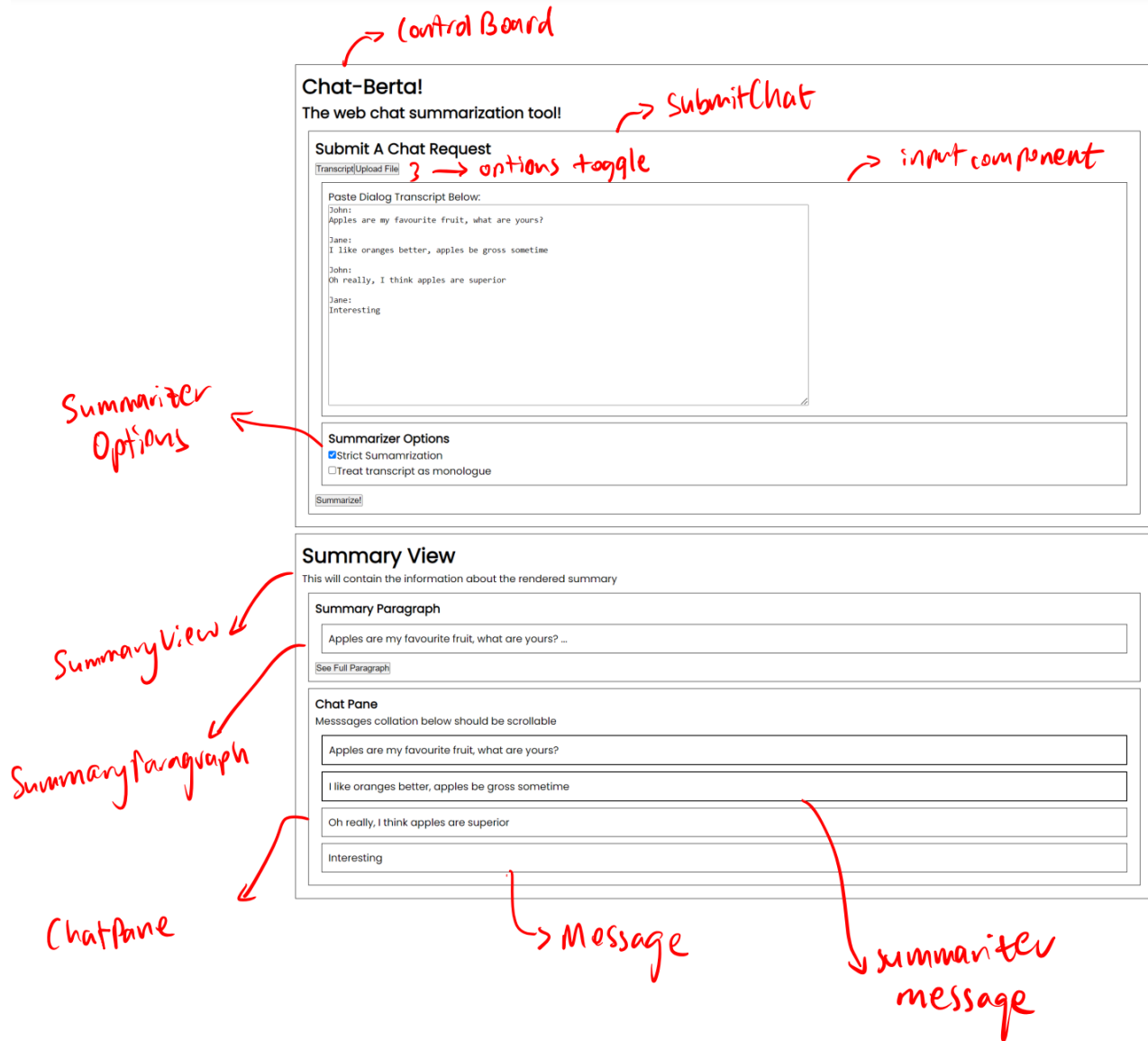


# Front-End Design Specifications

## Current Layout

First, a quick overview of the current layout of the application:



The labelled items above are the React components used in the source code. Each of these components is wrapped with a `div` container with the class name `basic-container`, and that is what generates the simple black borders on the side.

The message components with thicker lines are summarizer messages. (That is, messages included in the summary paragraph)

The OptionsToggle will be implemented as a checkbox component, in a few days but as for now it is just implemented as two buttons. (The CSS can design it as a checkbox).

We also have different components for different states of the `SummaryView` content.

