Links - Project Proposal

Abhijith Madhav, Arjun S Bharadwaj

Common Utilities for OS funtion simulators

Team Members

Name	Role No.	Email id
Abhijith Madhav	MT2013002	abhijith.madhav@iiitb.org
Arjun S Bharadwaj	MT2013026	arjun.s.waj@iiitb.org

Supervisor

Prof. Srisha Rao.

Version

Sl. No.	Date.	Version No.
1	29/03/2014	1

Project Timelines

- Start Date: 31/01/2014
- End Date: 30/06/2014

Objectives of the project

Traditional bookmarking services do not allow storing information on private servers. The application should allow deploying a local server which is capable of the following for every user/client:

- Bookmarking URLs
- Tagging the URLs
- Searching among the links
- Annotations for a given URL
- Multi-user
- Expose an API

Functionalities

This section provides the list of functionalities/features that are planned to be supported in the course of this project.

- Create user accounts/signup.

- Links Management
  - Save links.
  - Edit links.
  - Delete links.
  - Tag links.
  - Annotate links.
  - Classify/Organize links in folders.
- Group Management
  - Create user groups.
  - Join group.
  - Unjoin the group.
  - Add members to the group (by group owner).
  - Remove members from group (by group owner).
- Share Options
  - Via Email.
  - Via Twitter.
  - Via Facebook.
  - Share links with groups.
- Suggestions
  - Suggest tags if same link was shared by others if the links privacy is public.
  - Suggest public links belonging to the same tags.
- Expose APIs
  - Search based on links, annotations, tags at various granularity levels of ownership of the links.
  - APIs to add, delete and edit the links.
  - Android App that consumes the data from the provided APIs.
- Shorten the links - URL shortener.

Project Deliverables

This section gives an overview of the deliverables of the project 'Links':

Milestones

The project 'Links' shall have the following milestones that shall be delivered.

- Freezing on the project requirements through requirement gathering sessions with the client.
- Finalization of the Software Requirement Specifications composing both the functional requirements and the non functional requirements.
- High level design including the Architecture Design of the core system and the plugin interface.
- Low level design including the UML diagrams that conforms to MVC architecture.
- Implementataion of the basic and the core functionality of the system.
- Implementataion of the plugin architecture to support plugins.
- Implementation of the REST APIs.
- Implementation of the Android App to consume data through REST APIs.
- Unit Testing of the modules.

Estimated total time

Sl. No.	Item	Duration (in hours)
1	Analysis	10
2	Requirements Gathering	10
3	SRS	15
4	Use Cases	15
5	High Level Design	30
6	Low Level Design	40
7	Implementation	100
8	Testing	30

H/W and S/W requirements

This section deals with the H/W and the S/W requirements for the implementation of the project 'Links'.

H/W requirements

The server should have the following configuration:

- 32/64-bit processor.
- At least 2 GB of physical memory.
- At least 10 GB of free space.

S/W requirements

The server should have the stable version of the following softwares installed:

- Unix based OS.
- Ruby.
- Rails.
- MySQL.

Standards

The project conforms to the following standards:

- Separation of Concerns as laid down by MVC framework.
- Conventions over Configurations as laid down by Rails framework.
- DRY - Don't Repeat Yourself as laid down by Rails framework.
- Mapping resource to the URLs as laid down by REST architecture.

Technology/Architecture

The project 'Links' uses the MVC architecture for the implementation. Below is the architecture diagram of the same.

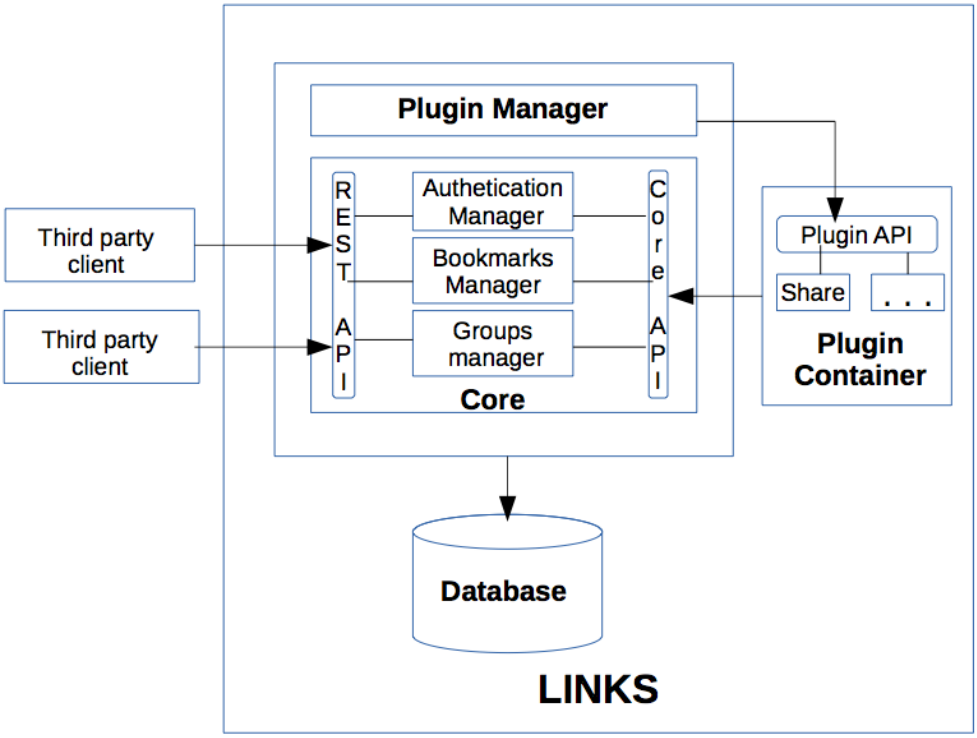

Figure 1: Links - Architecture Diagram.