Table of Contents

Contents

[1. Introduction 3](#_Toc146719764)

[1.1 Purpose 3](#_Toc146719765)

[1.2 Project Scope 3](#_Toc146719766)

[2. Overall Description 4](#_Toc146719767)

[2.1 Product Perspective 4](#_Toc146719768)

[3. Why do we need Phase 2 for Screen Genius 4](#_Toc146719769)

[4. Integrations 5](#_Toc146719770)

[4.1 Notes Apps 5](#_Toc146719771)

[4.2 Productivity Apps 5](#_Toc146719772)

[4.3 Social Media 5](#_Toc146719773)

[4.4 Chat GPT & BARD 5](#_Toc146719774)

[4.5 Messaging Apps 5](#_Toc146719775)

[4.6 Mail Clients 6](#_Toc146719776)

[5. Specific Requirements 7](#_Toc146719777)

[5.1 Integration with Notes App 7](#_Toc146719778)

[5.2 Integration with Productivity Apps 7](#_Toc146719779)

[5.3 Integration with Social Media 7](#_Toc146719780)

[5.4 Integration with Chat GPT & BARD 7](#_Toc146719781)

[5.5 Integration with Messaging Apps 7](#_Toc146719782)

[5.6 Integration with Mail Client 7](#_Toc146719783)

[5.7 Administrative Features 8](#_Toc146719784)

[5.8 User Management 8](#_Toc146719785)

[5.9 Additional Screenshot/Video Features 8](#_Toc146719786)

[5.10 Q & A Feedback 8](#_Toc146719787)

[6. User Interface Steps 9](#_Toc146719788)

[7. User Interface Display and Pop Ups 9](#_Toc146719789)

[8. Integration API’s 9](#_Toc146719790)

[9. Functional Requirements 10](#_Toc146719791)

[9.1 Notes Apps Integration 10](#_Toc146719792)

[9.1.1 Evernote 10](#_Toc146719793)

[9.1.2 OneNote 11](#_Toc146719794)

[9.1.3 Notion 12](#_Toc146719795)

[9.2 Productivity Apps Integration 13](#_Toc146719796)

[9.2.1 Microsoft Teams 13](#_Toc146719797)

[9.2.2 Dropbox 13](#_Toc146719798)

[9.2.3 Google Drive & Docs 14](#_Toc146719799)

[9.2.4 SharePoint 16](#_Toc146719809)

[9.2.5 Zapier 16](#_Toc146719810)

[9.2.6 Slack 17](#_Toc146719811)

[9.3 Social Media Integration 19](#_Toc146719812)

[9.3.1 Twitter 19](#_Toc146719813)

[9.3.2 Facebook 21](#_Toc146719814)

[9.3.3 LinkedIn 22](#_Toc146719815)

[9.3.4 Instagram 22](#_Toc146719816)

[9.3.5 YouTube 23](#_Toc146719817)

[9.4 ChatGPT & BARD Integration 24](#_Toc146719818)

[9.4.1 ChatGPT 24](#_Toc146719819)

[9.4.2 BARD 26](#_Toc146719820)

[9.5 Messaging Integration 26](#_Toc146719821)

[9.5.1 WhatsApp 26](#_Toc146719822)

[9.5.2 Telegram 28](#_Toc146719823)

[9.6 Mail Clients Integration 29](#_Toc146719824)

[9.6.1 Gmail 29](#_Toc146719825)

[9.6.2 Outlook 29](#_Toc146719826)

[9.7 Screenshot URL Summary Generation 30](#_Toc146719827)

[9.8 AdSense and Ads Placement 30](#_Toc146719828)

[9.9 Super Admin – Management Tool 30](#_Toc146719829)

[9.9.1 User Management 30](#_Toc146719830)

[9.9.2 Image & Video Management 30](#_Toc146719831)

[9.9.3 Subscription Management 30](#_Toc146719832)

[9.9.4 Pricing Plans Management 30](#_Toc146719833)

[9.9.5 Extension’s Static UI & Text 30](#_Toc146719834)

[9.9.6 General Settings 30](#_Toc146719835)

[9.9.7 Pro Features Settings 30](#_Toc146719836)

[10. Screen Genius – User Management 31](#_Toc146719837)

[10.1 Pricing Plans 31](#_Toc146719838)

[10.2 Membership 31](#_Toc146719839)

[10.3 Pro & Free User Roles 31](#_Toc146719840)

[10.4 Ads Free 31](#_Toc146719841)

[10.5 Watermark 31](#_Toc146719842)

[10.6 Pro Features 31](#_Toc146719843)

[10.7 Unsubscribe 32](#_Toc146719844)

[10.8 Account Management 32](#_Toc146719845)

[11. Sharing Features 32](#_Toc146719846)

[11.1 Remember Email 32](#_Toc146719847)

[11.2 Email Suggestions 32](#_Toc146719848)

[12. Screenshot/Video additional features 33](#_Toc146719849)

[12.1 Screenshot Background (Gradient) 33](#_Toc146719850)

[12.2 Screenshot Padding 33](#_Toc146719851)

[12.3 Emoji & Content Creator Elements 33](#_Toc146719852)

[13. Q & A Feedback 33](#_Toc146719853)

[14. Non-Functional Requirements 33](#_Toc146719854)

[14.1 Performance 33](#_Toc146719855)

[14.2 Security 33](#_Toc146719856)

[14.3 Usability 34](#_Toc146719857)

[14.4 Reliability 34](#_Toc146719858)

[14.5 Compatibility 34](#_Toc146719859)

[14.6 Scalability 34](#_Toc146719860)

[15. Design & Implementation Constraints 34](#_Toc146719861)

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Changes Made** | **Version** |
| Vijayalakshmi Mali | 22nd Sep 2023 |  | Draft |
| Vijayalakshmi Mali | 8th Dec 2023 | 1. Added detailed requirements for Adsense and pricing plans |  |

# Introduction

This document provides additional features of Screen Genius Chrome extension that serves as a 2-in-1 screenshot and screen record tool.

The Screen Genius Chrome extension is a standalone tool that integrates with the user's web browser. It allows users to capture screenshots and record screens while providing additional functionality.

Screen Genius Phase 2 aims to facilitate the capture and sharing of screenshots and videos with multiple integration possibilities into popular productivity apps, social media platforms, messaging apps, mail clients, and more. Additionally, it includes functionality for managing user subscriptions, user roles, and other administrative features.

## Purpose

The purpose of this Software Requirements Specification (SRS) document is to provide a comprehensive overview of the requirements for the development of "ScreenGenius Phase 2”

This document provides additional features of Screen Genius Chrome extension that serves as a 2-in-1 screenshot and screen record tool.

This phase focuses on adding several integrations, including Notes Integration, Manual Upload, Social Media Sharing, AI Integration, Messaging Apps Integration, and Email Integration, User Management, etc.

## Project Scope

The Phase 2 development aims to enhance the functionality of the extension by integrating with various external services and tools. This document outlines the specific requirements for each integration and provides a comprehensive understanding of the extension's capabilities.

* Integrate with notes app tools like Evernote, One Note and Notion which should allow users to save screenshots/videos directly to the respective application.
* Integrate with Productivity apps like Teams, GDrive, Docs, Dropbox, Sharepoint, Zapier and Slack. This will allow users to share screenshots/videos directly to the application.
* Integrate with Social Media platforms like Twitter, Facebook, LinkedIn, Instagram and You tube.
* Integrate with AI tools like ChatGPT and BARD. This will allow users to send the screenshot as input and receive a response.
* Integrate with messaging apps like Whatsapp, Telegram. This will allow users to share screenshots/videos directly to the application.
* Integrate with apps like Gmails and Outlook. This will allow users to share screenshots/videos directly to the mail.
* Adsense and Ads Placement – To show ads at appropriate locations.
* Super Admin Management – Develop User Management system in the application that allows universal control to manage user access/Image-Video Management/Subscription Management/Pricing/Extension’s UI/general settings and Pro feature settings etc.
* Add pricing plans and payment plans and watermark to the photos/videos. Users shall subscribe/unsubscribe plans and edit their profile picture, name, password etc,
* Add additional features to screenshot/Video such Gradient, Image padding, Emoji’s and content creator elements.
* User emails shall be recorded in the system and shall suggest emails
* Add QA and client feedback

# Overall Description

## Product Perspective

Screen Genius is a standalone chrome extension designed to provide users with the capability to capture and share screenshots and videos seamlessly with various integration possibilities. It interfaces with third-party applications and APIs to enhance user productivity.

# Why do we need Phase 2 for Screen Genius

In Phase 2 the focus is more aligned towards enhancing user experience and expanding the capabilities of the application. Building upon the foundation established in Phase 1, this phase introduces a range of new functionalities and improvements aimed at providing users with a more comprehensive and feature-rich screen capture and editing solution.

Key Enhancement in Phase 2:

1. Integrations
2. User Management
3. Color Grading
4. Better UI/UX experience

# Integrations

## Notes Apps

* **Evernote Integration**: Users can save screenshots/videos directly to Evernote for documentation and note-taking purposes.
* **OneNote Integration**: Users can save screenshots/videos directly to OneNote for note-taking and collaborative work.
* **Notion Integration**: Users can save screenshots directly to Notion for note-taking, planning, and documentation.

