

## Recommendation System for Dining

Pramila M. Chawan\* Kaushal Suvarna\*\*, Yash Goyal\*\*, Jay Thakker\*\*,  
Advait Bandiwadekar\*\*

\*Associate Professor (Department of Computer Engg. and Info. Tech. VJTI, Mumbai, Maharashtra, India.

\*\*B. Tech. Students (Department of Computer Engg. and Info. Tech., VJTI, Mumbai, Maharashtra, India.

Corresponding author : Pramila M. Chawan

### ABSTRACT

In the advent of a smartphone dependent world, applications have become an integral part of life. Existing apps are providing information which is missing a connection to get to the goal of having a good plan. Planning is one of the most time consuming activities and all existing applications focus on single services (be it a movie, or a dine-out etc.) while a complete plan requires a combination of such services. (What if we had to plan a movie AND a dine-out). Into the frame comes the Service Oriented Architecture that puts together all the needed services in a single place (application) from where users can very conveniently plan out an evening based on the recommendations provided. A weighted k-Nearest Neighbours(kNN) algorithm sorts out the recommendations based on filter parameters provided by the user (such as their location, budget etc.), while API services provided by the respective single service applications allow us to bring together multiple services on one platform to provide a complete planning experience.

**Keywords** – API, kNN, Recommendations, Service, Service Oriented Architecture,

Date of Submission:03-05-2018

Date of acceptance: 19-05-2018

### I. INTRODUCTION

How many times have we backed out on an outing just because we couldn't narrow down on a plan. We have a lot of mobile applications which answer a single or a subset of the aforementioned questions but not all. To answer all these, you would probably spend hours on already existing services like Zomato, Uber, Tapzo, Google Maps or BookMyShow etc. but indecisiveness, a characteristic, inherent in humans, leads to a lot of switching in between single service applications. Our application aims at combining a set of these services, providing a feature rich interface to set a number of filter parameters (like location, number of people, budget, time to spare etc.) and get restaurant and movie recommendations tailored to these parameters.

The following article focuses on the methodology used to bring together a set of services (namely services provided by Google, Uber, Zomato etc) to develop an all-in-one evening planner in a strictly Service Oriented Architecture. A weighted k-Nearest Neighbours algorithm decides the plan recommendations to be returned to the user based on the parameters.

### II. PREVIOUS SCENARIO

All along we have missed out on a platform which could not only bring multiple services together but also provide us with a satisfactory plan

combining them. The evolution began with menu cards being circulated around the city advertising the cuisine of the restaurants. Later on, these menus were published online for the purpose of reaching a larger audience. Applications began with reviewing and rating the restaurants. Restaurants were forced to excel in a competitive sphere to maintain a good rating. Travelling was addressed by providing online cab services making the restaurants easily accessible. The issue was that there was no common platform to integrate all these applications to make these tasks simple and accessible at a single click.

Several single service applications have addressed issues in one sphere of a complete plan (For eg. Zomato, Uber, BookMyShow) while other service oriented architecture based applications have concentrated on a different domain (For. Eg Trivago). Therefore our aim is to propose a service oriented system that brings together a set of single services to recommend a personalized and customized plan.

### III. LITERATURE SURVEY

#### 1.1 Service Oriented Architecture

The term service-oriented architecture expresses a perspective of software architecture that defines the use of loosely coupled software services to support the requirements of the business processes and software users. In an SOA environment,

resources on a network are made available as independent services that can be accessed without knowledge of their underlying platform implementation. A service-oriented architecture is not tied to a specific technology. It may be implemented using a wide range of interoperability standards, including Web Services. The key is independent services with defined interfaces that can be called to perform their tasks in a standard way, without the service having pre-knowledge of the calling application, and without the application having or needing knowledge of how the service actually performs its tasks. These services inter-operate based on a formal definition (or contract) that is independent of the underlying platform and programming language.

