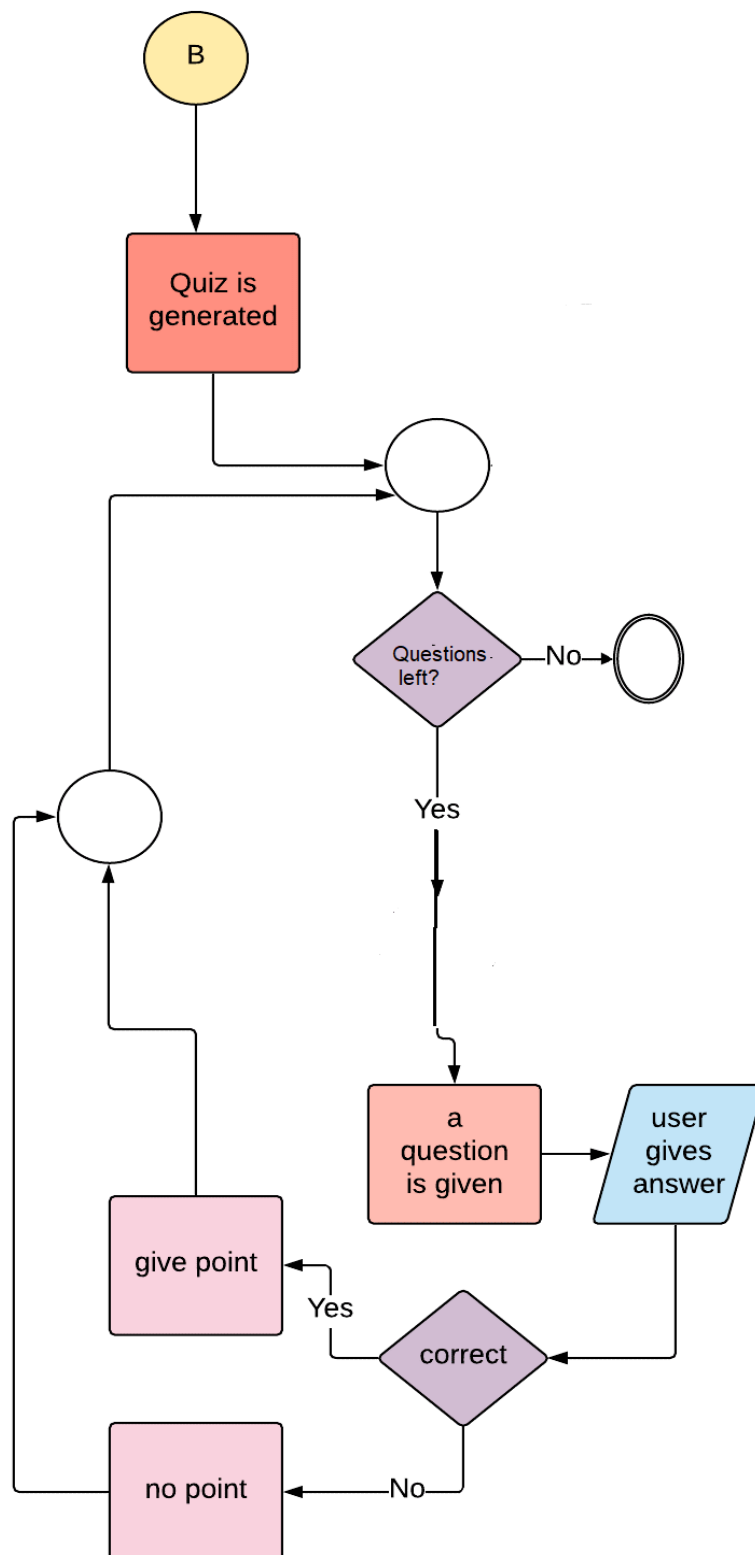


Fig: Quiz Activity Diagram

4.3.4.2 Memorization

Activity name: Memorization

Activity id: 3.4.2

Version: 1.1

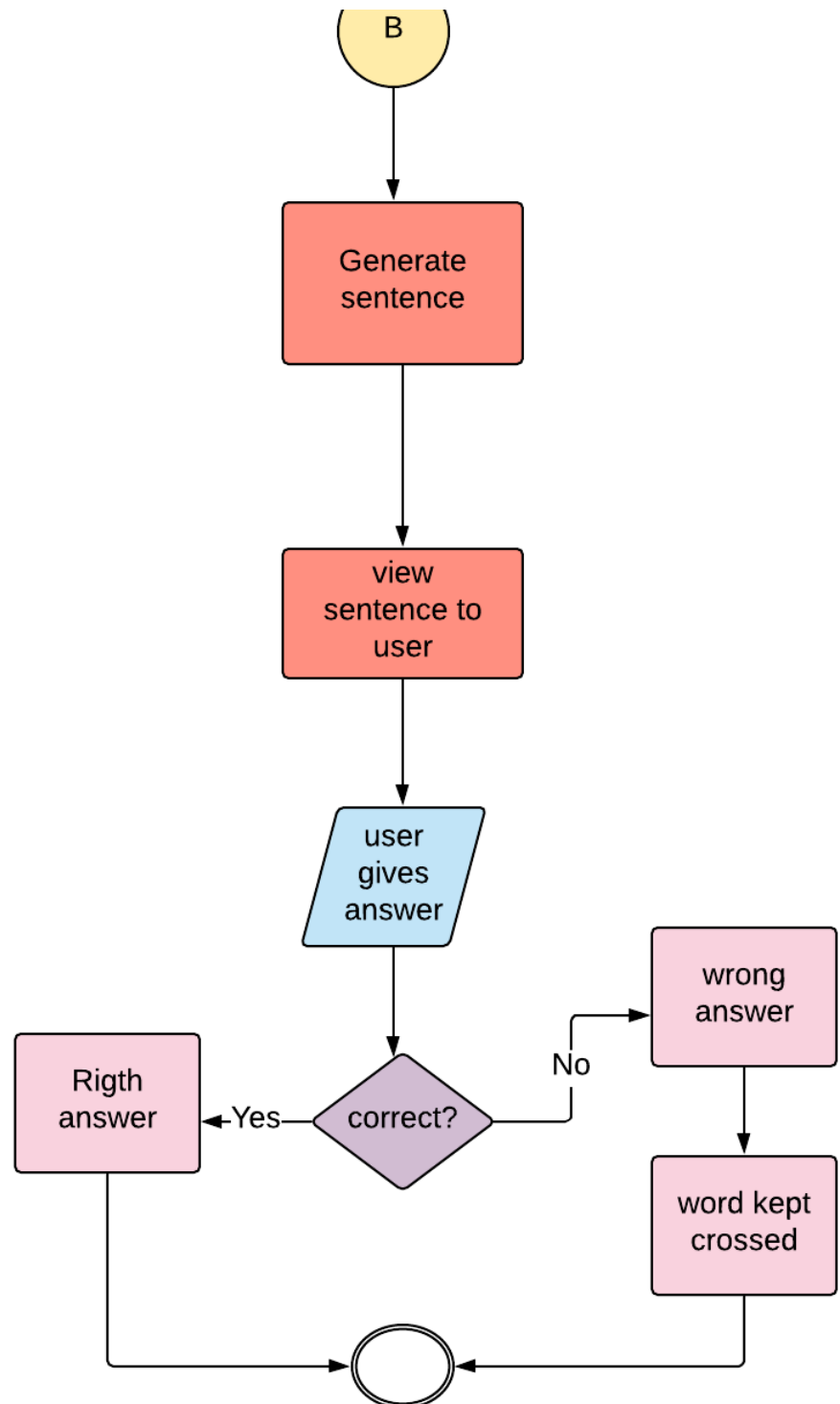


Fig: Memorization by repetition activity diagram

4.3.4.3 Daily Training

Activity name: Daily Training

Activity id: 3.4.3

Version: 1.1

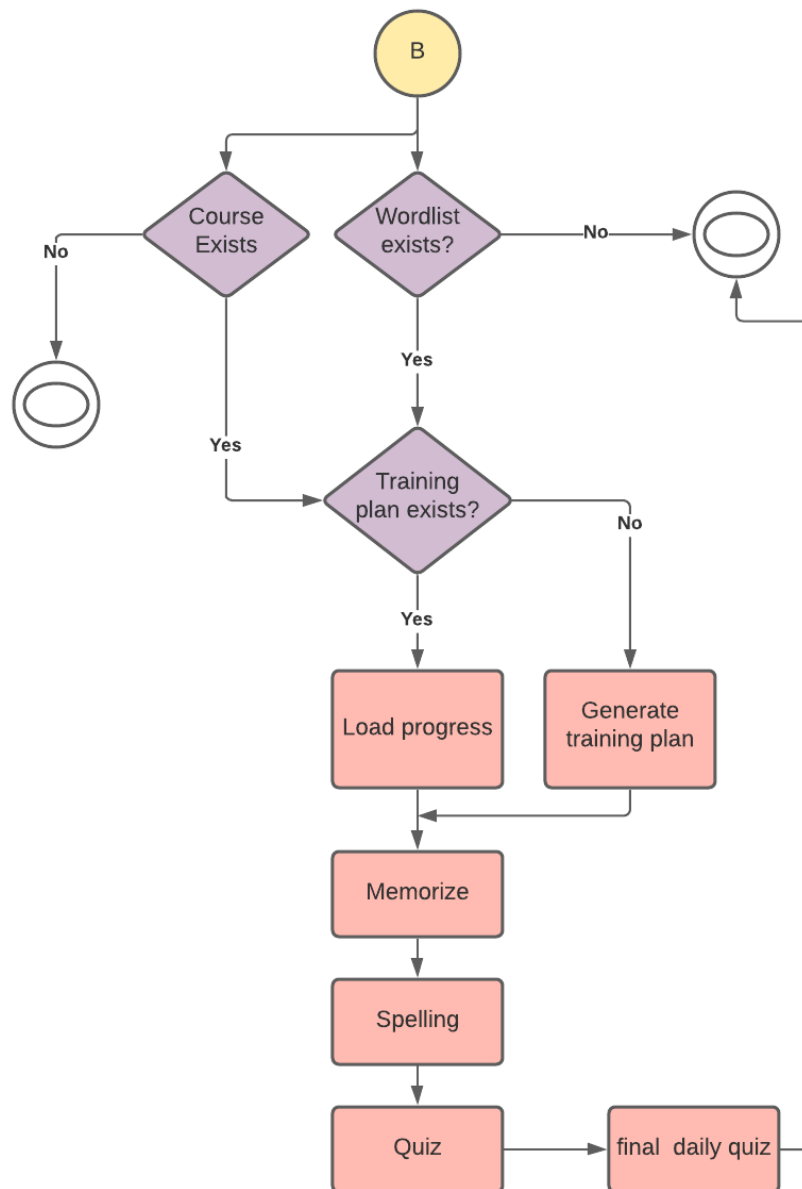


Fig: Daily Training activity diagram

4.3.5 Spelling Master

Activity name: Spelling Master

Activity id: 3.5.2

Version: 1.0

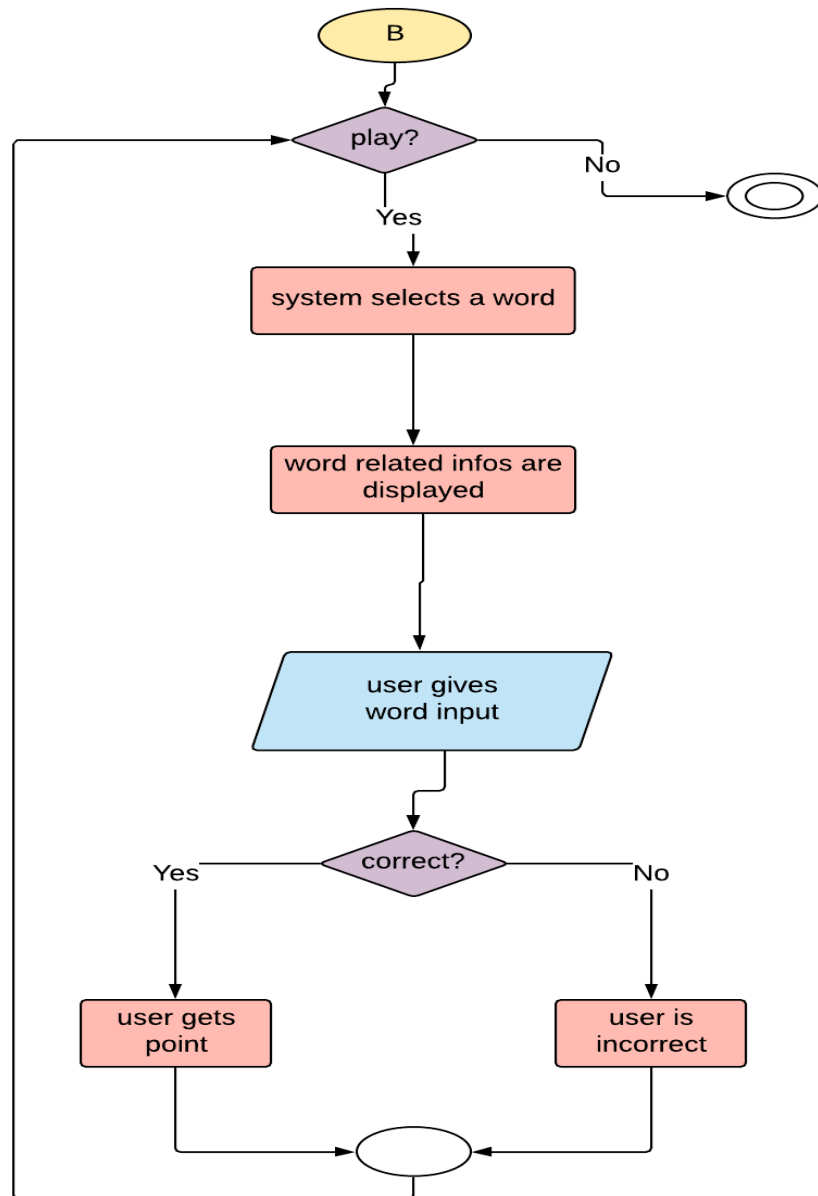


Fig: Spelling Bee Activity Diagram

4.3.6 Market Place

Activity name: Market Place

Activity id: 3.6

Version: 1.0

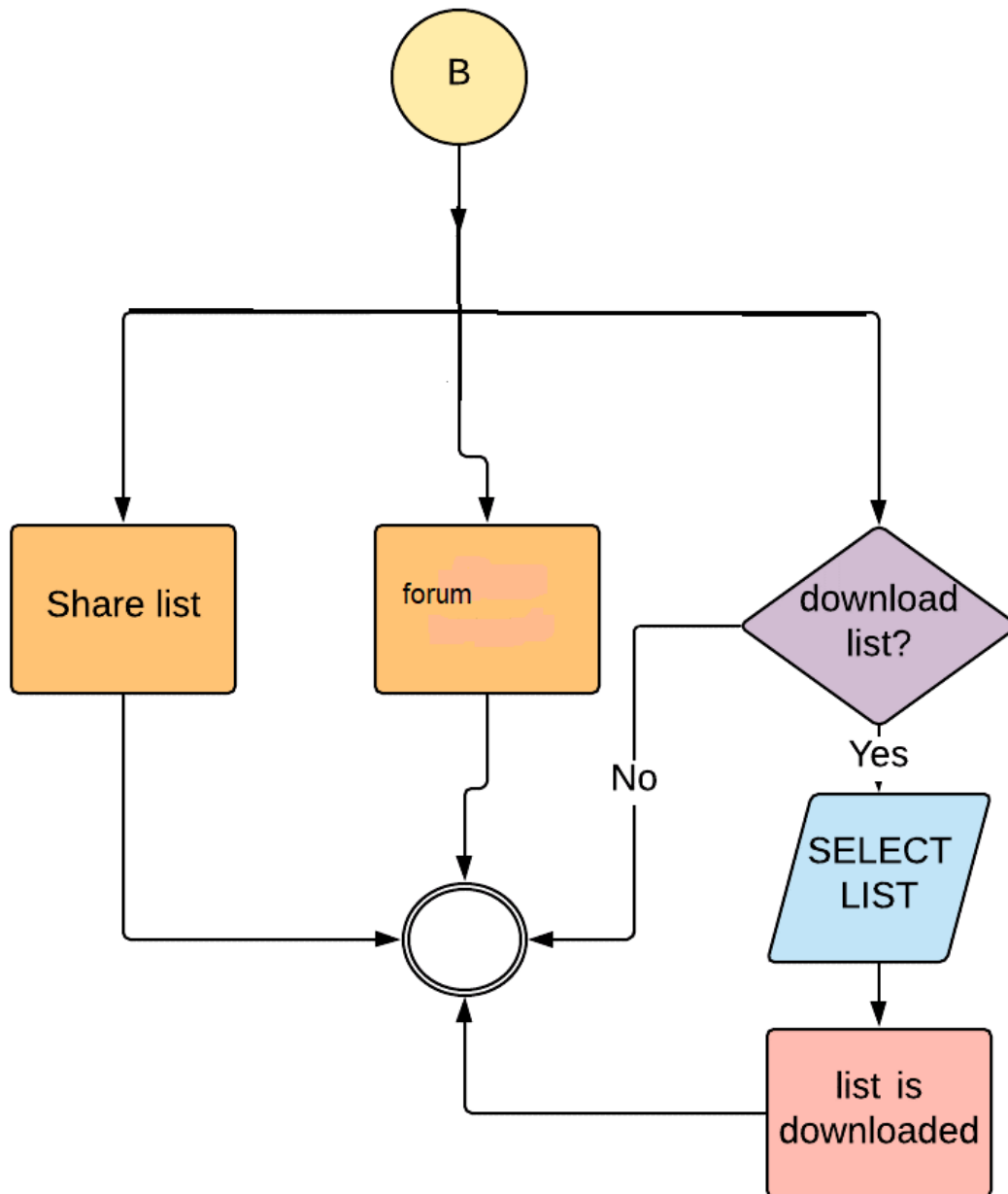


Fig: Market place Activity Diagram

4.3.6.1 Forum

