Recent Advances in Technology and Engineering (RATE-2017)
6th National Conference by TJIT, Bangalore

International Journal of Science, Engineering and Technology
An Open Access Journal

Smart Recipe Application for all your Culinary Needs

Shajan Jacob, Qubin Jose, Sadique Ali Khan, Jabez V, Sejal Nimbhorkar, Maria John

Abstract

Nowadays, people use various applications for ordering food, looking for recipes and restaurants, shopping for food and groceries. At present, autonomous mobile applications are available to find recipe, to search for the restaurants, to read the reviews of the restaurants, to order the food and to purchase grocery. In this paper, we present a mobile application to integrate all the above mentioned services into one unified application. This application is a food service platform that suggests, orders and guide to make recipes, a smart companion for home cooks and professional chefs. It also organizes recipes, create cookbooks, and manage health and diet. It helps users accomplish more in the kitchen with the available ingredients.

Keywords- recipe, culinary, food.

Introduction

In our daily lives we are constrained with busy schedules, most of us tend to make use of mobile applications for most of our daily activities including cooking. Cooking is an activity for which we need to take proper time, have the idea of the ingredients and the utensils that we would use in our recipe.

Now a days there are many cooking recipe websites and the recipe applications that are coming up with new recipes like *Kitchen Stories, BigOven, Cookpad,* and so on. These application provides us the details on how to cook and the filtering of the contents based on the ingredient to use and place independent recipes. In these kinds of recipe applications the user have to spend his time in browsing through the recipes provided and also face difficulty in gather the ingredients and the utensils required to cook and eventually fail to go ahead with the recipe preparation and end up in a starving condition.

It would always be a helpful and an excellent way if the user gets their recipe as accurate as possible and order the ingredient online and check the availability of the food in the restaurants nearby. In this paper we come up with an application where the user will be able to find his recipe based on the ingredients used, time to prepare, utensils that will be needed to cook, order the ingredients online, make the check-list, helps us to check-out the Best restaurants that provide this recipe meal to order the food or to eat out and other features as well.

The proposed system is very useful for any person who is facing culinary difficulties due to time constrained lifestyle. This application is very convenient as it integrates many autonomous mobile applications such as recipe finder, restaurant explorer, to read the reviews of the restaurants, to order the food and to purchase grocery.

It has been observed from personal experience that a lot of inconvenience is caused from using the existing systems like recipe apps, websites, apps for ordering food and groceries, and looking for restaurants all at once. There arose a need to bring all of these together into one single app and that was the motive behind the given proposed system.

The proposed system plans to bring together the functionalities of recipe lookup using advanced filters for finding recipes according to time required to prepare them, the utensils required, and the ingredients available, it also serves for looking up restaurants, ordering food, and ordering groceries.

It also includes a list to store all the ingredients that are required which will be further sent for grocery shopping. It will also suggest healthy and hygienic cooking and eating habits for the users' benefit.

The proposed system will be applied for use by any persons facing culinary difficulties due to time constrained lifestyle. It will also bring about a great deal of convenience for the same.

Related Works

A recipe recommendation system has been developed which uses recipe content along with the user's preference [1]. Recipe from AJINOMOTO is estimated based on the scoring recipe (TRS) and the system outputs recommended recipe. The system deals with user preferences and the behavioral patterns. The system does not have quantitative evidence.

A method of extraction and visualization of the user's recipe preferences using KeyGraph is developed [2]. To evaluate validity of KeyGraph, two analyses are conducted, one to find periodic fluctuation by using Google Trend data, and other is POS analysis. Better results are shown with the preference extraction. Only few positive revives can be analyzed. Not really supportive in the real markets.

An application which is able to retrieve, within the cards of the ECHI-ICH inventory then identify the main ingredients and extract the nutrients has been developed [3]. The users on the web can also query, in a simple and straightforward manner with the ICH inventory and obtain the results. The system retrieves the cards about foods, integrating a combination of methods, and offers the results divided into categories. The recipes have no quantities, but rather "how to", so the user can only have broad ideas about nutrients.

