JS:99 Food Recommendation

A dissertation submitted in partial fulfilment of

the requirements for the degree of

BACHELOR OF *SCIENCE* in Computer Science

in

The Queen's University of Belfast

by

Aedan Vallely

April 2020

SCHOOL OF ELECTRONICS, ELECTRICAL ENGINEERING and COMPUTER SCIENCE

CSC3002 – COMPUTER SCIENCE PROJECT

Dissertation Cover Sheet

A signed and completed cover sheet must accompany the submission of the Software Engineering dissertation submitted for assessment.

Work submitted without a cover sheet will **NOT** be marked.

|  |  |  |  |
| --- | --- | --- | --- |
| Student Name: | Aedan Vallely | Student Number: | 40181558 |
| Project Title: | Food Recommendation | | |
| Supervisor: | Jerry Shen | | |

Declaration of Academic Integrity

Before submitting your dissertation please check that the submission:

1. Has a full bibliography attached laid out according to the guidelines specified in the Student Project Handbook
2. Contains full acknowledgement of all secondary sources used (paper-based and electronic)
3. Does not exceed the specified page limit
4. Is clearly presented and proof-read
5. Is submitted on, or before, the specified or agreed due date. Late submissions will only be accepted in exceptional circumstances or where a deferment has been granted in advance.

**By submitting your dissertation you declare that you have completed the tutorial on plagiarism at** [**http://www.qub.ac.uk/cite2write/introduction5.html**](http://www.qub.ac.uk/cite2write/introduction5.html) **and are aware that it is an academic offence to plagiarise. You declare that the submission is your own original work. No part of it has been submitted for any other assignment and you have acknowledged all written and electronic sources used.**

1. If selected as an exemplar, I agree to allow my dissertation to be used as a sample for future students. (Please delete this if you do not agree.)

|  |  |  |  |
| --- | --- | --- | --- |
| *Student’s signature* | Aedan Vallely | *Date of submission* | 30-04-19 |

Acknowledgements

I would like to thank:

* My Supervisor **Jerry Shen** for his guidance and support throughout the project.
* My other lectures at **Queen’s University Belfast** who have helped me gain the knowledge to complete the project.
* My **family and friends** who have supported me throughout this Final Year Dissertation

Abstract

The purpose of this project is to explore the possible success of creating a Multimodal Food Recommendation system. It focuses on recommending food to a user by using Cosine Similarity. The focus will be on displaying information about the food though the forms of images, text descriptions and nutritional information. The application produced positive results when multiple users tested the system.

This report will compliment the system and expand details on how the system was designed and created.

The project code repository can be found on the Queen’s University EEECS GitLab [1].

Table of Contents

[1.0 Introduction and Problem Area 8](#_Toc39156658)

[1.1 Properties of Food 8](#_Toc39156659)

[1.2 Types of Food Recommendation Systems 8](#_Toc39156660)

[User Preferences 8](#_Toc39156661)

[\*During this project, the words Recipe and Food will be used interchangeable and are taken to have the same meaning 8](#_Toc39156662)

[User Nutritional Needs 9](#_Toc39156663)

[User Preferences and Nutritional Needs 9](#_Toc39156664)

[Group Recommendations 9](#_Toc39156665)

[1.3 Previous Approaches 9](#_Toc39156666)

[2.0 Solution Description and System Requirements 11](#_Toc39156667)

[2.1 Solution Description 11](#_Toc39156668)

[2.2 System Requirements 11](#_Toc39156669)

[Functional Requirements 11](#_Toc39156670)

[Non-Functional Requirements 12](#_Toc39156671)

[3.0 Design 13](#_Toc39156672)

[3.1 Architectural Design 13](#_Toc39156673)

[3.2 Software Design 13](#_Toc39156674)

[3.3 User Interface Design 14](#_Toc39156675)

[Initial Design Ideas 14](#_Toc39156676)

[User Preferences Page 14](#_Toc39156677)

[Recommendations Page 15](#_Toc39156679)

[Recipe Information Page 16](#_Toc39156681)

[Type of User Interface 16](#_Toc39156682)

[3.4 Food Dataset Design 17](#_Toc39156683)

[Dataset Size 17](#_Toc39156684)

[Dataset Details 17](#_Toc39156685)

[3.5 User Database Design 17](#_Toc39156686)

[3.6 Recommendation Algorithm 18](#_Toc39156687)

[Cosine Similarity 18](#_Toc39156688)

[Requirements for Cosine Similarity 18](#_Toc39156689)

[4.0 Implementation 19](#_Toc39156690)

[4.1 Use of Supporting Tools 19](#_Toc39156691)

[Languages Used 19](#_Toc39156692)

[Development Environment 19](#_Toc39156693)

[Version Control 19](#_Toc39156694)

[Python Libraries Used 20](#_Toc39156695)

[4.2 Data Collection 20](#_Toc39156696)

[4.3 Creating the User Interface 20](#_Toc39156697)

[4.4 Creating the Recommendation System 21](#_Toc39156698)

[Creating the Vectors 21](#_Toc39156699)

[Cosine Similarity 21](#_Toc39156700)

[Filtering the Results 22](#_Toc39156701)

[4.5 Linking the Components 22](#_Toc39156702)

[4.6 Implementation Issues 22](#_Toc39156703)

[5.0 Testing 23](#_Toc39156704)

[5.1 Software Testing 23](#_Toc39156705)

[5.2 User Testing 23](#_Toc39156706)

[Facilitating User Tests 23](#_Toc39156707)

[Criteria for Success 23](#_Toc39156708)

[5.3 Test Results 24](#_Toc39156709)

[Time Results 24](#_Toc39156710)

[User Survey Results 24](#_Toc39156711)

[6.0 System Evaluation 25](#_Toc39156712)

[6.1 Interpretation of Test Results 25](#_Toc39156713)

[Background Information 25](#_Toc39156714)

[UI Feedback 25](#_Toc39156715)

[Recommendation System Feedback 26](#_Toc39156716)

[6.2 Evaluation of Project Goals 26](#_Toc39156717)

[Functional Requirements 26](#_Toc39156718)

[Non-Functional Requirements 27](#_Toc39156719)

[6.3 Evaluation of Project Process 28](#_Toc39156720)

[Project Management 28](#_Toc39156721)

[Future Improvements 28](#_Toc39156722)

[Overall Conclusion 29](#_Toc39156723)

[7.0 Appendices 30](#_Toc39156724)

[7.1 Bibliography 30](#_Toc39156725)

[7.2 How to Use the System 31](#_Toc39156726)

[Requirements 31](#_Toc39156727)

[Configuration 31](#_Toc39156728)

[User Interface Guide 32](#_Toc39156729)

[7.3 Test Results 34](#_Toc39156731)

1. Introduction and Problem Area

What to eat is one of the most important and fundamental choices one makes on a day to day basis. According to studies, food takes up to one third of a person’s daily thoughts [2]. Such an important part of life is one that is easily overlooked, and most people do not adhere to the recommended healthy lifestyle [2]. This is where a food recommendation system can be useful. Food recommendation systems recommend food to the user based on a whole range of different factors. These could range from taste, ingredients, ease of cooking or spice levels. There are multiple different ways to recommend food and there have been multiple different approaches to creating a food recommendation system in the past.

* 1. Properties of Food

It is important to understand how food is perceived by people in order to be able to recommend it to them. Food and meals have many different properties: ingredients, nutritional information, taste, smell, colours, textures etc. There are two main ways to describe these properties: a text description or an image. A text description is able to go into a lot of detail about the food, however a picture can tell a thousand words. People will be able to clearly see the different colours and textures in the food and will help them to determine if they would like it, potentially more so than the text descriptions.

* 1. Types of Food Recommendation Systems

There are 4 main groups that most recommendation systems fall into [3] and each one focuses on a different need of the user. There are benefits and costs to each one therefore it is important to understand the different types of system before deciding which one to create.

User Preferences

The first type of system is a system based solely on user preferences. This system will obtain information about the user through different methods about the food they like using meal history, user input etc. and use this to recommend food that the user will hopefully like. This method should return a high rate of successful recommendations. However, this method negates the issue that the nutritional needs of the user are not being accounted for.