Activity name: Forum

Activity id: 3.6.1

Version: 1.0

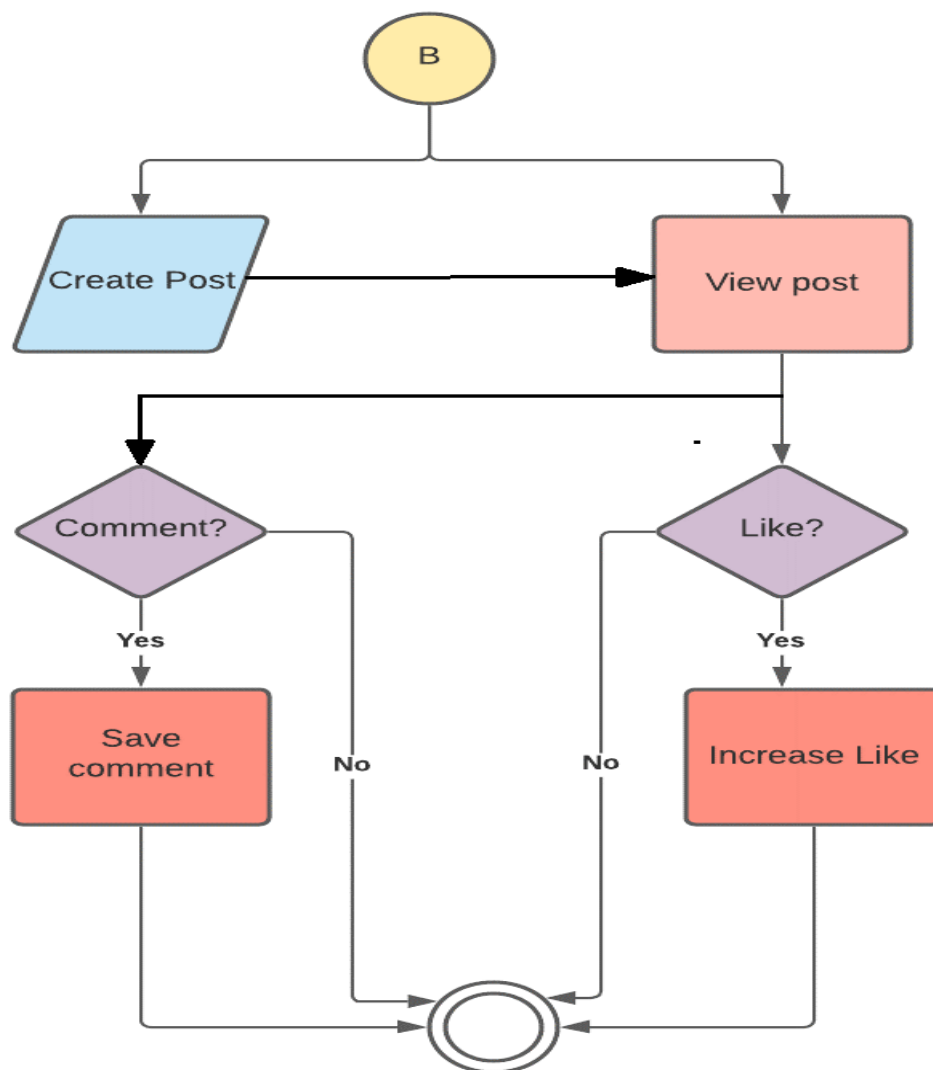


Fig: Forum activity diagram

4.3.6.2 Share wordlist

Activity name: Share wordlist

Activity id: 3.6.2

Version: 1.0

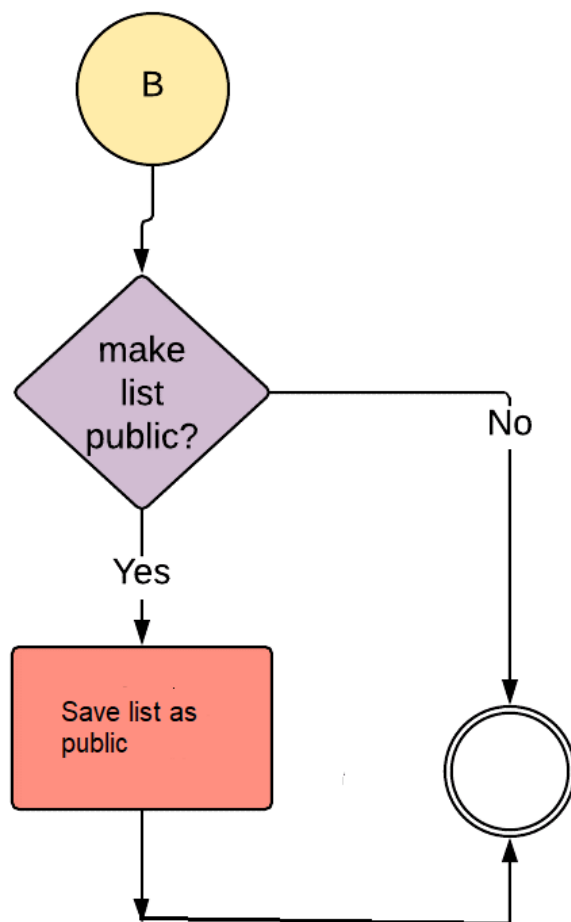


Fig: Share wordlist activity diagram

4.4 Swimlane Diagram of WM 3000

The following figures are the swim lane diagram of our project.

4.4.1. Authentication

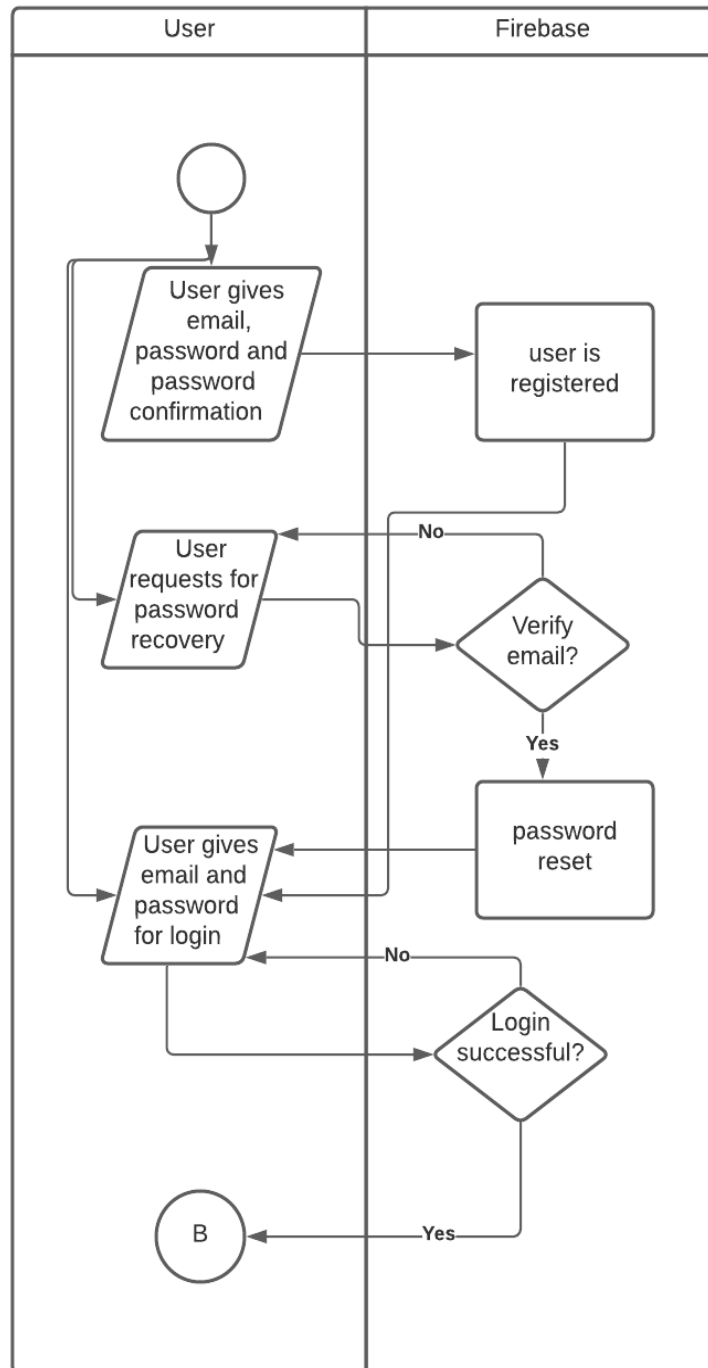


Fig: Authentication Swimlane diagram

Description: The figure shows the swimlane diagram for Authentication.

4.4.2 Dictionary Words

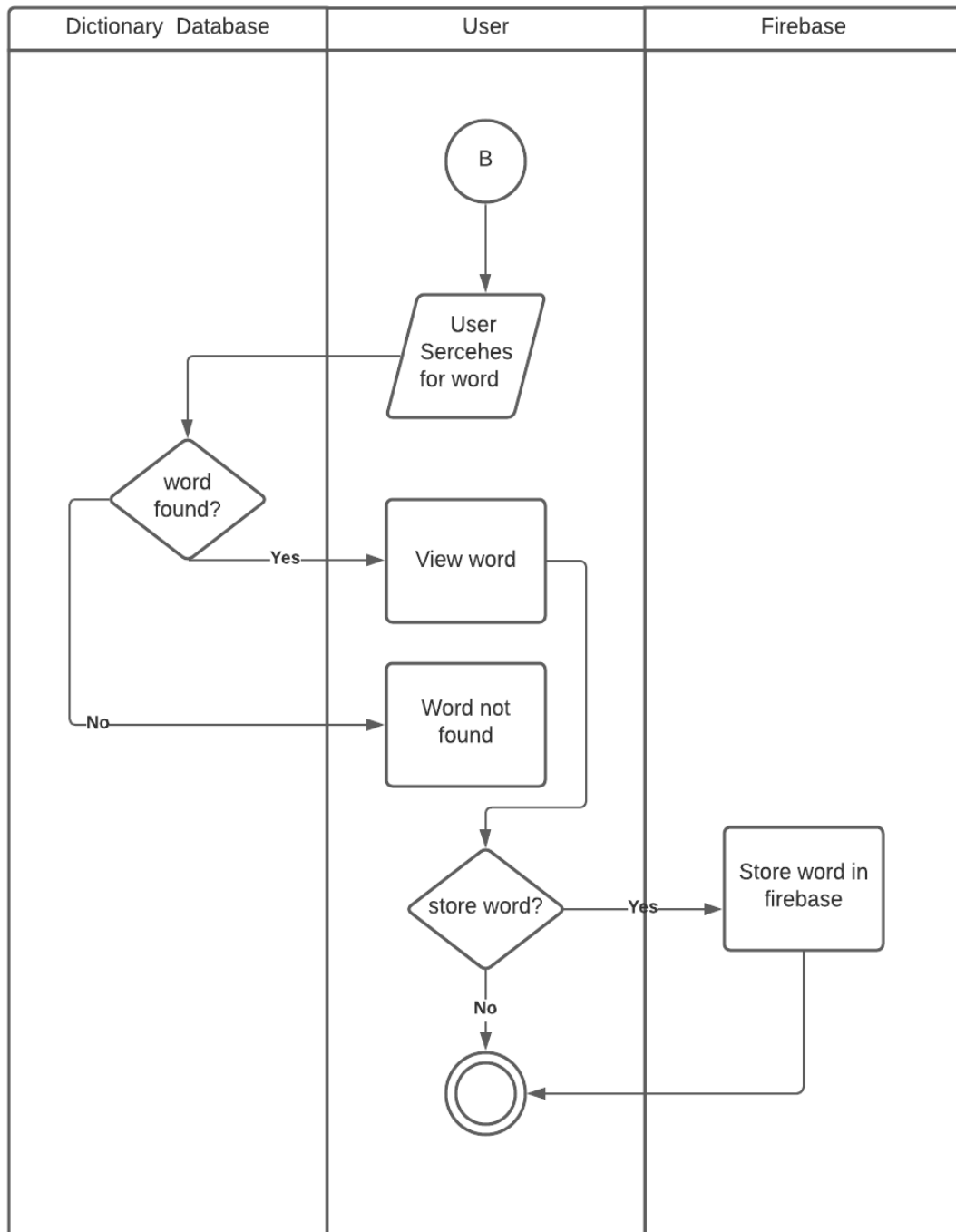


Fig: Dictionary Words Swimlane diagram

Description: The figure shows the swimlane diagram for Dictionary words module..

