COSC 4P02 Project Requirements Document
Web Summarizer and Shortener

1. Introduction

1.1. About this document

This document is used to analyze the detailed requirements(functional/non-functional/domain), expected system architecture, and the final goal of our group's project. This document will also introduce the tool that we use to manage projects.

1.2. Background of this project

Now, as the Internet becomes more and more developed, the information people have access to daily is increasing rapidly. The same goes for videos, we know that some videos are very long but do not have enough useful information. Many video creators churn out long but meaningless videos just for traffic. This situation makes it increasingly difficult for people to read and understand lengthy articles and news in a limited time. Advanced Internet users are beginning to pursue shorter and more effective information, which makes software like Web Summarizer and Shortener useful.

This is also the reason why our group decided to choose this project theme. The final goal of this project is to build a convenient, fast and powerful web/video content summary software that can also shorten web page URLs.

1.3. About our software

This software project is a web-side application that can summarize the different types of web pages and video files and shorten web links. The application will allow users to enter the web page or video link that they want the system to summarize in the input text box. After the system reads and analyzes the web page text or video input by the user, it will give users a shorter summary text that contains enough information for the user's reference. At the same time, if you are a user of the pro version, you can further customize the level of information summary (the initial plan is to divide it into three different levels: low/medium/high). At the same time, users of the pro version can also share the generated summary text and shortened links to social media (such as Google and Facebook) by using API access.

2. Requirements

2.1. Functional Requirements:

I. The system can read user input
II. The system can check whether the web page/video link entered by the user is valid

III. When the link entered by the user fails, the system can give a reminder
IV. The system can analyze user-entered web links and shorten them
V. The system can call one or more large language models to analyze and summarize the web page/video content provided by the users.
VI. The system allows pro version users to customize the level of summary (high/medium/low levels).
VII. The system allows any user to create a private account
VIII. The system allows pro version users to create accounts and log in using third-party social media accounts (such as Google and Facebook).
IX. The system allows pro version users to bind their created accounts with third-party social media accounts.
X. The system can use third-party social media API access
XI. The system allows pro version users to use API access to share the generated summary text and shortened web-links to third-party social media with one click.
XII. The system allows pro-version users to use the User Dashboard
XIII. User Dashboard can track activity
XIV. User dashboard can be used to manage shortened links
XV. User dashboard allows users to customize settings (e.g. UI font size/color)
XVI. The system can store short links and summary texts generated by users in the past and let pro version users access them
XVII. The system must have an API access port for other software or programs to access

2.2. Non-functional Requirements:

I. The system should be working with 7/24.
II. The system should have less than 1 hour of downtime when meeting problems.
III. The system should work on all kinds of browsers.
IV. The interface is reasonably designed and user-friendly.
V. Quick response after user input.

2.3. Domain Requirements:

I. The system needs to be *Data Encryption Standard (DES)* compliant

3. Tools

Due to the multiple requirements in this project, after discussions with the team members, to make the final released program stable and reliable, we decided to focus on only one function in each sprint. Fortunately, according to the course outline, this project has six requirements, allowing us to implement it in six work cycles. Because of that, we decided to use *Jira Software* to manage the user stories and all sprints.

4. Each Person's Contribution

William Li (6471254/wl18sc@brocku.ca/Product owner):
Host discussion, responsible for the program backend.

Xiaobin Liu (6246789/xl17gl@brocku.ca/Scrum Master):
Write this report, responsible for the program backend.

Di Wu (6345912/dw17kf@brocku.ca/developer team):
Participate in discussions, responsible for the front-end of the program.

Daniel Yang (6937601/dy19hh@brocku.ca/developer team):
Participate in discussions, responsible for the front-end of the program.

Yuchen Ding (6677967/yd18jd@brocku.ca/developer team):
Participate in discussions, responsible for the program backend.

Yuhao Cao (6594899/yc18do@brocku.ca/developer team):
Participate in discussions, responsible for the program backend.

5. Sprints

Sprint 1
Feature 1 - Short link creation
User Story:
As a social media influencer, I want to create engaging short links to my videos and blog posts so that they are easier to share in social media posts and video descriptions and attract a larger audience.

