Software Requirements Specification for "Links"
Version 1.0

Abhijith Madhav, Arjun S Bharadwaj

February 7, 2014

Revisions

.

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Abhijith Madhav(MT2013002) Arjun S Bharadwaj(MT2013026)	Initial version sent to Client	February 5, 2014
2.0	Abhijith Madhav(MT2013002) Arjun S Bharadwaj(MT2013026)	Elaboration of product scope. Made interoperability as an FR.	February 7, 2014

Contents

1 Introduction

This document provides the Software Requirement Specifications of the project. Specifically it provides the details on the functional and non functional aspects of the project.

1.1 Document Purpose

The document describes the software requirements of "Links", version 1.0. The software requirements encapsulate the whole of the product.

1.2 Product Scope

"Links" is website offering bookmarking services with the specific requirement being that it should be deployable on a private server within an organization.

The purpose is to enable people in an organization to share bookmarks with each other.

The specific benefit delivered by "Links" is that sharing/publishing of bookmarks can be controlled at the required level of granularity.

"Links" is intended to be an extensible, modular system with an exposed API. Third parties must be able to write plugins and applications using this API.

It is envisioned to seemlessly fit into the "share" motto of the net community, be it through simple email or through popular social platforms like facebook and twitter. A trigger based action execution like that of ifttt.com augments this motto and is intended to be a part of "Links".

2 Overall description

This section provides the overall description of the project. This includes the product perspective and the functionality.

2.1 Product Perspective

A bookmarking service is a centralized online service which enables users to add, annotate, edit and share bookmarks of web documents, the most popular being Delicious. "Scuttle" is one of the other well known open source alternative. "Links" is envisioned on similar lines with the specific objective that it is aimed at deployment within an organization.

2.2 Product Functionality

1. Provide for user accounts/signup.
2. Provide for website bookmark management.
3. Provide for management of user groups for sharing bookmarks.
4. Provide for options to share website bookmarks amongst users and groups.
5. Provide options to push website bookmarks onto popular social media and via email.

3 Functional Requirements

This section specifies the functional requirements of the project.

3.1 Functional Requirements

This section provides the functional requirements of the project.

1. Login
    - User logs into the system.
    - *Input:* User enters the login details and clicks on login.
    - *Output:*
        - If the validation of the user credentials is successful:
            - The user is redirected to the Homepage.
        - If the validation of the login details is unsuccessful:
            - An appropriate error message is shown to the user.
2. Signup
    - User clicks on Signup.
    - *Input:* Signup details are asked.
    - *Output:*
        - If validation of the details is successful:
            - Account is created.
        - If validation of the details is unsuccessful:
            - An appropriate error message is shown to the user.
3. Save Links
    - User bookmarks the links.
    - *Input:* A Title, URI, Tags and Annotations are provided.
    - *Output:*
        - After the link is saved:

        * Acknowledgement for the action is provided.
        * Suggest tags if same link was shared by others based on the visibility level of the user.

4. Edit Links

    - User edits the links.
    - *Input:* An existing link, modified data (title, tags, annotations).
    - *Output:*
        - After the link is updated:
            * Acknowledgement for the update is provided.

5. Delete Links

    - User deletes the links.
    - *Input:* An existing link shared by the user.
    - *Output:*
        - After the link is deleted:
            * Acknowledgement for the deletion is provided.

6. Create User Groups

    - Creation of Groups by the user.
    - *Input:* Group name and group description corresponding to the group.
    - *Output:*
        - If the group name is not already used:
            * Acknowledgement for the creation is provided.
        - If the group name already exists:
            * Appropriate error message.

7. Join Group

    - User joins the group.
    - *Input:* Group name of the group that the user wishes to subscribe to.
    - *Output:* Acknowledgement of the subscription is provided.

8. Unjoin Group

    - User unjoins the group.
    - *Input:* Group name of the group that the user wishes to unsubscribe to.
    - *Output:* Acknowledgement of the un-subscription is provided.

9. Share

- Sharing of the links.
- *Input:* A set of bookmarks to be shared via email or popular social media
- *Output:* Share bookmarks via the specified media.

10. Add Members
    - Add members to the group.
    - *Input:* User names by the group owner.
    - *Output:* Acknowledgement of addition to group.
11. Remove Members
    - Remove members to the group.
    - *Input:* User names by the group owner.
    - *Output:* Acknowledgement of deletion to group.
12. Interoperability
    Expose APIs through which the following can be done
    - Search based on bookmarks, annotations, tags in accordance to the visibility of the user.
    - Add, delete and edit the bookmarks in accordance to the visibility of the user.

    As a proof of concept, an android application that consumes the data from the provided APIs will be created

4 Non-functional Requirements

This section specifies the non-functional aspects of the project.

4.1 Performance Requirements

1. Any interactive transaction with “Links” should not take more than 5 seconds

4.2 Safety and Security Requirements

1. All users must be authenticated before being given modification privileges in “Links”.