4.4.3 Quiz

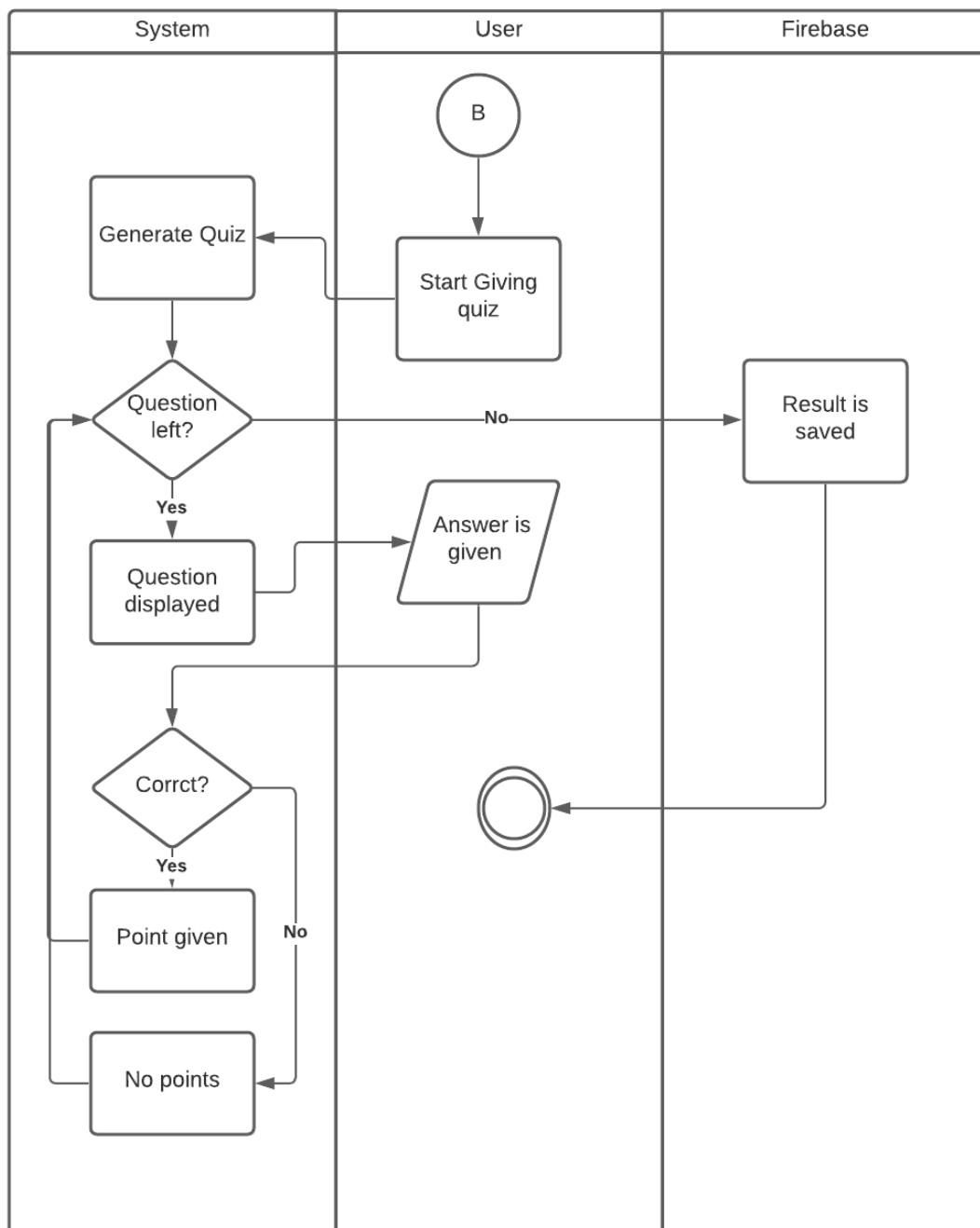


Fig: Quiz Swimlane diagram

Description: The figure shows the swimlane diagram for Quiz.

4.4.4 Daily Training

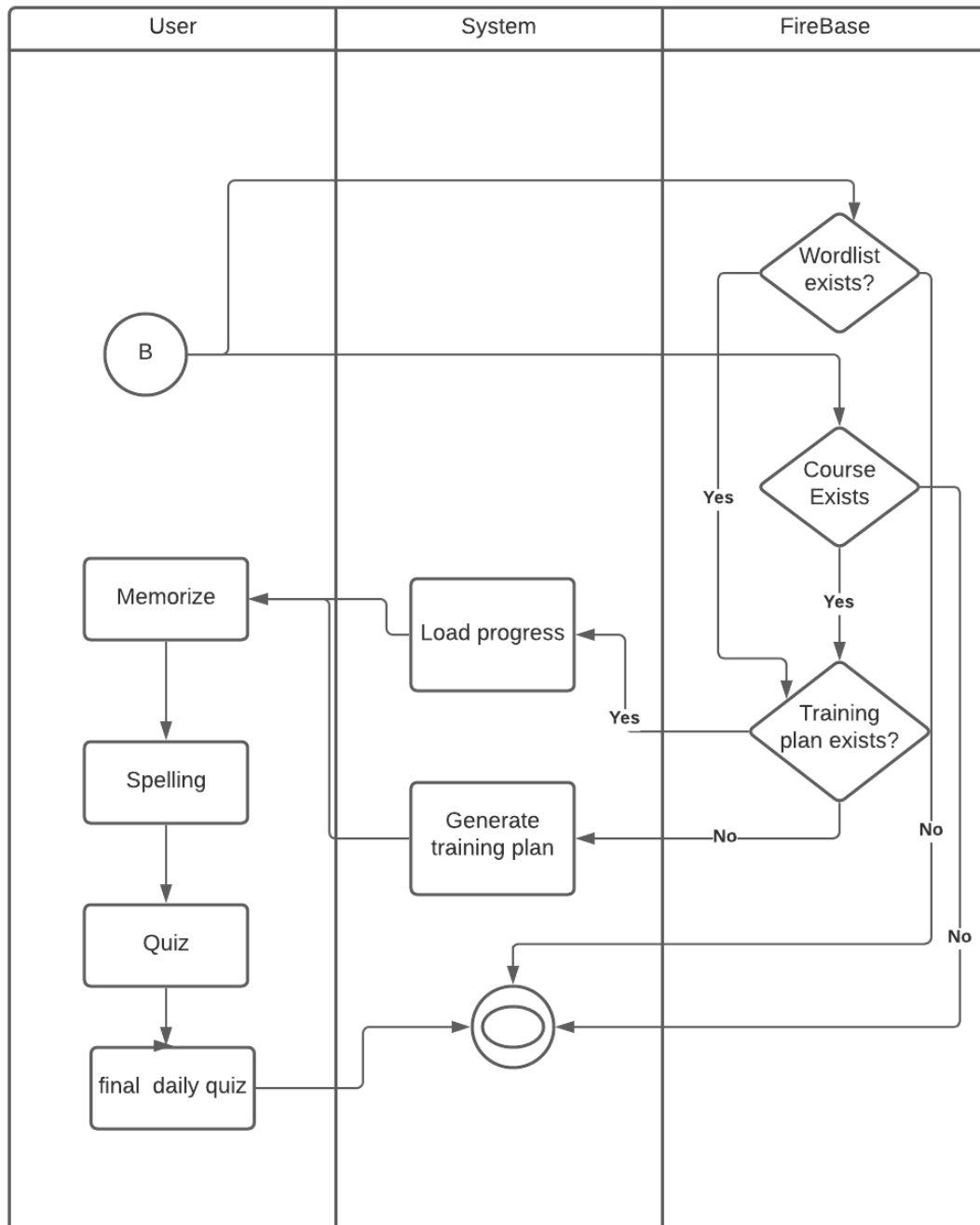


Fig: Daily Training Swimlane diagram

Description: The figure shows the swimlane diagram for Daily Training.

4.4.5 Memorize

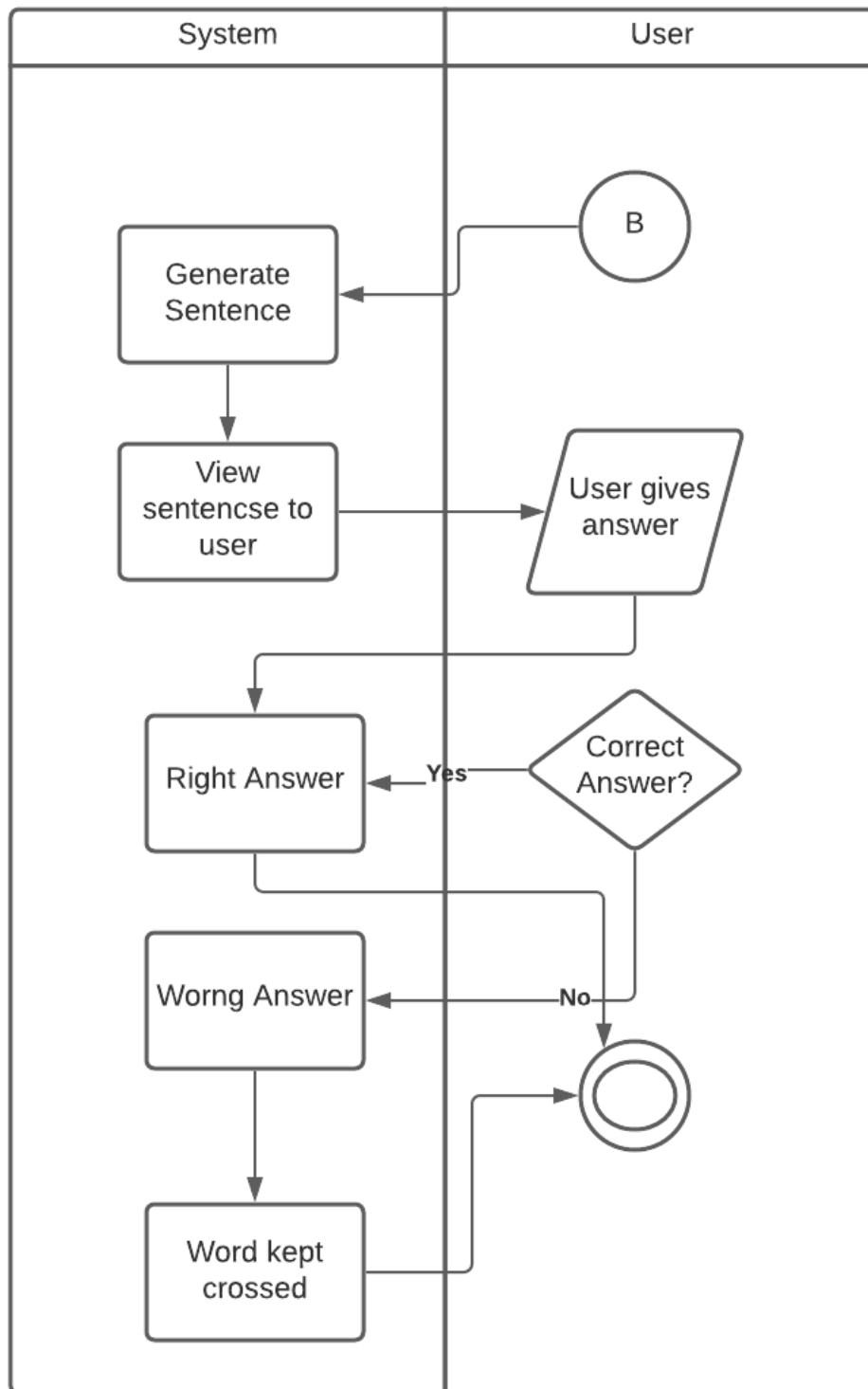


Fig: Memorize by Repetition Swimlane diagram

Description: The figure shows the swimlane diagram for Memorization module.

