mySquareMeal

A Project Report
Presented to
The Faculty of the College of
Engineering

San Jose State University
In Partial Fulfillment
Of the Requirements for the Degree
Master of Science in Software Engineering
Master of Science in Computer Engineering

By
Mandar Gade, Neha Kumar, Setu Pandya, Varun Mahendra Shah
May 2018

APPROVED

Dan Harkey, Project Advisor

David Bruck, Director, MS Computer Engineering

Dan Harkey, Director, MS Software Engineering

Xiao Su, Department Chair

ABSTRACT

mySquareMeal
By Mandar Gade, Neha Kumar, Setu Pandya, Varun Mahendra Shah

Food management is one of the biggest problems facing the world today. There are people that eat more than their requirement or those that eat less. There are multiple solutions available where upon entering physical parameters one can get an information about number of calories to consume. There are also solutions that suggest diets to users. What we find lacking is a solution in which user gets tailored options for meals.

Another area is where user can select favorite cuisines and is suggested restaurant options based on that. We are looking to enhance this area. For example, if a user eats lunch between 1:00 – 2:00 pm every day and likes to eat yoghurt, ham, cheese and kale, there is no system available that can suggest restaurant options based on individual food preference open at that time.

In our application, a user can select his favorite foods and what time of the day they like to eat them. We have used the Zomato API along with Spoonacular's Food Nutrition information API as data sources. For the above-mentioned example, our system suggestz a Ham, Kale and Cheese Sandwich with a yoghurt side at IHop or a Ham Sub with Cheese at Subway. User can review the choices our system makes if they decide to try them out. Based on the reviews, our system will learn more about the user's food eating patterns and choices and will make better recommendations with more data. We can also use the data we collect to show eating trends per geographical zones and popularity wise food ranking. These visualizations can be used by healthcare professionals to target specific areas of promotion for healthy eating awareness programs.

Acknowledgments

The authors are deeply indebted to Professor Dan Harkey, Professor Vicki Parrish, Mr. Sunil Garg, Mr. Siddhant Gujar, Mr. Ankur Sheth and Mr. Hanumant Jagtap for their invaluable guidance throughout the duration of this project.

Table of Contents

Chapter 1. Project Overview 1
1.1 Introduction 1
1.2 Proposed Areas of Study and Academic Contribution 2
1.3 Current State of the Art 5

Chapter 2. Project Architecture 7
2.1 Introduction 7
2.2 Architecture Subsystems 8

Chapter 3. Technology Descriptions 10
3.1 Client Technologies 10
3.2 Middle-Tier Technologies 10
3.3 Data-Tier Technologies 10

Chapter 4. Project Design 11
4.1 Client Design 11
4.2 Middle-Tier Design 14
4.3 Data-Tier Design 17

Chapter 5. Project Implementation 19
5.1 Client-Side Application 20
5.2 Application Server 20
5.3 Machine Learning Server 22
5.4 NoSQL Datastores 23

Chapter 6. Testing and Verification 24
6.1 Test and QA strategy 24
6.2 Verification Plan 26

Chapter 7. Performance and Benchmarks 27
7.1 Performance Plan 27
7.2 Benchmarking Strategy 28

Chapter 8. Deployment, Operations, Maintenance 29
8.1 Deployment Strategy 29
8.2 Operational Dependencies 29
8.3 Maintenance Schedule 30

Chapter 9. Summary, Conclusions, and Recommendations 31
9.1 Summary 31
9.2 Conclusions 31
9.3 Recommendations for Further Research 32

Glossary **33**

References **34**

List of Figures

Figure 1 Project Architecture.....7
Figure 2 Use Case Diagram.....13
Figure 3 Activity Diagram.....14
Figure 4 Sequence Diagram (User Registration).....15
Figure 5 Sequence Diagram (User engagement).....16
Figure 6 Project implementation.....19
Figure 7 Rest Web Services.....21
Figure 8 MVC Request Lifecycle.....22
Figure 9 Agile Lifecycle.....24
Figure 10 Deployment Strategy.....29

Chapter 1. Project Overview

1.1 Introduction

mySquareMeal provides users with an option to choose healthy meals based on their pre-existing health conditions and food preferences. We have many smartphone applications that use machine learning to study user preferences and recommend users healthy meals but none take into consideration factors like a user's potential reaction to an ingredient due to medical drugs they are taking and other preexisting allergens. We also have applications, that recommends a meal to user based on calorie count, user's physical activity, location, and user preference.

Our application makes recommendations based on a user's health condition in terms of certain disease. It is important for a person with a non-curable disease like diabetes or blood pressure anomalies to maintain a healthy diet and our app will provide easy food management for such people. Healthy eating not only maintains physical health of a person, but also helps a person to be mentally strong when he or she is dealing with it.

Users create profiles like those created when visiting a General Practitioner for the first time. Based on this profile and location of a user we can provide them with a list of meal options they can have. It includes real time analysis of restaurant data, location, user preferences and a machine learning model that will study all characteristics of this profile and provide the final recommendations. One of our main challenges is to integrate data

from multiple sources and derive a meaningful dataset that we can use to train our ML model. We not only need to study the popularity of a meal, but also the ingredients and then weigh this against health parameters. Users will also provide feedback for the recommendations made and system will consider user preferences for making better recommendations. This will improve the accuracy of the recommendation algorithm over time. Our project's implementation is heavily dependent on availability of relevant datasets and rich user profiles.

1.2 Proposed Areas of Study and Academic Contribution

This project is primarily an application of Machine Learning. Machine learning is a computer science concept that involves training a computer program to output a decision with the simplest being a 'yes' or a 'no' answer based on pattern or other recognition employed by the program. Essentially, it aims to aid the decision-making process by creating hierarchies of objects for each input it receives and then uses these hierarchies to form a pattern that it compares with a master pattern to *learn* which patterns lead to which decisions. The ability of the program to take decisions with accuracy depends on the type algorithm used to build patterns from object hierarchies. Some of the more popular algorithms are Isotonic Regression, Linear Regression, Convolutional Neural Networks and Naïve Bayesian Algorithm. Each algorithm provides a different level of accuracy for the same input data even though fundamentally they are designed to provide the same output. Since Machine Learning is a tool that aims to substitute human decision-making process, it must process information in the same way humans do. But the decisions we take

are extremely complex problems from a computational point of view and hence these algorithms are trained by subjecting them to a sample pattern set.

