**DELL Technologies Envision the Future Competition 2019-20**

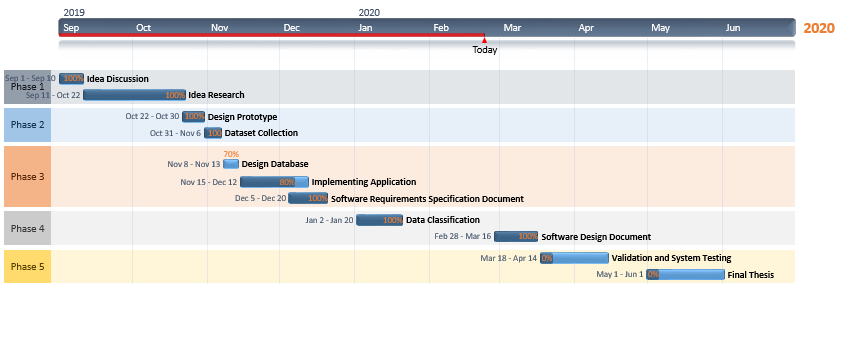
**Interim Progress Report Structure**

**Section 1: Refined Project Description**

* 1. Our main goal is to secure the watching experience on online video streaming platforms, especially targeting the younger audience by creating a plugin tool working with a web browser granting extra privileges. By enhancing the recommendation accuracy by searching within video content, and giving the authority to elders to control what younger audience are watching to keep them safe, in which the selected items will be removed from a given video while displaying the rest of the video. Also when recommending videos, this tool ensures that recommended content is relevant as possible as the objects inside the video are used to compare the relevancy of the video.
  2. The project is a plugin tool, which helps the users to search by a video they upload and the system will recommend videos that they desired. This system will help people to block or cut some scenes from the video they aren’t interested in it. The user will provide the system with a video as an input; the system will start processing on it, then recommend the most similar videos related to it as the system depend on feature extraction on the video so the resulted will be more accurate to the input one rather than the other platforms which used some calculations and algorithms. The system will give the user the chance to cut some scenes from the video he uploads if there are some scenes not desired to him. This software needs internet access.
  3. A new feature is added to our system, which is the audio detection. This will enhance the recommendation system more and more. The content will be analyses based on objects found in the video or the audio file upload. Calculations based on user’s data are used by almost every video platform which makes our proposed solution unique and reliable enough to solve this problem.

**Section 2: Refined Project Plan**

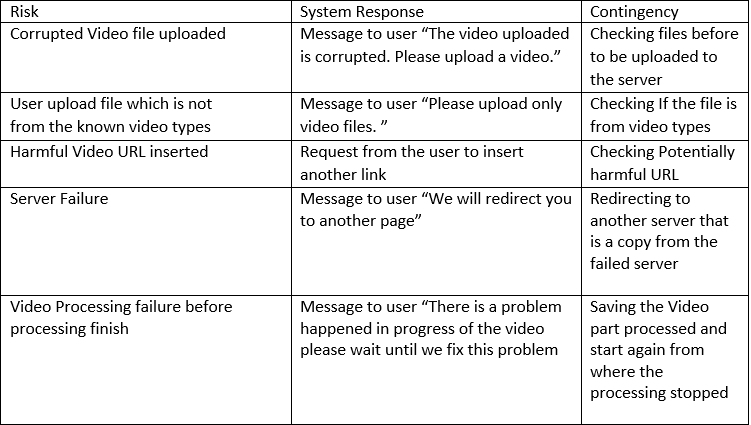
2.1. Detailed schedule and milestones:



2.2. Team members and their roles:

1. Amr Sherif: Creating the documents related to the project, also participated in writing papers that will be submitted in a conference, journal, and competitions.
2. Aly Mekawy: Participated in implementation of the plugin and improving its features.
3. Fouad Osama: Participated in implementation of the plugin and improving its features.
4. Youssef Roshdy: Creating the diagrams related to the project, also participated in writing papers that will be submitted in a conference, journal, and competitions.

