INDRAPRASTHA COLLEGE FOR WOMEN

UNIVERSITY OF DELHI

DEPARTMENT OF COMPUTER SCIENCE

SOFTWARE ENGINEERING PROJECT

(UNIQUE PAPER CODE: 32341402)

‘Morबोल’

<u>Submitted by:</u>

Aditi Upadhyay (21/CS/02)

Durgesh Choudhary (21/CS/13)

Nagadi Leela Rao (21/CS/28)

<u>Submitted to:</u>

Ms. Vimala Parihar

ACKNOWLEDGEMENT

It is our privilege to express our sincerest regards to our Software Engineering professor, Ms. Vimala Parihar, for her invaluable inputs, able guidance, encouragement, whole-hearted cooperation, and constructive criticism without which this project would not have been able to sail through. We deeply feel honoured and blessed for getting this opportunity of being guided by her. We express our sincere thanks to her for being there constantly for help and supporting, encouraging, and guiding us to successfully complete and present this project named “Morबोल”.

Aditi Upadhyay

Durgesh Choudhary

Nagadi Leela Rao

CERTIFICATE

This is to certify that Ms. Aditi Upadhyay, Ms. Durgesh Choudhary and Ms. Nagadi Leela Rao have successfully carried out the completion of the project entitled “Morबोल” under my supervision. The project has been submitted as per the requirements of Lab based on Software Engineering in the fourth semester of B.Sc.(H) Computer Science.

Ms. Vimala Parihar

(Project-in-charge)

INDEX

PRODUCT DESCRIPTION

1.1 INTRODUCTION

As per the Indian constitution, there are 22 official languages of India including Hindi, Tamil, Telugu, Kannada, Punjabi, Bhojpuri, Bodo, Malayalam, Sanskrit, etc. According to the 2011 census, Hindi is the mother tongue of around 40% Indians. Thus, most Indians have a mother tongue other than Hindi.

If you are trying to learn to speak a language at your own pace, then you can use apps like Duolingo, Mango languages, Busuu, etc to do so. There are options easily available to learn Hindi, but it’s not the same for other Indian languages, especially not one like Duolingo.

Thus, anyone wanting to learn an Indian language in an easy and fun manner would like to have access to an app like “MorBol”.

MorBol, that stands for ‘Mor’ (peacock, i.e., the national bird or ‘More’) and ‘Bol’ (words), is a user-friendly Indian language learning app with the following features:

1. Structured learning courses with questions for regular assessment through verbal, listening and reading practice.
2. Sets up daily goals for the user to achieve, encouraging them to take small steps at a comfortable pace and keeps them interested. The user is also notified as a reminder for the same.
3. Option of conversing with a chatbot and joining communities to help the user in improving their communication skills in that language, since conversation builds better understanding of any language.
4. Gamified quiz to compete with other learners and practice.
5. Feedback option to register a complaint, query or provide feedback about the usage of the app or course.
6. App language options in Indian languages other than English for accommodating a larger diverse audience.

In the wake of the disappearance of languages around the world, it is imminent for us to conserve and propagate the use of our native languages. This app aims to aid the same goal and make the process of learning an Indian language fun!

OBJECTIVES

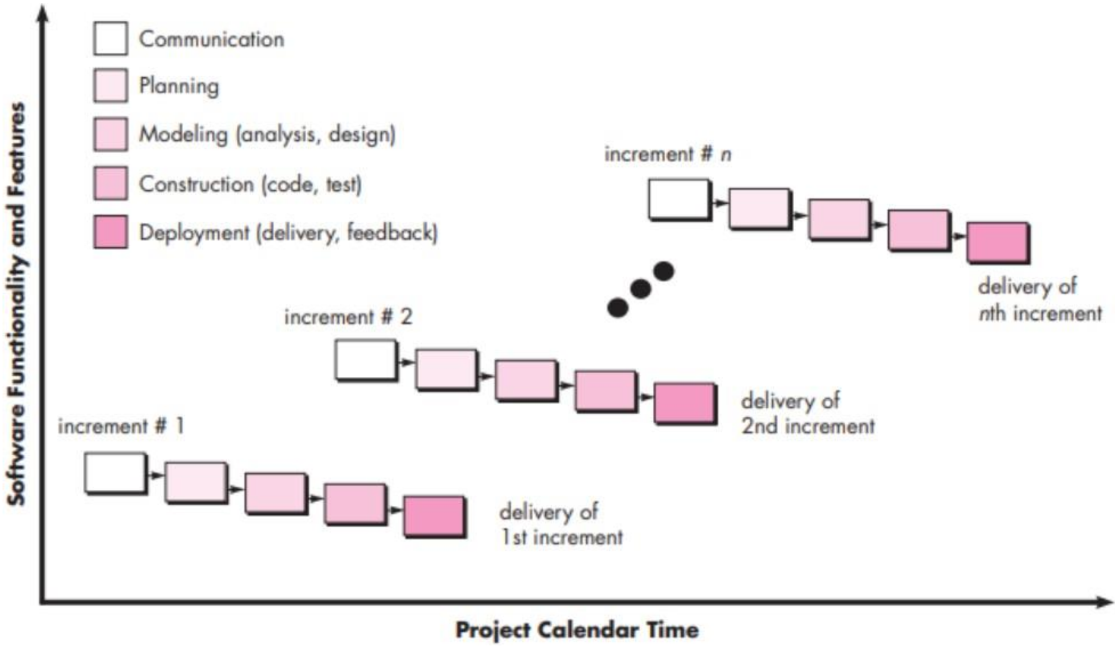
Following are the objectives of ‘MorBol’:

1. **Learning activity:** To train the app users effectively in speech comprehension via listening exercises and visual content as well as helping them in learning to read and understand text.
2. **Language Practice:** To allow users to create or choose a community where they can chat and exchange their knowledge. They can also converse with a chatbot for the same. Users are also presented with questions to answer after completing each level.
3. **Quizzes:** Not making learning boring by adding interesting quizzes to engage users and to motivate them through accomplishments and recognition.
4. **Progress-tracking:** To check level of proficiency and find all information regarding user’s progress.
5. **Vocal practice questions:** To improve user’s pronunciation.
6. **Daily goals:** To keep the user on track with the learning course and notify them accordingly.
7. **Feedback:** To allow the users to register a complaint or give their inputs for future improvements.

1.2 PROCESS MODEL

Incremental Model

Incremental Model is a process of software development where requirements are divided into multiple standalone modules of the software development cycle. In this model, each module goes through the requirements, design, testing and implementation phases. Every subsequent release of the module adds function to the previous release. The process continues until the complete system is achieved.



The various phases of incremental model are as follows:

1. Requirement analysis: In the first phase of the incremental model, the product analysis expertise identifies the requirements. And the system functional requirements are understood by the requirement analysis team. To develop the software under the incremental model, this phase performs a crucial role.

2. Design & Development: In this phase of the Incremental model of Software Development Life Cycle, the design of the system functionality and the development method are finished with success. When software develops new practicality, the incremental model uses style and development phase.

3. Testing: In the incremental model, the testing phase checks the performance of each existing function as well as additional functionality. In the testing phase, various methods are used to test the behaviour of each task.

4. Implementation: Implementation phase enables the coding phase of the development system. It involves the final coding after design in the designing and development phase and testing the functionality in the testing phase. After completion of this phase, the number of the product working is enhanced and upgraded up to the final system product.

Why are we using this model?

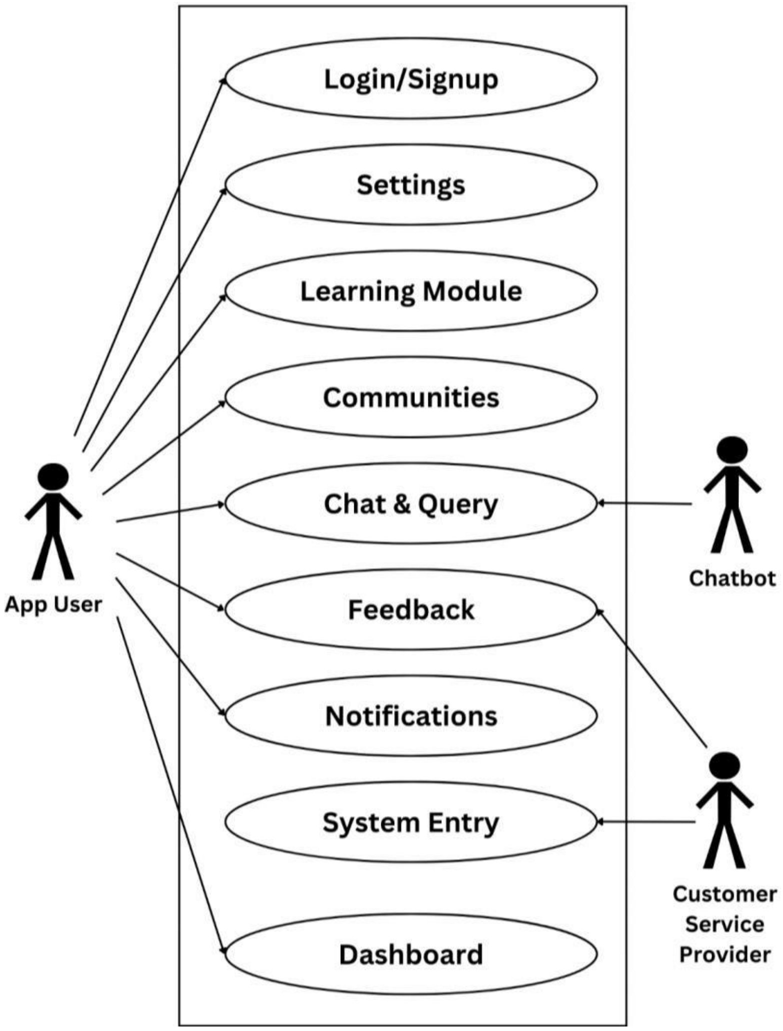
- Incremental process is best suited for our language learning app (MorBol) because this model is very flexible and can be adapted to changing requirements and features. Our app needs to be updated frequently to accommodate new questions, course material, and user preferences.
- In the app, it's important to continuously gather feedback from users to improve the app and make sure to meet their needs. The incremental model allows for continuous iteration.
- It is effective in risk management, as it allows for risks to be identified which helps to reduce overall project risk.

Overall, the incremental process model is a good fit for our app as it allows for continuous feedback and improvement, flexibility and adaptability, risk management and early delivery of working software.

REQUIREMENT ANALYSIS

2.1 USE CASE DIAGRAM

Use cases are structured outline or template for the description of user requirements, modelled in a structured language like English. It describes the sequence of interactions between actors and the system necessary to deliver the services that satisfies the goal. Here, the system is treated as a 'black box', and the interactions with the system including responses, are as perceived from outside the system.



2.2 USE CASE TEMPLATE

1. Login/Signup:

1.	**Brief Description:** This use case manages the user's entry into the app to access the language courses and services provided by the app. The user is not required to login every time they use the app unless the user has logged out.
2.	**Actors:** The following actors interact and participate in this use case: App User
3.	**Flow of Events:** **3.1 Basic Flow** This use case begins when the user wants to access the app's features. 1. User is asked to choose an app interface language. 2. The system asks for the user's credentials, i.e., a login id (in case the user is logging in using Google or Facebook account) and a password (in case the user wants to create a MorBol account). 3. The user provides the login id and/or password. 4. The system validates the credentials and provides access to the user. **3.2 Alternate Flow** **3.2.1 User does not exist in the system** The use case allows new users to sign up on the app by creating an account with a login id and a password or by signing up through Google or Facebook account. 1. User is prompted to provide a login id and password. 2. The app makes an entry in the users database and prompts the user to enter a name, username and choose an avatar. 3. User provides the username and avatar preference for the profile. 4. The system provides access to the user with the created profile. **3.2.2 User has forgotten the password** It allows the user to access the app in case they forget the password by resetting it after providing a valid login id. **3.2.3 User has provided incorrect credentials** In such a case the user will be displayed with a message that they have entered incorrect credentials. They can choose to re-enter the credentials.
4.	**Special Requirements:** None
5.	**Pre-conditions:** In case the user wants to login or sign up with a Google or Facebook account then they should have a valid login id.
6.	**Post-conditions:** If the use case is successful, the user gains access to the app's courses and services- communities, chatbot, settings, dashboard, and feedback option
7.	**Extension Points:** None

2. Settings:

1.	**Brief Description:** This use case manages the user's profile and app preferences.
2.	**Actors:** The following actors interact and participate in this use case: App User
3.	**Flow of Events:** **3.1 Basic Flow** The use case starts when the user wishes to edit profile or change app settings and appearance or access the help desk. **3.1.1 Profile settings** User can change name, username, avatar, and password. **3.1.2 General settings** User can change the app interface language, mute/unmute notifications and can manage sound effects, theme. **3.1.3 Help desk** In case of any common query, the user can access the helpdesk to find answers to Frequently Asked Questions (FAQs). The user can also access the terms and conditions here. **3.2 Alternate Flow** **1. 3.2.1 Question is not present in FAQs** The app user will be given an option to submit feedback in the feedback section to submit a query that will be taken up by a customer service provider
4.	**Special Requirements:** None
5.	**Pre-conditions:** User must login into the app to change preferences.
6.	**Post-conditions:** If the use case is successful, changes as per the preferences will be reflected in the app. The user can get answers to common questions from the help desk and view the terms and conditions.
7.	**Extension Points:** Notifications, Feedback

3. Learning Module:

1.	**Brief Description:** This use case allows the user to access a language course and learn a language through audio, video as well as textual content. It also allows the user to assess themselves through gamified quizzing and compete with other users. Daily goals are set for the user to complete and maintain the momentum. Users can also select and switch language courses.
2.	**Actors:** The following actors interact and participate in this use case: App User
3.	**Flow of Events:** **3.1 Basic Flow** The use case starts when the user wants to learn a language or assess themselves through quizzing games.

3.1.1 Language Course preference

The user chooses the language that they want to learn. If the user is learning more than one language simultaneously, then they can switch between language courses.

3.1.2 Course work

It allows the user to learn the chosen language. As the user proceeds through the course, the progress is updated by the app.

1. The module presents the user with various levels to complete, with each level consisting of some tasks.
2. The user starts with the current level and completes the task (listening to an audio, watching a video, or reading textual information).
3. After completion of each task, the user answers 2-3 questions.
4. On answering all correctly, the app awards the user with some points which are added to the dashboard and unlocks the next level.

3.1.3 Daily goal completion

Based on the user’s progress, i.e., the number of levels cleared, the app notifies the user to complete daily goals, which consist of 5 questions chosen to revise what has been learnt previously and the tasks to be completed for the day. Completing these goals everyday will allow the user to maintain his/her streak and gain points.

1. The app notifies the user through a notification to complete the daily goal.
2. The user clicks on the notification.
3. The user answers the 5 questions.
4. If answered correctly, then the app awards the user with 50 points.
5. The user is then redirected to the course, to complete a specified number of tasks.
6. The app awards the user with a streak after completing these tasks and updates the dashboard.

3.1.4 Quizzing game

The user can choose to compete with other users in a fun gamified quiz with 15 questions. The standings are displayed on the leader board on a real time basis and the result is updated in the dashboard.

1. User clicks on ‘Play’.
2. The app creates a room for 15 users currently logged in and starts the game.
3. The user competes with other users by answering the interactive questions as fast as possible, each question answered wrong will lead to loss of 5 points.
4. The game ends after answering all the questions. The leader board is updated with users ranked based on number of correctly answered questions (10 points for each correct answer) and the least time taken.
5. The dashboard is updated with the result.