Machine Learning while being a powerful tool on its own can be further boosted. Boosting is a technique that can be used to better the efficiency of any learning algorithm. AdaBoost (Adaptive Boosting) developed by Yoav Freund and Robert Schapire can be employed to improve the performance of our Model (Freund & Schapire, 1995).

Since the advent of first mobile phones and then smartphones, the sensors embedded in these phones provide a treasure trove of data that if processed can lead to meaningful information gain. One such parameter is location data tracked from a sensor on all smartphones that provides latitude and longitude of the device by communicating with the Global Positioning Satellite(GPS). From an architectural point of view, the location-based service (LBS) of mobile communication and the personalization of information recommendation are two important trends in the development of e- commerce. Many previous researches focus on one of the two things. In our application, we are trying to integrate both by finding how can we use LBS and integrate a user's preferences and prevalent health conditions while recommending the meal options (Kuo, Chen & Liang, 2009, p 3543-3554).

Our application involves building a user profile based on type of food habits and preexisting diseases. We will then match this profile with healthy food choices and will

correlate these choices with a location based dataset of restaurant menus. This will enable us to recommend to users, healthy meal options in restaurants near their current locations. An existing application like what we are planning: SmartDiet is a system for diet planning. It provides a user a combination of meals and restaurants based on user's food preferences. It is a recommendation system for personal diet management. Moreover, the recommendations can be improved with user's feedbacks (Hsiao & Chang, 2010, p 421-425).

Our biggest dependency is availability of quality data. The problem we are trying to solve is unique and hence there is no readily available source of data for us to feed to a Machine Learning algorithm. Research into the most promising available datasets, Diabetes dataset (Lichman, 2013), Food Environment Atlas (US Department of Agriculture, N.D.) and approaches for what we are trying to accomplish (Li & Arandjelovic, 2017, p 357-360) reveals two things: 1) There is no dataset that is readily available and provides all the parameters that we need. We will need to perform ETL (Extract, Transform, Load) Process on two or more datasets to tailor them for our use. 2) There are multiple algorithms to choose from and we need to select and test the best candidates to identify the one that gives us an optimal result that balances accuracy with performance (Shaikh, Blog, Rizvi & Jain, 2017).

The ETL process employs the use of statistical and mathematical formulae to manipulate data sources and transform them into viable streams that can be fed continuously to our

ML implementation. Specific algorithms with their pros and cons for this implementation are already discussed above. From studies that were referred during our research, our biggest challenge is to find a suitable algorithm that provides consistently accurate results.

1.3 Current State of the Art

The growth of machine learning both as a concept and on an application level has been phenomenal especially in the last couple of years. Everyone in the information technology industry talks about applying machine learning whether they understand the concept fully or not.

To explain the simplest machine learning algorithm in a layman's language: It is a tool that takes a set of inputs, processes them and gives a 'yes' or a 'no' answer. An example specific to what we are aiming to accomplish is, imagine you have some preexisting health conditions and call your doctor. You ask him/her, "Can I eat a Caesar Salad?" The doctor will respond with 'yes' or 'no'. Our system will substitute your doctor and go a step further. It will recommend other meal options to you based on your location. The more recommendations our system makes, the higher its accuracy rate will be. But there is a limitation to this approach as latest research suggests, as the amount of data keeps growing, machine learning algorithms are unable to keep up. To keep up with this growth in data it is essential to look to an even newer and more exciting concept; deep learning.

Deep learning builds hierarchies of objects instead of providing simple ‘yes’ or ‘no’ answers and uses these hierarchies to improve performance benchmarks of machine learning algorithms like interoperability, data dependencies and execution time. State of the art research in this field points to the direction that deep learning algorithms will not slow down and are able to keep up with increasing amounts of data.

To summarize, we can say that both machine learning and deep learning are nascent concepts and work done in either of them or their combination will help push forward the boundaries of research in computer science.

Chapter 2. Project Architecture

2.1 Introduction

mySquareMeal is a smartphone application that communicates with a central server before redirecting output recommendations back through the app to a user. It provides location and user information to the server. The recommendation system has data on multiple sources integrated into three main data stores: user profile data set, restaurant's data set, data set for learning model. Recommendation system further mines these data stores to get meaningful data to perform analysis and learning.

