

# From Watch to Work

Transforming Video Learning with Automated Assessment

Software Design and Requirement Specification



Session: 2021 – 2025

## Submitted by:

Muhammad Shakeel	2021-CS-625
Muhammad Adeel Tariq	2021-CS-684
Huzaifa Shahzad	2021-CS-700

## Supervised by:

Dr. Farah Adeeba

Department of Computer Science, New Campus  
**University of Engineering and Technology**  
**Lahore, Pakistan**

# Contents

<b>List of Figures</b>	<b>iii</b>
<b>List of Tables</b>	<b>v</b>
<b>1 Software Requirement Specification</b>	<b>1</b>
1.1 Functional Requirement . . . . .	1
1.2 Non-functional Requirement . . . . .	5
1.3 Use case . . . . .	6
1.3.1 Sign Up . . . . .	8
1.3.2 Log In . . . . .	9
1.3.3 Log Out . . . . .	10
1.3.4 Input Video Link . . . . .	12
1.3.5 Select Assessment Type . . . . .	13
1.3.6 Complete Assessment . . . . .	14
1.3.7 Submit Assessment . . . . .	16
1.3.8 Evaluate Assessment . . . . .	17
1.3.9 View Score . . . . .	19
<b>2 Design Specification</b>	<b>21</b>
2.1 Wireframes . . . . .	21
2.1.1 Home Page . . . . .	21
2.1.2 User Dashboard . . . . .	23
2.1.3 View Transcript . . . . .	24
2.1.4 View Summary . . . . .	25
2.1.5 User History . . . . .	26
2.1.6 Take Quiz . . . . .	27
2.1.7 Multiple Choice Questions (MCQs) . . . . .	28
2.1.8 Short Questions . . . . .	29
2.1.9 Score . . . . .	30
2.2 Activity Diagram . . . . .	31
2.2.1 User Registration . . . . .	31
2.2.2 User Login . . . . .	31
2.2.3 User Dashboard . . . . .	32
2.2.4 Forget Password . . . . .	33
2.2.5 Link Input and Transcript Generation . . . . .	34

---

2.2.6	User Choices after Transcription . . . . .	34
2.2.7	Assessment Generation . . . . .	35
2.3	State Diagram . . . . .	37
2.3.1	User Registration . . . . .	37
2.3.2	User Login . . . . .	37
2.3.3	User Dashboard . . . . .	38
2.3.4	Forget Password . . . . .	39
2.3.5	Link Input and Transcript Generation . . . . .	40
2.3.6	User Choices after Transcription . . . . .	41
2.3.7	Assessment Generation . . . . .	42
2.4	Sequence Diagram . . . . .	45
2.4.1	User Registration . . . . .	45
2.4.2	User Login . . . . .	45
2.4.3	Forget Password . . . . .	46
2.4.4	Video Upload . . . . .	47
2.4.5	Transcription Process . . . . .	48
2.4.6	Assessment Generation . . . . .	48
2.4.7	Assessment Evaluation . . . . .	49
2.5	Collaboration Diagram . . . . .	50
2.5.1	User Registration . . . . .	51
2.5.2	User Login . . . . .	52
2.5.3	Forget Password . . . . .	52
2.5.4	Upload Video . . . . .	53
2.5.5	Transcription . . . . .	54
2.5.6	Assessment Generation . . . . .	54
2.5.7	Assessment Evaluation . . . . .	55
2.6	Class Diagram . . . . .	56
2.7	Component Diagram . . . . .	58
2.7.1	User Authentication Module . . . . .	60
2.7.2	Video Processing and Transcription Module . . . . .	62
2.7.3	Assessment Generation Module . . . . .	64
2.7.4	Database Service Module . . . . .	67
2.8	Deployment Diagram . . . . .	69
2.9	ER Diagram . . . . .	71

# List of Figures

1.1	Use-Case diagram . . . . .	7
2.1	Wireframe: Home Page . . . . .	22
2.2	Wireframe: User Dashboard . . . . .	23
2.3	Wireframe: View Transcript . . . . .	24
2.4	Wireframe: View Summary . . . . .	25
2.5	Wireframe: View History . . . . .	26
2.6	Wireframe: Take Quiz . . . . .	27
2.7	Wireframe: MCQs . . . . .	28
2.8	Wireframe: Short Questions . . . . .	29
2.9	Wireframe: Score . . . . .	30
2.10	Activity Diagram: User Registration . . . . .	31
2.11	Activity Diagram: User Login . . . . .	32
2.12	Activity Diagram: User Dashboard . . . . .	33
2.13	Activity Diagram: Forget Password . . . . .	33
2.14	Activity Diagram: Link Input and Transcript Generation . . . . .	34
2.15	Activity Diagram: User Choices after Transcription . . . . .	35
2.16	Activity Diagram: Assessment Generation . . . . .	36
2.17	State Diagram: User Registration . . . . .	37
2.18	State Diagram: User Login . . . . .	38
2.19	State Diagram: User Dashboard . . . . .	39
2.20	State Diagram: Forget Password . . . . .	40
2.21	State Diagram: Link Input and Transcript Generation . . . . .	41
2.22	State Diagram: User Choices after Transcription . . . . .	42
2.23	State Diagram: Assessment Generation-1 . . . . .	43
2.24	State Diagram: Assessment Generation-2 . . . . .	44
2.25	Sequence Diagram: User Registration . . . . .	45
2.26	Sequence Diagram: User Login . . . . .	46
2.27	Sequence Diagram: Forget Password . . . . .	47
2.28	Sequence Diagram: Video Upload . . . . .	48
2.29	Sequence Diagram: Transcription Process . . . . .	48
2.30	Sequence Diagram: Assessment Generation . . . . .	49
2.31	Sequence Diagram: Assessment Evaluation . . . . .	49
2.32	Collaboration Diagram . . . . .	51
2.33	Collaboration Diagram: User Registration . . . . .	52

---

2.34	Collaboration Diagram: User Login . . . . .	52
2.35	Collaboration Diagram: User Login . . . . .	53
2.36	Collaboration Diagram: Upload Video . . . . .	53
2.37	Collaboration Diagram: Transcription . . . . .	54
2.38	Collaboration Diagram: Assessment Generation . . . . .	55
2.39	Collaboration Diagram: Assessment Evaluation . . . . .	55
2.40	Class Diagram . . . . .	57
2.41	Component Diagram . . . . .	60
2.42	Component Diagram: User Authentication Module . . . . .	62
2.43	Component Diagram: Video Processing and Transcription Module . . . . .	64
2.44	Component Diagram: Assessment Generation Module . . . . .	66
2.45	Component Diagram: Data Service Module . . . . .	68
2.46	Deployment Diagram . . . . .	70
2.47	ER Diagram . . . . .	73

# List of Tables

1.1	Functional Requirements . . . . .	2
1.2	Non-Functional Requirements . . . . .	5
1.3	Use Case: Sign Up . . . . .	8
1.4	Use Case: Login . . . . .	9
1.5	Use Case: Logout . . . . .	11
1.6	Use Case: Input Video Link . . . . .	12
1.7	Use Case: Select Assessment Type . . . . .	13
1.8	Use Case: Complete Assessment . . . . .	15
1.9	Use Case: Submit Assessment . . . . .	16
1.10	Use Case: Evaluate Assessment . . . . .	17
1.11	Use Case: View Score . . . . .	19
2.1	Wireframe: Home Page . . . . .	21
2.2	Wireframe: User Dashboard . . . . .	23
2.3	Wireframe: View Transcript . . . . .	24
2.4	Wireframe: View Summary . . . . .	25
2.5	Wireframe: User History . . . . .	26
2.6	Wireframe: Take Quiz . . . . .	27
2.7	Wireframe: Multiple Choice Questions (MCQs) . . . . .	28
2.8	Wireframe: Short Questions . . . . .	29
2.9	Wireframe: Score . . . . .	30

# Chapter 1

## Software Requirement Specification

In the context of this project, “From Watch to Work: Transforming Video Learning with Automated Assessment”, both the Functional and Non-functional Requirements have been carefully organized. This categorization aligns with the specific relationships these requirements hold and the precise objectives they serve within this project’s scope. This organized approach ensures a well-structured presentation and simplifies future referencing. It contains one of the following two values:

- Functional Requirements (FR)
- Non-Functional Requirements (NFR)

### 1.1 Functional Requirement

This project focuses on developing a system for the learners who intend to make use of the videos they watch on YouTube to gather the understanding of the video. This system will provide them with a platform where they can get the transcription of the video, generate the summary, get the key concepts discussed in the video, generate different kinds of assessment and get the scores after completing those assessments. To achieve this, here are the outlined functional requirements that describe the specific tasks and capabilities this system should possess.

The functional requirements are represented with Requirement IDs which consist of three parts separated by “-“. The first part is “FR” representing “Functional Requirements”, second part is the requirement identifier which is represented by the first letters of the requirement name, and the third part is the serial number assigned to the requirement and has unique value for instance, FR-LP-01 is the

‘Functional Requirement’ with the heading of ‘Landing Page’.

The Functional Requirements for this project are listed below in (Table 1.1). All the functional requirements listed below have the same priority level in this project and will be implemented in their entirety.

TABLE 1.1: Functional Requirements

ID	Title	Description
FR-LP-01	Landing Page	The system shall have the Home Page which will include information about the system and instructions on how to make use of it efficiently.
FR-LS-01	Login/Sign-up	The system Home page shall have the ‘Log in’ and ‘Sign up’ buttons for the user.
FR-UR-01	User Registration	The system shall have the options to create a new account using a valid email, unique username, and personalized password or the user can continue with their Google or Facebook account.
FR-LG-01	User Login	The user shall be able to ‘Log in’ using their credentials.
FR-LG-02	Credential Validation	The system shall validate the user credentials by fetching the data from the stored data in the system database.
FR-RP-01	Reset Password	The user shall have the option to reset the password by using the ‘Forgot Password’ option while logging into the portal, after providing a valid username or email address.
FR-UD-01	User Dashboard	The user shall land, after signing in with the correct credentials, on the personalized dashboard page.
FR-UD-02	User History	The system shall display the user history on the dashboard including video links the user has provided and the scores of the assessments they have taken.
FR-UD-03	User History	The user shall be able to visit the details of any video link such as transcript, summary, generated assessment, assessment results that user already used in earlier sessions.
FR-VL-01	Video Link Input	The user shall be able to provide a YouTube video link for the input field on the system.
FR-VL-02	Video Link Validation	The system shall validate the YouTube video link.



FR-VL-03	Audio Extrac- tion	The system shall download the video and extract the audio in '.wav' format, with 16kHz frame rate and mono-audio channels, from the provided YouTube link.
FR-NF-01	Audio Extrac- tion Notification	The system shall notify the user whether the download was successful or an error occurred.
FR-TR-01	Transcription Process	The system shall call the Whisper API for the transcrip- tion process of the extracted audio.
FR-TR-02	Language Detec- tion	The Whisper API shall detect the language of the tran- scribed audio.
FR-TR-03	Urdu/Hindi Translation	The API shall translate the transcript to English and keep a copy of the original transcription as well, if the language of the transcribed audio is Urdu/Hindi.
FR-TR-04	Transcription Display Option	The user shall be able to view the transcription in ei- ther Urdu/Hindi or English, if the language of the tran- scribed audio is Urdu/Hindi.
FR-TR-05	Transcription Formatting	The system shall use LLMs for the Bloom's taxonomy deduction for the transcript.
FR-TR-06	Transcription Formatting	The system shall format the transcript into different paragraphs, with each paragraph consisting of a unique topic and labeled according to Bloom's taxonomy levels.
FR-SM-01	Summarization Process	The system shall use LLMs to provide the user with a concise version of the transcribed content.
FR-SM-02	Summarization Analysis	The system shall apply the Bloom's taxonomy to the summarized text to provide the taxonomy level of the entire transcript.
FR-KT-01	Key Topic Iden- tification	The system shall use LLMs to identify the key topics from the transcribed text which hold the most signifi- cance.
FR-KT-02	Key Topic Dis- play	The system shall display all the extracted key topics to the user to provide a better understanding of the transcribed text for the given video.
FR-AD-01	Assessment Type Selection	The user shall select the desired assessment type from the following options: Short Answer Questions, Multiple Choice Questions (MCQs), and Fill-in-the-Blanks.
FR-AD-02	Assessment Generation	The system shall generate assessments using LLMs as per the user's selection.

FR-AD-03	Short Answer Questions Assessment	The system shall generate Short Answer Questions using LLMs from the transcribed text according to the identified levels of Bloom's taxonomy.
FR-AD-04	MCQs Assessment	The system shall generate Multiple Choice Questions (MCQs) using LLMs related to the transcript according to different levels of Bloom's taxonomy.
FR-AD-05	Fill-in-the-Blanks Assessment	The system shall generate fill-in-the-blanks questions using LLMs leaving out essential information from the transcript as blank parts.
FR-AD-06	Assessment Display	The system shall display the generated assessment, as per the user's selection, on the interface.
FR-AI-01	Assessment Interaction	The user shall be able to interact with the displayed assessment by selecting (MCQs) or inputting the answers (Short Q&A and fill-in-the-blanks).
FR-AI-02	Assessment Interaction	The user shall be able to move back and forth between the questions and provide the answers in any order.
FR-AI-03	Assessment Interaction	The system shall keep track of the number of questions already answered by the user and the number of questions remaining to be answered.
FR-AI-04	Assessment Interaction	User shall be able to choose to skip and not answer any question.
FR-AS-01	Assessment Submission	The user shall be able to submit the assessment after completing to system for further processing.
FR-AS-02	Assessment Evaluation	The system shall evaluate the submitted answers.
FR-AS-03	Assessment Evaluation	The system shall consider the unanswered questions as incorrect.
FR-AS-01	Assessment Result	The system shall provide the result in a score based format to the user.
FR-EH-01	Error Handling	The system shall notify the user in case of any errors in a feasible way, like invalid video link, so the user can remove the error easily.
FR-NT-01	System Notifications	The user shall be notified about the downloading status of the video, generation of transcript, generation of summary and extraction of the key-topics. Furthermore, user shall also be notified after the assessment generation and evaluation results.

## 1.2 Non-functional Requirement

In the context of this project, the non-functional requirements will ensure that the system not only functions well but also delivers a superior user experience, comply with the privacy regulations, and adapts to varying conditions. The non-functional requirements are categorized into different groups with each group providing the same functionality. Each group is assigned a unique identifier which are DS for 'Data Storage', SP for 'Scalability and Performance', 'US' for 'Usability', 'AB' for 'Availability and Backup', 'CM' for 'Compatibility', 'MA' for 'Maintainability' and 'DI' for 'Data Integrity'. The requirements are then numbered within each category, starting with 01. The Non-functional Requirements for this project are listed in (Table 1.2):

TABLE 1.2: Non-Functional Requirements

ID	Description
NR-DS-01	The system shall store user data securely, including registration details such as email addresses, usernames, and passwords (using encryption), submitted video links, and generated data.
NR-DS-02	The database shall be available at all times for the retrieval of user data at any time.
NR-DS-03	Submitted video links data shall be used to analyze the learning trends among users.
NR-DS-04	Generated data will be stored for further training and advancement of the system.
NR-SP-01	The system shall handle a significant number of users simultaneously without noticeable performance degradation.
NR-SP-02	The downloading, transcribing, summarization, and assessment generation processes should be optimized for efficiency.
NR-US-01	The system shall have a user-friendly interface, resulting in easy navigation between different pages and smooth interaction with the available features.
NR-US-02	The system shall inform the user in case of errors (if any) and display them in a meaningful way so the user can understand and proceed to solve the error.
NR-AB-01	The system shall have uptime of 99% to ensure users can access their accounts and process videos without interruptions.
NR-AB-02	The system shall perform regular backups of user data and system settings to ensure recovery in case of system failure.

NR-CM-01	The system shall be compatible with modern web browsers such as Chrome.
NR-CM-02	The system shall be responsive and accessible from both desktop and mobile devices.
NR-MA-01	The system shall be designed in a modular fashion, allowing developers to easily update or add new features (e.g., additional question types).
NR-MA-02	The system shall maintain comprehensive logs for debugging and tracking errors.
NR-DI-01	The system shall ensure that video transcriptions, summaries, and generated assessments are stored correctly without data corruption.
NR-DI-02	The system shall validate inputs (e.g., video links) to prevent system errors.

### 1.3 Use case

The Use Case Diagram (Figure 1.1) provides a visual representation of the system's functionality from the perspective of the users (actors). It outlines the interactions between the user and the system, showcasing the various actions a user can perform and the system's responses. Each use case represents a specific goal that the user can accomplish, such as signing up, logging in, providing a YouTube video link, and choosing to view the transcription, summary, or key points of the video. The diagram also includes system-driven actions, such as validating user credentials, processing video links, generating assessments, and providing feedback. By illustrating these interactions, the use case diagram helps clarify the core functionalities of the system and how different processes are interconnected to enhance the user experience.

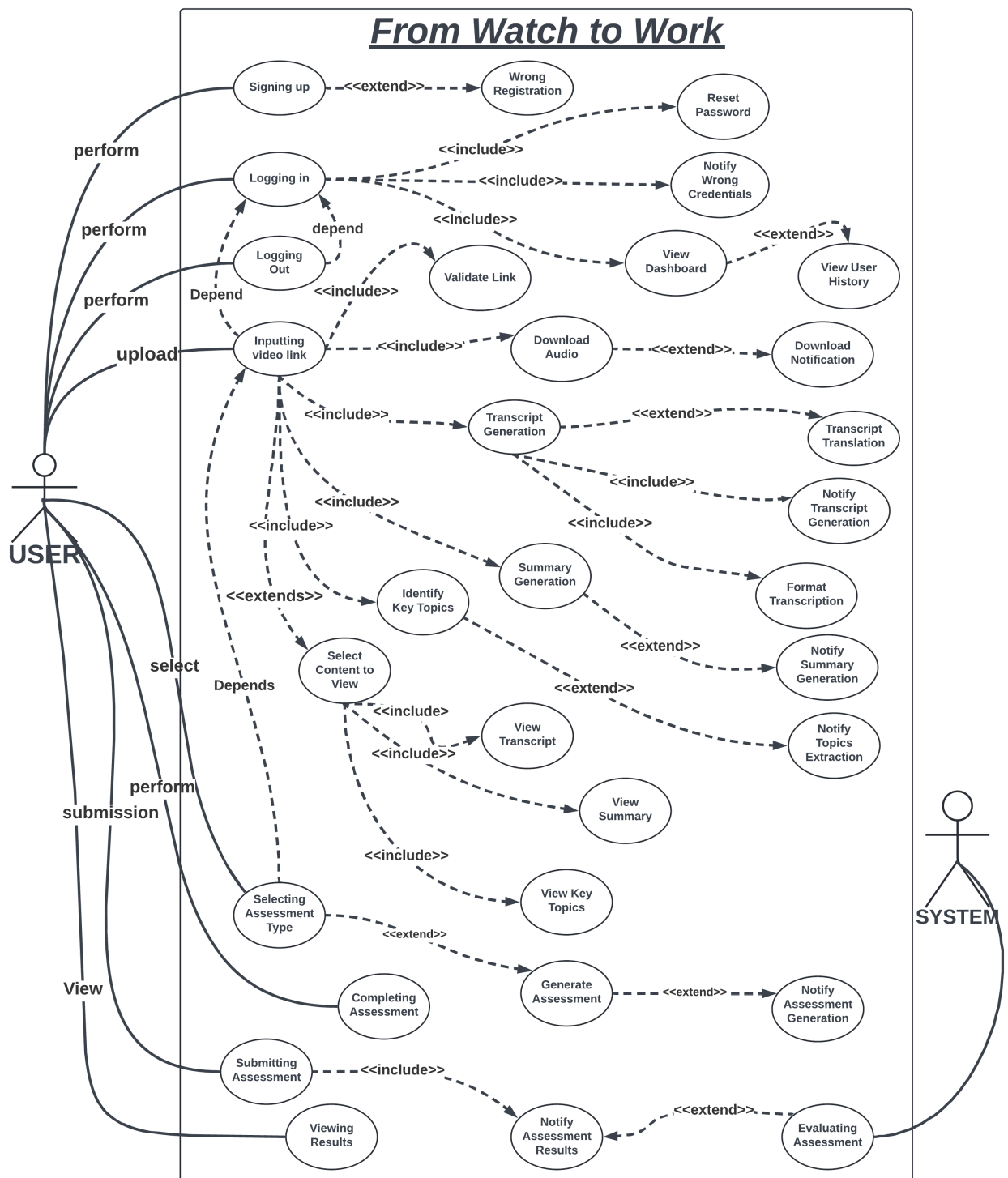


FIGURE 1.1: Use-Case diagram

### 1.3.1 Sign Up

The use case presented in Table 1.3 outlines the process for users to sign up for an account on the platform. It details the primary actor, goal, and triggers involved in the sign-up process. The table further describes the pre-conditions that must be met, the post-conditions following a successful sign-up, and the basic flow of events. Additionally, it addresses potential exceptions and outlines the qualities expected from the sign-up process, emphasizing efficiency and user-friendliness.

TABLE 1.3: Use Case: Sign Up

Use Case	Sign Up
ID	US-01
Primary Actor	User
Goal	To allow the user to sign up for an account.
Trigger	The user initiates the sign-up process.
Pre-conditions	<ul style="list-style-type: none"><li>• The website is operational.</li><li>• The user's credentials are valid.</li></ul>
Post-conditions	<ul style="list-style-type: none"><li>• The user account is created.</li></ul>
Trigger	The user initiates the login process.
Basic Flow	<ol style="list-style-type: none"><li>1. The user enters their username, email and password.</li><li>2. The system validates the input.</li><li>3. Upon successful validation, the user account is created.</li></ol>
Exception	<ul style="list-style-type: none"><li>• If the system is not running: The sign-up process cannot proceed, and an error message is displayed.</li></ul>

Use Case	Sign Up
Qualities	<ul style="list-style-type: none"><li>• The sign-up process should be quick and efficient, allowing users to create an account without significant delays.</li><li>• The sign-up interface should be user-friendly, making it easy for users to enter their username, email, and password accurately.</li><li>• In the event of sign-up failures (e.g., due to incorrect credentials or system issues), the system should provide clear and informative error messages to assist the user in resolving the problem.</li></ul>

1.3.2 Log In

Table 1.4 outlines the login process for users accessing their dashboard. It specifies the primary actor, goals, and triggers that initiate the login process. The table includes pre-conditions necessary for a successful login, as well as post-conditions that confirm user access to the dashboard. Additionally, it details the basic flow of events, potential exceptions that may occur, and the qualities expected from the login process, highlighting the need for efficiency and user-friendly design.

