iJRASET

International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL
FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

www.ijraset.com

Call: ○08813907089 | E-mail ID: ijraset@gmail.com

FOODIE: An Android Application

Rahul Dandekar [1], Sanket Gadkari [2], Mayuri Sadudia [3], Sayali Kamble [4]

[1,2,3,4]*Under the Guidance of Prof. Shrikant Sanas, Department of Information Technology*
Padmabhushan Vasantdada Patil Pratishthan's College of Engineering, Mumbai

Abstract – In today's hectic world of work loaded life, there is no time for an individual to give a dedicated schedule for making his daily meals. You just want to relax after returning from work, either by ordering in a takeaway or heading out to a delicious establishment near you. At times it becomes a very tedious work for humans to think on, for what to make at dinner in such limited materials available. The food industry is an underserved market. The possibilities are wide-open, and the competition is scarce. Entrepreneurs can easily make a quick turn around by providing a much needed service. We are hence, developing an Android Application which would make the task of preparing meals a less tensed for an individual. The system is an innovative new recipe search engine which finds recipes you can make with the ingredients you currently have at hand. The system is an Android application - Foodie, which will display the recipes according to the inputs given by text, voice and image and filter them accordingly to their variety and list as per the most appropriate recipes first. Foodie answers the question, "What can I make with what I've got?" helping you get the most out of your groceries. Just select your ingredients to instantly find recipes you can make right now at home. Save on your grocery bills and help reduce food waste by using Foodie to easily find new and exciting ways to use the food in your home before it ends up in the trash. The project is mainly designed to fulfil the meal plans of an individual for proper and timely schedule of his taste and meals, along with taking care of trash and grocery bills.
Keywords — Recipe, Food, Cook Book, Recipe Retrieval, Android, Recipe Generator, The Meal Maker.

I.   INTRODUCTION

In today's world a person can prepare a dish either by asking another person who has acquired such knowledge of cooking or by reading certain cooking recipe books, and gaining self knowledge by owning such books. Another way of making delicious recipes is by searching for the recipes over the internet, read the instructions and prepare it. In most of the cases every person visiting a website, already know a name, by which he searches the website for the dish. We have come up with this unique idea of preparing meals just by providing ingredients to the system. Just add your ingredients and Foodie instantly finds matching recipes to give the required meals you want in a jiffy. We're all about good recipes, and about quality home cooking that everyone can enjoy. If you have a little gourmand at home, let him try his hand at making dinner with these simple methods to prepare a meal. It is like having a TV cooking show 'cheftestant' with you in the kitchen who always knows what to make, no matter what random ingredients are in your 'mystery basket'. The study therefore aimed at developing an application to enable user's search for whatever recipes they are interested in .The feature of this application is that recipes would be displayed to the user based on the ingredients provided in form of text, voice or image. With such a system in place, Foodie would be more competitive and adaptive to all types of users. The goal of this project is to create an Android software application that is accessible to everyone. The app will be easy to use with a variety of features available. These features will include a hierarchy of how different foods are related to one another and a recipe generator.

II.   LITERATURE SURVEY

In survey we have observed following issues with the current available applications for Recipe providers:
Applications like Cookpad, CookBook, etc. are recipe provider apps which provide the recipes' directions, ingredients, and reviews. It does not accept ingredients as input in form of voice and image. Most of the available means of Recipe apps provide less Indian recipes, while our app is mainly dedicated for Indian users by providing more than ever Indian recipes. The available applications do not provide active user interface with the app.
In the conducted survey we learned that, to make the app more user-friendly and efficient, the inputs to the system have to be provided in three ways - text, voice and image. Also it will show different types of dishes and their preparation time.

