



Term Project Proposal

Capstone Project

Version 1.0.1

Zekai Lin Shizhan Ma Zhenyang Li

Wentao Chen

CIS 389: Web System Arch/Prog  
 Professor: Edmund Yu

Last Updated: Nov.13 2023

# Modified Record

( ps. most updated version : [Proposal.docx](https://docs.google.com/document/d/1rH4Eu2JNPgtP_h9i1nisRK9-5EAcSJXx/edit?usp=sharing&ouid=112602052240691928431&rtpof=true&sd=true))

## \*todos

* maybe add more features on tasks, such as user login, cookbook bind with user
* develop schedule update
* UML Class diagrams

## ModifiedHistory

| **Date** | **version** | **Title** | **Comment** | **People** |
| --- | --- | --- | --- | --- |
| Nov.13 23 | V 1.0.1 | Submission | Submission for the Project Proposal | Zekai L |
| Nov.27 23 | V 1.0.2 | Project Finished | Updated Tasks Progress | Zekai L |
|  |  |  |  |  |

# 

# Outline:

Modified Record

Project Detail, Team Member General & Team Rules

Intro to the Existing Open Source Web Project

Proposal Description

* Short Description
* System Structure
* Tech Stack

Development Process

Plan & Documentation

Develop Tasks

* front end
* back end
* Combined

Development Schedule

Reference

**Project Details**

| **Option:** | Option 2: Build a web site, using an existing Web server, |
| --- | --- |
| **Project Name:** | Cook Book |
| **Description:** | Back End Server, with an Admin Website for a Simple Open Source Web App |

**Team Members & Roles**

| **Team Members** | **Contact Information** |
| --- | --- |
| Zekai Lin <Lead> | [zlin48@syr.edu](mailto:zlin48@syr.edu) |
| Shizhan Ma | sma100[@syr.edu](mailto:kchen38@syr.edu) |
| Zhenyang Li | zli115[@syr.edu](mailto:bkhatiwo@syr.edu) |
| Whentao Chen | wchen01@syr.edu |

**General Team Rules:**

* Respect each other's suggestions.
* Must be on time for all meetings and due dates.
* Allow each team member to speak without judgment and interruption.
* Communication about being absent or inability to complete work should be 24 and 48 hours in advance respectively.
* All members must contribute a minimum of 15% of their work to each assignment.
* Require consensus on all project decisions.
* Accept responsibility and accountability along with the authority given.
* All team members confront issues directly and promptly with the group lead or as a whole group

# Proposal Description

## Short Description

For existing options, after meeting with group mates, we came up with an agreement to build a website. Considering the limit of time, after negotiating with the Professor, we will develop a **Back End**, with an **Admin Website** for a Simple Open Source Web App.

## Intro to the Existing Open Source Web Project



### Original Project: Simple Cook Book

Project Link: <https://github.com/YunYouJun/cook>

The original Website is an Online Cook Book, Users can select Vegitables, meat, main food, and cooking tools to filter what food can be cooked with chosen materials and methods.

Once there are any Results, the User can Just click the Result they Want to Check, and it will open the link showing how to make the food. It could be YouTube video links, social media posts and etc.

