Echos

“One place to learn and translate American Sign Language”

System Requirements Specification

Group 11

Prayag Savsani [AU1841035]
Darshan Navadiya [AU1841061]
Kishan Mehta [AU1841072]
Harsh P. Patel [AU1841075]
Dhruv Sharma [AU1841102]
Harin Khakhi [AU1841116]
Nilay Gandhi [AU1841118]
Sanket J Shah [AU1841120]

TABLE OF CONTENTS

1. Introduction

1.1 Purpose

The purpose of this document is to provide the information about the design process of the ‘Echos’ application and provide an overview of the implementation. The design and technical perspective for the application, different system diagrams and the assumptions and constraints of the application are elucidated further in this document.

1.2 System Scope

The system design provided in this document builds upon the scope defined in the System Requirements Specification document.

1.3 Audience

The audience for this application are:

- The people who would like to learn sign language for their personal use like to understand a deaf friend, family member, etc or the person who just likes to learn new things.
- The people who do not understand sign language can translate the sign language to text by pointing their device’s camera to that person and convert it.

2. Design Overview

2.1 Architectural Goals

The key architectural goal is to use the best software engineering practices for designing and developing a scalable and maintainable Android application that is compatible with most devices.

2.2 Technologies

Front-end:

- **Dart:**
  Dart is a programming language designed for client development, such as for the web and mobile apps. It is developed by Google and can also be used to build server and desktop applications. Dart is an object-oriented, class-based, garbage-collected language with C-style syntax.
- **Flutter:**
  Flutter is an open-source UI software development kit created by Google. It is used to develop applications for Android, iOS, Linux, Mac, Windows, Google Fuchsia, and the web from a single codebase.

Back-end:

- **Firebase:**
  Firebase is a platform developed by Google for creating mobile and web applications. It can be used as a NoSQL database as well as a platform for hosting machine learning models.
- **Flutterfire:**
  FlutterFire is a set of Flutter plugins which connect your Flutter application to Firebase.

Machine Learning:

- **Tensorflow:**
  Tensorflow is an open source library for developing machine learning models. The model used in our application is trained and tested using Tensorflow. We also took advantage of Tensorflow Lite stack which allowed us to deploy the model right into our Android application.

- **tflite:**
  Tflite is a Flutter plugin for accessing TensorFlow Lite API. Using its helper functions, we could easily capture the live feed and classify it using our model.

2.3 Design considerations

Compatibility:
The software is made to be compatible across all the android devices available in the market.
Extensibility:
As this application is made in flutter and it is a very flexible application. We can add any new feature or functionality very easily. This application can be easily extended to the iOS devices as the same flutter code can be used for both Android and iOS.
Maintainability:
It is easy to maintain the application as the code of the application is divided into multiple modules.
Modularity:
All the components of the software are modularised and divided into sections like screens, reusable-widgets, services, models, helpers and utilities. This makes all the components easily accessible and maintainable.
Reusability:
All the reusable components are designed to be used in different contexts and are easily accessible and maintainable in the reusable-widgets sections.
Scalability:
The scalability of the tutorial part of the application is high as it dynamically renders the components according to the data available on the database. The translator part is as scalable as the firebase hosting service allows it to be.
Usability:
The UI of the application has been designed using the Human-Computer Interaction principles to make the user experience pleasing. This makes the application simple to use for any beginner and also keeps the daily user interested.
Testability:
The application can easily be tested by making any changes because flutter has a feature called “Hot Reload” which instantaneously reflects the changes to the application. As the application is modularised, unit testing becomes easy and effective.

3. System Architecture and Diagrams

3.1 Overall Architecture Diagram

The following diagram shows the overview of the system architecture:

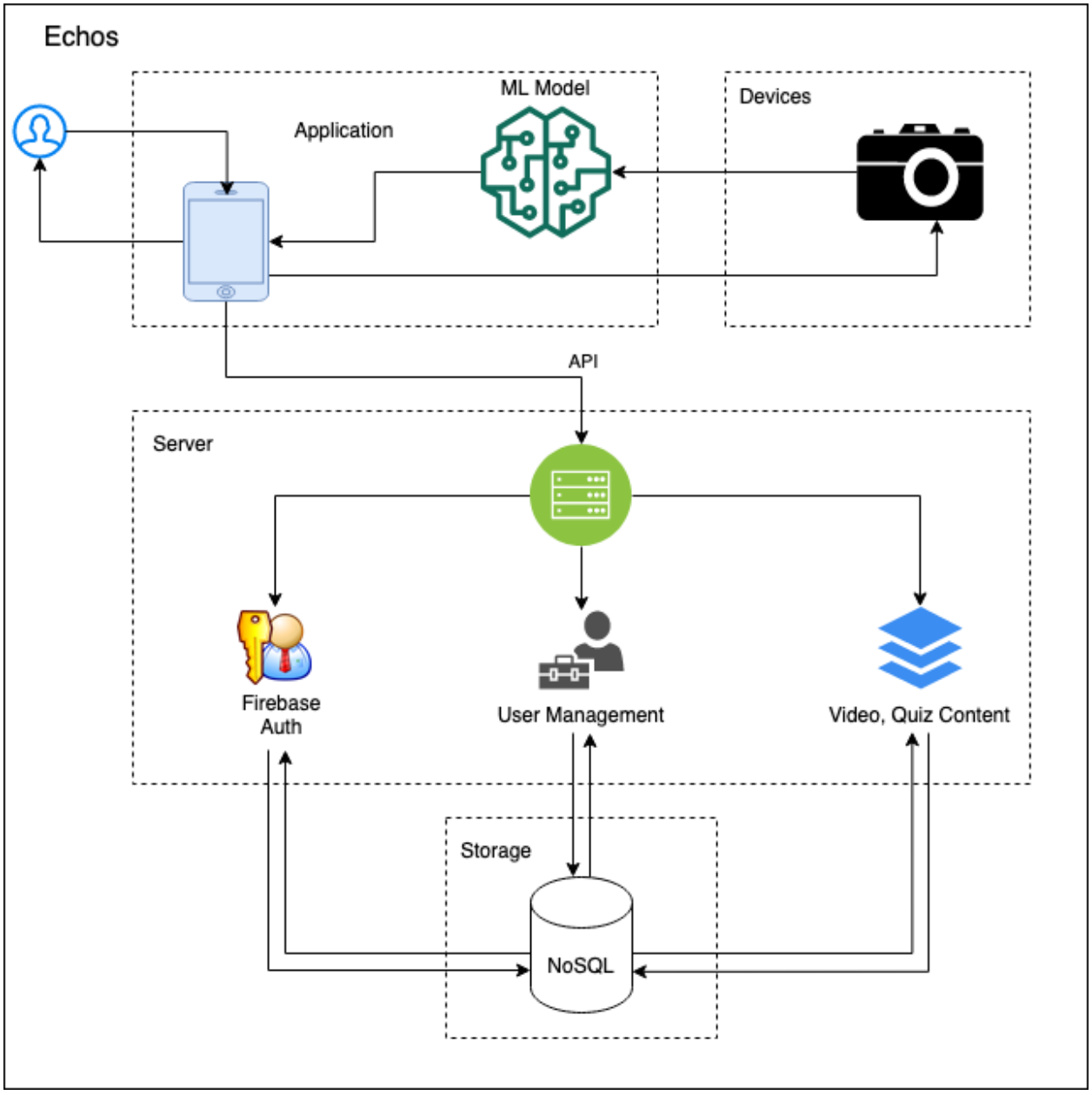


Fig. System architecture diagram

3.2 Use-Case Diagram

Use Case Diagram:

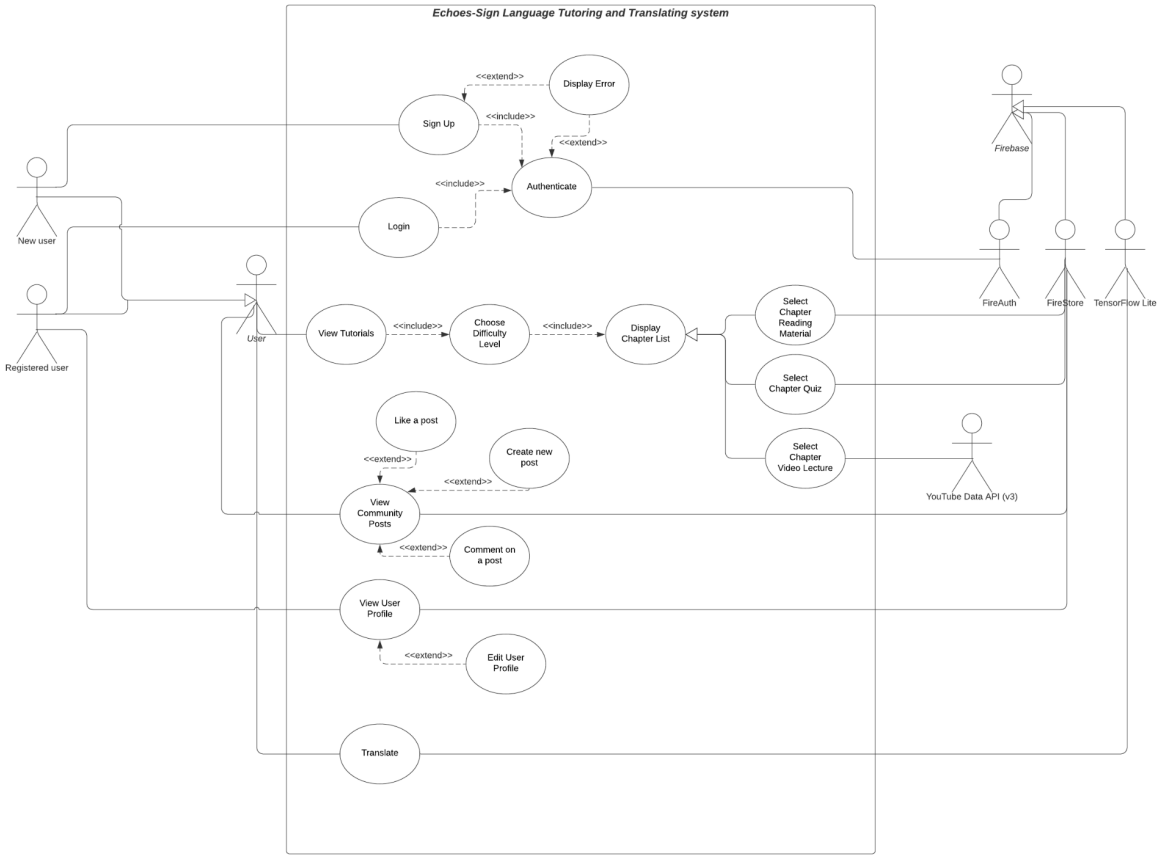
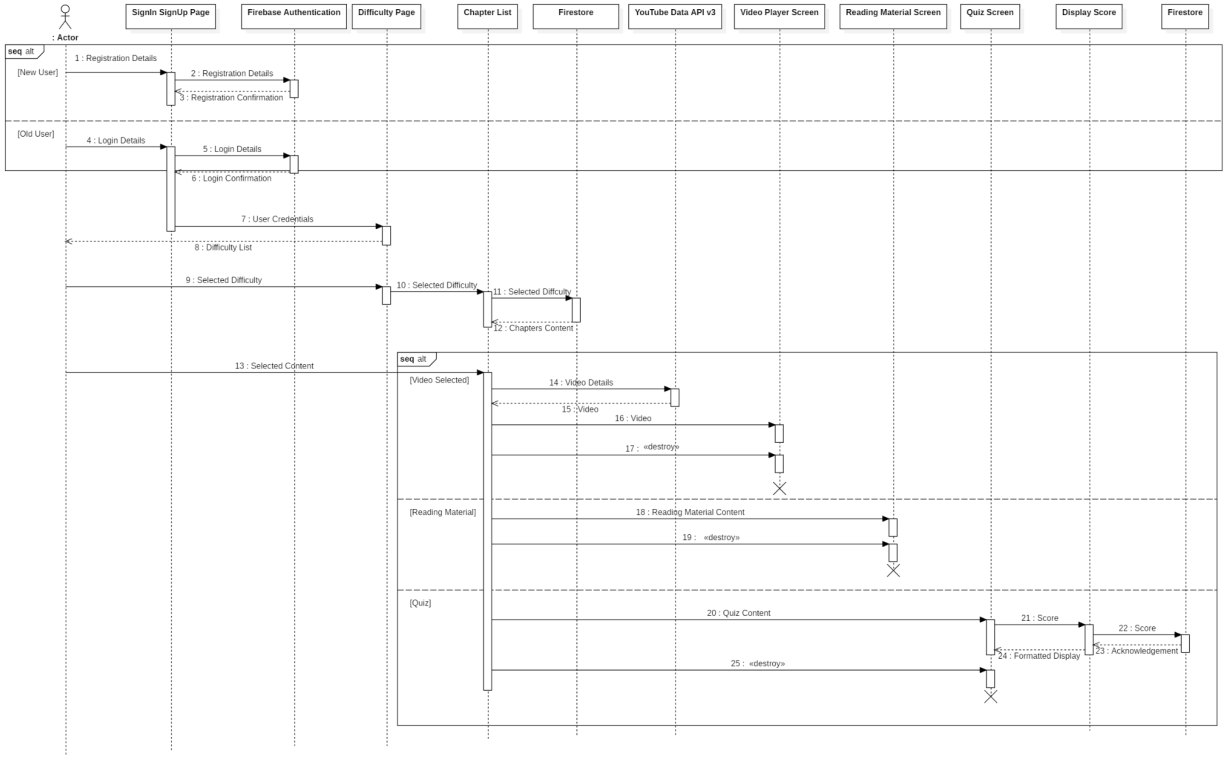