Product backlog:

1. Database creation
2. Set up query for holding information for user and their origin URL and shortened URL
3. Link Java program with database
4. Url shorten algorithm

Sprint 2
Feature 2 - API access
User story:
As a user I want to share the content to my Google or Facebook account directly from the website, so that it's sufficient for me which saves me time from editing my post.

As a developer, I want to integrate web summarizer and url shortener into our own applications, products, or services.

As a user, I want to integrate a link-shortening function into the app I design so that my users can quickly learn summaries of news articles.

Product backlog:

1. Frontend website GUI
2. Backend setup spring boot framework
3. Create API access

Sprint 3

Feature 3 - Social media integration

User story:

As a user I want to link my Google or Facebook account to this website so that I can share the summarized content to my account easily.

As a user, I would like to be able to track and view clicks and user interaction data for the shortened links I share through Google and Facebook to protect my privacy.

Product backlog:

1. Build a connection between our website and Google and Facebook to let users log in with their Google/Facebook accounts.
2. Output database post and get request log to track activities.

Sprint 4

Feature 4 - Summarize web content using AI models

User story:

As a user I would like to let AI summarize the text or video on the website for me so that it helps me understand

Product backlog:

1. Import AI model
2. Build an automated web scraper to gather information from the website
3. Content summarization algorithm
4. Output summarized content
5. Adjust the database to save summarized content as well

Sprint 5

Feature 5 - Custom summarization level

User story:

As a user I want to have 3 summarization levels, highly concise, medium concise and briefly summarized, so that I can use them for different purposes.

Product backlog:

1. Adjust frontend to make summarization-level options
2. Adjust the summarize algorithm to meet different summarization levels

Sprint 6

Feature 6 - User dashboard for registering users to track activity, manage links, and customization

User story:

As a user I want to save or delete my original and shortened link so that I'm able to manage my website links.

As a user I want to track and view my links' history activity so that I can protect my privacy.

As a user I want to customize my shortened URL link with my own decision so that I can make my URL easier to remember.

As a user I want to customize my web summarizer and shortener website like switching dark mode or light mode, so that I can customize the website in my favor.

Product backlog:

1. Adjust shortened the algorithm to let user customize their link
2. Let users have access to the database to let them manage their links
3. Create output for the link used in history
4. create dark/light mode switch button
5. create button that let user change font size