TABLE 1.4: Use Case: Login

Use Case	Login
ID	US-02
Primary Actor	User
Goal	To allow the user to log in to the dashboard.
Pre-conditions	<ul style="list-style-type: none"><li>• The website is operational.</li><li>• The user’s credentials are correct.</li></ul>

Use Case	Login
Post-conditions	<ul style="list-style-type: none"><li>• The user gains access to the dashboard.</li></ul>
Trigger	The user initiates the login process.
Basic Flow	<ol style="list-style-type: none"><li>1. The user enters their username and password.</li><li>2. The system validates the credentials.</li><li>3. Upon successful validation, the user gains access to the system.</li></ol>
Exception	<ul style="list-style-type: none"><li>• If the system is not running: The login process cannot proceed, and an error message is displayed.</li></ul>
Qualities	<ul style="list-style-type: none"><li>• The login process should be quick and efficient, allowing users to access the system without significant delays.</li><li>• The login interface should be user-friendly, making it easy for users to enter their username and password accurately.</li><li>• In the event of login failures, the system should provide clear and informative error messages.</li></ul>

### 1.3.3 Log Out

Table 1.5 details the logout process for users wishing to exit their accounts. It identifies the primary actor, the goals, and the triggers that initiate the logout action. The table outlines the pre-conditions required for a successful logout and the post-conditions confirming that the user has been logged out. Additionally, it describes the basic flow of events, potential exceptions that may arise, and the qualities expected from the logout process, emphasizing the importance of efficiency and user clarity.



TABLE 1.5: Use Case: Logout

<b>Use Case</b>	<b>Logout</b>
<b>ID</b>	US-03
<b>Primary Actor</b>	User
<b>Goal</b>	To allow the user to log out from the account.
<b>Pre-conditions</b>	<ul style="list-style-type: none"><li>• The website is operational.</li><li>• The user is logged in already.</li></ul>
<b>Post-conditions</b>	<ul style="list-style-type: none"><li>• The user is successfully logged out of the account.</li></ul>
<b>Trigger</b>	The user initiates the logout process.
<b>Basic Flow</b>	<ol style="list-style-type: none"><li>1. The user selects the "Logout" option.</li><li>2. The system confirms the user's intention to log out.</li><li>3. Upon confirmation, the system terminates the user's session and logs them out.</li></ol>
<b>Exception</b>	<ul style="list-style-type: none"><li>• If the system is not running: The log-out process cannot proceed, and an error message is displayed.</li></ul>
<b>Qualities</b>	<ul style="list-style-type: none"><li>• The log-out process should be quick and efficient, allowing users to log out without significant delays.</li><li>• In the event of log-out failures, the system should provide clear and informative error messages to assist the user in resolving the problem.</li></ul>

### 1.3.4 Input Video Link

Table 1.6 describes the process by which users can input or paste a video URL on the platform. It specifies the primary actor, the goals of the action, and the triggers that initiate the input process. The table outlines the necessary pre-conditions and post-conditions, confirming that the link has been successfully entered. Additionally, it details the basic flow of events, potential exceptions that may occur, and the expected qualities of the input process, highlighting the need for efficiency and clear error messaging in case of failures.

TABLE 1.6: Use Case: Input Video Link

Use Case	Input Video Link
<b>ID</b>	US-04
<b>Primary Actor</b>	User
<b>Goal</b>	To allow the user to input/paste a URL.
<b>Pre-conditions</b>	<ul style="list-style-type: none"> <li>• The website is operational.</li> <li>• The user is logged in already.</li> </ul>
<b>Post-conditions</b>	<ul style="list-style-type: none"> <li>• The link is successfully entered.</li> </ul>
<b>Trigger</b>	The user initiates the input process.
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1. The user selects the input box.</li> <li>2. The user pastes or inputs the URL.</li> <li>3. The system validates the input URL.</li> <li>4. The URL is confirmed and validated, showing loading for further processing.</li> </ol>

Use Case	Input Video Link
<b>Exception</b>	<ul style="list-style-type: none"> <li>• If the system is not running: The input process cannot proceed, and an error message is displayed.</li> <li>• If the URL is not valid: The input process cannot proceed, and an error message is displayed.</li> </ul>
<b>Qualities</b>	<ul style="list-style-type: none"> <li>• The input process should be quick and efficient, allowing easy navigation of the input box.</li> <li>• In the event of any kind of failure, the system should provide clear and informative error messages to assist the user in resolving the problem.</li> </ul>

### 1.3.5 Select Assessment Type

Table 1.7 outlines the process for users to select their desired assessment type within the platform. It identifies the primary actor, the goals, and the triggers that initiate the assessment selection process. The table includes necessary pre-conditions and post-conditions, confirming that the selected assessment type is generated based on the video transcript. Additionally, it describes the basic flow of events, potential exceptions that may arise, and the qualities expected from the assessment selection process, emphasizing the importance of an intuitive and user-friendly interface.

TABLE 1.7: Use Case: Select Assessment Type

Use Case	Select Assessment Type
<b>ID</b>	US-05
<b>Primary Actor</b>	User
<b>Goal</b>	To allow the user to select a desired assessment type.

Use Case	Select Assessment Type
<b>Pre-conditions</b>	<ul style="list-style-type: none"><li>• The user is logged in.</li><li>• The transcript and key points of the video are generated.</li></ul>
<b>Post-conditions</b>	<ul style="list-style-type: none"><li>• The system generates the selected assessment type.</li></ul>
<b>Trigger</b>	The user initiates the assessment selection process.
<b>Basic Flow</b>	<ol style="list-style-type: none"><li>1. The user selects the "Assessment" option.</li><li>2. The user is presented with the types of assessments: Short Answer Questions, MCQs, or Fill-in-the-Blanks.</li><li>3. The user selects their preferred assessment type.</li><li>4. The system generates the selected assessment type based on the transcript.</li></ol>
<b>Exception</b>	<ul style="list-style-type: none"><li>• If the system is not running: The assessment selection cannot proceed, and an error message is displayed.</li></ul>
<b>Qualities</b>	<ul style="list-style-type: none"><li>• The assessment selection process should be intuitive and user-friendly.</li><li>• The system should provide a clear interface to choose the desired assessment type.</li></ul>

### 1.3.6 Complete Assessment

Table 1.8 outlines the process for users to complete an assessment on the platform. It identifies the primary actor, goals, and triggers that initiate the assessment completion process. The table specifies the preconditions that must be met before starting the assessment and the post-conditions confirming the completion of the

assessment. Additionally, it describes the basic flow of events, potential exceptions that may occur, and the qualities expected from the assessment interface, emphasizing the need for clarity, ease of navigation, and timely completion.

TABLE 1.8: Use Case: Complete Assessment

Use Case	Complete Assessment
ID	US-06
Primary Actor	User
Goal	To allow the user to complete an assessment.
Preconditions	<ul style="list-style-type: none"><li>• The user is logged in.</li><li>• The user has selected an assessment type.</li></ul>
Postconditions	<ul style="list-style-type: none"><li>• The user completes the assessment.</li></ul>
Trigger	The user initiates the completion of the assessment.
Basic Flow	<ol style="list-style-type: none"><li>1. The user is presented with the assessment questions.</li><li>2. The user answers each question according to the selected assessment type.</li><li>3. The user finishes the assessment.</li></ol>
Exception	<ul style="list-style-type: none"><li>• If the system is not running: The assessment process cannot proceed, and an error message is displayed.</li></ul>

Use Case	Complete Assessment
Qualities	<ul style="list-style-type: none"> <li>• The assessment interface should be easy to navigate and user-friendly.</li> <li>• The questions should be clearly presented.</li> <li>• The user should be able to complete the assessment without delays.</li> </ul>

### 1.3.7 Submit Assessment

Table 1.9 details the process for users to submit their completed assessments. It identifies the primary actor, the goals of the submission, and the triggers that initiate this action. The table outlines the necessary pre-conditions for submitting the assessment and post-conditions confirming that the assessment is stored in the system. Additionally, it describes the basic flow of events, potential exceptions that may occur, and the qualities expected from the submission process, emphasizing the importance of speed, reliability, and clear user notifications upon successful submission.

TABLE 1.9: Use Case: Submit Assessment

Use Case	Submit Assessment
ID	US-07
Primary Actor	User
Goal	To allow the user to submit the completed assessment.
Pre-conditions	<ul style="list-style-type: none"> <li>• The user is logged in.</li> <li>• The user has completed an assessment.</li> </ul>
Post-conditions	<ul style="list-style-type: none"> <li>• The assessment is submitted and stored in the system.</li> </ul>
Trigger	The user submits the completed assessment.

Use Case	Submit Assessment
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1. The user clicks the "Submit" button after completing the assessment.</li> <li>2. The system stores the user's responses in the database.</li> <li>3. The user is notified of successful submission.</li> </ol>
<b>Exception</b>	<ul style="list-style-type: none"> <li>• If the system is not running: The assessment submission process cannot proceed, and an error message is displayed.</li> </ul>
<b>Qualities</b>	<ul style="list-style-type: none"> <li>• The submission process should be fast and reliable.</li> <li>• The system should provide clear confirmation that the assessment was submitted successfully.</li> </ul>

### 1.3.8 Evaluate Assessment

Table 1.10 describes the process for the system to automatically evaluate submitted assessments. It identifies the primary actor as the system and outlines the goal of calculating the score and storing the result. The table specifies the pre-conditions necessary for evaluation and the post-conditions confirming that the score is saved in the user's account. Additionally, it details the basic flow of events, potential exceptions that may arise during the evaluation process, and the qualities expected from the evaluation, emphasizing the need for efficiency, accuracy, and clear error messaging in case of failures.

TABLE 1.10: Use Case: Evaluate Assessment

Use Case	Evaluate Assessment
<b>ID</b>	US-08
<b>Primary Actor</b>	System

Use Case	Evaluate Assessment
Goal	To automatically evaluate the submitted assessment, calculate the score, and store the result.
Pre-conditions	<ul style="list-style-type: none"><li>• The user has submitted a completed assessment.</li><li>• The system has access to the correct answers or a scoring algorithm for the assessment type.</li></ul>
Post-conditions	<ul style="list-style-type: none"><li>• The score is calculated and saved in the user's account.</li><li>• The user is notified that the evaluation is complete and the score is available for viewing.</li></ul>
Trigger	The user submits the completed assessment.
Basic Flow	<ol style="list-style-type: none"><li>1. The user submits the completed assessment.</li><li>2. The system retrieves the submitted answers and compares them with the correct answers.</li><li>3. The system calculates the score based on the user's responses.</li><li>4. The score and assessment results are stored in the user's account.</li><li>5. The system sends a notification to the user that the score is ready.</li></ol>
Exception	<ul style="list-style-type: none"><li>• If the system encounters an issue retrieving the answers: An error message is displayed, and the issue is logged.</li><li>• If the system cannot access the correct answers or scoring mechanism: The system logs the error and informs the user that the assessment cannot be evaluated at this time.</li></ul>



Use Case	Evaluate Assessment
Qualities	<ul style="list-style-type: none"> <li>• The evaluation process should be efficient, providing timely results after submission.</li> <li>• The grading should be accurate and reliable.</li> <li>• Any evaluation failures should trigger meaningful error messages and proper logging for debugging.</li> </ul>

### 1.3.9 View Score

Table 1.11 outlines the process for users to view their scores after submitting an assessment. It identifies the primary actor as the user and describes the goal of enabling score visibility. The table lists the pre-conditions required for viewing the score, such as being logged in and having submitted an assessment. The post-conditions confirm that the user can view their score and any associated feedback. The basic flow details the steps taken by the user to access their score, while exceptions account for potential issues like system unavailability. The qualities emphasize the need for a quick and user-friendly interface for score viewing.

TABLE 1.11: Use Case: View Score

Use Case	View Score
ID	US-09
Primary Actor	User
Goal	To allow the user to view their score after submitting the assessment.
Pre-conditions	<ul style="list-style-type: none"> <li>• The user is logged in.</li> <li>• The user has submitted an assessment.</li> </ul>
Post-conditions	<ul style="list-style-type: none"> <li>• The user views their score and feedback, if any.</li> </ul>
Trigger	The user navigates to the score viewing page.

Use Case	View Score
Basic Flow	<ol style="list-style-type: none"><li>1. The user clicks on the "View Score" option.</li><li>2. The system retrieves the score and assessment feedback.</li><li>3. The user views their score.</li></ol>
Exception	<ul style="list-style-type: none"><li>• If the system is not running: The user cannot view the score, and an error message is displayed.</li></ul>
Qualities	<ul style="list-style-type: none"><li>• The score viewing process should be quick and efficient.</li><li>• The system should provide a clear interface to display scores and feedback.</li></ul>

# Chapter 2

## Design Specification

### 2.1 Wireframes

Wireframes hold significant importance in the design and development process of software applications, websites, and user interfaces. They serve as visual blueprints that outline the layout, structure, and basic elements of a user interface before the actual design and development work begins.

#### 2.1.1 Home Page

The wireframe activity outlines the user tasks for the Home Page, as detailed in Table 2.1. This table enumerates specific tasks that users can perform, such as logging in, viewing the header, interacting with the Call-to-Action button, and accessing various sections of the website. Figure 2.1 visually represents these tasks and the overall layout of the Home Page, providing a clear framework for user interaction.

TABLE 2.1: Wireframe: Home Page

ID	User Task
T1	As a user, I shall be able to click the Login / Sign-up button.
T2	As a user, I shall be able to view the header.
T3	As a user, I shall be able to use the link button for pages.
T4	As a user, I shall be able to see the website's logo.
T5	As a user, I shall be able to view the hero section.
T6	As a user, I shall be able to view the image.
T7	As a user, I shall be able to read the headline.
T8	As a user, I shall be able to read the description text.
T9	As a user, I shall be able to interact with the CTA (Call-to-Action) button.

T10	As a user, I shall be able to view the body of the website.
T11	As a user, I shall be able to view the footer of the website.
T12	As a user, I shall be able to view the content of the footer.



FIGURE 2.1: Wireframe: Home Page

## 2.1.2 User Dashboard

The wireframe activity outlines the user tasks for the User Dashboard, as detailed in Table 2.2. This table enumerates specific tasks that users can perform, such as seeing their name after login, inputting a URL, viewing the video thumbnail, and accessing the transcript and summary. Additionally, it includes the ability to take quizzes and check their history. The wireframe diagram fig.2.2 visually represents these tasks and the overall layout of the User Dashboard, providing a clear framework for user interaction.

TABLE 2.2: Wireframe: User Dashboard

ID	User Task
T1	As a user, I shall be able to see my name after login.
T2	As a user, I shall be able to input the URL.
T3	As a user, I shall be able to see the video thumbnail.
T4	As a user, I shall be able to view the transcript.
T5	As a user, I shall be able to view the summary.
T6	As a user, I shall be able to take the quiz.
T7	As a user, I shall be able to go to the input URL interface.
T8	As a user, I shall be able to check my history.

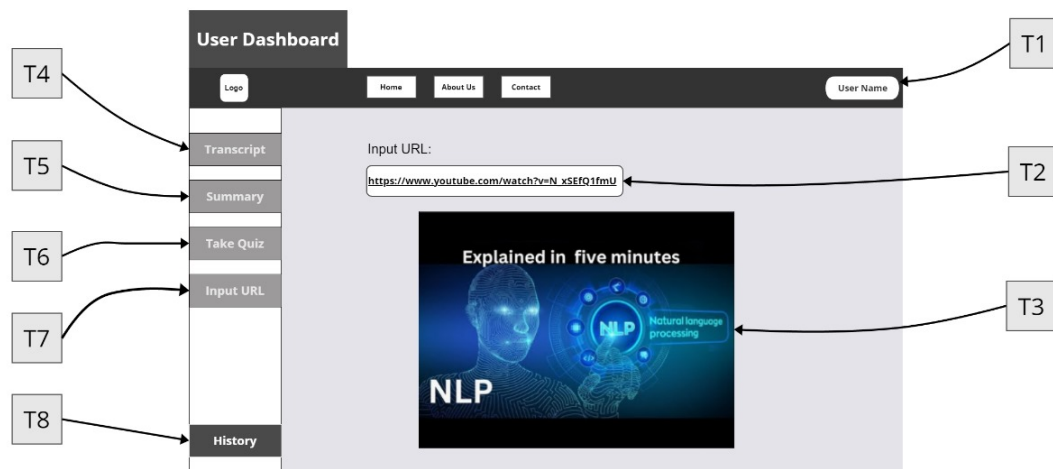


FIGURE 2.2: Wireframe: User Dashboard

### 2.1.3 View Transcript

The wireframe activity outlines the user tasks for viewing the transcript, as detailed in Table 2.3. This table specifies the task that users can perform, which is viewing the generated transcript. Figure 2.3 visually represents this task and the overall layout for the transcript viewing interface, providing a clear framework for user interaction.

TABLE 2.3: Wireframe: View Transcript

ID	User Task
T1	As a user, I shall be able to view the transcript generated.



FIGURE 2.3: Wireframe: View Transcript

### 2.1.4 View Summary

The wireframe activity outlines the user tasks for viewing the summary, as detailed in Table 2.4. This table specifies the task that users can perform, which is viewing the generated summary. Figure 2.4 visually represents this task and the overall layout for the summary viewing interface, providing a clear framework for user interaction.

TABLE 2.4: Wireframe: View Summary

ID	User Task
T1	As a user, I shall be able to view the summary generated.

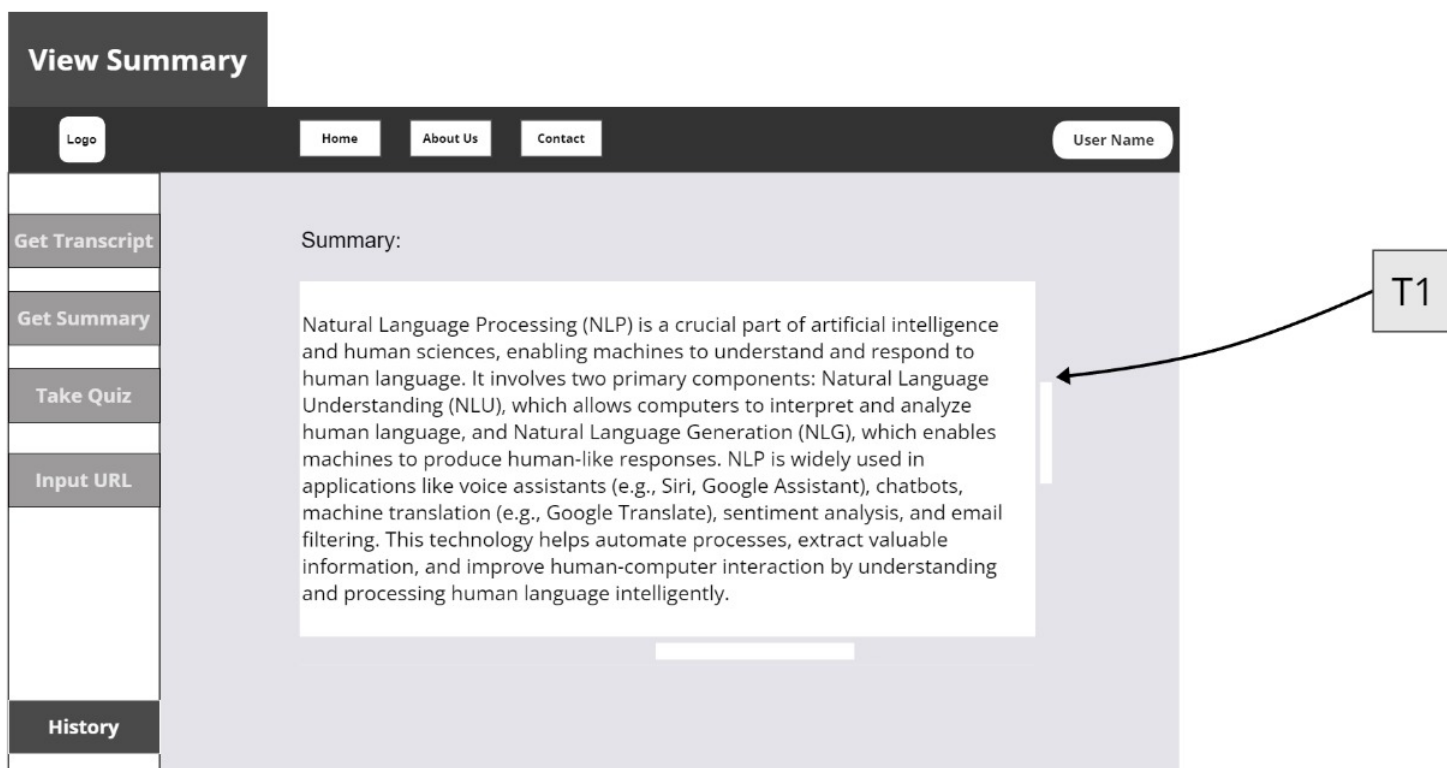


FIGURE 2.4: Wireframe: View Summary

2.1.5 User History

The wireframe activity outlines the user tasks for viewing user history, as detailed in Table 2.5. This table specifies the task that users can perform, which is viewing their user history. Figure 2.5 visually represents this task and the overall layout for the user history interface, providing a clear framework for user interaction.

TABLE 2.5: Wireframe: User History

ID	User Task
T1	As a user, I shall be able to view my user history.

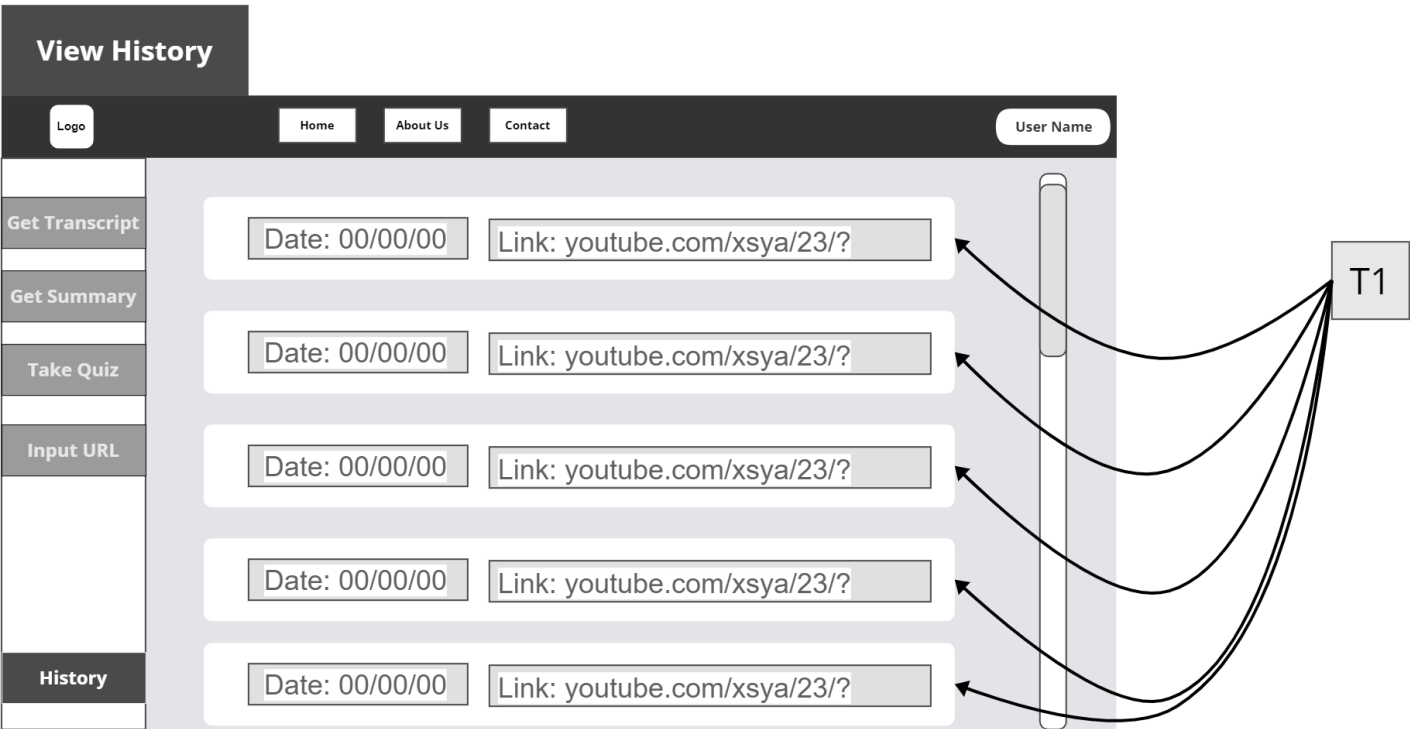


FIGURE 2.5: Wireframe: View History



2.1.6 Take Quiz

The wireframe activity outlines the user tasks for taking a quiz, as detailed in Table 2.6. This table specifies the tasks that users can perform, including selecting the question type and the quiz type. Figure 2.6 visually represents these tasks and the overall layout for the quiz interface, providing a clear framework for user interaction.