When a request has not been made, the `SummaryContentUnset` component is in use:

## Summary View

This will contain the information about the rendered summary

No Content Available

When a request is made and is loading, we use the `SummaryContentLoading` component:

## Summary View

This will contain the information about the rendered summary

Loading...

We also have an error component `SummaryContentError` that is rendered if the summary request fails for some reason:

## Summary View

This will contain the information about the rendered summary

Error Occured!  
TypeError: Failed to fetch

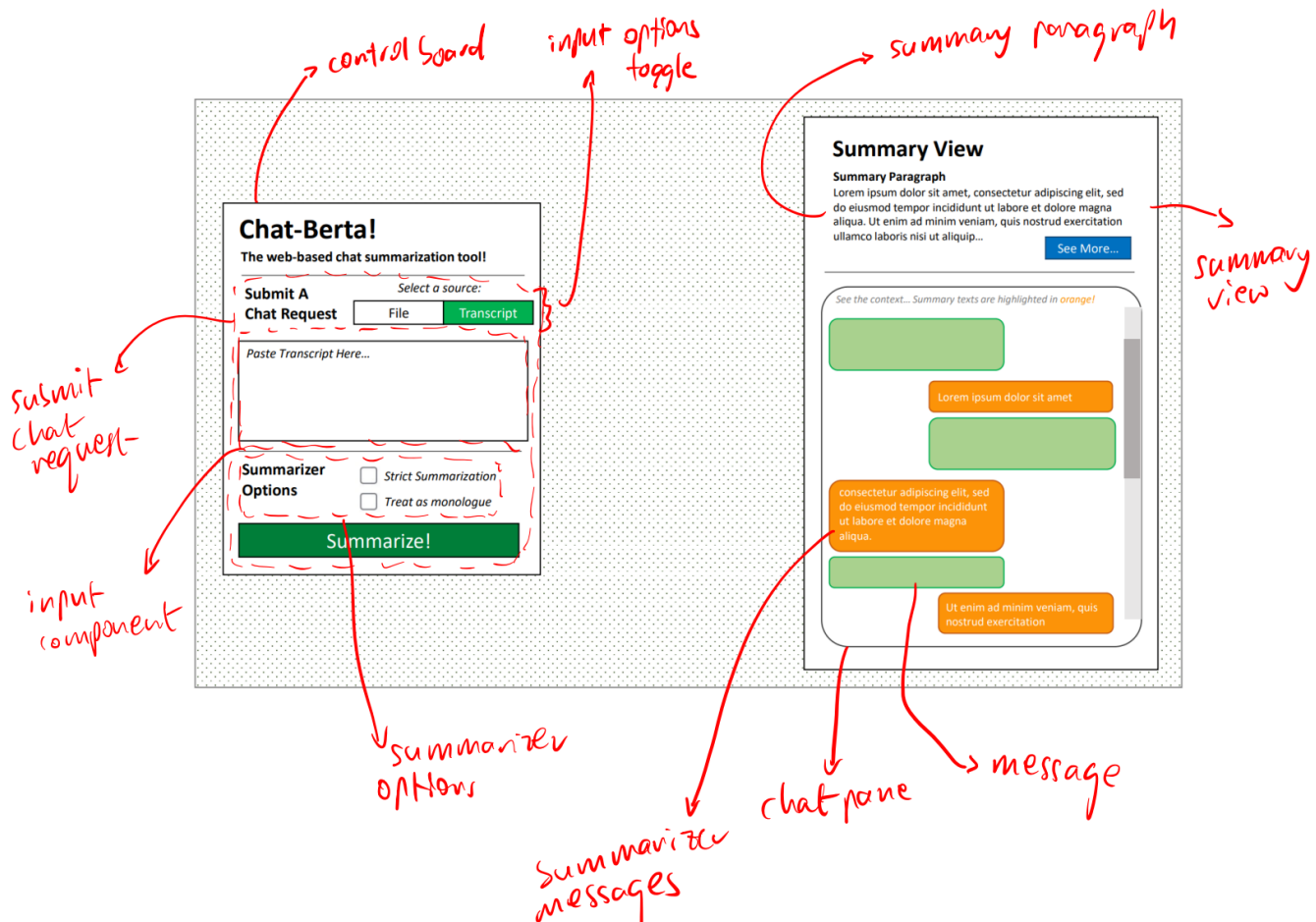
The idea with this frontend CSS design is that you can simply make the CSS classes to implement the frontend design we desire for each component. I can then use those class names in the React source code to implement the different required items.