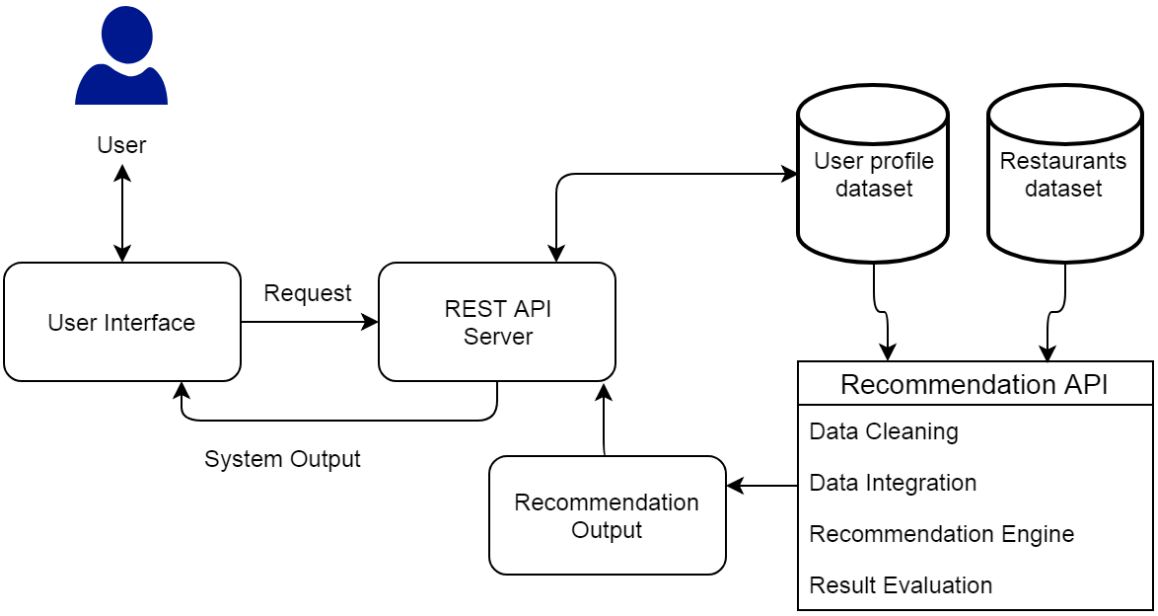


Figure 1 Project Architecture

2.2 Architecture Subsystems

2.2.1 Smartphone application

- The application will be a responsive web based UI. The advantage of having this over a native Android/iOS/Windows application is that we get device and OS based independence.
- Smartphone application is important in terms of user interaction with the system. It is location independent and thus has an advantage of collaborating with system in different conditions.
- Smartphone sends request to server with input parameters required for recommendation analysis and displays recommendations.
- System operation is dependent on location tracking capability of smartphone application that is implemented using Google Places API.

2.2.2 Application Server

- Server performs an important task of integrating operations of different modules such as data stores, recommendation system, smartphone application communication and security policy configuration.
- Server is also responsible for maintaining and updating user profiles, which is an important input parameter for our recommendation system.

2.2.3 Data Analysis, Integration and Recommendation system

- mySquareMeal is dependent on data availability and integration. Data analysis and integration is important for deriving meaningful data that we can use for training and testing our recommendation model.
- Recommendation system uses machine learning techniques to derive the recommendation. It uses the user's profile and location to categorize the available data and operate learning on this model to cater to unique recommendations.

2.2.4 NoSQL Data Stores

- The system has three data stores, that contain integrated data from multiple sources such as user profile data, restaurant data and data to train the learning model. These three data stores are as follows:
    1. User profile data set
    2. Restaurants data set
    3. Data set for learning model

Chapter 3. Technology Descriptions

3.1 Client Technologies

HTML 5, CSS 3 are used as markup language for our UI along with **Bootstrap 4.0 that** is a library containing design and formatting elements for front end web development. Bootstrap themes and designs support responsive layouts by default hence we do not need to handle these explicitly while building the UI of the application. **AngularJS 5** is used to enable dynamic binding to give a native app like experience while also enabling cross platform framework use for Bootstrap enabled front end.

3.2 Middle-Tier Technologies

Java Spring is used to create and provide REST API endpoints for use in the Front End. Python libraries (pandas, numpy, scikit-learn) are used to implement Machine Learning that is housed on a NGrock server.

3.3 Data-Tier Technologies

MongoDB is used to store our restaurant dataset as well as machine learning data. All data operations are performed by formatting the data using **JSON** (**JavaScript Object Notation**).

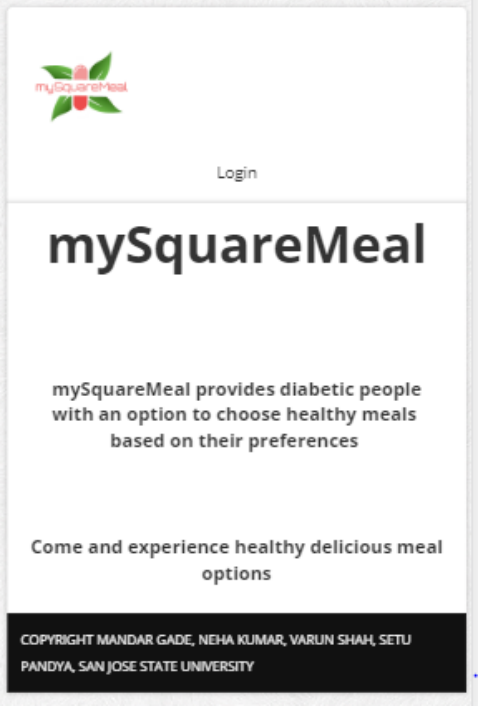
Chapter 4. Project Design

In our project, there are three actors associated with the system. User can perform registration, login and view results of food. Moreover, user can add his/her details (food preferences). Application server authenticates the user. Machine Learning System is responsible for providing recommendation and data integration.

4.1 Client Design

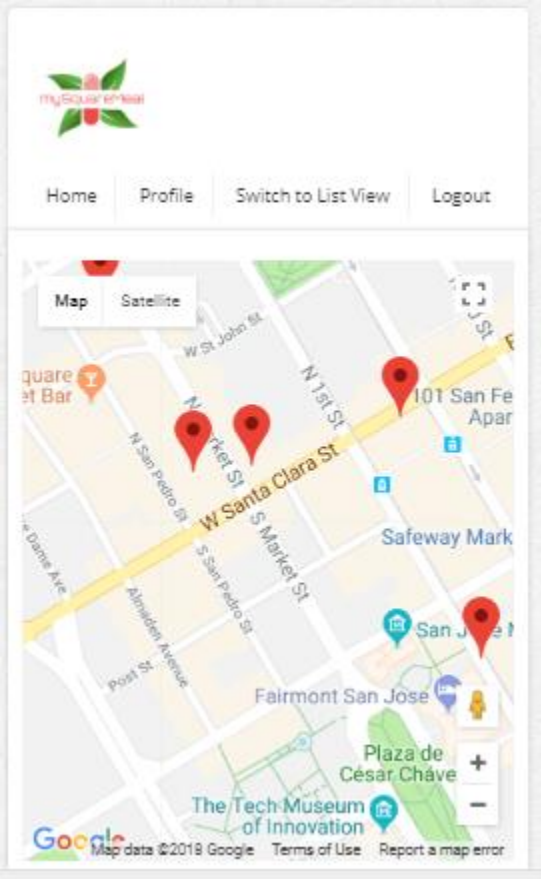
4.1.1 UI Mockups

Welcome



Registration

Map View


Home
Profile
Switch to List View
Logout
Map
Satellite
W St John St
N 1st St
W Santa Clara St
N San Pedro St
S Market St
S San Pedro St
Almaden Avenue
Post St
Safeway Mark
Fairmont San Jose
Plaza de
César Chávez
The Tech Museum
of Innovation
Google
Map data ©2019 Google
Terms of Use
Report a map error