### What We Gonna Do

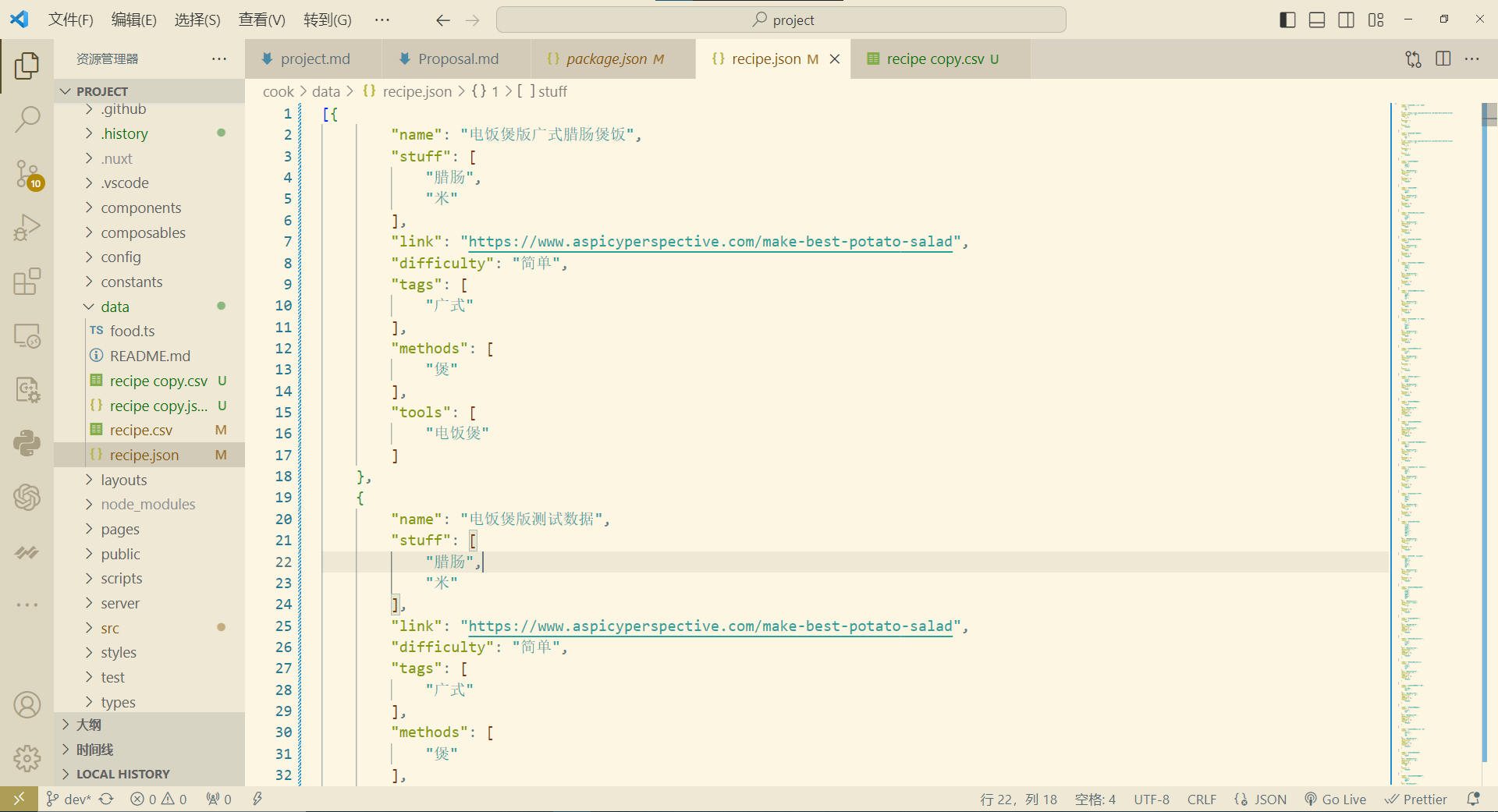
What we gonna do is mainly build an **admin website** with a **backend server** operating a database to store those food materials, available cooking methods, and dish info with corresponding URL links as a complement to the Existing project.

That means we want to build a REST website and treat the origin website as a front end. (Graph 1. Index of Simple Cook Book Web App)

In addition, We plan to add an English version to the website, so that it can be easily present in the final presentation. If Our Progress is Better than expected, we will also add some content or features to the existing web page.

### Why are we doing so

Once we checked the app, we found that the web-only store has limited food in the list. That is because the data (materials and dishes info) is stored in a JSON file locally.



(Graph 2. Food Data Stored in Local .json file )

And here comes a problem. If the User wants to modify the cookbook, such as by adding new food guide links or deleting hard food items, they need to dive into the project fundamentals and deal with the code directly. However, directly modifying the data is not convenient and may lead to strange problems.