A method for generation and recommendation of new recipes based on information available about the existing ones is developed [4]. The study uses an ontology based diet management system. The system recommends recipes taking into account the user's health-preferences. The system is not providing accurate ODM

Another method proposes an ingredient suggestion method for allergy safe cooking based on food context as extracted by examining recipe similarity [5]. Uses both recipe metadata and context metadata for calculating the recipe similarity (use-context) and Use only recipe metadata for calculating the recipe similarity (no-context). Considering the use of food context for allergy ingredients, the precision of extracting allergy-free ingredients is proved. Method manually extract the food context of each recipe hence it takes too much time and effort. Also does not provide a larger recipe data set.

Another system proposes a clustering method for recipe and user-generated sites based on page structure, main ingredient and the seasonings used in the food [6]. Two types of clustering are done, first based on the name and cooking method of the dish, and second based on ingredients and seasoning of the recipe. The proposed system benefits the classification of similar recipes in a better way than other applications. The system does not support the cook flow and value of ingredients.

FoodForCare is intends to provide support to users, especially patients with non-communicable diseases (NCDs), to record their eating habits [7]. The application records user's intake of food, along with body metrics based on calculation of user's BMR and notifies the user of his health metrics, calories consumption per day. Information about daily nutrition and calorie intake enables users to regulate their food intake and improve their health. The system does not have the list of restaurants that sell healthy food and food menu for patients.

A method proposes a framework for the user to find a recipe the user needs [8]. The framework provides an analysis of the combination of ingredients in order to assist the arrangement of ingredients for the user. The framework allows the user to efficiently arrange the list of ingredients taking his/her intention into consideration. The proposed framework does not distinguish seasonings from ingredients which will reduce performance of the arrangement assistance.

A system that helps users to build a diet management plan that can provide them recipes with accurate nutritional information based on recipe information [9]. The Recipe ontology (RO) is built based on SPARQL Protocol and RDF Query Language (SPARQL); and a dietary recommendation system (DRS) and a JENA Framework is designed to analyze whether a user is having a balanced diet and to recommend suitable changes in diet. It analyses the health information and provides users with personalized dietary recommendations, and maintains a long-term personal diet record. A large recipe collection database is required in for the system to function accurately.

The authors perform a case study on the mobile application Foodie.fi to understand the grocery shopping habits of users [10]. They categorize users into three major classes based on their shopping behavior; Holistic Meal Planners, Memory Extenders, and Online Shoppers. Table 1 shows the closely related work with our application.

System Architecture

The main components of the system include a database, a web application, and a mobile application. Fig 1. gives an overview of the main components of the system.

The main modules of the system are users, recipes, restaurants, groceries, grocery stores, and orders. The helping important modules include badges and other miscellaneous modules. Fig 2. Illustrates all the modules implemented by our system.

The web application is accessed through the browser. The android mobile application works on all android versions above 4.3 (lollipop).

We have developed a unified API system for both the web application as well as the Android app. The API system receives request from the client app and returns data in JSON (JavaScript Object Notation) format. The JSON data that is returned from the API

Fig 1. Main Components of the System

system in form of a response is rendered in the client side along with the UI. Since our APIs follow the REST API principles, other applications can be built using our API.

We are using NodeJS in the server-side for handling request response lifecycle. NodeJs is a well-known server side scripting language which runs on Google's chromium V8 engine. NodeJS is good at handling asynchronous operations. For the storage, we are using MongoDB, which is a document based NoSQL database system. Its schema-less model makes it easier to store unstructured data. MongoDB returns JSON objects and uses JavaScript for making queries. The JSON is serialized and saved internally as BSON (Binary JSON) format.