## Productivity Apps

* **Teams Integration**: Users can share screenshots/videos directly in Microsoft Teams conversations or channels for collaboration and problem-solving.
* **Dropbox Integration**: Users can save screenshots/videos directly to a Dropbox folder for file management and sharing.
* **Google Drive & Docs Integration**: Users can save screenshots/videos directly to Google Drive or insert them into Google Docs for collaboration and document creation.
* **SharePoint Integration**: Users can save screenshots/videos directly to a SharePoint site for collaboration and document management.
* **Zapier Integration**: Users can trigger automated workflows (Zaps) using captured screenshots/videos.
* **Slack Integration**: Users can share screenshots/videos directly to Slack channels or with individuals for quick collaboration.

## Social Media

* **Twitter Integration:** Users can share screenshots/videos directly to twitter for social media posting.
* **Facebook Integration**: Users can share screenshots/videos directly to Facebook for social media posting.
* **LinkedIn Integration:** Users can share screenshots/videos directly to LinkedIn for social media posting.
* **Instagram Integration:** Users can share screenshots/videos directly to Instagram for social media posting (if supported by Instagram API).
* **YouTube Integration:** Users can upload videos directly to YouTube for video sharing and content creation.

## Chat GPT & BARD

* **ChatGPT Integration:** Users can use the ChatGPT API to send captured screenshots for generating responses.
* **BARD Integration:** Users can use the BARD API to send captured screenshots for generating responses.

## Messaging Apps

* **WhatsApp Integration:** Users can send screenshots/videos directly to WhatsApp contacts for quick sharing and communication (if supported by WhatsApp Web/Desktop).
* **Telegram Integration:** Users can send screenshots/videos directly to Telegram chats for quick sharing and communication.

## Mail Clients

* **Gmail Integration:** Users can attach screenshots/videos directly to Gmail emails for communication and information sharing.
* **Outlook Integration:** Users can attach screenshots/videos directly to Outlook emails for communication and information sharing.

# Specific Requirements

The following use cases describe the primary interactions users can perform with Screen Genius:

## Integration with Notes App

* Users can select the desired Notes App integration (Evernote, OneNote, Notion) after capturing a screenshot.
* Users can authenticate with the respective API.
* Users can use the API to create a new note or page and save the screenshot/video.

## Integration with Productivity Apps

* Users can select the desired Productivity App integration (Teams, Dropbox, Google Drive & Docs, SharePoint, Zapier, and Slack) after capturing a screenshot/video.
* Users can authenticate with the respective API.
* Users can use the API to perform actions like sharing in a conversation, saving to a specific folder, triggering a Zap, or sending to a channel/person.

## Integration with Social Media

* Users can select the desired Social Media integration (Twitter, Facebook, LinkedIn, Instagram, and YouTube) after capturing a screenshot/video.
* Users can authenticate with the respective API.
* Users can use the API to post the screenshot/video on the selected social media platform.

## Integration with Chat GPT & BARD

* Users can select either ChatGPT or BARD integration after capturing a screenshot.
* Users can use the selected API to send the screenshot for generating responses.

## Integration with Messaging Apps

* Users can select the desired Messaging App integration (WhatsApp, Telegram) after capturing a screenshot/video.
* Users can authenticate with the respective API.
* Users can use the API to send the screenshot/video to a contact or chat.

## Integration with Mail Client

* Users can select the desired Mail Client integration (Gmail, Outlook) after capturing a screenshot/video.
* Users can authenticate with the respective API.
* Users can use the API to compose an email with the screenshot/video attached.

## Administrative Features

* Super admin can manage user accounts, including activation, suspension, deletion, and updates.
* Super admin can manage user subscriptions, allowing for cancellations or extensions.
* Super admin can create, edit, or delete pricing plans.
* Super admin can configure extension's static UI elements and text.
* Super admin can set general and pro features settings.

## User Management

* Users can select pricing plans and manage their memberships.
* Users can have different roles based on their subscription, with varying access to features.
* Users may encounter ads, and free users may need to watch ads before downloading high-resolution screenshots or videos.
* Watermarking options are available for free and pro users.

## Additional Screenshot/Video Features

* Users can add background colors, gradients, or images to screenshots (pro feature).
* Users can adjust screenshot padding.
* Users can add emoji’s and content creator elements to screenshots (pro feature).

## Q & A Feedback

* QA processes and client feedback will be incorporated to address any issues or make minor updates as needed.

# User Interface Steps

The user interface steps for each integration will include the following common steps:

* Authentication with the respective API.
* Use of API commands to perform actions related to screenshots/videos.

# User Interface Display and Pop Ups

* For each integration, a dropdown menu will display the available integration options.
* Popups may appear for selecting specific options within the integration, such as choosing a notebook, folder, or channel.
* While logging into other applications from the extension, if the user's authentication fails, an error message is displayed, and the user is prompted to retry the authentication process.
* If the selected API is unavailable or returns an error during the sharing process, an error message is displayed, and the user is given the option to retry or report the issue.
* If the user cancels the sharing process at any point, the extension returns to its default state, allowing the user to take other actions or capture additional screenshots.
* Users shall receive notifications when screenshot captures or screen recordings are completed successfully, and in case of any errors or interruptions during the process.

# Integration API’s

Screen Genius shall integrate with the following external APIs for third-party application integration:

* Evernote API
* OneNote API
* Notion API
* Microsoft Teams API
* Dropbox API
* Google Drive API
* Google Docs API
* SharePoint API
* Zapier API
* Slack API
* Twitter API
* Facebook API
* LinkedIn API
* Instagram API
* YouTube API
* ChatGPT API
* BARD API
* WhatsApp API (if supported by WhatsApp Web/Desktop)
* Telegram API
* Gmail API
* Outlook API

# Functional Requirements

In this Software Requirements Specification (SRS), we have included comprehensive API references to empower the developers. These references serve as an invaluable resource, offering clear and detailed insights into the functionality and usage of APIs. By providing this documentation, we aim to facilitate seamless integration and development, enabling our team to leverage the full potential of our software with ease and confidence.

# Notes Apps Integration

# Evernote

The following functional requirements outline the specific steps and interactions between the user, the ScreenGenius extension, and the Evernote API for sharing a screenshot directly to Evernote.

Step 1: User captures a screenshot/video using ScreenGenius Extension

Step 2: User clicks on edit Video/Screenshot and clicks on Share Video and selects Evernote.

Step 3: Authentication with Evernote API

**Condition 1:** If the user has not previously authenticated with the Evernote API:

* The extension prompts the user to authenticate with their Evernote account.
* The user clicks on the "Authenticate with Evernote" option.
* The extension opens a new window or tab with the Evernote authentication page.
* The user provides their Evernote login credentials and grants necessary permissions to the extension.
* Upon successful authentication, the extension receives an authentication token from Evernote.

**Condition 2:** Previously Authenticated User

* The extension uses the received authentication token to interact with the Evernote API securely.

**API Reference**:

| **Module** | **Services** | **Purpose** |
| --- | --- | --- |
| NoteStore | createNote | To create a new note with the screenshot or video capture. |
|  | updateNote | To update an existing note with new captures. |
|  | getNote | To retrieve a specific note by it’s ID. |
|  | getNoteContent | To retrieve the content of a specific note. |
|  | getResource | To get the binary content of a resource, which could be your screenshot or video. |
|  | getResourceData | To get the binary data of a resource, which could be your screenshot or video. |
|  | createNotebook | To create a new notebook where you can store the captures. |
|  | updateNotebook | To update the properties of an existing notebook. |
|  | getDefaultNotebook | To get the default notebook for the authenticated user. |
|  | listNotebooks | To get a list of all notebooks owned by the user. |
|  | shareNote | To share a note with others. |
|  | stopSharingNote | To stop sharing a note that was previously shared. |
| UserStore | getUser | To get information about the authenticated user. |
|  | getUserUrls | To get URLs for the user's profile, photo, etc. |
| Types | Note | Represents a single note. |
|  | Resource | Represents a piece of data that is associated with a note, like your screenshot or video. |
|  | Notebook | Represents a collection of notes. |
|  | SharedNote | Represents the sharing state of a note. |
|  | User | Represents an Evernote user. |

Step 4: The extension provides the user with a dropdown menu or dialog to select the destination notebook within Evernote where the screenshot should be saved.

Dropdown Options:

* Create New Note
* Notebook – select notebook that were already created
* Tasks
* Search for Notes

Step 5: The user should be able to preview the video/screenshot and should be able to share to selected notebook folder.

Step 6: The user's screenshot is successfully saved in their Evernote account within the selected notebook.

Step 7: The extension provides feedback to the user, confirming the successful sharing of the screenshot.

# OneNote

The following functional requirements outline the specific steps and interactions between the user, the ScreenGenius extension, and the One Note API for sharing a screenshot directly to One Note.

Step 1: User captures a screenshot/video using ScreenGenius Extension

Step 2: User clicks on edit Video/Screenshot and clicks on Share Video and selects One Note.