### 1.2 Service

A web service is a service offered by an electronic device to another electronic device, communicating with each other via the World Wide Web. In a web service, the Web technology such as HTTP—originally designed for human-to-machine communication—is utilized for machine-to-machine communication, more specifically for transferring machine-readable file formats such as XML and JSON. In practice, a web service typically provides an object-oriented web-based interface to a database server, utilized for example by another web server, or by a mobile app, that provides a user interface to the end user. We can identify two major classes of web services:

1. REST-compliant web services, in which the primary purpose of the service is to manipulate XML representations of web resources using a uniform set of “stateless” operations, and
2. Arbitrary web services, in which the service may expose an arbitrary set of operations.

### 3.3 REST Services

Representational state transfer (REST) or RESTful web services are a way of providing interoperability between computer systems on the Internet. REST-compliant Web services allow requesting systems to access and manipulate textual representations of Web resources using a uniform and predefined set of stateless operations. Other forms of Web services exist which expose their own arbitrary sets of operations such as WSDL and SOAP. In a RESTful Web service, requests made to a resource’s URI will elicit a response that may be in XML, HTML, JSON or some other defined format. The response may confirm that some alteration has been made to the stored resource, and it may provide hypertext links to other related resources or collections of resources. Using HTTP, as is most common, the kind of operations available include those predefined by the HTTP methods GET, POST, PUT, DELETE and so on. By using a stateless protocol and standard operations, REST

systems aim for fast performance, reliability, and the ability to grow, by re-using components that can be managed and updated without affecting the system as a whole, even while it is running.

### 3.4 Spring Framework

The Spring Framework is an application framework and inversion of control container for the Java platform. The framework’s core features can be used by any Java application, but there are extensions for building web applications on top of the Java EE (Enterprise Edition) platform. Although the framework does not impose any specific programming model, it has become popular in the Java community as an addition to, or even replacement for the Enterprise JavaBeans (EJB) model. The Spring Framework is open source.

### 3.5 k-Nearest Neighbours

In pattern recognition, the k-nearest neighbours algorithm (k-NN) is a non-parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space. The output depends on whether k-NN is used for classification or regression. Our methodology requires a way to classify whether to recommend a restaurant to the user or not.

In k-NN classification, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small).

k-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. Both for classification and regression, a useful technique can be to assign weight to the contributions of the neighbours, so that the nearer neighbours contribute more to the average than the more distant ones which is what we’ve done in the paper by setting priorities to the filter parameters entered by the users.

## IV. APPROACH

The application being developed is meant for public usage on a day to day basis the number of users accepting service is vast hence the system to be developed should fulfill these requirements a scalable approach using service oriented architecture has been developed and deployed.

### 4.1 Spring Framework

Spring framework was used to implement services. Spring enabled to loosely follow MVC Model in implementation of the framework with the help of annotations denoting various aspects of MVC in the code. Loose coupling was achieved using Inversion of Control. The objects gave their own dependencies instead of creating or looking for dependent objects. We made sure to use Spring Framework to make the services lightweight with

respect to size and transparency. Spring Framework itself created and managed the life cycle and configuration of application objects.

We used PostgreSQL database for storing the data. The communication between the framework and database was separated by multiple layers. We implemented Repository Classes which handled all the layers. The repositories were extension of JPA (Java persistence API).

The Java Persistence API (JPA) is a Java specification for accessing, persisting, and managing data between Java objects / classes and a relational database. JPA was defined as part of the EJB 3.0 specification as a replacement for the EJB 2 CMP Entity Beans specification. JPA is now considered the standard industry approach for Object to Relational Mapping (ORM) in the Java Industry.

We then configured Hibernate and c3p0 properties in the application property file of the spring framework. Hibernate is a high-performance Object/Relational persistence and query service, which is licensed under the open source GNU Lesser General Public License (LGPL). Hence the code was not needed to be specific to any particular RDMS. Hibernate is already implemented inside JPA. c3p0 is an easy-to-use library for making traditional JDBC drivers "enterprise-ready" by augmenting them with functionality defined by the jdbc3 spec. Its mainly used for pooling and handling the volume of interaction with Database. c3p0 was also inherently implemented.