	3.2 Alternative Flow **3.2.1 User wants to exit a course** If the user doesn't wish to continue a particular course, then they choose to reset and remove a course while switching courses. **3.2.2 User wants to go back to previous tasks** The user can simply click on the task tab that they want to retake and proceed. If it's a task from a previous level, then the user can click on the respective level. The progress of the levels will remain intact. **3.2.3 User has given incorrect answer** For every question answered wrong, including the ones in daily goals, the app will present the user with an extra question to ensure that they have understood the concept and can then proceed ahead. **3.2.4 User wants to exit a game in between** If the user doesn't want to continue the game, then they can click on the exit option, but they will not gain any points. **3.2.5 User wants to jump levels** The user will be presented with 10 questions based on the levels preceding the one that they want to unlock. On getting all questions correct, the level will be unlocked, and the user will gain 100 bonus points.
4.	**Special Requirements:** None
5.	**Pre-conditions:** User must login into the app to access the courses based on their preference.
6.	**Post-conditions:** If the use case was successful, the user can: 1. Select the language the user currently wants to learn. 2. Switch languages. 3. Learn the selected language. 4. Complete daily goals. 5. Assess their learning through gamified quiz and compete with other users by playing games/quizzes.
7.	**Extension Points:** Notifications, Dashboard

4. Communities:

1.	**Brief Description:** This use case manages the communities (based on languages) that an app user is a member of and provides chatting options with other members. This also handles individual community's notifications and allows the app user to create their own community as well. The user can be part of more than one community.
2.	**Actors:** The following actors interact and participate in this use case: App User
3.	**Flow of Events:** **3.1 Basic Flow** This use case begins when the app user is ready to choose a community based on the language they are learning or want to learn. 1. The app user searches for communities and selects one to join. 2. The app allows the app user entry into the chosen community and allows the user to see the members, admins and previous chats. 3. The app user can send and react to others' messages and mute/unmute notifications for that community. **3.2 Alternate Flow** **3.2.1 User wants to create a new Community** The use case allows the app user to create his/her own community and invite people to it. As the admin, user can set group profiles and descriptions, and remove users if they disregard community guidelines. **3.2.2 User wants to exit the Community** It permits the app user to exit a community as and when they want to. After doing so, they won't be able to access the community chats. **3.2.3 A member is not adhering to community guidelines** The use case permits the user to report a member for causing tension or ruckus in the group. A background check will be done on that member and his/her account will be terminated accordingly. The admin can also choose to remove that member and report them.
4.	**Special Requirements:** None
5.	**Pre-conditions:** All the new users must login into the app and choose/create a community to execute the use case.
6.	**Post-conditions:** If the use case is successful, it provides the user the following options: 1. Create, choose, or exit a community. 2. Chat with fellow members in the community and react to their messages. 3. Handle community-based notifications: Mute/Unmute.
7.	**Extension Points:** Notifications

5. Chat & Query:

1.	**Brief Description:** This use case allows the app user to ask their doubts/interact with the chatbot.
2.	**Actors:** The following actors interact and participate in this use case: App User and Chatbot
3.	**Flow of Events:** **3.1 Basic Flow** This use case begins when the user clicks on the chatbot icon to have a chat with it or to get rid of a doubt they have regarding their courses' language. **3.2. Alternate Flow** None
4.	**Special Requirements:** None
5.	**Pre-conditions:** This use case requires the user to be logged in to the app.
6.	**Post-conditions:** If the use case is successful, it allows the user to do the following: 1. Interact with the chatbot in any of the supported languages. 2. Clear doubts regarding meanings or semantics of a particular language.
7.	**Extension Points:** None

6. Feedback:

1.	**Brief Description:** This use case takes the feedback given by the users and is accessed by the customer service provider who looks into these feedbacks and provides response to the same.
2.	**Actors:** The following actors interact and participate in this use case: App User and Customer Service Provider
3.	**Flow of Events:** **3.1 Basic Flow** This use case begins when the app user wants to send feedback to the developers regarding the interface, usability, or the course material itself or submit a complaint. The customer service provider replies to these feedbacks, informs the developer team regarding any inconvenience the users face and resolves any issues as mentioned in the app user's complaint. **3.2 Alternate Flow** **3.2.1 The server is down** In this case, feedbacks cannot be sent so- 1. Users can report the issue. 2. Or check after some time. 3. Write an email to the organization's mail id provided.
4.	**Special Requirements:** None
5.	**Pre-conditions:**

	This use case requires the user to be logged in to the app and the CSP to be logged into the back-end system.
6.	**Post-conditions:** If the use case is successful, the app user will be able to 1. Send feedback. 2. Write complaints. 3. View feedback status. 4. Receive replies to their feedback and resolution of their issues.
7.	**Extension Points:** None

7. Notifications:

1.	**Brief Description:** This use case displays all the necessary notifications to the user regarding their progress, daily goals, and new messages they get on the communities that they are a member of. It also notifies the user when an updated version of the app is available.
2.	**Actors:** The following actors interact and participate in this use case: App User
3.	**Flow of Events:** **3.1 Basic Flow** The app user gets the new notifications immediately on his/her smartphone on which the app is installed or on their Google email id(only to notify about pending goals and weekly progress) while the user is not using the app or is not on the respective community's page currently. The use case will receive data from the following: • The learning module sends an update on the daily goals to be completed and shares weekly progress. • Any new messages in the user's communities will be redirected. It also displays notifications for app updates. **3.2 Alternate Flow** Users may mute the notifications from app settings or system settings and can only see the notifications while using the app by clicking on the notifications icon.
4.	**Special Requirements:** None
5.	**Pre-conditions:** User needs to be logged into the app.
6.	**Post-conditions:** User will be able to view and access notifications regarding progress, daily goals, and messages from communities.
7.	**Extension Points:** Learning Module, Communities

8. System Entry:

1.	**Brief Description:** This use case manages the entry of the backend users into the system for usage or service providing purposes.
2.	**Actors:** The following actors interact and participate in this use case: Customer Service Provider
3.	**Flow of Events:** **3.1 Basic Flow** This use case begins when a backend user tries to enter the system to provide various services to the users: 1. The CSP provides a CSP number and password to enter the system. 2. The CSP’s credentials are verified and provided entry into the system accordingly. 3. They will get access to feedbacks that have to be addressed. 4. The CSP will address the feedbacks and issues and update the status of each one as it is resolved. **3.2 Alternate Flow** **3.2.1 Backend User entered invalid credentials** If in the basic flow the actor enters invalid credentials, they won’t be allowed entry into the system and the system will display an error message. The actor can choose to either return to the beginning of the Basic Flow or cancel the login, at which point the use case ends.
4.	**Special Requirements:** None
5.	**Pre-conditions:** The backend users must have the correct system entry credentials to login.
6.	**Post-conditions:** If the use case is successful, the user seeking entry will be allowed to enter in the system and handle operations.
7.	**Extension Points:** None

9. Dashboard:

1.	**Brief Description:** This use case allows the user to view their progress and leader board positions along with different statistics.
2.	**Actors:** The following actors interact and participate in this use case: App User
3.	**Flow of Events:** **3.1 Basic Flow** This use case begins when the app user has selected a language, which they want to learn and can see the following in the dashboard tab: • Name, username, avatar. • Progress in each language course. • Number of streaks after daily goal completion. • Leader board positions. • Points collected from answering questions and completing levels. All updates from the learning module will be redirected to the dashboard to be displayed. **3.2 Alternate Flow** If the user didn’t select any language or didn't start the lesson, they won't be able to see any progress.
4.	**Special Requirements:** None
5.	**Pre-conditions:** User is logged into the app and has chosen a language to learn.
6.	**Post-conditions:** If the use case is successful, the user can see his/her progress in the selected language and other stats.
7.	**Extension Points:** None

2.3 DATA FLOW DIAGRAMS (DFD)

Data flow diagram takes an input-process-output view of a system. It is a graphical representation of flow of data in an information system. It can depict incoming data flow, outgoing data flow, and stored data. That is, data objects flowing into the software, are transformed by processing elements, and resultant data objects flow out of the software.

DFD Components

DFD can represent source, destination, storage, and flow of data using the following set of components:

Entities: Entities are sources and destinations of information data. Entities are represented by rectangles with their respective names.

Process: Activities and action taken on the data are represented by Circle or Round edged rectangles.

Data Flow: Movement of data is shown by pointed arrows. Data movement is shown from the base of arrow as its source towards head of the arrow as the destination.

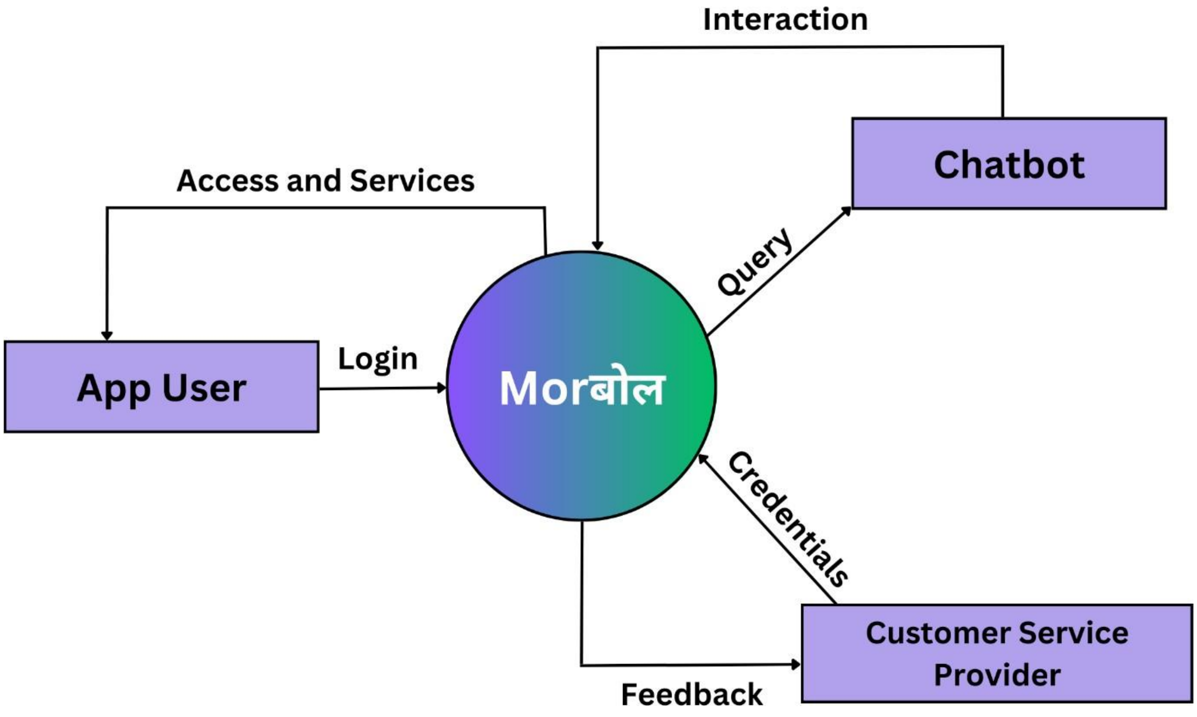
Database: The databases and data stores used in this project are represented by two parallel lines.

A few simple guidelines can aid immeasurably during the derivation of a data flow diagram:

- The level 0 data flow diagram should depict the software/system as a single bubble.
- Primary input and output should be carefully noted.
- Refinement should begin by isolating candidate processes, data objects, and data stores to be represented at the next level; all arrows and bubbles should be labelled with meaningful names.
- Information flow continuity must be maintained from level 1 to level 2 and one bubble at a time should be refined. This is a natural tendency to overcomplicate the data flow diagram.

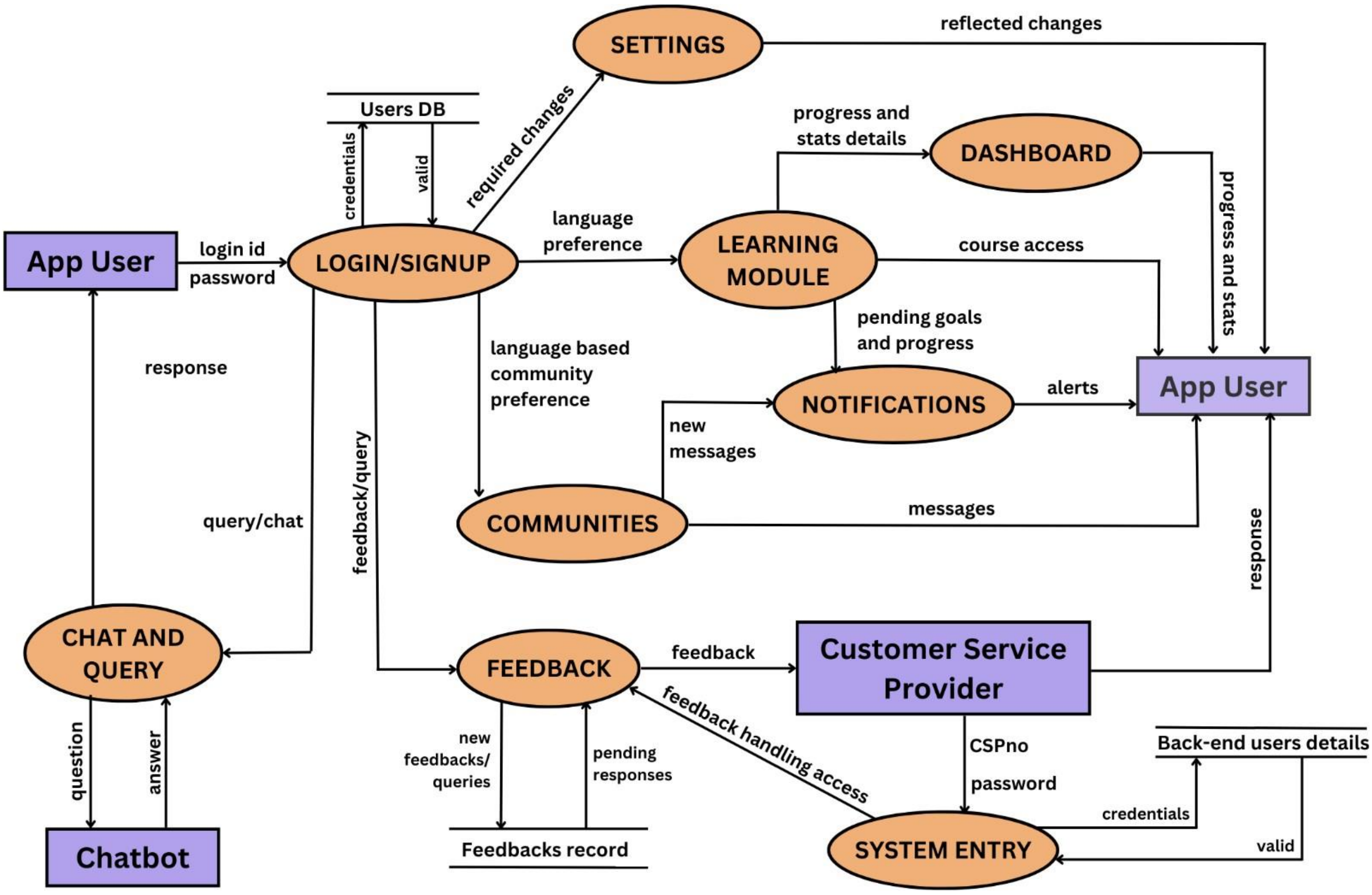
ZERO-LEVEL DFD:

Zero level DFD depicts various external entities.