A. Tools And Technology
So we chose to develop this android application which is very much cost effective and dynamic, affordable by every end user who wants to schedule his daily meals with whatever ingredients he has at hand. The system would be two way integration with the user

550

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

and the database in the system. The platform used for the development of the system application is Android Studio, which is open source and contains both XML and Java for code compilation. The Eclipse version of Android development is obsolete, hence, Android Studio is chosen over it. The minimum version of Android that our application would be compatible with is Jelly Bean version of Android i.e. 4.1.2. The database connectivity would be done through SQLite.

Why Android?
Since our basic intention is to provide an application to the user, which can be accessed by him in all possible means, Android phones are being used by almost every individual. Hence, the selection of making the project in Android is the need of the hour.

Android supports gradle (this allows you to really have control over the build, create different application flavors, different signing configurations and so on).

In the layout view you have the option to view both the actual layout and the xml at the same time, while in Eclipse you must choose between the two tabs.

It feels better. From doing a refactoring (like renaming a method, to extracting a class) to viewing the Log cat. The refactoring seem more solid and the Log cat does not "disappear" as it happens in Eclipse.

B. How Is Foodie Different From Regular Recipe Apps?
The Foodie returns recipes you can actually make right now with the ingredients you have. It even tells you exactly how many recipes you can make, broken down into 'Starters', 'Main Dishes' and 'Dessert' categories. Foodie allows you to enter in the contents of your kitchen. It assumes you have basics like salt and pepper and then cranks out a list of potential recipes you can make using just what you have or with the addition of another item. There are recipes for starters, main-course, and desserts. You can have Foodie focus on a particular item in your list, so if you're hungry for a chicken dish, you can click on chicken on the list and Foodie will show recipes featuring chicken first. If you've got 80% of what you need for a meal, your list taps into what you need to cook and save what you have. Foodie is a free service. Signing up for a free account however does allow you to save your list for future recipe search.

C. Software Context
The intended use for this application is for day-to-day purposes. The application must meet the needs of the average android mobile user. Foodie will appeal to a very wide audience since both job workers, proficient cooks, and those who only wish to cook more but lack the knowledge of recipes can benefit from this application.

D. Usage Scenario
Foodie will be designed as a single-user application that can be helpful to anyone in the position of making food preparation decisions based on known ingredients of food. The system will primarily support these decisions, guided by interactions with the user to gather enough data to provide appropriate Recipes.

E. Android: An MVC Architecture
Android is a framework that provides java programmers the ability to control different aspects of smart devices. This interaction happens through the Android SDK (Software Development Kit). Models represent the raw information or access points to the raw information. Views represent the visual representation of the models. More generally they represent the layout of the information. Model can be a collection of menu items and prices (Strings and doubles). The view may be a scrollable list that has the menu item names on the left of each cell (list item) and the details on the right. Controllers connect the Views to the Models, they contain all the logic. Every event that is intercepted, such as a touch, is sent to its respective controller. The controller then decides, based on the input, which models to query. The information collected is then sent to the view which renders the visual representation of the models and the associated actions. The users' actions are completed by the view being rendered.
Android attempts to use this by providing the user with resources and layouts (res folder) that are either raw data files, such as images, or layouts (in XML format). These are the views, they dictate exactly what the program will look like, how everything is laid out. Android follows the "Main idea" from above by doing absolutely no work in the views (i.e. no loops, no procedural

551

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

programming). The controllers are called activities, they interact with the views to create a user experience. The SDK triggers events in the activities. The activity collects the information from the models, builds the view and sets up the event triggers needed to accomplish the tasks. Models are left to the user's discretion, usually they are implemented as classes, it is recommended to build as many classes that have no ties to the activities as possible to provide modularity.