TABLE 2.6: Wireframe: Take Quiz

ID	User Task
T1	As a user, I shall be able to select the question type.
T2	As a user, I shall be able to select the quiz type.

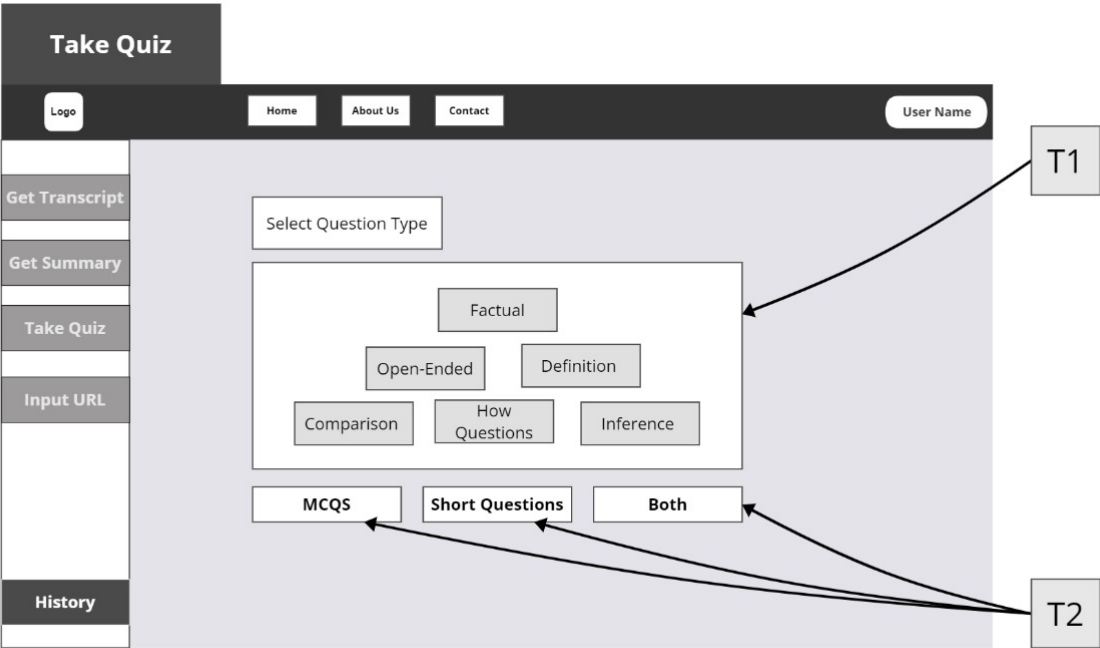


FIGURE 2.6: Wireframe: Take Quiz

### 2.1.7 Multiple Choice Questions (MCQs)

The wireframe activity outlines the user tasks for Multiple Choice Questions (MCQs), as detailed in Table 2.7. This table specifies the tasks that users can perform, including reading the question, selecting an option, and moving to the next question. Figure 2.7 visually represents these tasks and the overall layout for the MCQs interface, providing a clear framework for user interaction.

TABLE 2.7: Wireframe: Multiple Choice Questions (MCQs)

ID	User Task
T1	As a user, I shall be able to read the question.
T2	As a user, I shall be able to select an option.
T3	As a user, I shall be able to move to the next question.

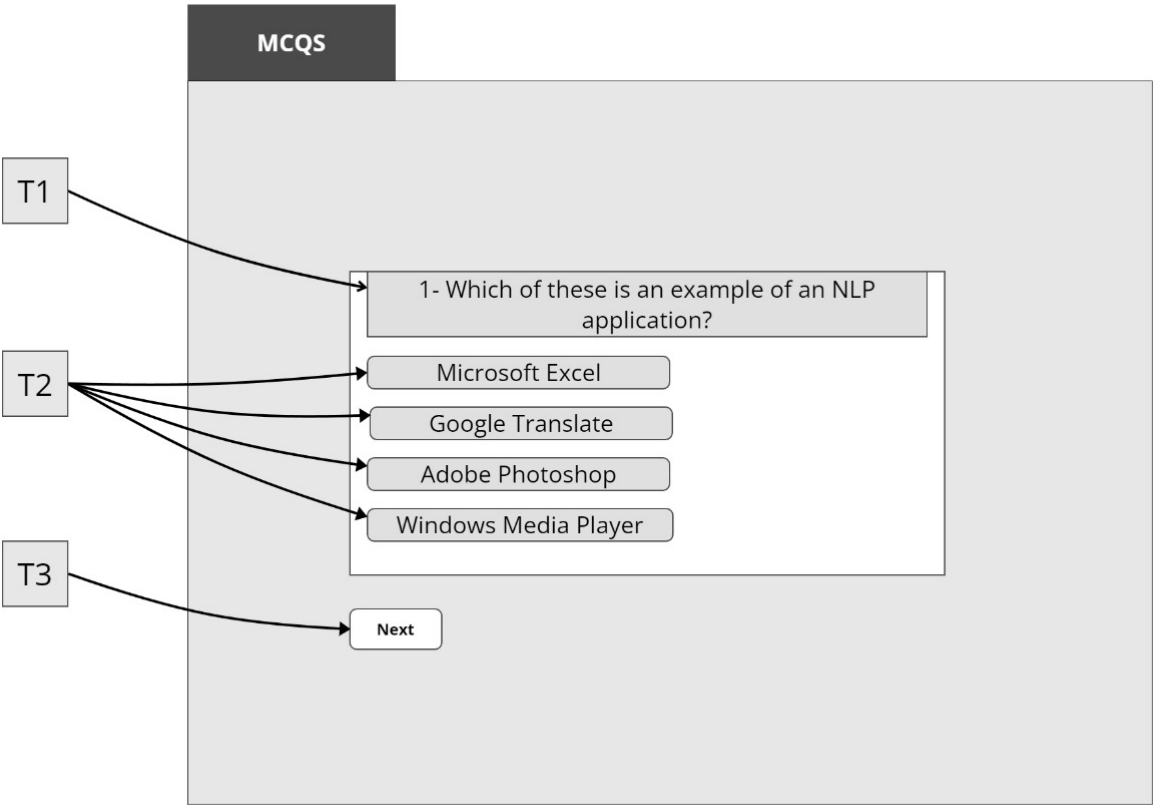


FIGURE 2.7: Wireframe: MCQs

2.1.8 Short Questions

The wireframe activity outlines the user tasks for Short Questions, as detailed in Table 2.8. This table specifies the tasks that users can perform, including reading the question, inputting their answer, navigating to previous and next questions, and submitting the quiz. Figure 2.8 visually represents these tasks and the overall layout for the Short Questions interface, providing a clear framework for user interaction.

TABLE 2.8: Wireframe: Short Questions

ID	User Task
T1	As a user, I shall be able to read the question.
T2	As a user, I shall be able to input my answer.
T3	As a user, I shall be able to go to the previous question.
T4	As a user, I shall be able to go to the next question.
T5	As a user, I shall be able to submit the quiz.

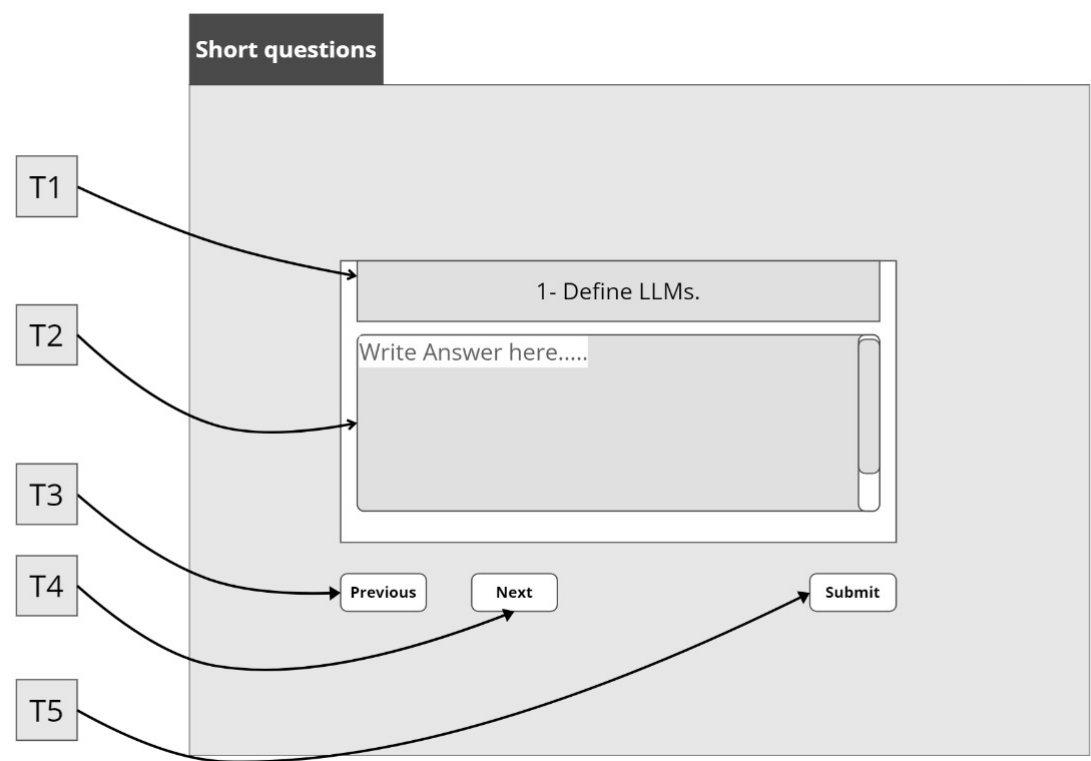


FIGURE 2.8: Wireframe: Short Questions

2.1.9 Score

The wireframe activity outlines the user tasks for viewing the Score, as detailed in Table 2.9. This table specifies the tasks that users can perform, including viewing their score and ending the quiz. Figure 2.9 visually represents these tasks and the overall layout for the Score interface, providing a clear framework for user interaction.

TABLE 2.9: Wireframe: Score

ID	User Task
T1	As a user, I shall be able to see my score.
T2	As a user, I shall be able to end the quiz.

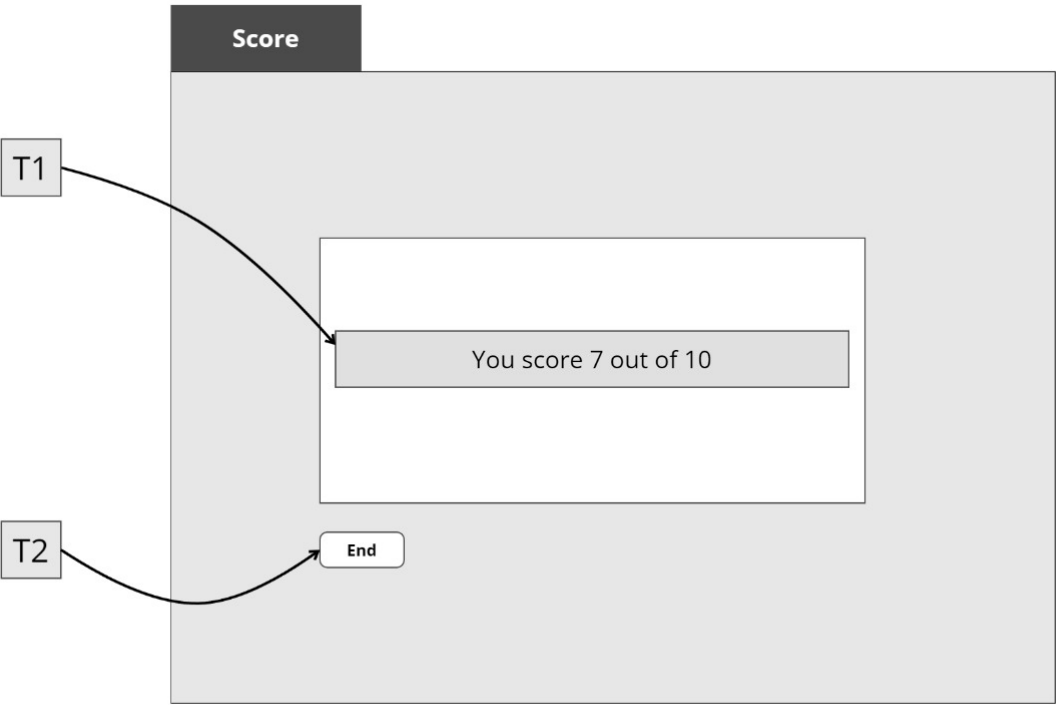


FIGURE 2.9: Wireframe: Score

## 2.2 Activity Diagram

Activity Diagrams represent the dynamic behavior of the system by showcasing the sequence of actions from one activity to another. The activity diagrams below illustrate the various stages of user interaction, video transcription, and assessment generation in the project, highlighting the decision points and concurrent actions.

### 2.2.1 User Registration

The User Registration Activity Diagram (Figure 2.10) outlines the step-by-step process for new users to create an account in the system. It begins with the user accessing the registration page and provides options for entering a valid email, unique username, and a personalized password. The diagram captures decision points, such as validating input and handling errors, ensuring a streamlined registration process. Once registered, the user receives confirmation and is directed to the login screen to access their account.

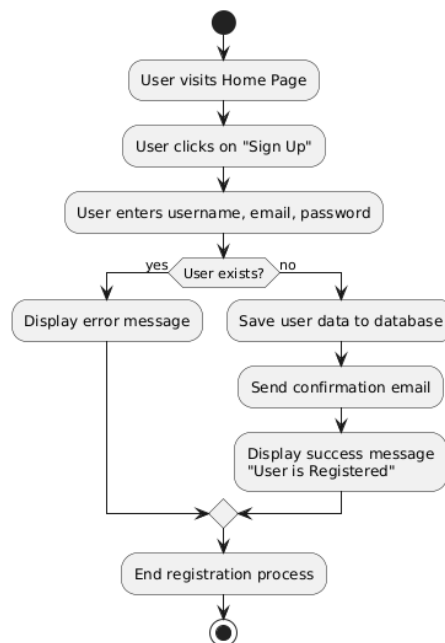


FIGURE 2.10: Activity Diagram: User Registration

### 2.2.2 User Login

The User Login Activity Diagram (Figure 2.11) depicts the login process for users. It starts with the user initiating a login request, with options to enter credentials or log in using Google or Facebook. The system validates the provided credentials, and in case of success, the user is directed to the dashboard. If the credentials are invalid, an error message is displayed, with the option to retry. The diagram

effectively captures the parallel paths of logging in through traditional credentials or social media.

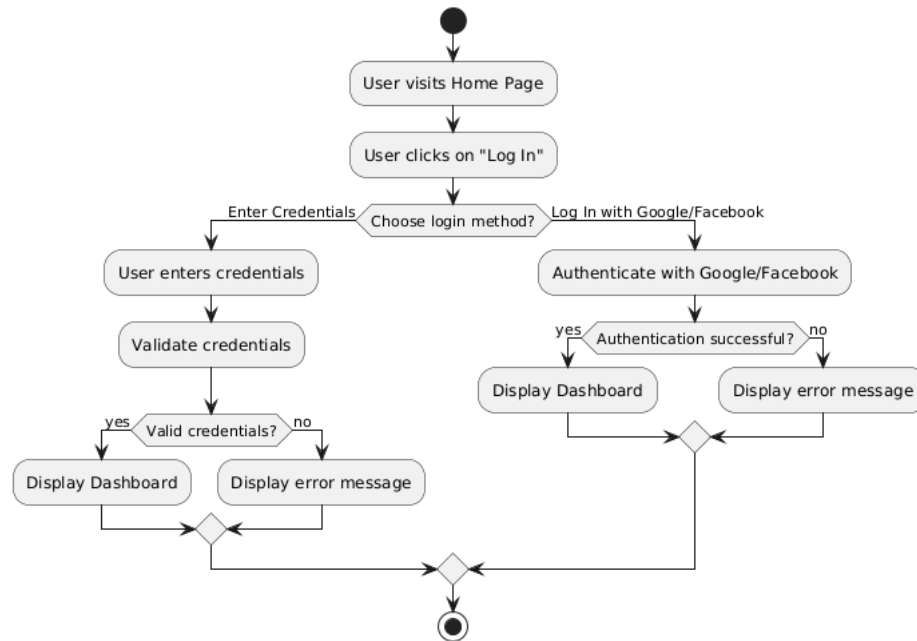


FIGURE 2.11: Activity Diagram: User Login

### 2.2.3 User Dashboard

The User Dashboard Activity Diagram (Figure 2.12) outlines the process through which users can access and review their past interactions within the system. It begins with the user selecting the option to view their history, which includes previously uploaded video URLs and results from assessments they have completed. Once the user initiates this action, the system retrieves the relevant data from its database. The diagram illustrates the flow of activities, including the presentation of video links and assessment results in an organized manner, allowing users to easily navigate through their past activities.

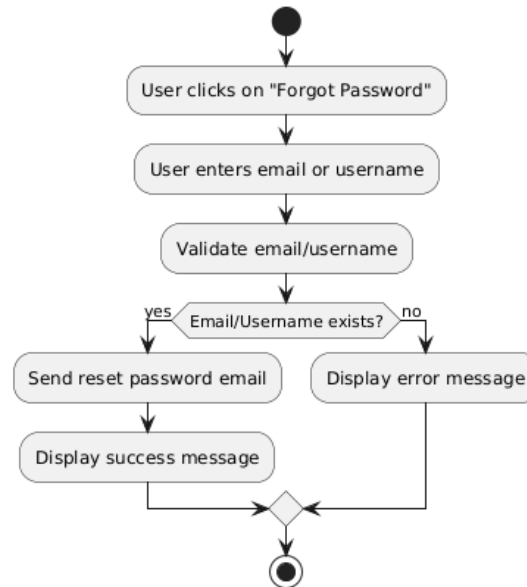


FIGURE 2.12: Activity Diagram: User Dashboard

## 2.2.4 Forget Password

The Forget Password Activity Diagram (Figure 2.13) illustrates the process a user follows to reset their password within the system. It begins with the user initiating the password recovery by entering their registered email address. The system then verifies whether the provided email exists in its database. If the email is valid, the system sends a reset link to the user's email, allowing them to create a new password. If the email does not exist, an error message is displayed, prompting the user to try again or check their email for accuracy.

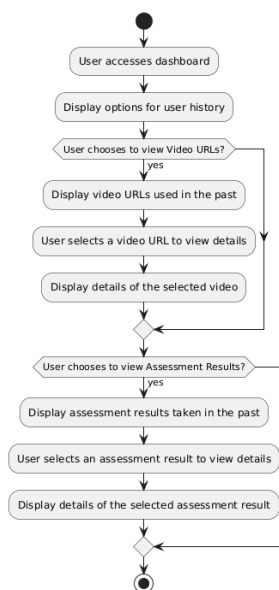


FIGURE 2.13: Activity Diagram: Forget Password

### 2.2.5 Link Input and Transcript Generation

The Video Upload and Transcription Activity Diagram (Figure 2.14) delineates the workflow involved when a user uploads a YouTube video for transcription. The process begins with the user selecting a video to upload, which triggers the system to validate the provided YouTube link. Upon successful validation, the system downloads the video and extracts the audio in the specified format. Once the audio is prepared, the system calls the transcription API to generate a textual transcript of the audio content.

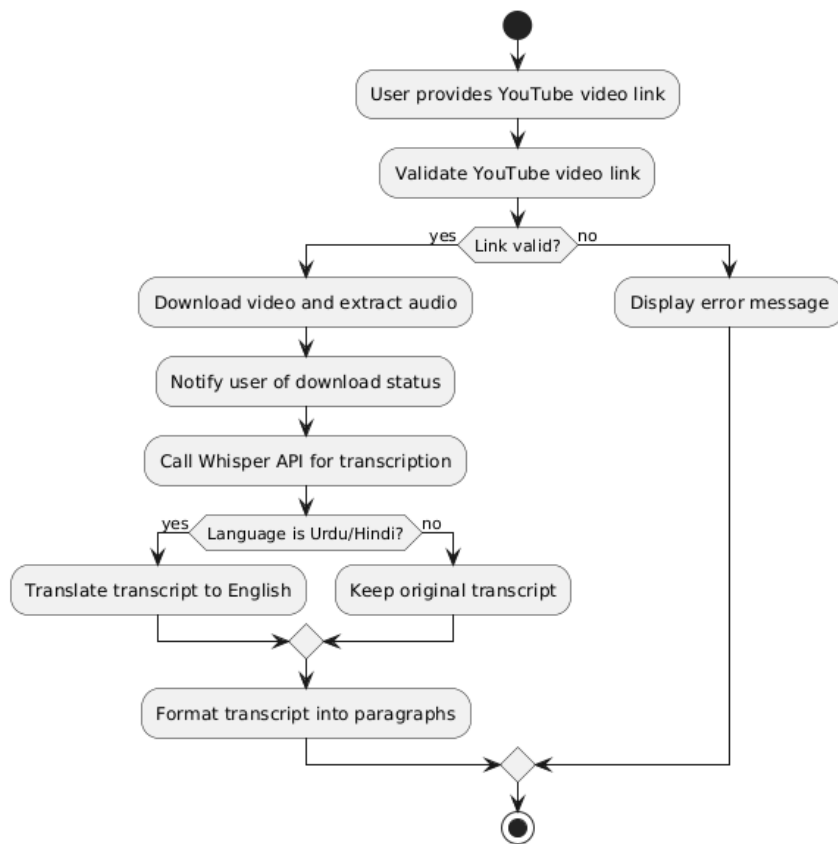


FIGURE 2.14: Activity Diagram: Link Input and Transcript Generation

### 2.2.6 User Choices after Transcription

The User Choices After Transcript Generation Activity Diagram (Figure 2.15) illustrates the options available to users immediately following the successful generation of a transcript from their uploaded video. After the transcription process is complete, users are presented with three simultaneous choices:

1. View Summary - Users can opt to see a concise summary of the transcript, highlighting key points and essential information.