Step 3: Authentication with One Note

**Condition 1:** If the user has not previously authenticated with the One Note:

* The extension prompts the user to authenticate with their One Note account.
* The user enters Microsoft mail id clicks submit🡪 A dialogue box to enter password pop ups, user enters password and clicks on submit.
* The user provides their One Note/Microsoft login credentials and grants necessary permissions to the extension.
* Upon successful authentication, the extension receives an authentication token from Evernote.

**Condition 2:** Previously Authenticated User

* The extension uses the received authentication token to interact with the One Note API securely.

Step 4: The extension provides the user with a dropdown menu or dialog to select the destination notebook within One Note where the screenshot should be saved.

Dropdown Options:

* Select Notebook (This dropdown shall show all the notebooks user have created in One Note)
* Add a notebook (User shall be able to create a notebook and save the screenshot)

Step 5: The user's screenshot is successfully saved in their One Note account within the selected page/task.

Step 6: The extension provides feedback to the user, confirming the successful sharing of the screenshot.

**API Reference link for One Note**: [Link](https://docs.axway.com/bundle/Amplify_Integration_Builder_allOS_en/page/microsoft_onenote_api_documentation.html)

# Notion

The following functional requirements outline the specific steps and interactions between the user, the Screen Genius extension, and the Notion API for sharing a screenshot directly to Notion.

Step 1: User captures a screenshot/video using Screen Genius Extension

Step 2: User clicks on edit Video/Screenshot and clicks on Share Video and selects Notion.

Step 3: Authentication with Notion

**Condition 1:** If the user has not previously authenticated with the Notion:

* The extension prompts the user to authenticate with their Notion account.
* The user enters Gmail credentials🡪 A dialogue box to enter password pop ups, user enters password and clicks on submit.
* The user provides their Notion/Gmail login credentials and grants necessary permissions to the extension.
* Upon successful authentication, the extension receives an authentication token from Notion.

**Condition 2:** Previously Authenticated User

* The extension uses the received authentication token to interact with the Notion API securely.

Step 4: The extension provides the user with a dropdown menu or dialog to select the destination notebook within Notion where the screenshot should be saved.

Dropdown Options:

* Select Page (This dropdown shall show all the pages user have created in Notion)
* Task List
* Add a page (User shall be able to create a page and save the screenshot)

Step 5: The user's screenshot is successfully saved in their Notion account within the selected page/task.

Step 6: The extension provides feedback to the user, confirming the successful sharing of the screenshot.

# Productivity Apps Integration

# Microsoft Teams

The following functional requirements for ScreenGenius extension's integration with Microsoft Teams, allowing users to capture a screenshot or video and directly share it with Teams, including individual persons or groups.

**Step 1**: The user captures a screenshot or video using the ScreenGenius extension.

**Step 2**: After capturing the screenshot or video, the user decides to share it with Microsoft Teams.

**Step 3**: Authentication with Microsoft Teams API

**Condition 1:** If the user has not previously authenticated with the Microsoft Teams API:

* The extension prompts the user to authenticate with their Microsoft account.
* The user clicks on the "Authenticate with Microsoft Teams" option.
* The extension opens a new window or tab with the Microsoft authentication page.
* The user provides their Microsoft login credentials and grants necessary permissions to the extension.
* Upon successful authentication, the extension receives an authentication token from Microsoft.

**Condition 2:** Previously Authenticated User

* If the user has previously authenticated with the Microsoft Teams API, proceed to the next step.
* The extension uses the received authentication token to interact with the Microsoft Teams API securely.
* The extension provides the user with a dropdown menu or dialog to select the target destination within Microsoft Teams. Dropdown options should contain the following:
* All the Groups
* Individuals

**Step 4**: User shall be able to share the screenshot/video to an individual or Teams Channel or group

**Step 5**: User shall confirm the selection

**Step 6**: The extension uses the Microsoft Teams API to send the screenshot or video to the selected Individual or group.

**Step 7**: The user's screenshot or video is successfully shared in Microsoft Teams with the selected recipient(s).

**Step 8**: The extension provides feedback to the user, confirming the successful sharing of the screenshot or video.

API Reference link for Teams API: [Link](https://learn.microsoft.com/en-us/graph/use-the-api?context=graph%2Fapi%2F1.0&view=graph-rest-1.0#call-a-rest-api-method)

# Dropbox

The following functional requirements for ScreenGenius extension's integration with Dropbox, allowing users to capture a screenshot or video and directly share it with Dropbox.

**Step 1**: The user captures a screenshot or video using the ScreenGenius extension.

**Step 2**: After capturing the screenshot or video, the user decides to share it with Dropbox.

**Step 3**: Authentication with Dropbox.

**Condition 1:** If the user has not previously authenticated with the Dropbox

* The extension prompts the user to authenticate with their Dropbox.
* The user clicks on the "Authenticate with Dropbox" option.
* The extension opens a new window or tab with the Dropbox authentication page.
* The user provides their Dropbox login credentials and grants necessary permissions to the extension.
* Upon successful authentication, the extension receives an authentication token from Dropbox.

**Condition 2:** Previously Authenticated User

* If the user has previously authenticated with the Dropbox API, proceed to the next step.
* The extension uses the received authentication token to interact with the Dropbox API securely.
* The extension provides the user with a dropdown menu or dialog to select the target destination within Dropbox.

**Step 4**: User shall be able to share the screenshot/video to single or shared folder.

**Step 5**: User shall confirm the selection

**Step 6**: The extension uses the Dropbox API to send the screenshot or video to the following:

* A specific Dropbox folder: User shall select a specific folder from dropbox
* A shared folder or group within Dropbox: User shall select a shared folder or group within Dropbox
* An individual or email address associated with a Dropbox account
* User shall enter recipients email

**Step 7**: The user's screenshot or video is successfully shared in Dropbox with the selected recipient(s) or folder.

**Step 8**: The extension provides feedback to the user, confirming the successful sharing of the screenshot or video.

# Google Drive & Docs

The following functional requirements for ScreenGenius extension's integration with Google Drive & Docs, allowing users to capture a screenshot or video and directly share it with Google Drive & Docs or insert it into Google Drive.

**Step 1**: The user captures a screenshot or video using the ScreenGenius extension.

**Step 2**: After capturing the screenshot or video, the user decides to share it with Google Drive & Docs.

**Step 3**: Authentication with Google Drive & Docs.

**Condition 1:** If the user has not previously authenticated with the Google Drive & Docs

* The extension prompts the user to authenticate with their Google Drive & Docs.
* The user clicks on the "Authenticate with Google” option.
* The extension opens a new window or tab with the Google Drive & Docs authentication page.
* The user provides their Google login credentials and grants necessary permissions to the extension.
* Upon successful authentication, the extension receives an authentication token from Microsoft.

**Condition 2:** Previously Authenticated User

* If the user has previously authenticated with the Google Drive & Docs, proceed to the next step.
* The extension uses the received authentication token to interact with the Google Drive & Docs API securely.
* The extension provides the user with a dropdown menu or dialog to select the target destination within Google Drive & Docs.

**Step 4**: User shall share the document to:

* A specific Google drive folder- User selects Gdrive folder
* Inserting the screenshot/video into a google docs document –
* Users shall choose a Google Docs document from their account.
* The extension uses the Google Drive & Docs API to insert the screenshot or video into the selected Google Docs document.

**Step 5**: The user's screenshot or video is successfully shared in Google Drive or inserted into Google Docs.

**Step 6**: The extension provides feedback to the user, confirming the successful sharing or insertion of the screenshot or video.

