Report

Lab Exercise/Assignment week : Lab 5

Lab Group (CL/01, CL/02,…) : Lab CL/01

Family name : Le

First name : Quang Long

Student number : 6482132

Email address : qll998@uowmail.edu.au

YouLink. An application on to gather Youtube’s videos data

Project Manager: Le Quang Long

Contents

[Requirements Management Plan 2](#_Toc38556871)

[Functional Specifications 3](#_Toc38556872)

[Project scope statement 5](#_Toc38556873)

[Product scope statement 5](#_Toc38556874)

[Project Deliverables 5](#_Toc38556875)

[User Acceptance Criteria 5](#_Toc38556876)

[Project Boundaries 5](#_Toc38556877)

[Project Assumption 5](#_Toc38556878)

[WBS 6](#_Toc38556879)

[WBS review 6](#_Toc38556880)

[WBS Dictionary Entry 7](#_Toc38556881)

# Requirements Management Plan

Requirement collection

|  |
| --- |
| * Analyze Youtube Videos Data * Analyze Metadata of Youtube Videos * Gather users opinions, what users want and need from questionnaire interviews and group sessions * Gather features and attractive points of other data scraping app and data scrapping services * Learning and understanding Youtube APIs |

Requirement Analyzation

|  |
| --- |
| * Gather data of customer questionnaire, gathering and grade their requirements from most needed to least needed * Base the requirements around Youtube videos data and functionalities and grade them. Compare it against the customer questionnaires requirements * Base the requirements on user usability and the software need to have. Compare it against two other requirements charts |

# Functional Specifications

Using FURPS+ model to classify requirements

* Functionality
* Usability
* Reliability
* Performance
* Security
* + extras

1. Functionality specification
   1. Application customers have to be divided up to 3 tiers Free, Paid and Premium with the order low to high
   2. Application have to gather daily data from user usages and their metadata
   3. Customer can customize the data and metadata they would like to gather
   4. Customers are allowed to mass gather data on video categories that they specify
   5. The application have to warn the user if the video contains explicit contents or is unavailable in their countries. Before extracting the data.
   6. The application must have access to all Youtube videos without any regional limits.
2. Usability Specification
   1. Application must be adaptable for a worldwide launch. Meaning that there have to be multiple versions to fit with countries legality
   2. Application must be able to adapt on mobile and desktop platforms
   3. Application must have to be usable both web and app version
   4. User payments and experience has to be streamlined and foolproof
3. Reliability Specifications
   1. Depending on users’ tier be request limit have to be the following
      1. Free tier : 2 requests at the same time
      2. Paid tier : 20 requests at the same time
      3. Premium tier: unlimited requests at the same time
   2. Same goes for request limit this concerns the region restrictions
      1. Free tier: videos produced by country’s content creator only
      2. Paid tier: videos produced by regions content creator only (Ex: Oceania, Asia, America, Europe etc…)
      3. Premium tier: no restriction, all videos are available
4. Performance Specifications
   1. Ensure that the application versions have 3 modes depending on user latency: no lag, low lag and high lag
   2. Add cloud requests system for premium users
5. Security Specifications
   1. Ensure that data gathered from use is secure
   2. Ensure that no request can be made until user is signed up and verified
   3. Ensure that Premium user connect to cloud is always SSH tunneled
   4. Ensure that user can not modify request and break the server
   5. DDOS proof
   6. Custom request proof
6. Extras
   1. Allow user to customize their application layout, coloring and format.

(dark mode, light mode, etc…)

* 1. Trial Premium/Paid version for users.

FURPS+ requirement approved by Long Le

# 

# Project scope statement

## Product scope statement

|  |
| --- |
| YouLink is a software project managed by Long Le and team in order to streamline, minify and change the way how youtube videos data are gathered. YouLink is expected to help user multiple ways on how to gather datas easily. Upon project finish the software is expected to launched worldwide and managed by the company. |

## Project Deliverables

|  |
| --- |
| Project Deliverables: User cases, User data report, WBS schedule, cost baseline, progress reports, user stories, ROI plan |

## User Acceptance Criteria

|  |
| --- |
| To be accepted by users at launch the product have to accomplish everything in usability and performance specifications stated in the FURPS model |