\*During this project, the words Recipe and Food will be used interchangeable and are taken to have the same meaning

User Nutritional Needs

This system will attempt to recommend food based on a given set of needs for the user. These needs could be general, such as a daily calorie limit, or be more individual and fitted to the specific user. There will be no focus on the likes or dislikes on the user and will only recommend the best suited foods for the users needs. While this system will help the user adhere to a healthier diet, it may not be recommending food that the user likes. Previous attempts at recommendation systems [4] found that only accurate recommendations of food that the user will enjoy will ensure the user will continue to use the system. The benefits of this system will therefore be negated if the user will not use it.

User Preferences and Nutritional Needs

An obvious solution to the shortfalls of the above approaches would be to combine them. Basing a system on both the preferences and nutritional needs of the user would surely produce the best results in both user satisfaction and helping the user adhere to a healthy diet. This is clearly the optimum solution for creating a food recommendation system. However, this system does come with its disadvantages, such as the need for a large dataset to include healthy food that the user will like, as there may not be a large overlap between these groups. The additional computation time to find foods based on these factors could also cause a loss of interest from the user.

Group Recommendations

In many cultures, meals are social events. Whether it is dinner at home with family or a party with friends, some people rarely eat alone. This type of system aims to find food that would be suitable for a group to eat. This is particularly challenging as everyone in the group may have distinct likes and dislikes, making it difficult to find food that would be enjoyed by everyone.

* 1. Previous Approaches

There have been many different attempts at creating food recommendation systems each exploring different avenues that could produce a successful system. Some ideas can be used from previous approaches to aide the creation of a new system. The use of heuristics [5] was shown to give a high degree of precision. This allowed the user to input some information about what they wanted and get a more accurate recommendation. Another approach [6] used the ingredients liked by the user to recommend food. Using the ingredients the user liked to recommend food was able to produce an accuracy of 81%. This approach however required the collection of information about previous meals that the user enjoyed.   
An interesting study [7] found that it was possible to ‘nudge’ users towards selecting healthier choices based on placement of the recommended recipes. This discovery would make it easier to implement the 3rd type of recommendation system and nudge users towards healthier foods that they might like.

These previous approaches along with the different types of recommendation system will help with the decision of what avenue to approach with this food recommendation system. All advantages and disadvantages will be analysed to come to the final decision.

1. Solution Description and System Requirements

Over the course of this project, there will be a set of objectives and aims that will have to be decided on and agreed to. This will ensure that the project can be completed, and metrics are able to be evaluated.

* 1. Solution Description

This project will be to create a Multimodal Food Recommendation system that will focus on food images, food ingredients and the nutritional information of the food. This will use a User Interface to display this information to the user in the best way possible. The aim will be to recommend food to the user and display the properties of the recommended food in a meaningful way.

The general objectives will be:

* Study previous attempts and carry out a literature review in order to decide on the best possible approach to the problem.
* Learn about the different metrics that can be used to recommend food to users.
* Research different recommendation techniques and decide on the best suited approach.
* Explore the most effective User Interface designs and create a suitable user interface.
* Create a system to recommend food and display information about that food in the User Interface.
  1. System Requirements

To achieve the goals for this project, there must be a list of functional and non functional requirements to follow and meet. This is necessary to use as a guideline for implementing the system and to use as a benchmark to see how the project is progressing.

Functional Requirements

* The system should be able to read in information from the user without error.
* The system should use this information to recommend recipes to the user.
* There should be a suitable dataset of recipes for the system to use.
* There should be a suitable number of recipes recommended to the user on each occasion.
* The system should be able to store the user’s preferences for use in future recommendations.
* The user interface should provide multimodal description about each recipe, such as images and text descriptions.
* The user interface should provide the nutritional information for each recipe.
* The system should provide the recommendations in a reasonable time frame.
* The user interface and recommendation system should work together seamlessly and allow easy transfer of data between the two.

Non-Functional Requirements

* The system should be easy to access by the user.
* The user interface should be simplistic and clear to use.
* The user interface should provide the information that isn’t overwhelming and is easy for the user to understand.
* The code should be clearly laid out and easy to understand.
* The system should produce accurate recommendations for each user.

1. Design

The design of the system is very important to ensure that the requirements set out in the above section will be met. This section will explore the decisions that have been made and the reasons behind them.

* 1. Architectural Design

The system will contain a User Interface, a recommendation system, a dataset of food details and a database to store the user’s details.

Recommendations

Stored User Information

User Information

User Information

Food

Information

Recommendation System

User

Interface

Food

Dataset

Users

Database

*Figure 1. Architectural Design of the System*

Figure 1 shows the main components in the system link together, and how data will flow between the different components. The food dataset is stored separately from the user information. This is so that there can be more food data added to the dataset easily, making it easier to manage and maintain.

* 1. Software Design

The Recommendation System, the Users Database and the Food Dataset make up the backend part of the system. The core recommendation system will work by using data passed to it from the user interface, and data taken from the database to calculate suitable foods to recommend to the current user. The food dataset will store details about the food and the Users database will store the previous likes and dislikes from the user.

* 1. User Interface Design

The user interface is a very important part of this system as ease of use is a vital requirement of a recommendation system. The user interface must be clear and understandable, simple to use but also display all of the necessary information.

Initial Design Ideas

An initial design of the system included one page to input the required details and another page to display information about a recommended recipe. However, it was decided that this would not be providing enough information to the user, and only displaying one recipe would not be satisfactory. Displaying information about more than one recipe at a time was found to be too overwhelming and was too cluttered.

User Preferences Page

Drop Down Box

Asks the user to select the amount of time they have available to them. Options of 15 mins to 90+ mins.

Multi Select Box

Allows users to select one or more categories of food

Text Box

Allows users to enter list of ingredients they like

*Figure 2. Design of the User Preferences Page*

Figure 2 shows the design for the user preferences page. This page is essential to the design of the system. The user’s preferences are needed in order to recommend suitable recipes. The required values to gather are Time, Categories and Ingredients. The user can select the length of time they have available, one or more categories of food they would like and a list of ingredients that they would like. There is the choice of selecting any category. Select boxes and drop-down menus have been used to reduce the chance of error and to ensure only categories present in the dataset can be chosen.

Recommendations Page

Image of the Food

Button

Allows users to view more information about the food

*Figure 3. Design of the Recommendations Page*

Button

Allows users to “Disike” this recipe

Button

Allows users to “Like” this recipe

**Recipe Name**

**Recipe Name**

Figure 3 shows the design of the Recommendations page. This is the page that will display the different recipes that have been recommended to the user. It will only display an image of the recipe and its name. This was decided to ensure the user was not overwhelmed by the information displayed on this page. The focus of the project is a multimodal recommendation system; therefore, the focus of this page will be on image.

There will be five recommendations displayed per page. For each recipe, there will be three buttons: Like, Dislike and View Info.

The Like and Dislike buttons will allow the user to decide if that recipe is something they like or not. This recipe will then be added to the corresponding part of the database. The View Info button will allow the user to access the information page for that recipe.

Recipe Information Page

Recipe Name

Time to Prepare

Link To Recipe

Text Description

Nutritional Information for the Food

Images of the Food

Text information about the food

*Figure 4. Design of the Food Information Page*

Figure 4 shows the design of the Recipe Information page. This is the page where the user can get more details about each specific recipe. The page will contain images of the food, text descriptions as well as the nutritional information. This entails images, text and numerical information. There will also be a link to a step by step recipe the user can use to make the food themselves. The page will be laid on in a way that ensure all text is readable, and the nutritional information can be easily understood. It is important for the user to be able to take in all the information so they can decide if it is something they will like.

The text information will include the name, length of time required to make it and a brief description of the recipe. The images that are displayed will be randomised so that there are different images shown each time. This will mean that the user will be able to see multiple different images and learn more about that recipe.

Type of User Interface

There are many types of user interface that could be used for a recommendation system. A standard computer program, a mobile app or a web app were all considerations. Each option had strengths and weaknesses. A standard program would be easy to develop and have a fast run time, however it would only be accessible via computer. A mobile app can be accessed on the go, which would be useful when eating out or shopping for food. A mobile app however has considerably more development time. A web app was found to have the best balance between accessibility and ease of development. A web app provides an easily customisable environment for creating a UI and also opens the system up for hosting on a server in the future to accommodate multiple users.