**API Reference:**

| **Module** | **Services** | **Purpose** |
| --- | --- | --- |
|  | Base URL | The base URL for all API requests is https://api.cloud-elements.com/elements/api-v2/. |
| Delete | /files | [Deletes a file from your cloud service by its specified path](https://cloud-elements.github.io/docs/elements/googledrive/ko/api-documentation.html?elementId=21#!/files/deleteFiles) |
| GET | /files | * [Retrieves a single file from your cloud service from a specified path. The file is directly streamed from the cloud storage provider through Cloud Elements, and is not buffered to disk.](https://cloud-elements.github.io/docs/elements/googledrive/ko/api-documentation.html?elementId=21#!/files/getFiles) |
| POST | /files | * [Uploads a file to your endpoint cloud service. If the destination path includes directories that do not exist, they will be created](https://cloud-elements.github.io/docs/elements/googledrive/ko/api-documentation.html?elementId=21#!/files/createFile) |
| GET | /files/comments | * [Search for comments for a file](https://cloud-elements.github.io/docs/elements/googledrive/ko/api-documentation.html?elementId=21#!/files/getFilesComments) |
| POST | /files/comments | * [Search for comments for a file](https://cloud-elements.github.io/docs/elements/googledrive/ko/api-documentation.html?elementId=21#!/files/getFilesComments) |
| DELETE | [/files/comments/{commentId}](https://cloud-elements.github.io/docs/elements/googledrive/ko/api-documentation.html?elementId=21#!/files/deleteFilesCommentByCommentId) | Create a comments for a file |
| POST | [/files/copy](https://cloud-elements.github.io/docs/elements/googledrive/ko/api-documentation.html?elementId=21#!/files/createFileCopy) | * [Copies a file specified by a file path to another file path, leaving the original intact. If the source file path does not exist an error response is returned. If the destination path includes directories that do not exist, they will be created. Copying a file that already exists results in an error response.](https://cloud-elements.github.io/docs/elements/googledrive/ko/api-documentation.html?elementId=21#!/files/createFileCopy) |
| [GET](https://cloud-elements.github.io/docs/elements/googledrive/ko/api-documentation.html?elementId=21#!/files/getFilesLinks) | [/files/links](https://cloud-elements.github.io/docs/elements/googledrive/ko/api-documentation.html?elementId=21#!/files/getFilesLinks) | [Returns a link that can be used to download the specified file through Cloud Elements. The link can be used to download the file without providing credentials.](https://cloud-elements.github.io/docs/elements/googledrive/ko/api-documentation.html?elementId=21#!/files/getFilesLinks) |
| [GET](https://cloud-elements.github.io/docs/elements/googledrive/ko/api-documentation.html?elementId=21#!/files/getFilesPermissions) | [/files/permissions](https://cloud-elements.github.io/docs/elements/googledrive/ko/api-documentation.html?elementId=21#!/files/getFilesPermissions) | * [Retrieves the permissions from a single file on your cloud service from a specified path](https://cloud-elements.github.io/docs/elements/googledrive/ko/api-documentation.html?elementId=21#!/files/getFilesPermissions) |
| POST | [/files/permissions](https://cloud-elements.github.io/docs/elements/googledrive/ko/api-documentation.html?elementId=21#!/files/createFilePermission) | [Create a permission for a file](https://cloud-elements.github.io/docs/elements/googledrive/ko/api-documentation.html?elementId=21#!/files/createFilePermission) |
| DELETE | [/folders](https://cloud-elements.github.io/docs/elements/googledrive/ko/api-documentation.html?elementId=21#!/folders/deleteFolders) | [Deletes an entire folder from your cloud service by its specified path. The folder specified is deleted from your cloud service.](https://cloud-elements.github.io/docs/elements/googledrive/ko/api-documentation.html?elementId=21#!/folders/deleteFolders) |
| POST | [/folders](https://cloud-elements.github.io/docs/elements/googledrive/ko/api-documentation.html?elementId=21#!/folders/deleteFolders) | [Creates a new folder on your cloud service using its specified path. Creation of the new folder goes through Cloud Elements. If the destination path includes directories that do not exist they will be created. Creating a folder that already exists results in an error response.](https://cloud-elements.github.io/docs/elements/googledrive/ko/api-documentation.html?elementId=21#!/folders/createFolder) |
| GET | [/folders](https://cloud-elements.github.io/docs/elements/googledrive/ko/api-documentation.html?elementId=21#!/folders/deleteFolders)/contents | [Get a list of Cloud Files that are contained in a specified folder on your cloud service specified by a path.](https://cloud-elements.github.io/docs/elements/googledrive/ko/api-documentation.html?elementId=21#!/folders/getFoldersContents) |
| POST | [/folders](https://cloud-elements.github.io/docs/elements/googledrive/ko/api-documentation.html?elementId=21#!/folders/deleteFolders)/copy | [Copies a folder specified by a file path to another file path, leaving the original intact. If the source path does not exist an error response is returned. If the destination path includes directories that do not exist, they will be created.](https://cloud-elements.github.io/docs/elements/googledrive/ko/api-documentation.html?elementId=21#!/folders/createFolderCopy) |

API reference link: [link](https://docs.axway.com/bundle/Amplify_Integration_Builder_allOS_en/page/google_drive_api_documentation.html)

# SharePoint

The following functional requirements for Screen Genius extension's integration with SharePoint, allowing users to capture a screenshot or video and directly share it with SharePoint.

Step 1: The user captures a screenshot or video using the Screen Genius extension.

Step 2: After capturing the screenshot or video, the user decides to share it with SharePoint.

Step 3: Authentication with SharePoint.

**Condition 1:** If the user has not previously authenticated with the SharePoint

* The extension prompts the user to authenticate with their SharePoint.
* The user clicks on the "Authenticate with SharePoint" option.
* The extension opens a new window or tab with the SharePoint authentication page.
* The user provides their SharePoint login credentials and grants necessary permissions to the extension.
* Upon successful authentication, the extension receives an authentication token from SharePoint.

**Condition 2:** Previously Authenticated User

* If the user has previously authenticated with the SharePoint API, proceed to the next step.
* The extension uses the received authentication token to interact with the SharePoint API securely.
* The extension provides the user with a dropdown menu or dialog to select the target destination within SharePoint.

**Step 4**: User shall be able to share the screenshot/video to specific SharePoint site or document library.

**Step 5**: User shall confirm the selection

**Step 6**: The extension uses the SharePoint API to send the screenshot or video to the SharePoint site or document library.

**Step 7:** The user's screenshot or video is successfully shared in SharePoint with the selected recipient(s) or folder.

**Step 8:** The extension provides feedback to the user, confirming the successful sharing of the screenshot or video.

# Zapier

The following functional requirements for ScreenGenius extension's using Zapier API to send a captured screenshot from the ScreenGenius extension as input and receive a generated response

**Step 1**: The user captures a screenshot or video using the ScreenGenius extension.

**Step 2**: After capturing the screenshot or video, the user decides to send it to the Zapier.

**Step 3:** The extension provides the user with a dialog or form to configure the request to the Zapier API.

**Step 4**: Authentication with Zapier API

**Condition 1:** If the user has not previously authenticated with the Zapier:

* The extension prompts the user to provide their Zapier API key or authenticate with their details.
* The extension opens a new window or tab with the authentication page.
* The user provides their Zapier login credentials/Gmail credentials and grants necessary permissions to the extension.

**Condition 2:** Previously Authenticated User

* If the user has previously authenticated with the Zapier, user shall be able to share the screenshot/video as post or send it to individual persons.

**Step 5:** Once user logged into the application. User shall grant necessary permissions.

**Step 6**: A small dialogue box should be opened and User shall be asked to choose destination from the dropdown. The dropdown options may contain:

* Folder Names – User shall select one of the folder from dropdown
* Create a Folder – Click on create Folder🡪Enter Folder Name🡪Select Folder

**Step 7**: Once user selects the destination and submit, user shall be asked to type a message and click on send message.

**Step 8**: The extension uses the Zapier API to send the screenshot or video to the selected recipient destination.

**Step 9**: The user's screenshot or video is successfully shared in Zapier with the selected recipient(s) or destination.

**Step 10**: The extension provides feedback to the user, confirming the successful sharing of the screenshot or video.

# Slack

The following functional requirements for ScreenGenius extension's using Slack API to send a captured screenshot from the ScreenGenius extension as input and receive a generated response

**Step 1**: The user captures a screenshot or video using the ScreenGenius extension.

**Step 2**: After capturing the screenshot or video, the user decides to send it to the Slack.

**Step 3:** The extension provides the user with a dialog or form to configure the request to the Slack API.

**Step 4**: Authentication with Twitter API

**Condition 1:** If the user has not previously authenticated with the Slack:

* The extension prompts the user to provide their Slack API key or authenticate with their details.
* The extension opens a new window or tab with the authentication page.
* The user provides their Slack login credentials/Gmail credentials and grants necessary permissions to the extension.

**Condition 2:** Previously Authenticated User

* If the user has previously authenticated with the Slack, user shall be able to share the screenshot/video as post or send it to individual persons.

**Step 5:** Once user logged into the application. User shall grant necessary permissions.

**Step 6:** A small dialogue box should be opened and User shall be asked to choose destination from the dropdown. The dropdown options may contain:

* Individual names
* Group Names

**Step 7:** Once user selects the destination and submit, user shall be asked to type a message and click on send message.

**Step 6:** The extension uses the Slack API to send the screenshot or video to the selected recipient destination.

**Step 7:** The user's screenshot or video is successfully shared in Slack with the selected recipient(s) or destination.