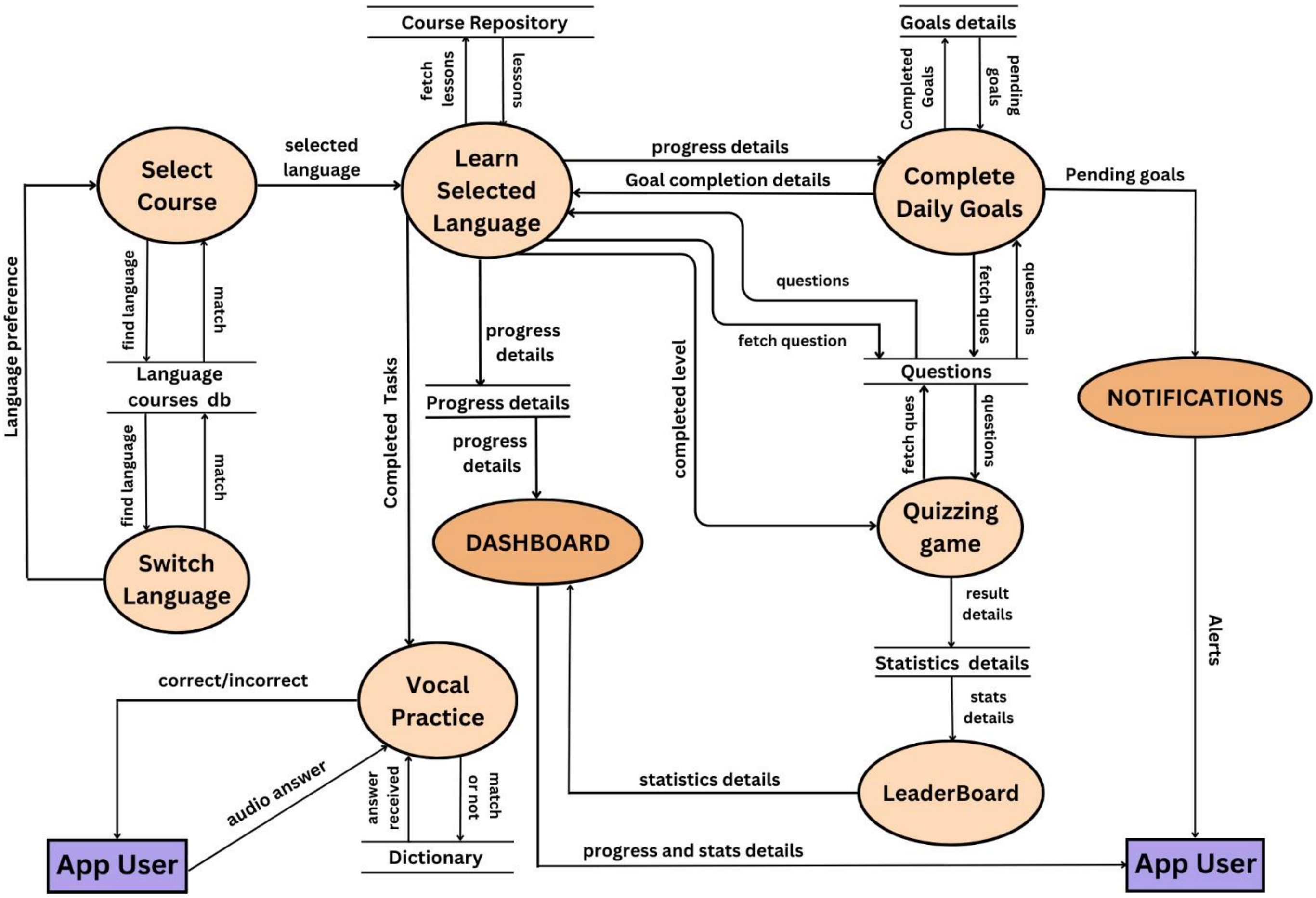
LEVEL - 1 DFD:

First level DFD depicts the process how the user is going to login in the system and the processing of different data going into the different modules and databases, and from there back to the user.



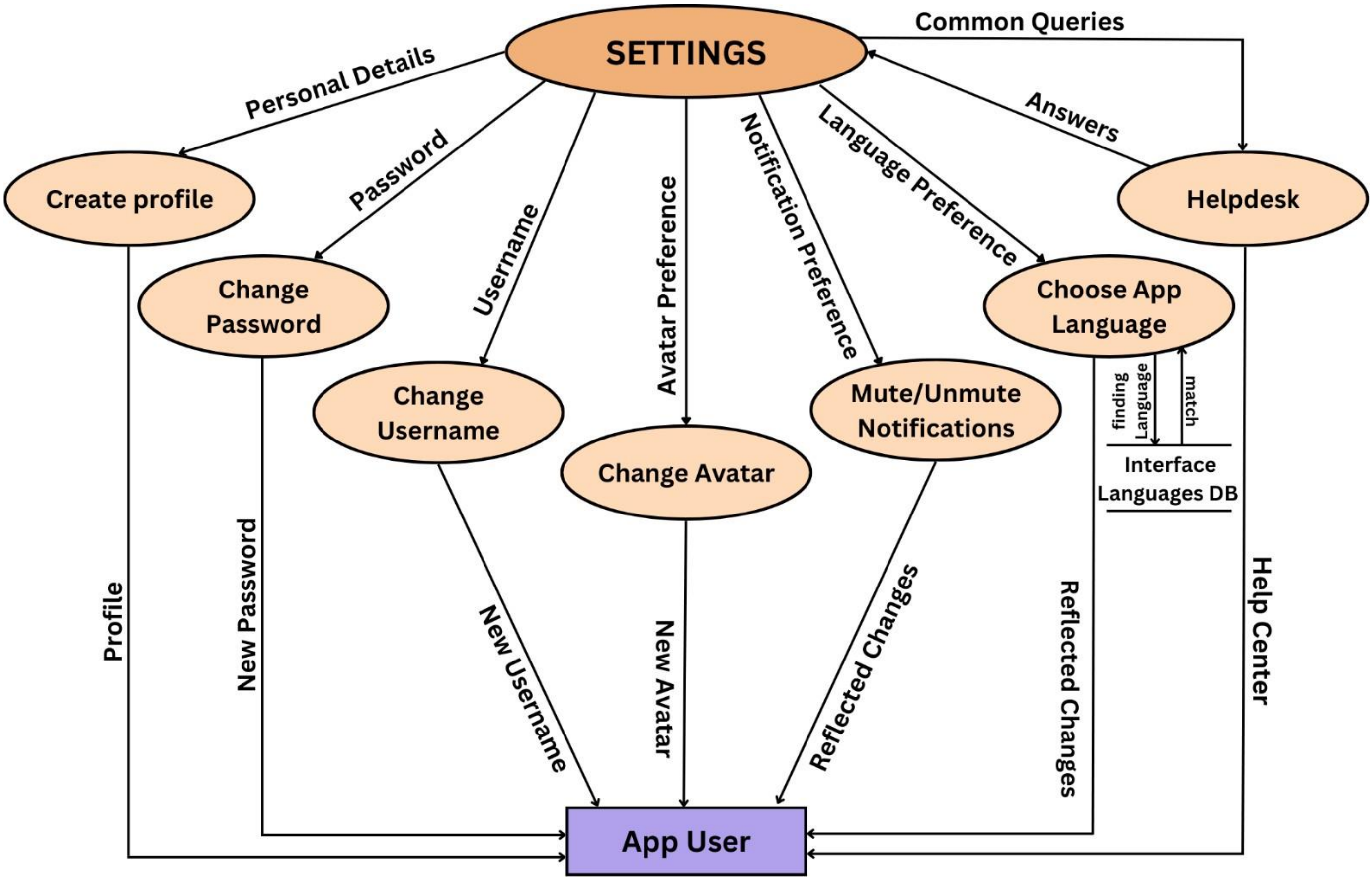
LEVEL – 2A DFD:

Shows the inner view of the Learning Module.



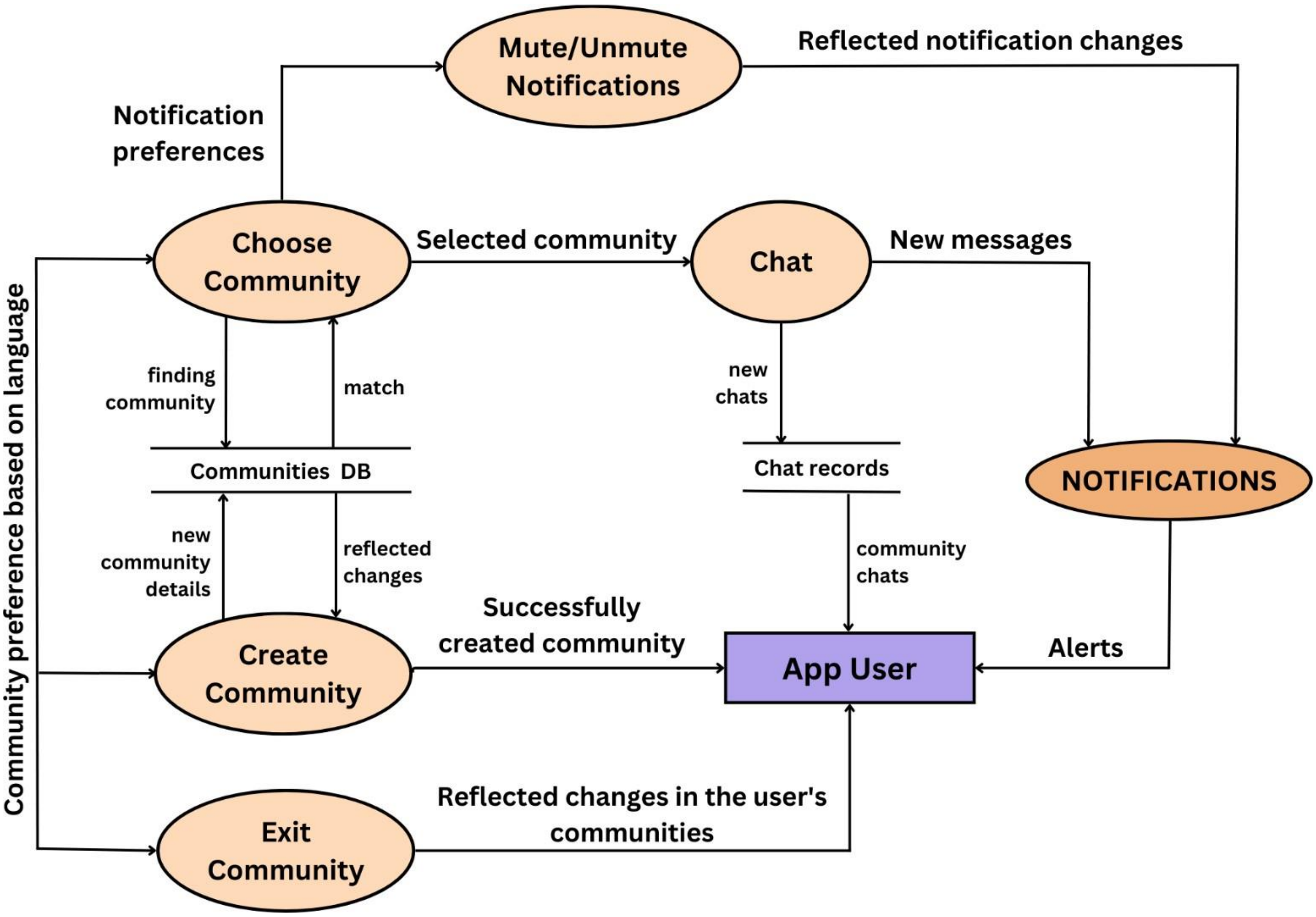
LEVEL – 2B DFD:

Shows the inner view of Settings module.



LEVEL – 2C DFD:

Shows the inner view of Communities module.



2.4 DATA DICTIONARY

A set of information describing the contents, format, and structure of a database and the relationship between its elements, used to control access to and manipulation of the database.

The data dictionary is a crucial component of any relational database. Ironically, because of its importance, it is invisible to most database users. Typically, only database administrators interact with the data dictionary.

Most of the data dictionary contains the following information:

- **Name -** the primary name of the data or control item, the data store or an external entity.
- **Alias** - other names used for the first entry.
- **Where used -** a listing of the processes that use the data or control item and how it is used external.
- **Description** - a notation for representing content.

S.NO.	NAME	ALIAS	WHERE USED	DESCRIPTION
1	App user	User	Used wherever the user is involved	Represents a registered user of the app, and contains information such as name, email, password, username, IdType and avatar preference
2	Login id and password	Credentials	Used during login/signup	It is specific to every entity and is used as a login detail
3	Customer Service Provider	CSP	Used during system entry and feedback response	Refers to the person employed to respond to users' queries and feedbacks
4	CSPNo and password	CSP credentials	Used during system entry	It is specific to every customer service provider and is used to log into the back-end system
5	Gamified quiz	Quizzing game	Used when the user wants to play the language-based game	Refers to the quiz-based game played by the user to assess himself and compete against other users for a leader board position
6	Learning module	Language courses	Used when the user wants to select a language to learn	Refers to the list of language wise courses available for the user to take up and learn, play games and complete daily goals
7	Course repository	Course Coure work Course content	Used when user wants to learn the selected language	Refers to the course content of each language course which is distributed into levels, that are further divided into tasks
8	Feedback	Query Complaint	Used when the user wants to submit feedback regarding the app or coursework	Refers to a feedback, query or complaint that the user has about the usage of the app or the course content which may be positive or negative
9	Statistics	Stats	Used when the user plays a game and the details about his performance are accessed and sent to dashboard from leader board	Refers to the score, time taken and leader board position earned by the user in each game played
10	Chatbot	ChatBot API	Used when the user opens an interface to the Artificial Intelligence chatbot to chat or query regarding a language	Refers to the third-party software Application Programming Interface inculcated into the app through an interface for the user to interact with

11	System entry	Back-end entry	Used when a CSP is trying to log into the back end of the software	Refers to the back-end part of the software not visible to app users that CSPs are part of
12	Community	Language based community	Used when the user wants to join groups that focus on a particular language	Refers to the language-based groups available for the user to join or create their own and interact with other app users
13	Dashboard	—	Used when the user wants to check his progress and stats	Refers to the page visible that shows the user’s name, username, avatar along with progress and stats
14	Avatar	Avatar image	Used when the user sets up the profile image when signing up or changing it from settings	Refers to the graphic image provided by the application for the user to use as the profile image and remain pseudonymous
15	Progress	Progress details Progress bar	Used when the user completes a level and is reflected in the dashboard	Refers to the number of levels completed in a language course that is depicted as a bar in the dashboard
16	Notification	Alert	Used when the user is notified about pending goals and new chats	Refers to the notifications generated to remind the user to complete his goals or any new messages in the joined communities
17	Chats	Messages	Used when the user has joined or created a community	Refers to the messages sent by app users in a community
18	Daily Goals	Pending goals	Used when the user has to complete particular tasks to maintain a streak	Refers to 5 questions and specified tasks that the user has to complete in order to practice regularly
19	Interface language	App language	Used when the user has to choose what language is to be used for the app	Refers to the language chosen by the user to be able to use the app in that specified language
20	Settings	—	Used when the user has to manage profile or display settings	Refers to the gear icon present in apps in general that is used to control settings related to user profile and app appearance

2.5 SOFTWARE REQUIREMENT SPECIFICATION (SRS)

1. Introduction

This document aims at defining the overall software requirements for the application ‘**MorBol**’. Efforts have been made to define the requirements exhaustively and accurately. The final product will be having only features/functionalities mentioned in this document and assumptions for any additional functionality/feature should not be made by any of the parties involved in developing/testing/implementing/using this product. In case it is required to have some additional features, a formal change request will need to be raised and subsequently a new release of this document and/or product will be produced.