## Desired Design

The following diagram outlines the required design, highlighting the different components in the pretty version:

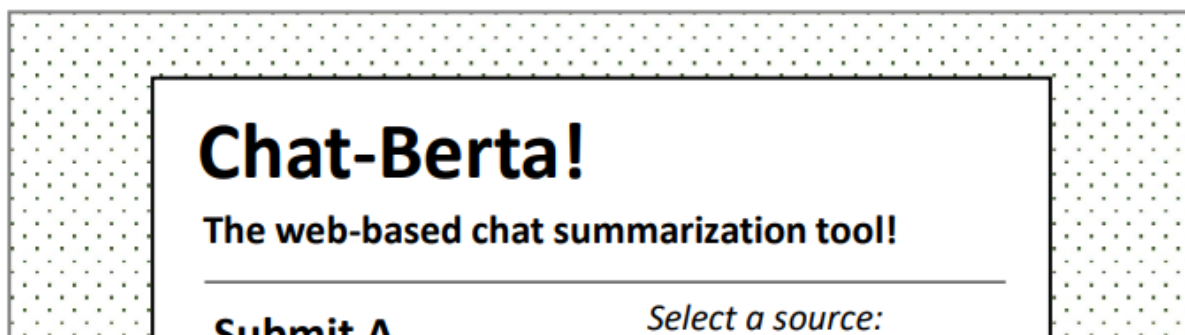
## Main View

The main view will look something similar to this, I have labelled the required components in the below picture.



## Phone Layout

On a phone, the SummaryView component should automatically fall under the ControlBoard component (most likely implemented with CSS Flex):



Paste Transcript Here...

### Summarizer Options

- ☐ Strict Summarization
- ☐ Treat as monologue

Summarize!

## Summary View

### Summary Paragraph

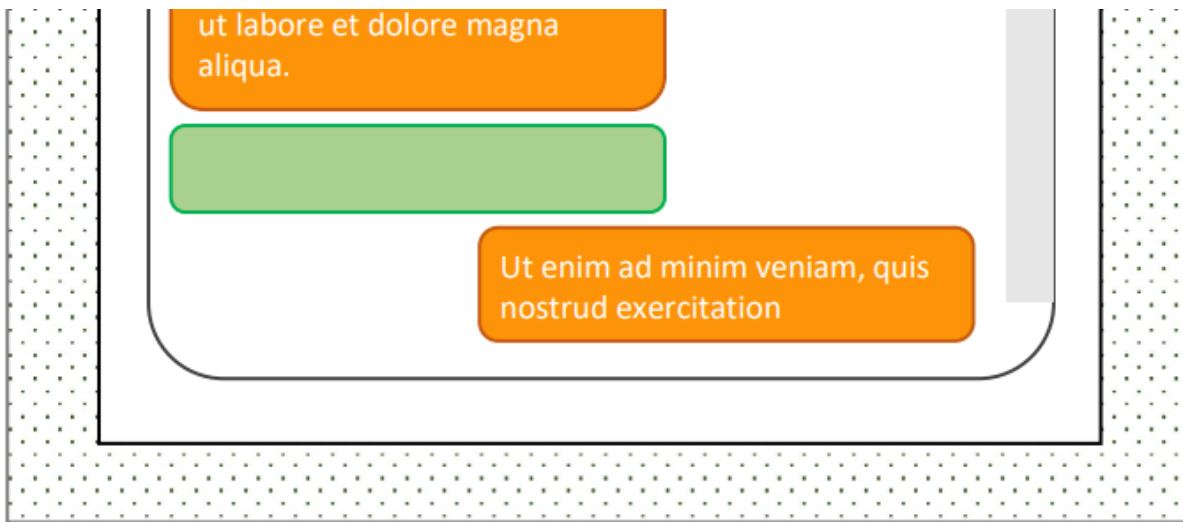
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip...

See More...

See the context... Summary texts are highlighted in orange!