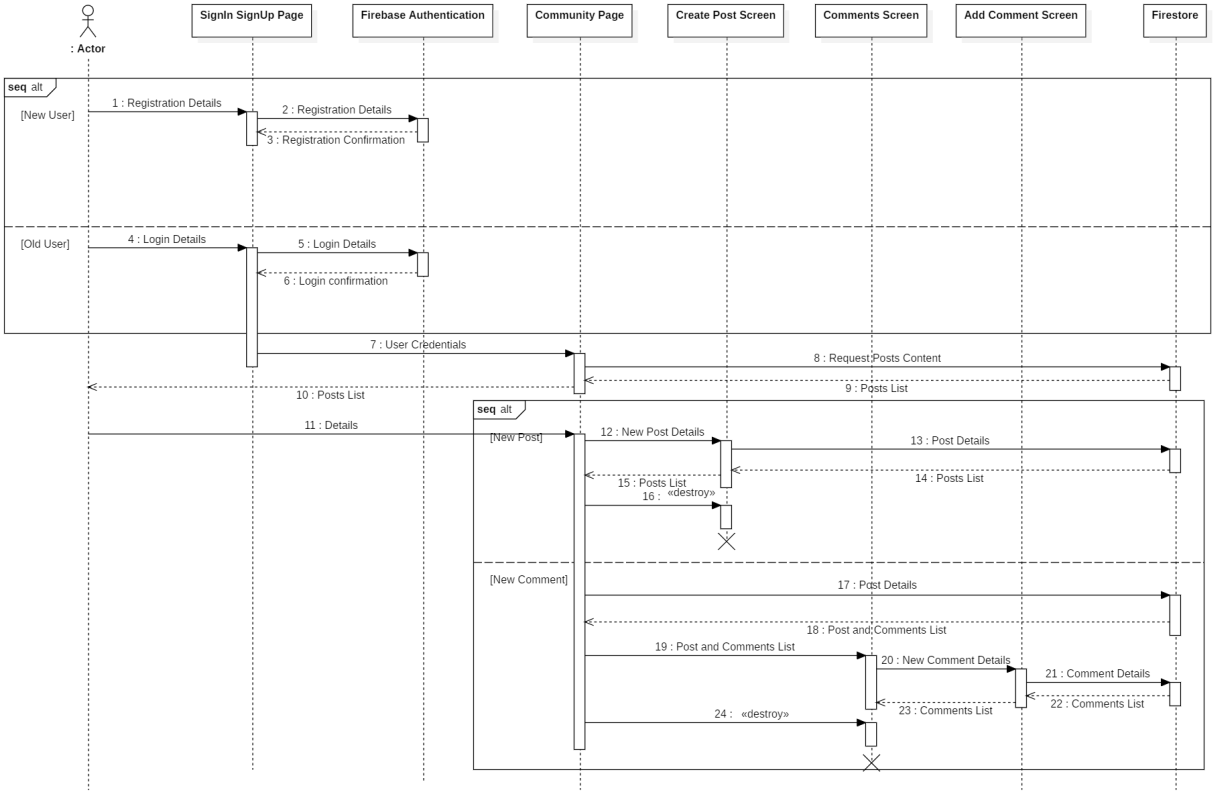
Fig. UseCase Diagram

3.3 Sequence Diagram

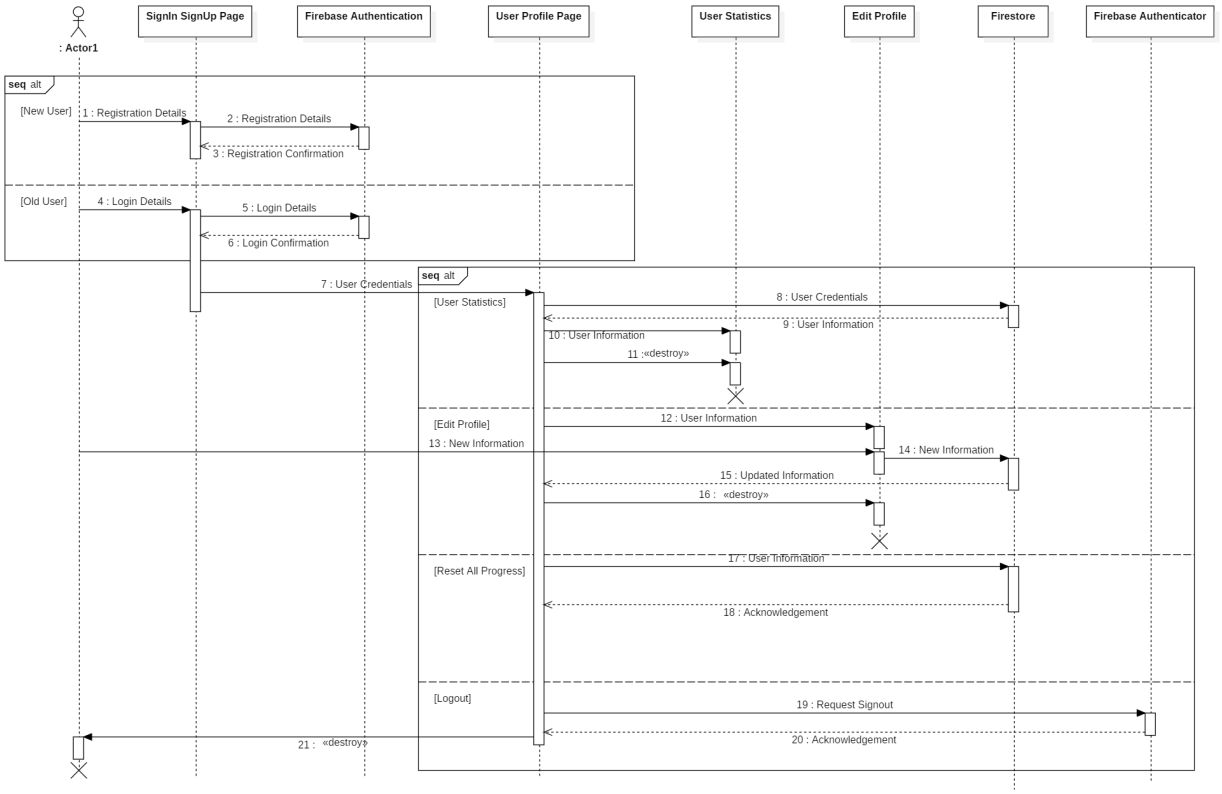
Sequence Diagram for attempting video, reading material or quiz by a new user or an old user:



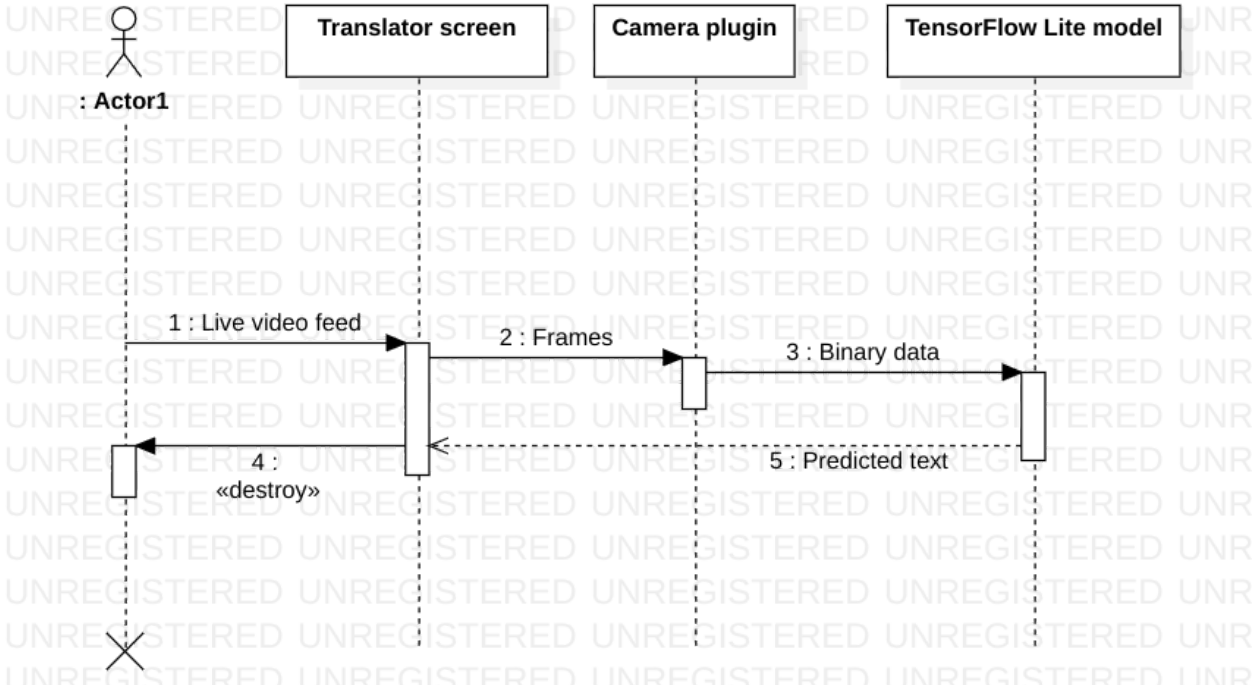
Sequence Diagram for creating and viewing posts and comments by old or new users:



Sequence diagram for viewing use profile:

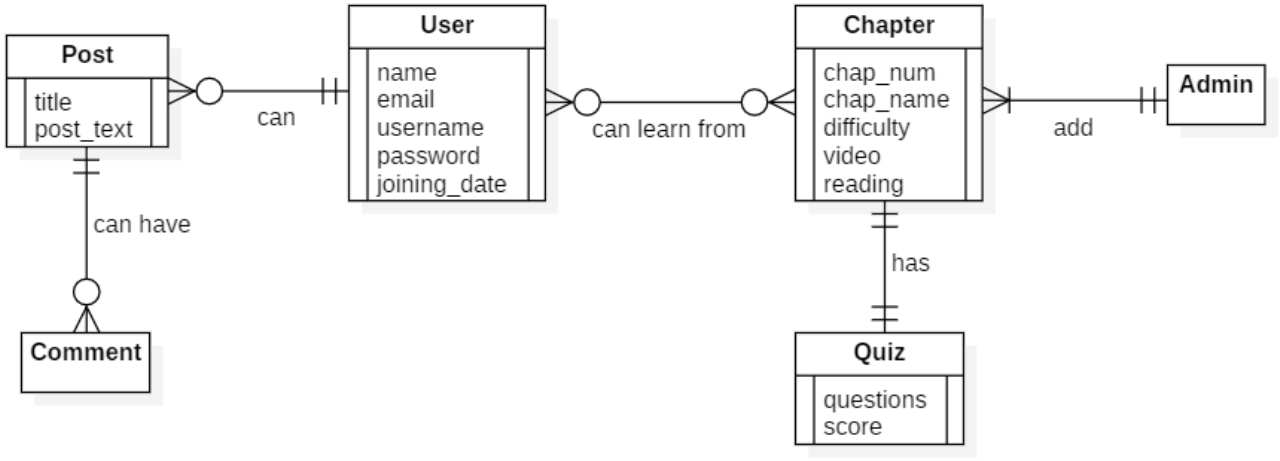


Sequence diagram for translating ASL to text:

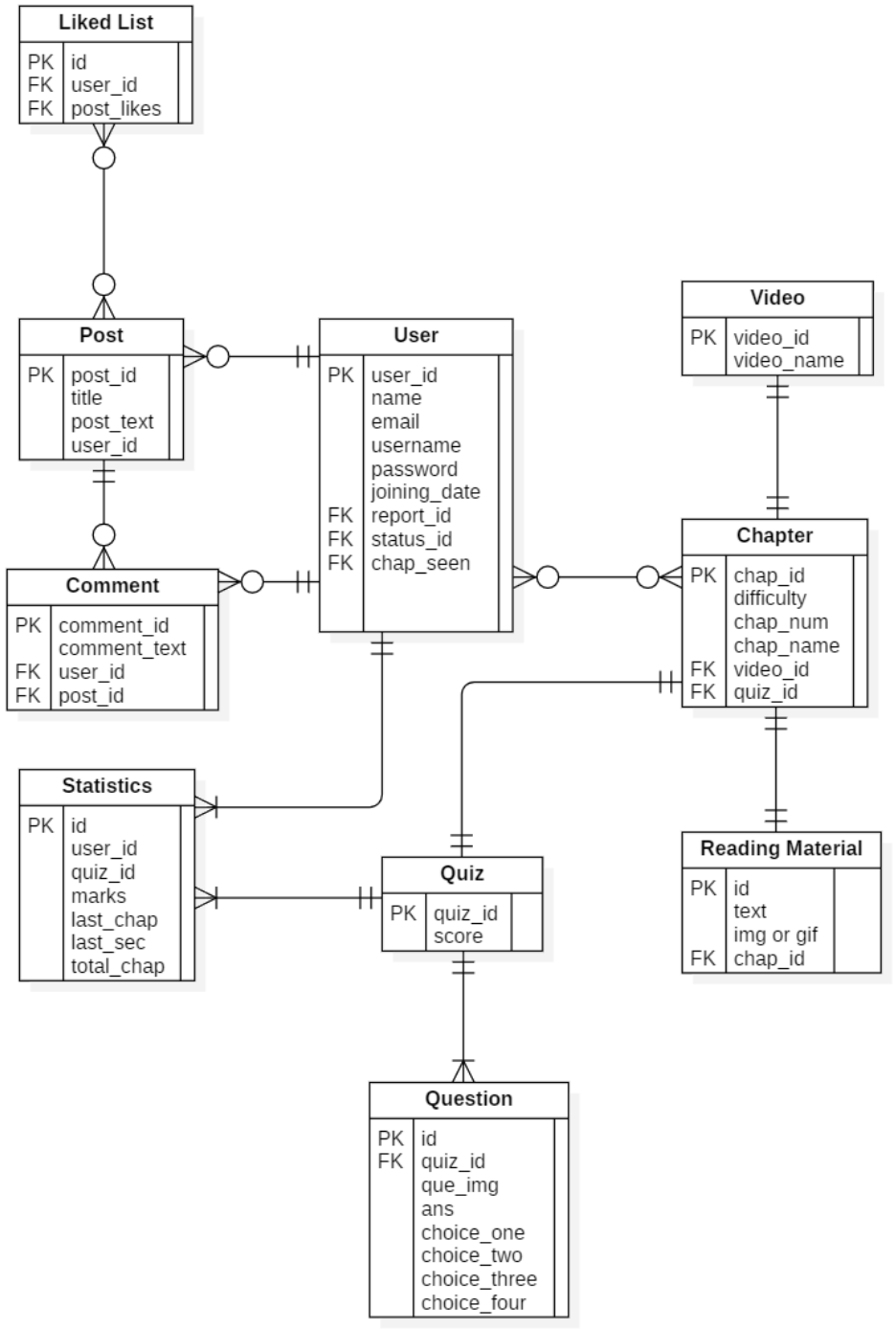


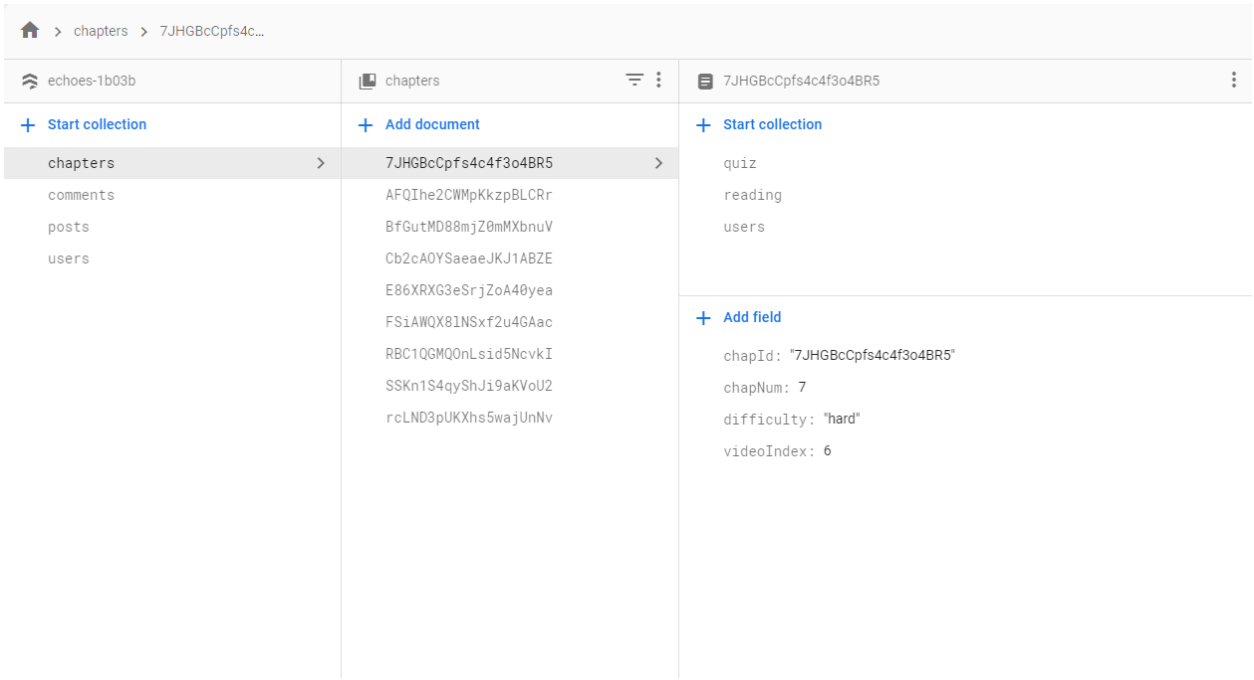
3.4 ER Diagram

Logical ER Diagram:



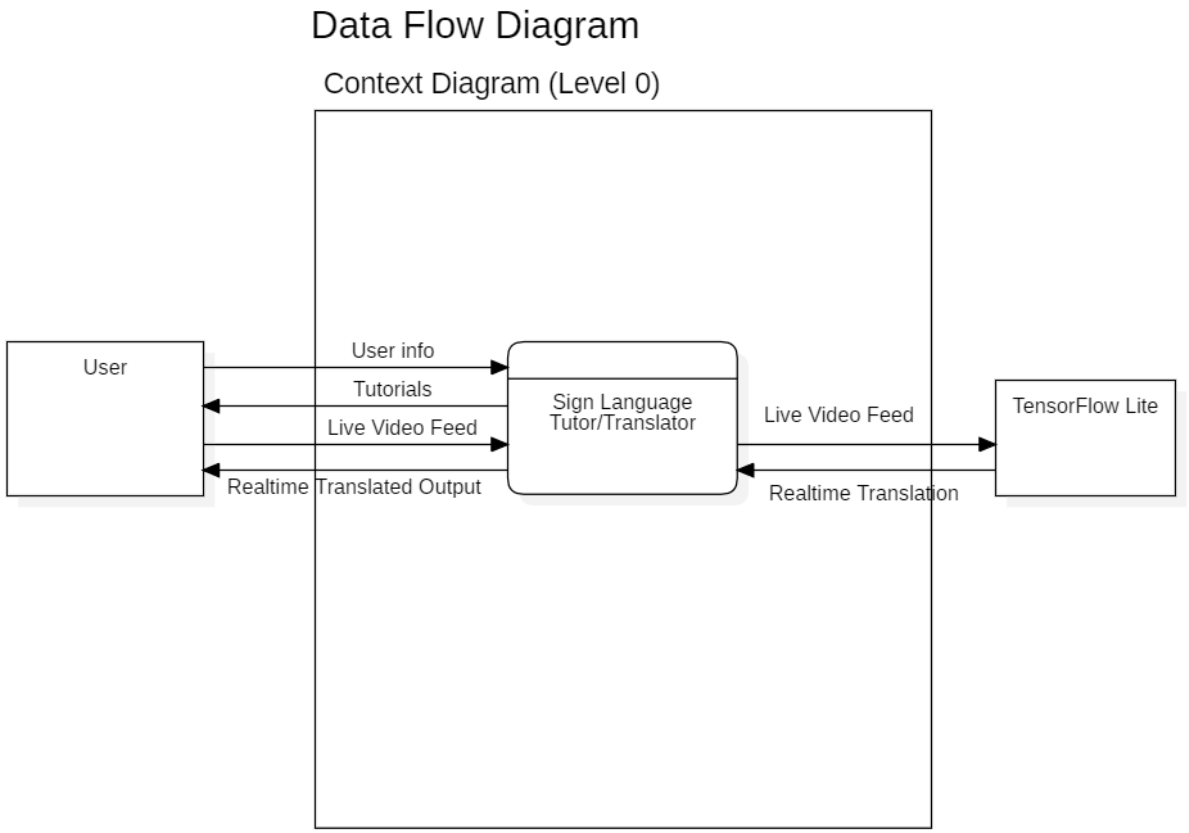
Physical ER Diagram



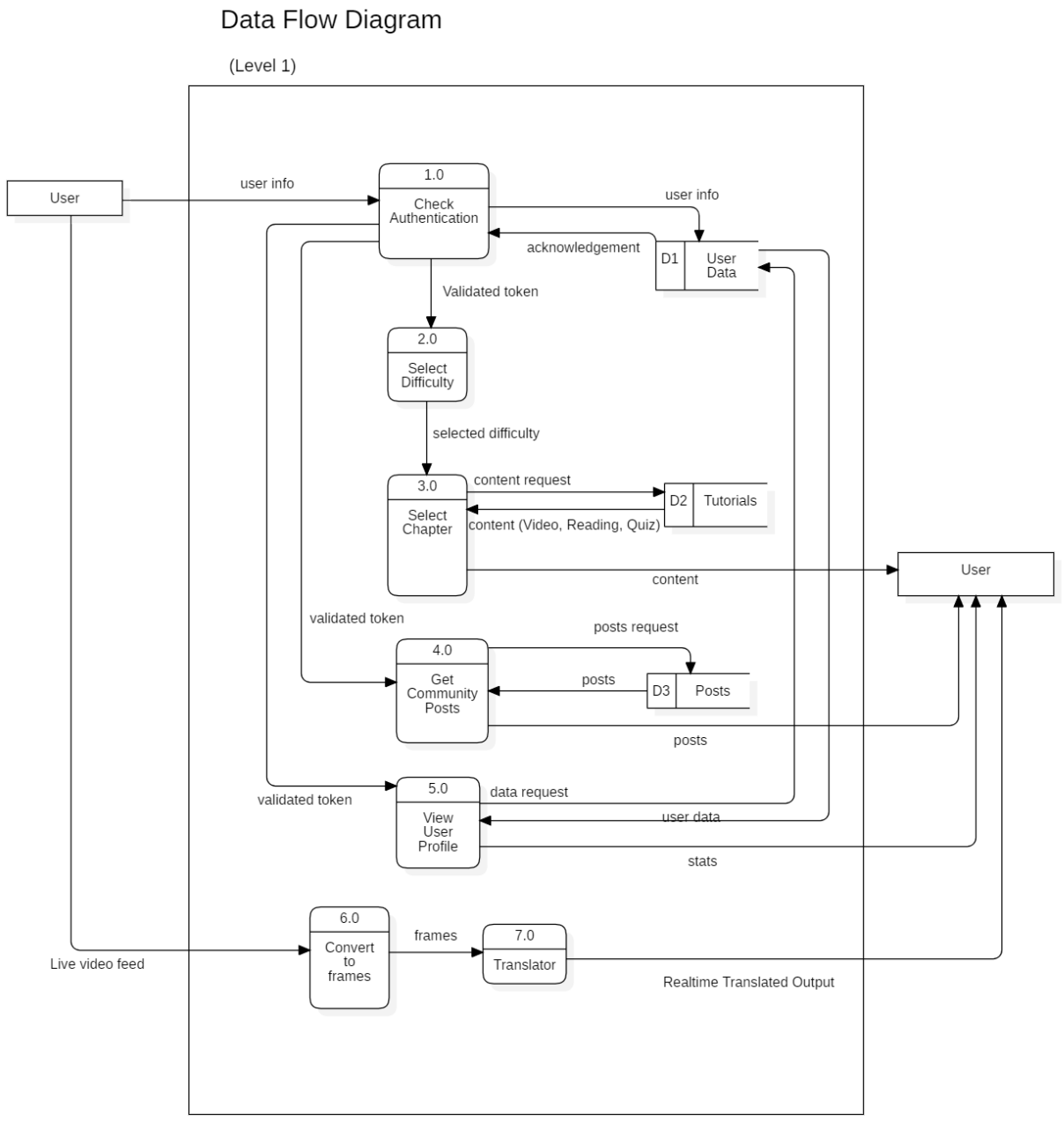

chapters > 7JHGBcCpfs4c...
echoes-1b03b
Start collection
chapters
comments
posts
users
chapters
Add document
7JHGBcCpfs4c4f3o4BR5
AFQIhe2CWMpKkzpBLCRr
BfGutMD88mjZ0mMXbnuV
Cb2cAOYSaeaeJKJ1ABZE
E86XRXG3eSrjZoA40yea
FSiAWQX8lNSxf2u4GAac
RBC1QGMQOnLsid5NcvkI
SSKn1S4qyShJi9aKVoU2
rcLND3pUKXhs5wajUnNv
7JHGBcCpfs4c4f3o4BR5
Start collection
quiz
reading
users
Add field
chapId: "7JHGBcCpfs4c4f3o4BR5"
chapNum: 7
difficulty: "hard"
videoIndex: 6