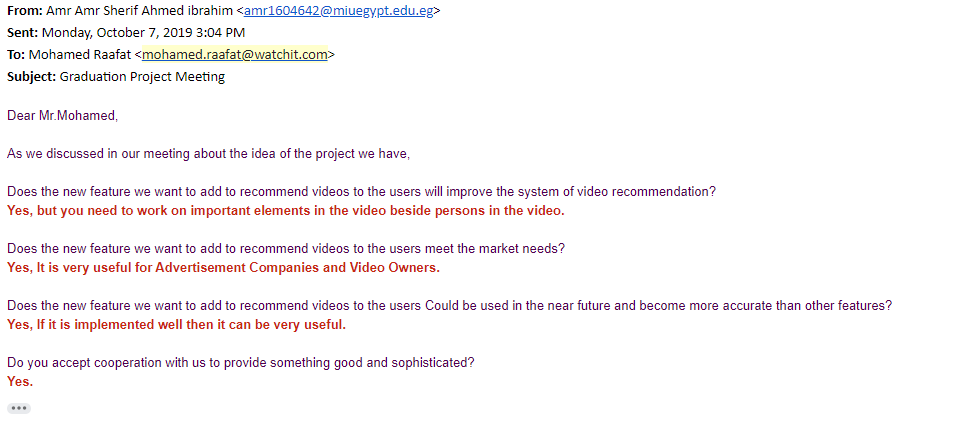
2.3. Contingency and risk mitigation plan



**Section 3: System Requirements**

* 1. 3.1. Requirements Elicitation Process
  2. a) Requirements elicitation: Our goal is to enhance and improve the recommendation system so it can meet up with the user’s needs. We gathering the requirements throughout meetings with our professor and discussing the importance of features that our system will include. We brainstorm together and prioritize the requirements by giving consideration to the time the university gave. Also, our client agrees to the term of the process we make as the core of the project has been implemented.
  3. Analysis: A Feasibility study have been done before choose which requirement should do first and has the priority over the other and which feature will improve the project by giving consideration to the time the university gave.
  4. Prioritization: During the meetings with our professor we ranking the requirements and what is need to improve also, the features which will be added and the plan for the next features according to the prioritization plan we have.

* 1. Change management: We have an agreement with the client about the plan of the project (Requirements and features) that will do though and any change or additional features will be after the university plan time for the project.
  3. b) The system users:
  4. - Any type of user could use our plugin especially the cold start one as the system will recommend the videos that meet the users’ interest after upload a video or enter an URL of a video that need a recommendation for something alike without needing for example; the history of searching for the user, location, country, as some platforms needed. Also, the users who doesn’t like to watch a movie or a video that include scenes they could be cutting and filtered them from the video they watched, for example; If the parents doesn’t like their children to watch a video that contains any undesired scenes, they could cut them by using our online plugin tool.
  5. The system client:
* Our client is called Watch IT which is a web and mobile application.



* 1. c) Challenges encountered:
  2. - We got the experience of how such a plugin implemented

1. - Our Dataset that took a lot of time to get it correctly.
2. - The experience of how to train and test on files and objects we have.

3.2. System Requirements List:

* 1. a) Functional requirements (Most Important Ones):

**FR1:**

•Name: Show Recommendation

•Description: It allows the user to see the recommended videos based on the relevance calculated from the similarity.

•Input: Video

•Output: List of recommended videos

•Precondition: None

•Post-condition: Related videos

•Priority: 10/10

•Expected risks: No recommended videos displayed

**FR2**

•Name: Search with video

•Description: It allows the user to search with a video that he selected to get similar videos.