**Step 8:** The extension provides feedback to the user, confirming the successful sharing of the screenshot or video.

**API Reference:** [link](https://api.slack.com/apis/calls)

|  |  |  |
| --- | --- | --- |
| **API Name** | **Uses** | **Argument** |
| [**auth.revoke**](https://api.slack.com/methods/auth.revoke) | Revokes a token. |  |
| [**auth.test**](https://api.slack.com/methods/auth.test) | Checks authentication & identity. | Authentication token bearing required scopes. Tokens should be passed as an HTTP Authorization header or alternatively, as a POST parameter. |
| [**auth.teams.list**](https://api.slack.com/methods/auth.teams.list) | Obtain a full list of workspaces your org-wide app has been approved for. |  |
| [**chat.meMessage**](https://api.slack.com/methods/chat.meMessage) | Share a message into a channel. |  |
| [**chat.postMessage**](https://api.slack.com/methods/chat.postMessage) | Sends a message to a channel. |  |
| [**chat.delete**](https://api.slack.com/methods/chat.delete) | Deletes a message. |  |
| [**conversations.acceptSharedInvite**](https://api.slack.com/methods/conversations.acceptSharedInvite). | Accepts an invitation to a Slack Connect channel |  |
| [**conversations.approveSharedInvite**](https://api.slack.com/methods/conversations.approveSharedInvite) | Approves an invitation to a Slack Connect channel |  |
| [**users.list**](https://api.slack.com/methods/users.list) | Lists all users in a Slack team. |  |
| [**usergroups.list**](https://api.slack.com/methods/usergroups.list) | List all User Groups for a team |  |
| [**conversations.create**](https://api.slack.com/methods/conversations.create) | Initiates a public or private channel-based conversation |  |
| [**conversations.history**](https://api.slack.com/methods/conversations.history) | Fetches a conversation's history of messages and events. |  |
| [**conversations.join**](https://api.slack.com/methods/conversations.join) | Joins an existing conversation. |  |
| [**conversations.list**](https://api.slack.com/methods/conversations.list) | Leaves a conversation. |  |
| [**reminders.add**](https://api.slack.com/methods/reminders.add) | Creates a reminder. |  |
| [**reminders.complete**](https://api.slack.com/methods/reminders.complete) | Marks a reminder as complete. |  |

# Social Media Integration

# Twitter

The following functional requirements for ScreenGenius extension's using Twitter API to send a captured screenshot from the ScreenGenius extension as input and receive a generated response

**Step 1**: The user captures a screenshot or video using the ScreenGenius extension.

**Step 2**: After capturing the screenshot or video, the user decides to send it to the Twitter.

**Step 3:** The extension provides the user with a dialog or form to configure the request to the Twitter API.

**Step 4**: Authentication with Twitter API

**Condition 1:** If the user has not previously authenticated with the Twitter:

* The extension prompts the user to provide their Twitter API key or authenticate with their details.
* The extension opens a new window or tab with the authentication page.
* The user provides their twitter login credentials/Gmail credential and grants necessary permissions to the extension.

**Condition 2:** Previously Authenticated User

* If the user has previously authenticated with the Twitter, user shall be able to share the screenshot/video as post or send it to individual persons.

**Step 5:** Once user logged into the application, a dialogue box opens for the user to enter a message and share the screenshot/video along with the preview of screenshot/video.

**Step 6:** The extension uses the Twitter API to send the screenshot or video to the selected destination.

**Step 7:** The user's screenshot or video is successfully shared in Twitter with the selected recipient(s) or destination.

**Step 8:** The extension provides feedback to the user, confirming the successful sharing of the screenshot or video.

**API Reference**: [link](https://developer.twitter.com/en/docs/twitter-api/data-dictionary/introduction)

|  |  |  |  |
| --- | --- | --- | --- |
| Tweets | [GET statuses/show](https://developer.twitter.com/en/products/twitter-api/tweets/post-and-engage/api-reference/get-statuses-show-id) [GET statuses/lookup](https://developer.twitter.com/en/products/twitter-api/tweets/post-and-engage/api-reference/get-statuses-lookup) |  | [Tweets lookup](https://developer.twitter.com/en/docs/twitter-api/tweets/lookup/introduction) |
| [POST statuses/update](https://developer.twitter.com/en/docs/twitter-api/v1/tweets/post-and-engage/api-reference/post-statuses-update) [POST statuses/destroy/:id](https://developer.twitter.com/en/docs/twitter-api/v1/tweets/post-and-engage/api-reference/post-statuses-destroy-id) |  | [Manage Tweets](https://developer.twitter.com/en/docs/twitter-api/tweets/manage-tweets/introduction) |
| [GET statuses/user\_timeline](https://developer.twitter.com/en/docs/twitter-api/v1/tweets/timelines/api-reference/get-statuses-user_timeline.html) [GET statuses/mentions\_timeline](https://developer.twitter.com/en/docs/twitter-api/v1/tweets/timelines/api-reference/get-statuses-mentions_timeline.html) [GET statuses/home\_timeline](https://developer.twitter.com/en/docs/twitter-api/v1/tweets/timelines/api-reference/get-statuses-user_timeline) |  | [Timelines](https://developer.twitter.com/en/docs/twitter-api/tweets/timelines/introduction) - User Tweet timeline - User mention timeline - Reverse chronological home timeline |
| GET search/tweets | [Search API](https://developer.twitter.com/en/docs/twitter-api/enterprise/search-api/overview) | [Search Tweets](https://developer.twitter.com/en/docs/twitter-api/tweets/search/introduction) - Full-archive search |
| [POST favorites/create](https://developer.twitter.com/en/docs/twitter-api/v1/tweets/post-and-engage/api-reference/post-favorites-create) [POST favorites/destroy](https://developer.twitter.com/en/docs/twitter-api/v1/tweets/post-and-engage/api-reference/post-favorites-destroy) |  | [Manage Likes](https://developer.twitter.com/en/docs/twitter-api/tweets/likes/introduction) - Like a Tweet - Unlike a Tweet |
| Users | [GET users/show](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/follow-search-get-users/api-reference/get-users-show) [GET users/lookup](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/follow-search-get-users/api-reference/get-users-lookup) |  | [Users lookup](https://developer.twitter.com/en/docs/twitter-api/users/lookup/introduction) |
| [GET users/search](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/follow-search-get-users/api-reference/get-users-search) |  | [Replacement under consideration] |
| [GET followers/ids](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/follow-search-get-users/api-reference/get-followers-ids) [GET followers/list](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/follow-search-get-users/api-reference/get-followers-list) [GET friends/ids](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/follow-search-get-users/api-reference/get-friends-ids) [GET friends/list](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/follow-search-get-users/api-reference/get-friends-list) |  | [Follows lookup](https://developer.twitter.com/en/docs/twitter-api/users/follows) |
| [GET mutes/users/ids](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/mute-block-report-users/api-reference/get-mutes-users-ids) [GET mutes/users/list](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/mute-block-report-users/api-reference/get-mutes-users-list) |  | [Mutes lookup](https://developer.twitter.com/en/docs/twitter-api/users/mutes) |
| [GET account/settings](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/manage-account-settings/api-reference/get-account-settings) [GET account/verify\_credentials](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/manage-account-settings/api-reference/get-account-verify_credentials) [GET users/profile\_banner](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/manage-account-settings/api-reference/get-users-profile_banner) [POST account/settings](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/manage-account-settings/api-reference/post-account-settings) [POST account/update\_profile](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/manage-account-settings/api-reference/post-account-update_profile) [POST account/update\_profile\_banner](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/manage-account-settings/api-reference/post-account-update_profile_banner) [POST account/remove\_profile\_banner](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/manage-account-settings/api-reference/post-account-remove_profile_banner) [POST account/update\_profile\_image](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/manage-account-settings/api-reference/post-account-update_profile_image) |  |  |
| [GET saved\_searches/show/:id](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/manage-account-settings/api-reference/get-saved_searches-show-id) [GET saved\_searches/list](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/manage-account-settings/api-reference/get-saved_searches-list) [POST saved\_searches/create](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/manage-account-settings/api-reference/post-saved_searches-create) [POST saved\_searches/destroy/:id](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/manage-account-settings/api-reference/post-saved_searches-destroy-id) |  |  |
| [POST users/report\_spam](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/mute-block-report-users/api-reference/post-users-report_spam) |  |  |
|  | [Account Activity API](https://developer.twitter.com/en/docs/twitter-api/enterprise/account-activity-api/overview) |  |
| Direct Messages |  |  | [Direct Messages lookup](https://developer.twitter.com/en/docs/twitter-api/direct-messages/lookup/introduction) [Manage Direct Messages](https://developer.twitter.com/en/docs/twitter-api/direct-messages/manage/introduction) |
| Lists | [GET lists/show](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/create-manage-lists/api-reference/get-lists-show) |  | [Lists lookup](https://developer.twitter.com/en/docs/twitter-api/lists/list-lookup/introduction) |
| [POST lists/create](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/create-manage-lists/api-reference/post-lists-create) [POST lists/destroy](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/create-manage-lists/api-reference/post-lists-destroy) [POST lists/update](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/create-manage-lists/api-reference/post-lists-update) |  | [Manage Lists](https://developer.twitter.com/en/docs/twitter-api/lists/manage-lists/introduction) |
| [GET lists/statuses](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/create-manage-lists/api-reference/get-lists-statuses) |  | [Lists Tweets lookup](https://developer.twitter.com/en/docs/twitter-api/lists/list-tweets/introduction) |
| [GET lists/members](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/create-manage-lists/api-reference/get-lists-members) [GET lists/memberships](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/create-manage-lists/api-reference/get-lists-memberships) [POST lists/members/create](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/create-manage-lists/api-reference/post-lists-members-create) [POST lists/members/destroy](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/create-manage-lists/api-reference/post-lists-members-destroy) |  | [List members](https://developer.twitter.com/en/docs/twitter-api/lists/list-members/introduction) |
| [GET lists/subscribers](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/create-manage-lists/api-reference/get-lists-subscribers) [GET lists/subscriptions](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/create-manage-lists/api-reference/get-lists-subscriptions) [GET lists/lists](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/create-manage-lists/api-reference/get-lists-list) [POST lists/subscribers/create](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/create-manage-lists/api-reference/post-lists-subscribers-create) [POST lists/subscribers/destroy](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/create-manage-lists/api-reference/post-lists-subscribers-destroy) |  | [Lists follows](https://developer.twitter.com/en/docs/twitter-api/lists/list-follows/introduction) |
| [GET lists/ownerships](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/create-manage-lists/api-reference/get-lists-ownerships) |  |  |
|  |  |  |
| [GET lists/members/show](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/create-manage-lists/api-reference/get-lists-members-show) [GET lists/subscribers/show](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/create-manage-lists/api-reference/get-lists-subscribers-show) |  |  |
| [POST lists/members/create\_all](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/create-manage-lists/api-reference/post-lists-members-create_all) [POST lists/members/destroy\_all](https://developer.twitter.com/en/docs/twitter-api/v1/accounts-and-users/create-manage-lists/api-reference/post-lists-members-destroy_all) |  |  |
| Collections | [GET collections/entries](https://developer.twitter.com/en/docs/twitter-api/v1/tweets/curate-a-collection/api-reference/get-collections-entries) [GET collections/list](https://developer.twitter.com/en/docs/twitter-api/v1/tweets/curate-a-collection/api-reference/get-collections-list) [GET collections/show](https://developer.twitter.com/en/docs/twitter-api/v1/tweets/curate-a-collection/api-reference/get-collections-show) [POST collections/create](https://developer.twitter.com/en/docs/twitter-api/v1/tweets/curate-a-collection/api-reference/post-collections-create) [POST collections/destroy](https://developer.twitter.com/en/docs/twitter-api/v1/tweets/curate-a-collection/api-reference/post-collections-destroy) [POST collections/entries/add](https://developer.twitter.com/en/docs/twitter-api/v1/tweets/curate-a-collection/api-reference/post-collections-entries-add) [POST collections/entries/curate](https://developer.twitter.com/en/docs/twitter-api/v1/tweets/curate-a-collection/api-reference/post-collections-entries-curate) [POST collections/entries/move](https://developer.twitter.com/en/docs/twitter-api/v1/tweets/curate-a-collection/api-reference/post-collections-entries-move) [POST collections/entries/remove](https://developer.twitter.com/en/docs/twitter-api/v1/tweets/curate-a-collection/api-reference/post-collections-entries-remove) [POST collections/update](https://developer.twitter.com/en/docs/twitter-api/v1/tweets/curate-a-collection/api-reference/post-collections-update) |  |  |