* 1. Food Dataset Design

The dataset of Food is the basis on which the recommendations are made. Without a suitable dataset, the system would not work.

Dataset Size

The size of the dataset is an important consideration. A dataset too small will be unable to provide enough recommendations for different users and could become stale by constantly recommending the same recipes. A dataset too large would be able to recommend a variety of recipes to the user, however would take up a lot of space, increase run times and take a long time to collect.

A decision was made to collect data for 50 different recipes. This would ensure enough variety in recommendations as well as a quick run time.

Dataset Details

There are many different characteristics of food that could be collected. It was important to collect enough information for the user to understand the recipe. It was decided that the data should be split into different food categories. There would be 10 different food categories with 5 different recipes in each one. For each recipe, the following details will be collected:

* Name of the recipe
* Category it belongs to
* Time to prepare the recipe.
* List of ingredients if the recipe.
* Text description of the recipe,
* Images of the recipe.
* Nutritional Information: Calories, Fat, Saturated Fat, Carbohydrate, Sugar, Protein, Salt.
  1. User Database Design

This will be a simple database will just one table. The table will store details for each user: their likes and dislikes. There will be 5 columns in the table: UserID, Name, Liked Ingredients, Liked Recipes, Disliked Recipes.

As discussed in section 1.3, using ingredients that the user previously liked gave a high score of accuracy in a different system [6]. However, that system also found that using ingredients the user disliked had a very low accuracy score, therefore it was decided that storing ingredients the user dislikes would not be used in this project.

* 1. Recommendation Algorithm

There are many different methods that can be used to recommend things to users. After reading many different attempts at a recommendation system, it became clear that a heuristic approach that used ingredients liked by the user would be a good solution to the problem. This means that the recommendation algorithm would be a recommendation by similarity method.

Cosine Similarity

A method to carry out recommendation by similarity is Cosine Similarity. “Cosine similarity is a [measure of similarity](https://en.wikipedia.org/wiki/Measure_of_similarity) between two non-zero vectors of an [inner product space](https://en.wikipedia.org/wiki/Inner_product_space) that measures the [cosine](https://en.wikipedia.org/wiki/Cosine) of the angle between them” [8]. The problem contains two sets of information: the information about the food and the information from the user. By converting the food information and the user’s information into vectors, measuring the cosine of the angle between those vectors would give a score of similarity. The food that produce the highest similarity score would then be suitable foods to recommend to the user. Two identical vectors i.e. angled at 0° would have a score of 1, whereas two vectors that were completely different and angled at 90° would have a score of 0.

Cosine similarity is a very powerful but cost-effective method to generate a set of recommendations and fits the needs of this project very well.

Requirements for Cosine Similarity

Cosine similarity requires two vectors to compare. Designing a vector of labels for each food will therefore be necessary. This set of labels should include the category of food, as well as a list of ingredients in the food.

This set of labels will then be compared to the set of labels of the users likes. This will be a combination of the information inputted by the user, along with the labels gathered by the other foods they have liked. This will mean that the more recipes the user likes, the better the recommendations will become.

1. Implementation

This section will discuss how the project was implemented, what languages and tools were used and some of the decisions that were made throughout the development process.

* 1. Use of Supporting Tools

Languages Used

The first step to creating this project was making the UI and creating a web app. The web pages themselves would obviously use HTML however, a framework was required in order to facilitate routing and form submission etc. Node JS was chosen for this as it was a simple to use, powerful tool and I also had previous experience using node from my placement year. A prototype of the web app was then developed using Node JS.

The recommendation system using Cosine Similarity required a language that could facilitate data science and packages focused on data manipulation. The choice made for this was python. Python is an incredibly powerful language with multiple packages focused on data science.

A problem however arose when considering how to link the recommendation system to the front-end user interface, it became clear that it was not efficient to link python scripts to a node js server. A decision was then made to use a python-based framework for the web app. Flask was the framework that was chosen. Flask is a python web framework that allows easy links to python scripts.

Development Environment

The development environment chosen was PyCharm. PyCharm is a premium development environment created by JetBrains and can be used for free by students. It easily facilitated creating the web app and even generates a skeleton Flask application. PyCharm also creates a virtual environment for development. This is very important when using python. Python uses many different languages and has many different working versions. A virtual environment allows the use of a specific version of python independent of any other project being created on the device.

Version Control

A EEECS provided Gitlab was used for the version control of the system. Using a version control allows the project supervisor to access the code at any time. It also means that if anything goes wrong in development, there is an option to rollback to a previous version, due to the incremental nature of commits.

Python Libraries Used

Python is a language that has many different supporting libraries. These libraries make development must easier and faster, and can reduce complex steps into simple function calls. The main library that was used was Flask. Flask was decided as the framework for the web app. It has many features to allow routing and requests.   
A quality of life library used was pandas. Pandas is a library that focuses on data frames and data structures. It was very useful during the implementation to structure the food data and the users data.

To carry out the cosine similarity, a library called sklearn was used. Sklearn is a machine learning library that has many different uses. A cosine similarity function and a feature extraction function were the uses for the library in this project.

* 1. Data Collection

The recommendation system requires data in order for it to successfully recommend food to the users. It was established in the design section that there would be 10 categories of food with 50 total recipes. The 10 categories were gathered from the BBC Good Food website [9]. This is a very popular website which gave a good indication of popular recipes in the UK. The categories selected were: Pasta, Soup, Noodles, Pizza, Rice, Salad, Beef, Chicken, Seafood, Pork. The information that was set out in section 3.4 were collected. The information about each food was stored in an excel document. This allows for easy editing of information and the ability to change the details as required. It will also allow a host of the system to easily add in custom information.

When the app is running, the data is loaded into a pandas dataframe which allows it to easily be accessed and allow the correct information to be used.

100 images for each food were collected. This was achieved by downloading images from Google Images [10].

* 1. Creating the User Interface

In the design section, it was decided that a Flask powered Web Application would be used for the user interface. As PyCharm was used as the IDE, a shell for the web app was automatically generated. The structure is a central server application (app.py) and different templates that can be rendered to display different information. The central application controls the routing and displaying of templates. This is also how the user interface interacts with the recommendation system. This structure can be seen in figure 5.

Each template refers to a screen on the site. The rendering of these screens is controlled by the app.py screen. When the user clicks a button or submits a form, they are then routed to another page.

app.py

Users.html

home.html

recommendations.

html

home.html

Recommend.py

*Figure 5 Structure of the System*

The passing of information between screens is done using POST requests. This ensures that the data is accurately transferred. The data is passed to the app.js by a POST request and then given to the next screen when it is rendered.

* 1. Creating the Recommendation System

The recommendation system uses cosine similarity to determine similar foods to the information the user has provided. To implement this system, the sklearn library was used.

Creating the Vectors

Firstly, a vector of information had to be created. As there was a set of keywords about the foods gathered, it was easy to turn this into an array using simple python functions.

The user’s information is passed to the recommendation function from the user interface. This is the information that the user has liked. This too is converted to an array and all of the vectors are added to a data frame. The sklearn library has a function which creates a vector from a column of a data frame and this is used to create the vectors needed for cosine similarity.

Cosine Similarity

The sklearn library is used again to create a cosine similarity matrix. This matrix contains the scores of each food compared with each other. This is filtered down to only show the scores of the user’s information. This can be sorted into descending order, showing the best recommendations at the top.

Filtering the Results

This does not however give the results for the recommendations. The user will have picked a certain amount of time that they have available, therefore all of the foods that take longer than the specified time must be removed.

The top five results from the remaining foods are then passed to the user interface to be displayed.

* 1. Linking the Components

The initial design of the system using node js would have required complicated processes to run the python recommendation system. Using Flask made it easy to link the components together. As the components are inside the same directory and environment, the recommendation system can simply be imported into the app.py file and the functions inside can easily be called. This is much faster and more efficient that other methods that would have to have been used in a different implementation.