•Input: Video

•Output: Similar videos

•Precondition: None

•Post-condition: Related videos to the video searched with

•Priority: 10/10

•Expected risks: Searched with corrupted video, Searched with image

**FR3**

•Name: Upload video

•Description: It allows the user to upload his own video

•Input: Video file

•Output: Notify the user for the uploading completeness

•Priority: 10/10

•Precondition: None

•Post-condition: Video uploaded successfully

•Expected risks: Corrupted file

**FR4**

•Name: Cosine similarity

•Description: This function calculates the cosine similarity based on their objects

•Input: Two videos including objects and objects’ frequencies

•Output: Get the cosine similarity between the two input videos

•Priority: 10/10

•Precondition: None

•Post-condition: Cosine similarity is calculated

•Expected risks: Calculation error

**FR5**

•Name: Create a filter

•Description: This function lets the user creates a filter for mature content.

•Input: User puts filter boundaries

•Output: Filter enabled automatically

•Priority: 10/10

•Precondition: None

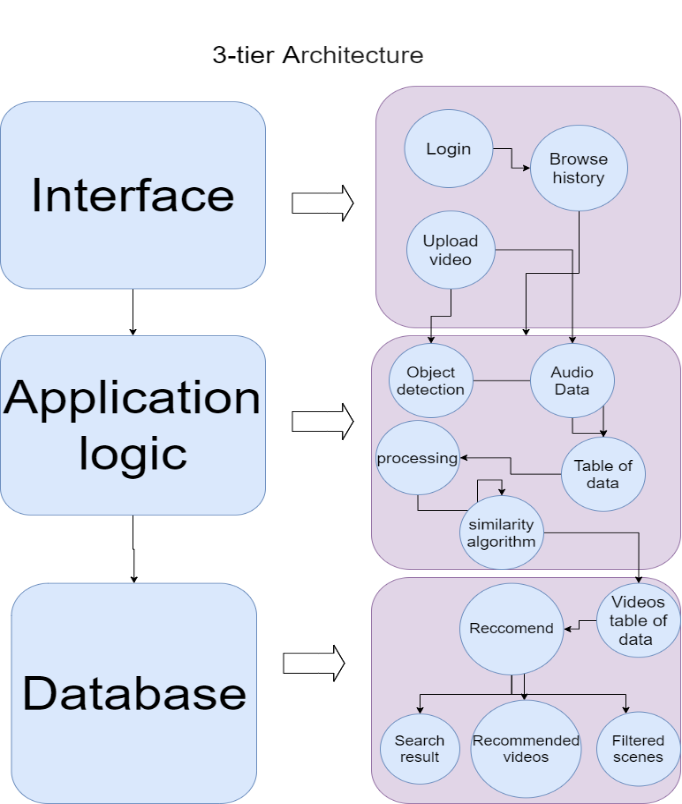
•Post-condition: Filtered videos displayed