Communication between various other service APIs was through RestTemplate package provided by the spring framework. RestTemplate makes interacting with most RESTful services a one-line incantation. And it can even bind that data to custom domain types. Performance of spring when compared with other server side implementations :

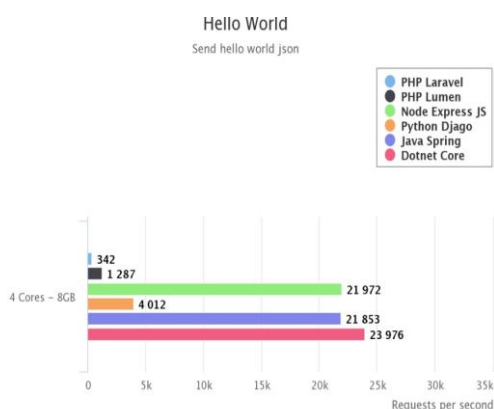


Figure 1

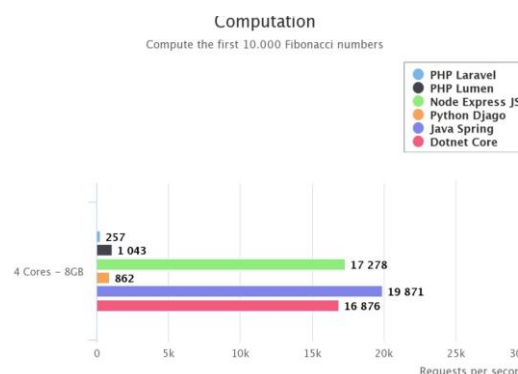


Figure 2



Figure 3

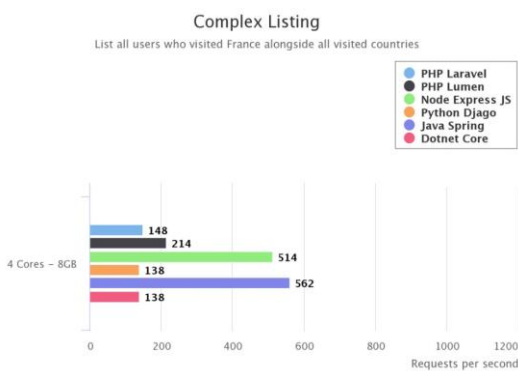


Figure 4

#### 4.2 PostgreSql

Seamless integration of the application with the postgresql database has been provided. PostgreSQL was majorly used because of it being Open Source. Over the time because of open source community postgresql has become one of the most efficient database available. For our case we used a single database instance for all the tables. The data is backed up periodically. Indexing was used to improve the efficiency of the DB. The sensitive data was hidden to prevent direct access and breach of security.

#### 4.3 Security

User security being compromised for authentication OAuth2 has been implemented to match the current standards in security.

OAuth 2.0 is the industry-standard protocol for authorization also is an open standard for access delegation, commonly used as a way for Internet users to grant websites or applications access to their information on other websites but without giving them the passwords. This mechanism is used by companies such as Amazon, Google, Facebook, Microsoft and Twitter to permit the users to share information about their accounts with third party applications or websites.

We implemented OAuth 2.0 using spring security which comes part of spring framework. We secured most of the APIs with OAuth 2.0. For the APIs which were not secured by OAuth 2.0 we made sure that they did not communicate or were connected to any secured API in any way.

#### 4.4 Android Studio

Considering the context of the application and wide popularity of smart phone app is implemented in Android Studio.

The UI provides current location and location filling with appropriate details leading to appropriate results. A layout has been created to display results.

We Implemented Google APIs and our own APIs and communicated through it using volley.

#### 4.5 Web Scraping

For necessary real time data we are using Beautiful Soup library is used along with request library to pull out the data from websites. Tag based matching and attribute extraction and data typecasting has been used to filter the data and make it available for further processing.