Fig 2. Main Modules of the System

Table 1 : Comparison of closely related works with our application

Concept	Advantages	Disadvantages
Uses Content Based Filtering Algorithm (CBFA) to recommend recipes to users [11]	Eliminates cold-start problem Dependent only on users for preference identification	Increased dependency on user interaction to increase accuracy
Analyses user comments in order to understand the response to a recipe [12]	Classification of comments into negative or positive or neutral Polarity of words related to food domain used for classification Helps the user to make a smart decision to choose the preferred recipe	Offers only 90% accuracy
Identifies the cultural preferences of users based on the recipes they have viewed [13]	Users are recommended recipes based on their cultural preferences	Huge amount of recipe data is needed for analysis
Recommends alternative ingredients to users to replace ingredients that may not be available [14]	Alternative ingredient recommended Users preference is given more importance Tries to keep the authenticity of the recipe and taste Finds similar ingredients	Might not work with all ingredients because some ingredients maybe irreplaceable
Analyses user's circles to identify recipes that might be preferable to them [15]	Provides personalized recommendation engine using UCM(user-class matrix) Has scalable computational capabilities to process massive recipe information	Suffers from cold-start.

Users module handles all the authentication part and user details. Authentication is done using JWT (JSON Web Token), which basically encrypts and signs the user data with the secret key and makes token out of it. This authentication token is stored in the client side as browser cookie and x-access-token. Since the JWT is passed along with each request, server will decrypt the JWT and verify the credentials.

User can do CRUD (Create Read Update Delete) operations in all the modules. Groceries module takes care of the grocery data that is used in recipes and stores module. Recipes module is our main module that integrates all other modules. The recipe filtering, recipe recommendation, nutrition and other details of the recipe is handled by this module.

Restaurants module keeps track of restaurants data and the recipes served by them. Grocery Stores module lets the user to search and order groceries online and compare the prices offered by the individual stores. Orders module handles the online payment and cart checkout. Badges module is a reward system that encourages users to use the app more often by giving them knock-off deals.

Implementation

The entire system is divided into two:

Back End

This is where the actual data processing and storage takes place. The backend API is built using the REST API standards and returns JSON formatted response. We are using the following technologies to make the backend of our system.

NodeJS: For handling the request response cycle of the API.

MongoDB: For querying and handling huge amount of recipes, restaurants, stores and groceries data.

JWT: For handling user authentication.

Front End

We are developing both a web application and an android application.

Web application is built with HTML5, CSS3, JavaScript and it also makes use of popular frameworks like Semantic-UI, Jquery etc.

Android Application is done using pure Java and XML. It uses Retrofit library to communicate with the backend API.

Both the Web and Android application are using the same backend API to fetch and alter data.

Data Collection

Our proposed system requires data of recipes, restaurants, groceries and stores. All the data can be entered manually by the user. User can also enter his restaurant into the application and specify the dishes that are prepared in his restaurant.

The store owners can also specify the groceries that they sell through their user accounts. There is also data collection done from user's inputs to the system.

Experimental Setup

Software part of the experimental setup is as follows;

Server Side Consists of NodeJs, and MongoDB.

Client Side requires Android 4.0 or above.

Hardware portion should have the following;

Server Side of 1.5 GHZ processor or above with 512 MB RAM or above, and *Client Side* can have 1 GHZ processor or above, and 1 GB RAM or above.

Software Required for Development is Android Studio, Sublime Text, and Git.

Results

A survey was conducted by downloading and using similar and existing system product applications on platforms similar to the proposed system for the sake of convenience and ease.

An application called 'Kitchen Stories' was downloaded from the play store which provides similar features as the proposed system. Later, on using the application it was found that it provides only recipes, whereas the restaurant and grocery features are not provided in the application 'Kitchen Stories'.

	Existing System	Proposed System
Recipes	Yes	Yes
Restaurant	No	Yes
Groceries	No	Yes

Table 1. Comparing proposed system with Kitchen Stories.

The same as above was done for an application called "Amazon Now". Later, on using the application it was found that it provides only groceries, whereas the restaurant and recipe features are not provided in the application 'Amazon Now'.

	Existing System	Proposed System
Recipes	No	Yes
Restaurant	No	Yes
Groceries	Yes	Yes

Table 2. Comparing proposed system with Amazon Now.

