Software Requirements Specification

Connexity ShopYourLikes

Link Journal

Ronald Gao - 904476004
Luis Lino - 704315045
Vincent Cheng - 604584095
Juan Hernandez - 904476103
Benjamin Hutchins - 204636614

1 Introduction:

1.1 Purpose

The following Software Requirements Specification(SRS) document will provide the basis for the overall vision and description of the functionality that the web application should provide as well as the constraints that should be satisfied and the external interfaces that should be supported. Each of the features specified should all aim towards providing users with an easy to learn interface and providing software analytics that are relevant to non-technically inclined users.

1.2 Definitions and Acronyms

SYL - ShopYourLikes, the name of the application responsible for handling SYL links and content creators' information.
User - Content creators and social influencers who generate SYL links to a retailer's link and are paid per clicks and sales leading from the SYL link.
Instagram - A social media platform that allows people to post pictures on their personalized account with a brief description. People who opt to "follow" an account will receive updates when that account adds a new post.
Instagram Shop - An internal shopping mechanism that allows Instagram users to advertise a merchant's content in a simpler way.
API -

1.3 Project Scope

The SYL application servers as a way to connect content creators and social influencers to merchant retailers through the generation of an application specific SYL link that maps 1-1 to a URL belonging to the product that is to be advertised. The users have the option of managing the SYL link in various ways from a simple hyperlink included in personal websites, blogs, or social media platforms, to a more integrated environment by allowing SYL to access to their Instagram story and basic personal information that allows users to post SYL links tagged on an Instagram picture, this allows integration with the Instagram Shop.

2 Overall Description:

2.1 Project Perspective

The SYL Link Journal provides a convenient and detailed web interface for investigating analytics of ShopYourLikes links. ShopYourLikes is primarily used by invited social influencers to generate monetized links to retailer items. The current SYL web interface does not provide certain desirable features such as batch URL conversion or specific link analysis, so the SYL Link Journal provides a way for social influencers to access this data if desired. The SYL Link

Journal extends the functionality of the main SYL website, so while the journal exists independently of the main website, the journal's users remain the same: invited social influencers.

To accomplish its tasks, the SYL Link Journal integrates with the ShopYourLikes API so that the users can view the effectiveness of their individual URLs and groups of URLs. For each SYL link, the SYL Link Journal provides detailed information such as the number of clicks, earnings generated, and primary geographic origin of clicks. This data can also be viewed at a group level, in which links can be categorized by retailer, popularity, and other factors.

In other words, the SYL Link Journal allows the user to look through an ongoing history of their generated SYL URLs. They can create re-copy old URLs, create new URLs, and see the images and other metadata associated with URLs. This journal is designed to be as user-friendly as possible, with easy navigation and mobile-friendly design.

The Link Journal also comes with some extra functionalities, one of which is the ability to convert batches of URLs into ShopYourLike URLs through a user interface. Another extra functionality is the use of external APIs (the specific origins of these APIs are up to change and will be described in the interface section) to help visualize a user's data. For example, social influencers may view a heat map of the geographic locations of their links' clicks, helping them view where their primary followers live.

3 External Interface Requirements:

All the requirements for the expected input and output of the "Connexity ShopYourLikes" application are listed here.