4.4.6 Spelling Master

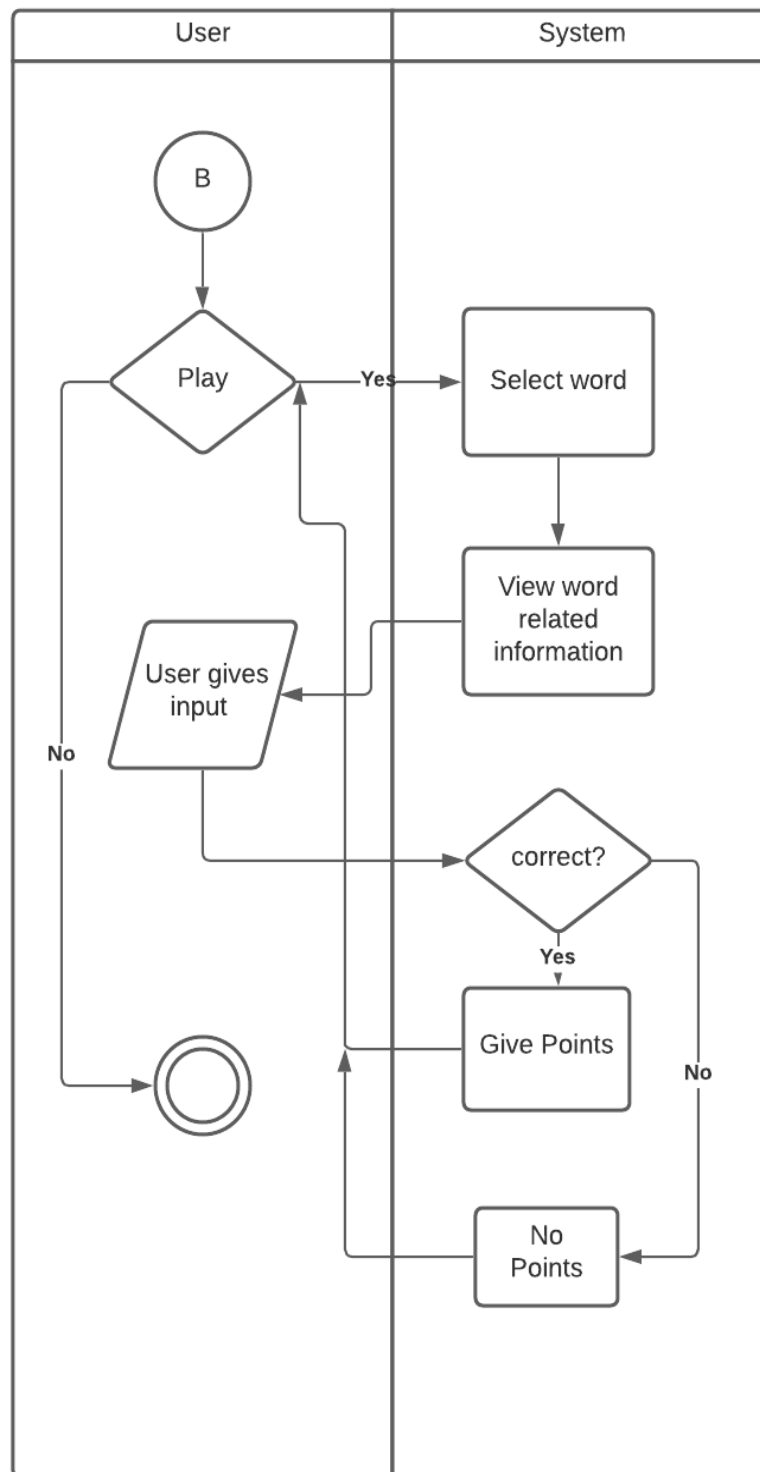


Fig:

Spelling Master Swimlane diagram

Description: The figure shows the swimlane diagram for Spelling Master.

4.4.7 Forum

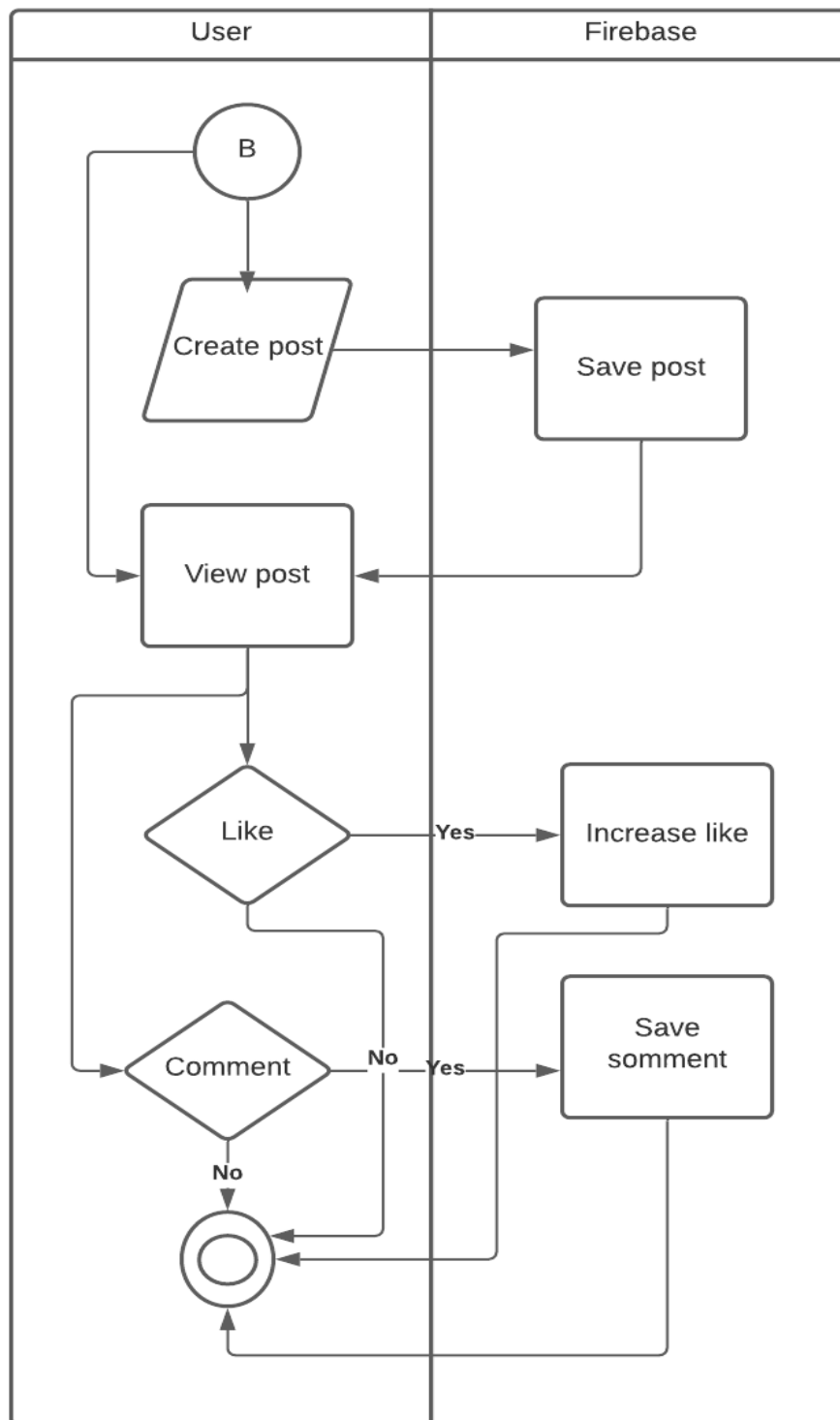
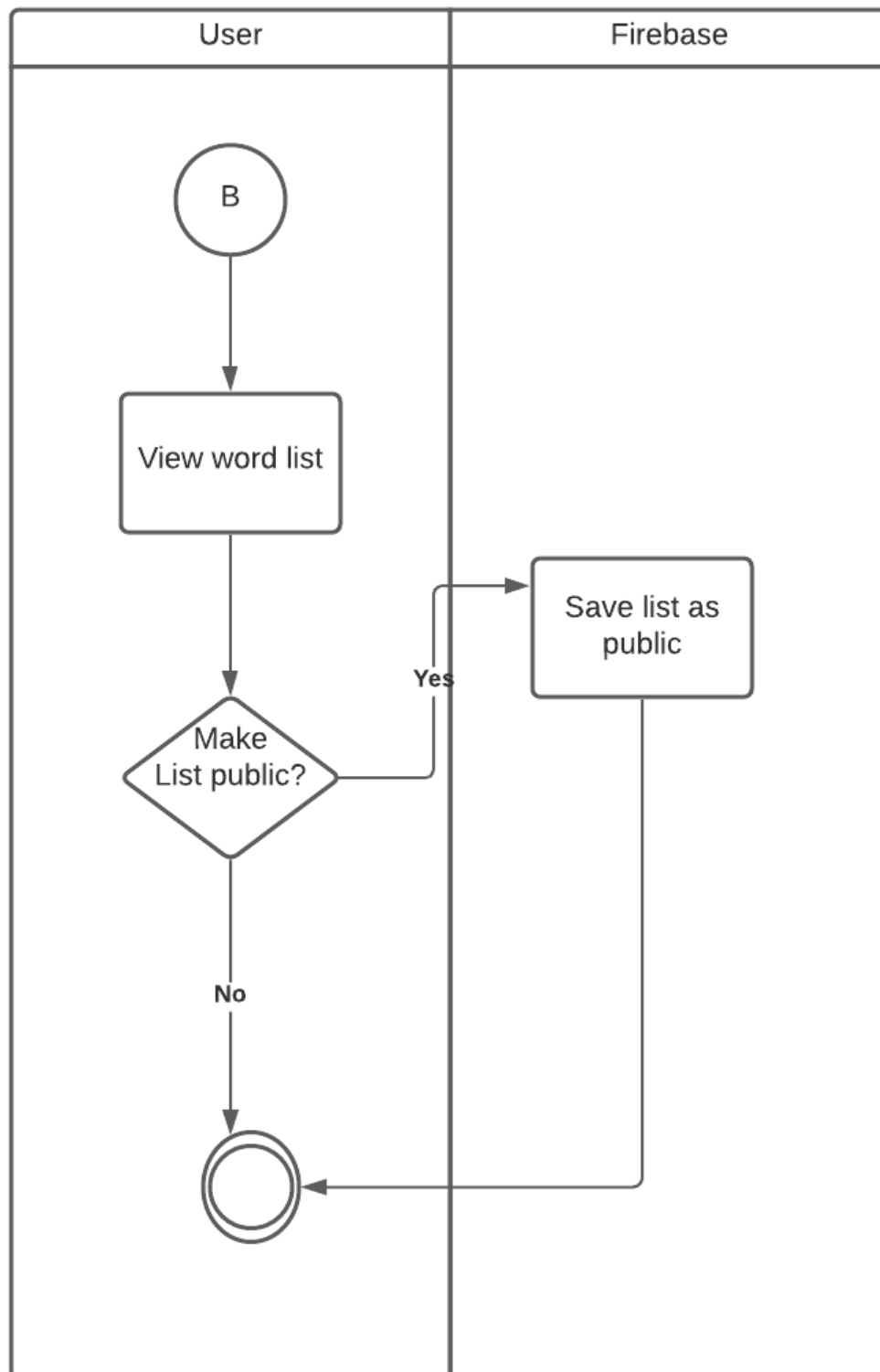


Fig: Forum Swimlane diagram

Description: The figure shows the swimlane diagram for Forum module.

4.4.8 Share Wordlist

Fig: Share Wordlist Swimlane diagram



Description: The figure shows the swimlane diagram for Share Wordlist module..

5. Data Based Modeling for Word Master 3000

The application depends on two separate databases. One is an offline database for the dictionary, and the other is an online database for storing user and his training related information.

5.1 ER DIAGRAM (Dictionary database)

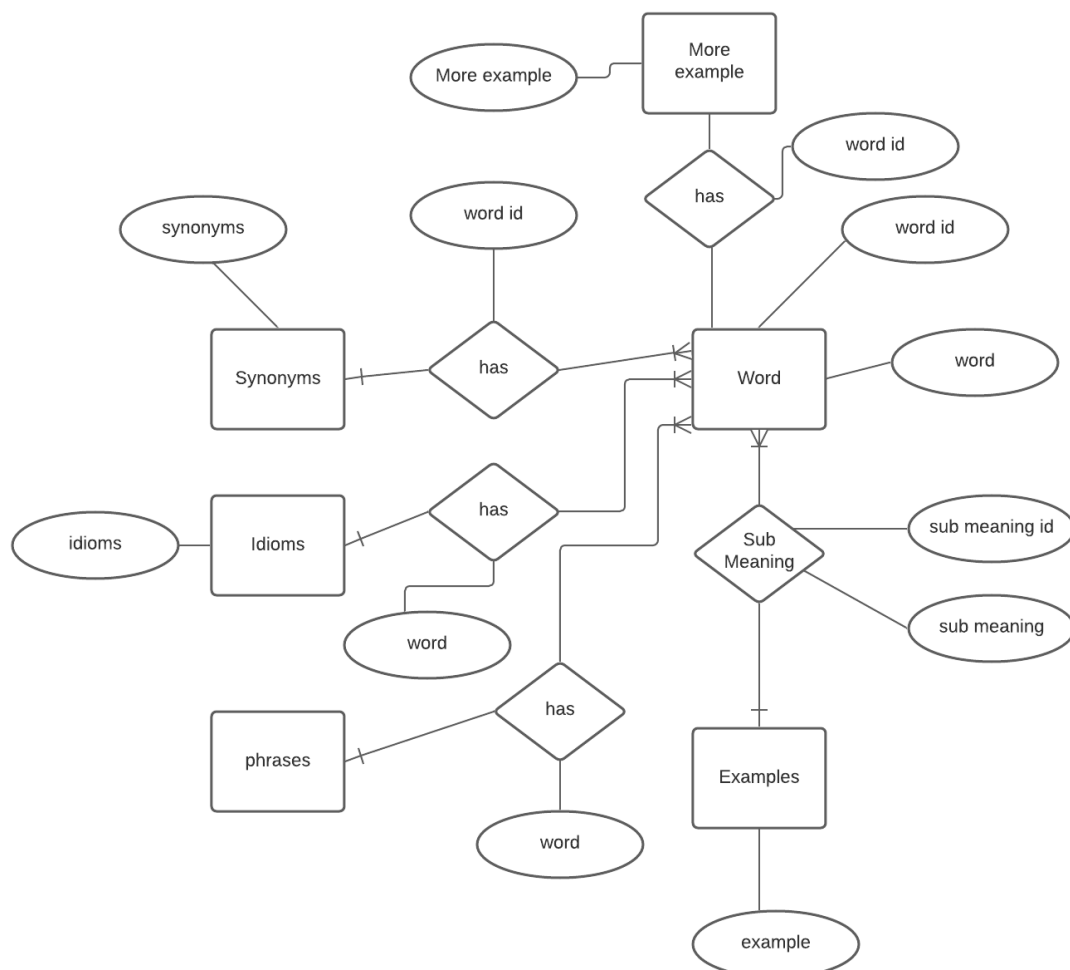


Fig: ER Diagram for word master 3000

5.2 Schema Diagram (Dictionary database)

Word		
Attribute	Type	Size
<u>Word ID</u>	Int	8
Word	String	200
Parts of speech	String	200
Meaning	String	200