Suppose the user taps on one of the menu items, the device collects the information and sends it to the controller, the controller now retrieves the Model for the information for that menu item. In this case the information contains the ingredients, the wine pairing and an image of the item (these may be several Models that have been queried to collect these results). The controller then packs the information up nicely for the view that will show this information to the user in an organized fashion
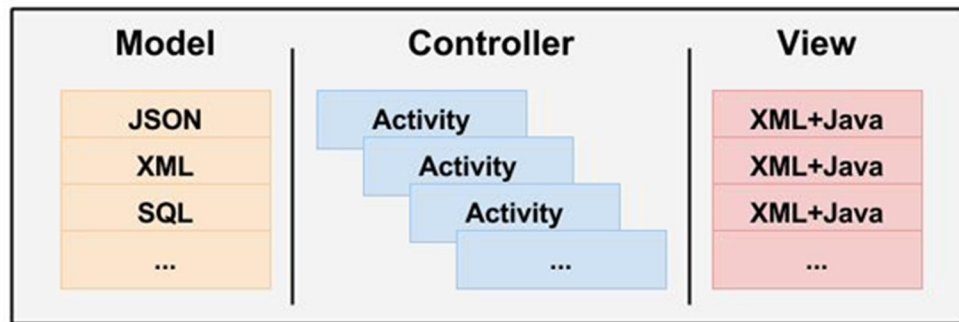

Fig: overview of MVC in Android

III.   THE PROPOSED SYSTEM

Foodie will require minimal user input, yet still provide substantial information regarding the desired output. A database will be used to keep track of all the foods owned. From this database, ingredients will be searched when the user decides what kind of dish he wishes to create. The app will feature direct integration with the user's input ingredients, so it will know what ingredients you have so you can cook great meals. The system will show all kinds of food vegetarian as well as non-vegetarian cuisines. It will provide different types of dishes ranging from Indian, Chinese, Italian, Mexican, etc. depending on type of ingredients provided. Recipe information will include details about a recipe, instructions for preparing a dish, ingredients, preparation time along with pictures as guides to great food.

The App begins with a homepage where you need to login. If you've not registered in the app, then you will have to do the registration process first. To make the registration process less time consuming, Facebook and Google account integration is provided, which lets you access the app easily. Once you login you can avail features of the App.

The system then lets you either to find a recipe by giving its name or by providing input of ingredients into the search field to make a custom search. If text is chosen as input, you will have to enter as much as ingredients you want in the search box query. It cranks out a list of potential recipes you can make using just what you have or with the addition of another item. Similarly if voice is chosen as input user can easily search for recipes just by saying ingredients by voice and discover great dishes. Lastly when image is chosen user can click pictures of ingredients and post it in search box. Accordingly results will be provided. The App also allows saving your favourite recipes for further referral.

IV.   IMPLEMENTATION

A. Data Model And Description

The primary data object for this system is an ingredient, each belonging to a food group. The highest level data objects are a recipe, an ingredient, and a cuisine.

   1)   Data Objects:
      a)   Ingredient
         o   Name (String)
         o   Quantity (Double)
         o   itemID (Integer)
         o   Food Group (String)
      b)   Food Group

www.ijraset.com   *Volume 4 Issue III, March 2016*
IC Value: 13.98   *ISSN: 2321-9653*
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

- o   ingredientID(Integer)
- o   Name (String)

c)   Recipe
- o   ingredientID(Integer)
- o   Quantity(Double)

d)   Image
- o   Object

e)   Favourites
- o   recipeID(integer)
- o   name(string)

B.   Functional Model And Description

1) Adding Ingredients:

Description: Allows users to add ingredients through various ways. Image Scan and Manual input will be the sub functions of the Add Ingredients option, which allows the user to manually input the ingredients.

Outputs: Sub categories

Destination of output: User

Processing: Once the app has been downloaded, the user clicks the Add Ingredients button. The user then selects how to input ingredients. After input, Foodie will save ingredients in the database for further use.

Pre-condition: The user must have registered and downloaded the app through the OS store.

Post condition: Display sub categories

Side effects: Doesn't display sub categories

Responses to Abnormal behaviour: Restart app

Stability: Stable

Necessity: Mandatory

2) Find/Search Recipes:

Description: Allows user to find recipes through three sub functions, text, voice and image.