* 1. Implementation Issues

There were some problems during the implementation phase. As already discussed in the design section, the initial user interface was developed in node js, meaning that significant development time was lost. As the end of the project drew closer, it was necessary to remove certain things from the implementation of the project in order to have a suitable system created by the end. The decision was made to remove the user database from the system. This would mean the functionality of the system was based only on what the user had inputted to the application.

The addition of the user’s page and the implementation of a database would have added on more development time which was not available in order for the project to be completed on time. It is very unfortunate that this had to happen and will be discussed further in section 6.

1. Testing

Testing of the system is vital to ensure that the needs of the project have been met. Testing will also be able to measure just how well those requirements have been met. This section will explore the test plan for this project. Section 6.1 will then evaluate the results gathered from the testing.

* 1. Software Testing

The software in the system should be properly tested to ensure that it all works to the required standard. Unfortunately, due to time constraints full software testing could not be carried out on this system. This will be further discussed in the evaluation section.

The time for the recommended recipes to be produced was tested. To carry out these tests, the time before and after calling the recommendation function was recorded, and the time elapsed calculated. This was calculated with:

* One Category and One Ingredient
* One Category and Five Ingredients
* Two Categories and Five Ingredients
* Ten Categories and Five Ingredients

Each of these variations was tested 10 times in order to gain a suitable average.

* 1. User Testing

This system suggests food to the user based on the users likes. The success of the system is therefore subjective. Everyone’s tastes are different meaning that users will have to use the system to determine its success.

Facilitating User Tests

The initial plan for user tests was to ask many people to come and use the system as it was working. However, due to recent situations this had to be altered. The revised test plan was then to get users to use the system remotely. This would be facilitated via a screen sharing system such as Skype. The user would use the system, then be asked to complete a survey to give their thoughts on the system and evaluate it. This compromise meant that the number of users able to use the test would be much lower than originally planned.

Criteria for Success

The survey carried out by the user should obtain the users thoughts on how the system matched up to the requirements set out in section 2. Each user will give a score for each question on how well the system performed. A scoring system of 1 to 5 will be used, with 1 being “very poor” and 5 being “very well”.

The following questions will be asked:

* How easy was the system to use?
* How did you find the layout of the site?
* How clearly was information displayed on the site?
* How much did you like the foods recommended?
* How accurate where the foods that were recommended to you?
* Did seeing the nutritional information of the food influence their decision.

Other questions will need to be asked to gather information about the user. This information will be needed to evaluate the results and to determine their validity. The following information will need to be gathered:

* Age range
* How often they cook their own meals
* What they look for when deciding what to eat: Image, Nutritional information, time etc.

The user will also be prompted to give some additional feedback about the system which can also be used in the evaluation of the system. As the limit of questions on a survey was 10 questions, users were asked to provide verbal feedback and this will be discussed in the evaluation section.

* 1. Test Results

Time Results

|  |  |
| --- | --- |
| Variation | Average Time (seconds) |
| 1 Category 1 Ingredient | 0.02061 |
| 1 Category 5 Ingredients | 0.03795 |
| 2 Categories 5 Ingredients | 0.03471 |
| 10 Categories 5 Ingredients | 0.04041 |

Average Time taken: 0.03342

User Survey Results

In total, 10 people used and tested the system. The survey was completed using Survey Monkey [11]. The charts of results can be seen in the appendices section at the end of the document.

1. System Evaluation
   1. Interpretation of Test Results

This section will discuss the information received from the user tests.

Background Information

Several questions were asked to determine some information about the tester. This was important to ensure there was a wide range of backgrounds, meaning the test results were not skewed. This was the case for the age ranges. In terms of how often users cooked their own meals, everyone said that they cooked a meal at least once a week, with most users answering that they cooked more than that. It would have been interesting to see the thoughts of someone who did not regularly cook their own meals, as they may have a different idea of what they look for in food.

For the question about the most important factor for choosing what to eat, there was a spread of answers across all the possible responses. This is a very positive sign. This proves that a multimodal form of recommendation is the best way to recommend food to a wide range of people. By providing all of the information, each user will be able to identify the particular feature they look for and make an informed decision on what to eat.

UI Feedback

The consensus about the User Interface was positive. Most responses claimed that the User Interface was easy to use and clear to understand. There were some different levels to responses however they were all on the positive side of the answers. This suggests that there was some room for improvement on the User Interface.

Some of the verbal feedback suggested that there should be more text indicating how to use the system. During the design section it was decided to keep the text on screen to a minimum and that a clearer UI would be easier to use. This however appears to be a poor decision, and this should be improved in the future.

Another suggestion was that the application appeared bland. This again was a conscious decision in order to make the application clear and easy to use. The users did think that the information was very clearly displayed therefore this decision appears to be successful. The concern with this feedback is that the user may be unwilling to continue to use the system if they find it boring or hard to look at. Keeping users interested is a very important part of the system, and the UI appearing bland could be a turn off to the user.

Recommendation System Feedback

The feedback from the users about the recommendation was quite average. They seemed to find the system was ok at recommending suitable food items, but not great. This is a positive result as it meant that the system was working as intended and recommending suitable food items. Users explained that the foods that were recommended were mostly accurate, however there were some food items that felt out of place in the recommendations. This can be highlighted by the results from the survey. While most users liked the foods that they were recommended, they noted that the foods were not always very accurate.

Some possibilities for this could be that the time indicated by the user did not allow for accurate predictions, or that there was not a significant amount of suitable recipes in the dataset allowing for an accurate prediction.

Users noted that when they added a larger number of ingredients they liked, they received more suitable recommendations and a better variety of meals. This is due to the cosine similarity method of recommendations. A larger number of ingredients provides more scope for a match in the food dataset.

This also highlights how the user database of previous likes would have added a higher degree of accuracy in the recommendations. The addition of the users likes would have increased the number of ingredients being matched in the user database and made the recommendations more suited to each user. It is therefore very unfortunate that it was not possible to add this feature and suitably test it.

* 1. Evaluation of Project Goals

This section will explore how the project has met the goals set out at the beginning of the project.

Functional Requirements

It is not possible to fully determine if the functional requirements of the project have been met without full software testing. It can however be assessed via observation of the system and by user testing.

The system can read in information from the user, and produce recommendations based on this information. The information provided by the user is accurate as it uses drop boxes and combo boxes, reducing the chance of user error. By making the inputs mandatory, it ensures that the user must input information in order to produce the recommendations.

A suitable number (decided in the design section to be 5) of recommendations are produced each time. This is ensured by using cosine similarity. As each recipe in the dataset is given a score, there will be a full dataset of recipes available to recommend. This means that there will always be at least 5 recipes produced each time the application is used. This was verified during the user testing phase.

Each food in the dataset includes multiple images, a text description, nutritional information and a preparation time. Through human testing, it was ensured that all the recipes have this information stored about them. All this information is displayed each time the user navigates to the user information page.

The average computation time of 0.033 seconds is a very suitable time for computation. This was a target set out in the design section when deciding on the size of the food dataset. The final decision of dataset size was therefore successful, and it resulted in a fast recommendation time.

The system however is not able to store the user’s preferences. This is a failure of the project and has potentially impacted the success of the project. Due to situations outlined in the project management sub section in section 6.4, it was not possible to include this in the final iteration of the project.

Non-Functional Requirements

The non-functional requirements were ones which were tested in the user testing series of testing. The non-functional requirements can be seen as a success. Although the system is not currently easily accessed by users, if it were to be hosted on a web server it would be simple to access.

The test results showed that the users thought the system was easy to use and that the information and layout of the application was very clear and easy to understand. This is a very important part of the recommendation system and vital to ensure repeat use.

The users were mostly satisfied with the recommendations but that the recommendations were not incredibly accurate.

The initial interpretation of these results would mean that the requirement has not been fully met. However, after receiving the feedback, users were still satisfied with the recommendations even if they weren’t fully accurate. The added variety where one specific ingredient would produce an obscure recommendation was something that the users liked. The satisfaction of the user should be seen as more important that just focusing on the accuracy of the system, meaning this requirement has been met.

* 1. Evaluation of Project Process

This section will discuss the strengths and weaknesses of the project. This will include the design and development stages and an overall evaluation of the solution.