1.1 Purpose

This specification document describes the capabilities that will be provided by the software application ‘MorBol’. It also states the various required constraints by which the system will abide. The intended document audience for this document are the development team, testing team, language experts, stakeholders, service providing team and the end users of the product.

1.2 Scope

The software product ‘MorBol’ will be an Indian Languages Learning application that will be used to learn any of the 22 official Indian languages and in an interactive and interesting manner. The application will provide a user-friendly interface to take up one or more language courses, play quizzing games related to the ongoing course, be a part of various communities and enhance their speaking skills through vocal practice questions. The application will also store the information of the progress made by the user, statistics of all the games he has played and display all those details in the dashboard. He can chat with a chatbot to query about any language related doubts.

Daily goals are set and notified to the user to remind him to complete required tasks to stay on track.

The user also has the option to provide feedback about the courses or the application’s interface which is handled by a customer service provider. User can also manage his profile and display features by managing the settings.

The application will greatly simplify the process of learning an Indian language and help the interested users get what they are looking for.

1.3 Definitions, Acronyms and Abbreviations

Following abbreviations have been used throughout this document:
DBA- Database Administrator
API- Application Programming Interface
DB- Database
CSP- Customer Service Provider
RAM- Random Access Memory

1.4 Overview

The rest of the SRS document describes the various system requirements, interfaces, features, and functionalities in detail.

2. Overall Description

This portion describes the general factors that affect the product and its requirements.

2.1 Product Perspective

This application will be interacting with the chatbot API provided by a third-party firm. This is to allow the user to interact with the chatbot through the app.
A speech recognition software will be used when the user must answer a question for vocal practice. An AI based translation engine will be used to translate the app language from English to an Indian language.
The databases of the system will contain information about:

a. Users' and CSPs' details
b. Languages and courses
c. Course material
d. User's progress and stats
e. Communities and chats
f. Feedbacks

2.1.1 Hardware interfaces

- RAM: Minimum of 6 GB RAM required to be able to use along with other uses like web browsing, social media, video streaming, and some popular mobile games
- Support for Wifi or data connectivity
- Phone should have a speaker and microphone
- Minimum refresh rate of 60Hz

2.1.2 Software interfaces

1. Operating System: minimum ANDROID 8 Version or IOS 6 version
2. Database: Oracle database
3. Front End: HTML, CSS, JavaScript- for user interface development purpose
4. Back End: Java - for development purpose
5. Speech Recognition software: for processing speech to understand pronunciation
6. AI based translation engine: to implement different languages for app interface.
7. Chatbot API: which is an actor that interacts with the app user.

2.1.3 Communications interfaces

User will be notified to complete daily goals and view new chats in communities via notifications.

The notifications regarding goals will also be sent via mail to the Google email id used by the user to sing up and log into the app.

2.1.4 Memory Constraints

At least 40 MB RAM should be freely available to install and run the app.

2.1.5 Operations

The DBA will be responsible for handling the databases, including entries or modifications that are to be done manually, like course and questions repository based on the inputs provided by language experts.

Feedbacks will be handled by the CSPs who will update the feedbacks records with the responses and change the status of the query to 'resolved' once the customer is satisfied with the help provided.

Data entered or generated by the app user is directly stored into respective data stores.

2.1.6 Site Adaptation requirements

The app user's device will have to support the hardware and software interfaces as mentioned in the above sections.

2.2 Product functions

The app will allow access to app users who log into their account, after which they can access the various modules except the system entry module, which is present at the back end, that is accessed by the CSPs.

Major functions that the app will perform are-

1. Login facility to enable only existing users to access the app's features.
2. App user will be able to make changes and manage settings of profile and the app.
3. They'll be able to learn any language supported by the app and complete daily goals.
4. App user will be able to play quizzing games and assess their standing amongst the app users.
5. App user will be able to interact with other app users through communities.
6. App user and chatbot can interact with each other.
7. App user can provide feedback that will be taken up by the user (with the role of CSP) and resolved.
8. The user (with the role of CSP) will be able to record the conversations with the app user into the feedback records.

2.3 User characteristics

The app will be used by the app user with the following characteristics-

Educational qualification: No minimum educational qualification required but the user should be able to read and write in any one of the languages supported by the app. So even primary school kids can use it and if required, parents can help them in learning to use the app.

Experience: No specific experience required

Technical expertise: the user should be comfortable with using an android or iOS smartphone device, with or without accessibility options enabled.

2.4 Constraints

1. The Developers will follow a privacy policy that details how the user's information collected will be handled.
2. App users will have to agree to the terms and conditions that set the way in which the app may be used, in a legally binding way.
3. The app users must follow community guidelines to avoid any kind of content that could trigger people or social communities

2.5 Assumptions and dependencies

1. The app users are mostly of Indian nationality and if otherwise, they are well versed in English.
2. The accuracy of answers expected by the app user depends on the chatbot software while using the chatbot option.
3. The accuracy of the different language interfaces of the app depends on the AI based translation engine employed.
4. The user satisfaction with the courses depends on the capabilities of language experts in designing them.
5. The accuracy of the speech recognition depends on the tool being used.

2.6 Apportioning of requirements

Accessibility options will be looked at in future versions of the app that enable haptic response and audio option for each task and question. The software will be designed and updated accordingly.

3. Overall Specification

The major modules in this project are:

1. **Login/Signup:** helps the user login or signup in the application and use it.
2. **Learning Module:** manages language course selection, includes courses for different languages, has an option to play quizzing games and complete daily goals.
3. **Settings:** provides various options to manage profile, app display settings and access helpdesk.
4. **Dashboard:** displays progress and stats of the user.
5. **Communities:** provides access to language-based communities that user can join and allows the user to create his own community.
6. **Feedback:** handles the feedback/complaints received from the user and involves the CSP in resolving it.
7. **System entry:** provides access to the backend users (CSPs) into the system.
8. **Notifications:** handles notifications related to community chats and pending goals and any app updates available.
9. **Chat & Query:** provide interface to the chatbot that the user can interact with it as and when required.

4. Specific Requirements

This section contains all the software requirements at a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements.

4.1 External Interfaces

4.1.1 User interface

The following screens will be provided-

1. **Login/Signup page:** when user first opens the app, he will be asked to login or signup with valid credentials.
2. **Create profile:** if the user is signing up, then he will have to enter a name, username, choose an avatar and choose the app language.
3. **Home page:** this is the learning module screen that will be visible when the valid user opens the app, which will have all courses listed.
4. **Language Course page:** it will contain the learning module tab, game tab and daily goals tab.
5. **Course page:** this will contain various levels with tasks for the user to complete.
6. **Game page:** a quizzing game will be started for the user and a leader board will be shown after the end of the game.
7. **Dashboard:** will show the progress and stats of the user.
8. **Settings:** will show various options for the user to manage display and profile.
9. **Communities:** will show a list of communities joined along with a search bar.
10. **Community chat:** shown for each community.
11. **Notifications:** the user can check the notifications.
12. **Feedback page:** to provide feedback or register any complaint.
13. **Chatbot page:** to chat with the chatbot.

4.1.2 Hardware interfaces

As mentioned in section 2.1.1

4.1.3 Software interfaces

As mentioned in section 2.1.2

4.1.4 Communication interfaces

As mentioned in section 2.1.3

4.2 Performance Requirements

The performance of the software depends on the usability, automatic speech recognition, interface language translation and battery life.

- **Usability:** Usability is defined as the amount of time it takes users to complete a task using mobile applications. Usability testing will be conducted via user testing of the software product.
- **Automatic-speech recognition:** The ASR results and analysis should take no more than 20 seconds to complete; regular reports will be given to the software development firm about accuracy
- **Interface language translation:** the switch between languages should be seamless in translation; regular reports will be given to the software development firm.
- **Battery-life:** the amount of battery life and CPU usage of the phone must be considered.

PROJECT MANAGEMENT

3.1 FUNCTION POINT

The function point (FP) metric can be used effectively as a means for measuring the functionality delivered by a system. The FP metric can be used to:

1. Estimate the cost or efforts required to design, code, and test the software.

2. Predict the number of errors that will be encountered during testing.

3. Forecast the number of components and/or the number of projected source lines in the implemented system.

Number of external inputs: Each external input that provides distinct application-oriented data to the software is counted separately.

Number of external outputs: Each external output that provides application-oriented information to the user is counted. In this context output refers to reports, screen, and error messages, etc. Individual data items within a report are not counted separately.

Number of external inquiries: An external inquiry is defined as an on-line input that results in the generation of some immediate software response in the form of an on-line output. Each distinct inquiry is counted.

Number of internal logical files: Each internal logical file (i.e., a logical grouping of data that may be one part of a large database or a separate file) is counted.

Number of external interface files: All external interfaces (e.g., data files on storage media) that are used to transmit information to another system are counted.

Measurement Parameter	**Count**	**Simple**	**Average**	**Complex**	**Total**
External inputs	**4**	3	4	6	12
External outputs	**5**	4	5	7	20
External Inquiries	**4**	3	4	6	12
Internal Logical Files	**2**	7	10	15	14
External Interfaces Files	**2**	5	7	10	10
Total count					**68**

To compute Function Point (FP), the following relationship is used:

F.P. = Total Count * [0.65 + 0.01 x Σ (Fi)]

Here, total count is the sum of all FP entries obtained from the above table. And the **Fi (i = 1 to 14)** are **"complexity adjustment values"** based on responses to the following questions:

1	Does the system require reliable backup and recovery?	5
2	Are specialised data communications required to transfer information to or from the application?	0
3	Are there distributed processing functions?	1
4	Is performance critical?	4
5	Will the system run in an existing, heavily utilized operational environment?	2
6	Does the system require online data entry?	5
7	Does the online data entry require the input transaction to be built over multiple screens or operations?	4
8	Are the ILFs updated online?	3
9	Are the inputs, outputs, files, or inquiries complex?	3
10	Is the internal processing complex?	3
11	Is the code designed to be reusable?	2
12	Are conversion and installation included in the design?	1
13	Is the system designed for multiple installations in different organisations?	0
14	Is the application designed to facilitate change and ease of use by the user?	4

VALUE ADJUSTMENT FACTOR Σ (Fi) = 37

Function Point (FP) = Total Count * [0.65 + 0.01 * Σ(fi)]

= 68 * [0.65 + 0.01 * 37]

= 68 * [0.65 + 0.37]

= 68 * 1.02

= 69.36

3.2 EFFORT ESTIMATION

Effort Estimation is a method for estimating the duration of software engineering projects. It is best suited to producing initial estimates for the length of a job, based on known time duration for preparing a specification.

For our system, we calculate Effort (E) in person months using the following formula:

Effort (E) = -37 + (0.96 * FP)

As calculated earlier, the Function Point (FP) of our project = 69.36

Hence,

Effort = -37 + (0.96 *69.36)

= -37+66.586

= 29.586

40% of Effort

= 40 *29.586/ 100

= 11.834

Since, Number of Members = 3

Therefore,

= 11.834/3

= 3.945

≈ **4 months per person (approximately)**

3.3 RISK TABLE

Risk is a probabilistic event i.e.; it may occur or may not occur. We frequently have an optimistic tendency to simply not see risk or wish they will not occur, which later leads us to trouble. Hence it is advisable to identify the critical areas, assess their probabilities, estimate the impact, and plan the contingency plan. The first step is Risk identification. Next each risk is analyzed to determine the likelihood that it will occur and the damage it will do if it occurs. Risks are then ranked by probability and impact. Finally, a plan is developed to manage those risks with high probability, moderate as well as low impact.

These risk analysis activities assist us in developing a strategy for dealing risks. An effective strategy is the RMMM plan.

It Includes:

- Risk identification
- Risk monitoring
- Risk management and contingency plan

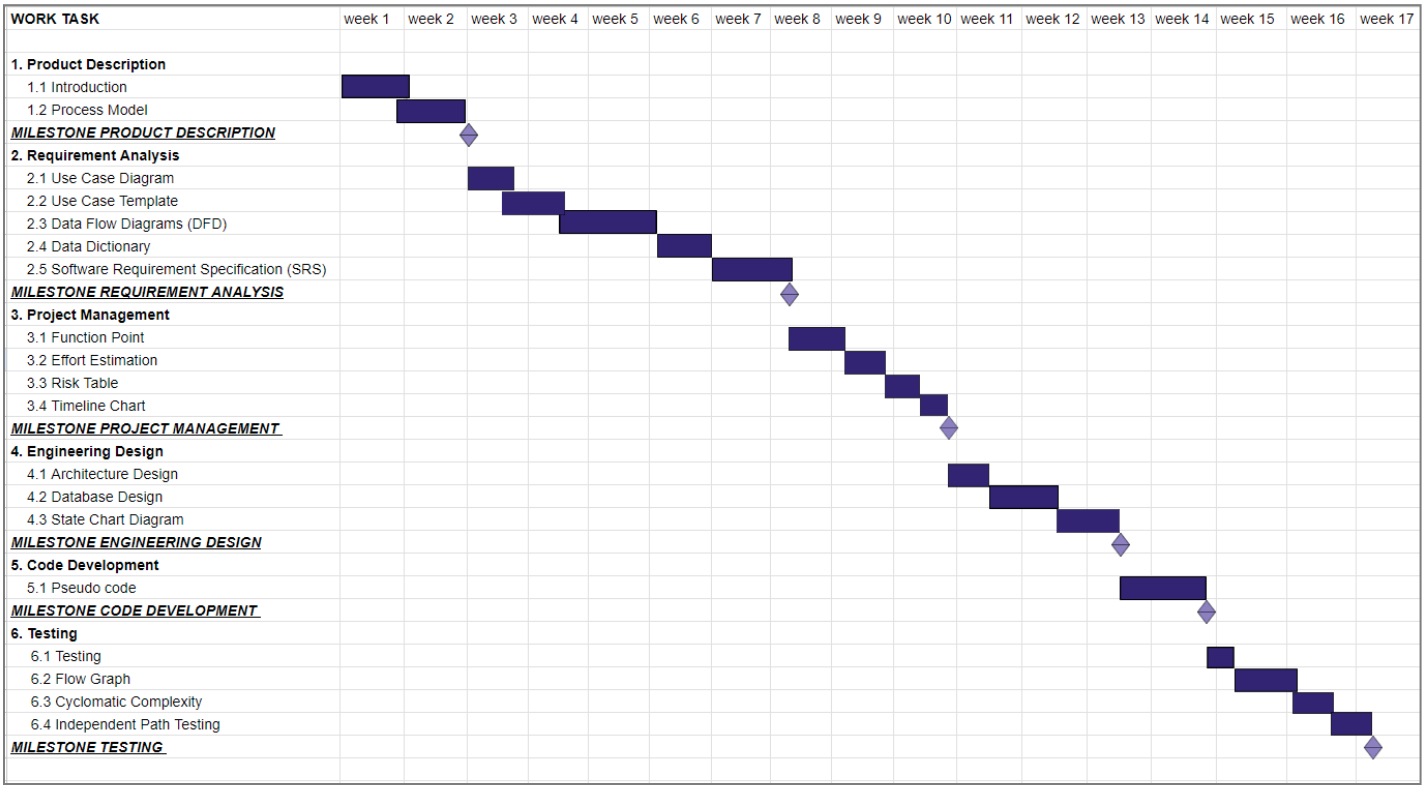
Impact Values:

- Negligible (4)
- Marginal (3)
- Critical (2)
- Catastrophic (1)

RISK	CATEGORY	PROBABILITY	IMPACT	RMMM
Low user adoption and engagement	Market risk	60%	1	A. Include more gamification elements and social features, improve UI/UX B. Conduct market research and advertise C. Gain government support and grants D. Train CSPs to handle complaints and gain feedbacks
Limited availability of content in some languages	Content-related	60%	2	A. Hire experienced language experts to design immersive courses
Technical issues with voice recognition	Development environment risk	40%	3	A. Provide regular feedback to software development firm
Inaccurate translations	Development environment risk	40%	3	A. Provide regular feedback to software development firm B. Keep options open for other firms if issues persist
Not enough funding available	Cost risk	30%	2	A. Approach Government for grants B. Hire competent managers
Loss of database	Quality risk	20%	1	A. Backup the database B. Closely monitor the database and backup copies
Low app performance on older devices	Technical risk	20%	3	A. Optimizing app for better performance on older devices
Size of product	Product size risk	15%	4	A. Upper bound of storage space is specified before-hand

3.4 TIMELINE CHART

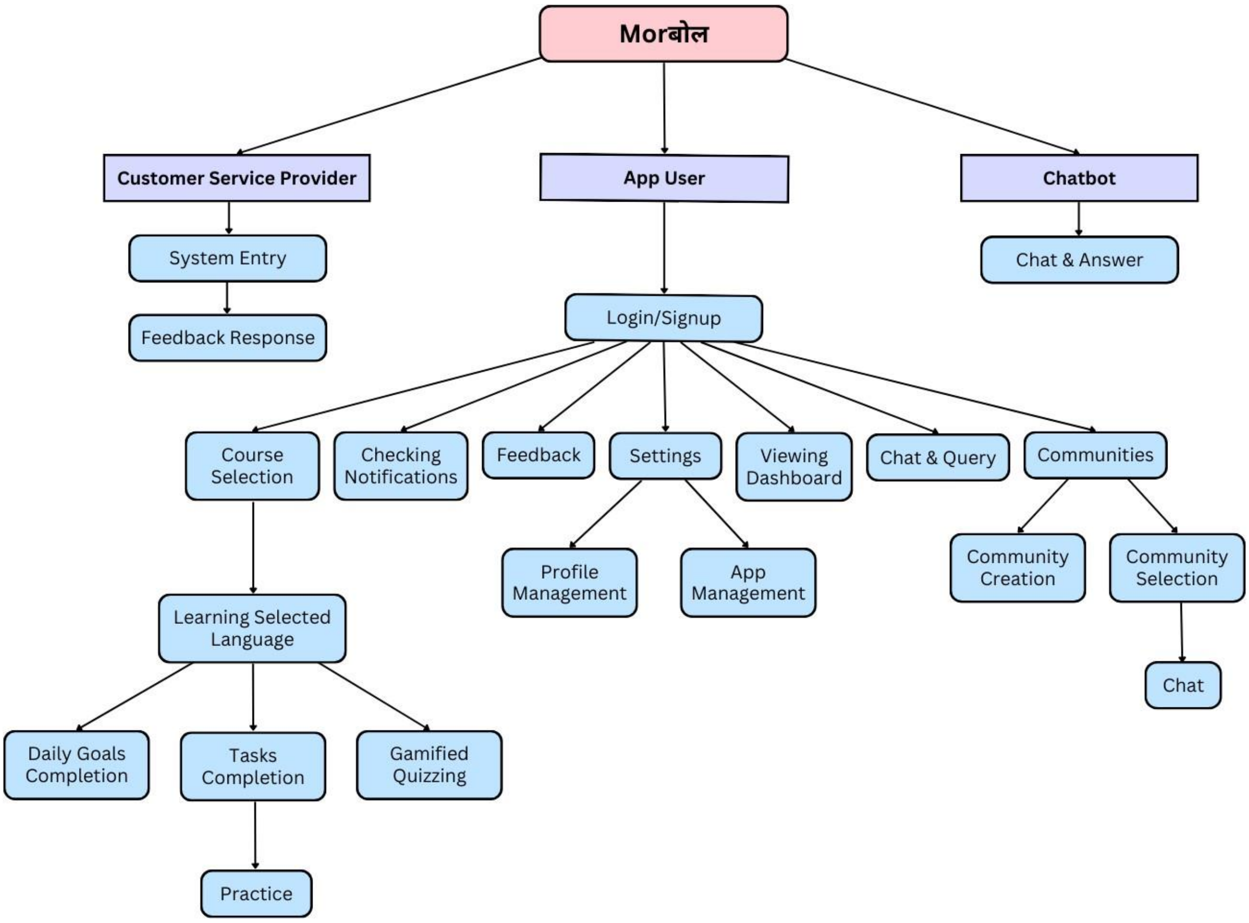
- All project tasks are listed in the far-left column.
- The next few columns may list the following for each task: projected start date, projected stop date, projected duration, actual start date, actual stop date, actual duration, task inter-dependencies (i.e., predecessors).
- The length of a horizontal bar on the calendar indicates the duration of the task the multiple bars that occur at the same time interval on the calendar, this implies task concurrency.
- A diamond in the calendar area of a specific task indicates that the task is a milestone; milestone has a time duration of zero.



ENGINEERING DESIGN

4.1 ARCHITECTURE DESIGN

The software architecture of a program or computing system is a depiction of the system that aids in the understanding of how the system will behave. Software architecture serves as the blueprint for both the system and the project developing it, defining the work assignments that must be carried out by design and implementation teams. The architecture is the primary carrier of system qualities such as performance, modifiability, and security, none of which can be achieved without a unifying architectural vision. By building effective architecture, you can identify design risks and mitigate them early in the development process.



4.2 DATABASE DESIGN

ENTITY RELATIONSHIP DIAGRAM

An entity relationship diagram (ERD), also known as an entity relationship model, is a graphical representation of an information system that depicts the relationships among people, objects, places, concepts, or events within that system. Entity relationship diagrams provide a visual starting point for database design that can also be used to help determine information system requirements throughout an organization.

There are three basic components of an entity relationship diagram:

Entities, which are objects or concepts that can have data stored about them. These are represented by rectangle.

Attributes, which are properties or characteristics of entities. An ERD attribute can be denoted as a primary key, which identifies a unique attribute, or a foreign key, which can be assigned to multiple attributes. These are shown in oval shaped boxes which are attached to their corresponding entity.

Relationships between and among those entities. These are shown in diamond shapes. The entities participating in that relation, their participation constraints are also shown.

SCHEMA

The schema of a database system is its structure described in a formal language supported by the database management system (DBMS). The term "schema" refers to the organization of data as a blueprint of how the database is constructed (divided into database tables in the case of relational database).

1. Users Database:

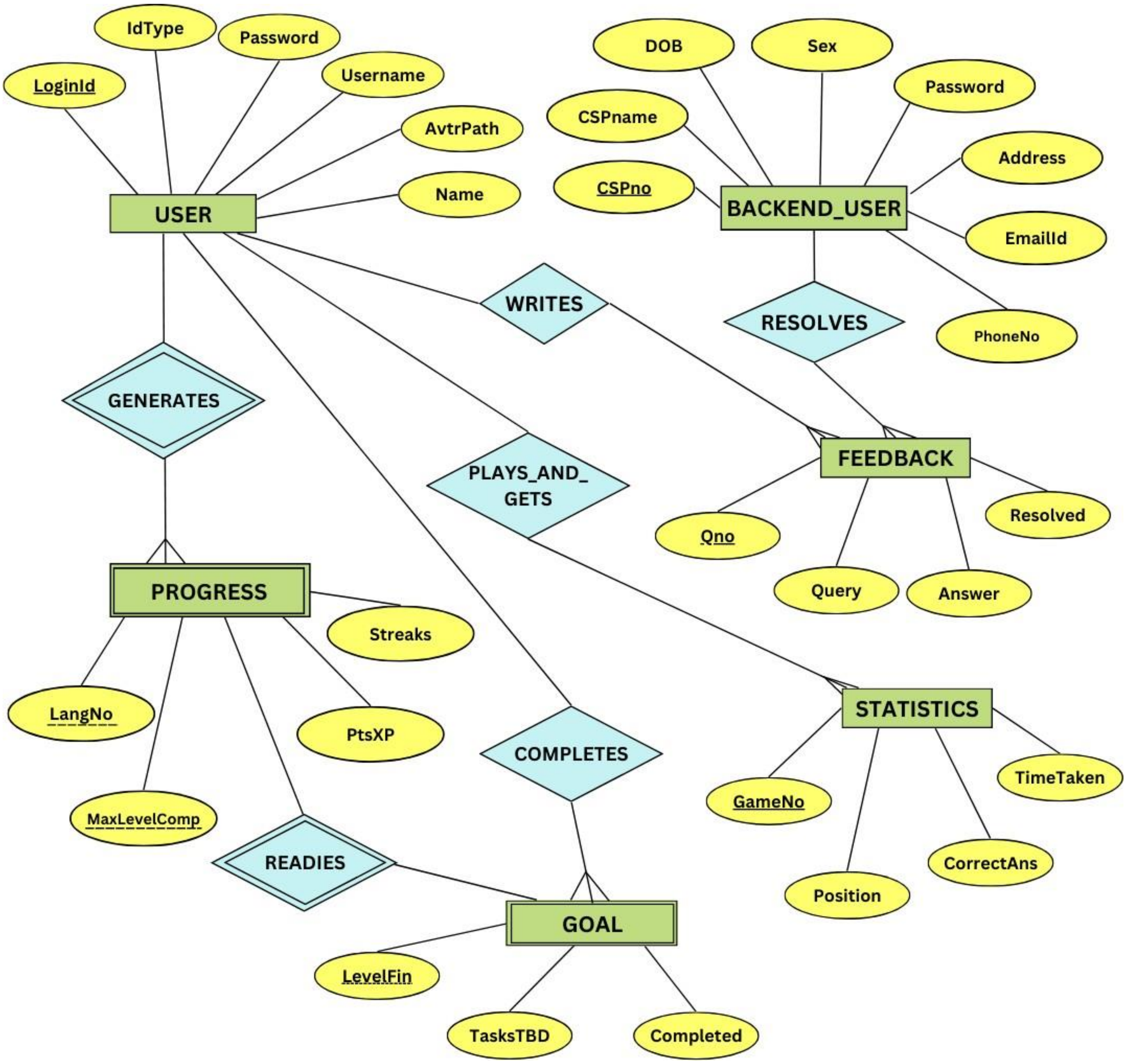


Table name: User

Schema:

Primary Key: LoginId

Description: This table holds the user details.

S.No.	Field Name	Data Type	Constraints	Description
1.	LoginId	VARCHAR(50)	PRIMARY KEY	This login id is used by the user to log into the app
2.	IdType	VARCHAR(10)	NOT NULL	The type of login id is specified
3.	Password	VARCHAR(20)	—	Password is set by the user for Morbol id
4.	Username	VARCHAR(15)	UNIQUE	Username is set by the user
5.	Name	VARCHAR(25)	NOT NULL	User provides their name
6.	AvtrPath	VARCHAR(30)	—	User may choose the avatar he wants to use

Sample table:

LoginId	**IdType**	**Password**	**Username**	**Name**	**AvtrPath**
umesh@morbol.in	Morbol	xy456t_23#	@umesh421	Umesh Sharma	C:\mb\img\avt5.png
tanisha@gmail.com	Google	NULL	@34taniiii	Tanisha Verma	C:\mb\img\avt9.png
eesharoy@yahoo.com	Facebook	NULL	@xyz645ee	Esha Roy	C:\mb\img\avt6.png

Table name: Backend User

Schema:

Primary Key: CSPno

Description: This table stores the details of the customer service providers.

S.No.	**Field Name**	**Data Type**	**Constraints**	**Description**
1.	CSPno	INT(4)	PRIMARY KEY	Backend user enters this unique number to log into the system
2.	CSPname	VARCHAR(25)	NOT NULL	Name of the CSP
3.	DOB	DATE	NOT NULL	DOB of the CSP
4.	Sex	CHAR(1)	NOT NULL	Gender of the CSP
5.	PhoneNo	BIGINT(10)	NOT NULL	Phone number of the CSP
6.	EmailId	VARCHAR(60)	—	Email id of the CSP
7.	Address	VARCHAR(100)	NOT NULL	Residential address of the CSP
8.	Password	VARCHAR(20)	NOT NULL	Password set by the CSP

Sample table:

CSPno	**CSPname**	**DOB**	**Sex**	**PhoneNo**	**EmailId**	**Address**	**Password**
4135	Anika Mishra	1978-09-12	F	8745126398	aniiimike@gmail.com	34, B block, Vasant Kunj, New Delhi-110070	trip4u&#
5612	Sumit Nadar	1982-02-14	M	7845169352	highsum22@gmail.com	122, Krishna Apartments, Civil Lines, New Delhi-110054	t4me&u!!

Table name: Feedback

Schema:

Primary Key: Qno

Description: This table stores the feedbacks and queries received from users.

S.No.	Field Name	Data Type	Constraints	Description
1.	Qno	CHAR(12)	PRIMARY KEY	Unique query number assigned to each query received
2.	Username	VARCHAR(25)	FOREIGN KEY	The user provides their username
3.	Query	VARCHAR(150)	NOT NULL	The query submitted by the user
4.	CSPno	INT(4)	FOREIGN KEY	CSP who answers the query
5.	Answer	VARCHAR(200)	—	The answer provided by the CSP to the user's query
6.	Resolved	BOOLEAN	NOT NULL	Tracks the status of resolution of issue which is set to true by CSP if resolved

Sample table:

Qno	Username	Query	CSPno	Answer	Resolved
220814 eng045	@34taniiii	I'm unable to retrieve my community chats	4135	Thank you for reaching out, the backend issue has been resolved now	TRUE
230105 hin033	@kalifor57	There are grammatical mistakes visible in the app	NULL	NULL	FALSE

Table name: Progress

Schema:

Primary Key: LangNo+MaxLevelComp+Username (Compound Key)

Description: This table stores the progress of each user in different language courses.

S.No.	Field Name	Data Type	Constraints	Description
1.	LangNo	INT(3)	PRIMARY KEY	A unique number assigned to each language course
2.	MaxLevelComp	INT(2)	PRIMARY KEY	Maximum number of levels completed by the user
3.	Username	VARCHAR(15)	PRIMARY KEY (FOREIGN KEY)	Unique Username that is set by the user
4.	PtsXP	INT(4)	NOT NULL	Points gained by the user
5.	Streaks	INT(3)	NOT NULL	Daily completion of lessons is Streak

Sample table:

LangNo	MaxLevelComp	Username	PtsXP	Streaks
202	15	@Kristee455	40	4
203	20	@Kristee455	100	7
204	25	@sarlav82get	200	3

Table name: Goal

Schema:

Primary Key: LangNo+LevelFin+Username (Composite Key)

Description: This table stores the number of tasks to be completed on completion of a level in a particular course.