Sub Meaning		
Attribute	Type	Size
<u>Word ID</u>	Int	8
<u>Sub meaning id</u>	int	8
Sub meaning	String	200

Example		
Attribute	Type	Size
<u>Sub meaning id</u>	int	8
<u>Example</u>	Dtring	200

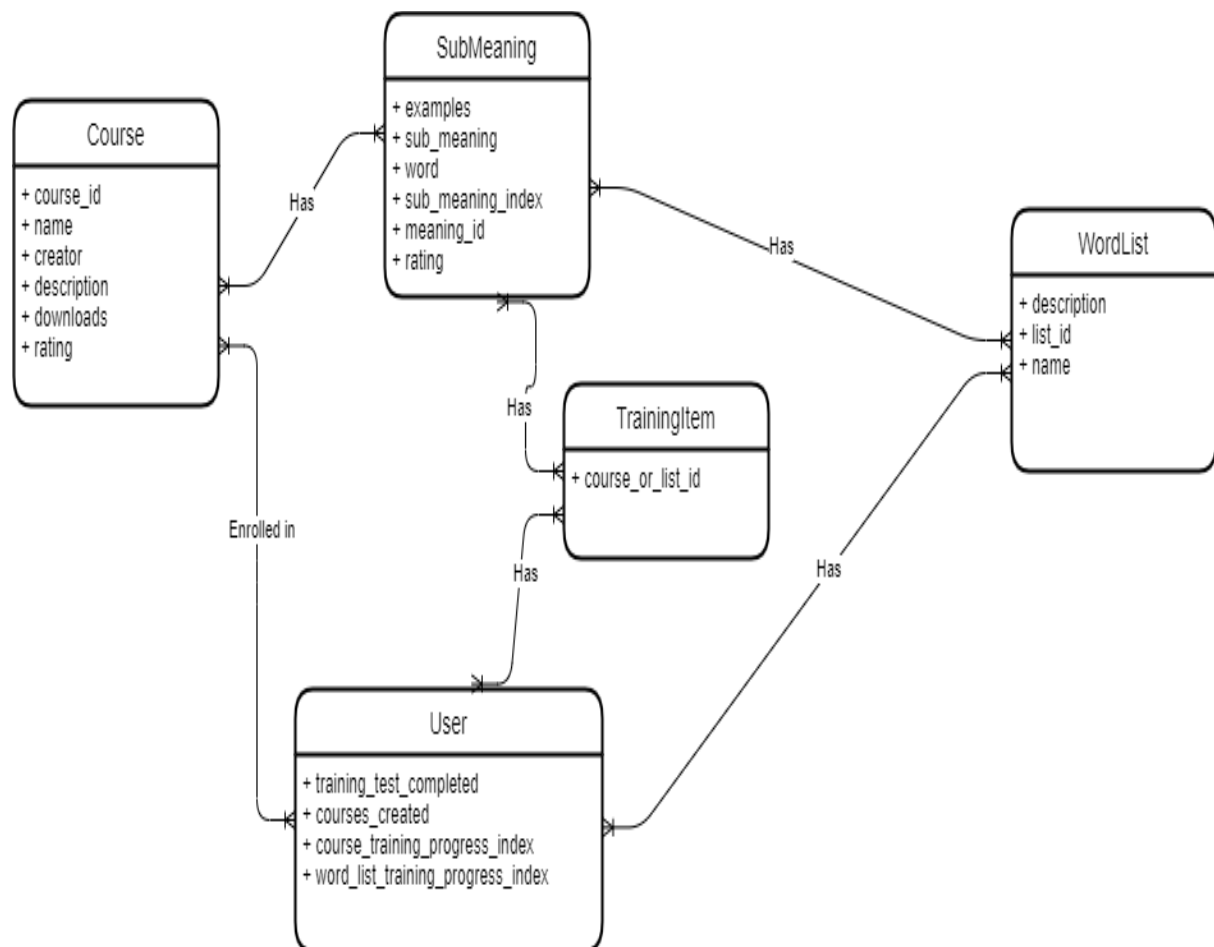
Synonyms		
Attribute	Type	Size
<u>Word ID</u>	Int	8
<u>synonyms</u>	string	200

phrases		
Attribute	Type	Size
<u>word</u>	string	200
phrases	String	200

Idioms		
Attribute	Type	Size
<u>word</u>	string	200
idioms	String	200

More Examples		
Attribute	Type	Size
<u>word ID</u>	Int	8
more example	String	200

5.3 Data objects (Cloud storage)





6. Class-Based Modeling for Word Master 3000

In this part we have discussed about the classes along with their attributes and methods of our application WORD MASTER 3000

6.1 CLASS BASED MODELING CONCEPT

Class-based Modeling represents the object. The system manipulates the operations. The elements of the class-based model consist of classes and objects, attributes, operations, class – responsibility - collaborator (CRS) models.

6.2 GENERAL CLASSIFICATION

To identify the potential classes, we have first selected the nouns from the solution space of the story. We have used seven general characteristics for this. The seven general characteristics are as follows:

1. External entities
2. Things
3. Events
4. Roles
5. Organizational units
6. Places
7. Structures

6.3 General Classification of Word Master 3000

Serial	Noun	
1	Authentication	3
2	MCQ	2,3
3	Fill – in – the blanks	2,3
4	account	2,3,4,7
5	antonyms	
6	check- box	3
7	clipboard	2,7
8	courses	2,7
9	database	1,2,7
10	dictionary	2
11	email	
12	example	
13	game	2,3,7
14	wordlist	2,7
15	leaderboard	2,3,7
16	link	
17	marketplace	2,3,6,7
18	meaning	
19	word	2,7
20	memorization	
21	username	2
22	password	
23	player	
24	point	
25	daily progress	2,3,7
26	pronunciation	
27	question	2
28	quiz	2,3,7
29	result	
30	round	2
31	score	
32	sentence	

33	server	1,2,7
34	spelling	
35	suggestion	2,3
36	synonyms	
37	system	2,3,7
38	Spelling master	2,3,7
39	time	
40	training	
41	user	2,4,5,7
42	forum	2, 7
43	post	2, 7
44	like	2
45	comment	2

6.4 SELECTION CRITERIA

The potential classes were then selected as classes by six Selection Criteria. A potential class becomes a class when it fulfills all six characteristics.

- **Retain information**
- **Needed services**
- **Multiple attributes**
- **Common attributes**
- **Common operations**
- **Essential requirements**

Serial	Noun	Selection criteria
1	MCQ	1
2	Fill – in – the blanks	1
3	account	1,2,3,4,5,6
4	clipboard	1
5	courses	1,2,3,4,5,6
6	database	1,2,3,6
7	game	1,2,5,6

8	leaderboard	1,2,5,6
9	marketplace	1,2,3,4,5,6
10	forum	1,2,3,4,5,6
11	player	1,2,3,4,5
12	daily progress	1,6
13	quiz	1,2,3,4,6
14	server	1,2,6
15	system	6
16	user	1,2,3,4,5,6
17	wordlist	1,2,3,6
18	word	2,3,4,6
19	Spelling master	1,2,3,4,5,6
20	Post	1,2,3,4,5,6
21	Comment	1,2,3,4,6

6.5 Associate Noun and Verb identification

We will now identify the nouns and verbs associated with the potential classes to find out the attributes and methods of each

Class

Serial	Potential Class	Noun	Verb
1	account	Name, User name, Email, password	Create, Login, recover
2	game	Player, score	Playing bee, Spelling Playing battle, Word Playing copter, Word Pairing,

			Scoring,
3	leaderboard	User, player	Daily progress, Weekly progress
4	marketplace	Wordlist, Meme,	Download wordlist,view,
5	player	account	Play game
6	quiz	Wordlist, user, answer, score	Generate new quiz, Save quiz, Judge quiz, Quiz analysis
7	user	Account, Frequent, Infrequent, quiz	Daily training, search words
8	Word	Spelling, meaning, multiple meaning, use in sentence, parts of speech, pronunciation	Search, input in app, long press in the dictionary, search on copy
9	wordlist	Name, user, Word, type	Memorize by repetitions, Save to list, training session, Share to market place,
10	Spelling master	Player, Meaning, Pronunciation, spelling	Launch game, Get word from dictionary, Show meaning, Show pronunciation, Check users answer,

			scoring
12	Post	User, text, timestamp, likes, comments	create post, edit post delete post, write comment, like post, unlike post

6.6 Identify Attributes

We have identified our potential classes, So, now we have to Identify the potential attributes of the potential classes

Serial	Potential Class	attributes
1	account	Name, User_name, Email, password
2	game	Player, Rounds, score
3	leaderboard	user, player
4	marketplace	Wordlist, user, Meme,
5	player	Account, game
6	quiz	Wordlist, user, answer, score
7	user	account, player, Frequent, Infrequent,

		Quiz, Wordlist,
8	Word	Spelling, meaning, multiple meaning, use in sentence, idioms, phrase
9	wordlist	ListName, user, Word, type
10	Spelling master	Player, Word,
11	post	creator, text, timestamp, likes, comments

6.7 Identify Methods

We have identified our potential classes, So, now we have
Identify the potential attributes of the potential methods

Serial	Class	Verb
1	account	createAccount(), Login(), recoverAccount()
3	leaderboard	getScore() displayRanking()

4	marketplace	EnrollInCourse(), PublishCourse(), UnPublishCourse()
5	player	PlayGame(), getScore();
6	quiz	GenerateNewQuiz(), JudgeQuiz(),
7	user	DailyTraining(), getQuizScore(). register(), takeQuiz(), Createlist();
8	Word	Savetolist(), selectMeaning(); selectAll(); viewWord() viewSynonyms()
9	wordlist	MemorizeByRepetition() TrainingSession(), publishToMarket(), generateQuiz(),
10	Spelling master	Launch game(), Gettingwordfromdictionary(), Showmeaning(), Showpronunciation(), Check users answer(), Scoring()
11	post	createPost() editPost() postComment() likePost() unlikePost()

6.8 Analysis

Here the player class and the user class are the same in this respect. Mainly, the user is a player. So, we can merge the classes. **We can merge player and user classes as user.**

And let us **rename the wordlist class to a course and post to forum post.**

There is no need for a separate Game class and Spelling Master class.

A word might have multiple meanings. And all the meanings will have their own example use etc. So, **we can split words into two classes: word and meaning**. Where words will be having a list of meanings. **Meaning will only have attributes**. Besides we will need a few handler class.

So, our final classes are

1. Account
2. User
3. MarketPlace
4. Quiz
5. Leaderboard
6. Word
7. Meaning
8. SubMeaning
9. WordList
10. SpellingMaster
11. ForumPost
12. DictionaryDatabaseHandler
13. FloatingBubbleHandler
14. Memorize
15. ForumComment
16. DailyTraining
17. FirebaseCrud

6.9 Class cards

After identifying our final classes we have generated the following class cards:

Word	
Attributes	Methods
<ul style="list-style-type: none"> • Meanings • Idioms • Phrases • Word body 	ProcessWordFromDataBase() ProcessIdiomsAndPhrases() ProcessMeanings()
Responsibility	Collaborator
<ul style="list-style-type: none"> • Hold information • One of the most valuable data entity • Hold information for the user to learn and practice 	DictionaryDatabaseController Meaning

Meaning	
Attributes	Methods
<ul style="list-style-type: none"> • PartsOfSpeech • MeaningBody • Synonyms • MoreExamples • MeaningID • SubMeanings 	ProcessMeaning() ProcessSynonyms() ProcessMoreExamples() ProcessSubMeanings()
Responsibility	Collaborator
<ul style="list-style-type: none"> • Hold information regarding a section of a word • Help complete the necessary information for a word 	DictionaryDatabaseController SubMeaning

SubMeaning	
Attributes	Methods
<ul style="list-style-type: none"> • SubMeaningBody • Examples 	LoadSubmeanings LoadExamples
Responsibility	Collaborator
<ul style="list-style-type: none"> • 	DictionaryDatabaseController

DictionaryDatabaseController	
Attributes	Methods

	OpenDatabase LoadSynonymsAsStrings LoadIdioms LoadPhrases LoadExamples LoadMoreExamples LoadAllWords
Responsibility	Collaborator
<ul style="list-style-type: none"> • Read raw data from local database • Help create sub meaning, meaning, and word. 	