Project Management

Throughout the project, there were some issues with the overall management of the project. The first issue came with the design of the user interface, and the decision to use node js. This proved to be a considerable error, as it meant the initial development time were wasted and the user interface had to be remade within flask. This meant there was a loss of time and effort, causing issues with following the timeline of the project. There were also some other issues affecting the timeline of the project, including problems from the current situation in the world as well as other personal issues. The absence of these problems could have led to further improvements added to the project. These will be discussed later on in this section.

Future Improvements

The next step for the system would be to host it on a web server so it could be accessed by users from any location, and not need to be hosted by the user on their machine. It was not felt that this was a necessary step for this project, as it was just exploring the ability to recommend food, and the different forms of media that could be used along with that.

Another step would be to increase the amount of food that can be recommended. This project only focused on dinner foods; however, this could be expanded to include all meals. This would make it a lot more accessible to users and the system would be able to give more accurate predictions.

As the focus of the project was on multimodal recommendations, it would be interesting to explore adding a food recognition system to the project. This is an idea that was discussed in the initial stages of the project, but it was decided that it didn’t fit the main objectives of the project. A food recognition system would allow the user to input an image of food. The system would recognise this image, and recommend similar foods based on it. This could help if a user saw an image on social media of an appetizing meal and wanted to eat something similar. This could be expanded based on an idea found in another paper [9]. This paper used a machine learning tool to identify ingredients from an image of food. This would mean that the system would be able to use those specific ingredients to recommend food to the user.

Adding in these improvements could lead to this project having a commercial use. Food is an ever-present part of today’s life and the ability to access accurate recommendations is a very desirable feature. Increasing the likelihood of accurate recommendations by implementing the above ideas and making the system more accessible could lead to this system being used by many different people, providing a valuable service to the community.

For the system to be commercially viable, it would first need to be fully tested with a series of automated white and black box testing. Unfortunately, these tests could not be completed in this project time frame, however they would be necessary to ensure that there are no errors in the system.

Overall Conclusion

Overall this project is mildly successful. Most of the project’s objectives have been met as well as most of the functional and non-functional requirements. However, there are still some requirements that have not been met. For this reason, the project can not be seen as a full success. The main problem with this project was the poor project management. Better project management would have seen all of the requirements be met as well as potentially even more improvements.

1. Appendices

This section included a bibliography, a user guide and the charts from the user test results.

* 1. Bibliography

[1] A Vallely, “Gitlab Repository”, Queens University Belfast, 24 October 2019

Online <https://gitlab2.eeecs.qub.ac.uk/40181558/food-recommedation>

[2] Denise de Ridder, Floor Kroese, Catharine Evers, Marieke Adriaanse &

Marleen Gillebaart, 2019

Healthy diet: Health impact, prevalence, correlates, and interventions,

Psychology & Health

[3] Thi Ngoc Trang Tran M¨ usl ¨um Atas Alexander Felfernig Martin Stettinger, October 2016, “An overview of recommender systems in the healthy food domain”

[4] Xiaoyan Gao, Fuli Feng, Xiangnan He, Heyan Huang, Xinyu Guan, Chong Feng, Zhaoyan Ming, and Tat-Seng Chua, October 2018, “Visually-aware Collaborative Food Recommendation”

[5] M. A. El-Dosuky, M. Z. Rashad, T. T. Hamza, A.H. EL-Bassiouny “Food Recommendation using Ontology and Heuristics”

[6] Mayumi Ueda, Shinsuke Nakajima January 2011, “User’s Food Preference Extraction for Personalized Cooking Recipe Recommendation”

[7] Elsweiler, David, Trattner, Christoph and Harvey, Morgan (2017) Exploiting Food Choice

Biases for Healthier Recipe Recommendation.

[8] <https://en.wikipedia.org/wiki/Cosine_similarity>

[9] <https://www.bbcgoodfood.com/>

[10] <https://www.google.com/imghp?hl=en>

[11] <https://www.surveymonkey.co.uk/r/BD6TF9K>

* 1. How to Use the System

Requirements

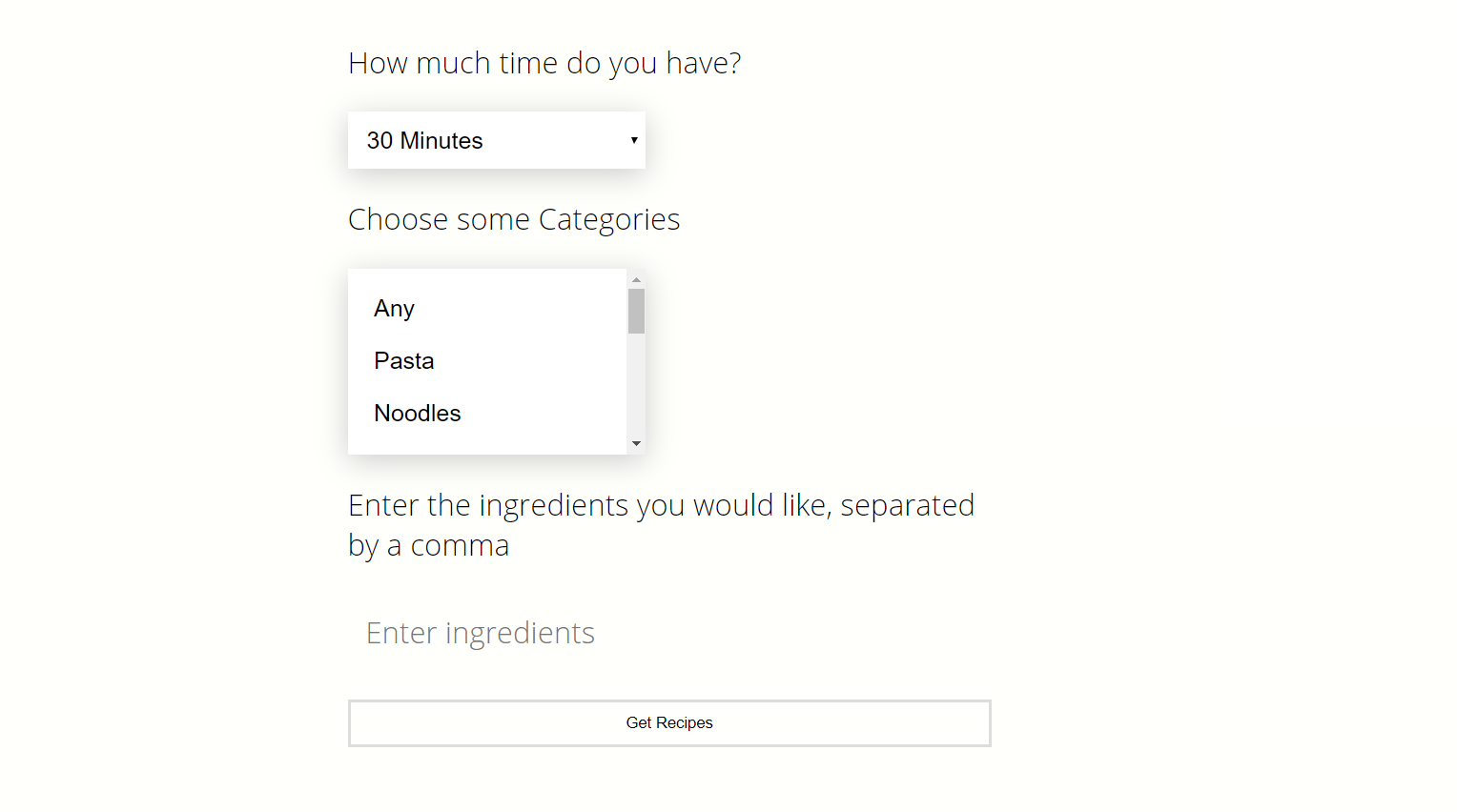
The following requirements are needed to use the system

* Python 3.7
* A suitable Python Interpreter

Configuration

* Download the “System” file from the gitlab repo
* Open this workspace in a python interpreter i.e. PyCharm
* It is recommended that a virtual environment is used
* Install the libraries described in section 4.1
* Run the “app.py” file.
* Open a web browser and navigate to <http://localhost:5000/>