3.5 Data-flow Diagram

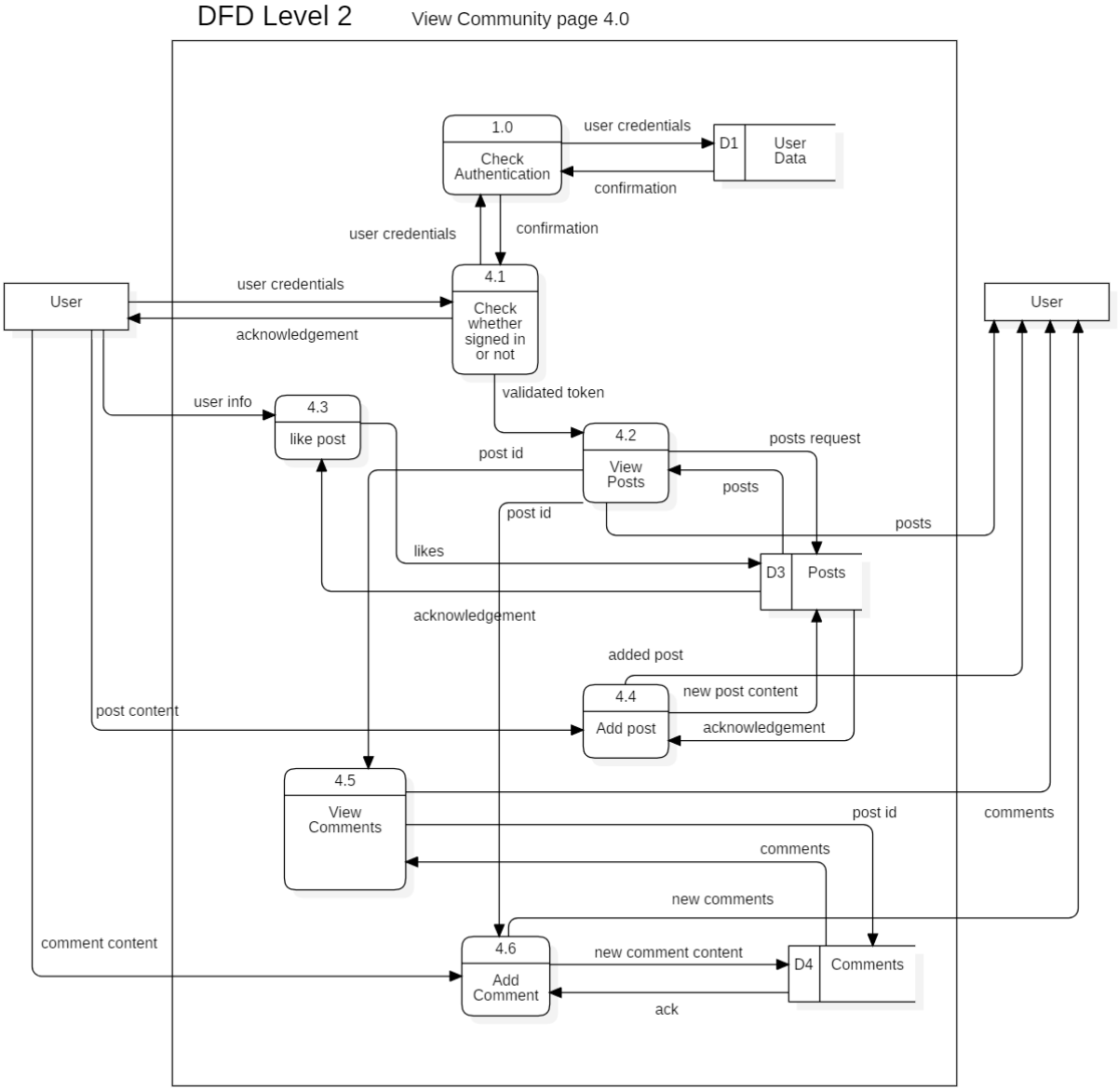
DFD Level 0 (Context Level)



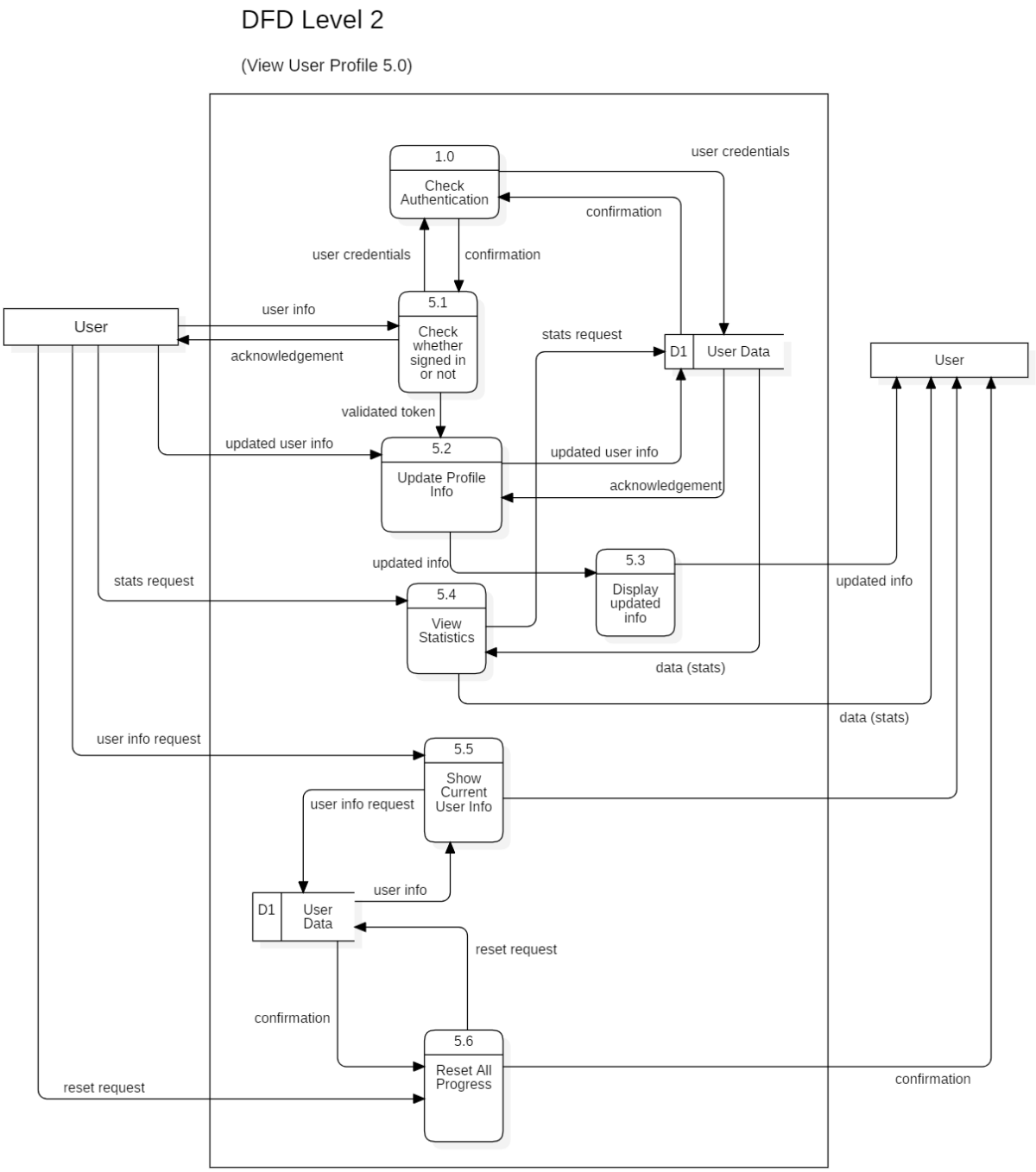
DFD Level 1


Data Flow Diagram
(Level 1)
User
user info
1.0
Check Authentication
user info
acknowledgement
D1
User Data
Validated token
2.0
Select Difficulty
selected difficulty
3.0
Select Chapter
content request
D2
Tutorials
content (Video, Reading, Quiz)
content
User
validated token
posts request
4.0
Get Community Posts
posts
D3
Posts
posts
validated token
5.0
View User Profile
data request
user data
stats
Live video feed
6.0
Convert to frames
frames
7.0
Translator
Realtime Translated Output