4.1.2 Use case Diagrams

A use case diagram involves use cases and actors. Use cases are the functions those are performed by different actors in the system. Use case diagram shows how the users interact with the system.

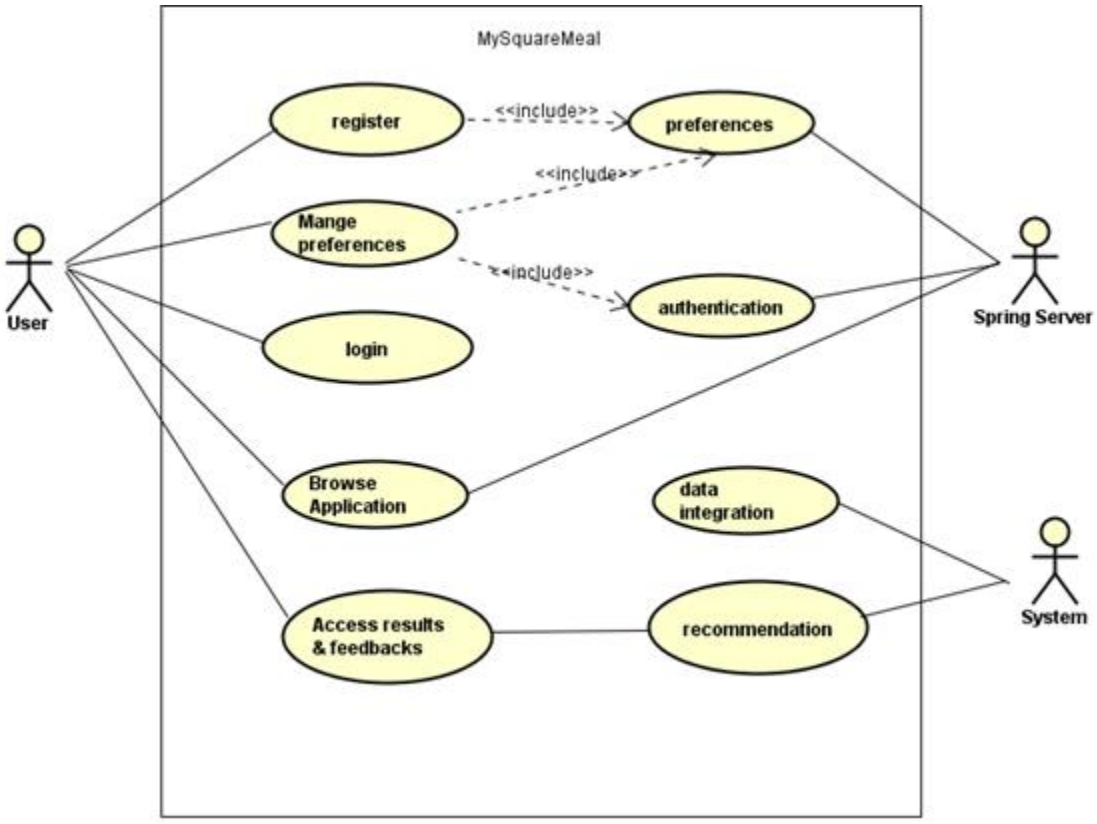


Figure 2 Use Case Diagram

4.2 Middle-Tier Design

4.2.1 Activity Diagram

Activity diagram involves activities in a stepwise manner. It represents workflow of the system. It shows the flow from operation to another operation. It represents control flow between activities. The flow in activity diagram can be sequential or concurrent. User first registers and login to the system. Then user can view recommendations, manage his food preferences and enter feedback.