2. View Key Concepts - This option allows users to access the most significant themes and ideas extracted from the transcript.
3. View Transcription - Users can select this option to review the full transcript, with the additional choice of viewing either the original or a translated version, specifically if the content is in Urdu or Hindi.

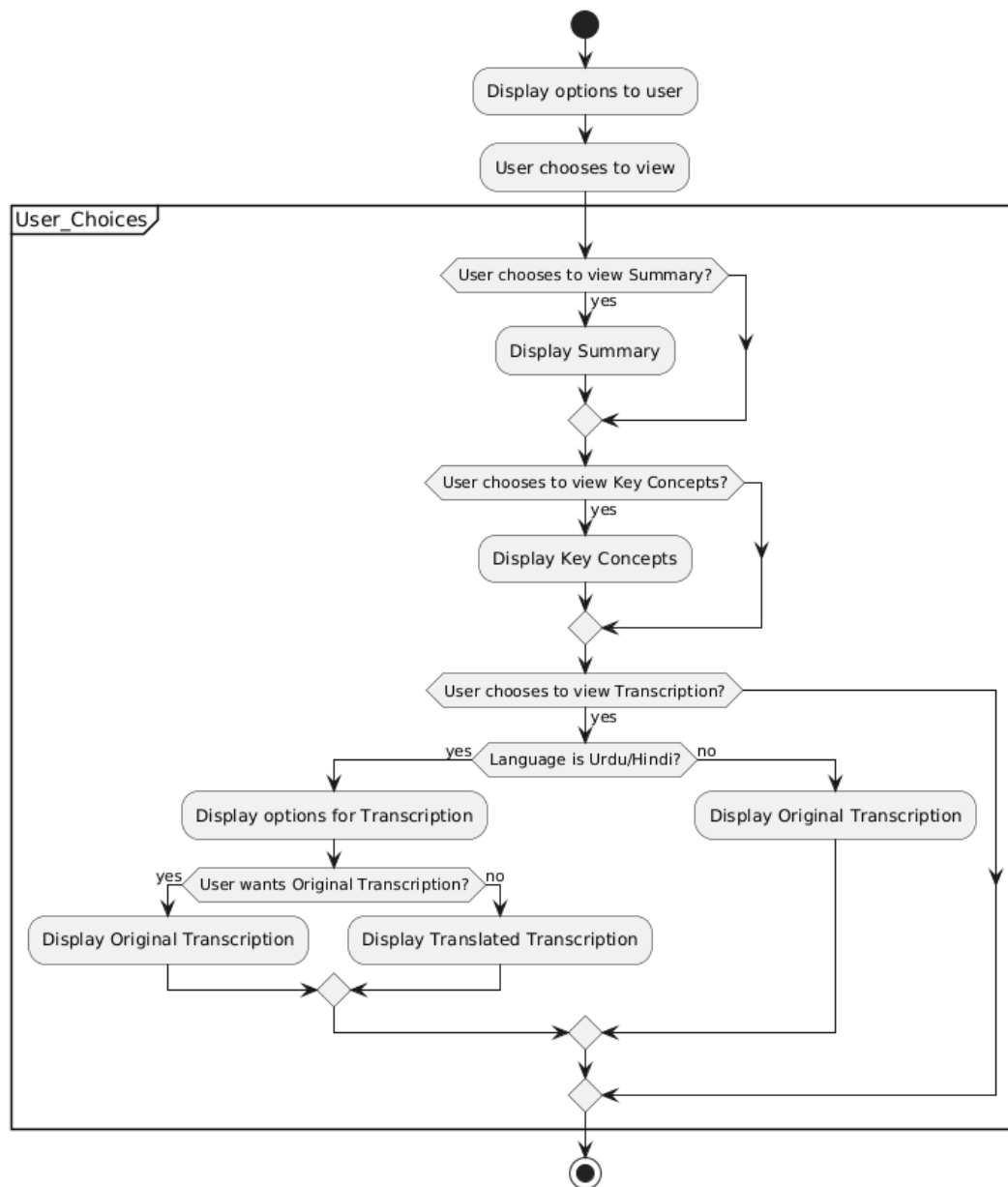


FIGURE 2.15: Activity Diagram: User Choices after Transcription

### 2.2.7 Assessment Generation

The Assessment Generation Activity Diagram (Figure 2.16) details the process a user undergoes to generate assessments based on the transcribed content of

their uploaded videos. The workflow begins with the user selecting the type of assessment they wish to create, such as Short Answer Questions, Multiple Choice Questions (MCQs), or Fill-in-the-Blanks. Following this selection, the system generates the assessment according to the user's requirements. Users are then presented with the assessment interface, where they can interact with the questions by providing answers or opting to skip any questions they choose not to answer. After completing the assessment, users submit their responses for evaluation.

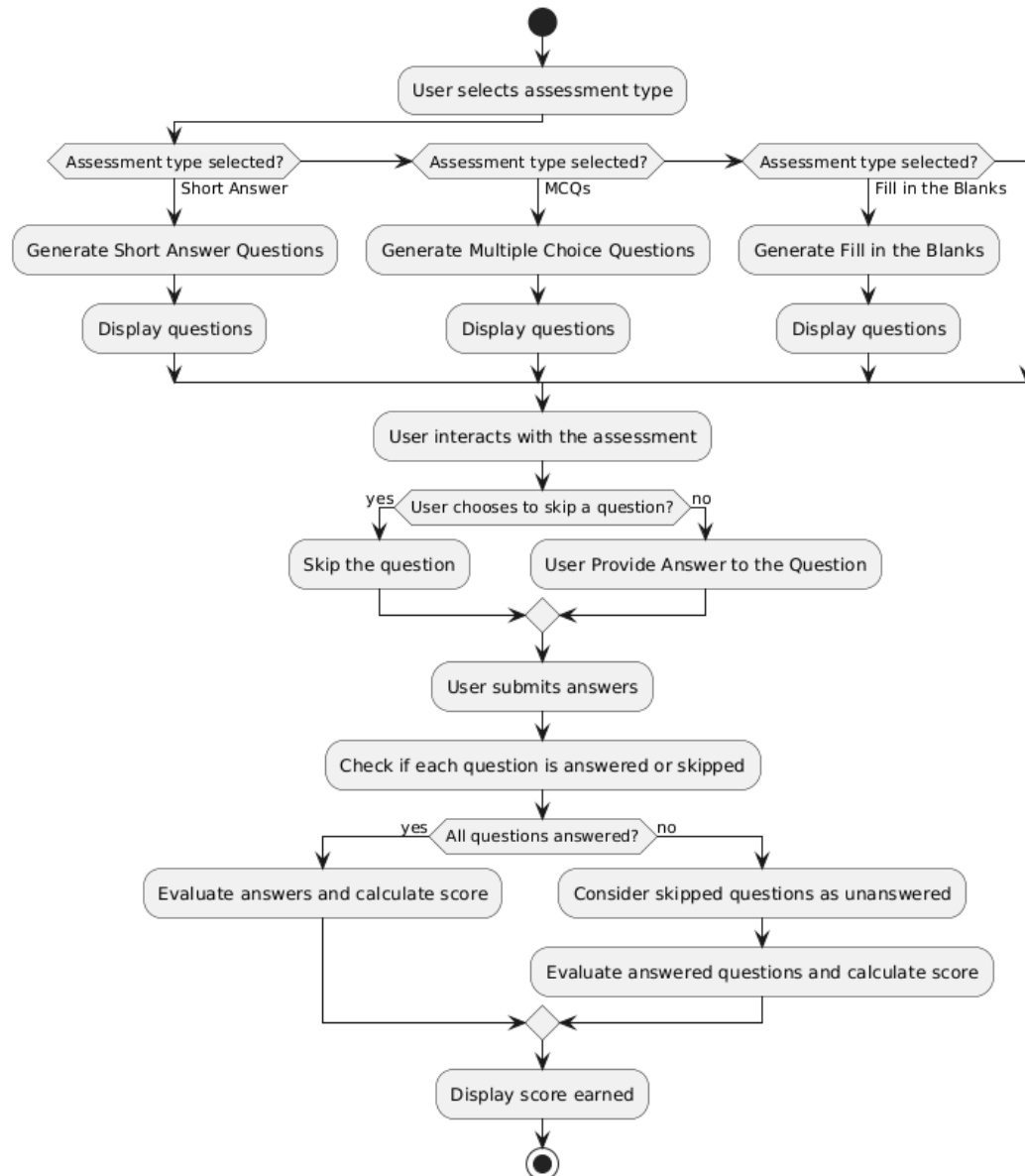


FIGURE 2.16: Activity Diagram: Assessment Generation

## 2.3 State Diagram

In this project, state diagrams are used to model the system's behavior in relation to user history, login, and assessment processes, helping to define how the system responds to different user actions or inputs.

### 2.3.1 User Registration

The User Registration State Diagram (Figure 2.17) illustrates the various states involved in the user registration process within the system. The diagram starts from the Idle state, where the system awaits user interaction. Once the user initiates the registration process, they enter the Enter User Details state, where they provide their email, username, and password. After the details are entered, the system moves to the Validate User Details state, where it checks the validity of the information. If the details are valid, the system transitions to the Create Account state, where the account is successfully set up, and the user is moved to the Registered state. In the case of invalid details, the system returns to the Idle state after displaying an error message.

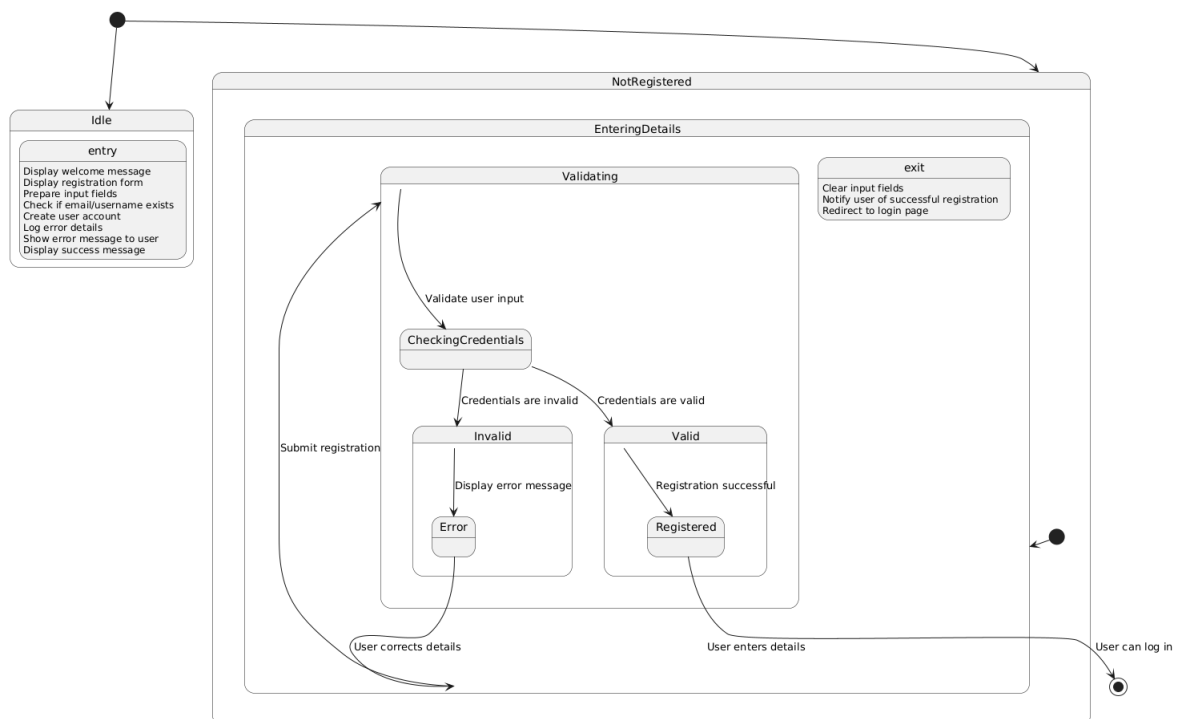


FIGURE 2.17: State Diagram: User Registration

### 2.3.2 User Login

The User Login State Diagram (Figure 2.18) illustrates the different states involved in the user authentication process. The diagram begins from the Idle state,

where the system waits for the user to initiate the login process. Upon clicking the login option, the system transitions to the Enter Credentials state, where the user provides their email and password or chooses to log in with Google or Facebook. The system then moves to the Validate Credentials state, where the provided information is checked for accuracy. If the credentials are valid, the system transitions to the Display Dashboard state, where the user is successfully logged in, and their personalized dashboard is shown. If the credentials are invalid, the system transitions to the Display Error state, prompting the user to try again.

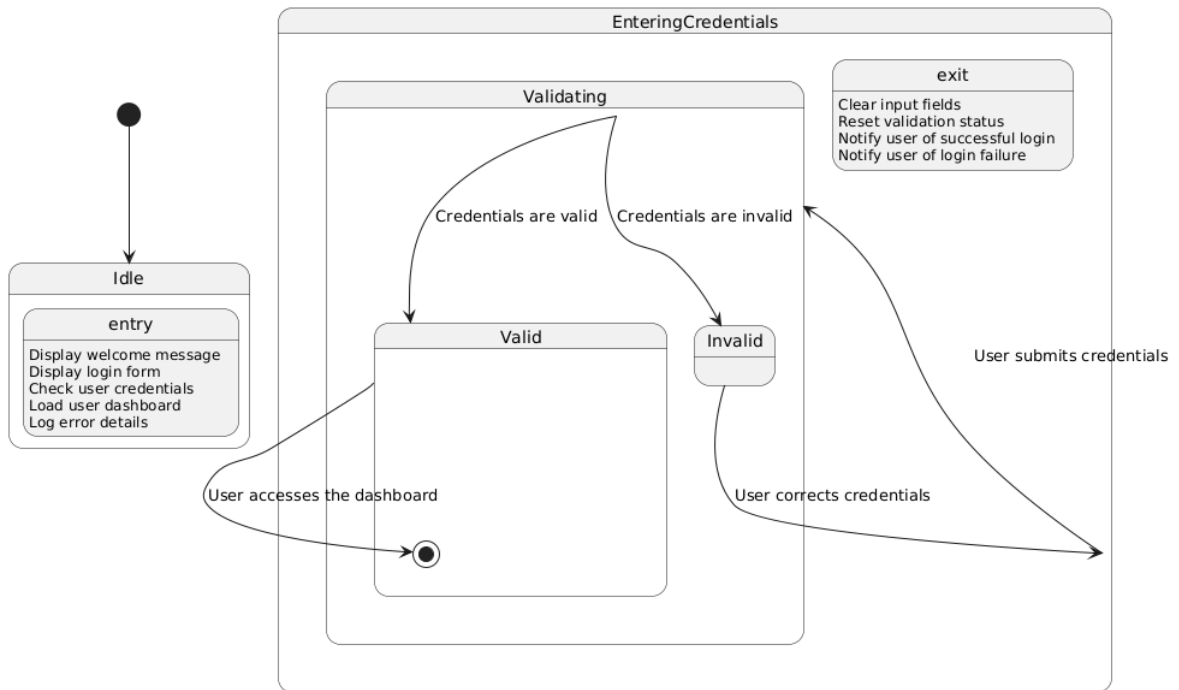


FIGURE 2.18: State Diagram: User Login

### 2.3.3 User Dashboard

The User History State Diagram (Figure 2.19) illustrates the various states the system transitions through when a user interacts with their past activity within the application. The process begins at the Idle state, where the system waits for the user to log in and access their dashboard. Upon selecting the option to view history, the system enters the Display User History state, where it retrieves and presents the user's previous interactions. From this state, the user has two main choices:

- **View Past Video URLs** – The system transitions to this state when the user chooses to view video links previously submitted for transcription and assessment.

- View Past Assessment Results – This state is activated when the user selects to view the results of assessments they have completed in the past.

Once the user has finished reviewing either their video history or assessment results, the system transitions back to the Dashboard state, allowing the user to continue other activities or log out.

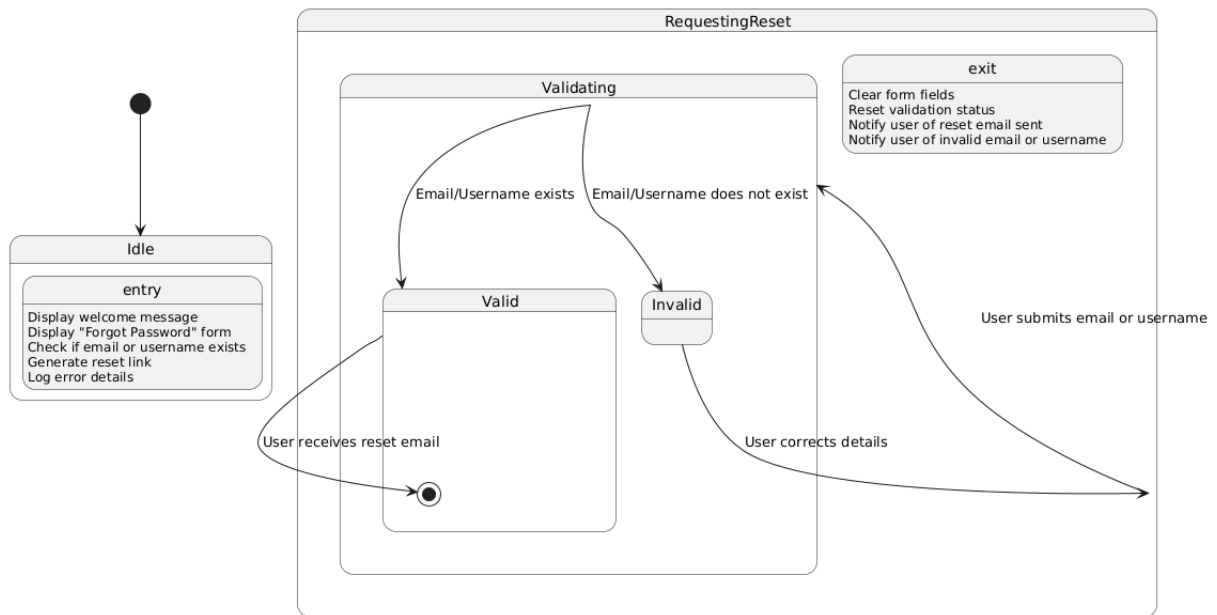


FIGURE 2.19: State Diagram: User Dashboard

### 2.3.4 Forget Password

The Forgot Password State Diagram (Figure 2.20) outlines the process that occurs when a user requests to reset their password. The flow begins in the Idle state, where the system awaits the user's input. Once the user selects the Forgot Password option, the system transitions to the Enter Email/Username state, prompting the user to provide their registered email or username. The system then moves to the Validate User Information state, where it checks if the entered email or username exists in the database:

- If the information is valid, the system proceeds to the Send Password Reset Email state, where a reset link is sent to the user's email.
- If the information is invalid, the system transitions to the Display Error Message state, notifying the user that the entered details are incorrect, and then returns to the Enter Email/Username state for re-entry.

Once the password reset email is sent, the process concludes, returning the system to the Idle state.

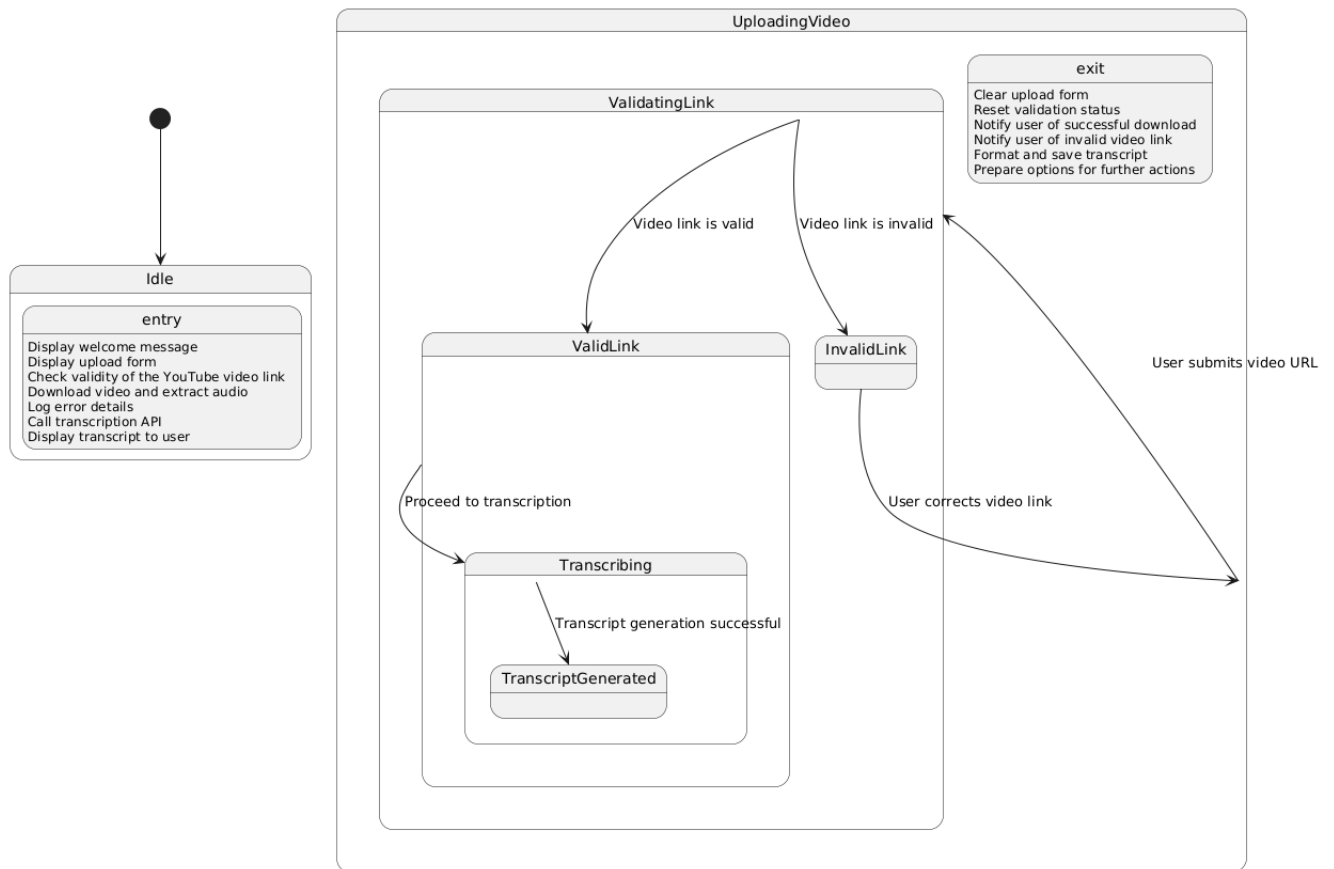


FIGURE 2.20: State Diagram: Forget Password

### 2.3.5 Link Input and Transcript Generation

The Link Input and Transcript Generation State Diagram (Figure 2.21) illustrates the steps the system goes through when a user uploads a video for transcription. The process begins at the Idle state, where the system waits for user input. Once the user selects to upload a video, the system transitions to the Enter YouTube Video URL state, where the user provides the video link. From there, the system enters the Validate URL state to check whether the provided URL is valid:

- If valid, the system transitions to the Download Video and Extract Audio state, where the video is downloaded, and its audio is extracted.
- If the URL is invalid, the system transitions to the Display Error Message state, notifying the user and allowing them to retry.