Inputs: There will be a Find Recipe button which will be selected by the user. Under any of the search methods, there will be a bookmark button that if clicked will store the recipe in the Manage Favourites button.

Source of input: User

Outputs: Sub categories, store favourite recipes.

Destination of output: User

Processing: The user clicks on find recipes. Then Foodie will display sub categories for further use.

Pre-condition: The user must have registered and downloaded the app through the OS store.

Post condition: Displays sub categories

Side effects: Doesn't display sub categories

Responses to Abnormal behaviour: Restart app

Stability: Stable

Necessity: Mandatory

3) Managing Favourites:

Description: Allows the user to edit favourite recipes through sub categories of the app.

Inputs: N/A

Source of input: N/A

Outputs: Display the sub categories View Favourites and Remove Favourites

Destination of output: User

Processing: User can view or remove Favourites.

Pre-condition: The user must have registered and downloaded the app through the OS store.

Post condition: Must display corresponding sub categories.

Side effects: Doesn't display sub categories

Responses to Abnormal behaviour: Restart app

Stability: Stable

Necessity: Mandatory

C. Behavioural Model And Description

1) Events: The following is a list of events within the RM system that will result in a change of state:

Add Ingredients: From the home screen (Start state) of the application, the user chooses to provide ingredients. The state changes to Add Type state.

Find Recipes: From the Start state, the user chooses to find recipes on the basis of ingredients provided. The state changes to Find Type state.

Favourite Recipe: From the Start state, the user chooses to view saved recipes. The state changes to Favourite state.

View Recipes: Within the Favourite state, the user can select the saved recipe they wish to view.

Voice Input: while in the Add Type state, the user can provide the ingredients by voice to the system.

Image Scan: While in the Add Type state, the user can scan ingredients to add. The state changes to the Verify Item state.

2) States: Unique states are presented below as a consequence of the events:

Start: The application starts in this state from launch. The user will be given the option to add ingredients, find recipes, and manage their meal preparation.

Add Type: When the user selects add ingredients, options will be given between manually adding the ingredients and using the image or voice to provide an input.

Find Type: Once the user chooses to manage the recipe, options will become available to do either text, voice or image search.

Verify Item: After the image scans the ingredients, the user will be presented with a list of items that were verified through the scan.

Favourite: Once the user selects Favourite Recipes, options will be given to either view or remove recipes from the user's Favourites.

D. User Interface Design
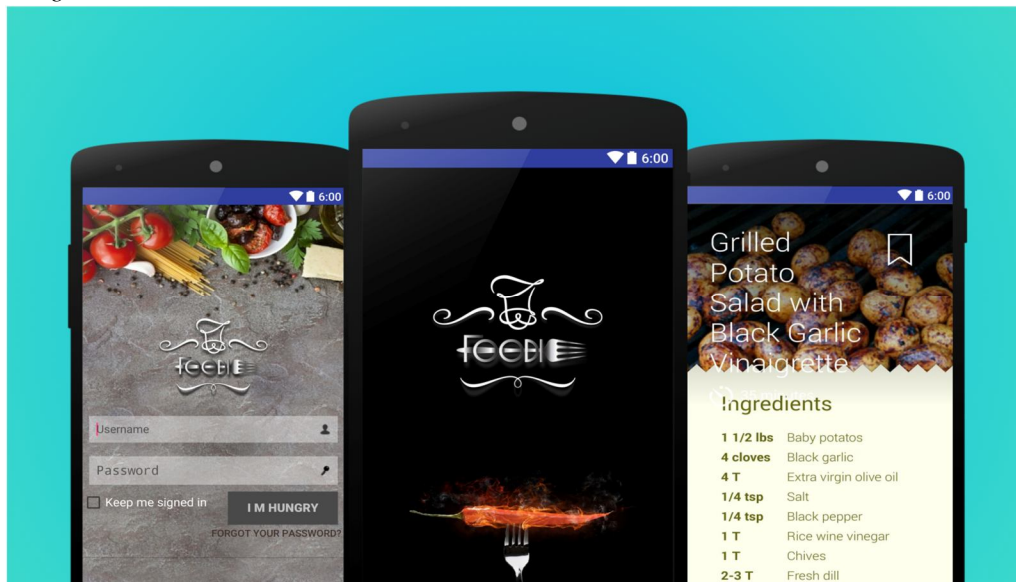
E. Database Design

1) Why SQLite?: The android platform uses SQLite databases in its applications and is one of five data storage options in android development. We will only be focusing on SQLite development in android because it is key to the construction of a