FloatingBubbleHandler	
Attributes	Methods
<ul style="list-style-type: none"> • CopiedWord • LemmatizedWord 	ListenToNewCopy CheckForValidInput LemmatizeWord SearchWord
Responsibility	Collaborator
<ul style="list-style-type: none"> • Wait for user to copy a valid input • Query word when user demands • Show meaning in floating window 	Word Meaning SubMeaning

WordList	
Attributes	Methods
<ul style="list-style-type: none"> • Name • Description • ListID • SubMeanings 	DeleteWordList AddSubMeaning SearchSubMeaning DeleteSubMeaning ShuffleSubMeanings PublishToMarket
Responsibility	Collaborator

<ul style="list-style-type: none"> ● Store submeanings that interest the user ● Trigger update to the cloud storage 	SubMeanings FirebaseCrud
---	-----------------------------

Marketplace	
Attributes	Methods
<ul style="list-style-type: none"> ● 	EnrollInCourse UnPublishCourse LoadUserContribution
Responsibility	Collaborator
<ul style="list-style-type: none"> ● Let user enroll in course ● Let user unpublish course ● Display courses that the user has created ● Trigger update to cloud storage 	FirebaseCrud

SpellingMaster	
Attributes	Methods
<ul style="list-style-type: none"> ● TotalQuestionsAttempted ● TotalCorrectAnswers 	LoadPronunciation CheckAnswer UpdateScore
Responsibility	Collaborator
<ul style="list-style-type: none"> ● Help user learn and practice spelling ● Check user answer ● Calculate user score ● Launch game, ● Show word details 	Word SubMeaning WordList

Memorize	
Attributes	Methods
<ul style="list-style-type: none"> ● TotalQuestionsAttempted ● MistakenSubMeanings 	CheckAnswer StartNextRound
Responsibility	Collaborator
<ul style="list-style-type: none"> ● Help user revise and memorize meaning 	Word Meaning

<ul style="list-style-type: none"> • Check user answer • Start next round with words left to memorize 	SubMeaning WordList
---	------------------------

Quiz	
Attributes	Methods
<ul style="list-style-type: none"> • TotalQuestionsAttempted • TotalCorrectAnswers 	GenerateOptions CheckAnswer UpdateMarks
Responsibility	Collaborator
<ul style="list-style-type: none"> • Present question to user • Evaluate answer • Calculate score 	Word Meaning SubMeaning WordList

Leaderboard	
Attributes	Methods
<ul style="list-style-type: none"> • UserPosition • UserScore 	getLeaderBoardStandings getUserPosition
Responsibility	Collaborator
<ul style="list-style-type: none"> • Fetch leaderboard positions 	FirestoreCrud

User	
Attributes	Methods
<ul style="list-style-type: none"> • Account • TrainingProgress • CoursesCreated • CoursesEnrolled • WordLists 	UnenrollFromCourse CreateWordList
Responsibility	Collaborator
<ul style="list-style-type: none"> • Hold user information • Trigger update to cloud storage 	Account FirestoreCrud

Account	
Attributes	Methods
<ul style="list-style-type: none"> • Email • Name • Password 	LogIn SignUp RecoverPassword

<ul style="list-style-type: none"> • UserName 	
Responsibility	Collaborator
<ul style="list-style-type: none"> • Provide authentication services to user 	FirestoreCrud

ForumPost	
Attributes	methods
<ul style="list-style-type: none"> • creator, • text, • timestamp, • likes, • comments 	createPost() editPost() postComment() likePost() unlikePost()
Responsibility	Collaborator
<ul style="list-style-type: none"> • create post • edit post • delete post • post comment • like post 	FirestoreCrud

ForumComment	
Attributes	Methods
<ul style="list-style-type: none"> • creator, • text, • timestamp 	PostComment(), DeleteComment(),
Responsibility	Collaborator
<ul style="list-style-type: none"> • Store information about comments 	ForumPost FirestoreCrud

DailyTraining	
Attributes	Methods
<ul style="list-style-type: none"> • TrainingSequence 	LoadTrainingSequence UpdateRating EvaluateResult TrackTrainingProgress
Responsibility	Collaborator
<ul style="list-style-type: none"> • 	User FirestoreCrud

FirebaseCrud	
Attributes	Methods
<ul style="list-style-type: none"> • Firebase_connection_info 	Create(), Read(), Update(), Delete(), CallCloudFunctions(),
Responsibility	Collaborator
<ul style="list-style-type: none"> • Create Documents • Update Documents • Read Documents • Delete Documents 	User

6.10 Class Modeling Diagram

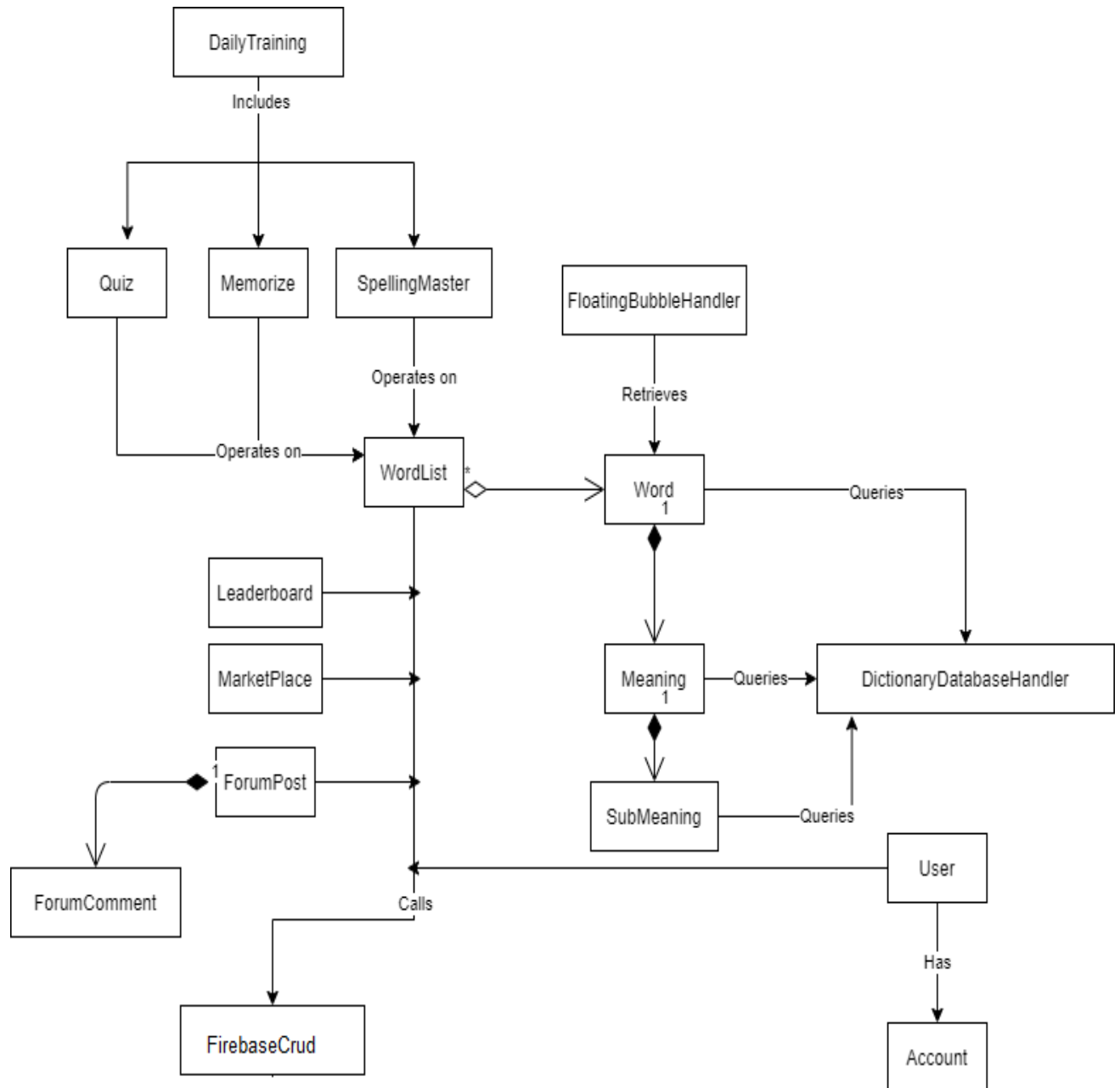


Fig: Class modeling diagram

7. BEHAVIORAL MODELING OF WORD MASTER 3000

7.1 STATE TRANSITION

State diagram represents active states for each class the events (triggers). For this we identified all the events, their initiators and collaborators.

Serial	Event	initiator	collaborator	State name
1	Aim to help			
2	Spending time	User		
3	Create account	User	User	create Account
4	Access all feature	User	User	
5	To provide info	user	Account	Provide info
6	Login	user	Account	Login
7	Recover id	User	Account	Recover
8	Show word details	User	Word, meaning	Show word details
9	Select specific meaning	User	Word, meaning	Select specific meaning
10	Select all meanings	User	Word, meaning	Select All
11	Group all meanings to get all	System	Word, system meaning	Group all meanings
12	Search words	user	Word	search
13	Input in app	user	System	Input In App,

14	Search on copy	user	System	Search on Copy
15	Create list	user	Word, Course, firebaseCrud	Create list
16	Store words	User	Course, words, firebase, firebaseCrud	Store word
17	Share list	User,	Marketplace, course, firebaseCrud	Share list
18	Make list public	User,	Marketplace, course, firebaseCrud	Make list public
19	Memorize words	User,	Course, firebaseCrud, meaning	Memorize by repetitions
20	Take part in quizzes	user	Quiz,	Take part in quizzes
21	Create new quiz	User	Quiz,	Create new quiz
22	Smart suggestions	Quiz,	firebaseCrud, Course	Smart suggestions
23	Generate short training session	course	System,	Generate short training session
24	Play game	User,	Game,	Play game
25	Launch game	Game		Launch game
26	Show meaning and pronunciation	Spelling bee	Game, firebaseCrud, meaning	Show meaning and pronunciation
27	Match the answer of the user	Spelling bee	Game, meaning	Match the answer of the user
28	Display question	Quiz,	system	Display question
29	Get first one answered			
30	Scoring	Daily Training	User	Scoring

31	Generate certain question	System	Course	Generate certain question
32	Show questions to user	Game, quiz	User, firebaseCrud	Show questions to user
33	Generate leaderboard	System	User	Generate leaderboard
34	Access market place	User	Marketplace	Access market place
35	Download word list	User	Course	Download word list
36	Delete post	user	post	Delete post
37	Like post	User	ForumPost	Like post
38	Daily training	User	System	Daily Training
39	Post in forum	ForumPost	User	Posting in forum
40	Commenting in forum	ForumPost	User	Commenting in forum

7.1.1 Events after analysis:

After some analysis, we can merge some events and states which are of same types. The analyzed data are given below:

serial	event	initiator	collaborator	State name
1.	Create account	User	User	Create account
2.	To provide info	user	Account	To provide info
3.	Login	user	Account	Login
4.	Recover id	User	Account	Recover id
5.	Show word details	User	Word, meaning	Show word details

6.	Select specific meaning	User	Word, meaning	Select specific meaning
7.	Select all meanings	User	Word, meaning	Select all meanings
8.	Group all meanings to get all	System	Word, system meaning	Group all meanings to get all
9.	Search words	user	Word	Search words
10.	Input in app	user	System	Input in app
11.	Search on copy	user	System	Search on copy
12.	Create list	user	Word, Course, firebaseCrud	Create list
13.	Store words	User	Course, words, firebaseCrud, system	Store words
14.	Share list	User,	Marketplace, course, firebaseCrud	Share list
15.	Make list public	User,	Marketplace, course, firebaseCrud	Make list public
16.	Memorize words	User,	Course, meaning	Memorize words
17.	Take part in quizzes	user	Quiz	Take part in quizzes
18.	Create new quiz	User	Quiz,	Create new quiz
19.				
20.	Smart suggestions	Quiz,	firebaseCrud, Course	Smart suggestions
21.	Generate short training session	firebase	System,	Generate short training session
22.	Play game	User,	Game	Play game
23.	Launch game	Game	firebaseCrud	Launch game
24.	Show meaning and pronunciation	Spelling bee	Game, meaning	Show meaning and pronunciation

25.	Match the answer of the user	Spelling bee	Game, meaning	Match the answer of the user
26.	Display question	Quiz,	firebaseCrud, system	Display question
27.	Scoring	Game	firebaseCrud, User	Scoring
28.	Generate certain question	System	Course	Generate certain question
29.	Show questions to user	Game, quiz	User, firebase	Show questions to user
30.	Get points from games and quizzes	Game, quiz	user	Get points from games and quizzes
31.	Generate leaderboard	leaderboard	User,	Generate leaderboard
32.	Access market place	User	MarketPlace	Access market place
33.	Download word list	User	Course	Download word list
34.	Daily training	User	System	Daily Training
35.	Post in forum	ForumPost	User	Posting in forum
36.	Commenting in forum	ForumComment	User	Commenting in forum
37.	Liking posts	ForumPost	User	Liking posts

7.1.2 State Transition Diagram:

The following are the state transition diagram of our project

User

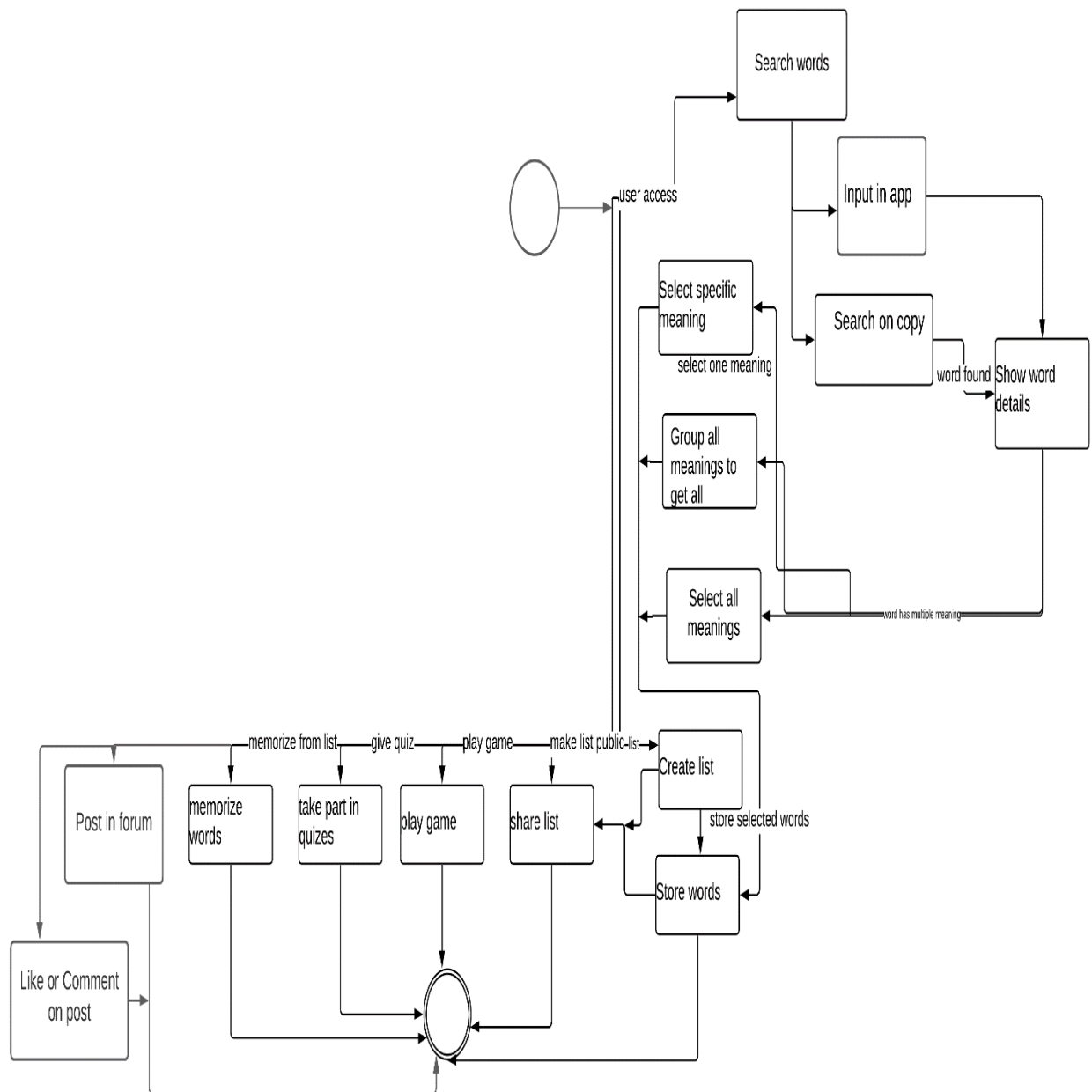


Fig: User State Transition Diagram

Description:

The diagram mainly shows the state of the user class of our application. The account class is also associated with it.