Once the audio is successfully extracted, the system moves to the Call Transcription API state, where the audio is processed and transcribed. The transcription

process marks the end of this stage, allowing the user to proceed to the next steps, such as viewing the summary, key concepts, or transcription.

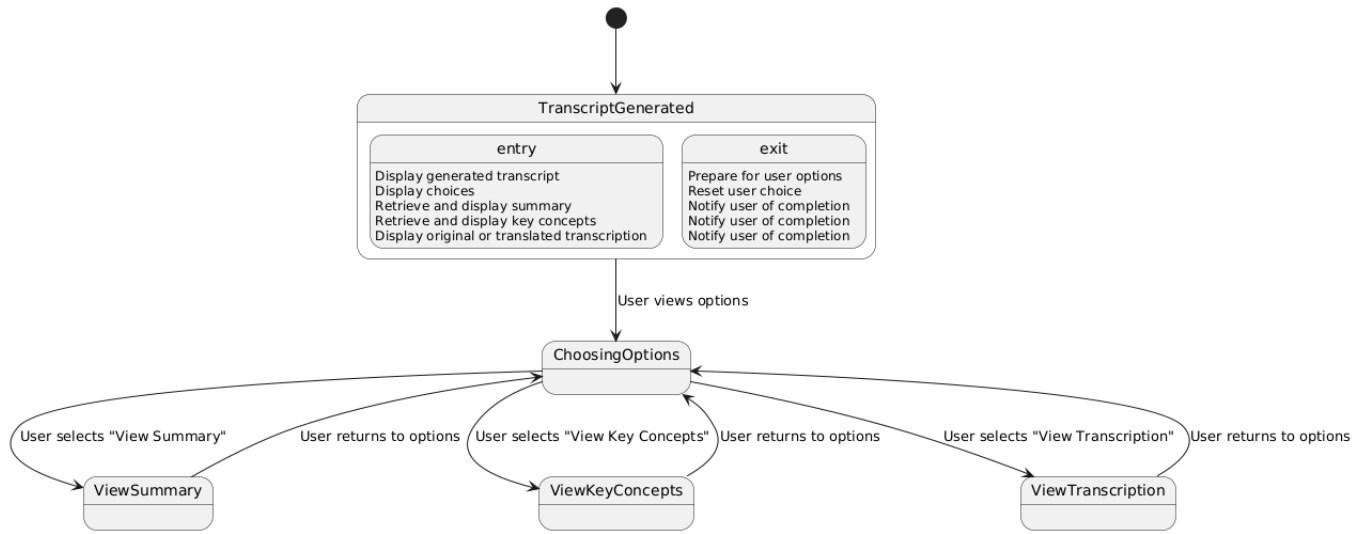


FIGURE 2.21: State Diagram: Link Input and Transcript Generation

### 2.3.6 User Choices after Transcription

The User Choices After Transcription State Diagram (Figure 2.22) outlines the possible actions a user can take after the transcription of a video has been successfully generated. The process starts from the Idle state, where the system waits for user interaction following the transcription process. Once the transcription is available, the system transitions to the Display User Choices state, where the user is presented with three options:

1. View Summary – The system transitions to this state if the user opts to see a concise summary of the transcribed content.
2. View Key Concepts – If the user chooses this option, the system moves to the state where it displays key topics and concepts extracted from the transcription.
3. View Transcription – This state allows the user to view the full transcription of the video.

If the user selects to view the transcription and the language of the transcription is Urdu or Hindi, an additional choice is provided to either view the Original Transcription or the Translated Transcription (translated into English).

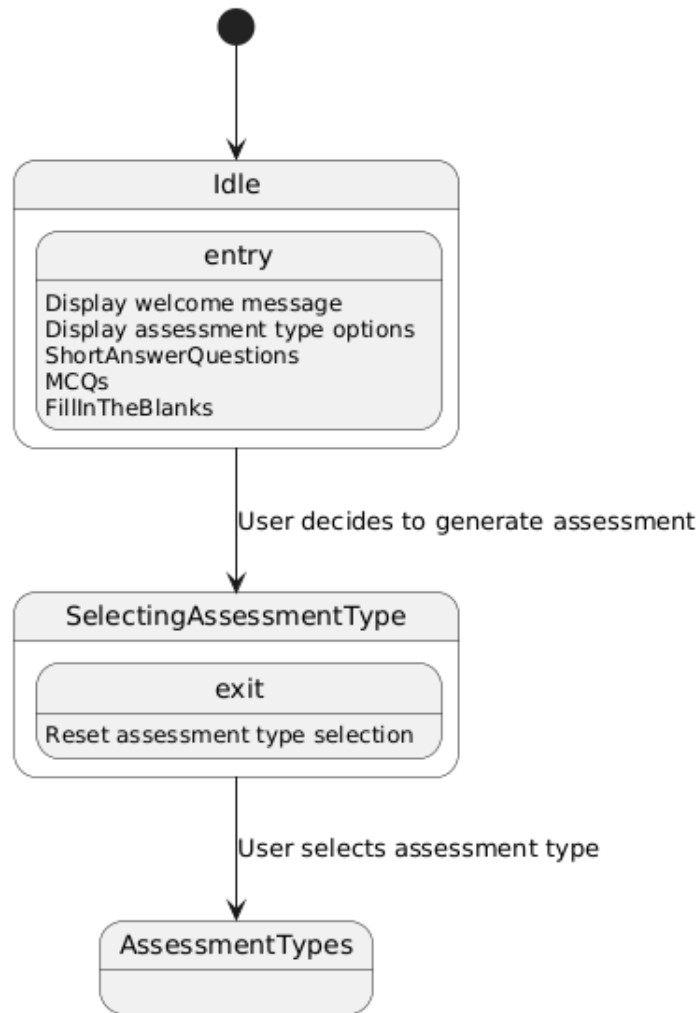


FIGURE 2.22: State Diagram: User Choices after Transcription

### 2.3.7 Assessment Generation

The Assessment Selection State Diagram (Figure 2.23) outlines the process a user goes through when choosing an assessment type after a video has been transcribed and processed. The diagram starts from the Idle state, where the system waits for user interaction following the generation of transcription, summary, or key concepts. Once the user chooses to take an assessment, the system transitions to the Display Assessment Options state. Here, the user is presented with several types of assessments:

1. Short Answer Questions – The system transitions to this state if the user selects this assessment type.
2. Multiple Choice Questions (MCQs) – The system moves to this state if the user prefers to answer MCQs.



3. Fill in the Blanks – This state is activated when the user selects to complete fill-in-the-blank questions.

After the user makes their selection, the system enters the Generate Selected Assessment state, where the chosen assessment is generated and displayed. The user can then begin interacting with the assessment in subsequent stages.

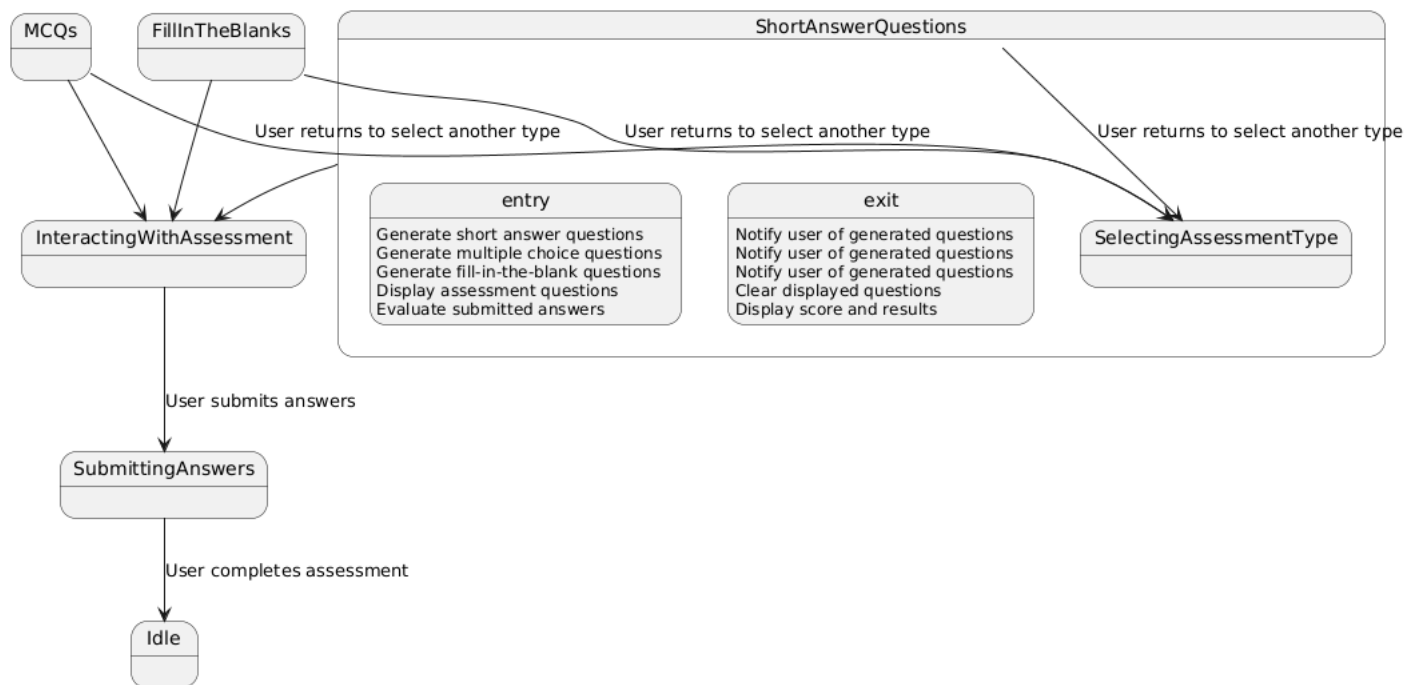


FIGURE 2.23: State Diagram: Assessment Generation-1

The Assessment Selection State Diagram (Figure 2.24) illustrates the sequence of actions a user goes through when interacting with and submitting an assessment after it has been generated. Starting from the Idle state, the system awaits user interaction with the generated assessment. When the user begins interacting, the system transitions to the User Interacts with Assessment state. In this state, the user answers questions, and they are given the option to:

- Answer the Question
- Skip the question without providing an answer

The system then transitions to the Submit Answers state once the user completes answering or skips the questions. At this point, the system processes all the answers in the Check Answered or Skipped Questions state, evaluating which questions were answered and which were skipped. Finally, the system moves to

the Display Score state, where the user's performance is shown based on their submitted answers.

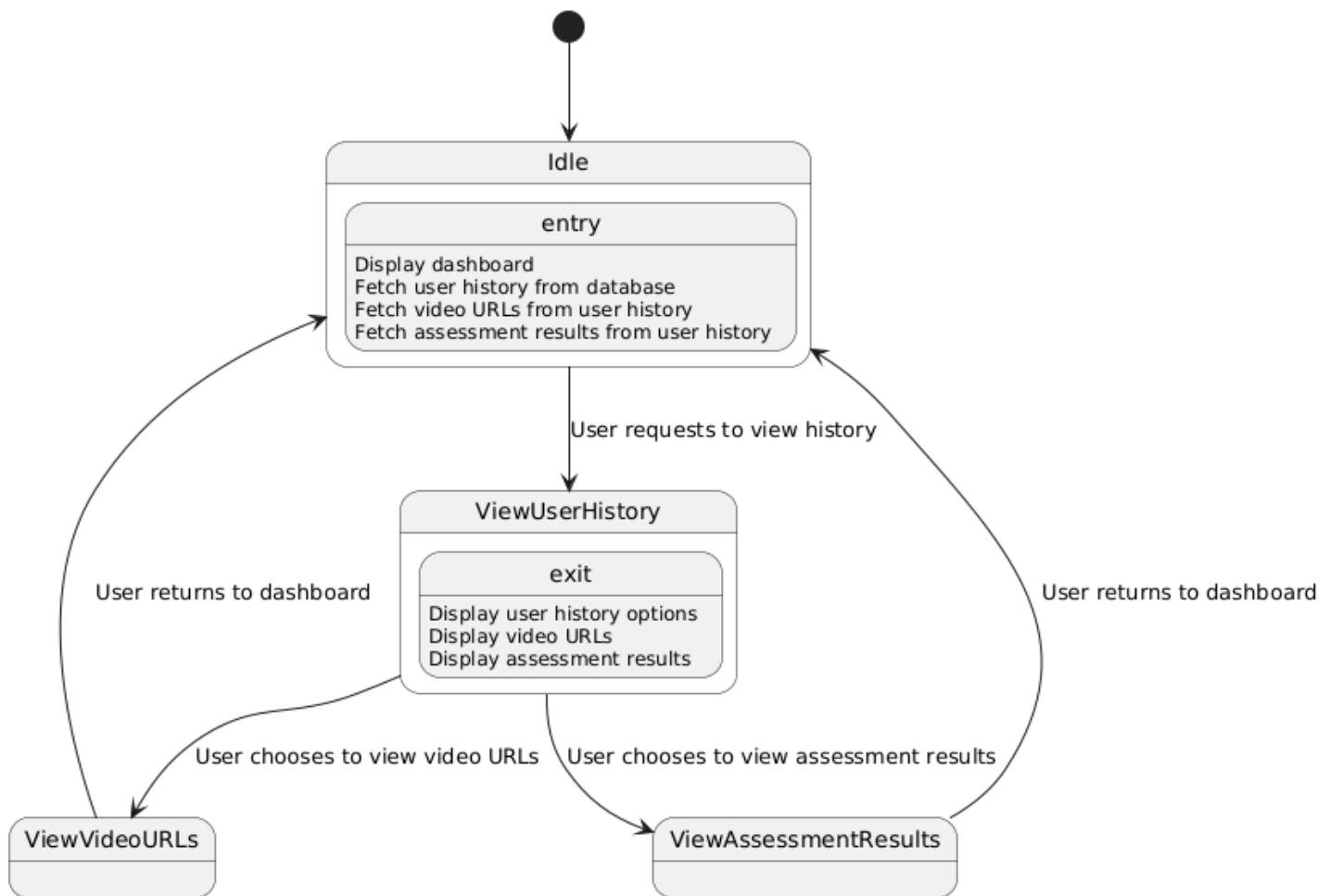


FIGURE 2.24: State Diagram: Assessment Generation-2

## 2.4 Sequence Diagram

In the context of this project, sequence diagrams visually depict how objects and entities within the system communicate to accomplish various tasks, such as video transcription, assessment generation, and user account management. By illustrating the sequence of messages exchanged between the user, system, and external APIs (such as YouTube or Whisper API), these diagrams provide a clear understanding of the order in which operations are executed. They help trace the flow of requests and responses, showing how the system processes user inputs, validates credentials, downloads videos, generates transcripts, and handles assessments.

### 2.4.1 User Registration

The User Registration sequence diagram (Figure 2.25) demonstrates the process of a user registering into the system. The user initiates the registration by entering their details such as username, email, and password, which are sent to the backend system. The backend system communicates with the database to verify if the user already exists. If no existing user is found, a new record is inserted into the database, and the registration is completed. Finally, a success message is sent back to the user.

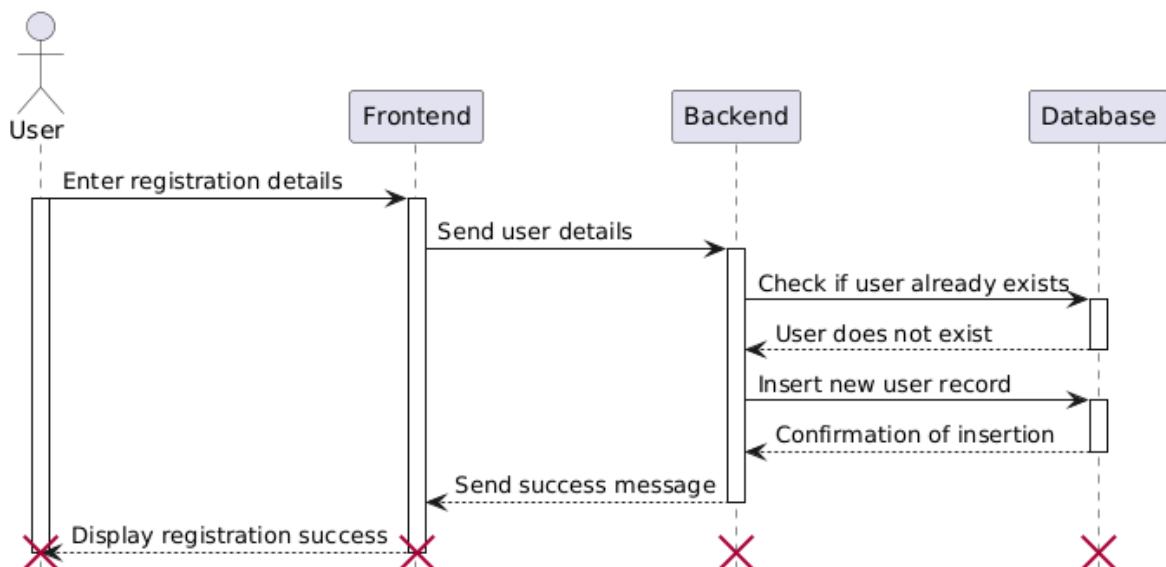


FIGURE 2.25: Sequence Diagram: User Registration

### 2.4.2 User Login

The User Login sequence diagram (Figure 2.26) illustrates the process of a user attempting to log into the system. The user initiates the login by sending their username and password to the system. The system then communicates with the

database to verify the credentials. Upon successful authentication, the system completes the login process and informs the user.

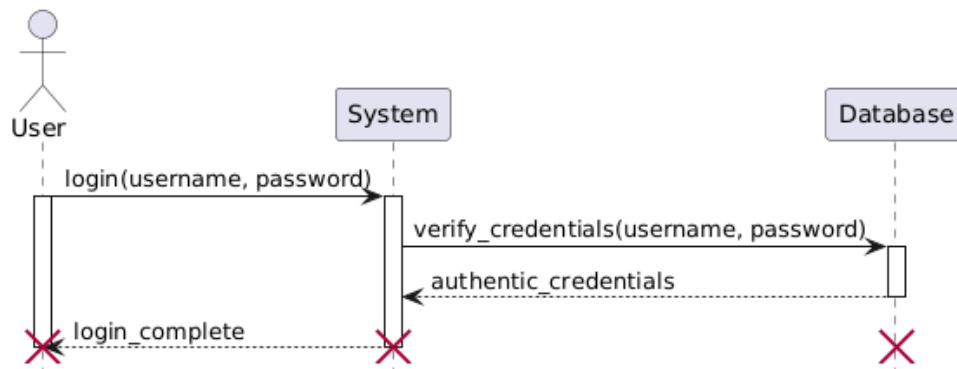


FIGURE 2.26: Sequence Diagram: User Login

### 2.4.3 Forget Password

The Forget Password Sequence Diagram (Figure 2.27) illustrates the process of resetting a user's password. The user initiates the process by requesting a password reset, providing their registered email address. The system checks if the email exists in the database. If valid, the system sends a reset link to the user's email through the email service. The user clicks the reset link, and the system verifies the token. Upon verification, the user is allowed to input a new password, which is updated in the database. A success message is then returned to the user upon completion.

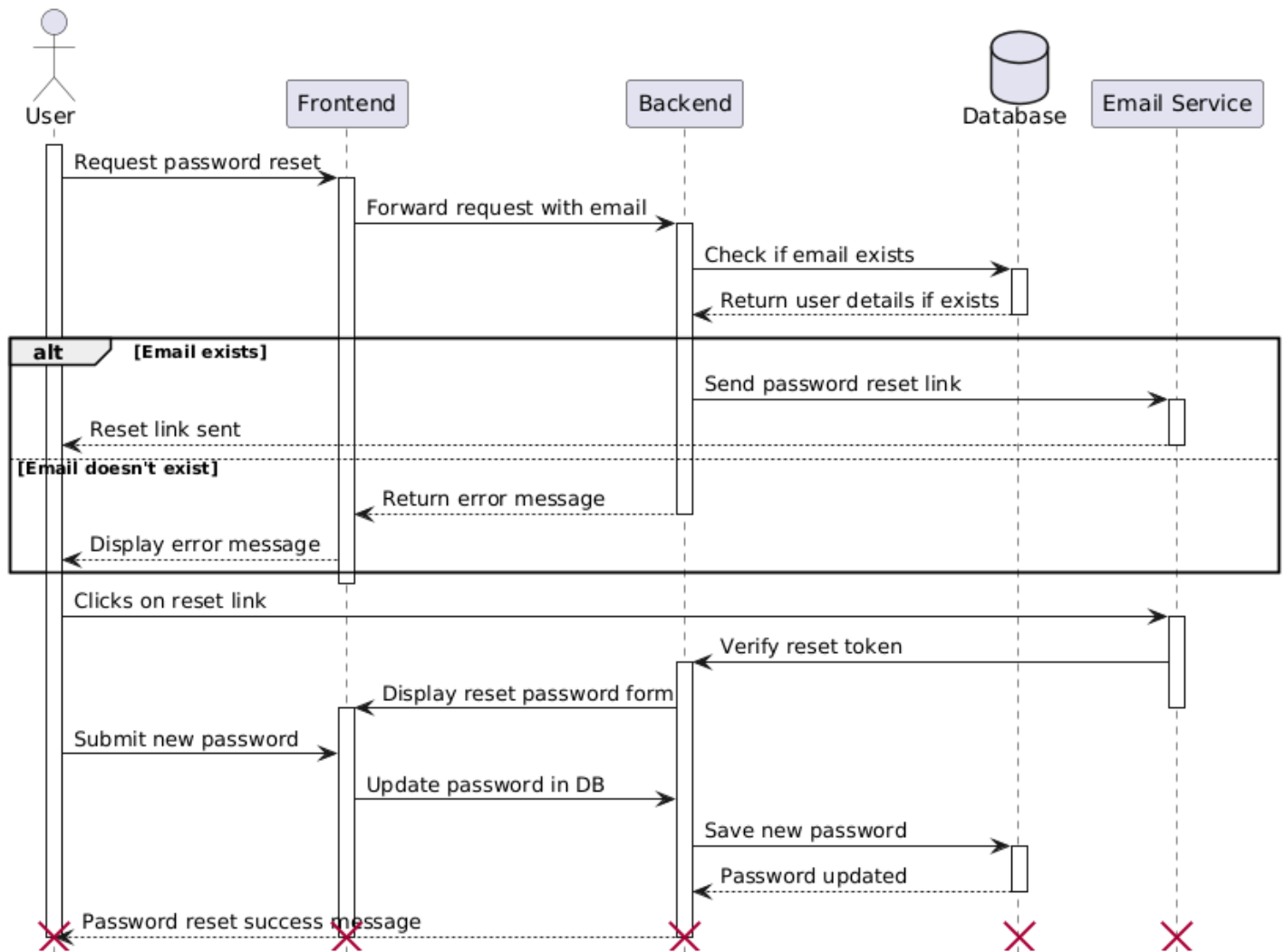


FIGURE 2.27: Sequence Diagram: Forget Password

#### 2.4.4 Video Upload

The Video Upload sequence diagram (Figure 2.28) demonstrates how a user uploads a video to the system. The user provides the video URL to the system, which then sets various attributes for the video, including its title, length, size, and thumbnail. Once the metadata is stored successfully, the system confirms the successful upload to the user.