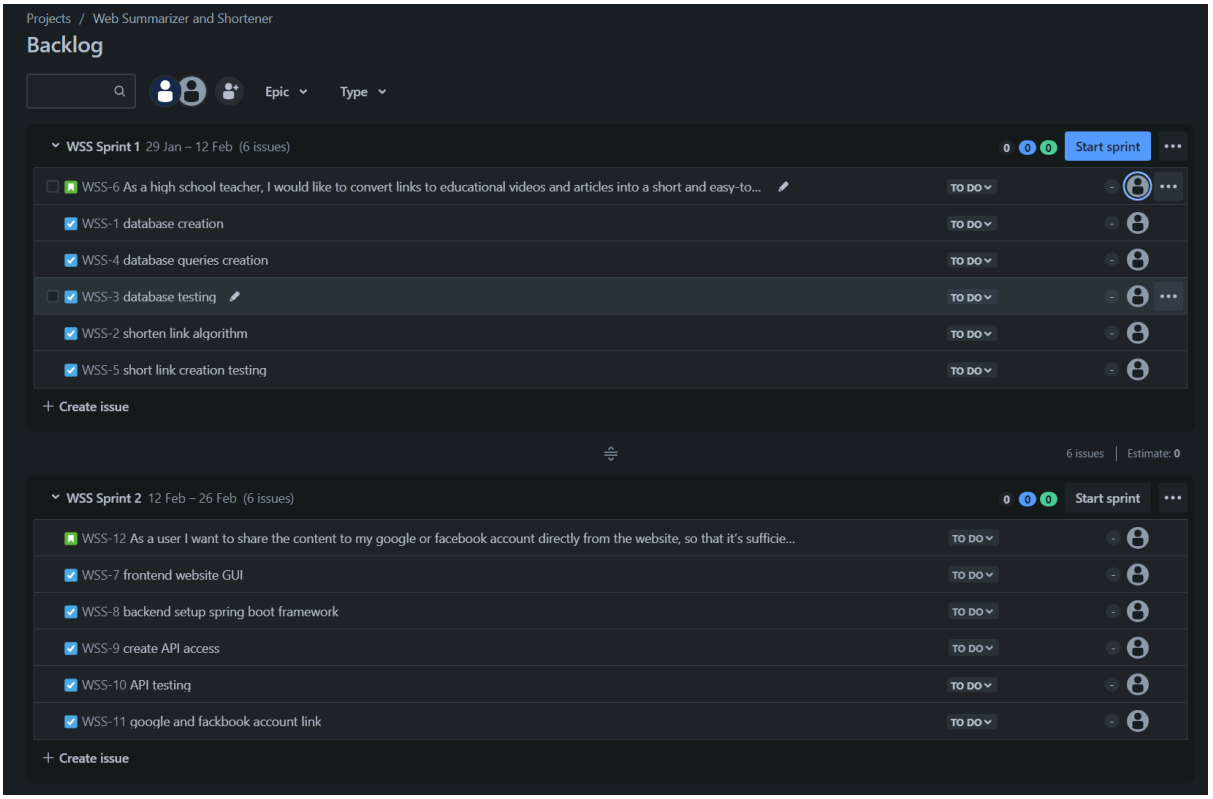
6. Expected System Architecture

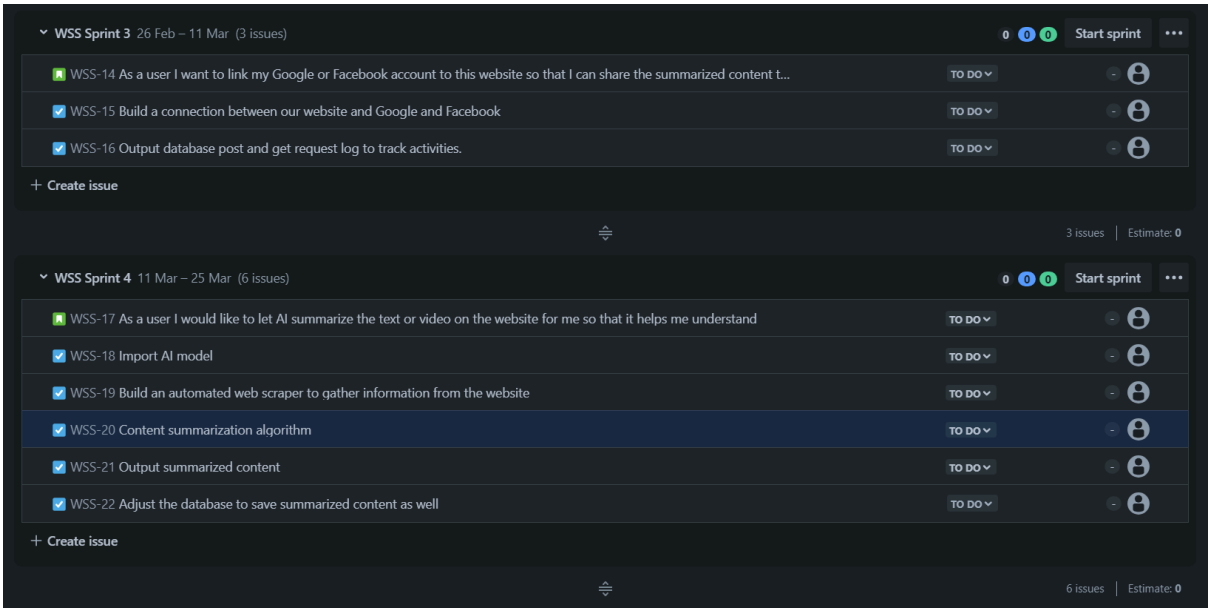
Based on the current discussion, we decided to use the *Model–view–controller(MVC) Model* to model our program to help us implement the code. **However, please note** that the feasibility of this model remains to be verified later.