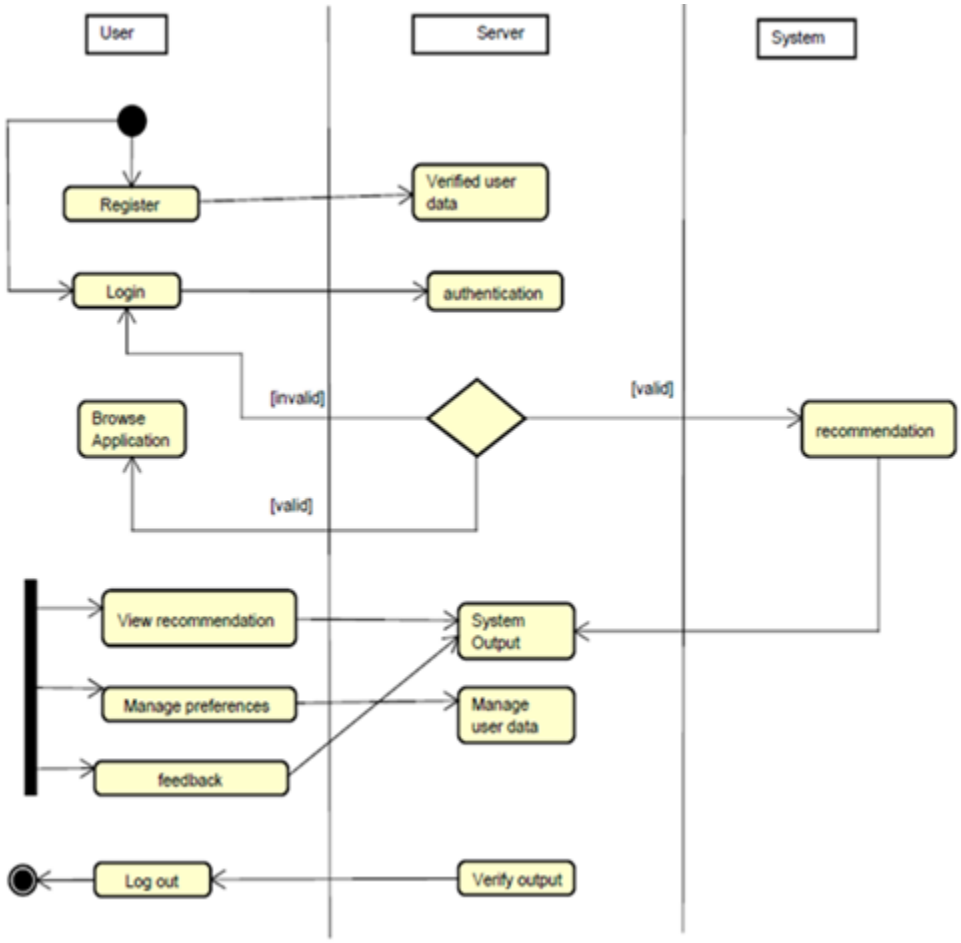


Figure 3 Activity Diagram

4.2.2 Sequence Diagrams

Sequence diagram for user registration:

When user registers to the system, the system validates the details and creates a user profile that is stored in the database. Also, user provides the preferences that are essential for recommendation system. Preferences are part of user profile.

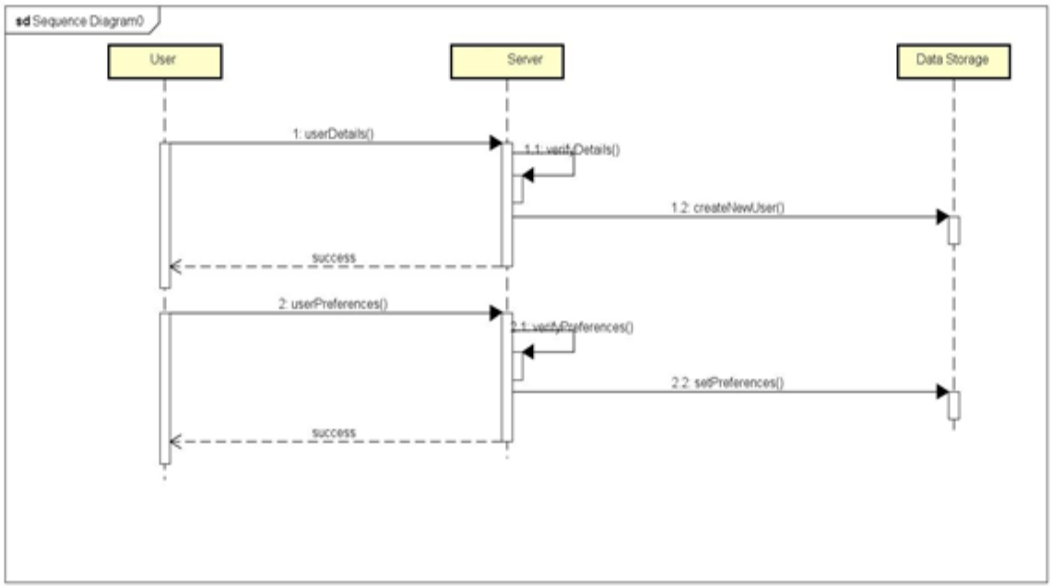


Figure 4 Sequence Diagram (User Registration)

Sequence diagram for user engagement with the system:

The following diagrams represents the user engagement with the system. User has a session with the system in which recommendation is provided to the user according to his location and preferences.

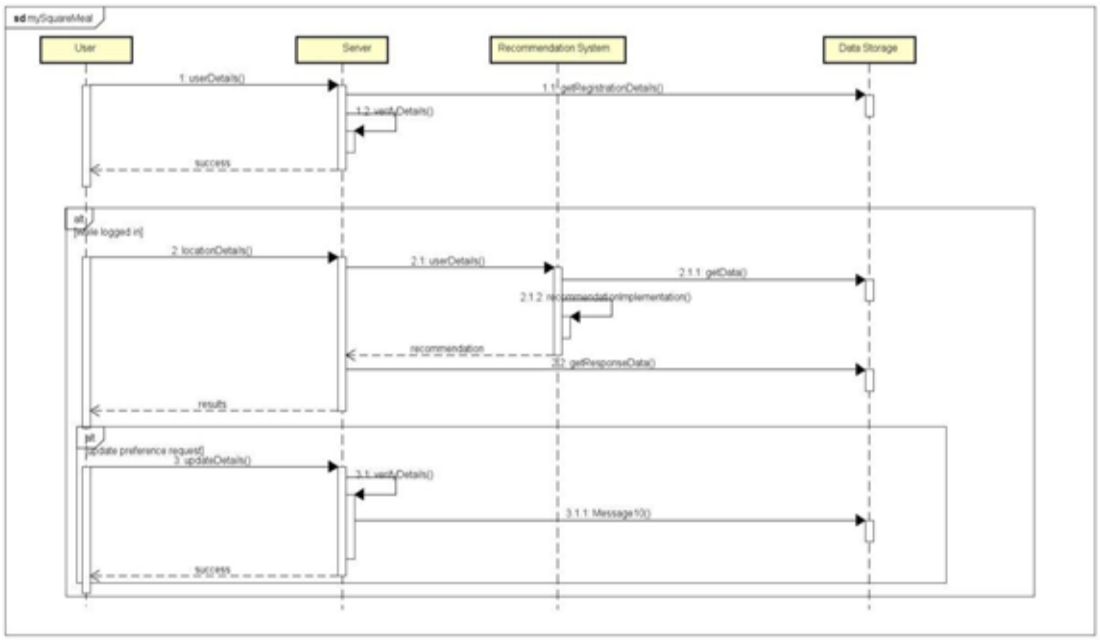


Figure 5 Sequence Diagram (User engagement)