# Facebook

The following functional requirements for ScreenGenius extension's using Facebook API to send a captured screenshot from the ScreenGenius extension as input and receive a generated response

**Step 1**: The user captures a screenshot or video using the ScreenGenius extension.

**Step 2**: After capturing the screenshot or video, the user decides to send it to the Facebook.

**Step 3:** The extension provides the user with a dialog or form to configure the request to the Facebook API.

**Step 4**: Authentication with Facebook API

**Condition 1:** If the user has not previously authenticated with the Facebook:

* The extension prompts the user to provide their Facebook API key or authenticate with their details.
* The extension opens a new window or tab with the authentication page.
* The user provides their Facebook login credentials credential and grants necessary permissions to the extension.

**Condition 2:** Previously Authenticated User

* If the user has previously authenticated with the Facebook, user shall be able to share the screenshot/video as post or send it to individual persons.

**Step 5:** Once user logged into the application, a dialogue box opens for the user to enter a message and share the screenshot/video along with the preview of screenshot/video.

**Step 6:** The extension uses the Facebook API to send the screenshot or video to the selected destination.

**Step 7:** The user's screenshot or video is successfully shared in Facebook with the selected recipient(s) or destination.

**Step 8:** The extension provides feedback to the user, confirming the successful sharing of the screenshot or video.

# LinkedIn

The following functional requirements for ScreenGenius extension's using LinkedIn API to send a captured screenshot from the ScreenGenius extension as input and receive a generated response

**Step 1**: The user captures a screenshot or video using the ScreenGenius extension.

**Step 2**: After capturing the screenshot or video, the user decides to send it to the LinkedIn.

**Step 3:** The extension provides the user with a dialog or form to configure the request to the LinkedIn API.

**Step 4**: Authentication with LinkedIn API

**Condition 1:** If the user has not previously authenticated with the LinkedIn:

* The extension prompts the user to provide their LinkedIn API key or authenticate with their details.
* The extension opens a new window or tab with the authentication page.
* The user provides their LinkedIn login credentials credential and grants necessary permissions to the extension.

**Condition 2:** Previously Authenticated User

* If the user has previously authenticated with the LinkedIn, user shall be able to share the screenshot/video as post or send it to individual persons by selecting the account.

**Step 5:** Once user logged into the application, a dialogue box opens for the user to select a individual user/ multi users/share as a post in the dropdown and submit.

**Step 6**: Once the user selects the destination user shall able to enter a message and submit.

**Step 7**: The extension uses the LinkedIn API to send the screenshot or video to the selected destination.

**Step 8**: The user's screenshot or video is successfully shared in LinkedIn with the selected recipient(s) or destination.

**Step 9**: The extension provides feedback to the user, confirming the successful sharing of the screenshot or video.

# Instagram

The following functional requirements for ScreenGenius extension's using Instagram API to send a captured screenshot from the ScreenGenius extension as input and receive a generated response

**Step 1**: The user captures a screenshot or video using the ScreenGenius extension.

**Step 2**: After capturing the screenshot or video, the user decides to send it to the Instagram.

**Step 3:** The extension provides the user with a dialog or form to configure the request to the Instagram API.

**Step 4**: Authentication with Instagram API

**Condition 1:** If the user has not previously authenticated with the Instagram:

* The extension prompts the user to provide their Instagram API key or authenticate with their details.
* The extension opens a new window or tab with the authentication page.
* The user provides their Instagram login credentials credential and grants necessary permissions to the extension.

**Condition 2:** Previously Authenticated User

* If the user has previously authenticated with the Instagram, user shall be able to share the screenshot/video as post or send it to individual persons by selecting the account.

**Step 5:** Once user logged into the application, a dialogue box opens for the user to select a individual user/ multi users/share as a post in the dropdown and submit.

**Step 6:** Once the user selects the destination user shall able to enter a message and submit.

**Step 7:** The extension uses the Instagram API to send the screenshot or video to the selected destination.

**Step 8**: The user's screenshot or video is successfully shared in Instagram with the selected recipient(s) or destination.

**Step 9**: The extension provides feedback to the user, confirming the successful sharing of the screenshot or video.

# YouTube

The following functional requirements for ScreenGenius extension's using YouTube API to send a captured screenshot from the ScreenGenius extension as input and receive a generated response

**Step 1**: The user captures a screenshot or video using the ScreenGenius extension.

**Step 2**: After capturing the screenshot or video, the user decides to send it to the YouTube.

**Step 3:** The extension provides the user with a dialog or form to configure the request to the YouTube API.

**Step 4**: Authentication with YouTube API

**Condition 1:** If the user has not previously authenticated with the YouTube:

* The extension prompts the user to provide their YouTube API key or authenticate with their details.
* The extension opens a new window or tab with the authentication page.
* The user provides their YouTube login credentials credential and grants necessary permissions to the extension.

**Condition 2:** Previously Authenticated User

* If the user has previously authenticated with the YouTube, user shall be able to share the screenshot/video as post or send it to individual persons by selecting the account.

**Step 5**:Once user logged into the application, a dialogue box opens for the user to select a individual select a private channel/share as a post in the dropdown and submit.

* If the user selects a private channel, all the channels created by the user should be shown in the dropdown and user shall select one of the channel and submit. The screenshot/video should be saved in the selected destination.

**Step 6**: Once the user selects the destination user shall able to enter a message and submit.

**Step 7**: The extension uses the YouTube API to send the screenshot or video to the selected destination.

**Step 8**: The user's screenshot or video is successfully shared in LinkedIn with the selected recipient(s) or destination.

**Step 9**: The extension provides feedback to the user, confirming the successful sharing of the screenshot or video.

# ChatGPT & BARD Integration

# ChatGPT

The following functional requirements for ScreenGenius extension's using ChatGPT API to send a captured screenshot from the ScreenGenius extension as input and receive a generated response

**Step 1**: The user captures a screenshot or video using the ScreenGenius extension.

**Step 2**: After capturing the screenshot or video, the user decides to send it to the ChatGPT API for generating a response.

**Step 3:** The extension provides the user with a dialog or form to configure the request to the ChatGPT API.

**Step 4**: Authentication with ChatGPT API

**Condition 1:** If the user has not previously authenticated with the ChatGPT:

* The extension prompts the user to provide their ChatGPT API key or authenticate with their details.
* The extension opens a new window or tab with the authentication page.
* The user provides their login credentials and grants necessary permissions to the extension.

**Condition 2:** Previously Authenticated User

* If the user has previously authenticated with the ChatGPT, proceed to the next step.
* The user configures the ChatGPT API request, which includes the following parameters:
* **Input**: The user attaches the captured screenshot to the request as input.
* **Prompt**: The user can provide a text prompt or context to guide the ChatGPT response generation.
* The extension uses the configured parameters to send an API request to the ChatGPT API endpoint.

**Step 5:** The ChatGPT API processes the request, using the provided screenshot as input and generating a response based on the input and prompt.