User Interface Guide

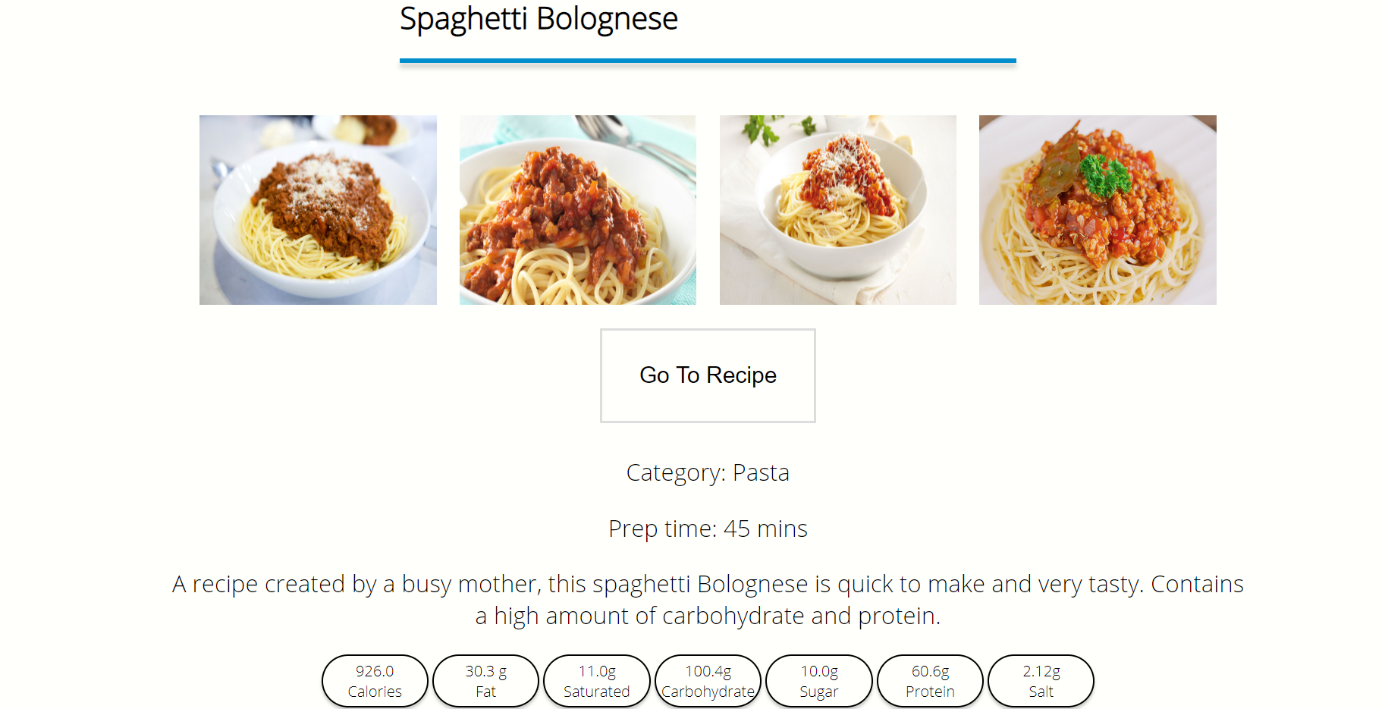


Select the amount of time you have from the dropdown menu

Select some food categories you would like from the multi select box. Hold the Ctrl key to select multiple options.

Type in some ingredients that you like. Separate these ingredients with a comma.

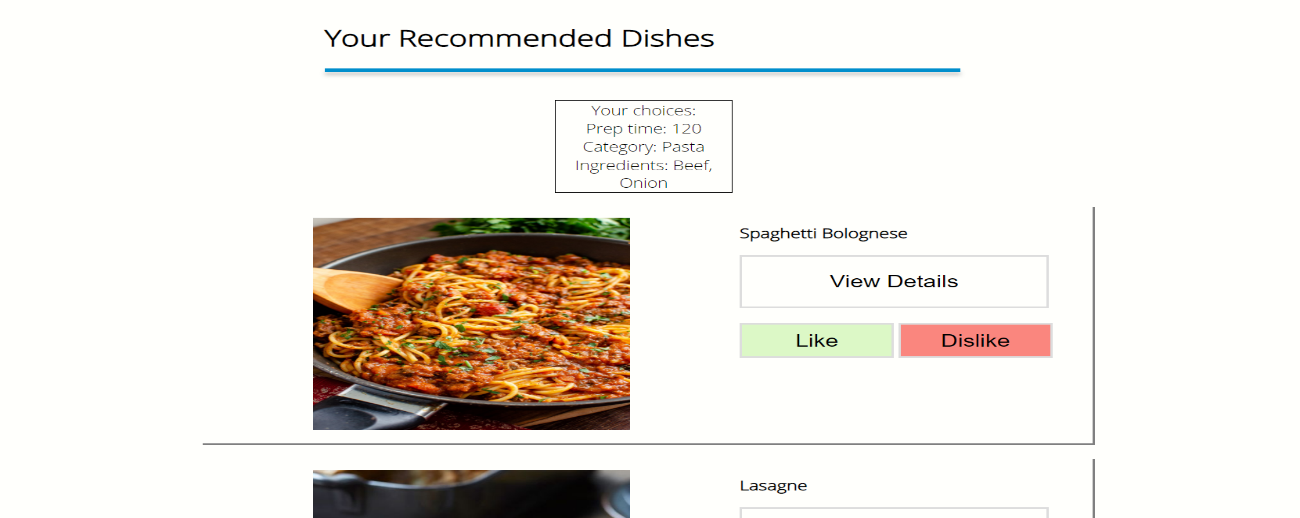
Once all the information has been added, click the Get Recipes button



Text information about the food

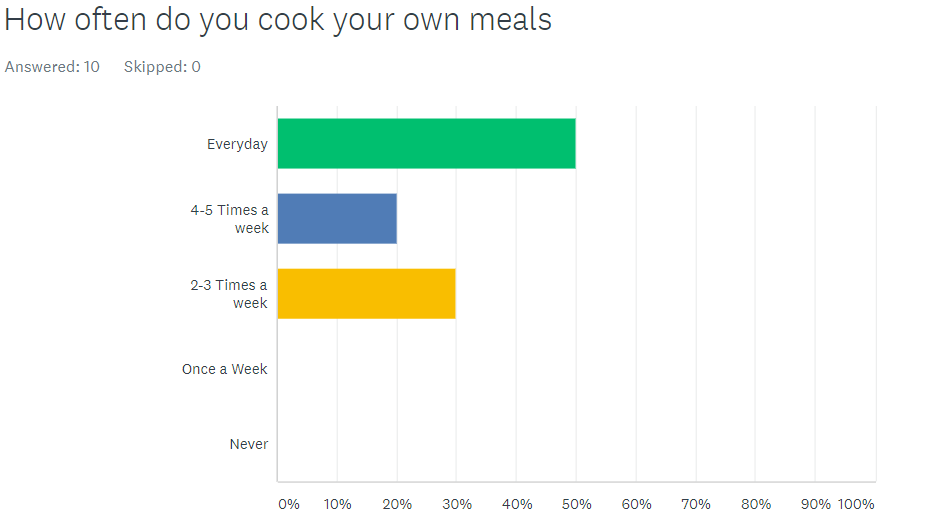
Nutritional Information of the food

Click here to few a step by step recipe for this food



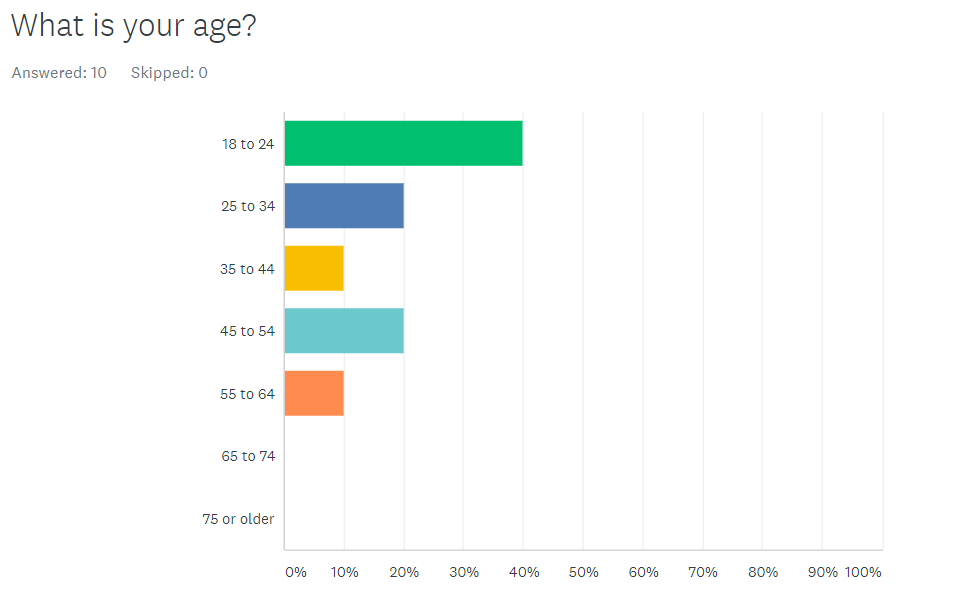
Your Recommendations are displayed on this page. Scroll down to view them all

