Release Planning Document

**GitHub Repository:** <https://github.com/MaulikMann/COSC-4P02.git>

|  |  |  |
| --- | --- | --- |
| Maulik Mann | 6994214 | [mm20as@brocku.ca](mailto:mm20as@brocku.ca) |
| Adrian Binu | 6970677 | [ab19xe@brocku.ca](mailto:ab19xe@brocku.ca) |
| Thanushan Pirapakaran | 6890206 | [tp19uy@brocku.ca](mailto:tp19uy@brocku.ca) |
| Vishant Khamar | 7062920 | [vk20if@brocku.ca](mailto:vk20if@brocku.ca) |
| Devaan Zastrow | 6932842 | [dz19xr@brocku.ca](mailto:dz19xr@brocku.ca) |
| Michael Boulos | 6973523 | [mb19ep@brocku.ca](mailto:mb19ep@brocku.ca) |
| Jason Hunter | 6744247 | [jh19pp@brocku.ca](mailto:jh19pp@brocku.ca) |
| Brett Terpstra | 6920201 | [bt19ex@brocku.ca](mailto:bt19ex@brocku.ca) |
|  |  |  |

# Roles:

Product Owner: Maulik Mann

Scrum Master: Vishant Khamar

UI Team: Jason Hunter, Thanushan Pirapakaran, Adrian Binu

Web Summarizer Team: Vishant Khamar, Michael Boulos, Brett Terpstra, Deevan Zastrow

URL Shortening: Maulik Mann

# User Stories & Tasks:

## Sprint 1:

## January 30th – February 13th:

### [A] As a user, I want to be able to input a URL and receive a shortened link, so that I can easily share web content with others:

TASKS:

1. Build the Interface for URL input (URL Shortening).
2. Use pyshortner API to implement URL shortening.

FEATURES:

1. To enter a valid URL, there will be an input area on the user interface.
2. Upon entering the URL, the system will have the capability to create a URL shortener for the supplied domain.
3. A component of the user interface that lets users copy the shortened URL that is created to the clipboard with only one click.

[B] As a user, I want to input a URL and receive a summary of the content or video:

TASKS:

1. Create a page that gives the option between the two types of content that can be summarized and shortened, text and video.
2. Add a button and URL input box specifically for webpage summarizing.
3. Add a button and URL input box specifically for YouTube video summarizing.
4. Create a result page that provides the summarized content and the shortened URL.
5. Integrate AI API to summarize the video transcript.
6. Integrate AI API to summarize other web content.

FEATURES:

1. The system will have a feature to retrieve the video or website content using the supplied URL.
2. A summarization algorithm that highlights the most important information in a brief overview of the webpage or video material.
3. A component of the user interface that presents the generated summary in an organized manner.

[C] As a user I would like to log in to a dashboard through Google and FB authentication so that I can track activity, manage links, and customize settings.

TASKS:

1. Create a page that allows users to login with a google account or Facebook account.
2. Integrate Google and Facebook authentication API

FEATURES:

1. Facebook and Google authentication options will be integrated into the login page.
2. Personalized user dashboard with selections and important information displayed.
3. Features of the user interface to handle shortened links, such as viewing, changing, and removing certain characters
4. A section of the dashboard that enables users to monitor link-related behaviour, such as the quantity of clicks.

[D] As a developer I would like to use lamma.cpp and whisper.cpp to implement the LLM/AI features such as video -> transcript and text -> summary conversion.

TASKS:

1. Integrate lamma.cpp to transcribe video content to text. Develop a mechanism to process and store the transcribed text.
2. Integrate whisper.cpp for text summarization. Create a module to generate concise summaries from textual content.
3. Develop and execute test cases to validate the accuracy and reliability of LLM/AI features. Ensure robustness through thorough testing, including edge cases.
4. Document the integration process of lamma.cpp and whisper.cpp. Provide clear documentation on how users can utilize video transcription and text summarization features.

FEATURES:

1. Users can transcribe video content to text using lamma.cpp. Processed transcripts are stored for future reference.
2. Users can generate concise summaries from textual content using whisper.cpp.
3. Detailed documentation guides users on utilizing video transcription and text summarization functionalities.

Sprint 2:

[A] As a user I would like to adjust the level of summarization so that can control how big I want the summarization to be

TASKS:

1. Implement a slider component in the user interface for adjusting summarization levels.
2. Allow users to manually enter the desired depth or length of the summary.

FEATURES:

1. The feature for this interface will include a slider in the user interface that lets users change how summarized information is.
2. Length will be measured as the number of words used in the summary.
3. It will allow users to manually enter how deep or how long they want the summary to go.

[B] As a pro user, I want a user dashboard where I can track my link creation activity, manage my links, and customize my settings.

TASKS:

1. Design and implement a dashboard page for pro users.
2. Implement features for identifying and verifying pro users during login.
3. Add link management tools for pro users to observe, modify, and remove links.

FEATURES:

1. There will be a system for identifying and verifying pro users when they log in.
2. Pro users can observe, modify, and remove links, among other link management tools.

Sprint 3:

[A] As a user, I want to be informed about the free and pro features, so I can make an informed decision about whether to upgrade to the pro version.

TASKS:

1. Design and implement a UI section offering a concise summary of free and pro features.
2. Include a graphic comparison table displaying features offered in the free and pro editions.

FEATURES:

1. There will be a portion of the UI that offers a concise summary of the pro and free features.
2. It will have a graphic comparison of the features offered in the free and pro editions, like a table
3. Have analytics for the shortened URLs

[B] As a pro user, I want a history of my summarized URLs

TASKS:

1. Create a special section in the user dashboard to display the history of summarized URLs.
2. Include information such as the original URL, summarizing date, and prepared summary for each URL.

FEATURES:

1. The system will include the creation of a special section in the user dashboard to hold and show the history of summarized URLs.
2. That section will also include information about the original URL, the summarizing date, and the summary which was prepared for each summarized URL.

Sprint 4:

[A] As a user I want a dark theme to allow easier visibility on my eyes

TASKS:

1. Implement a feature to switch between bright and dark themes in the user interface.
2. Add functionality to remember the user's preferred theme for subsequent visits.

FEATURES:

1. There will be a feature which will have the ability to switch between bright and dark themes in the user interface.
2. The system features that allow the user's preferred theme to be remembered for subsequent visits.

[B] As a user I want a text-to-speech capability for the summary, so I can listen to it instead of reading it

TASKS:

1. Implement an option in the user interface to enable text-to-speech for the summary.
2. Provide controls for play, pause, and stop options during text-to-speech playback.

FEATURES:

1. There will be an option in the user interface to turn on text-to-speech for the summary created.
2. It will also include an option for users to control the text-to-speech playback, there are play, pause, and stop options.

Document created and edited by Jason Hunter and Vishant Khamar.