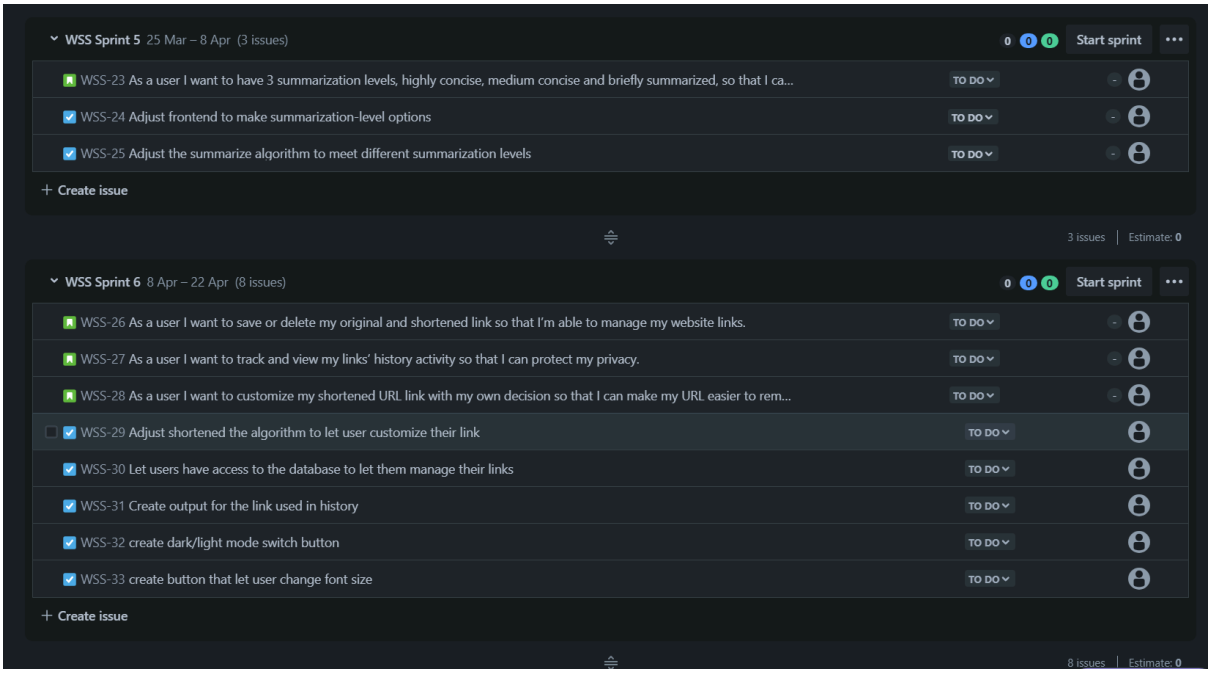
7. External link

Here is our GitHub page:
https://github.com/MTKdesu/Web-Summarizer-and-Shortener

8. Jira page screenshots






WSS Sprint 5 25 Mar – 8 Apr (3 issues)
0 0 0
Start sprint
WSS-23 As a user I want to have 3 summarization levels, highly concise, medium concise and briefly summarized, so that I ca...
TO DO
WSS-24 Adjust frontend to make summarization-level options
TO DO
WSS-25 Adjust the summarize algorithm to meet different summarization levels
TO DO
Create issue
3 issues
Estimate: 0
WSS Sprint 6 8 Apr – 22 Apr (8 issues)
0 0 0
Start sprint
WSS-26 As a user I want to save or delete my original and shortened link so that I'm able to manage my website links.
TO DO
WSS-27 As a user I want to track and view my links' history activity so that I can protect my privacy.
TO DO
WSS-28 As a user I want to customize my shortened URL link with my own decision so that I can make my URL easier to rem...
TO DO
WSS-29 Adjust shortened the algorithm to let user customize their link
TO DO
WSS-30 Let users have access to the database to let them manage their links
TO DO
WSS-31 Create output for the link used in history
TO DO
WSS-32 create dark/light mode switch button
TO DO
WSS-33 create button that let user change font size
TO DO
Create issue
8 issues
Estimate: 0