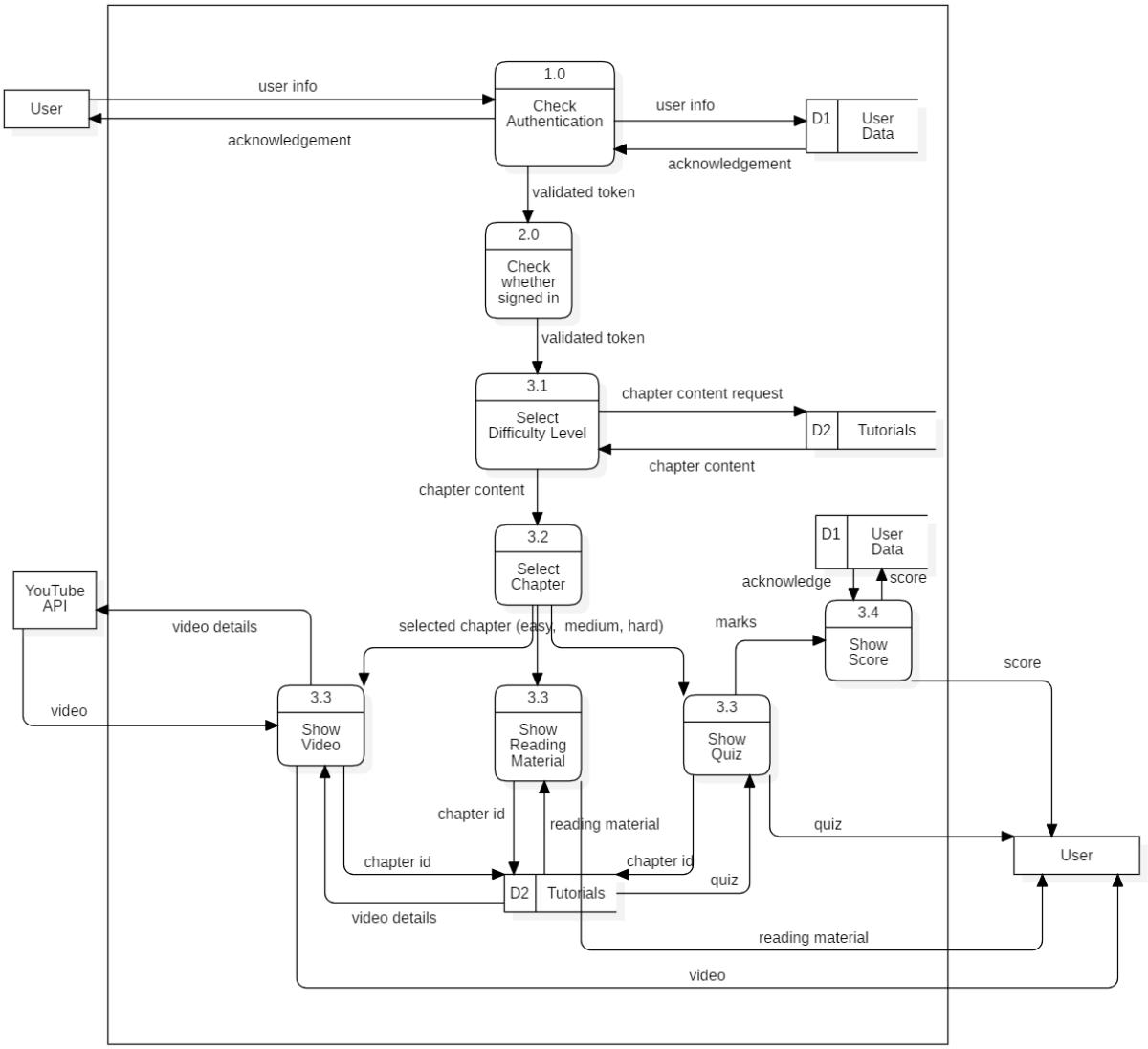
DFD Level 2 (4.0) View Community Page



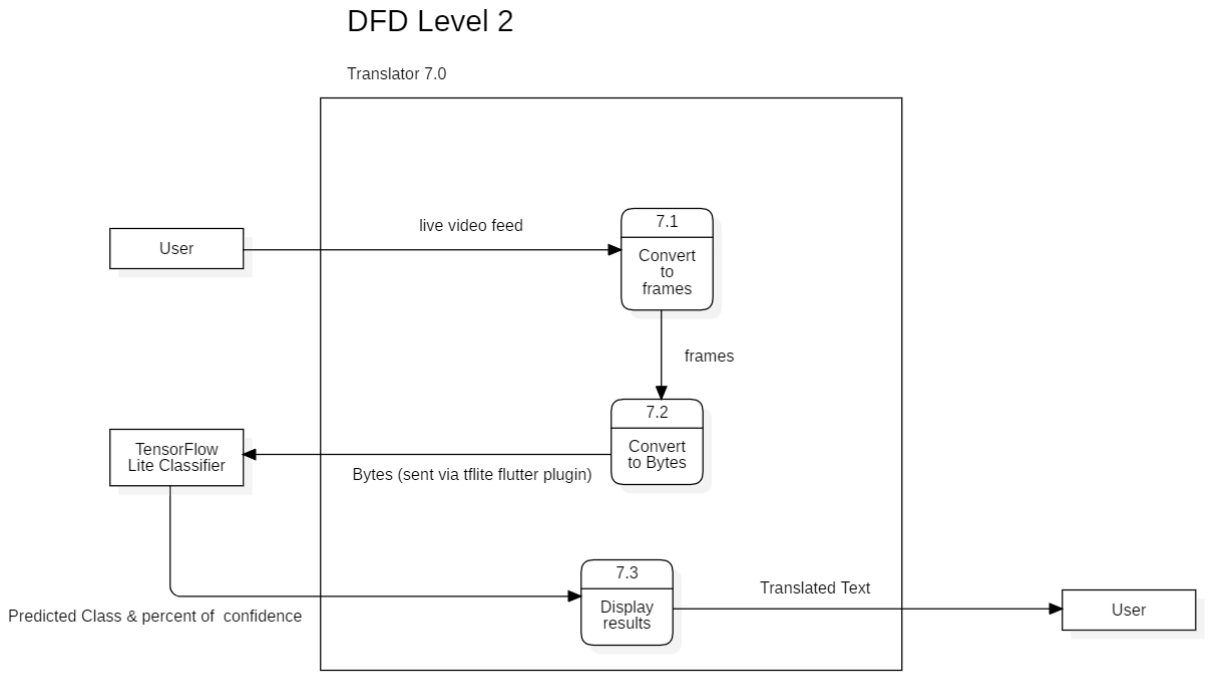
DFD Level 2 (5.0) View User Profile



DFD Level 2 (2.0) View Difficulty Page and Chapter List



DFD Level 2 (7.0) Translator



3.7 UI design

Please refer to the Echos_UI_Design document.

4. Assumptions and Constraints

4.1 Assumptions

The basic assumptions made for the application are:
[i] The user has a stable internet connectivity for viewing the video tutorials and other related content.
[ii] The user's device has a compatible camera that can be accessed by the translator module of the application.

4.2 Constraints

The basic constraints of the application are:
[i] If the user has limited internet connectivity, the usability of the application greatly decreases as most of the content would not be accessible easily.
[ii] If the camera on the user's device malfunctions, the translator module of the application becomes unusable.