**Step 6**: The ChatGPT API returns the generated response to the extension.

|  |  |
| --- | --- |
| **API Key** | **Function** |
| Authorization: Bearer OPENAI\_API\_KEY | Authorizing the user |
| POST https://api.openai.com/v1/audio/transcriptions | Transcribes audio into the input language. |
| Chat Completion Object - Represents a chat completion response returned by model, based on the provided input. | {  "id": "chatcmpl-123",  "object": "chat.completion",  "created": 1677652288,  "model": "gpt-3.5-turbo-0613",  "choices": [{  "index": 0,  "message": {  "role": "assistant",  "content": "\n\nHello there, how may I assist you today?",  },  "finish\_reason": "stop"  }],  "usage": {  "prompt\_tokens": 9,  "completion\_tokens": 12,  "total\_tokens": 21  }  } |
| Create Chat completion – Creates a model response for the given chat conversation.  Request: curl https://api.openai.com/v1/chat/completions \  -H "Content-Type: application/json" \  -H "Authorization: Bearer $OPENAI\_API\_KEY" \  -d '{  "model": "gpt-3.5-turbo",  "messages": [  {  "role": "system",  "content": "You are a helpful assistant."  },  {  "role": "user",  "content": "Hello!"  }  ]  }' | Response:  {  "id": "chatcmpl-123",  "object": "chat.completion",  "created": 1677652288,  "model": "gpt-3.5-turbo-0613",  "choices": [{  "index": 0,  "message": {  "role": "assistant",  "content": "\n\nHello there, how may I assist you today?",  },  "finish\_reason": "stop"  }],  "usage": {  "prompt\_tokens": 9,  "completion\_tokens": 12,  "total\_tokens": 21  }  } |
| The Image Object - Represents the URL or the content of an image generated by the OpenAI API.  Step 1: Request Body  Step 2: Returns  Step 3: Create Image Edit  **POST**[**https://api.openai.com/v1/images/edits**](https://api.openai.com/v1/images/edits)  Step 4: Create Image Variation  **POST https://api.openai.com/v1/images/variations** |  |

# BARD

The following functional requirements for ScreenGenius extension's using BARD API to send a captured screenshot from the ScreenGenius extension as input and receive a generated response

**Step 1**: The user captures a screenshot or video using the ScreenGenius extension.

**Step 2**: After capturing the screenshot or video, the user decides to send it to the BARD API for generating a response.

**Step 3:** The extension provides the user with a dialog or form to configure the request to the BARD API.

**Step 4**: Authentication with BARD API

**Condition 1:** If the user has not previously authenticated with the BARD:

* The extension prompts the user to provide their BARD API key or authenticate with their details.
* The extension opens a new window or tab with the authentication page.
* The user provides their login credentials/Gmail credentials and grants necessary permissions to the extension.

**Condition 2:** Previously Authenticated User

* If the user has previously authenticated with the BARD, proceed to the next step.
* The user configures the BARD API request, which includes the following parameters:
* **Input:** The user attaches the captured screenshot to the request as input.
* **Prompt:** The user can provide a text prompt or context to guide the BARD response generation.
* The extension uses the configured parameters to send an API request to the BARD API endpoint.

**Step 5:** The BARD API processes the request, using the provided screenshot as input and generating a response based on the input and prompt.

**Step 6**: The BARD API returns the generated response to the extension.

# Messaging Integration

# WhatsApp

The following functional requirements for ScreenGenius extension's integration with WhatsApp, allowing users to capture a screenshot or video and directly share it with WhatsApp, including individual persons or groups.

**Step 1**: The user captures a screenshot or video using the ScreenGenius extension.

**Step 2**: After capturing the screenshot or video, the user decides to share it with WhatsApp.

**Step 3**: Authentication with WhatsApp API

**Condition 1:** If the user has not previously authenticated with the WhatsApp:

* The extension prompts the user to authenticate with their WhatsApp
* The user clicks on the "Authenticate with WhatsApp” option.
* The extension opens a new window or tab with the WhatsApp authentication page.
* The user scans their QR Code with mobile and grants necessary permissions to the extension.
* Upon successful authentication, the extension receives an authentication token from WhatsApp.

**Condition 2:** Previously Authenticated User/Logged in to the web browser

* If the user has previously authenticated with the WhatsApp, proceed to the next step.
* The extension uses the received authentication token to interact with the WhatsApp API securely.

**Step 4**: The extension provides the user with a dropdown menu or dialog to select the target destination within WhatsApp. Dropdown options should contain the following:

* All the Groups
* Individuals

**Step 5:** Once user selects the group/individual and submit, space should be provided for the user to enter a message and submit.

**Step 6**: The extension uses the WhatsApp API to send the screenshot or video to the selected Individual or group.

**Step 7**: The user's screenshot or video is successfully shared in WhatsApp with the selected recipient(s).

**Step 8**: The extension provides feedback to the user, confirming the successful sharing of the screenshot or video.

**API Reference**: [Link](https://developers.facebook.com/docs/whatsapp/cloud-api/reference)

|  |  |
| --- | --- |
| **Node** | **Description** |
| [/PHONE\_NUMBER\_ID](https://developers.facebook.com/docs/whatsapp/cloud-api/reference/two-step-verification) | Represents a specific phone number. This endpoint is used to [set up two-factor authentication](https://developers.facebook.com/docs/whatsapp/cloud-api/reference/two-step-verification). |
| [/MEDIA\_ID](https://developers.facebook.com/docs/whatsapp/cloud-api/reference/media#retrieve-media-url) | Represents a specific media object. Use this endpoint to get and delete media objects. |
| [/MEDIA\_URL](https://developers.facebook.com/docs/whatsapp/cloud-api/reference/media#download-media) | Represents a specific media URL. Use this endpoint to download media from a URL. |
| Message Object | To send a message, you must first assemble a message object with the content you want to send. These are the parameters used in a message object: |
| audio | Required when type=audio**.** |
| context | Required if [replying to](https://developers.facebook.com/docs/whatsapp/cloud-api/guides/send-messages#replies) any message in the conversation. |
| Image | Required when type=image**.** |
| Messaging\_product | Messaging service used for the request. Use "whatsapp". |
| Preview\_url | Required if type=text**.**  Allows for URL previews in text messages — See the [Sending URLs in Text Messages](https://developers.facebook.com/docs/whatsapp/api/messages/text#urls). This field is optional if not including a URL in your message. Values: false (default), true. |
| recipient\_type | Currently, you can only send messages to individuals. Set this as individual. |
| text | A [text object](https://developers.facebook.com/docs/whatsapp/cloud-api/reference/messages#text-object). |
| id | **Required when**type**is**audio**,**document**,**image**,**sticker**, or**video**and you are not using a link.**  The media object ID. Do not use this field when message type is set to text. |
| link | **Required when**type**is**audio**,**document**,**image**,**sticker**, or**video**and you are not using an uploaded media ID (i.e. you are hosting the media asset on your public server).**  The protocol and URL of the media to be sent. Use **only**with HTTP/HTTPS URLs |
| caption | Media asset caption. Do not use with audio or sticker media. |
| filename | Describes the filename for the specific document. Use only with document media  The extension of the filename will specify what format the document is displayed as in WhatsApp |
| provider | This path is optionally used with a link when the HTTP/HTTPS link is not directly accessible and requires additional configurations like a bearer token. |

# Telegram

The following functional requirements for ScreenGenius extension's integration with Telegram, allowing users to capture a screenshot or video and directly share it with Telegram, including individual persons or groups/channels.

**Step 1**: The user captures a screenshot or video using the ScreenGenius extension.

**Step 2**: After capturing the screenshot or video, the user decides to share it with Telegram.

**Step 3**: Authentication with Telegram API

**Condition 1:** If the user has not previously authenticated with the Telegram:

* The extension prompts the user to authenticate with their Telegram.
* The user clicks on the "Authenticate with Telegram” option.
* The extension opens a new window or tab with the Telegram authentication page.
* The users can enter their Mobile No/email and password and login.
* Once the user logged in, the user shall grant necessary permissions to the extension.
* Upon successful authentication, the extension receives an authentication token from Telegram.

**Condition 2:** Previously Authenticated User/Logged in to the web browser

* If the user has previously authenticated with the Telegram, proceed to the next step.
* The extension uses the received authentication token to interact with the Telegram securely.

**Step 4**: The extension provides the user with a dropdown menu or dialog box to select the target destination within Telegram. Dropdown options should contain the following:

* All the Groups/Channels
* Individuals

**Step 5:** Once user selects the group/individual and submit, space should be provided for the user to enter a message and submit.

**Step 6**: The extension uses the Telegram to send the screenshot or video to the selected Individual or group.

**Step 7**: The user's screenshot or video is successfully shared in Telegram with the selected recipient(s).

**Step 8**: The extension provides feedback to the user, confirming the successful sharing of the screenshot or video.

# Mail Clients Integration

# Gmail

The following functional requirements for ScreenGenius extension's integration with Gmail, allowing users to capture a screenshot or video and directly share it with other people on Gmail.

