ASInt – MEEC
2020/2021 - Project
VQA – Video Question&Answer system

In this project students should develop a system (VQA) that will provide a Q&A (question and answers) application around videos.
Users will be able to see videos stored in the system and make questions concerning particular instants of the videos and answer previous made questions.
In this work students should

- Define the set of resources to be made available
- Define the relevant information (attributes) of such resources
- Define the interfaces (WEB and REST) to access such resources
- Implement a server prototype the illustrates the access to the information
- Implement a simple web server for access and management of resources

1 Users

The system will be operated by two classes of users:

- Regular users
- Administrator that will access the admin pages

All users will be authenticated using the FENIX login page.
The system should store a list of user that are administrators. The administrators will also login using the FENIX.

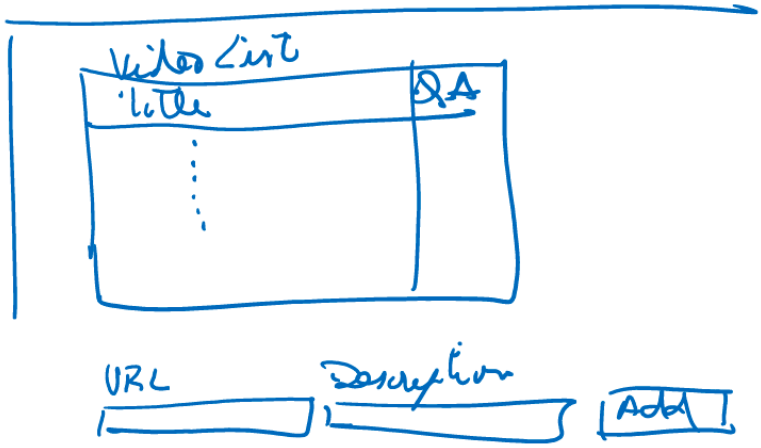
2 User Functionalities

The system will provide a set of web-pages where regular users will be able to perform the following functionalities:

- add a new video
- access a stored video
  - play the a video
  - post a question related to a particular instant of the video
  - answer a question posted by another user

2.1 Video Listing

In one of the application page, it is possible to see the list of videos registered into the system and add new videos.

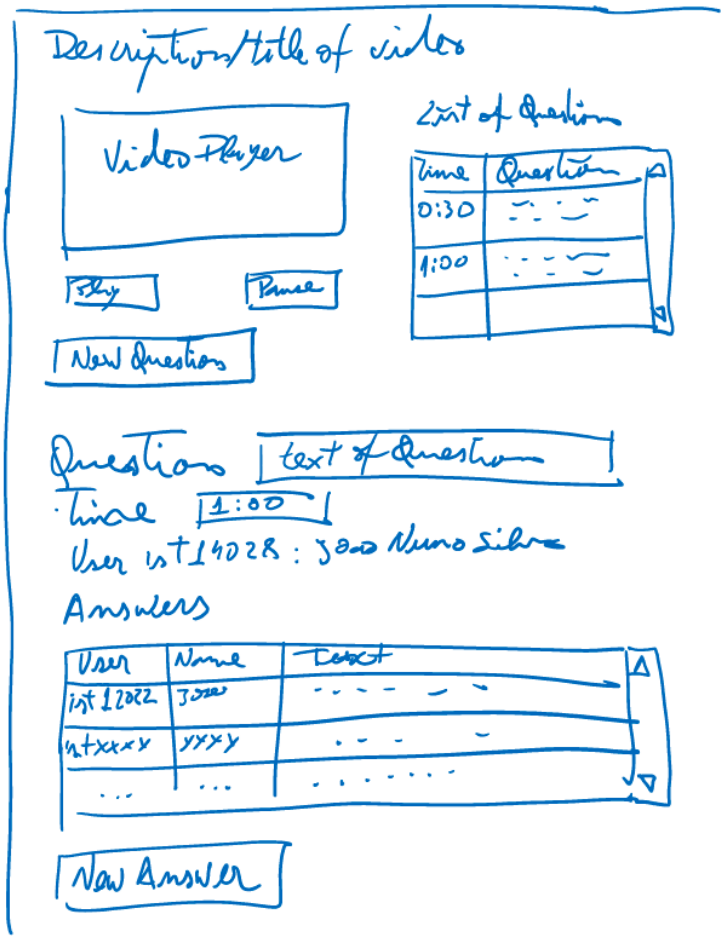


The Video List will provide the title/Description of the registers videos, a way to load the video page and the number of questions made in it
To register a new video to the system, users should provide the URL for the YouTube page of such video and a description/Title.