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

workable/functional program. **Content Values** allow us the ability to store a set of values for insert statements; **Context** allows us access to the application environment. **Cursor** is probably the most vital import we will need next to the SQLite imports. Cursor allows us access to the data returned to the cursor from a database query. **SQLException** allows us to throw SQL exceptions if there is ever an error, these messages provide more insight as to what the problem may be. **SQLiteDatabase** gives us the ability to manage a SQLite database using methods. **SQLiteOpenHelper** is basically a helper class that allows for creation and version management of a database. **Log** will basically log output in case there is an error. When you're using synchronized and decentralized storage it is important to access, transmit, and store data securely. To cover this completely, you need to address authentication, data at rest, data in motion, and read/write access.

V. RESTRICTIONS, LIMITATIONS AND CONSTRAINTS

The restrictions, limitations, and constraints for the Foodie will mostly be limited to the Android platform. The following list will describe these issues.

There are many Android Operating Systems, and making Foodie compatible with them all is impractical and unnecessary. Consequently, the Foodie will only work with the most recent OS series i.e. from Jellybean to Marshmallow, and provide updates for subsequent releases.

Camera quality may affect the scanner usage, limiting the user to manual input.

Only the most important functionality of the system will need to be tested. This includes image, text, and voice searching for recipes.

VI. CONCLUSION

The existing system shows us how the other applications work based on search recipes. This is used in our proposed system and we have developed an Android Application that will accept inputs in form of text, audio and images and will filter recipes accordingly and would provide results. The platform used is Android which is an open-source technology that can help in better availability of the environment for the user.

REFERENCES

[1] B.S. Afriyie, Concise ICT Fundamentals Volume One, Trafford Publishing, 2012
[2] Zigurd Mennieks, Laird Dornin, G. Blake Meike, & Mausmi Nakamura, "Programming Android Java Programming for the New Generation of Mobile Devices", 26 december 2013
[3] Bill Philips, Chris Stewart, Bran Hardy, Kristin Marsicano, "Android programming: The Big Nerd Ranch guide" , 2nd edition,2013
[4] Reto Meier, "Professional Android 4 Application Development", 8 June 2012
[5] Zigurd Menieks, G.Blake Meike, Laird Dornin and ZanePan, "Enterprise Android: Programming Android Database Applications for the Enterprise", 26 December 2013.
[6] R. Malloy, Internet and Personal Computing Abstracts: IPCA, Volume 22 Issues, Information Today, Incorporated, 2001, 2001.
[7] Milton Crawford, "The Hungover Cookbook Hardcover" –,May 31, 2011
[8] Hugh Fearnley -Whittingstall, "River Cottage Veg: 200 Inspired Vegetable Recipes Hardcover" , May 14, 2013
[9] Chris NewMan , "SQLite", November 2014
[10] Lauren Darcey and Shane Conder, "Android Wireless Application Development Volume II: Advanced Topics", 3rd Edition, Sep 25, 2012
[11] Joe Farcett, Danny Ayers, Liam R.E Quin, "Beginning XML", 5th edition, July 2012
[12] J. Pan, S. Chen, and N. Nguyen, Intelligent Information and Database Systems: 4th Asian Conference, ACIIDS, Proceedings Part 2, Kaohsiung, Taiwan, 2012

INTERNATIONAL JOURNAL
FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 ◎ (24*7 Support on Whatsapp)