Rake™ - All in One Mobile application Scraping Tool.

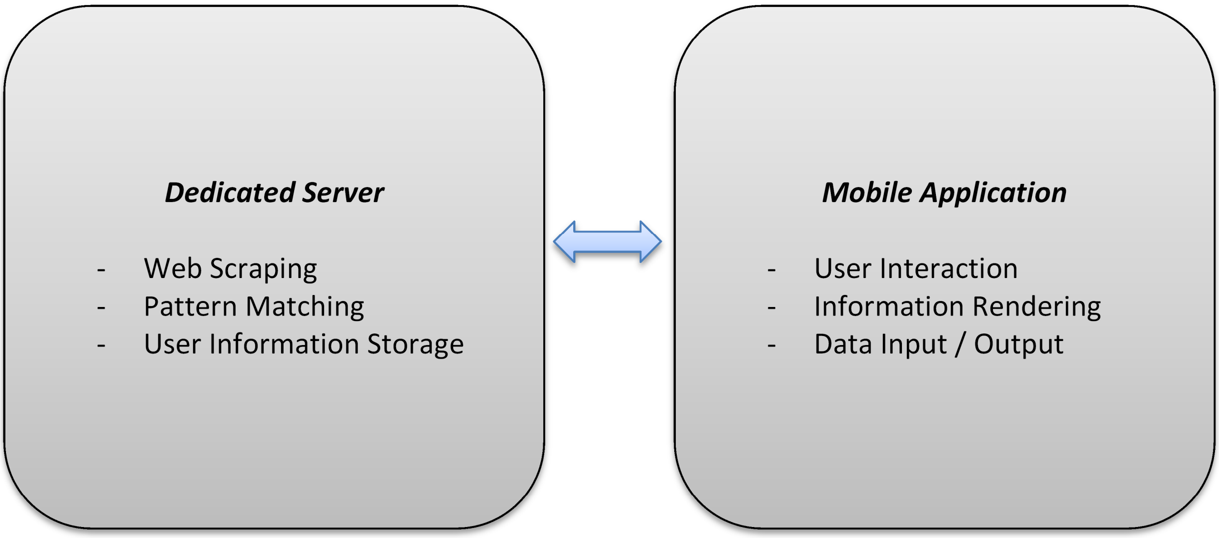
Anthony Silvano A. Suan

Abstract:

The project involves on developing an **all in one scraping system** that **extracts**, **transforms** and **renders** information from external web sources into a single mobile application. The project converts free-form public HTML and other mirrored or extended format, into a user friendly XML format or an RSS feed, by means of applying **search-patterns**. The output snippets will be joined by using output templates. The system will feature sharing environments for users defined **search-patterns / macros** and **output templates.**

The purpose of the project is to help mobile users remove the hassle of browsing their favorite blogs or news sites in order to get latest updates. The project will also have a web version for other platform users.

The project will be implemented in **two-tier**, **thin client** architecture. The system is composed of a dedicated **web server** and **a mobile app**. The dedicated web server performs the heavy computational roles such as web scraping and string search/processing. The mobile app provides the server to user interaction and favorable user experience. The communication between the server and the mobile app is through **AJAX**. The data interchange format is **JSON.**



Dedicated Server

- Web Scraping
- Pattern Matching
- User Information Storage

Mobile Application

- User Interaction
- Information Rendering
- Data Input / Output

Server Details:

Mobile Application Repository
https://github.com/anthonysuanschool/ict249/tree/master/Rake%20Project/mobile

Server Files Repository
https://github.com/anthonysuanschool/ict249/tree/master/Rake%20Project/server

Mobile Application Download Link
http://rake.anthonysuan.com/mobile/

Web Server Link
http://rake.anthonysuan.com/

User Story

There is only one type of user except of course for the server admin. The mobile user has the capability to login, logout, create scraper, view scraped information, create/delete/share/view search patterns, create/delete/share/view output templates, and logout.

Use Cases:

- Create Scraper (6hrs)
  - Manage Scraper Information (4 hrs)

Test Case	Expected Results
Input scraper information such as url and document type.	Store information to database and proceed to scrap contents.

  - Scrape Contents (2 hrs)

Test Case	Expected Results
Detect document type.	Document type detected.
Scrape page contents.	Fetch the public html strings.

