Contents

[Glossary 4](#_Toc9527632)

[Objectives 5](#_Toc9527633)

[Application Features 6](#_Toc9527634)

[Unauthenticated User 6](#_Toc9527635)

[AUTHENTICATED ORDINARY USER 6](#_Toc9527636)

[Authenticated User Administration 6](#_Toc9527637)

[Authenticated Admin User 6](#_Toc9527638)

[API Features 7](#_Toc9527639)

[Required Dashboard Design Elements 8](#_Toc9527640)

[Permissions 10](#_Toc9527641)

[Permissions 10](#_Toc9527642)

[Roles 10](#_Toc9527643)

[Roles & Permissions Map 10](#_Toc9527644)

[Key: 10](#_Toc9527645)

# Glossary

| Term | Meaning |
| --- | --- |
| 200 OK | The request has succeeded. The information returned with the response is dependent on the method used in the request. |
| admin | Administration |
| API | Application Programming Interface |
| BRAS | Browse, Read, Add, Search |
| BREAD | Browse, Read, Edit, Add, Delete actions as performed on a database within an application. |
| BREADS | Browse, Read, Edit, Add, Delete, and Search.  The BREAD concept with search capabilities highlighted. |
| BRS | Browse, Read, Search |
| CRUD | Create, Retrieve, Update, and Delete actions.  Another way to look at the basic database operations. See BREAD. |
| DELETE | DELETE is pretty easy to understand. It is used to \*\*delete\*\* a resource identified by a URI. |
| GET | The HTTP GET method is used to \*\*read\*\* (or retrieve) a representation of a resource. |
| guest | An unauthenticated user who has not registered, or has not logged into the system |
| HTTP | HyperText Transfer Protocol |
| HTTPS | HyperText Transfer Protocol, Secure |
| JSON | JavaScript Object Notation |
| JWT | JSON Web Token |
| MPA | Multi Page Application |
| PATCH | PATCH is used for \*\*modify\*\* capabilities. The PATCH request only needs to contain the changes to the resource, not the complete resource. |
| POST | The POST verb is most-often utilized to \*\*create\*\* new resources. |
| PUT | PUT is most-often utilized for \*\*update\*\* capabilities, PUTting to a known resource URI with the request body containing the newly-updated representation of the original resource. |
| REST | Representational State Transfer |
| SPA | Single Page Application |
| tag | A word or phrase used to identify, or classify, and item |
| UI | User Interface |
| User | The basic client of the application |

# **Objectives**

Create a bookmarking web application that has the following features:

1. User Profile system including storing of a photo
2. Tagging of Bookmarks
3. BREADS -> bookmarks and tags.
4. User vs Admin Views
5. Private and Public bookmark support
6. An API to allow mobile applications and alternative UI’s to access the database.

# Application Features

Each user level will have a set of permissions and features they will be allowed to perform.

## Unauthenticated User

An unauthenticated user needs to be able to:

* Register (with email verification),
* Login to an existing account,
* Reset their password for their existing account if they have forgotten it, and
* Perform BRS on non-private bookmarks.

## Authenticated Ordinary User

An authenticated ordinary user should be able to:

* Edit their own user profile,
* Perform BREADS on their own bookmarks, and
* Perform BRAS for tags.

## Authenticated User Administration

An authenticated user admin user should be able to:

* Reset a user password,
* Suspend a user,
* Unsuspend a user,
* Send a user password reset link, and
* Update basic details for a user account.

## Authenticated Admin User

Lastly, an authenticated admin user is required to be able to:

* Perform BREADS on any bookmark,
* Perform BREADS on any tag,
* Perform BREADS on any user, and
* Perform BREADS on any profile.

# API Features

The API will provide a method for the development of alternative front ends such as Native and Hybrid mobile device applications, Native and Hybrid desktop applications.

To allow this the API requires:

* The features of a Unauthenticated and Authenticated User
* Security for API access

# Required Dashboard Design Elements

The following are the features for each class of application user:

|  |  |
| --- | --- |
| Unauthenticated User (Guest) | MoSCoW |
| Home page access | M |
| Show last 10 bookmarks (non-private) | M |
| Show total bookmarks | S |
| Search capability | M |
| Site colour scheme | S |

|  |  |
| --- | --- |
| Authenticated User (Ordinary) | MoSCoW |
| Profile | M |
| Bookmarks | M |
| Tags | M |
| List of Own Bookmarks | M |
| User colour scheme | S |

|  |  |
| --- | --- |
| Authenticated User (Admin) | MoSCoW |
| Status of ‘App’ | M |
| Bookmarks | M |
| Users | M |
| Tags | M |
| Logged in users | M |
| Admin colour scheme | M |

|  |  |
| --- | --- |
| Authenticated User (User Admin) | MoSCoW |
| Status of ‘App’ | M |
| Users | M |
| Logged in users | M |
| User Admin colour scheme | C |

# Permissions

The site has a number of permission levels and roles.

## Permissions

|  |  |  |
| --- | --- | --- |
| Browse | Read | Search |
| Add | Edit | Delete |
| Authenticated | Suspended | Unauthenticated |

## Roles

|  |  |
| --- | --- |
| Administrator | User Administrator |
| User | Guest |

## Roles & Permissions Map

The table below shows the permissions to user role mapping. Details about specifics follows after the table.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Admin | User Admin | User | Guest | Suspended |
| Users | BREADS\* | BREADS1 | BRS | BRS | P0 |
| Tags | BREADS\* |  | BREADS | BRS |  |
| Links | BREADS\* |  | BREADS | BRS |  |
| Profiles | BREADS\* | BREADS\* | BREADS |  |  |

### Key:

P0: Password reset via email

BREADS1: Only able to perform action on ordinary users

BREADS2: Currently authenticated in user

BREADS3: Owned by currently authenticated admin user

# Bibliography

Fredrich, T. (2018, July 04). *REST API Tutorial*. Retrieved May 13, 2019, from https://www.restapitutorial.com/

REST Cheat Sheets. (2017, September 29). *API Design Cheat Sheet*. Retrieved May 23, 2019, from https://github.com/RestCheatSheet/api-cheat-sheet

REST Cheat Sheets. (2017, November 16). *Platform Cheat Sheet*. Retrieved May 23, 2019, from https://github.com/RestCheatSheet/platform-cheat-sheet