S.No.	Field Name	Data Type	Constraints	Description
1.	LevelFin	INT(2)	PRIMARY KEY	Levels completed by the user
2.	LangNo	INT(3)	PRIMARY KEY (FOREIGN KEY)	A unique number assigned to each language course
3.	Username	VARCHAR(15)	PRIMARY KEY (FOREIGN KEY)	Unique username that is set by the user
4.	TasksTBD	INT(1)	NOT NULL	Tasks that need to be done by the user
5.	Completed	BOOLEAN	NOT NULL	Status of the tasks whether completed or pending

Sample table:

LangNo	LevelFin	Username	TasksTBD	Completed
203	13	@himesh**77!	4	FALSE
204	20	@taarikajain^	3	TRUE
205	15	@himesh**77!	2	FALSE

Table name: Statistics

Schema:

Primary Key: GameNo+Username (Composite Key)

Description: This table stores the details of leader board positions achieved in games played by the users along with time taken and correct answers given.

S.No.	**Field Name**	**Data Type**	**Constraints**	**Description**
1.	GameNo	VARCHAR(11)	PRIMARY KEY	Unique number assigned to each game
2.	Username	VARCHAR(15)	PRIMARY KEY (FOREIGN KEY)	Unique username that is set by the user
3.	LangNo	INT(3)	FOREIGN KEY	Unique number associated with the language that the game is associated with
4.	Position	INT(2)	NOT NULL	Position of the user out of 20
5.	CorrectAns	INT(2)	NOT NULL	The number of correct answers given by the user
6.	TimeTaken	TIME	NOT NULL	The time taken to finish

Sample table:

GameNo	**Username**	**LangNo**	**Position**	**CorrectAns**	**TimeTaken**
210423HIN44	@himesh**77!	202	13	10	00:04:36
210423HIN44	@gayathree3	202	4	13	00:06:11
310123PUN66	@yashrajmukh1	205	5	13	00:07:45

2. Communities Database:

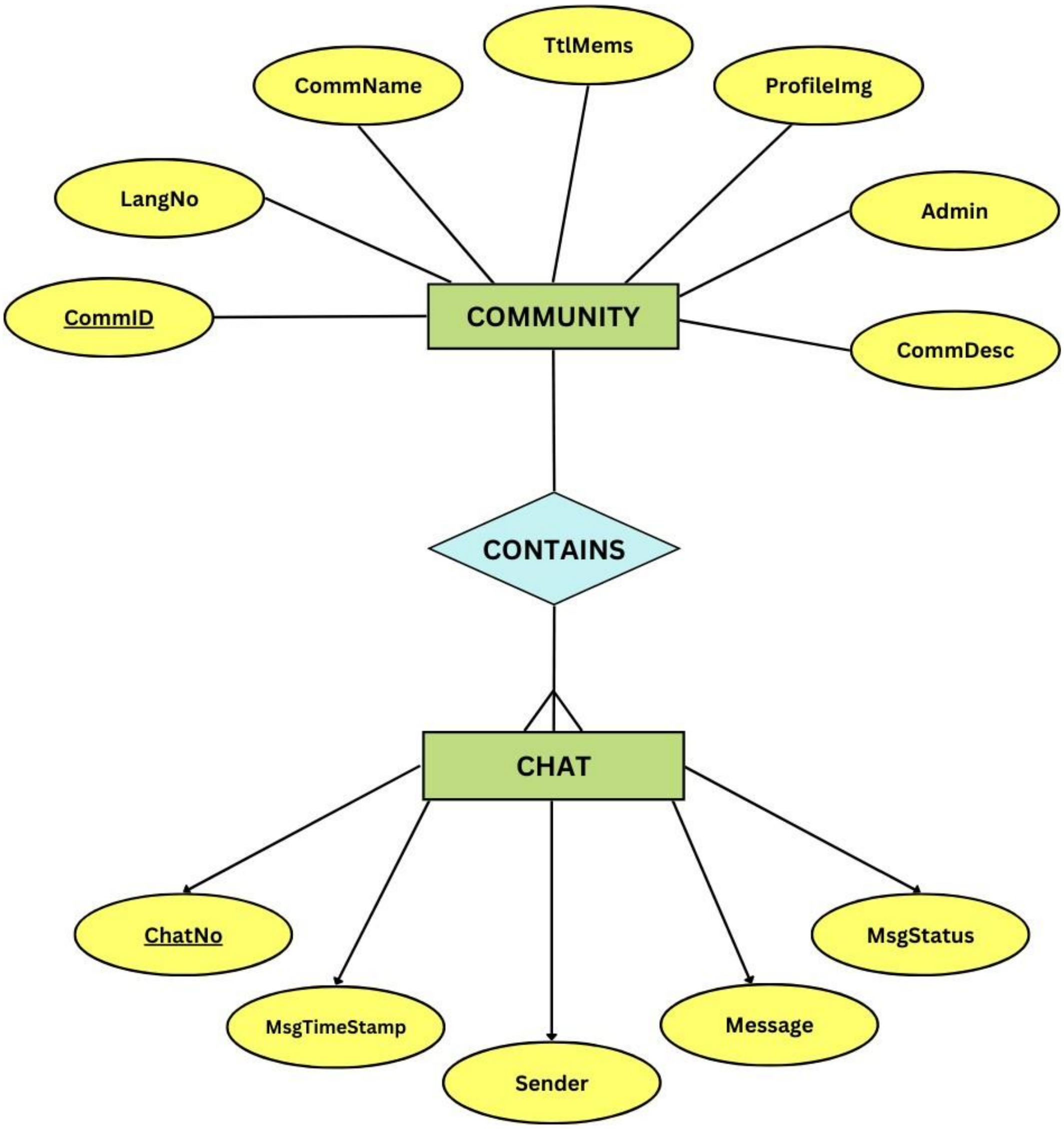


Table Name: Community

Schema:

Primary Key: CommID

Description: This table stores the details of all the existing communities.

S.No.	Field Name	Data Type	Constraints	Description
1.	CommID	INT(5)	PRIMARY KEY	Unique community id given to each community available
2.	LangNo	INT(3)	FOREIGN KEY	Unique Language number given to each language on which the community is based
3.	CommName	VARCHAR(20)	UNIQUE	Name of the community
4.	TtlMems	INT(3)	NOT NULL	Total members in the community
5.	CommDesc	VARCHAR(30)	-	Community description
6.	Admin	VARCHAR(20)	FOREIGN KEY	Username of Admin of the community group who has created it
7.	ProfileImg	VARCHAR(50)	-	Profile Image for the community is set by the admin

Sample table:

CommID	LangNo	CommName	TtlMems	CommDesc	Admin	ProfileImg
10051	203	Tamil Learners	150	Group for people who want to learn Tamil	@lithim22	C:\mb\hin\img\tamil_Icon.png
20032	204	Punjabi Learners	200	Group for people who want to learn Punjabi	@geetgo4	C:\mb\hin\img\punjabi_Icon.png

Table Name: Chat

Schema:

Primary Key: ChatNo

Description: This table stores the chats of a particular community. Each community has its separate chat database.

S.No.	Field Name	Data Type	Constraints	Description
1.	ChatNo	INT(10)	PRIMARY KEY	Unique Chat number assigned to each message sent on the community
2.	CommID	INT(5)	FOREIGN KEY	Unique community id given to each community available
3.	MsgTimeStamp	TIMESTAMP	NOT NULL	Timestamp of the message sent
4.	Sender	VARCHAR(20)	FOREIGN KEY	Username of the sender
5.	Message	VARCHAR(500)	NOT NULL	Message sent in the group
6.	MsgStatus	VARCHAR(10)	NOT NULL	Status of the message

Sample table:

ChatNo	CommID	MsgTime Stamp	Sender No	Sender	Message	MsgStatus
2103230032	10042	2023-03-21 03:14:07	110	@mehak242	Good Morning!	Delivered
0309230213	10042	2023-09-03 02:30:24	078	@sidduu	Kidda?	Sending

3. Interface_Languages Database:

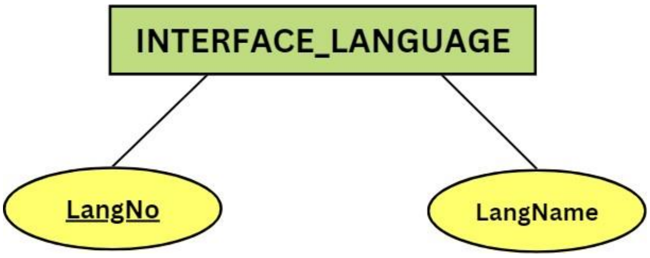


Table Name: Interface_Language

Schema:

Primary Key: LangNo

Description: This table stores the details of the languages which are available to choose from, for the interface of the application.

S.No.	**Field Name**	**Data Type**	**Constraints**	**Description**
1.	LangNo	INT(3)	PRIMARY KEY	Unique Language number present for each language available for the interface of the app
2.	LangName	VARCHAR(15)	NOT NULL	Name of the languages available for interface

Sample table:

LangNo	**LangName**
105	Hindi
108	English
109	Tamil

4. Language_Courses Database:

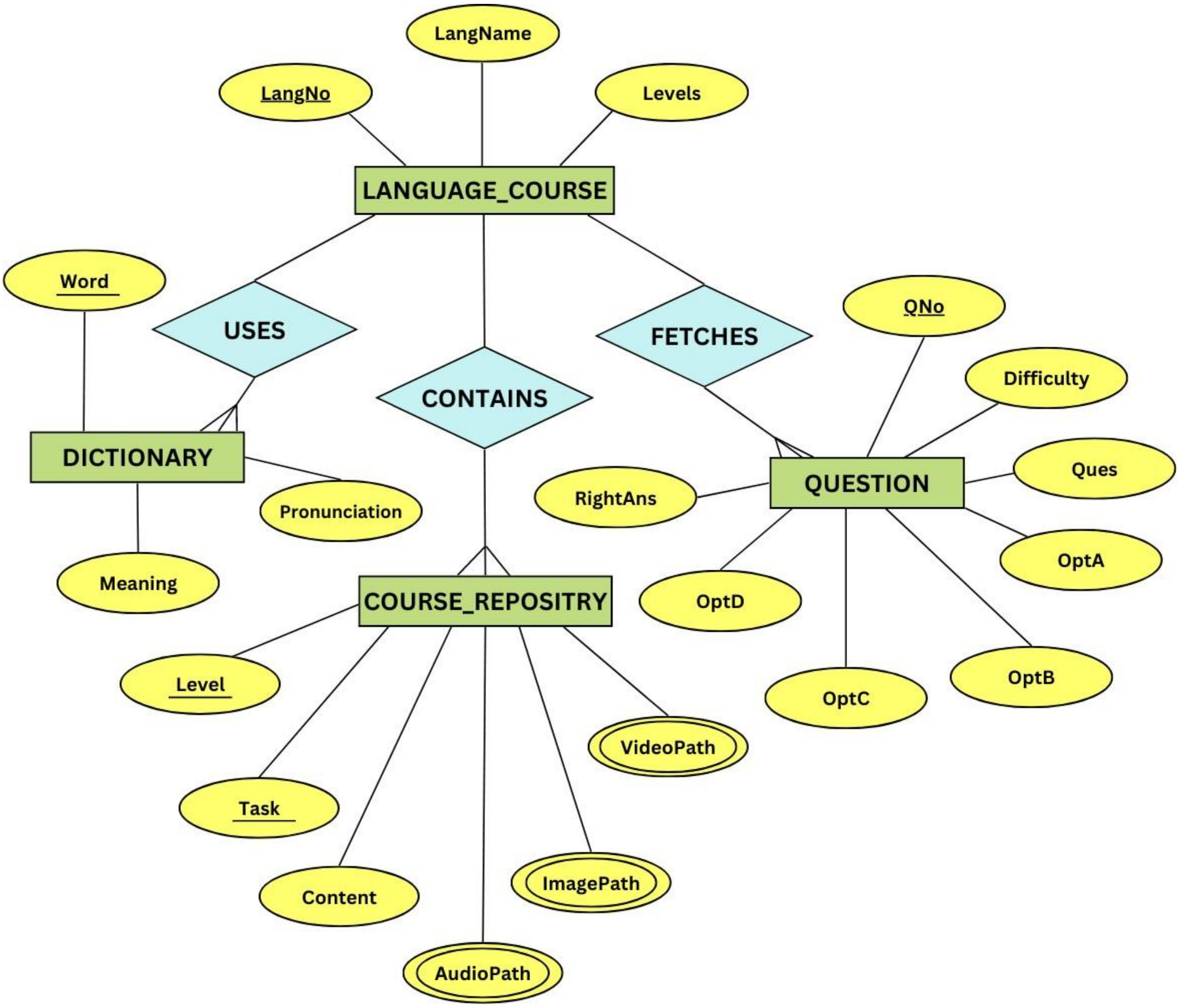


Table Name: Language Course

Schema:

Primary Key: LangNo

Description: This table stores information of the different language courses available.

S.No.	**Field Name**	**Data Type**	**Constraints**	**Description**
1.	LangNo	INT(3)	PRIMARY KEY	A unique number assigned to each language course
2.	LangName	VARCHAR(15)	NOT NULL	The language on which the course is based
3.	Levels	INT(2)	NOT NULL	Number of levels in a particular course

Sample table:

LangNo	**LangName**	**Levels**
202	Hindi	20
203	Tamil	20
204	Telugu	25

Table Name: Course Repository

Schema:

Primary Key: Level+Task (Composite key)

Description: This table stores course content along with paths to audio, video, and image files. It is maintained separately for each available language.

S.No.	**Field Name**	**Data Type**	**Constraints**	**Description**
1.	Level	INT(2)	PRIMARY KEY	The level number in the course
2.	Task	INT(1)	PRIMARY KEY	The task number in the level
3.	LangNo	INT(3)	FOREIGN KEY	A unique number assigned to each language course
4.	Content	VARCHAR(200)	NOT NULL	The textual content present in the task related to the course
5.	AudioPath	VARCHAR(50)	—	Path to any audio file to be included
6.	VideoPath	VARCHAR(50)	—	Path to any video file to be included
7.	ImagePath	VARCHAR(50)	—	Path to any image to be included

Sample table:

Level	**Task**	**LangNo**	**Content**	**AudioPath**	**VideoPath**	**ImagePath**
1	1	202	Listen carefully to the following 2 letters	C:\mb\hin\aud\ad1.mp3 C:\mb\hin\aud\ad2.mp3	NULL	NULL
1	2	202	Read the following words carefully- 1. Python is called 'ajagar' 2. Mango is called 'aam'	C:\mb\hin\aud\snake.mp3 C:\mb\hin\aud\mango.mp3	NULL	C:\mb\hin\img\snake.png C:\mb\hin\aud\mango.png

Table Name: Dictionary

Schema:

Primary Key: Word

Description: This table stores words with their meanings and audio files to cross check pronunciation. The table is stored separately for each language.