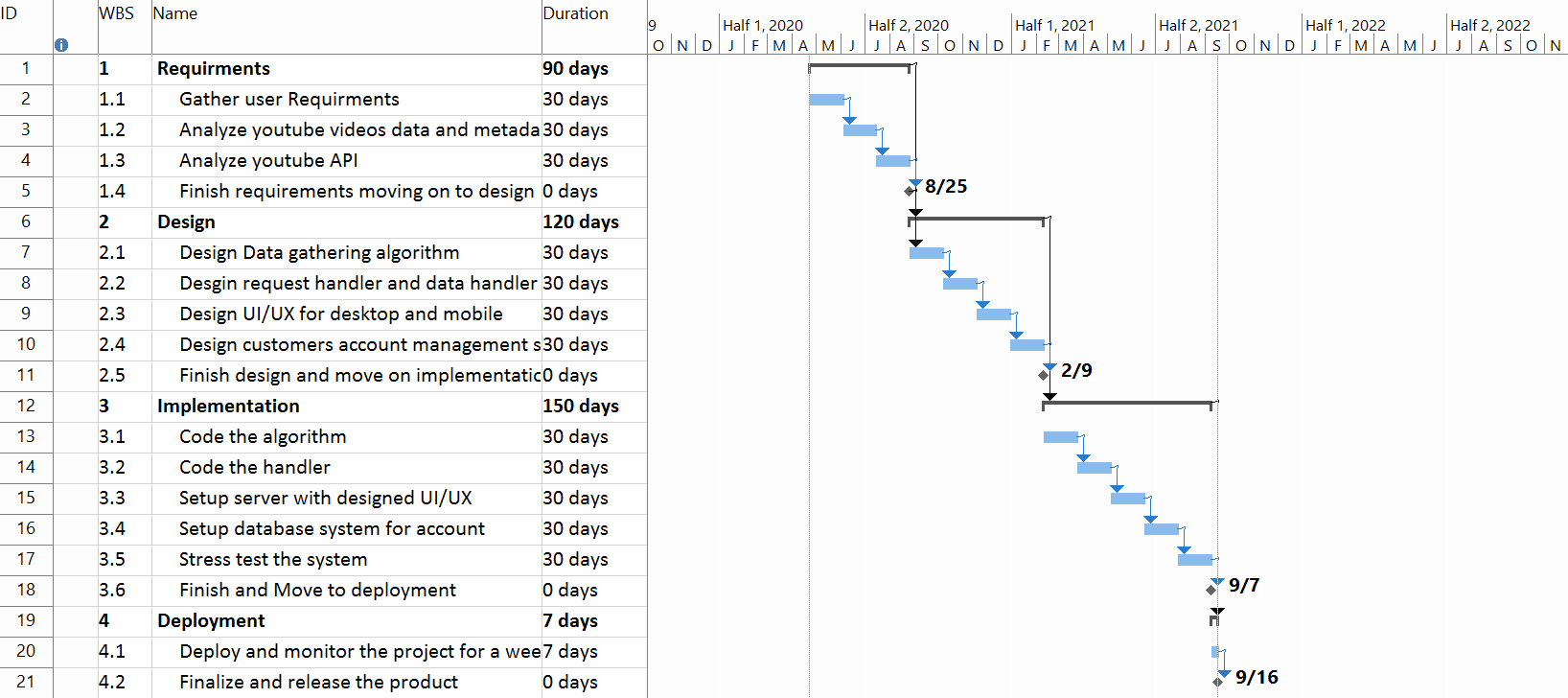
## Project Boundaries

|  |
| --- |
| Project is expected to complete within the requirements FURPS+ model and not create any extra requirements or unneeded requirements. Any decisions made to change, edit or add requirements or extra cost must be consulted with the project manager. |

## Project Assumption

|  |
| --- |
| Project is expected to be completed within 1 year and supposed that project cost is given without disruption. Project is assumed to be constantly monitored and reported. |

# WBS



# WBS review

By using waterfall model and taking the project step by step. We ensure the project stability, clarity and transparent both to the team and the sponsors. This will help to ensure project success.

# WBS Dictionary Entry

|  |  |
| --- | --- |
| WBS dictionary entry 22/04/2020 | |
| Project Title | YouLink: Youtube video data scrapping  application |
| WBS item number | 1.3 |
| WBS title | Analyze Youtube API |
| Description: Analyze Youtube API at <https://developers.google.com/youtube/v3>.  Analyzing Youtube API is an important step towards understanding system and software requirements. Our application might or might not depends on it. But from analyzing the API and see how YouTube videos data are organized we can move on replicating lots of their main functions or yet improve it to be better. Not only that we can aim towards serving semi-professionals user by creating composite functions and minify to just a click making the process of using Youtube API easier. Understanding how Youtube videos data is organized and managed is a big part of the requirements. Lack of knowledge on how it works can either slow or worse stop completely our project if our software can’t not go par to par to match with Youtube API. Hardware wise we can start using the API to check minimum system requirements that our software hypothetically required and keep moving up from there within limits. Our goal right now it to not take up more than twice the performances requirements of Youtube API. Our software hypothetical performances quality can be also based on Youtube API as long as the data is preserved we must deliver data successfully at least 95% of user requests. | |