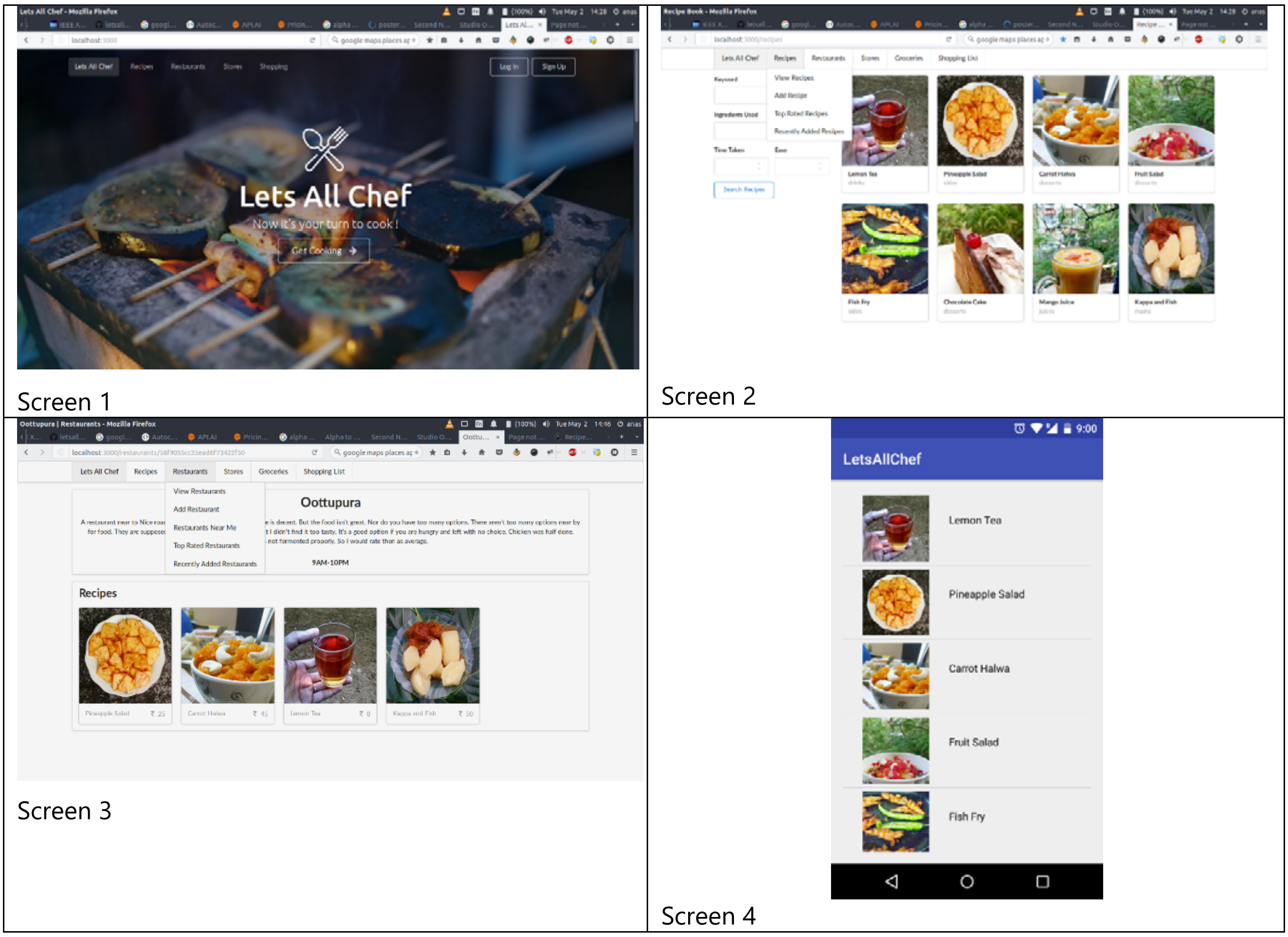

Fig 3. Proposed System Web pages

Fig 3. shows the webpages of proposed system. Screen 1 is the home page of the system. Screen 2 is the recipes option in that user can add new recipe, view top rated recipe and recently added recipe. Screen 3 is the restaurants option in that user can add new restaurants, view nearby restaurant, top rated restaurants and recently added restaurants. Screen 4 is the mobile app screenshot to view recipe list and Fig. 4 is the recipe details webpages. If user clicks on this recipe then ingredients, comment, instructions, nutrition details, combination recipe and related recipe.