S.No.	Field Name	Data Type	Constraints	Description
1.	Word	VARCHAR(20)	PRIMARY KEY	Different words stored for vocal practice under a course
2.	LangNo	INT(3)	FOREIGN KEY	A unique number assigned to each language course
3.	Meaning	VARCHAR(100)	NOT NULL	Meaning of the given word in English
4.	Pronunciation	VARCHAR(50)	NOT NULL	Path to the audio file of pronunciation

Sample table:

Word	LangNo	Meaning	Pronunciation
Namaste	202	Hello	C:\mb\hin\aud\namaste.mp3
Namaskar	202	Hello	C:\mb\hin\aud\namaste.mp3
Suprabhat	202	Good Morning	C:\mb\hin\aud\goodmorning.mp3

Table Name: Question

Schema:

Primary Key: QNo

Description: This table stores several questions of varying difficulty levels for different levels in a particular language course. The table is stored separately for each user.

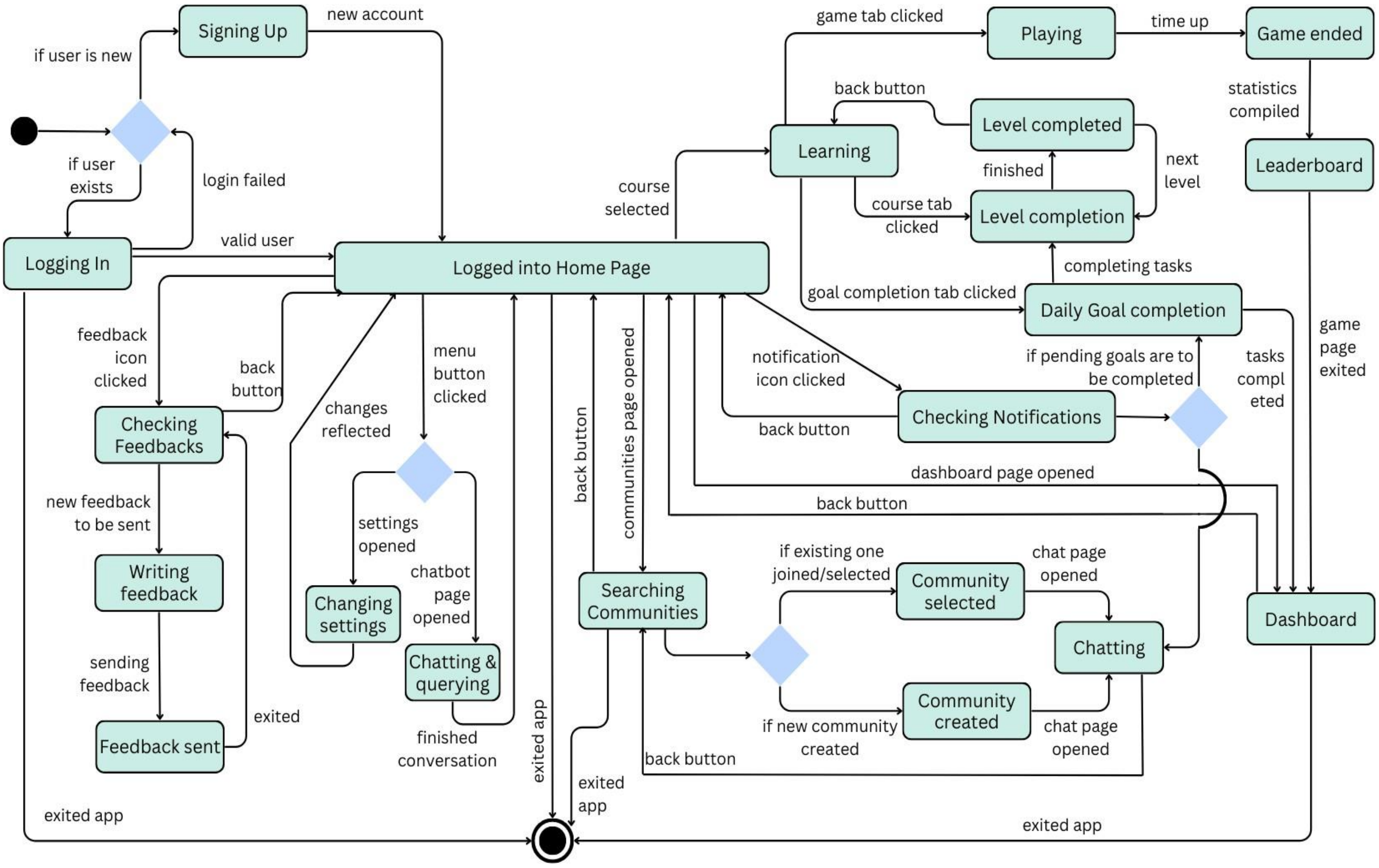
S.No.	Field Name	Data Type	Constraints	Description
1	QNo	STRING(4)	PRIMARY KEY	Question number of each question reflecting level number as well, stored internally
2	LangNo	INT(3)	FOREIGN KEY	A unique number assigned to each language course
3	Difficulty	VARCHAR(10)	NOT NULL	Difficulty level of the question
4	Ques	VARCHAR(80)	NOT NULL	Question to be answered by the user
5	OptA	NVARCHAR(50)	NOT NULL	First option of the question
6	OptB	NVARCHAR(50)	NOT NULL	Second option of the question
7	OptC	NVARCHAR(50)	NOT NULL	Third option of the question
8	OptD	NVARCHAR(50)	NOT NULL	Fourth option of the question
9	RightAns	VARCHAR(1)	NOT NULL	Correct option for given question

Sample table:

QNo	**LangNo**	**Difficulty**	**Ques**	**OptA**	**OptB**	**OptC**	**OptD**	**Right Ans**
0501	203	Easy	Correct pronunciation of ‘GOOD MORNING’ in Tamil ?	காலை வணக்கம்	இனிய இரவு	பிரியா விடை	வணக்கம்	A
0702	203	Medium	Make sentence ‘Hello, how are you ?’ in Tamil	நீ எங்கேஇருக்கிறாய்	அவர் யார்	வணக்கம் எப்படிஇருக்கிறாய்	நான்விழித்திருக்கிறேன்	C

4.3 STATE CHART DIAGRAM

A state chart diagram is used to show the state space of a given class, the event that cause a transition from one state to another, and the actions that result from a state change. It shows the changing behaviour of an object in response to different events. State transition diagrams are made only for the objects that are either very complicated or have a very dynamic behaviour with respect to various events. The state chart for 'MorBol' is as follows:



CODE DEVELOPMENT

5.1 PSEUDOCODE

Login/Signup:

```
1.  Start
2.  Choose AppLanguage from Interface Languages DB
3.  Set LanguageInterface=AppLanguage
4.  Choose Login/Signup
5.  If Login
6.          Choose loginType
7.          If loginType=’Google’ or ‘Facebook’
8.                  Enter loginId
9.                  If loginId exists
10.                         Print ‘Welcome back’+Name
11.                 Else
12.                         Print ‘Invalid Id’
13.                         Go to 7
14.         Else
15.                 Enter loginId
16.                 If loginId exists
17.                         Enter Password
18.                         If Password is correct
19.                                 Print ‘Welcome back’+Name
20.                         Else
21.                                 Print ‘Incorrect Password’
22.                                 Go to 17
23.                 Else
24.                         Print ‘Invalid User’
25.                         Go to 15
26.Else
27.         Choose SignupType
28.         If SignupType=’Google’ or ‘Facebook’
29.                 Enter loginId
30.                 If loginId exists
31.                         Add details to Users DB
32.                 Else
33.                         Print ‘Invalid Id’
34.                         Go to 29
35.         Else
36.                 Enter loginId
```

```
37.             Enter password    //atleast 8 characters
38.             Add details to Users DB
39.         Enter name
40.         Enter username
41.         Choose avatar
42.         Set AvatarPath in Users DB
43.         Print 'Welcome '+Name
44.Stop
```

Settings:

```
1.  Start
2.  Select settings
        1.  Profile
        2.  General
        3.  Helpdesk
3.  If settings=Profile
4.          Select Profileoptions
               i.  Name
               ii. Username
               iii.  Avatar
               iv.  Password
5.          Case i
6.              Change Name
7.          Case ii
8.              Change Username
9.          Case iii
10.             Change Avatar
11.         Case iv
12.             Change Password
13.         Update the Users DB
14.Else if settings=General
15.         Select options
               i.   AppInterfaceLanguage
               ii.  Notifications
               iii. SoundEffects
               iv.  Theme
16.         Case i
17.             Choose language from Interface Languages DB
18.             Set LanguageInterface= language
19.         Case ii
20.             If switch='On'
21.                 Unmute notifications
22.             Else
```

```
23.                 Mute notifications
24.         Case iii
25.             If switch='On'
26.                 Enable sound effects
27.             Else
28.                 Disable sound effects
29.         Case iv
30.             If switch='right'
31.                 Set to Light mode
32.             Else
33.                 Set to Dark mode
34.Else if settings=Helpdesk
35.         Select Help
            i.  FAQs
            ii. T&C
36.         If Help=FAQs
37.             Enter query
38.             If query exists
39.                 Display FAQ
40.             Else
41.                 Redirect to Feedback
42.         Else if Help=T&C
43.             Display T&C page
44.Stop
```

Learning Module:

```
1.  Start
2.  Choose LanguageCourse from language courses DB
3.  Select option:
            a.  LearningModule
            b.  DailyGoals
            c.  QuizzingGame
4.  If option=LearningModule
5.          Select level
6.          If level is unlocked
7.              Fetch tasks from course repository
8.              Complete tasks
9.              Fetch 3 questions from course repository
10.             While questions are still left
11.                 If answer is wrong
12.                     Add an extra question
13.                 Else
14.                     Add 10px to Points
```

```
15.             Unlock next level
16.         Else
17.             Fetch 10 RevisionQuestions from questions repository
18.             If correctAnswers=10
19.                 Add 100px to Points
20.                 Unlock level
21.                 Go to 7
22.             Else
23.                 Print ‘Sorry you still need to master the previous levels’
24.Else if option=DailyGoals
25.         Fetch PendingGoals from goals record
26.         If User not Active
27.             Send Streaks to Notifications module
28.         Answer 5 questions
29.             If answer is correct
30.                 Add 15px to Points
31.         Go to 7
32.Else if option=QuizzingGame
33.         Create room with 15 users online
34.         If users<15
35.             Add (15-users) Bots
36.         Start game
37.         Fetch 15 questions from questions repository
38.             If answer is correct
39.                 Add 10pts to GamePoints
40.             Else
41.                 Subtract 5pts from GamePoints
42.             Update statistics record
43.             Rank users based max(pts) and min(time) on the Leaderboard
44.         End game when all users have answered all questions
45.         Display Leaderboard on full screen
46.         Send User’s stats to Dashboard
47.Update Progress record
48.If PendingGoals completed
49.         Streaks+=1
50.         Update goals record
51.Stop
```

Communities:

```
1.  Start
2.  Display communities that the user has joined
3.  If user clicks on a community
4.          View chat page
5.          Chat or post a message
6.          Access more options from the icon (⋮) as chooseOption:
                i.    Community info
                ii.   Select messages
                iii.  Star messages
                iv.   Mute notifications
                v.    Exit community
7.          Case i
8.                  Display profile, description, and members
9.          Case ii
10.                 Choose msgOpt
                        1. Copy
                        2. Delete message
                        3. Star message
11.                 If msgOpt=Copy
12.                         The message is copied to the clipboard
13.                         Print ‘Copied to clipboard’
14.                 Else If msgOpt=Delete message
15.                         The message is deleted
16.                         Print ‘Message Deleted’
17.                 Else If msgOpt=Star message
18.                         The message is marked as a star message
19.         Case iii
20.                 Displays the messages marked with a star
21.         Case iv
22.                 The community chat notifications are muted
23.                 Prompts ‘Unmute notifications | Cancel’
24.                 If selected ‘Unmute notifications’
25.                         Notifications unmuted
26.                 Else If selected ‘Cancel’
27.                         Go to 4
28.         Case v
29.                 Prompts ‘Are you sure you want to exit the community?’
30.                 Prompts ‘Yes | Cancel’
31.                 If selected option is Yes
32.                         Exits the community
33. Prompts ‘You can’t message, you are no longer a part of this community’
```

```
34.                 Else If selected option is cancel
35.                         Go to 4
36.If user wants to create a community
37.         Enter the name of the community
38.         Select the language of the community chat for learning purpose
39.         Set group profile
40.         Enter group description
41.         Update Communities DB
42.If user wants to choose a community
43.         Select a community from the existing communities database
44.         Join the community
45.Exit the communities section
46.Stop
```

Chat & Query:

```
1.  Start
2.  User accesses the chatbot clicking on its icon
3.  Initialise the chatbot page and establish a connection to the Chatbot API
4.  Prompt the user to enter a message or query
5.  Send the user’s message to Chatbot API
6.  Receive the response from the Chatbot API
7.  Display the response to the user in the application
8.  If user wants to continue chatting
9.          Go to 4
10.Else
11.         Close the connection to the Chatbot API
12.Stop
```

System Entry:

```
1.  Start
2.  Enter CSPno
3.  If CSPno is valid
4.          Enter password
5.          If the entered password is correct
6.                  Access granted to Feedback module
7.          Else
8.                  Print ‘Incorrect Password’
9.                  Access denied
10.                 Go to 4
11.Else
12.         Print ‘Invalid CSPno’
13.         Go to 2
14.Stop
```

Feedback:

```
1.  Start
2.  Display all feedbacks/complaint sent by app user
3.  Select option:
            a. View feedbacks
            b. Write a feedback
4.  If option= View feedback
5.      Display feedback on full screen
6.      If issue not yet resolved
7.          Print ‘pending’
8.      Else
9.          Display CSP’s response
10.         Print ‘resolved’
11. Else if option= Write a feedback
12.     Write the feedback/complaint
13.     Submit the feedback
14.     If server down
15. Print “Sorry the server is down. You can wait for a while or report the issue or
    write an email to morbol@gmail.com’’
16.         If user selects Report option
17.             Report the issue
18.         Else if user wants to write an email
19.             Copy the email id to write feedback using a mailing app
20.         Else
21.             Go to 1
22.     A response is provided by a CSP
23.     If resolved
24.         CSP sets status to ‘Resolved’
25. Stop
```

Notifications:

```
1.  Start
2.  If current time=’00:00’ or reminder time
3.      Get streaks from language course module
4.      Display new notification->’Your current streak is ‘+ streaks + ’, complete
    today’s goal to maintain your streak’
5.      If user clicks on notification
6.          Redirect to daily goals page
7.  If app updates available
8.      Display new notification->’App update available, click to install new
    update’
9.      If user clicks on notification
10.         Redirect to app store
```

```
11.If new messages in a community
12.Display new notification->'You have a new message from '+community name
13.     If user clicks on notification
14.          Redirect to the community chat page
15.If user wants to manage notifications
16.     Select notification option by clicking on the icon (⋮)
            i.   Practice reminder //for daily goal completion
            ii.  Weekly progress
            iii. New app updates
17.     If notification option=Practice Reminder
18.          Send practice reminders options
                 a. Via phone notifications
                 b. Via email
19.          If phone notifications enabled
20.               Get practice reminders through notifications
21.          If email reminders enabled
22.               Get practice reminders through registered email account
23.          If smart scheduling enabled
24.               Automatically send scheduled reminders
25.          Else
26.               Set reminder time
27.     Else if notification option=Weekly progress
28.          If switch enabled
29.               Get progress from Language course module once a week
30.               Send the weekly progress to registered Gmail account
31.     Else if notification option=App updates
32.          If switch enabled
33.               Send app updates via notification
34.If user swipes a notification to left
35.     Delete the notification
36.Stop
```