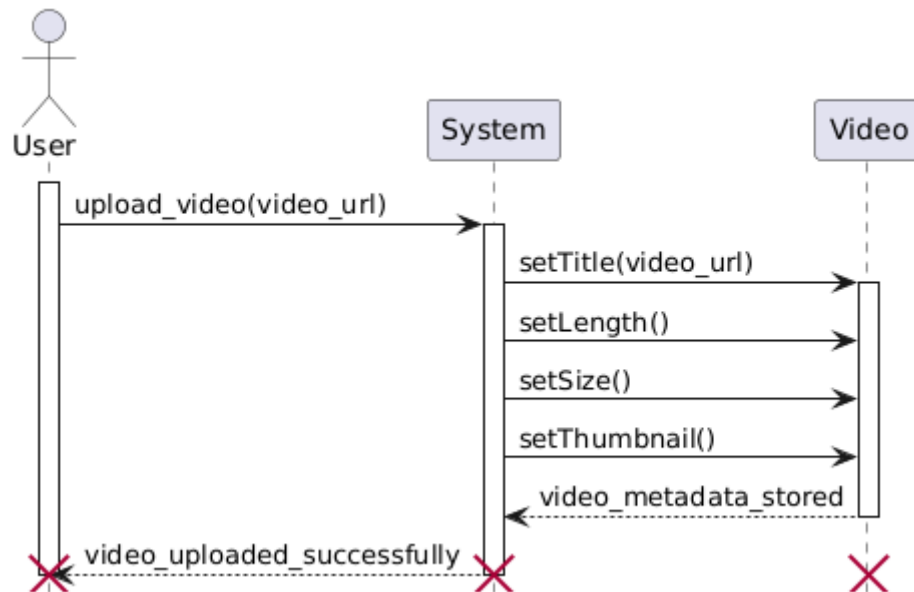


FIGURE 2.28: Sequence Diagram: Video Upload

### 2.4.5 Transcription Process

In the Transcription Process sequence diagram (Figure 2.29), the user requests the transcription of a video. The system downloads the audio from the provided YouTube link and sends a request to the Whisper API for transcription. The API processes the audio and returns the transcribed text to the system, which then displays it to the user.

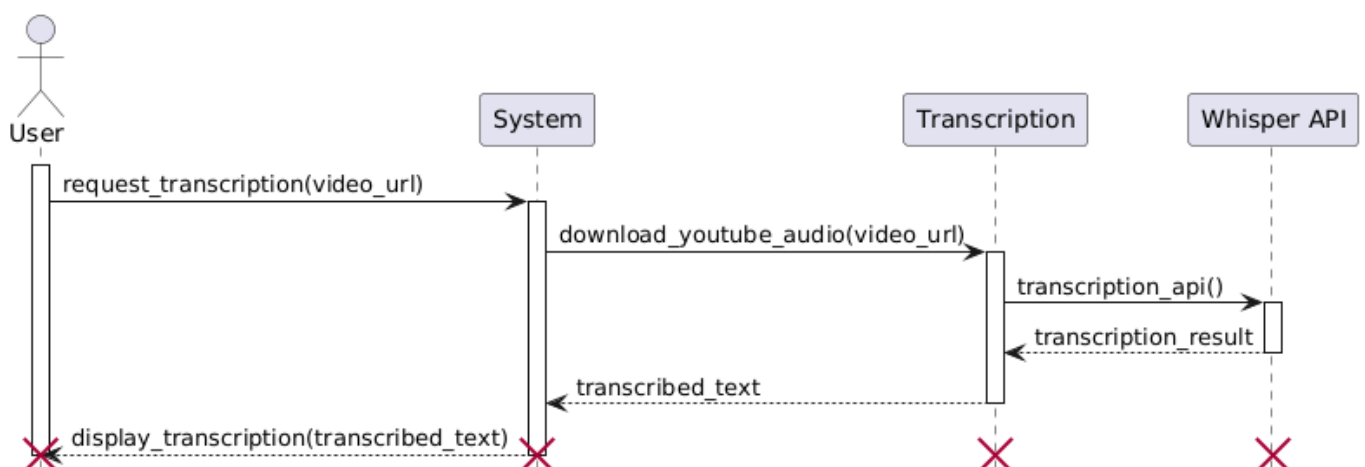


FIGURE 2.29: Sequence Diagram: Transcription Process

### 2.4.6 Assessment Generation

The Assessment Generation sequence diagram (Figure 2.30) outlines the steps taken when a user requests an assessment. The user specifies the type of assessment

they want. The system interacts with the Question class to generate the questions based on the specified type. Once the questions are generated, the system displays them to the user.

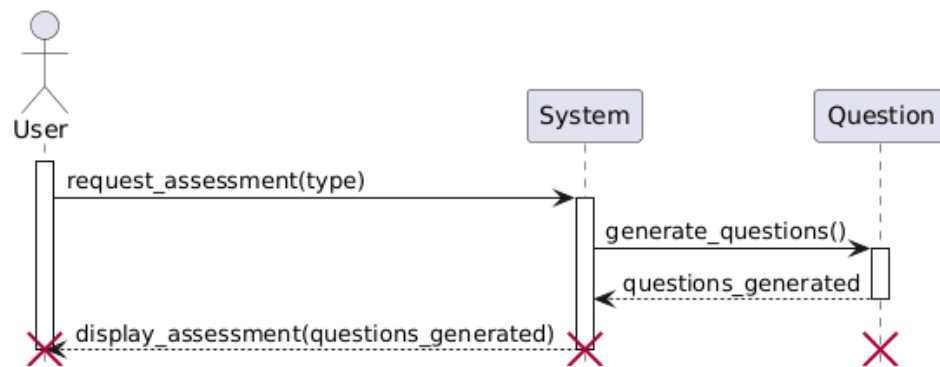


FIGURE 2.30: Sequence Diagram: Assessment Generation

### 2.4.7 Assessment Evaluation

The Assessment Evaluation sequence diagram (Figure 2.31) depicts the evaluation process after a user submits their answers to an assessment. The user sends their answers to the system, which then evaluates them using the Assessment class. The evaluation results are returned to the system, which subsequently displays them to the user.

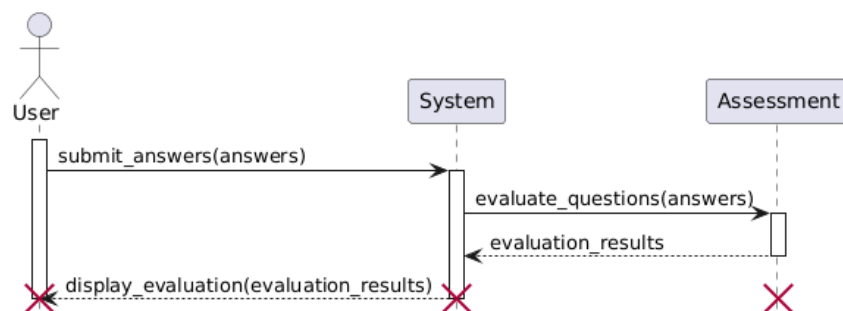


FIGURE 2.31: Sequence Diagram: Assessment Evaluation

## 2.5 Collaboration Diagram

In the context of this project, these diagrams highlight the structural organization of the system by showing relationships between components and how they work together to fulfill various functions, such as video transcription, assessment generation, and user interaction.

The collaboration diagram (Figure 2.32) illustrates the interactions and relationships among the various classes within the system. It serves as a comprehensive representation of how different components collaborate to achieve the functionality of the application.

- User interacts with the Video class to upload videos, which subsequently generates Transcription.
- The Transcription class communicates with the API to invoke necessary services for processing the uploaded videos.
- The Question class generates questions based on the transcribed content, while the Assessment class creates assessments from these questions.
- MediaProcessing serves as a central component that handles the interactions between videos, transcriptions, questions, and assessments, ensuring that each process is coordinated effectively.

This diagram provides a clear overview of the structural organization and the flow of data within the system, highlighting the collaborative nature of the classes involved in delivering the desired functionalities.



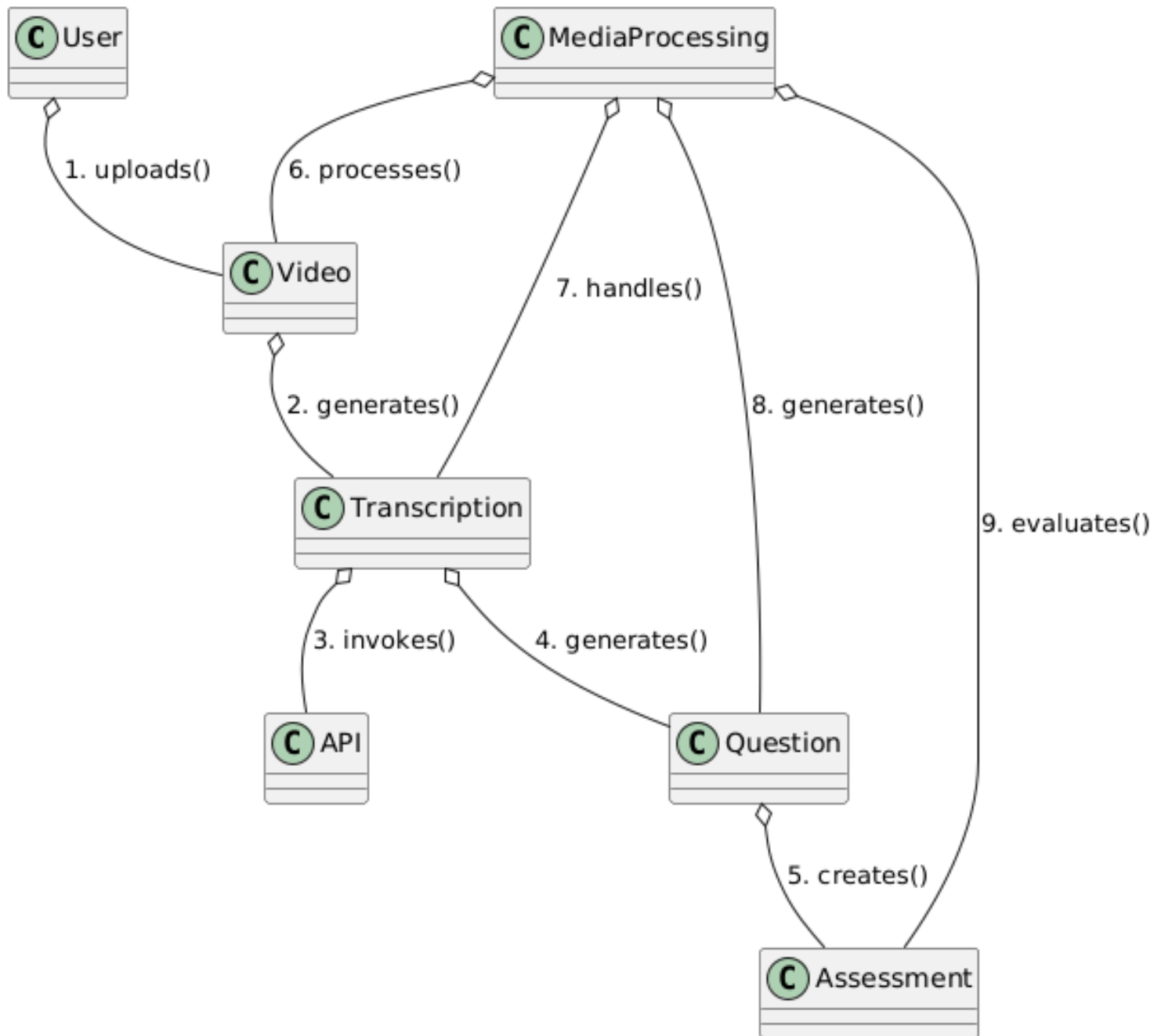


FIGURE 2.32: Collaboration Diagram

### 2.5.1 User Registration

The User Registration collaboration diagram (Figure 2.33) illustrates the interactions involved in the user registration process. It shows how the User class initiates the registration process by communicating with the API class to validate user information. Upon successful validation, the User class calls the User class methods to create a new account, showcasing the flow of data and control between these components during user registration.

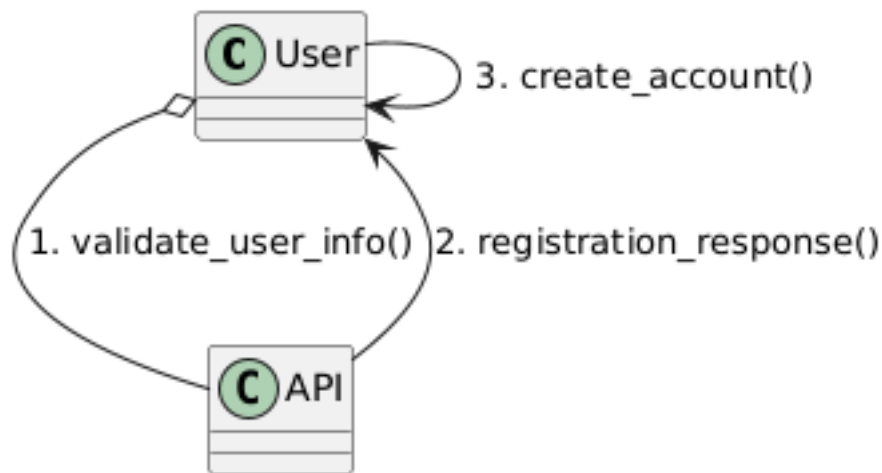


FIGURE 2.33: Collaboration Diagram: User Registration

### 2.5.2 User Login

The User Login collaboration diagram (Figure 2.34) depicts the sequence of interactions during the user login process. The User class triggers the login request, which is processed by the API class to authenticate the user. The diagram highlights how the system validates credentials and provides feedback to the user, emphasizing the importance of secure authentication in the application.

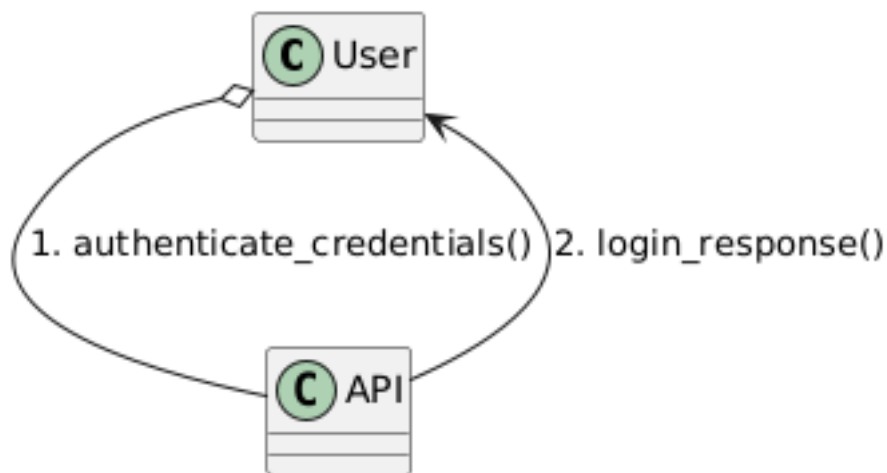


FIGURE 2.34: Collaboration Diagram: User Login

### 2.5.3 Forget Password

The Forget Password collaboration diagram (Figure 2.35) illustrates the interactions between the user, frontend, backend, database, and email service during the password reset process. The user submits a request to reset the password

through the frontend interface. The backend verifies the user's email by checking the database. If the email exists, a password reset link is sent via the email service. The user clicks the link to reset their password, and the backend verifies the token, displays a password reset form, and updates the user's password in the database upon submission.

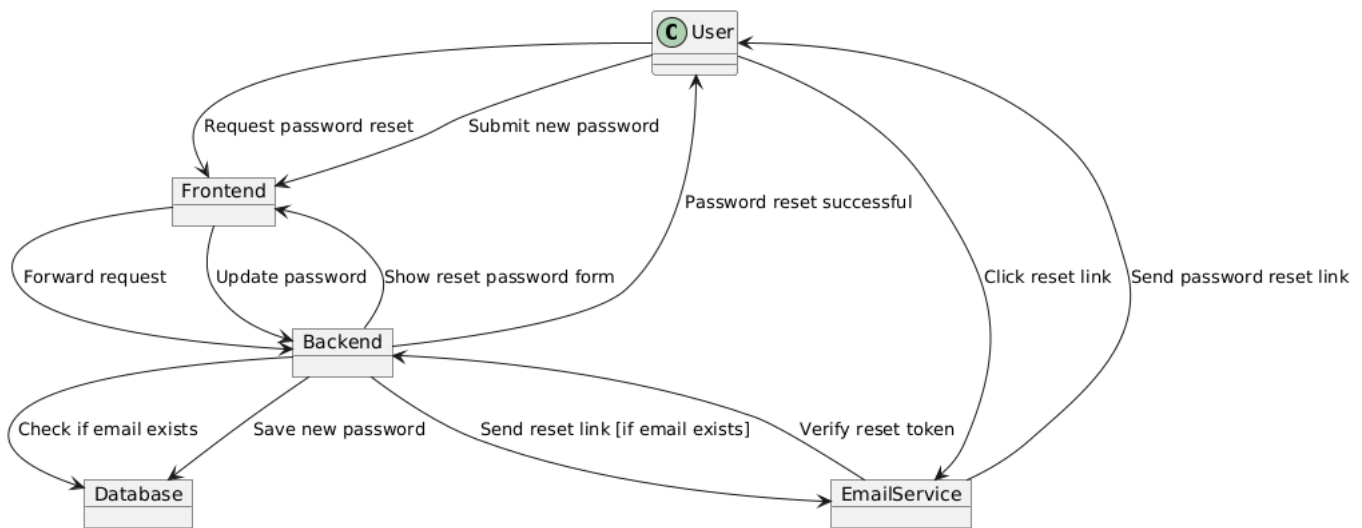


FIGURE 2.35: Collaboration Diagram: User Login

### 2.5.4 Upload Video

The Upload Video collaboration diagram (Figure 2.36) outlines the process of uploading a video. The User class interacts with the MediaProcessing class to upload the video URL, which then communicates with the Video class to store video metadata. This collaboration emphasizes the flow of data from the user to the system, highlighting the responsibilities of each class in managing video uploads.

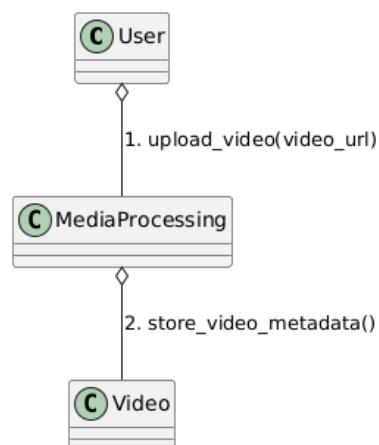


FIGURE 2.36: Collaboration Diagram: Upload Video

### 2.5.5 Transcription

The Transcription collaboration diagram (Figure 2.37) illustrates the interactions involved in requesting a transcription. The User initiates the process by calling methods from the Transcription class, which in turn interacts with the API class to perform the transcription via the external service. This collaboration showcases the integration of external APIs to enhance functionality.

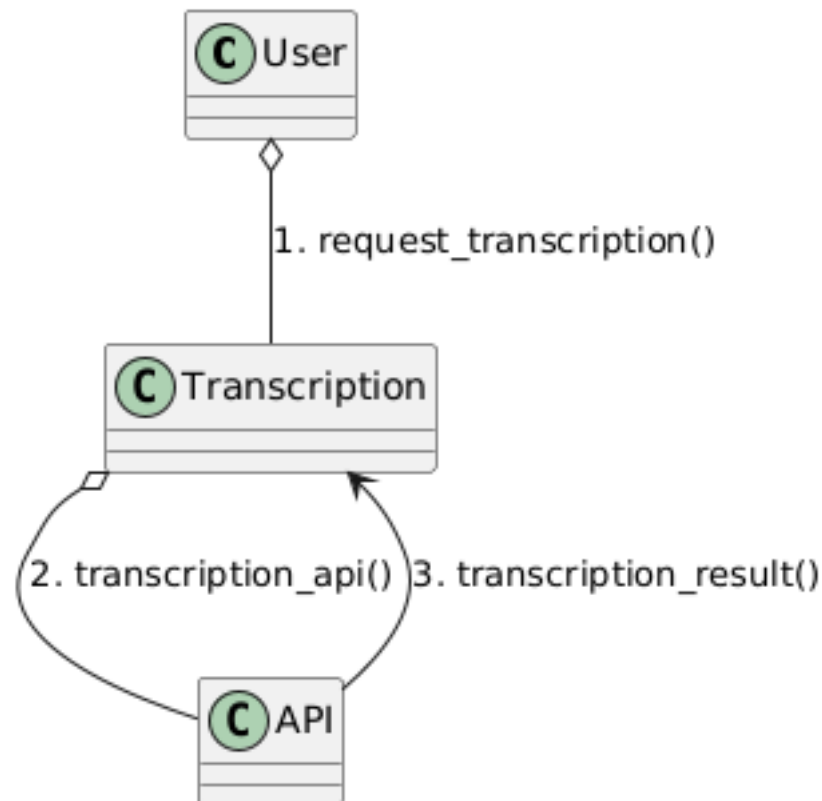


FIGURE 2.37: Collaboration Diagram: Transcription

### 2.5.6 Assessment Generation

The Assessment Generation collaboration diagram (Figure 2.38) depicts how questions are generated based on transcriptions. The MediaProcessing class communicates with the Question class to generate various types of questions from the transcription data. The interactions highlight how the system derives meaningful assessments from the transcribed content, demonstrating the educational focus of the application.