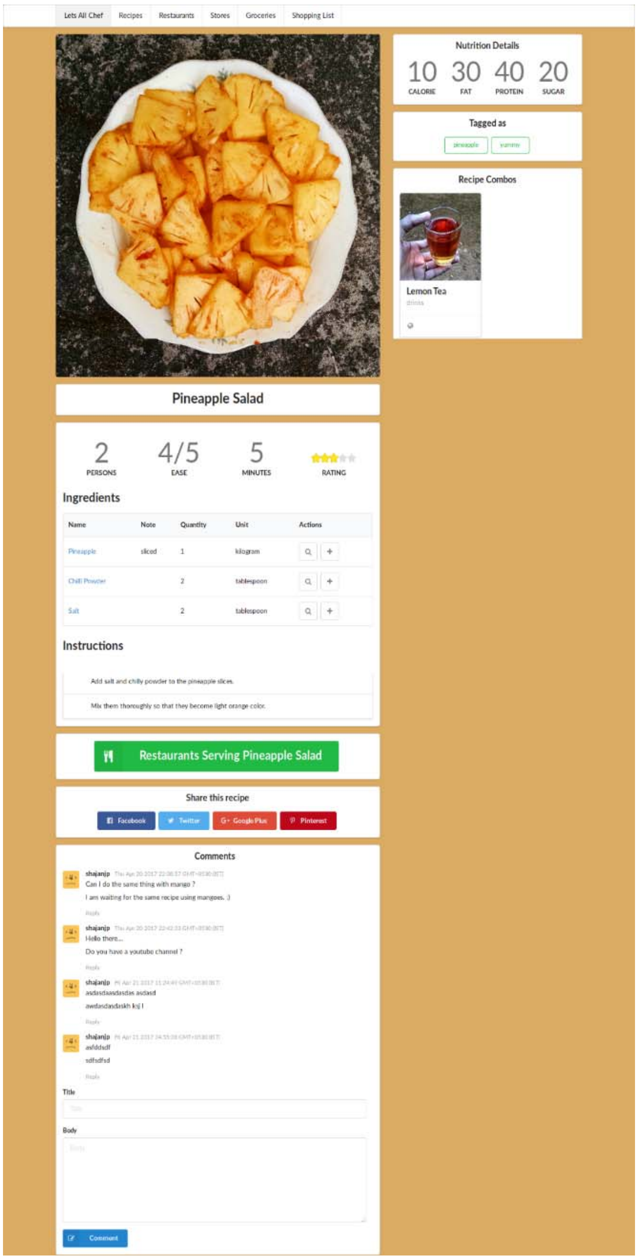

Figure 4. Recipe Details

Conclusions

The proposed system provides better user experience for culinary related problems for users living a time constrained lifestyle. The user enters the available groceries or the available utensils or the available time as input to the application into the filters.

The application will compute the best recipes satisfying the given inputs and show a list of recipes. If the user finds that he does not have some of the groceries in the recipe he shall put it up into the grocery list and that will become the shopping cart for another module of the application that will allow users to buy groceries online.

The user can also look up which restaurants these recipes and the user can also order them from those restaurants. The application also goes the further mile to suggest the user with healthy diet tips and healthy cooking habits and also recommend to them similar recipes.

References

[1] Asami Yajima, Ichiro Kobayashi, "Easy Cooking Recipe Recommendation Considering User's Conditions", International Conference on Web Intelligence and Intelligent Agent Technology Workshops, pp.13-16,2009.

[2] Mamoru Emoto, "Extraction of Preference of Recipe Providers and Users on Recipe-Sharing Websites", 15th International Conference on Data Mining Workshops, pp. 694-697, 2015.