#### 4.6 Modified k-NN

We modified the k-nn algorithm to fit our requirements of result. Regular k-NN algorithm returns as the k-closest points to the user's specified constraints. But as the user preference is strictly constrained, we reduce the data space by bounding it by the user's preference. By doing this, we restrict the results to lie within the user constraints. We have normalized the axes of time, distance and cost. Using these axes, we can easily find out the k-closest points to the user specified data point. We have also modified the behavior of final functions according to our requirements. The graph for a normalized function:

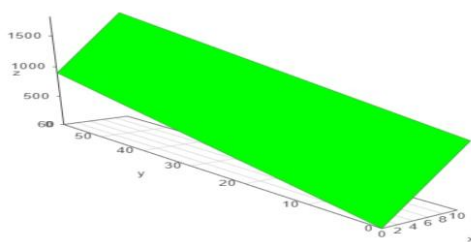


Figure 5

## V. ACQUIRED DATA

The data of all the restaurants and their details including the average cost per person, the location (i.e. the latitude and the longitude) and the cuisines have been taken from the Zomato API – a free to use API. The data is returned to the user in a JSON object, is accurate and doesn't require any preprocessing. For real time cab fares, we use the Uber API. Frequent calls to API are time intensive and hence, to reduce the overhead, a grid has been created which divides the entire region of Mumbai into hotspots from which the estimated travel costs are evaluated by computing the cost from the center of every hotspot to the center of every other. Each restaurant is allocated to a grid and hence cost to every restaurant can be calculated in O(1) time.

## VI. RESULTS AND CONCLUSION

We surveyed our Application with a hundred users – (hundred college students who are familiar with using such mobile applications) and had the following results:-

- 1.) Majority of the users found that the App significantly reduced the time to search relevant options.
  - 2.) Users found out many undiscovered places which they weren't aware of but did fit their criteria. (budget, distance, time)
  - 3.) Made planning group outings easier than before.
- Illustrated below are the results for searching a number of restaurants normally and via our App:-

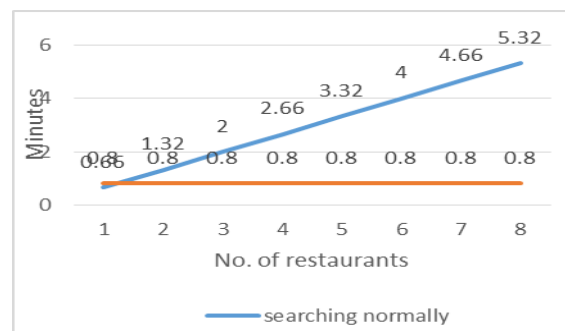


Figure 6

We haven't considered the application starting time as it will be similar in both (as good as deducting the values from both). App-switching time is considered to be negligible even though it might play a role in the user effort.

Furthermore we can improve the estimate latency by having a paid partnership with Uber while also integrating an Uber widget to allow users to book a cab directly from our application while at the same time promoting Service Oriented Architecture based applications.

Further modifications and improvements which can be implemented in the future:

1. User patterns can be analyzed and better results can be provided using content based filtering. This helps us find a relation between similar users and provide them with better results.
2. Support for advertisement pushing can benefit the recommended places as well as increase the revenue and provide better visibility.
3. Addition of movies, events, etc. in the planning phase which will result in a complete plan for the entire day. This can be done using a real time scraper on the data of a suitable website.

#### REFERENCES

##### Web Links:

- [1]. <https://developer.android.com> - Implementation of the android application.

- [2]. <https://maven.apache.org/guides/introduction/introduction-to-repositories.html> - For packages and their reference links.
- [3]. <https://spring.io/guides>
- [4]. <https://spring.io/docs> - For server side implementation.

##### Proceedings Papers:

- [1]. [http://cogprints.org/2124/3/wang\\_ICML2000.pdf](http://cogprints.org/2124/3/wang_ICML2000.pdf) - Reference for multiple instance kNN algorithm.
- [2]. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.139.2168&rep=rep1&type=pdf> Service Oriented Architecture Framework for Collaborative Services
- [3]. <http://www.joebm.com/papers/154-W00025.pdf> - A Decision Making Framework for SOA Adoption in e-Banking: A Case Study Approach