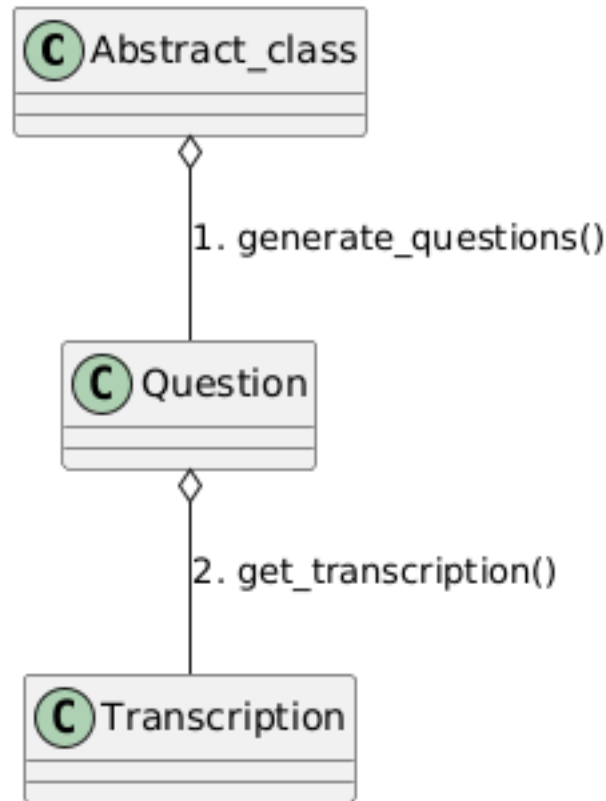


FIGURE 2.38: Collaboration Diagram: Assessment Generation

### 2.5.7 Assessment Evaluation

The Assessment Evaluation collaboration diagram (Figure 2.39) illustrates the process of creating and evaluating assessments. The User interacts with the Assessment class to submit answers, which are then evaluated by the system. This collaboration emphasizes the flow of user input and the subsequent feedback mechanism, showcasing how assessments are integrated into the user experience.

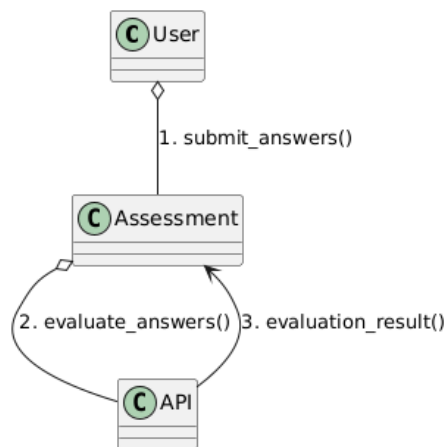


FIGURE 2.39: Collaboration Diagram: Assessment Evaluation

## 2.6 Class Diagram

In the context of this project, the class diagram represents the key components involved in managing user accounts, video transcription, and assessment generation, highlighting their attributes, methods, and the interactions between them.

The class diagram for this project captures essential entities like **User**, **Video**, **Transcription**, **Questions**, and **Assessment**, along with their respective properties and behaviors. It defines how these classes interact with each other—such as how a user uploads a video, the system processes the transcription, and questions are generated for assessments based on the transcribed content. The diagram also illustrates important associations, like the relationship between users and their uploaded videos, and how assessments are linked to transcriptions.

The Class Diagram (Figure 2.40) provides a comprehensive overview of the structure and organization of the system's components, detailing the various classes involved in the application. It outlines key classes such as **User**, **Video**, **Transcription**, **Question**, **Assessment**, **MediaProcessing**, and **API**, each with its attributes and methods.

- **User** class manages user-related functionalities, including account creation, login, and video uploads.
- **Video** class encapsulates information related to video metadata such as title, length, size, and URLs.
- **Transcription** class handles the transcription process, downloading audio, and managing transcribed text.
- **Question** class is responsible for generating various types of questions based on transcriptions and related data.
- **Assessment** class deals with user assessments, including evaluation and result management.
- **MediaProcessing** class integrates the various media-related functionalities, acting as a central point for processing video content.
- **VideoProcessingAPI** class defines the external APIs used in the system, facilitating interactions with services such as transcription and question generation.

- **LLMProcessing** class defines the Lama Models used in the system, facilitating interactions with services such as transcription and question generation.

The relationships among these classes, including associations and dependencies, illustrate how they interact to achieve the system's functionality. This class diagram serves as a foundational blueprint for the development of the application, guiding the implementation and integration of its components. The classes colored as Blue color are related to the frontend of the system while the classes colored as Green color are related to the backend precesses of the system.

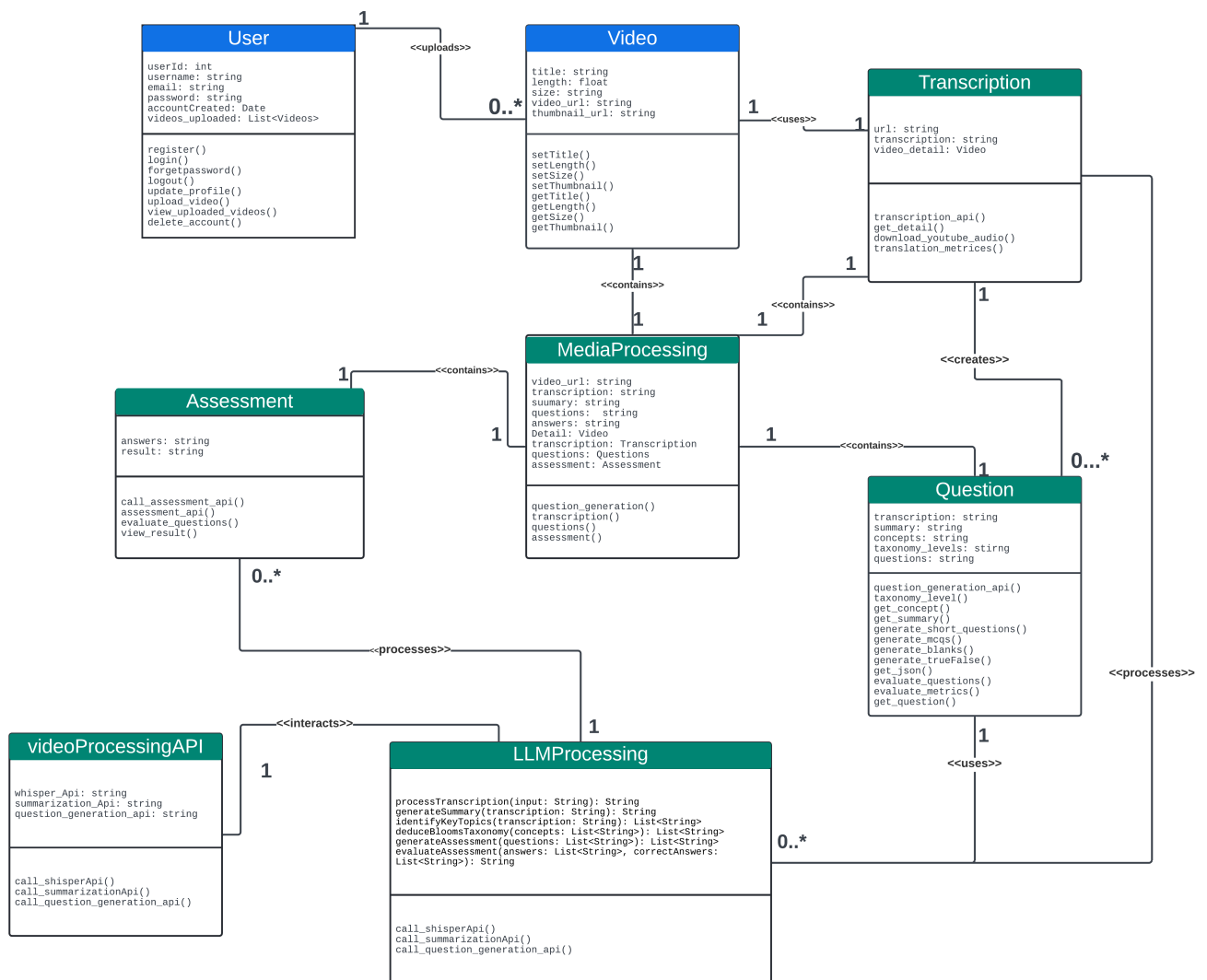


FIGURE 2.40: Class Diagram

## 2.7 Component Diagram

Component diagram provide a high-level view of the system's physical structure by showing how different software components are organized and interact with each other. In the context of this project, the component diagram illustrates the major system components, such as the user interface, back-end services, APIs, and databases, and how they collaborate to deliver key functionalities like video transcription, assessment generation, and user management.

The Component Diagram (Figure 2.41) illustrates the architecture of a system designed to process video content and generate assessments based on transcriptions. The diagram is organized into distinct packages representing the Frontend, Backend, External APIs, and Database, highlighting the modular structure and interactions among the components.

### 1. Frontend Package

- **User Interface:** This component represents the graphical interface through which users interact with the system. It handles user inputs, displays content, and communicates with the backend services via HTTP requests.

### 2. Backend Package

- **API Gateway:** Acts as the central entry point for all client requests. It routes requests to the appropriate backend services and orchestrates interactions between them.
- **User Service:** Manages user-related functionalities, including registration, login, and profile management.
- **Video Processing Service:** Handles video uploads, validates URLs, and interacts with external APIs to fetch video content.
- **Transcription Service:** Responsible for generating transcriptions from audio extracted from videos. It utilizes external APIs for transcription and summarization.
- **Assessment Service:** Generates assessments based on transcriptions and user preferences, interacting with external question generation APIs as needed.
- **Database Service:** Manages data storage and retrieval, interfacing with the MongoDB database to maintain user information, video details, transcriptions, and assessments.



### 3. External APIs Package

- YouTube API: Used by the Video Processing Service to fetch video data and content from YouTube.
- Whisper API: This transcription service converts audio files into text, allowing for accurate transcriptions of the video's audio content.
- Summarization API: Utilized by the Transcription Service to create concise summaries of the transcribed text.
- Question Generation API: Engaged by the Assessment Service to generate various types of questions based on the transcribed content.

### 4. Database Package

- MongoDB: The database component where all user data, uploaded video links, transcriptions, assessments, and related information are stored and managed.

**Interactions:** The diagram illustrates the relationships and data flow between these components:

- The User Interface sends HTTP requests to the API Gateway, which then routes these requests to the appropriate backend services.
- The API Gateway interacts with several services, including the User Service, Video Processing Service, Transcription Service, Assessment Service, and Database Service to fulfill user requests.
- The Video Processing Service communicates with the YouTube API to fetch video content.
- The Transcription Service sends requests to the Whisper API for transcribing audio and to the Summarization API for summarizing the transcriptions.
- The Assessment Service generates questions by interfacing with the Question Generation API.
- The Database Service manages data storage and retrieval from MongoDB, ensuring that all information is persistently maintained.

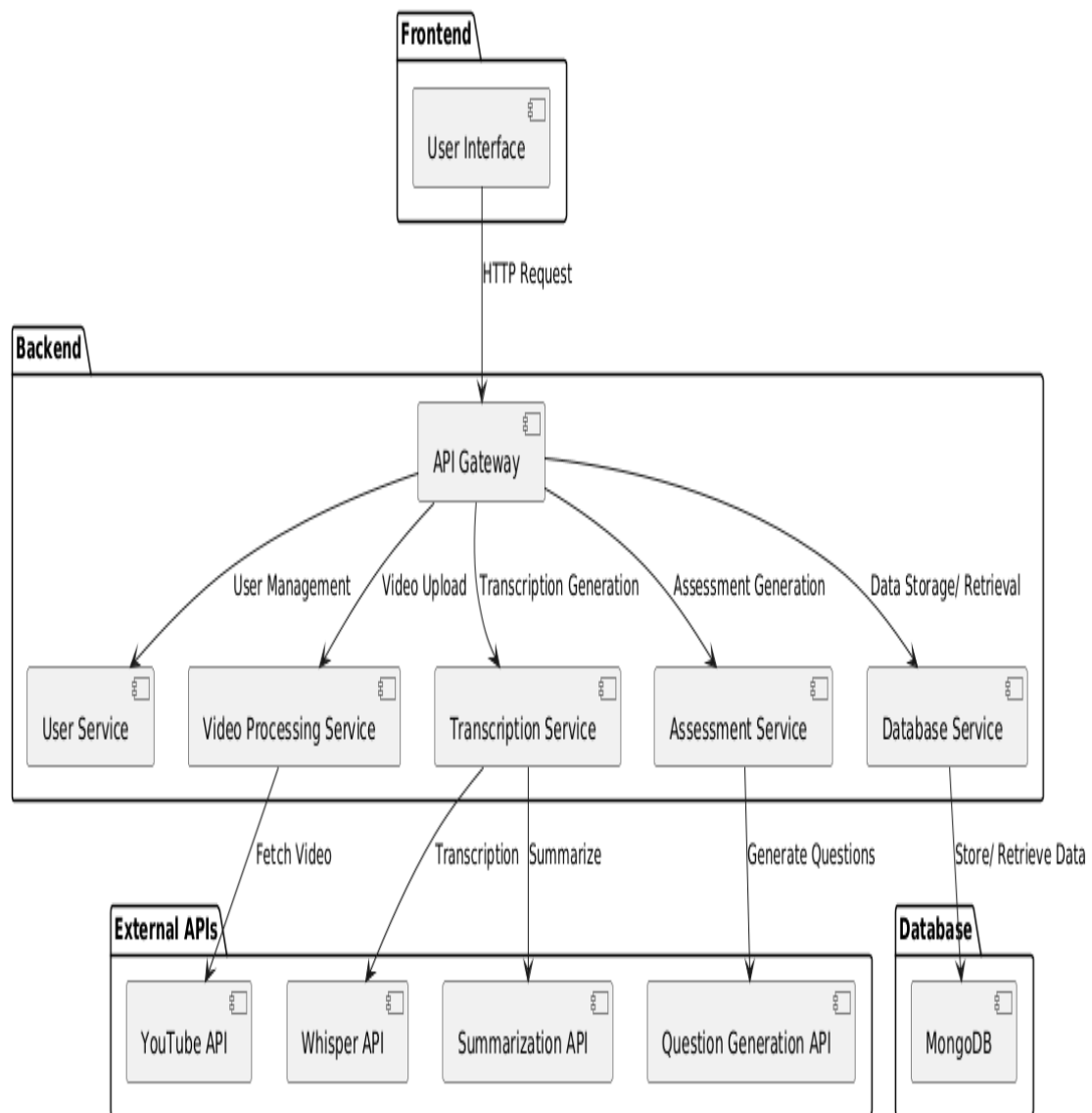


FIGURE 2.41: Component Diagram

### 2.7.1 User Authentication Module

The User Authentication Module diagram (Figure 2.42) outlines the interactions between the Frontend, Backend, and Database systems involved in managing user authentication and registration processes.

#### 1. Frontend Package

- Login Page: Represents the interface where users input their login credentials.
- Registration Page: The page where users can create a new account by providing their registration details.

- User Dashboard: The personalized dashboard available to users after successful login, where they can perform various actions.

## 2. Backend Package

- API Gateway: Serves as the intermediary for all communication between the frontend and backend services. It handles login, registration requests, and user actions from the dashboard by routing them to the appropriate backend services.
- User Service: Manages user-related processes, such as authentication, registration, and user data handling. This service validates the user's credentials during login and stores or retrieves user data during registration and authentication.

## 3. Database Package

- MongoDB: Acts as the persistent storage system, where user information such as credentials, registration details, and account data are stored and validated.

## Interactions

- The Login Page sends a Login Request to the API Gateway, which forwards it to the User Service for authentication. The User Service interacts with MongoDB to validate the user's credentials.
- The Registration Page sends a Registration Request to the API Gateway, which passes it to the User Service. The User Service stores the new user's information in MongoDB.
- Once the user is authenticated, actions on the User Dashboard are sent to the API Gateway, which may involve further backend processing.

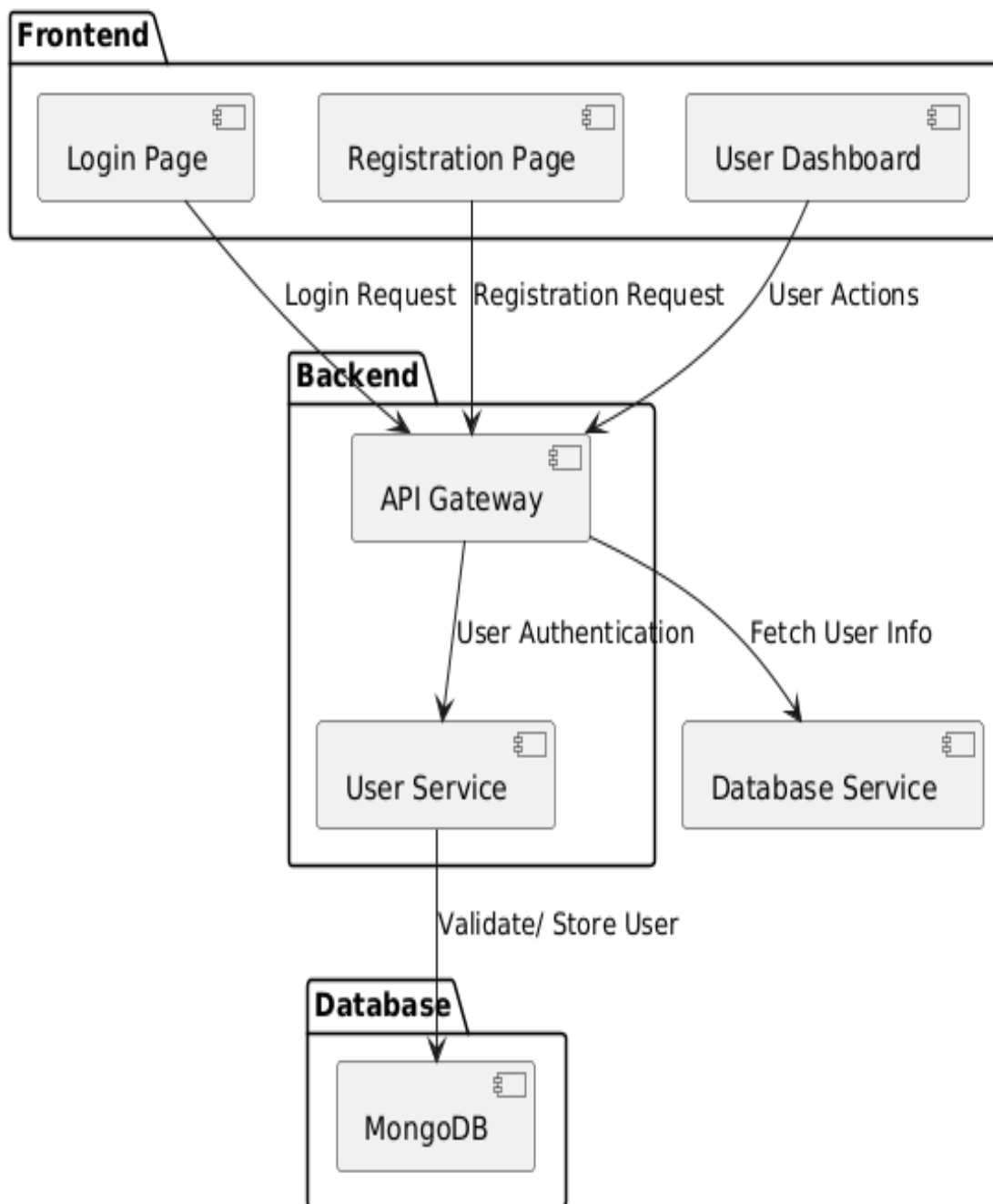


FIGURE 2.42: Component Diagram: User Authentication Module

### 2.7.2 Video Processing and Transcription Module

The Video Processing and Transcription Module diagram (Figure 2.43) illustrates the process flow for video uploading, transcription generation, and display within the system, showing the interactions between the Frontend, Backend, and External APIs components.

1. Frontend Package

- Video Upload Page: The user interface where users input and submit a YouTube video URL for transcription.
- Transcription Display: The interface where users can view the transcription and summarized content of the video.

## 2. Backend Package

- API Gateway: Acts as the central hub for all requests from the frontend to the backend services. It routes the video upload, processing, and transcription requests to the appropriate backend services.
- Video Processing Service: Responsible for handling the video URL submitted by the user. It processes the video and interacts with external APIs to fetch the video details.
- Transcription Service: Handles the transcription of the video's audio, sending requests to external APIs for audio transcription and text summarization.

## 3. External APIs Package

- YouTube API: Used by the Video Processing Service to fetch the video based on the URL provided by the user.
- Whisper API: Called by the Transcription Service to convert the audio into a transcription.
- Summarization API: Called by the Transcription Service to generate a concise summary of the transcribed content.

## Interactions

- The user uploads a video URL through the Video Upload Page, which sends a request to the API Gateway.
- The API Gateway forwards this request to the Video Processing Service, which then uses the YouTube API to fetch the video.
- Once the video is fetched, the API Gateway sends a transcription request to the Transcription Service, which calls the Whisper API to transcribe the audio into text.
- The Transcription Service also interacts with the Summarization API to generate a summary of the transcribed content.

- Finally, the transcription and summary are sent back to the Transcription Display for the user to view.

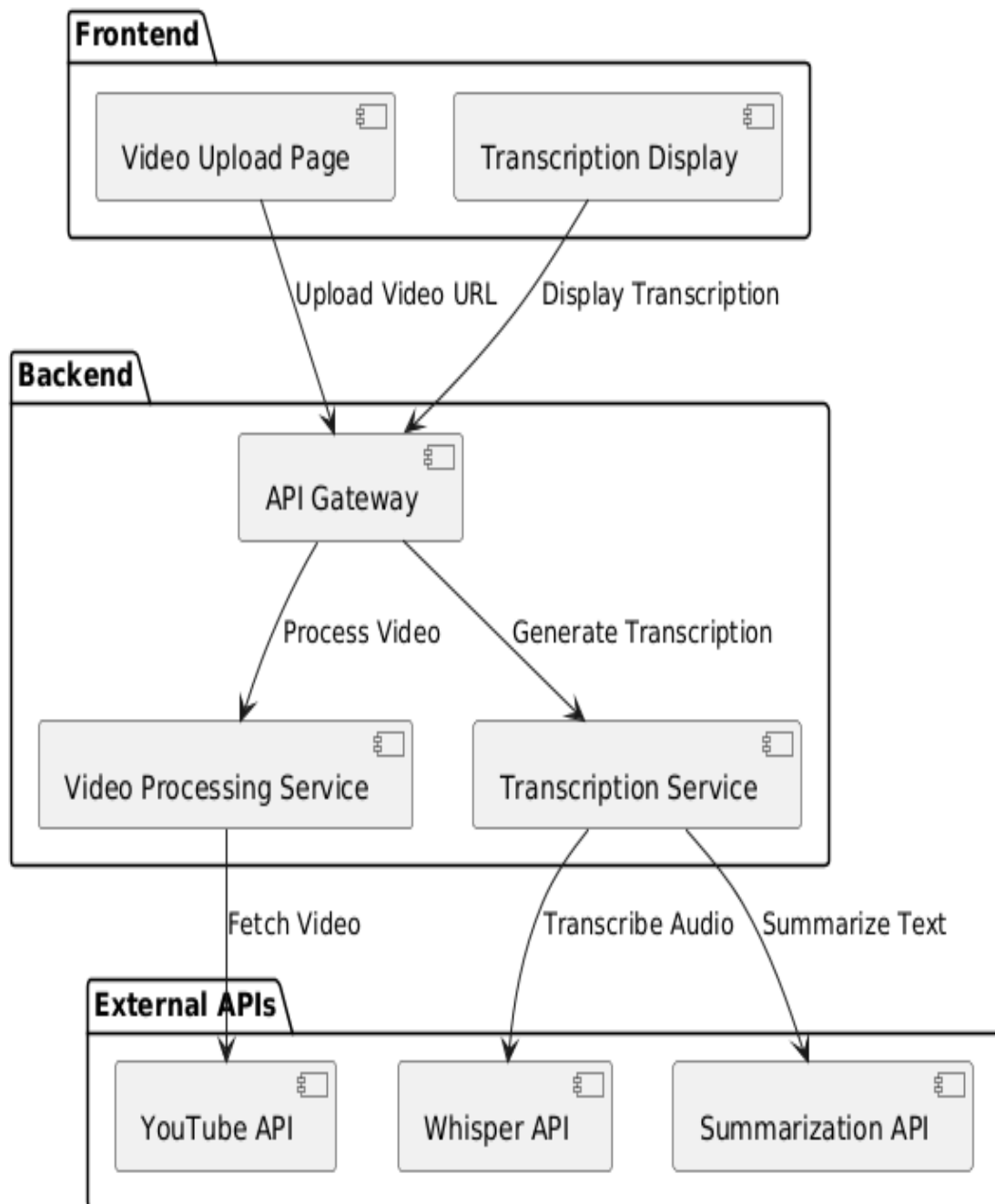


FIGURE 2.43: Component Diagram: Video Processing and Transcription Module

### 2.7.3 Assessment Generation Module

The Assessment Generation Module diagram (Figure 2.44) outlines the process of generating assessments, fetching questions, and displaying results in the system.