- Create Search Pattern (20 hrs)
  - Create Interface from regular expression to user friendly macros (10 hrs)

Test Case	Expected Results
Put user friendly pattern as input.	Generates a valid regular expression.

- Create Extraction Rules (4 hrs)

Test Case	Expected Results
Put {o} as input string	Determines that the string is considered as item/items of interest.
Put {*} as input string	Determines that the string is any item/items.

- Create Global Search Pattern Rules (2 hrs)

Test Case	Expected Results
Input extraction rules.	Fetch global items. Global Items examples are strings that appear only once.

- Create Repeatable Item Search Pattern Rules (2 hrs)

Test Case	Expected Results
Input extraction rules.	Fetch repeatable items. Repeatable items examples are strings that appear once or several times.

- Create tables for defined search pattern (2 hrs)

Test Case	Expected Results
Manipulate table using the database wrapper.	Database is changed according to the manipulation.

- Create Output Templates Implementation (11 hrs)
  - Global Template short-code Implementation (4 hrs)

Test Case	Expected Results
Input array of data and an xml template with shortcodes. Shortcodes are equivalent representation of the actual data.	Generates the complete output xml with the actual data.

- Repeatable Item Template short-code Implementation (5 hrs)

Test Case	Expected Results
Input array of data and an xml template with shortcodes. Shortcodes are equivalent representation of the actual data.	Generates the complete output xml with the actual data.

- Create tables for defined output templates (2 hrs)

Test Case	Expected Results
Manipulate table using the database wrapper.	Database is changed according to the manipulation.

- Create User Management Tool (6 hrs)
  - Database and tables implementation. (2 hrs)

Test Case	Expected Results
Manipulate table using the database wrapper.	Database is changed according to the manipulation.

  - Create User information (1 hr)

Test Case	Expected Results
Manage user.	Changes are done in accordance to the operations related to the user

  - User to Templates Association (3 hr)

Test Case	Expected Results
Associate templates to user.	User can manage templates associated.

- View Scraped Contents (8 hrs)
  - Aggregate JSON / XML (4 hrs)

Test Case	Expected Results
Request content to the server given user and scraper.	Server response is a JSON / XML String of the contents.

  - Render Contents (4 hrs)

Test Case	Expected Results
Put JSON / XML String and output template as input.	Generates the complete valid html string.

- Search / View / Use shared Scraper from other Users (6 hrs)
  - Search Scraper Item (2 hr)

Test Case	Expected Results
Enters a name of the scraper.	As the user enters a character, there is a suggestion to the scraper names it can find related to the current text entered.

- View Scraper Item (2 hrs)

Test Case	Expected Results
View Scraper.	Scraper Information can be viewed. Such as name, url, search patterns and output templates. Also the the user reviews on how many stars(1-5) it is rated.

- Use Scraper Item (2 hrs)

Test Case	Expected Results
Use Scraper.	It will be added to the users scraper list and can be used but cannot be edited by the user.

- Miscellaneous and Helpers: (8 hrs)
  - Create Database (MySQL) wrapper and CRUD functions (4 hrs)

Test Case	Expected Results
Connect to database.	Database connected.
Disconnect from database.	Database disconnected.
Insert into database.	Adds new item on database.
Remove from database.	Deletes selected item on database.
Select from database.	Retrieves items from database.
Update database.	Edits selected item on database.

  - Create JSON / XML Wrapper (4 hrs)

Test Case	Expected Results
Create JSON string from array.	Generates JSON string.
Create XML string from array.	Generates XML string.

  - User Interfaces (20 hrs)

Test Case	Expected Results
Test all based on user point of view.	Responded accordingly.

*Total of **83 hours** to implement all the use cases, excluding debugging and testing. Project timeframe is **2 months**. Weekly working hour budget is approximately **11 hrs/week**.*

Terms:

- <u>Whats with the name “Rake”?</u>
  - Inspired by the literal rake tool.

- <u>Scraper</u>
  - In this project, a scraper is the item that the user creates in order to defined get the contents from external web source. A scraper is composed of the web url, search pattern and output template. Scraper is also the main sense of this application.

- <u>Macro</u>
  - In this project, a macro is the dynamic representation of the part of the string to be search.
  - Example {o} and {*}

- Shortcode
  - In this project, similar to Macro, a shortcode is a representation of the actual data.
  - Example [g1], [g2], [g3] “g” means global items. [r1], [r2], [r3] “r” means repeatable items