4.3 Data-Tier Design

We are using a NOSQL database architecture. NOSQL databases do not have a schema. Data is represented in the form of documents. Below is the example of JSON document used in the module development of mySquareMeal:


```
{
        "id": "16839913",
        "name": "Adelita's",
        "location": {
                "address": "1896 Curtner Avenue, San Jose 95124",
                "locality": "Willow Glen",
                "city": "San Jose",
                "city_id": 10883,
                "latitude": "37.2752694444",
                "longitude": "-121.9235916667",
                "zipcode": "95124",
                "country_id": 216,
                "locality_verbose": "Willow Glen, San Jose"
        },
        "cuisines": "Mexican",
        "menu": [
                {
                        "name": "Pineapple Pork Tortillas",
                        "id": "168399131",
                        "ingredients": [
                                "ground cinnamon",
                                "fresh cilantro",
                                "chili powder",
                                "ground coriander",
                                "kosher salt",
                                "ground black pepper",
                                "garlic",
                                "plum tomatoes",
                                "avocado",
                                "lime juice",
                                "flank steak",
                                "salt",
                                "ground cumin",
                                "black pepper",
                                "olive oil",
                                "crushed red pepper flakes",
                                "onions"
                        ]
                },
                {
```

```
                "name": "Chilli Salad",
                "id": "168399132",
                "ingredients": [
                        "roma tomatoes",
                        "kosher salt",
                        "purple onion",
                        "jalapeno chilies",
                        "lime",
                        "chopped cilantro"
                ]
        }
    ]
}
```

Chapter 5. Project Implementation

We have followed the agile development methodology for the project and have used a Kanban Board: https://waffle.io/varundrogba/mySquareMeal to manage the project.

For source code management, version control and collaboration, we have used GitHub: https://github.com/varundrogba/mySquareMeal/ GitHub is a free version control system that also provides performance benchmarks and other useful metrics to track and manage projects.

We have added an MIT License our code repository on GitHub so our code can be used by anyone requiring only preservation of copyright and license notices.

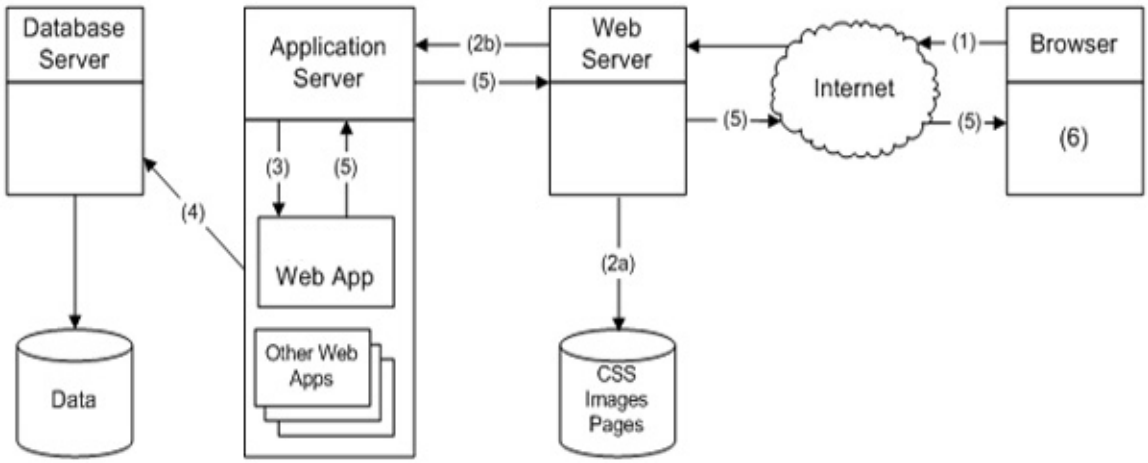


Figure 6 Project implementation

5.1 Client-Side Application

Below is the architecture design for implementing the client application and how it will interact with server.

Execution includes following steps:

- Creating data for the preprocessed recipe and foods to avoid, inSQL.
- Creating user information table that has details about user and his preferences.
- Creating the UI wireframes for application flow.
- Creating user registration page.
- Creating user login page.
- Creating user preference page to update his health conditions.

5.2 Application Server

- Implementing backend services in Flask Framework.
- Implementing authorization and validating if user is already registered for every user login and registration.

Figure 7 Rest Web Services

Keeping our system distributed enables our servers to share resources and perform tasks concurrently. It also distributes our points of failure so there is higher fault tolerance and increased scalability. Web Services are the ideal way that our platforms can use to communicate with one another. The application server handles this role. The service is exposed for use by defining a root URL so it can be accessed. Then I defined a collection of resources (food recipes) in the form of JSON data objects that can be operated upon by using the state transition methods defined above (GET/PUT/POST/DELETE).

Each resource in the collection had a unique identifier for access. Also implemented an authentication protocol to prevent unauthorized access to the Web Service.