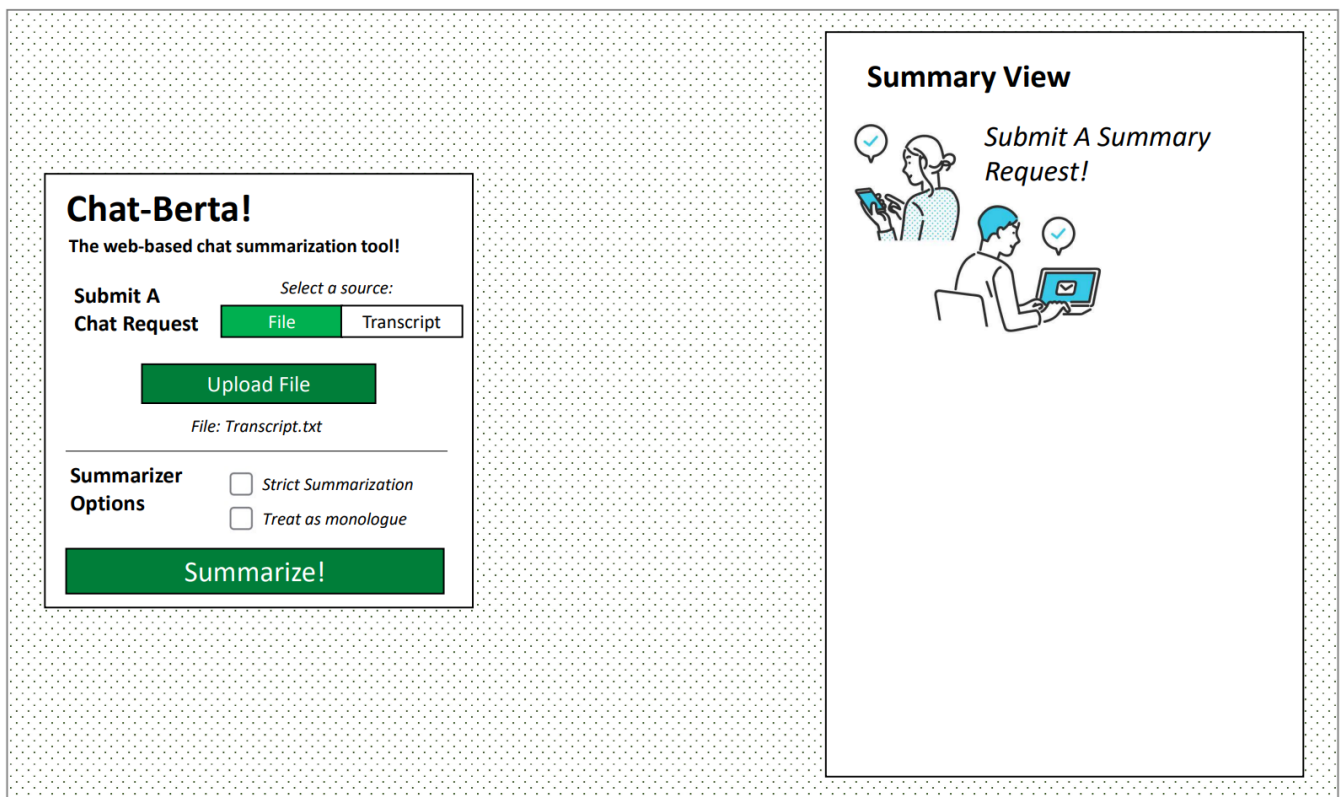
Lorem ipsum dolor sit amet

consectetur adipiscing elit, sed  
do eiusmod tempor incididunt



## Initial Page

The Initial page should have some basic info on the SummaryView component such as:



We also have the different layouts for the different input options, the above is for uploading a file, the below is for putting a transcript:

## Chat-Berta!

The web-based chat summarization tool!

Submit A  
Chat Request

Select a source:

File

Transcript

Paste Transcript Here...

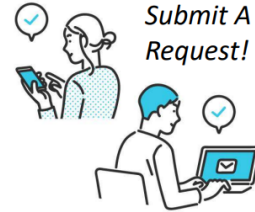
Summarizer  
Options

☐ Strict Summarization

☐ Treat as monologue

Summarize!

## Summary View



Submit A Summary  
Request!

## Loading Page

The loading page will have a simple loading gif (not rendered below):

## Chat-Berta!

The web-based chat summarization tool!

Submit A  
Chat Request

Select a source:

File

Transcript

Paste Transcript Here...

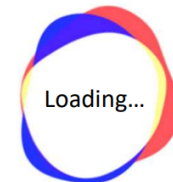
Summarizer  
Options

☐ Strict Summarization

☐ Treat as monologue

Summarize!

## Summary View





# Error Page

If the summary request failed, we can have a simple error page that returns the error information:

## Chat-Berta!

The web-based chat summarization tool!

Submit A Chat Request

Select a source:

FileTranscript


Paste Transcript Here...

Summarizer Options

☐ Strict Summarization☐ Treat as monologue

Summarize!

Summary View



Something went wrong.

Error: API Call Failure