3.1 User Interface

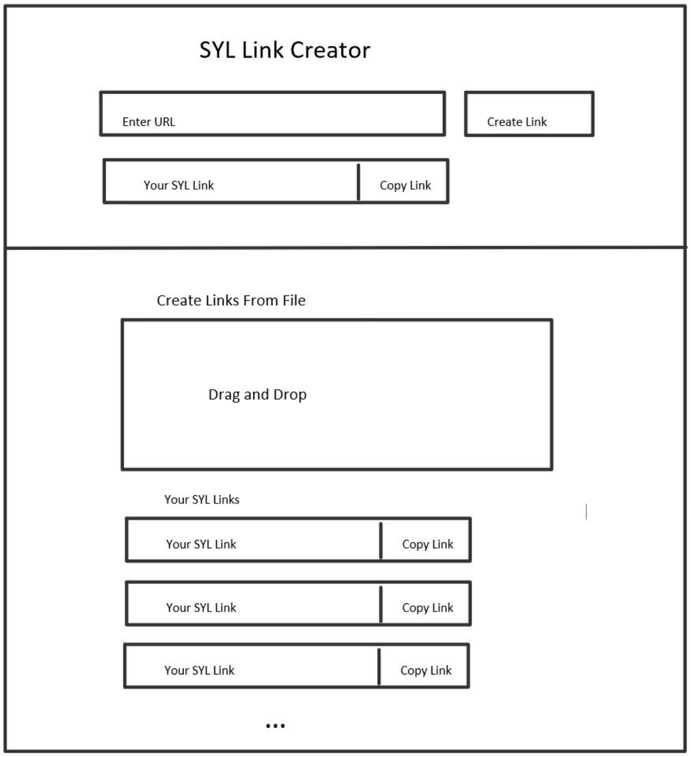
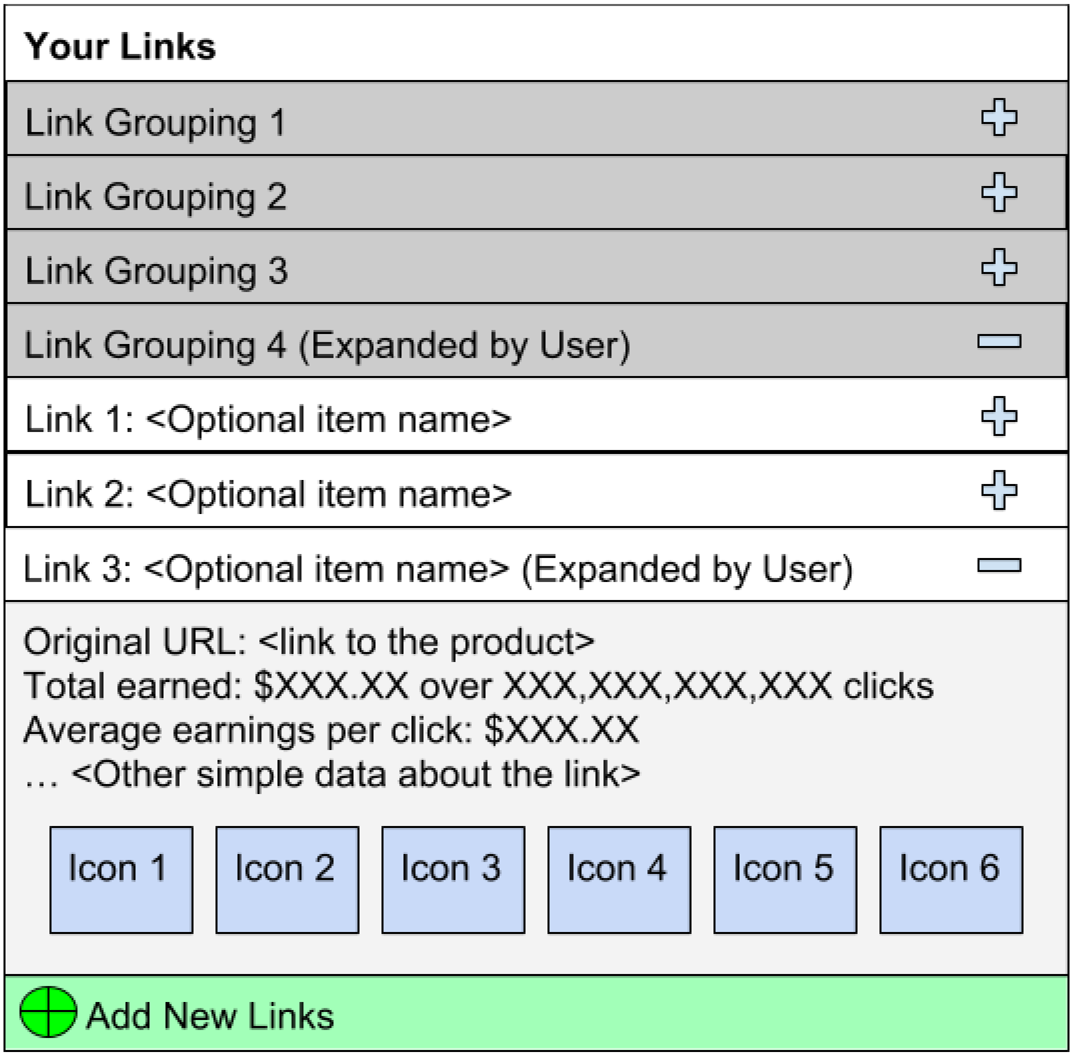


Figure 1. SYL Link Creation Page. Users can enter single links and get back an SYL link. Users can also batch create SYL links from a file (e.g. excel sheet of URLs).



Your Links

Link Grouping 1 ✚

Link Grouping 2 ✚

Link Grouping 3 ✚

Link Grouping 4 (Expanded by User) ▭

Link 1: <Optional item name> ✚

Link 2: <Optional item name> ✚

Link 3: <Optional item name> (Expanded by User) ▭

Original URL: <link to the product>
Total earned: $XXX.XX over XXX,XXX,XXX,XXX clicks
Average earnings per click: $XXX.XX
… <Other simple data about the link>