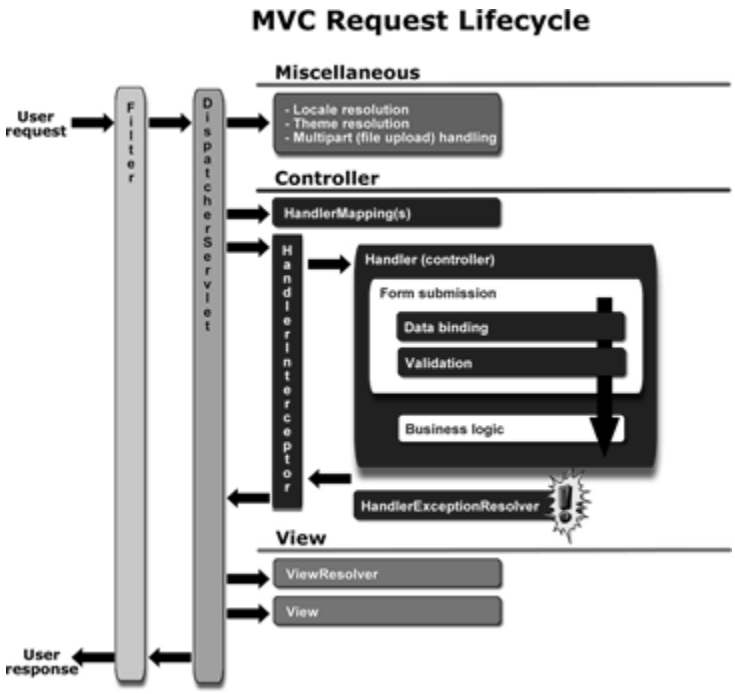


Figure 8 MVC Request Lifecycle

5.3 Machine Learning Server

- Read the json data using ‘read_json()’ function of pandas.
- Preprocess the data in pandas data frame using python.
- Divide the data into training set and testing set using ‘train_test_split()’ function from ‘cross_validation’ class of sklearn library.
- Implement the learning model.
- Test the predictions using ‘accuracy_score()’ function from ‘metrics’ class of sklearn.
- Repeat steps to increase accuracy

5.4 NoSQL Datastores

- Create a new MongoDB cloud instance using mLab - that hosts our NoSQL database on their Cloud.
- Reference this instance to the Machine Learning Server hosted using ngrock also on the cloud.
- User details and auth handled by application server and Auth(0) framework enabling us to not handle data storage using our database instances.

Chapter 6. Testing and Verification

Quality of a software product is determined by how well it performs when tested for all its possible use-cases. The most important aspects to define for our software QA process are the test cases and scenarios we will develop to make sure the product that we deliver always performs as expected.

6.1 Test and QA strategy

6.1.1 Environment Testing

Develop test cases simultaneously along with development and perform full system testing for each scrum iteration of the iterative development process.

Figure 9 Agile Lifecycle

We have employed the use of Test Case management tools to adjust and track our overall test strategy.

- Atlassian Jira for defect and bug tracking
- Atlassian Test Rail for Test Case Management and result logging
- Charles Proxy for API testing

6.1.2 Unit Testing

Team members have written unit test cases simultaneously for features they developed. Feature selection and task ownership is already defined on our Kanban Board so it is the responsibility of individual developers to write unit test cases that cover every line of code that is committed into our GitHub master branch.

6.1.3 Integration Testing

Given the number of actors in our product it is important to perform integration testing on multiple levels. The first step will be to integrate newly developed modules with existing modules for that specific environment. The second step will be to make sure that the module runs in the same way on the integrated development platform as it does on the individual environment. This is critical for isolated modules that may have no interaction with cross-environment modules.

6.1.4 System Testing

This is the final phase of testing in which the system is rigorously tested to weed out bugs. We have developed a bug priority list and will use root cause analysis methods to develop fixes and test them on each level before repeating full system testing.

6.2 Verification Plan

6.2.1 A/B (User acceptance testing)

We will release our product for user acceptance testing to a few users and monitor the performance of the product in this production setting. We will also employ performance testing techniques at this stage to develop performance metrics and benchmarks that we can use to optimize the production.

6.2.2 Product Backlog Review

The product backlog items left on the board after completion of the QA plan will be reviewed and further development or testing efforts if needed will be determined.

Chapter 7. Performance and Benchmarks

Performance testing is conducted iteratively, and benchmark identification is performed. The main reason for this is the presence of multiple actors interacting with each other. It is not practical to focus on performance optimization without having interactions-based performance benchmarked completely because only then will we be able to identify and prioritize performance bottlenecks.

7.1 Performance Plan

7.1.1 Load testing

We subject our servers to sudden spikes in number of requests and analyze how they manage this phenomenon to identify weak areas, see if we can manage spike load better.

7.1.2 Stress testing

We subject our servers and data stores to sustained levels of above average requests over extended periods and find ways to optimize performance in these environments. This is important from a product maintainability and long-term durability perspective.

7.2 Benchmarking Strategy

7.2.1 Capacity testing

Benchmarking is employed to push loads on the application as well as the servers to levels at which performance slows to an unacceptable level. This will allow us the gauge maximum capacity of individual components that we can then target specifically to improve performance.

To carry out the testing of the functionality we must check the basic functionality and cover all corner cases so that every user is unique. To make sure this is achieved, a unique screennameoremailidisrequiredduringregistration.

In addition, must verify that the required setup is established or not. All the dependencies are present so that the execution takes place properly. Any exception or error generated is logged and handled gracefully.

Chapter 8. Deployment, Operations, Maintenance

8.1 Deployment Strategy

Deployment plan was to begin development in a sandboxed mode where development as well as integration will be frequently deployed and tested without much control.

As development progresses, we began deployment control measures and move to a pre A/B testing sandbox integrated environment in preparation for full system and A/B testing. Once A/B testing is completed and high priority bugs are resolved we will move to a highly controlled deployment model on our final production environment.