2.2 Video Page

After selecting a video in the Video Listing, the video page appears.
There the user can

- see the video,
- pause and resume playing
- make a question
- see other question
- answer to a question.



The page will present a video player that can be controlled by two buttons: play and pause.
The list questions of present the various questions previously made about the current video.
For each question the list shows the time on the video and the question text.

2.3 New question

When viewing the video, if the user wants to make a new question, he should click the New Question button.
The videos should be paused to allow the user to write the text of the question.
After submitting the question text and time, the video continues executing.

2.4 View question

If the user selects one particular question he will see following information

- Text of the question
- The time instant on the video
- the user that made the question
- the various answers to that question

By clicking in the New Answer Button the user can add his only answer to the current question

3 Administrator functionalities

Administrator user will have access to two additional pages:

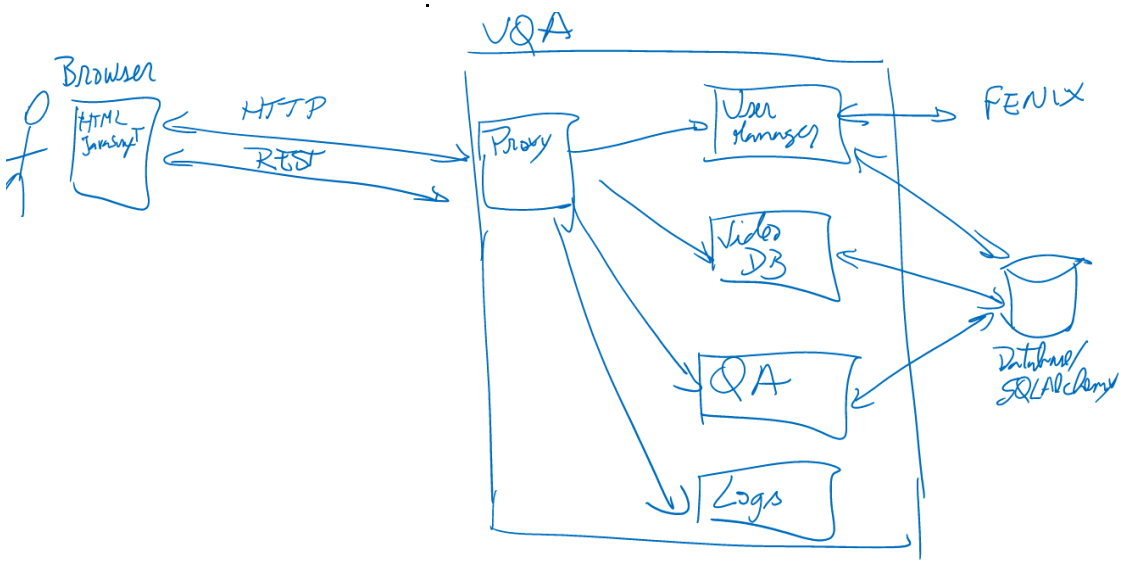
- Log listing
- User statistics

The Log listing will show all the events registers in the Log service.
The user statistics will show for each user the following:

- the number of videos registered
- the number of video views
- the number of questions made by him
- the number of answers made by him

4 Architecture of the system

The system will be implemented following the micro-service architecture, where each functionality is implemented by a simple web-service and will have the following components



List of components:

- FENIX – service that will allow user authentication and provides user names
- Proxy – web application that will hide all other components and provide to the user web pages and a set of REST APIs to be called from the JavaScript.
- User manager – Service that will store all user related information
  Video DB – component that will store the list of videos
- QA – component that will contain the list of questions and corresponding answers.
- Log – component that will record all events in the system