Dashboard:

```
1.  Start
2.  Select View Dashboard icon
3.  Display name, username and avatar
4.  From progress database fetch
        a.  Progress
        b.  Points
        c.  Streak
5.  Update progress
6.  Display Progress bar
7.  Add Points to Total Xp
8.  Display Total Xp
9.  If time='0:00' and Streak=0
10.     Set Current Streak=0
11.Else
12.     Add Streak to Current Streak
13.Display Current Streak
14.Fetch Stats from Leaderboard
15.Display user's position, pts and FinTime
16.Stop
```

TESTING PHASE

6.1 TESTING

BLACK-BOX TESTING

Black-box testing, also called *behavioural testing,* focuses on the functional requirements of the software. It enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program.
Black-box testing attempts to find errors in the following categories:

1. Incorrect or missing functions
2. Interface errors
3. Errors in data structures or external database access
4. Behaviour or performance errors
5. Initialization and termination errors

WHITE-BOX TESTING

White-box testing, also called *glass-box testing,* is a test-case design philosophy that uses the control structure described as part of component- level design to derive test cases. Using white-box testing we can derive test cases that:

1. Guarantee that all independent paths within a module have been exercised at least once
2. Exercise all logical decisions on their true and false sides
3. Execute all loops at their boundaries
4. Exercise internal data structures to ensure their validity

BASIC PATH TESTING

Basic-Path testing is a white-box testing technique. The basic path method enables the test-case designer to derive a logical complexity measure of a procedural design and use this measure as a guide for defining a basic set of execution paths.
Basic-path testing of the following modules are shown below:

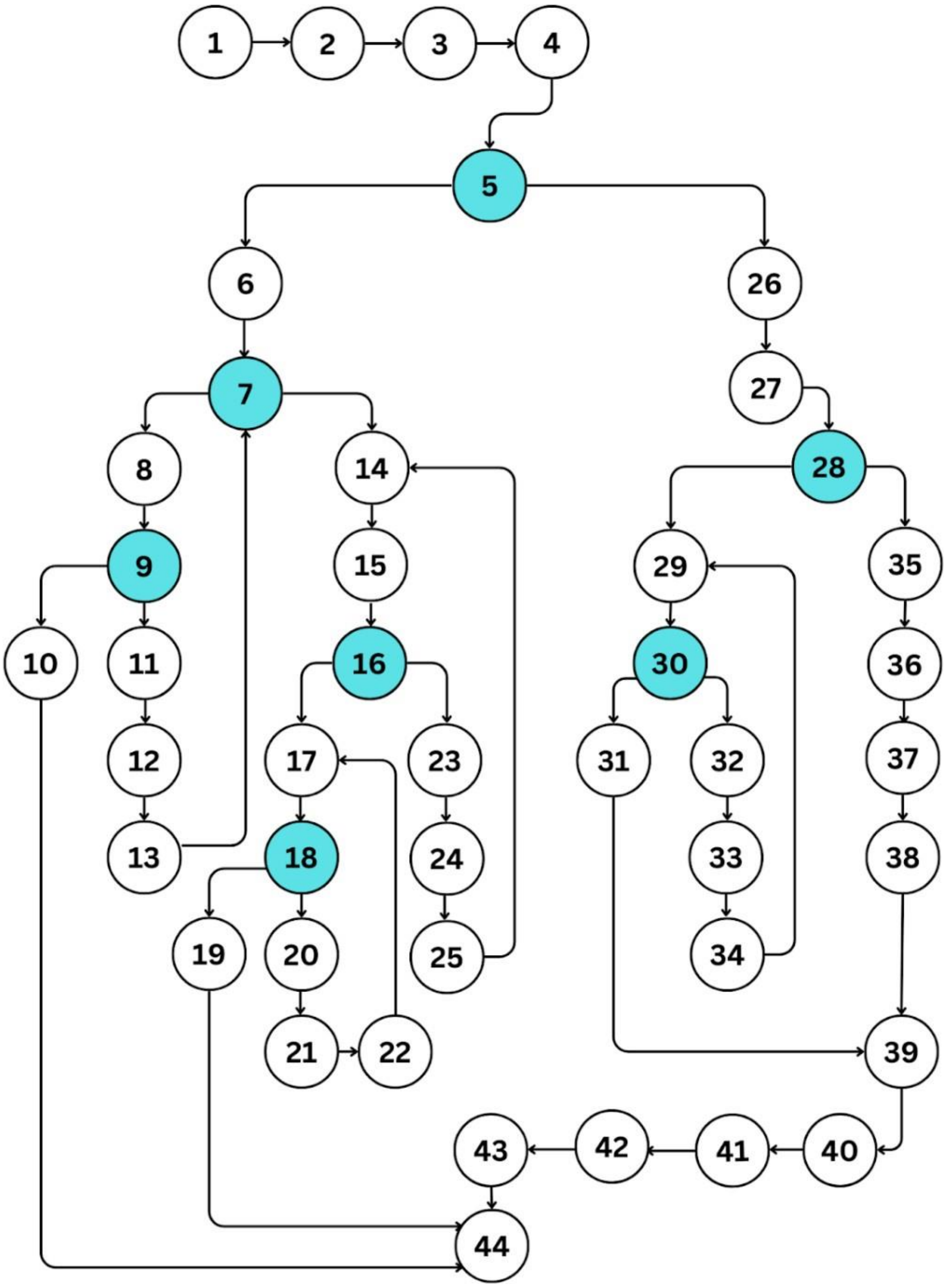
1. Login
2. Chat & Query
3. System Entry

6.2 FLOW GRAPH

In a control flow graph each node in the graph represents a basic block, i.e., a straight-line piece of code without any jumps or jump targets; jump targets start a block, and jumps end a block. Directed edges are used to represent jumps in the control flow. There are, in most presentations, two specially designated blocks: the entry block, through which control enters the flow graph, and the exit block, through which all control flow leaves.

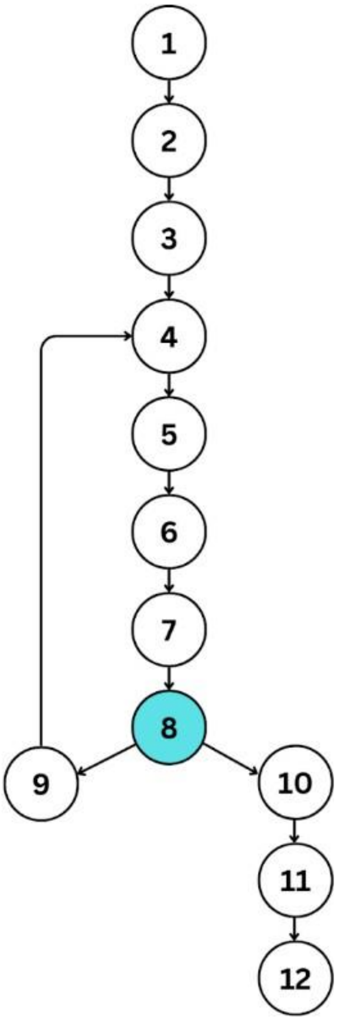
<u>Flow graph of Login/Signup module:</u>

The nodes coloured in blue are the predicate nodes.



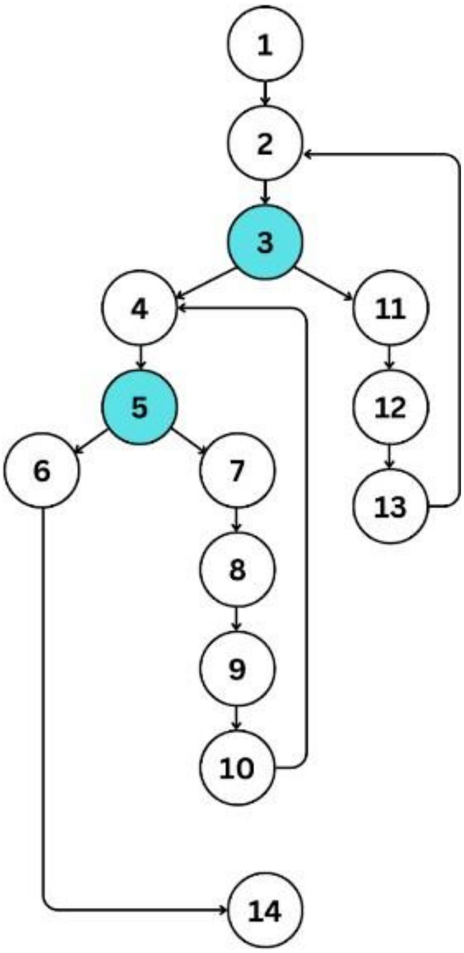
<u>Flow graph of Chat & Query module:</u>

The nodes coloured in blue are the predicate nodes.



Flow graph of System Entry module:

The nodes coloured in blue are the predicate nodes.



6.3 CYCLOMATIC COMPLEXITY

Cyclomatic complexity is computed using the control flow graph of the program, the nodes of the graph correspond to individual groups of commands of a program and a directed edge connects two nodes if the second command might be executed immediately after the first command. Cyclomatic complexity may also be applied to individual functions, modules, methods or classes within a program.
One testing strategy, called Basis Path Testing by McCabe, who first proposed it, is to test each linearly independent path through the program; in this case, the number of test cases will equal the Cyclomatic complexity of the program.

The formula of Cyclomatic Complexity is given by:
McCabe's Cyclomatic complexity V (G) for the flow graph is defined by:

V (G) = E – N+ 2P

E is the number of edges in flow graph
N is the number of nodes in flow graph
P is connected components in flow graph

<u>In case of Login module:</u>

E (Number of edges) = 50
N (Number of nodes) = 44
P (Connected Components) = 1
Therefore, V (G) = E – N + 2P
= 50-44+(2*1)
= 6+2
= 8

Complexity measure is defined in terms of independent paths. **Independent Path** is any path through the program that introduces at least one new set of processing statements or a new condition.

Since V(G) = 8,
There will be 8 independent paths for the above flow graph.
Path 1: 1-2-3-4-5-6-7-8-9-10-44
Path 2: 1-2-3-4-5-6-7-8-9-11-12-13-7...
Path 3: 1-2-3-4-5-6-7-14-15-16-17-18-19-44
Path 4: 1-2-3-4-5-6-7-14-15-16-17-18-20-21-22-17…
Path 5: 1-2-3-4-5-6-7-14-15-16-23-24-25-15…
Path 6: 1-2-3-4-5-26-27-28-29-30-31-39-40-41-42-43-44
Path 7: 1-2-3-4-5-26-27-28-29-30-32-33-34-29…
Path 8: 1-2-3-4-5-26-27-28-35-36-37-38-39-40-41-42-43-44

In case of Chat & Query module:

E (Number of edges) = 12
N (Number of nodes) = 12
P (Connected Components) = 1
Therefore, V (G) = E – N + 2P
= 12-12+(2*1)
= 0+2
= 2

Complexity measure is defined in terms of independent paths. **Independent Path** is any path through the program that introduces at least one new set of processing statements or a new condition.

Since V(G) = 2,
There will be 2 independent paths for the above flow graph.
Path 1: 1-2-3-4-5-6-7-8-9-4…
Path 2: 1-2-3-4-5-6-7-8-10-11-12

In case of System Entry module:

E (Number of edges) = 15
N (Number of nodes) = 14
P (Connected Components) = 1
Therefore, V (G) = E – N + 2P
= 15-14+(2*1)
= 1+2
= 3

Complexity measure is defined in terms of independent paths. **Independent Path** is any path through the program that introduces at least one new set of processing statements or a new condition.

Since V(G) = 3,
There will be 3 independent paths for the above flow graph.
Path 1: 1-2-3-4-5-6-14
Path 2: 1-2-3-4-5-7-8-9-10-4…
Path 3: 1-2-3-11-12-13-2…

6.4 INDEPENDENT PATH TESTING

In case of Login/Signup module:

PATH	CONDITION	TEST CASES	STATEMENT	STATUS
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 44	If user wants to login through an existing google or Facebook account	Yes	Welcome Back user	Login successful
1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 7, …	If user enters incorrect google or facebook id	No	Invalid id	Enter valid ld
1, 2, 3, 4, 5, 6, 7, 14, 15, 16, 17, 18, 19, 44	If user logs in with an existing MorBol id having a valid password	Yes	Welcome back User	Login successful
1, 2, 3, 4, 5, 6, 7, 14, 15, 16, 17, 18, 20, 21, 22, 17, …	If user enters the wrong password for the MorBol id	No	Incorrect password	Enter correct password
1, 2, 3, 4, 5, 6, 7, 14, 15, 16, 23, 24, 25, 15, …	If user does not exist in the system	No	Invalid user	Enter valid id
1, 2, 3, 4, 5, 26, 27, 28, 29, 30, 31, 39, 40, 41, 42, 43, 44	If user signs up with a valid google or facebook id	Yes	Welcome User	Signup successful
1, 2, 3, 4, 5, 26, 27, 28, 29, 30, 32, 33, 34, 29, …	If user tries to sign up with an invalid google or facebook id	No	Invalid Id	Enter valid id
1, 2, 3, 4, 5, 26, 27, 28, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44	If user wants to sign up with a MorBol id	Yes	Welcome User	Signup successful

In case of Chat & Query module:

PATH	CONDITION	TEST CASES	STATEMENT	STATUS
1, 2, 3, 4, 5, 6, 7, 8, 9, 4…	If user wants to continue chatting with the chatbot	Yes	Enter message or query	Allowed access to chat
1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12	If user wants to end the conversation	No	End conversation	Exit chat interface

In case of System Entry module:

PATH	CONDITION	TEST CASES	STATEMENT	STATUS
1, 2, 3, 4, 5, 6, 14	If CSPno is valid and password is correct	Yes	Access granted to feedback module	Successful system entry
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 4, …	If CSPno is valid and password is incorrect	No	Incorrect password	Enter correct password
1, 2, 3, 11, 12, 13, 2, …	If CSPno is invalid	No	Invalid CSPno	Enter valid CSPno

FUTURE SCOPE

For the present time this project holds its value for college level only but afterwards its scope can be expanded and geared up to national level to be accessible to people from various states.

We can add more Indian languages and even some dialects to be learnt in our app to cater to a larger audience.

We can also include support for more interface languages with subsequent improvements in AI translation tools.

Accessibility features can be added for people with visibility issues, hearing, and vocal impairment along with haptic feedback.

More interesting and engaging games can be formulated and added to pique the user’s interest in learning.

We can introduce different levels for each language, such as basic, intermediate, advanced, and expert. This will help learners learn at their own pace and will make the learning experience more personalized.

According to the feedback received from app users, we will keep on updating the software and enhancing user experience to create and maintain a satisfied user base with enthusiastic learners.

REFERENCES

[1] Software Engineering, K K Aggarwal and Yogesh Singh, New Age International Publishers, 3rd Edition (2008), Reprint: 2019

[2] Software Engineering- A Practitioner’s Approach, Roger S. Pressman and Bruce R. Maxim, McGraw Hill Publications, 8th Edition (2019)

[3] Sample projects for reference

[4] www.iubenda.com- laws and regulations every app developer should know

[5] Nau.edu- Requirements Specification

[6] www.duolingo.com- Duolingo website and App

[7] www.gliffy.com- ER diagram rules

[8] Fundamentals Of Database Systems, Ramez Elmasri and Shamkant B. Navathe, Pearson Publications, 7th edition

[9] edisplinas.usp.br

[10] Mobile Multi-domain Search Over Structured Web Data, by Atakan Aral, Research Gate