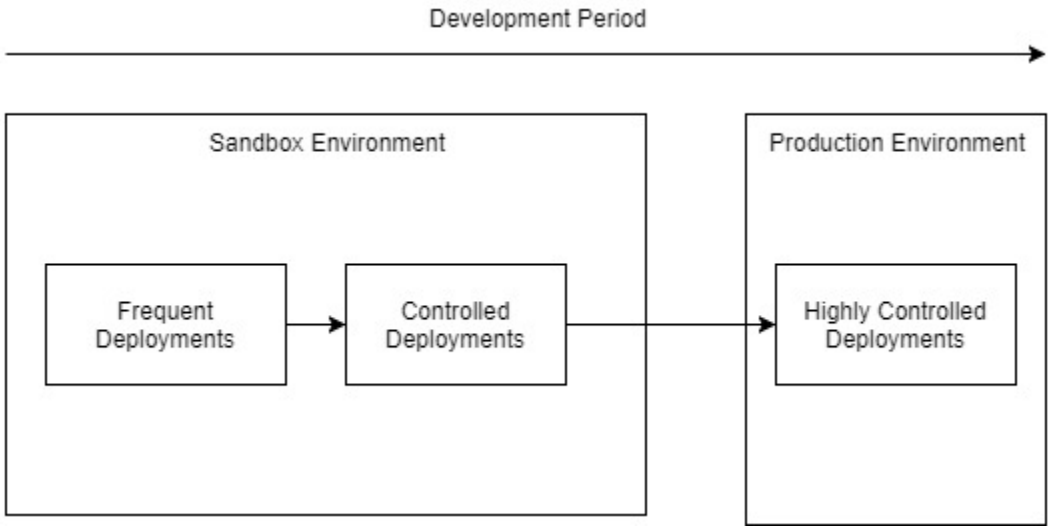


Figure 10 Deployment Strategy

8.2 Operational Dependencies

- Continuous availability of Application Server and Machine Learning Server.
- Location availability from client user.
- Availability of Data Stores to Application Server and Machine Learning Server.

8.3 Maintenance Schedule

- Inactive user accounts will be deleted after every six months.
- User location will be based on input by user. The app will not fetch user location in background ever so as not to breach user's privacy.
- Restaurants and menus databases will be updated once every month.

Chapter 9. Summary, Conclusions, and Recommendations

9.1 Summary

A meal recommendation system that can be accessed via a web browser from any platform to provide real time location based meal options was developed. No hardware was utilized on any level by moving all infrastructure needs in cloud environments. Object oriented programming principles along with REST api endpoints make sure to provide resources as is to a end user while abstracting implementation details completely. Server redundancy was created by splitting the middleware into two servers to avoid a single point of failure scenario making the environment truly robust and dependable. Human factors were extensively measured to give not only the best UX but also the safest UX while reducing chances of inadvertent privacy breach possibilities.

9.2 Conclusions

- Building native platform specific applications should be avoided as this increases overhead significantly while not providing significant performance returns over browser based progressive web applications. With the recent developments in web applications like HTML 5, CSS 3, Bootstrap 4.0 and controllers like Angular 5 make it increasingly easy to develop lightweight web apps while providing the full suite of features that a native application would have provided. They are only limited by the limitations of the browser in which they run.

- Machine learning is a powerful tool when employed on quality data. It is very important to have distinct measurements for data quality before thinking of machine learning models. The data must always come before the model.
- Significant optimizations in improving speed of recommendations delivered should only be done once model accuracy and dataset quality exceed expectations.
- Food and health management are one of the biggest crises our world faces and burgeoning human population directly correlates to this. It is important for further research to continue and expand in these areas.

9.3 Recommendations for Further Research

- Our machine learning server as well as our dataset will be available for free use under the MIT License. This can be very useful to nutritionists and dieticians to perform data analysis and develop visualizations.
- A long term project can be to create a data lake from this information that can be employed for historical trends analysis in future.
- Research into making applications that run in browsers more lightweight and have a non intrusive way for them to communicate with the native OS will reduce dependency on browser use as a layer.

Glossary

- API: Application Programming Interface – A way to provide (address) resources
- JSON: JavaScript Object Notation
- Kanban: A work management framework based on the Agile Methodology
- Machine Learning: A way to program decision making techniques by using mathematical and statistical modelling as an algorithm.
- MVC: Model – View – Controller – A web development methodology
- ORM: ManyToMany, ManyToOne, OneToMany and OneToOne relationships
- Responsive App: Have User Interface Layouts that can resize to fit all screen sizes without compromising the design integrity of the layout elements.
- REST: Representational State Transfer. For additional information on building REST API's: https://restfulapi.net
- SDLC: Software Development Lifecycle – How a software is developed
- Flask MVC: The model has no idea about the view part, the view only focuses on how to present data and the controller connects the model and view part
- Transaction: Four key properties of transaction is ACID that stands for Atomicity, Consistency, Isolation and Durability
- UI: User Interface – The visual interface of the application that a user can see
- UX: User Experience- The overall experience of the user while using an application

References

Frank, A., & Asuncion, A. (2010). UCI machine learning repository http://archive.ics.uci.edu/ml Irvine, CA: University of California. School of Information and Computer Science, 213

Food environment atlas. Retrieved 12/03, 2017, from https://www.ers.usda.gov/dataproducts/food-environmentatlas

Fielding, R. T. (2000). REST: Architectural styles and the design of network-based software architectures. Doctoral Dissertation, University of California

Hsiao, J., & Chang, H. (2010). SmartDiet. Paper presented at the Computer-Based Medical Systems (CBMS), 2010 IEEE 23rd International Symposium on, pp. 421-425.

Kuo, M., Chen, L., & Liang, C. (2009). Building and evaluating a location-based service recommendation system with a preference adjustment mechanism. Expert Systems with Applications, 36(2), 3543-3554.

Lichman, M. (2013). UCI Machine Learning Repository http://archive.ics.uci.edu/ml Irvine, CA: University of California, School of Information and Computer Science.

Li, J., & Arandjelovic, O. (2017). Glycaemic index prediction: A pilot study of data linkage challenges and the application of machine learning. Paper presented at the Biomedical & Health Informatics (BHI), 2017 IEEE EMBS International Conference on, pp.357-360.

REST API tutorial. (2017). Retrieved 11/27, 2017, from http://www.restapitutorial.com/

Shaikh, F., Blog, G., Rizvi, M. S., and Jain, K. (2017, April 11). Comparison between Deep Learning and Machine Learning. Retrieved October 04, 2017, from https://www.analyticsvidhya.com/blog/2017/04/comparison-between-deep-learningmachine-learning/

Tanenbaum, A. S., & Van Steen, M. (2007). Distributed systems: Principles and paradigms Prentice-Hall.