[3] Isabella Gagliardi, Maria Teresa Artese, "Create Your Menu Discovering Traditional Recipe", Expo2015 (2015 Milan Universal Exposition).

[4] Yu-Wen Lo, Qiangfu Zhao, Yu-Hsien Ting, and Rung-Ching Chen, "Automatic Generation and Recommendation of Recipes Based on Outlier Analysis", 7th International Conference on Awareness Science and Technology, 2015.

[5] Ami Ooi, Toshiya Iiba, and Kosuke Takano, "Ingredient Recommendation for Allergy-Safe Cooking based on Food Context", IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM), pp. 444-449,2015.

[6] Akiyo Nadamoto, Shunsuke Hanai, Hidetsugu Nanba, "Clustering for Similar Recipes in User-generated Recipe Sites based in Main Ingredients

and Main Seasoning", 19th International Conference on Network-Based Information , pp. 336-341,2016.

[7] Natnicha Suthumchai, Sirin Thongsukh, Pacharamai Yusuksataporn, Songsri Tangsripairoj, "FoodForCare: An Android Application for Self-Care with Healthy Food", 5th ICT International Student Project Conference (ICT-ISPC), 2016.

[8] Satoshi Yokoi, Keisuke Doman, Takatsugu Hirayama, Ichiro Ide, Daisuke Deguchi, Hiroshi Murase, "Typicality Analysis of the Combination of Ingredients in a Cooking Recipe For Assisting the Arrangements of Ingredients", IEEE International Conference on Multimedia & Expo Workshops (ICMEW),pp. 1-6,2016.

[9] Yu-Hsien Ting, Qiangfu Zhao, Rung-Ching Chen, "Dietary Recommendation Based on Recipe Ontolog", 6th International Conference on Awareness Science and Technology (iCAST), pp.1-6, 2016.

[10] Pia Tukkinen; Janne Lindqvist, "Understanding Motivations for Using Grocery Shopping Applications ", IEEE Pervasive Computing, Vol.14, Issue 4, pp.38-44, 2016.

[11] Teh Lee Cheng, Umi Kalsom Yusof, Mohd Nor Akmal Khalid, "Content-based Filtering Algorithm for Mobile Recipe Application", 8th Malaysian Software Engineering Conference, pp. 183-188, March 2014.

[12] Pakawan Pugsee and Monsinee Niyomvanich, "Comment Analysis for Food Recipe Preferences", 12th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2015.

[13] Kyung-Joong Kim, Chang-Ho Chung, "Tell Me What You Eat, and I will Tell You Where You Come From: A Data Science Approach for Global Recipe Data on the Web", IEEE Early Access Journal, Vol. PP, Issue: 99, pp 1-1 2016.

[14] Naoki Shino, Ryosuke Yamanishi Junichi Fukumoto, "Recommendation System for Alternative-ingredients Based on Co-occurrence Relation on Recipe Database and the Ingredient Category", 5th IIAI International Congress on Advanced Applied Informatics, pp. 173-178, 2016.

[15] ZhengXian Li, Jiazhao Shen, Yong Xu, "A Scalable Recipe Recommendation System for Mobile Application", 3rd International Conference on Information Science and Control Engineering, pp. 91-94, 2016.

Author Details

1. **Shajan Jacob**, UG Student (CSE), TJIT, Bengaluru, Karnataka, India. shajanjp@gmail.com
2. **Qubin Jose**, UG Student (CSE), TJIT, Bengaluru, Karnataka, India. qubinj@gmail.com
3. **Sadique Ali Khan**, UG Student (CSE), TJIT, Bengaluru, Karnataka, India. sadiquealikhan396@gmail.com
4. **Jabez V.**, UG Student (CSE), TJIT, Bengaluru, Karnataka, India. jabezv10@gmail.com
5. **Sejal Nimbhorkar,** Associate Professor (CSE), TJIT, Bengaluru, Karnataka, India. sejals@tjohngroup.com
6. **Maria John,** Assistant Professor (CSE), TJIT, Bengaluru, Karnataka, India. mariajohn@tjohngroup.com