**Step 1**: The user captures a screenshot or video using the ScreenGenius extension.

**Step 2**: After capturing the screenshot or video, the user decides to share it on Gmail.

**Step 3**: Authentication with Gmail API

**Condition 1:** If the user has not previously authenticated with the Gmail API:

* The extension prompts the user to authenticate with their Gmail account.
* The user clicks on the "Authenticate with Gmail" option.
* The extension opens a new window or tab with the Gmail.
* The user provides their Gmail login credentials.
* Upon successful authentication, the extension receives an authentication token from Gmail.

**Condition 2:** Previously Authenticated User

* If the user has previously authenticated with the Gmail API, proceed to the next step.
* The extension uses the received authentication token to interact with the Gmail API securely.
* The extension provides the user with a dialogue box in a new tab, gmail page is shown along with the screenshot/video.

The Gmail page should contain the following:

* To – This shall be blank and allow user enter single or multiple emails
* Sub – It is pre-defined with title of the screenshot/video
* Upon clicking on video user shall be able to view the link, edit the display name for URL, and remove the video.

**Step 4**: User shall be able to share the screenshot/video to single or multiple people.

**Step 6**: The extension uses the Gmail API to send the screenshot or video to the selected Individual or group.

**Step 7**: The user's screenshot or video is successfully shared in Gmail with the selected recipient(s).

**Step 8**: The extension provides feedback to the user, confirming the successful sharing of the screenshot or video.

# Outlook

The following functional requirements for ScreenGenius extension's integration with Outlook, allowing users to capture a screenshot or video and directly share it with other people on Outlook.

**Step 1**: The user captures a screenshot or video using the ScreenGenius extension.

**Step 2**: After capturing the screenshot or video, the user decides to share it on Outlook.

**Step 3**: Authentication with Outlook API

**Condition 1:** If the user has not previously authenticated with the Outlook API:

* The extension prompts the user to authenticate with their Outlook account.
* The user clicks on the "Authenticate with Outlook" option.
* The extension opens a new window or tab with the Outlook.
* The user provides their Outlook/Microsoft login credentials.
* Upon successful authentication, the extension receives an authentication token from Outlook.

**Condition 2:** Previously Authenticated User

* If the user has previously authenticated with the Outlook API, proceed to the next step.
* The extension uses the received authentication token to interact with the Outlook API securely.
* The extension provides the user with a dialogue box in a new tab, Outlook page is shown along with the screenshot/video.

The Gmail page should contain the following:

* To – This shall be blank and allow user enter single or multiple emails
* Sub – It is pre-defined with title of the screenshot/video
* Upon clicking on video user shall be able to view the link, edit the display name for URL, and remove the video.

**Step 4**: User shall be able to share the screenshot/video to single or multiple people.

**Step 6**: The extension uses the Outlook API to send the screenshot or video to the selected Individual or group.

**Step 7**: The user's screenshot or video is successfully shared in Outlook with the selected recipient(s).

**Step 8**: The extension provides feedback to the user, confirming the successful sharing of the screenshot or video.

# Screenshot URL Summary Generation

* Use the ChatGPT/BARD API to send the captured screenshot as input and receive a generated URL summary based on the screenshot.

# AdSense and Ads Placement

* Inject the code into extension pages and dashboard pages into appropriate locations.

# Super Admin – Management Tool

# User Management

* The system shall implement Role-Based Access Control to categorize users into roles, including "Super Admin," "Admin," and "User."
* Super Admin Privileges: The Super Admin shall have full access to all system features and data.
* The Super Admin shall be able to manage the access rights of other admins, including adding and removing and updating admin users.
* Super Admin should be able to add/remove admins account.

# Image & Video Management

* Super admin can delete a screenshot or video if it contains inappropriate content or for other reasons.
* Admin/Super Admin users shall be able to define retention policies for stored images and videos, specifying how long data is retained before automatic deletion

# Subscription Management

* Super admin can manage the subscription of various users, including cancellations and extensions.

# Pricing Plans Management

* Super admin can create, edit, or delete pricing plans.

# Extension’s Static UI & Text

* Super admin can manage the primary color of the extension, its static texts, font sizes, and other static data.

# General Settings

* Super admin can configure basic settings like video or screenshot resolutions, default video recording delay, etc.

# Pro Features Settings

* Super admin can configure pro settings like AdSense disable, upload size & number of files limit, etc.

# Screen Genius – User Management

## Pricing Plans

* Plans will be saved into the database for future management.

## Membership

* Use PayPal for payments and store user memberships.

## Pro & Free User Roles

* Assign different roles to users based on their subscription, enabling different levels of access and ad-free experiences.

## Ads Free

Display ads for free users in designated areas and offer ad-free experiences for pro users.

Display ads for the users in below places:

* Banner Ads
* **Dashboard – D**isplay Banner at right side of the dashboard by showing two videos in row.
* **Dashboard –** Display Ad in place of a video block for every three rows
* **Dashboard –** Display a banner Ad after every threerows
* **Sign Up –** Display an Ad in sign up page
* **View video in new tab –** Display an ad by reducing the size of the video
* **View Favorite/starred Videos –** Display ad in starred videos page
* **Edit Video –** Display an ad at left side of the page by reducing the size of the video
* Video Ads (Watch Ad to perform Action) Standard or Skippable videos
* Pro features usage after Ad/banner for one time usage or per session: Upload image + Share Public Link
* Add your watermark after watching a video or Watch a video to remove default watermark
* To Edit Video
* To share video

## Watermark

* Apply a watermark to screenshots and videos for free accounts, with an option for pro users to disable or customize it.

## Pro Features

|  |  |  |
| --- | --- | --- |
| **Feature** | **Pro** | **Free** |
| Price | Annually $4 (~~$5~~) Monthly $5 (~~$6~~) | Free |
| Unlimited Screenshots/Videos | Unlimited video/screenshots | Limited to 40 screenshots and 20 videos |
| Watermark | Can be added | Cannot added |
| Unlimited video length | Unlimited | Up to 5 min |
| Gradient | Available | Unavailable |
| Resolution | 4K | 720p |
| Data Recovery | Can be recovered till 30 days from trash | Cannot be recovered |
| User Management | Available | Not Available |
| Integrations | Available | Not Available |
| Download | GIF, MP4 | WebM |

Analysis for price allocation:

|  |  |  |
| --- | --- | --- |
| 7.2 | MB Data per Min' |  |
| 1440 | Mins per day |  |
| 10368 | MB |  |
| **10.125** | GB/Per day |  |
| **303.75** | **GB/Month** | **9.63 USD** |

## Unsubscribe

* Allow users to unsubscribe from memberships, with no refunds.

## Account Management

* Enable users to change their profile picture, name, password, etc.

# Sharing Features

## Remember Email

* Remember the email every time a user shares via email using default email share or any platform integrations.

## Email Suggestions

* Display a list of recent or frequent emails for user selection.
* The list can show up to 20 emails.

# Screenshot/Video additional features

## Screenshot Background (Gradient)

* Add a background color to screenshots, which can be a plain color, gradient, or an image for pro users.
* Users shall be able to add gradient backgrounds to screenshots and videos – The gradient colors must be customized as per the user needs.

## Screenshot Padding

* Provide a white area around screenshots for better manageability and padding.
* User shall be able to manage the thickness of padding in terms of %.

## Emoji & Content Creator Elements

* Allow users to add emojis and content creator elements to their screenshots and videos.
* Allow users to customize the size and position of emojis in screenshots/videos.

# Q & A Feedback

* Address any feedback provided by QA and implement minor updates as requested by the client.

# Non-Functional Requirements

## Performance

* ScreenGenius shall respond to user interactions and API requests within a reasonable time frame, typically less than 6 seconds.

## Security

* ScreenGenius shall implement appropriate security measures, including secure authentication and data encryption for API interactions.
* Comply with data privacy regulations such as GDPR (General Data Protection Regulation) and ensure that user data is handled in accordance with legal requirements.
* Audit Trails: Maintain comprehensive audit trails to track user activities, including screen captures and editing actions, for security and accountability purposes.
* Security Updates: Stay up-to-date with security patches and updates for all third-party libraries, frameworks, and dependencies used in the application.
* Secure API Design: If applicable, design APIs with security in mind, implementing authentication and authorization mechanisms for API access.

## Usability

* ScreenGenius shall provide a user-friendly interface with clear instructions for capturing, integrating, and managing screenshots/videos.
* Users shall have the ability to configure the quality and resolution settings for screen recordings to balance file size and video quality.

## Reliability

* Screen Genius shall be highly available and reliable, with minimal downtime and 99.9% uptime during normal operations.
* The tool shall be designed to gracefully handle system failures, ensuring that user data is not lost or corrupted, and that the tool can recover without data loss.

## Compatibility

* ScreenGenius shall be compatible with major chrome browsers and operating systems.
* Support Chromium browsers with Min optimization.

## Scalability

* ScreenGenius shall be scalable to accommodate increased user loads, additional features, increasing data volume.

# Design & Implementation Constraints

* Integration with third-party APIs requires authentication and adherence to their respective terms of use.
* Payment processing for subscriptions depends on third-party payment gateways (PayPal).
* Compliance with privacy regulations and data protection laws is mandatory.
* Compliance with google chrome store regulations.