Click the View Details button to view information about each food

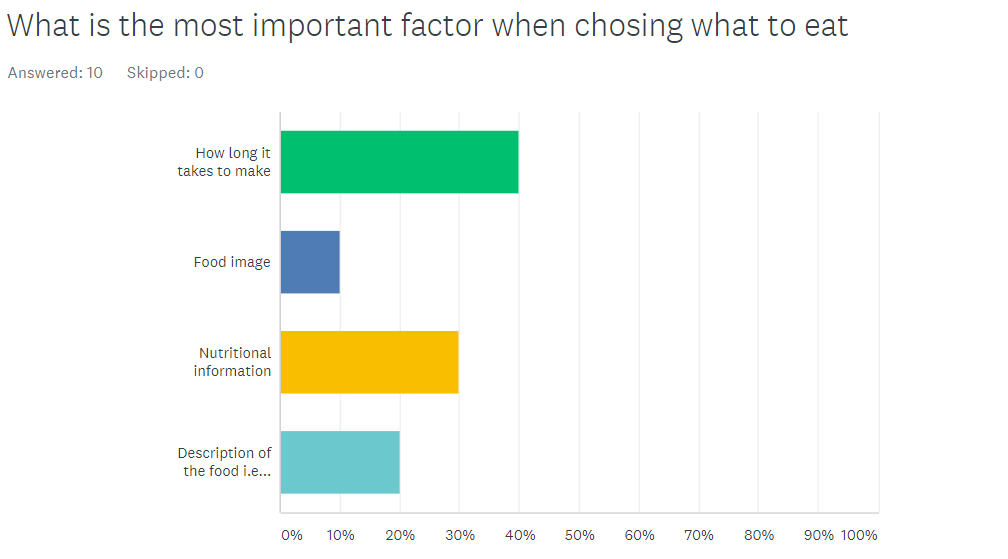
* 1. Test Results

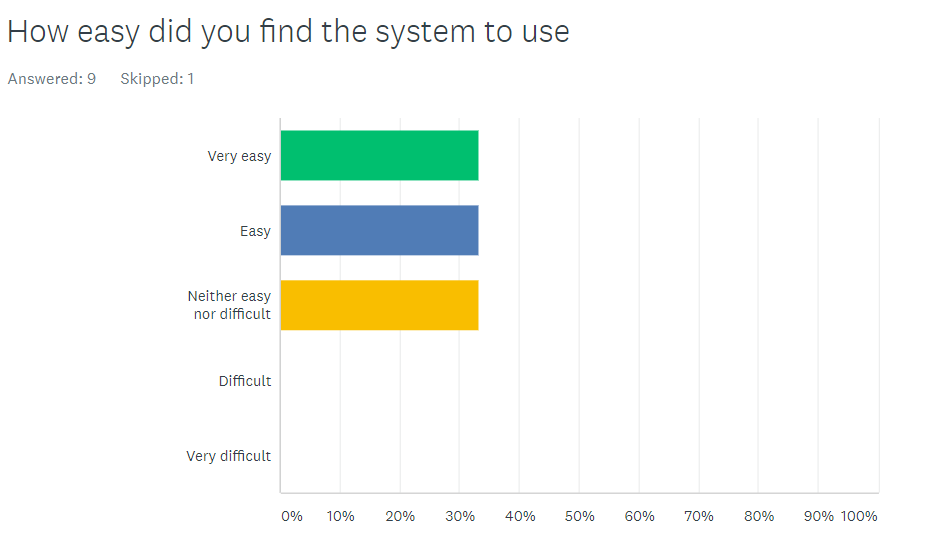
*Figure 7 Frequency of Home Cooking*

*Figure 6 Age of User*

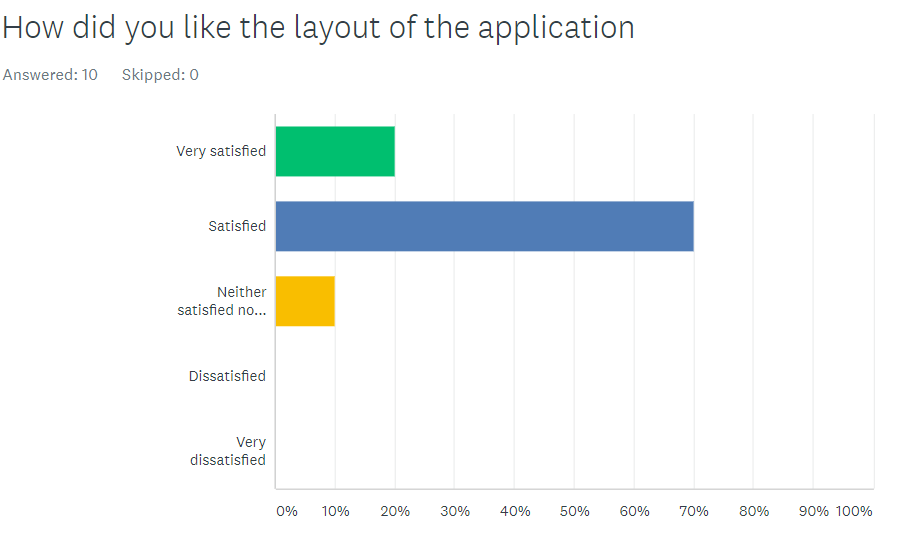
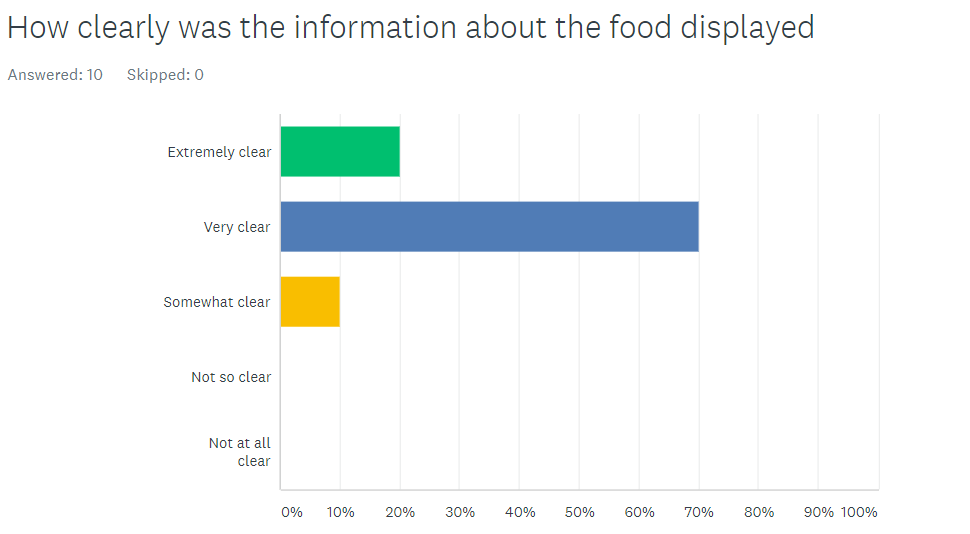