4.1 FENIX

The FENIX system will allow users to authenticate using the FENIX password.
It is possible to use the FENIX TecnicoID and password in web applications using the OAuth protocol.
A description of the way to integrate on web application with the Técnico authentication is available

here: https://fenixedu.org/dev/tutorials/use-fenixedu-api-in-your-application/
The FENIX system exports a set of APIs that provide information about the IST users.
The REST endpoints for FENIX are available in https://fenixedu.org/dev/api/

4.2 Proxy

The Proxy can be implemented in FLASK and export a set of endpoints
These endpoint will serve the HTML pages and the suitable REST API to be used by the browser JavaScript.
This proxy should store the minimum information possible.
In order to answers the requests, the Proxy should contact other components.
The proxy can be further divided into 3 components. Each component will have one specific set of functionalities.

4.2.1 User Web pages

The web pages module will present a set of web pages that will allow users to access information:

- Administrators will be able see the logs and users information.
- Regular will access the tow pages (Video Listing and Video Page) to process the information.

These web pages can have JavaScript code that call the supporting REST APIs

4.2.2 Authentication

This component will allow users to login into the VQA system using the FENIX password.
The authentication module will allow users to authenticate themselves using the FENIX TecnicoID and password.
VQA should implement the OAuth protocol.

4.2.3 API

The API component will export all the REST endpoint necessary by JavaScript code.
These endpoints can replicate those in the components or use as supporting data information from multiple components

4.2.4 Admin web pages

The admin pages will be accessed by the administrator to:

- list the logs
- list the registered users with statistics

Only administrators are authorized to access these pages.

4.3 User manager

This component will store all users information:

- User id
- Name
- Any other necessary information.

Whenever a user logins for the first time, a user record is stored.

4.4 Video DB

This component will store the basic information about a video:

- URL
- description/Title
- user that created it

4.5 QA

This component will store all the questions made along with its answers.

The minimal information for a question is:

- Video identifier
- Time
- User
- Text

If a question has some answers then it is also necessary to store for each answer:

- User
- Text

4.6 Log

The log component will store every relevant event of the system:

- every message exchange with any of the components (requests made to the proxy or any other component)
- data creation events (new user, new video, new question or new answer)

For the events it is necessary the request information (IP, endpoint) and a timestamp
For the data creation it is necessary to store the data type, the content of the data creation, the timestamp and the user responsible for such creation

5 Front-end

The Front-end runs on the browser and should be developed using basic HTML and JavaScript.
The pages that provide this front-end are delivered by the proxy.

6 Extensibility

Although this implementation of the system only contains 4 components the system should be implemented so that it can be easily extended in various ways

- new resources
- modification of the data of the existing resources.

7 Fault tolerance and scalability

Students should define the architectire of the components so that it is possibe to easyly add fault tolerance and scalibilitry characteristics to the maximum number of components.
At least, students should try to implement one of the componets with simple fault tolerance or scalability features and explayin in the report how to add fault tolerance and scalability to the other components.

8 Implementation

Students can use any technology and programming language (python, PHP, java, node.js) to develop the system. The selection the programming language will not affect the final grade.

9 Data persistence

In order to simplify implementation and future deployment in the cloud, students should use the sqlAlchemy or any other ORM

10 Resource identifiers

Students should decide the format of the identifier of resources and whether they are transparent or not.

11 Evaluation

The detailed evaluation criteria will be posted later, but the evaluation will follow these generic criteria:

- Number of implemented functionalities
- Quality of the suitable resource/endpoint
- Extensibility/ Fault Tolerance / Scalability
- Implementation of the back-end
- Implementation of the front-end
- Report

12 Report

The students should write a report describing:

- The system architecture
- Implemented REST API (documentations of the implemented endpoints)
- Used technologies/libraries
- User interface of the system
- FENIX integration
- Extesibility
- Fault Tolerance / Scalability

A section should also describe how the proposed implementation could be deployed on the cloud: presenting the selected environment, its advantages and the necessary changes.