•Expected risks: Filter is not working efficiently

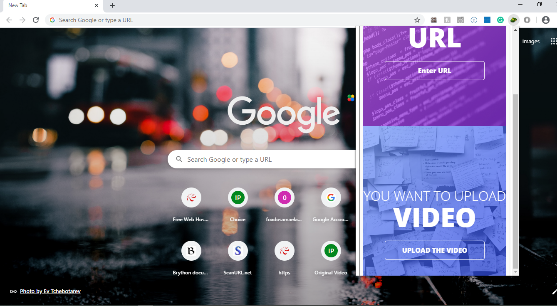
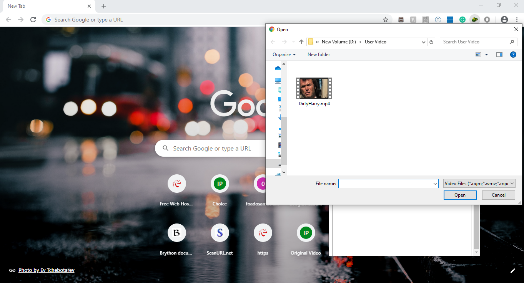
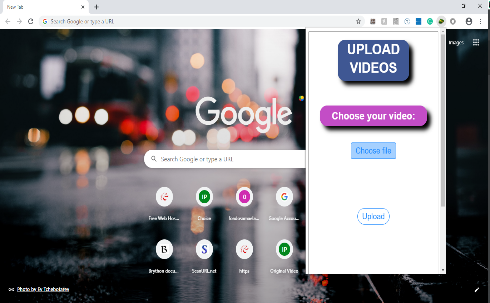
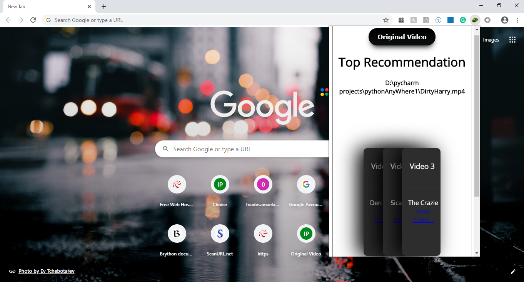
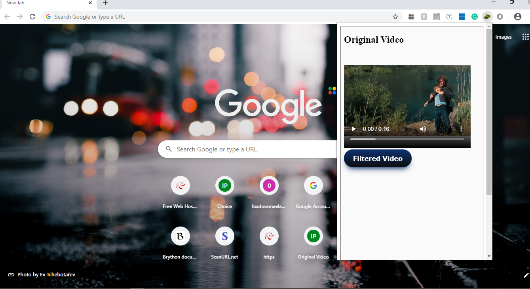
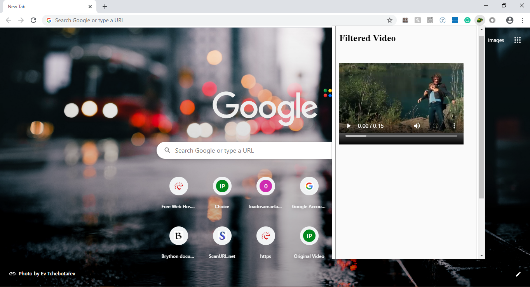
* 1. b) Non-functional requirements
  2. 1) Performance and Speed
  3. The recommender system must have high processing speed and performance, to give the user his video recommendation as with minimum delay.
  4. 2) Reliability
  5. The recommender system is reliable, where it provides the user with the most similar video to the one they provided on the first try, reducing the effort needed for the user to find similar videos.
  6. 3) Scalability
  7. The recommender system is scalable. The more scenes the user searches, the more resources it’ll have to recommend in the future.
  8. 4) Security and Safety
  9. The admin panel must be accessed with a password to ensure the protection of content.

**Section 4: System Design**

* 1. High level system architecture



* 1. User interfaces

4.3. Algorithmic components:

* YoloV3/ Darknet: Used for Training and Detecting objects in videos
* Cosine Similarity: Used to compare the content of the videos
* Spectral Clustering: Used to split videos into separate groups with their similar videos
* Convolutional Neural Network: Used for Training and Detecting type of Audio Files
* Recurrent Neural Network: Used for Training and Detecting type of Audio Files

4.4. Innovative aspects of the design

Our system is an extension to a browser which can be running in the background which makes it easier and flexible to use.

**Section 5: System Implementation**

5.1. Software platforms: Chrome Browser

Hardware platforms: None.

5.2. Software Development tools: Pycharm, Google Collaborator.

Software Development Languages: Python programming language, Flask framework, JavaScript, Html, CSS.

5.3. Open Source:

* Darknet YoloV3 (neural network for object detection).
* Online Web Service

5.4. Innovative aspects of the implementation: Labelling the detected objects from the video by the Yolo library and its frequencies to facilitate the process of getting similar videos. Making videos into clusters to separate videos into groups to get the related videos to the user’s video easily without comparing each individual video with the video the user uploaded.

Making the user insert a video URL from any website to be detected by the system and getting similar videos.

**Section 6: Other Relevant Issues and Challenges**

6.1. Technical

- In the beginning, when we want to build an offline version from the plugin (Desktop Application) the first problem we faced is the lack of the tools and the references to implement that version. There are no plugins to Desktop applications like what we want to do even the plugins that may have some common features is so complicated and hard to understand how it was implemented.

6.2. Other