*Figure 8 Most Important Factor when deciding what to eat*



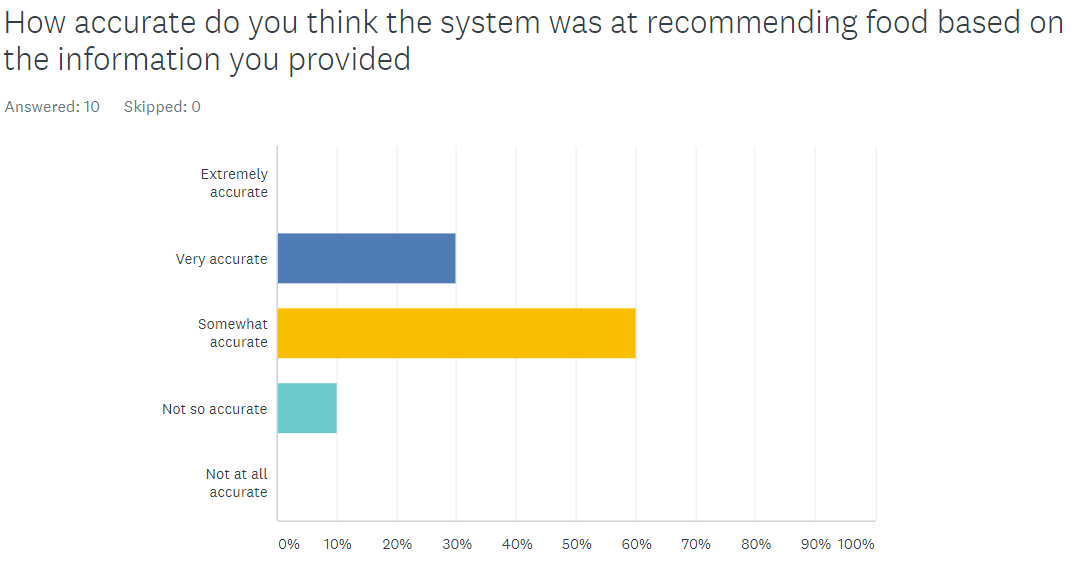


*Figure 9 Ease of Use of System*



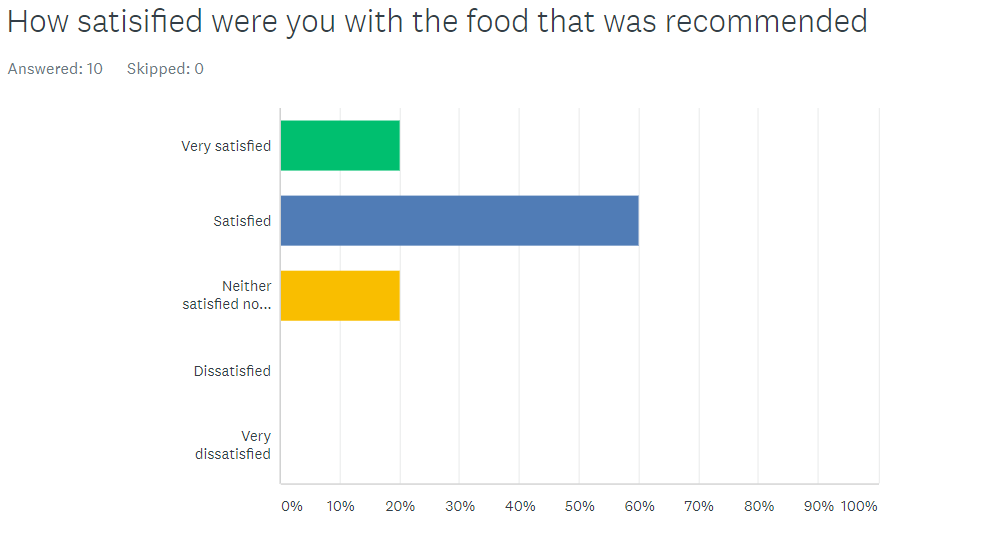
*Figure 11 Displaying Information*

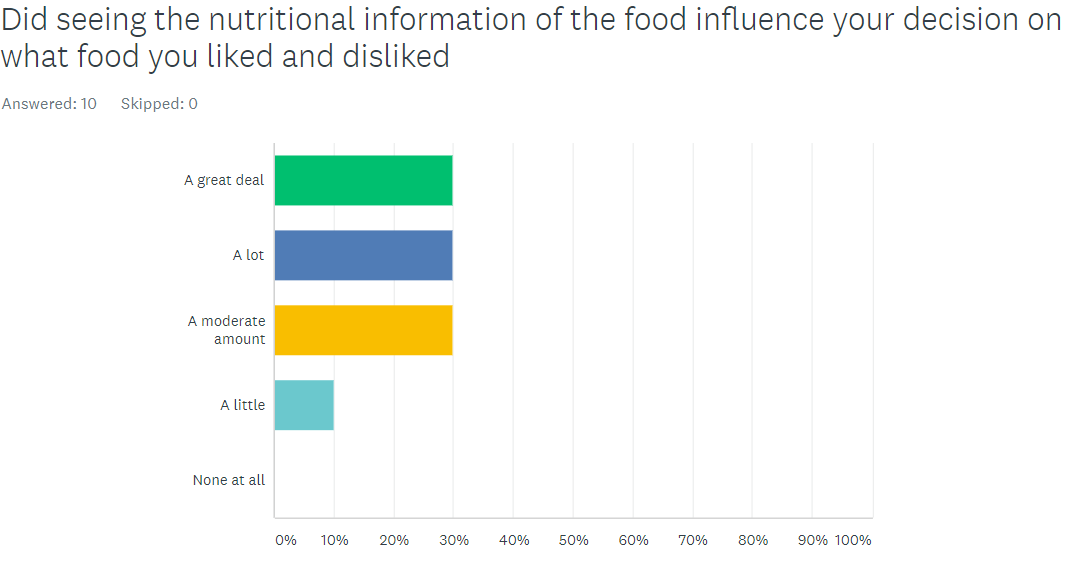
*Figure 10 Layout of System*



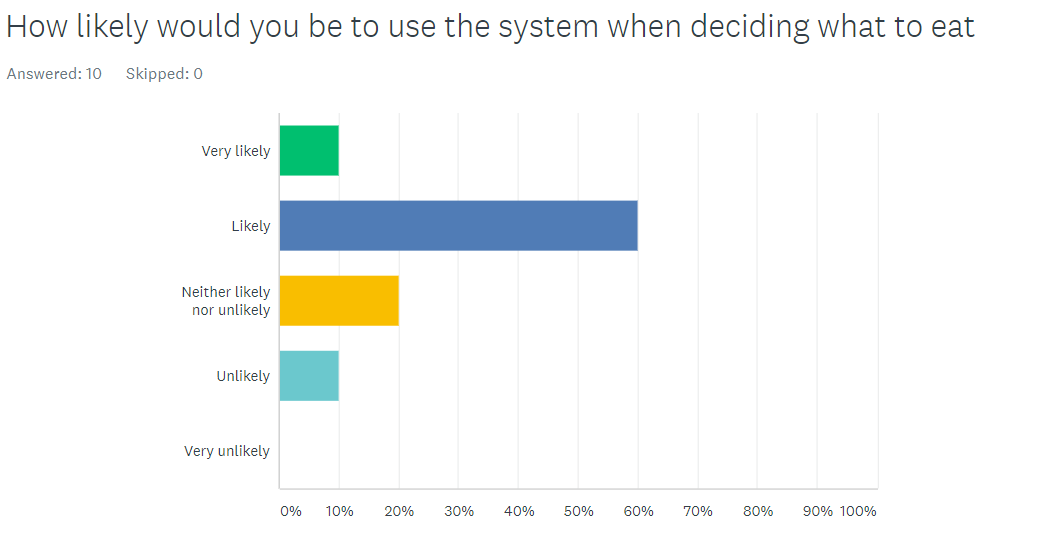
*Figure 13 Accuracy of Recommendation*

*Figure 12 Satisfaction with Recommendation*





*Figure 14 Influence of Nutritional Information*



*Figure 15 Would they use the system again*