Word

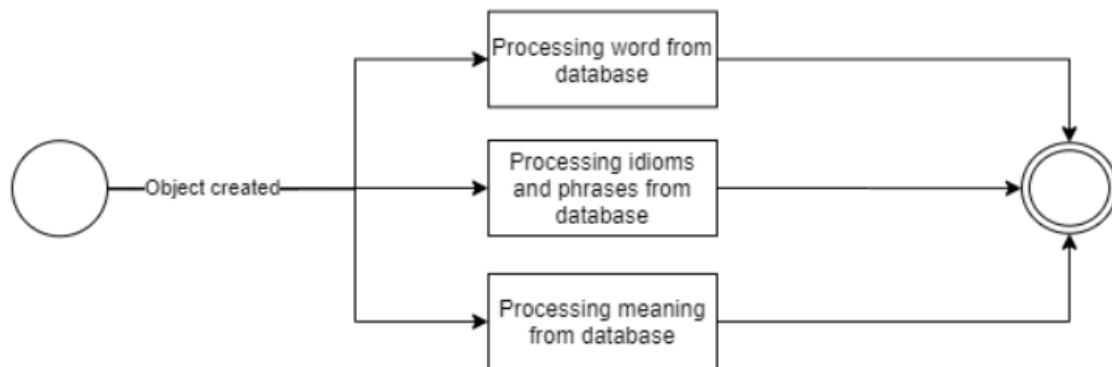
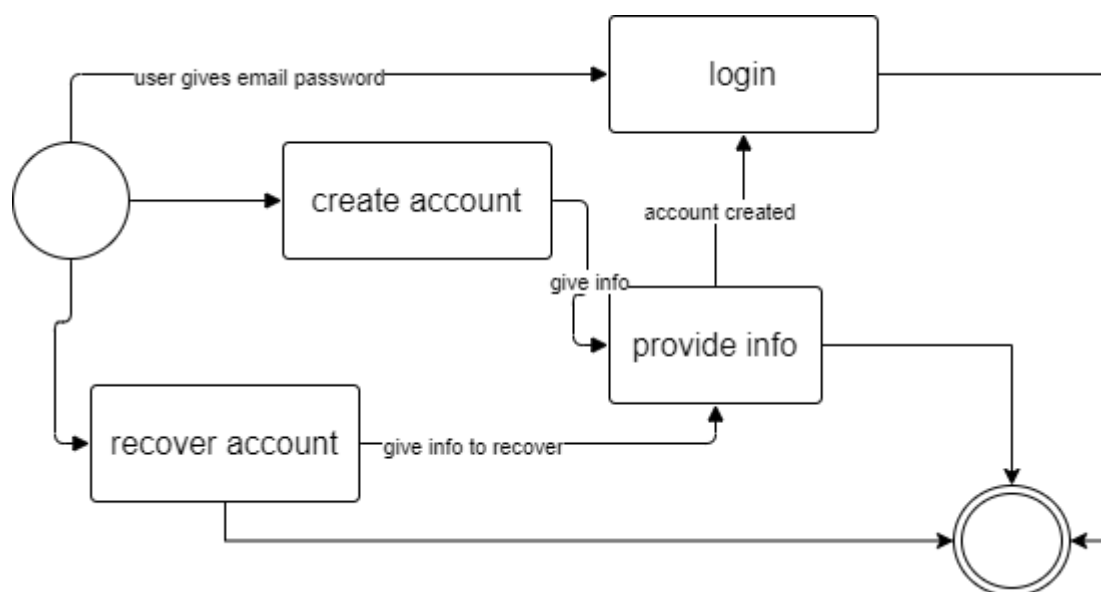


Fig: Word class state transition diagram

Description:

It mainly shows us how a word will be selected and other relating state with word class.

Account



Meaning

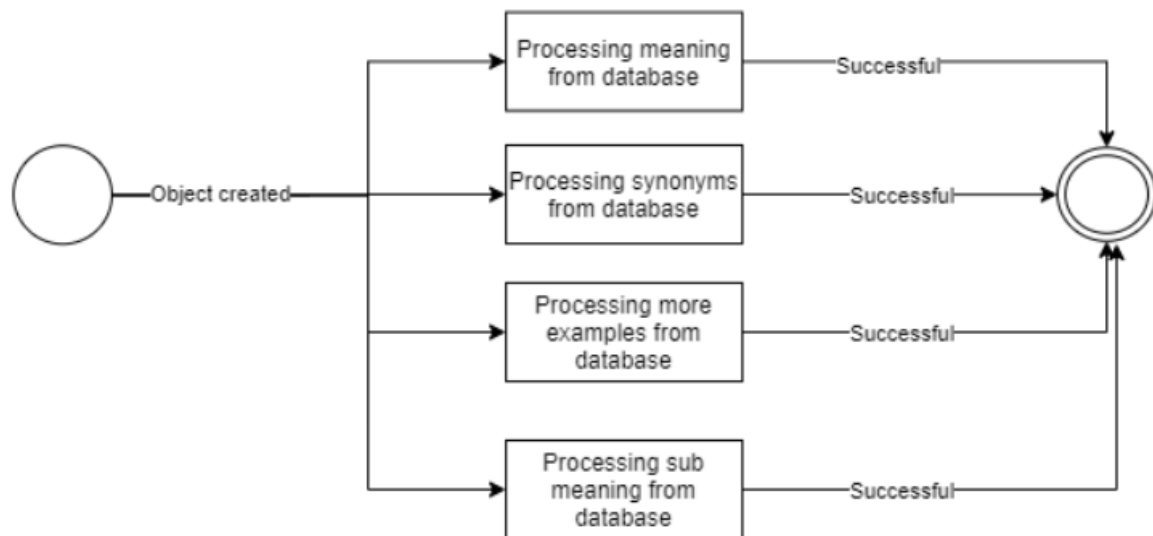


Fig: Account class state transition diagram

Fig: Meaning class state transition diagram

SubMeaning

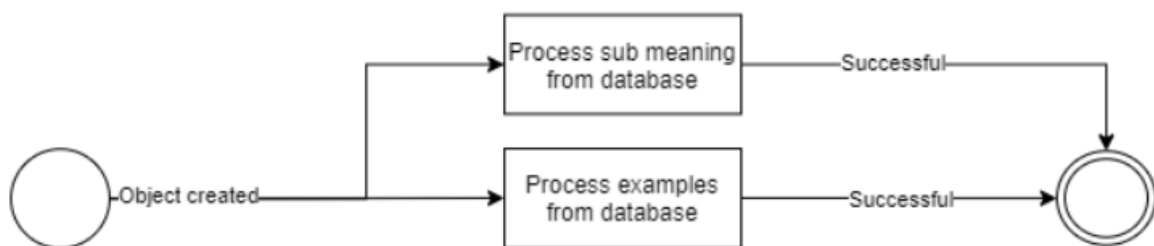
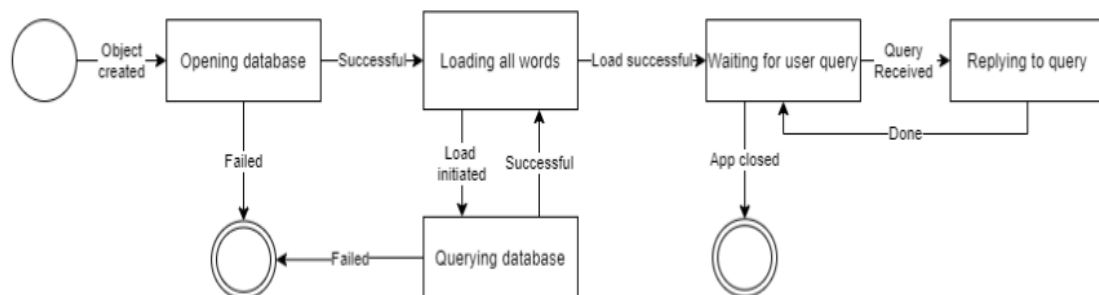
Dictionary database
controller