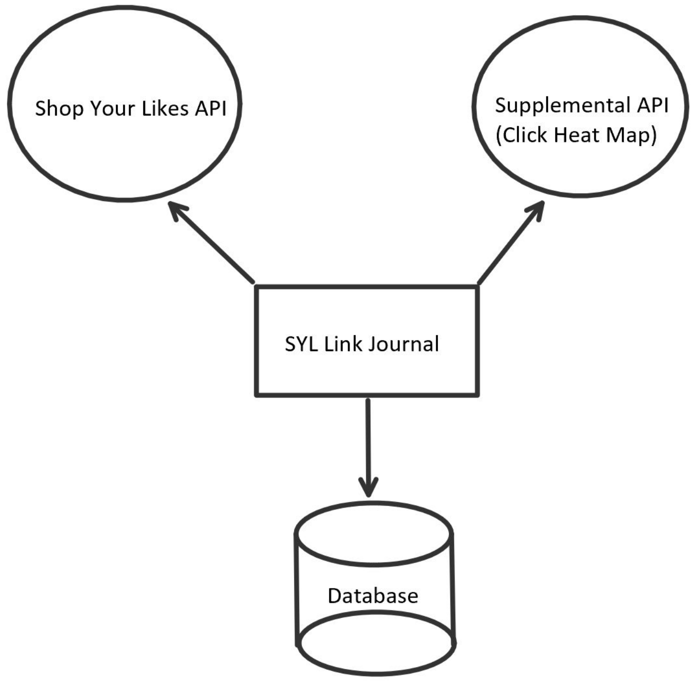
Figure 2: Example Link Journal interface. The user can expand link groupings to see individual links within that grouping. The user can also expand the individual links to see analytical data about those links.

3.2 Software Interface

“Connexity ShopYourLikes” is a web application so, naturally, the website domain and hosting will be done on Connexity servers. The application will be done in Java as typically Java is one of the fastest to deploy and to follow the technologies used by the main SYL website.

The application will use Spring Boot to create a stand-alone, production-grade with minimum configuration, which will also help get development done as fast as possible. Moustache is the templating framework that will be used since that is already the third-party software chosen in Connexity API. Persistent data such as links and link metadata will be stored in a database like PostgreSQL (subject to change).

Users should be able to provide multiple links to the application in order for “Connexity ShopYourLikes” to convert all the links into their monetizable format. So, naturally the application has to be connected to the ShopYouLikes API that Connexity provides. Finally, there will be an external API that is used in order to create a heatmap. For this, we will use a third-party software called heatmapper.js in order to graphically represent demographics of the location of links and accesses.



4 Functional Requirements:

ID: FR1

Feature: Single Link Creation

- Users can enter a single URL and get back a SYL link

ID: FR2

Feature: Batch Link Creation

- Users should be able to provide multiple links at once to the application and then get back a batch of converted links

-----

ID: FR3

Feature: Link Sorting

- Users can switch between different sorting options for viewing their list journal
- sorting options should include: alphabetical, date created, revenue earned, and number of clicks.

ID: FR4
Feature: Seeing Characteristics of the link after tapping or clicking on it
• Users should be able to see corresponding link attributes like number of clicks on that link, how much money the link made, etc.

ID: FR5
Feature: Putting a link into a certain group
• Users should be able to move the link to any group that they wish even though there will be an initial grouping already
-----
ID: FR6
Feature: Link Searching
• Users should be able to search their journal for links based on a variety of attributes (tag, retailer, link name, etc.)

ID: FRE1
Feature: Click Heat Map
• Users should be able to view a geographical heat map of the origins of clicks on their links

5 Non-functional Requirements:

ID: U1
Title: User-friendly Interface
• The interface should be designed in such a way so that the user can easily navigate
• The interface should also be pleasant on the eyes, with the links laid out in an aesthetic fashion
• Note: This requirement is notably vague, but still a significant consideration

ID: U2
Title: Mobile-friendly
• The interface should work easily with mobile devices
• Functionalities should be compatible with a touch interface, and interface elements should scale based on screen size

ID: P1
Title: Response Time
• All requests to the web interface should respond quickly (at most 2 seconds under standard server load)

ID: A1
Title: Database Atomicity

- A database transaction should either occur in its entirety and change nothing
- EX: Adding multiple links to the journal should add them all, or throw an error and add nothing

ID: C1
Title: Database Consistency

- Any transaction with the database should bring it from one valid state to another valid state

ID: I1
Title: Database Isolation

- Multiple users should be able to concurrently use the website at the same time without destroying the integrity of the database
- In other words, it should be as if their access is used sequentially

ID: D1
Title: Database Durability

- All requests from the user that affect the database should actually result in permanent changes to the database
- If this is not possible due to connection issues (or other issues), the user should be notified with a message/warning

ID: SC1
Title: Scalability - Users

- The database usage should scale efficiently with multiple concurrent users

ID: SC2
Title: Scalability - Links

- The database usage should scale efficiently for users with lots of links

ID: SC3
Title: Scalability - Clicks

- The database usage should scale efficiently for links with a lot of clicks

ID: M1
Title: Maintainability Constraint

- Code should be written in a way such that it can be easily modified in the event of requirements changes

ID: S1
Title: User Restrictions
- Users should be unable to access the website without proper credentials (an invite)

ID: CO1
Title: Compatibility
- The website should, at the very least, be compatible with Google Chrome

6 Project Timeline:

04/23/2018	Project timeline and SRS v.1
04/27/2018	SRS v.2
04/30/2018	Set up development environment
05/07/2018	Basic working implementation
05/11/2018	Midterm report, presentation, and demo
05/14/2018	Add features, integration testing, documentation
05/21/2018	Finish adding features, refactoring
6/06/2018	Final Report
6/08/2018	Final Presentation and Demos