It shows the interactions between the Frontend, Backend, External APIs, and Database components.

### 1. Frontend Package

- **Assessment Page:** The user interface where the user initiates the request for an assessment based on the transcribed video content.
- **Assessment Results Page:** The page where users view the results of their completed assessments.

### 2. Backend Package

- **API Gateway:** Acts as a central point for communication between the frontend and backend services, managing requests related to assessment generation and results display.
- **Assessment Service:** Handles the logic of generating assessment questions and saving the results once the user has completed the assessment.

### 3. External APIs Package

- **Question Generation API:** This external API is used by the Assessment Service to fetch relevant questions for the assessment, which may include short answer questions, MCQs, or fill-in-the-blank types.

### 4. Database Package

- **MongoDB:** The database where user data and assessment results are stored.
- **Database Service:** The service responsible for storing and retrieving data from MongoDB.

## Interactions

- The user begins the process by requesting an assessment through the Assessment Page, which sends a request to the API Gateway.
- The API Gateway forwards this request to the Assessment Service, which generates the assessment questions.
- The Assessment Service communicates with the Question Generation API to fetch questions related to the video transcript.

- After the assessment is completed, the Assessment Service sends the results to the Database Service, which stores them in MongoDB.
- The user can view the assessment results through the Assessment Results Page, which retrieves the data via the API Gateway.

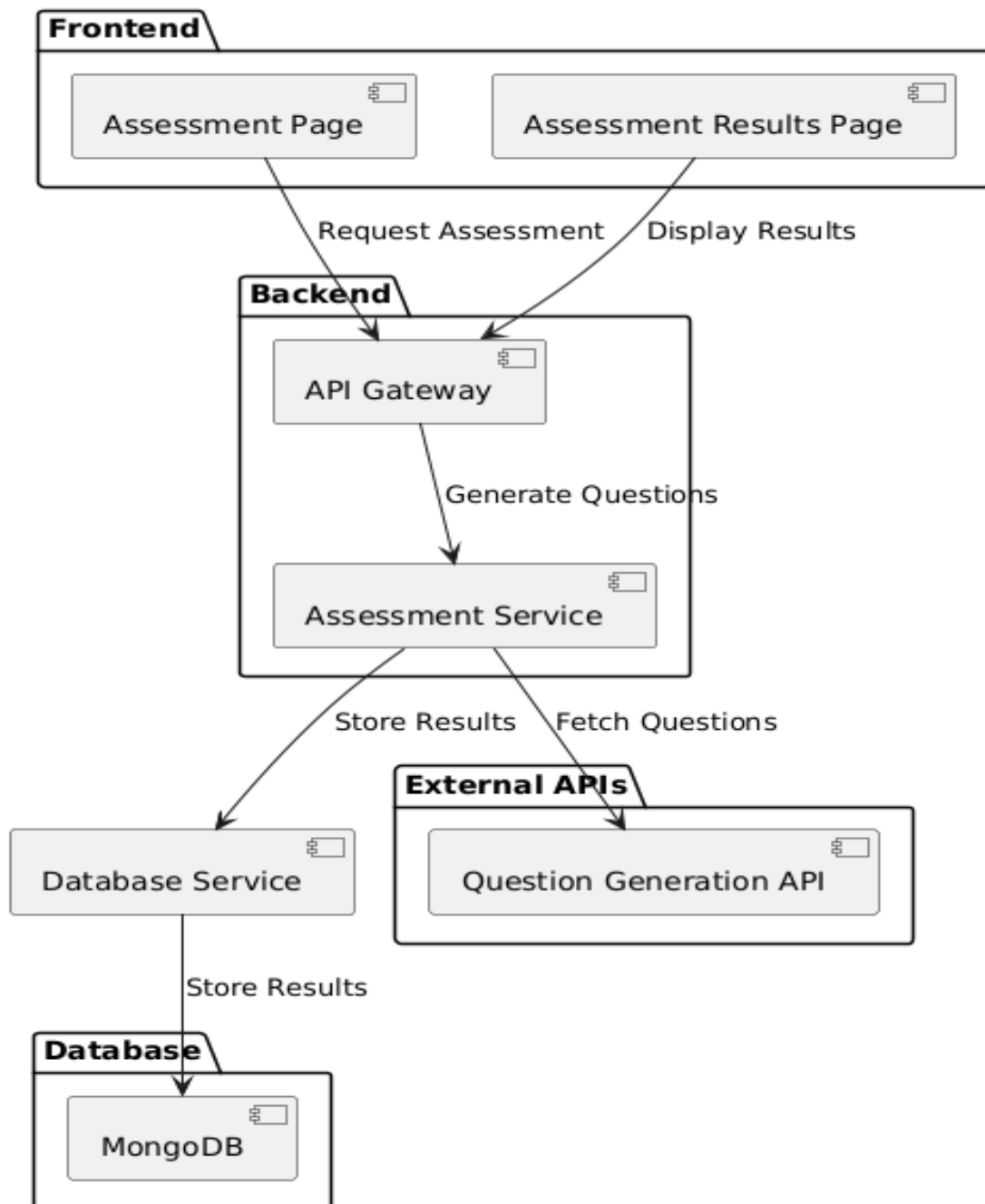


FIGURE 2.44: Component Diagram: Assessment Generation Module



### 2.7.4 Database Service Module

The Database Service Module diagram (Figure 2.45) represents the interaction between the Backend and Database in the process of fetching and retrieving data. It highlights the flow of data requests from the backend and how the database service processes those requests.

#### 1. Backend Package

- **API Gateway:** The central communication hub that handles requests from different parts of the system, such as the frontend or other backend services. In this case, it sends data requests to the Database Service.
- **Database Service:** Responsible for interfacing with the database to retrieve the requested data. It manages communication with the database to store, fetch, and return data.

#### 2. Database Package

- **MongoDB:** The database that stores all relevant data, such as user information, video data, transcription details, and assessments.

### Interactions

- The API Gateway sends a data request to the Database Service whenever the system needs to retrieve stored information, such as user history or video links.
- The Database Service then communicates with MongoDB to fetch the required data.
- Once the data is fetched from MongoDB, the Database Service returns the retrieved data back to the API Gateway for further processing or delivery to the frontend.

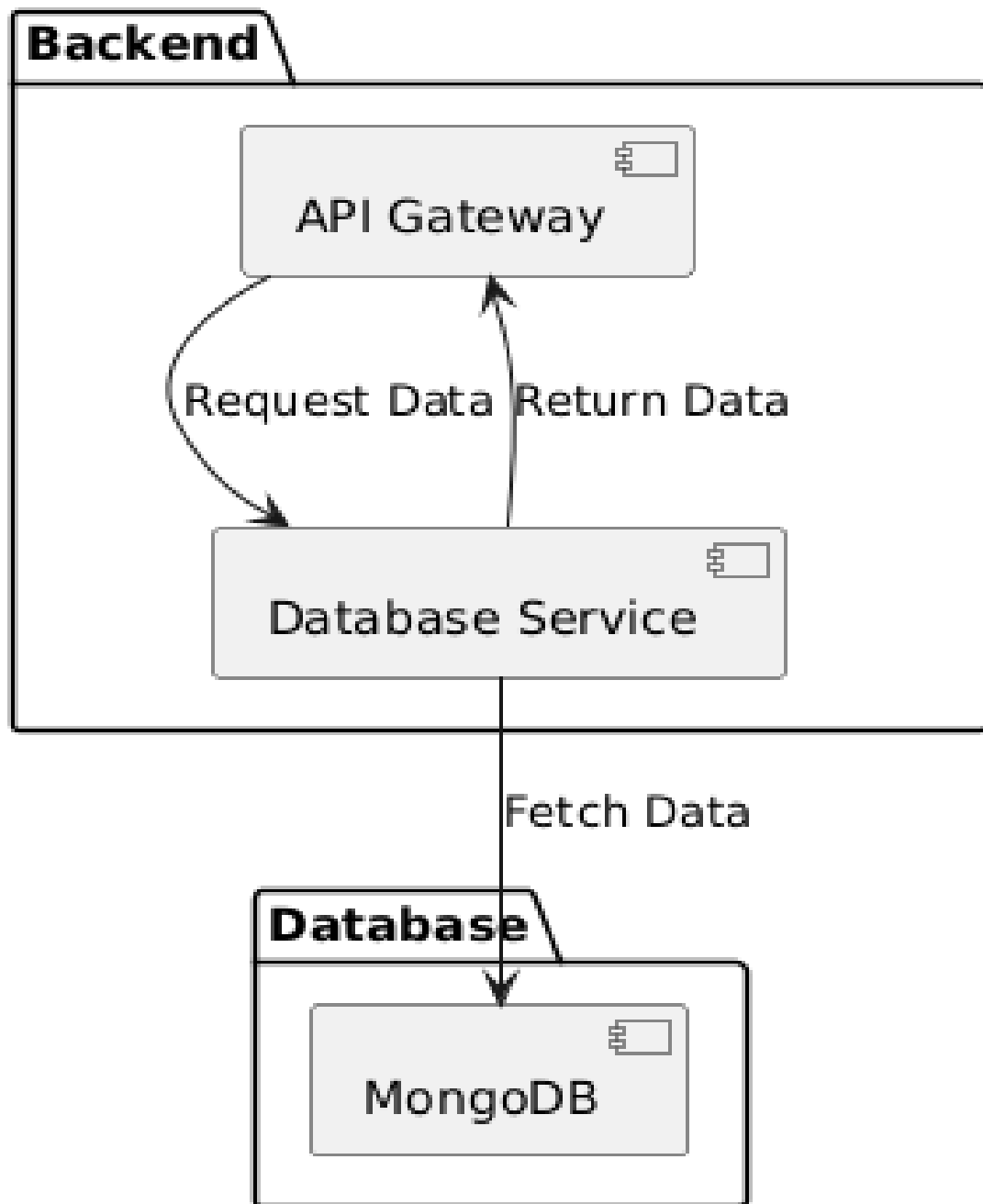


FIGURE 2.45: Component Diagram: Data Service Module

## 2.8 Deployment Diagram

Deployment diagrams are a type of structural diagram used in software engineering to illustrate the physical deployment of artifacts on nodes within a system architecture. In the context of this project, which focuses on a machine learning model for YouTube video transcription, summarization, and assessment generation, deployment diagrams provide a comprehensive view of how the software components are distributed across various hardware nodes and how they interact with each other in a live environment.

The Deployment Diagram (Figure 2.46) illustrates the architecture of the project, detailing the physical deployment of software components across various nodes in the system. It outlines how different parts of the project interact with each other and where each component is hosted.

1. Client Machine: Represents the user's environment, specifically the web browser, where users interact with the front-end of the system to upload videos, generate transcriptions, summaries, questions, and view assessments.
2. Application Server: This node hosts the backend services that handle core functionalities, including:
  - API Gateway: Acts as an intermediary between the front-end and back-end services, routing requests to the appropriate services.
  - Authentication Service: Manages user authentication (login, registration) and interacts with the database to store user credentials.
  - Video Processing Service: Handles video uploads and processes the video metadata, which may involve extracting details like title, length, and size.
  - Transcription Service: Sends video data to external APIs for transcription and stores the transcription data in the database.
  - Question Generation Service: Responsible for generating various types of questions (MCQs, fill-in-the-blanks, etc.) based on the transcription.
  - Assessment Service: Manages the assessment process by storing results and allowing users to view them.
3. External API Services: These APIs provide functionality that complements the system's internal capabilities:
  - Whisper API: Provides transcription services for video content.

- Summarization API: Generates summaries from the transcribed text.
  - Question Generation API: Helps generate questions for assessments based on the transcription data.
4. Database Server: The MongoDB server stores user information, uploaded video data, transcriptions, questions, and assessment results.

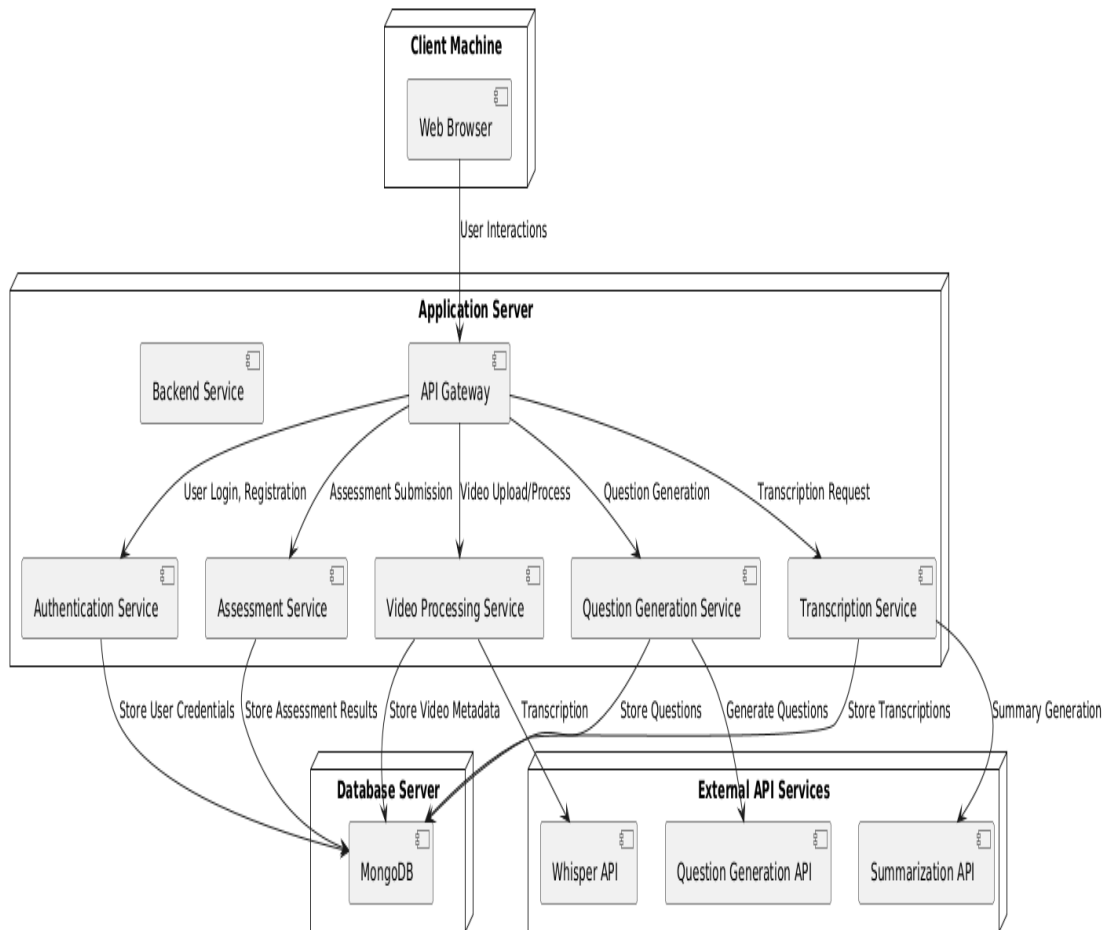


FIGURE 2.46: Deployment Diagram

## 2.9 ER Diagram

In the context of our project, which involves a machine learning model for processing YouTube videos, ER diagrams facilitate the understanding of the relationships between different entities in the database, such as users, videos, transcriptions, assessments, and questions.

The primary purpose of ER diagrams is to outline how entities interact within the system and to define the attributes of each entity. For example, in this project, the User entity may contain attributes such as user ID, username, email, and password, while the Video entity could include attributes like title, length, size, and video URL. By illustrating the connections between these entities—such as how users upload videos and receive assessments—ER diagrams help ensure that the underlying database structure is robust and supports the application’s functionality.

The Entity-Relationship (ER) Diagram (Figure 2.47) provides a conceptual representation of the database structure for the project. It highlights the various entities involved and the relationships between them:

1. **User:** Represents the users of the system. Each user has a unique `userId`, along with attributes like `username`, `email`, `password`, and `accountCreated`. A user can upload multiple videos and take multiple assessments.
2. **Video:** Represents the video content uploaded by the user. Each video has attributes such as `videoId`, `title`, `length`, `size`, `videoUrl`, and `thumbnailUrl`. A user can upload many videos, and each video generates a transcription and can be associated with multiple assessments.
3. **Transcription:** Stores the transcription data generated from the video. The `transcriptionId`, `url`, and `transcriptionText` attributes describe the transcription. Each video can generate one transcription, which can also lead to the creation of summaries and key concepts.
4. **Assessment:** Represents the assessments taken by a user based on a video. Each assessment is linked to a `userId`, `videoId`, and contains attributes like `assessmentId` and `result`. Each assessment is associated with multiple questions.
5. **Question:** Stores the questions generated for an assessment. The `questionId`, `assessmentId`, `questionText`, and `answer` describe the question. Each assessment can contain multiple questions.

6. VideoProcessingApi: Represents the different APIs used for the transcription, question generation and assessment purposed.
7. LLMProcessing: Represents the LLM model used for to process the video. It contains attributes like Process\_transcription, generate\_summary, identify\_key\_topics and other features of the system.
8. MediaProcessing: Stores the video details, transcription, summary and all the other extracted features of the video.

**Relationships:**

- User - Video: A user can upload multiple videos, represented by a one-to-many relationship.
- Video - Transcription: Each video generates one transcription, forming a one-to-one relationship.
- Transcription - Question: A transcription can be used to create multiple questions.
- Question - LLMProcessing: LLM processing is used to create questions.
- LLMProcessing - VideoProcessing: LLM Processing interacts with Video Processing API.
- Assessment - LLMProcessing: LLM Processing interacts with assessment to process answers and generate result.
- LLMProcessing - Transcription: LLM Processing processes transcription to do further operations.
- Video - MediaProcessing: MediaProcessing contains video for transcription purposes.
- Transcription - MediaProcessing: MediaProcessing contains transcription for summary extraction and other purposes.
- Assessment - MediaProcessing: MediaProcessing contains assessment for evaluation purposes.
- Question - MediaProcessing: MediaProcessing contains question to generate various questions.

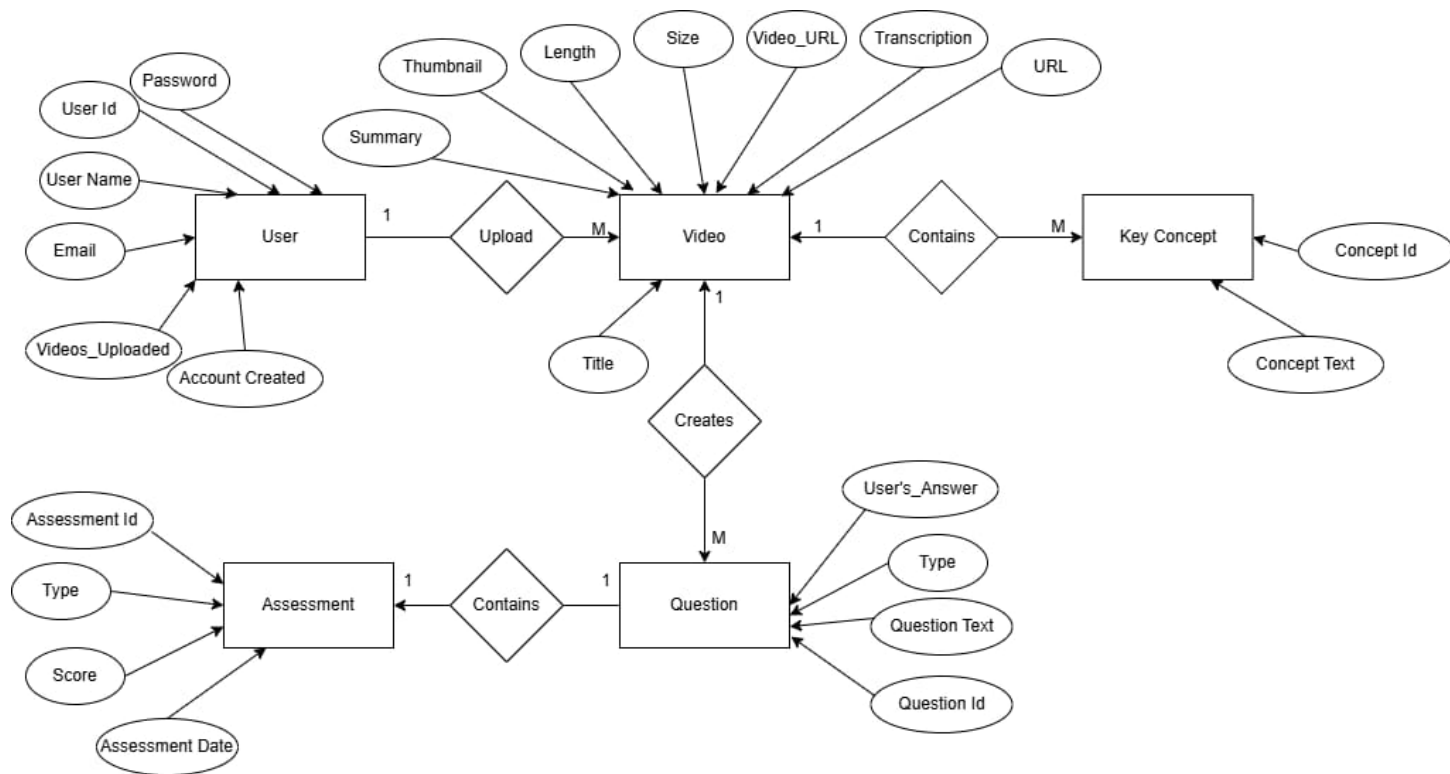


FIGURE 2.47: ER Diagram