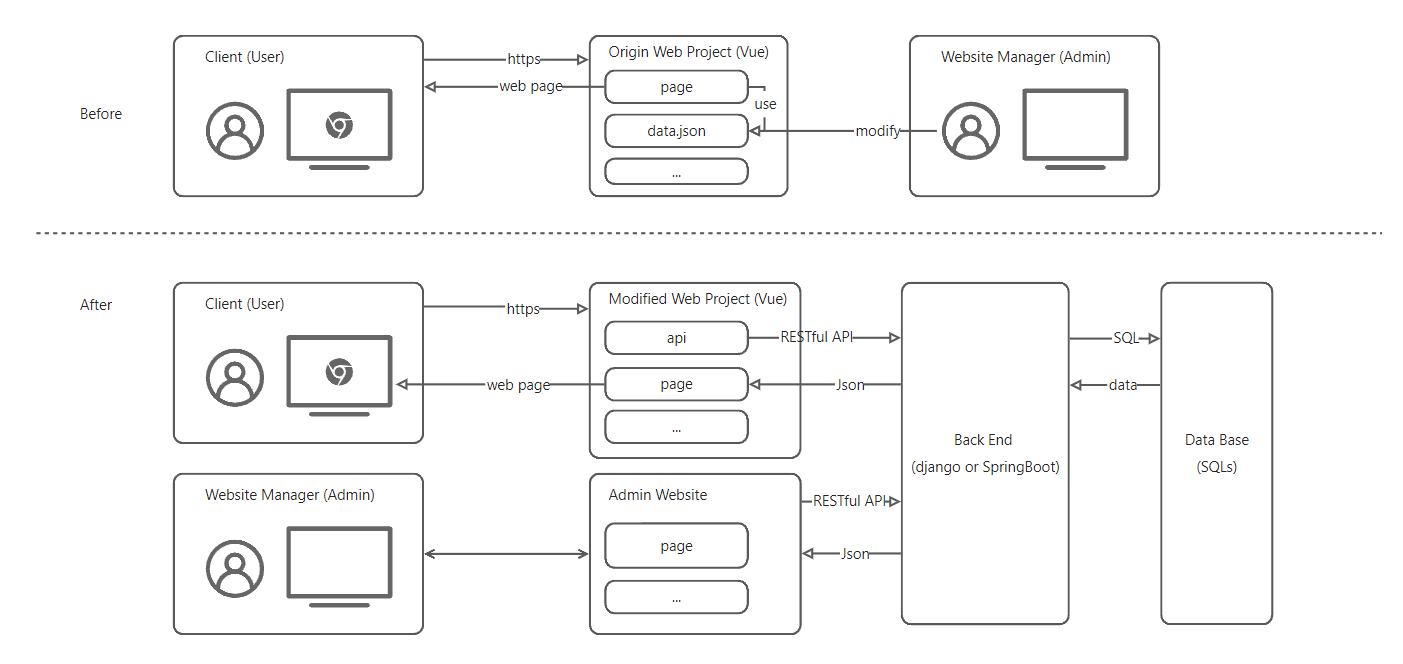
So we plan to build a **back-end admin website** storing the data instead, with a user-friendly User Interface to modify the food book items simply. And user clicks the search button on the original web page, the front-end server should send a request to a back-end server.

This type of website architecture is called **REST** (**Representational state transfer**) which is not only related to what we learned this semester but also commonly used in the Software industry. We believed that we would not only practice what we learned in the lecture but also get familiar with updated website-building techniques and tools.

REST(Representational state transfer) definition: <https://en.wikipedia.org/wiki/REST>

# 

## System Structure



(Graph 3. System Architecture )

To better explain what we mentioned in the Proposal description, we draw a diagram. The upper part is marked as before, showing the way the website's original structure. As our optimal part is under the dotted line shows, our work corresponding to the system structure is mainly three parts:

* Modified the original web project and added API of the back end
* back end and database server
* admin website

## Tech Stack

| **Usage** | **Tech Stack** |
| --- | --- |
| Front End | Vue, and Corresponding Packages… |
| Back End (Undecided) | Django(Python) or Springboot(Java) |
| Database (Undecided) |  |
| Version Control | Git & GitHub |
| Documentation | Github, Markdown, Google Drive |
| … | … |

# Development Process

## Develop Tasks

ps. Not necessary or low-priority goals marked as \*

### Plan & Documentation:

* [√] Project Proposal **Composition & Maintenance**
* [ ] API Documentation **Composition & Maintaining**
* [√] **Decide** Stack of the Back-End Server & Database
* [ ] \* **Draw** UML Class Diagram

### Front end

* [ ] **Develop** an English Version of the original Web Project
* [√] **Develop** the RESTful API according to the finished API Document
* [ ] \* **Plan & Develop** more features for the original Website Project

### Back end

* [√] **Read and learn** the decided backend framework
* [√] **Develop** the model part according to the finished UML diagrams
* [√] **Develop** the restful API according to the finished API documentation

### Admin Website

* [√] Filter components of stuffs according to type
* [√] to be continued

## 

## Development Schedule

| Date | Event & Task | Description & Comment | People | Status |
| --- | --- | --- | --- | --- |
| Nov.12 23 | Proposal Composition | First Submission of the project proposal | Zekai | **Finished** |
| … | Detailed Plan | coming soon … | Team | **Pending** |
| Dec.4-6 23 | Final Presentation |  | Team | **Pending** |

# Reference

Original Project Link: <https://github.com/YunYouJun/cook>

RESTful definition in wiki: <https://en.wikipedia.org/wiki/REST>