Fig: Sub meaning class state transition diagram

Fig: Dictionary Database class state transition diagram

Floating Bubble Handler

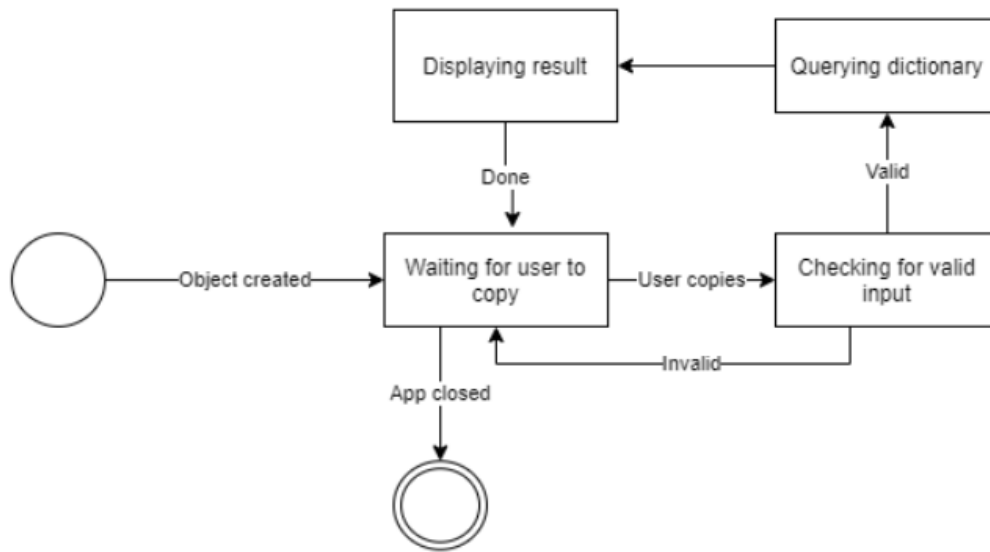


Fig: Floating Bubble Handler state transition diagram

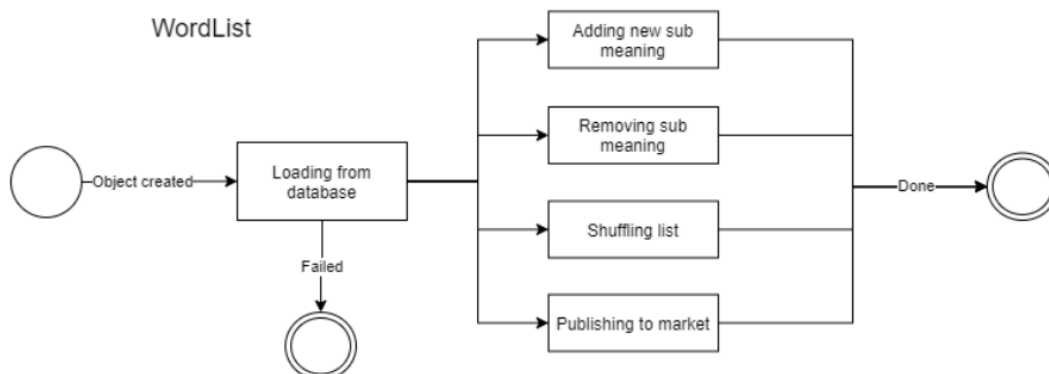


Fig: Wordlist state transition diagram

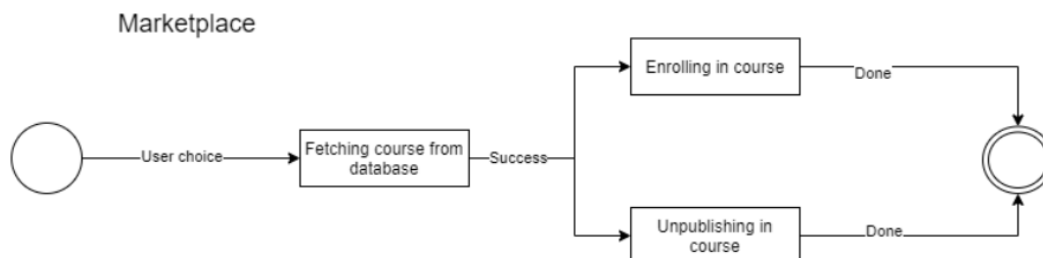


Fig: Marketplace state transition diagram

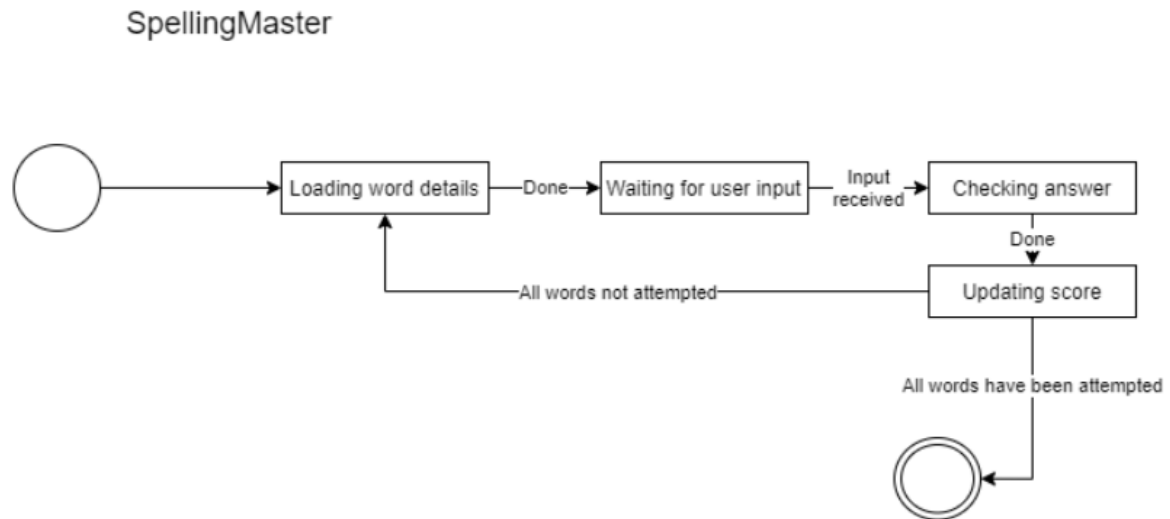


Fig: Wordlist state transition diagram

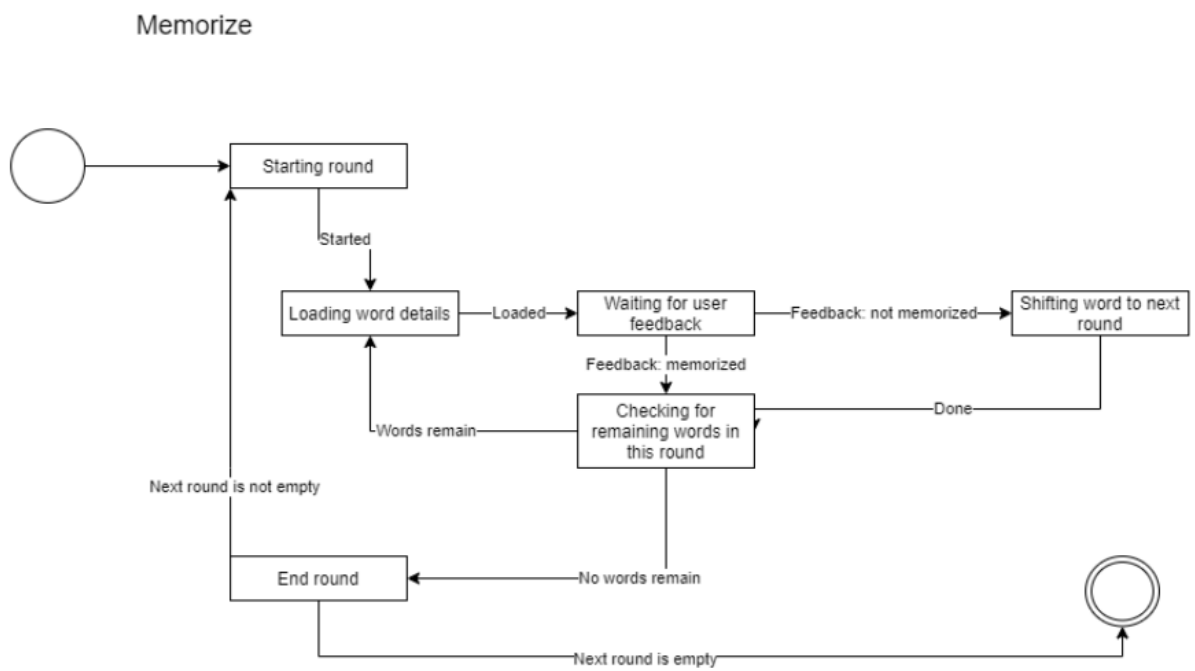


Fig: Spelling Master state transition diagram

Quiz

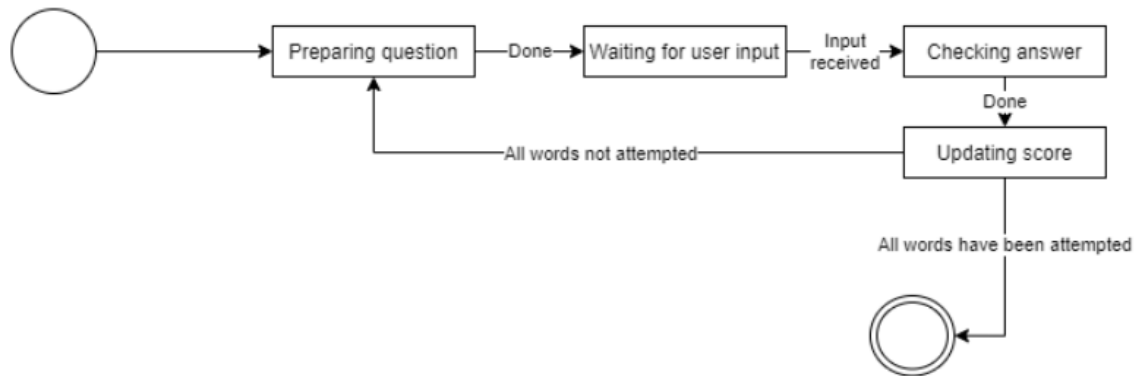


Fig: Quiz state transition diagram

Leaderboard

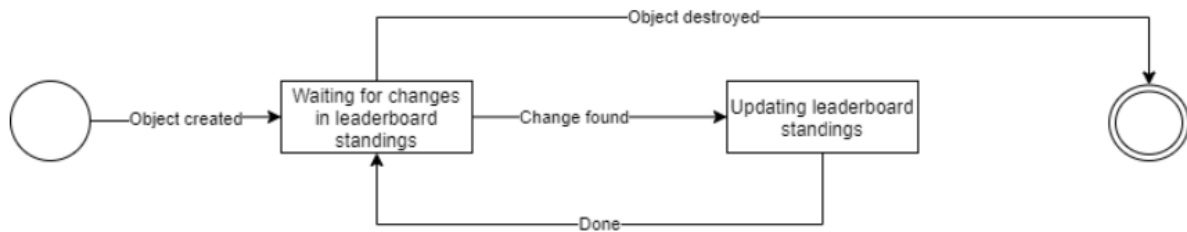


Fig: Leader Board state transition diagram

FirestoreCrud

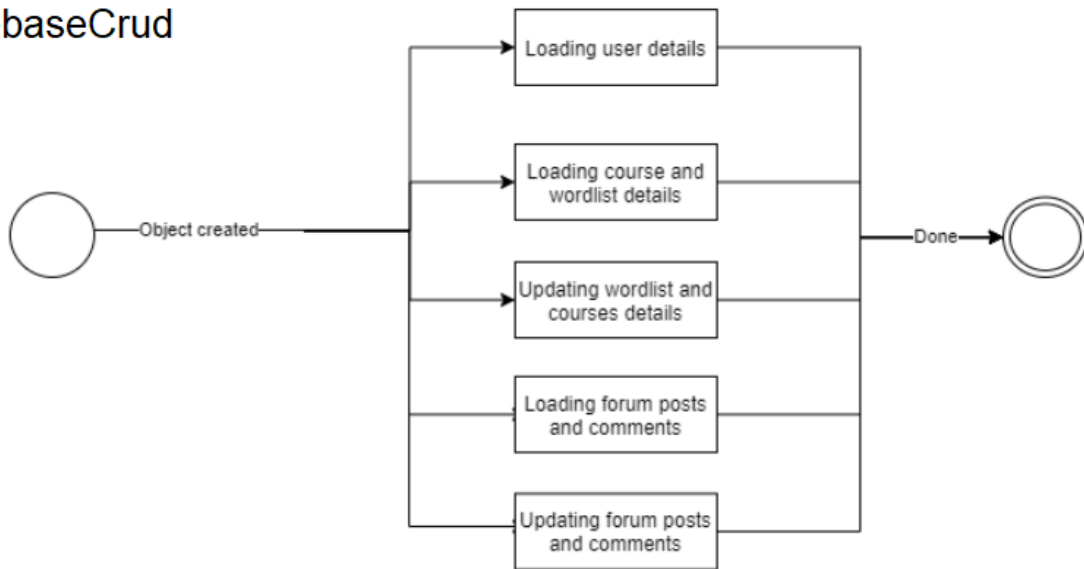


Fig: FirestoreCrud state transition diagram

ForumComment

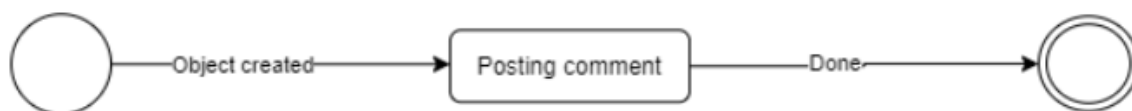


Fig: ForumComment state transition diagram

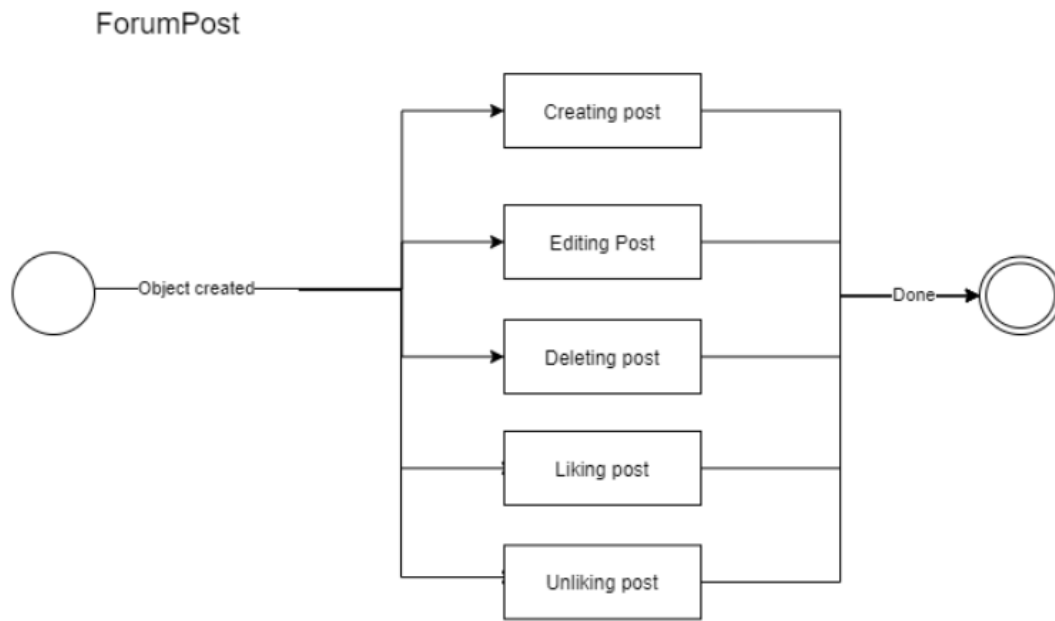


Fig: ForumPost transition diagram



Fig: Daily Training state transition diagram

7.2 Sequence Diagram

Fig: Daily Training state transition diagram

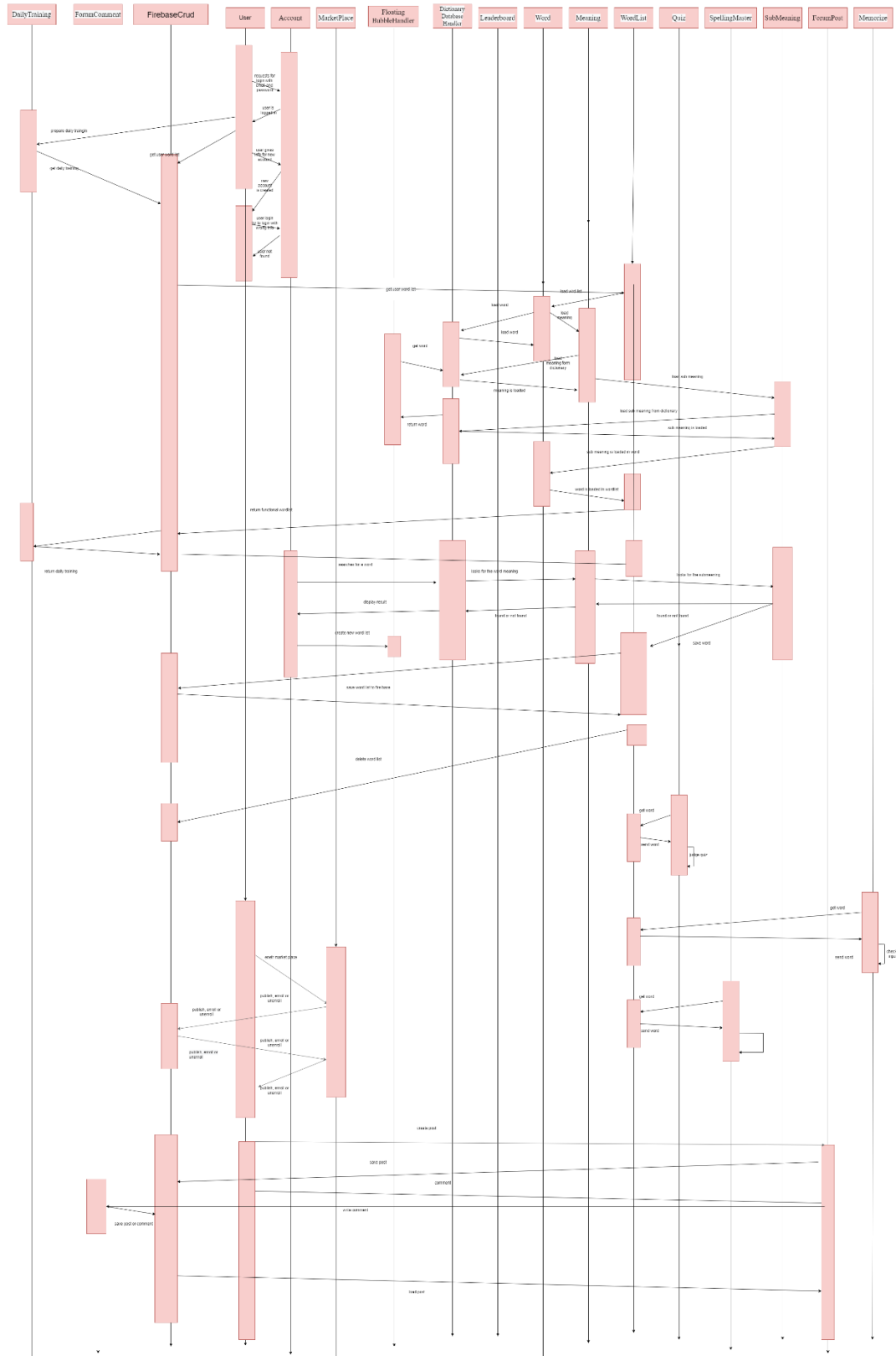


Fig: Sequence diagram