



**A Project report  
On**

## **App Market Analysis**

**Submitted by**

**USN-1PI11CS188, Vaishnavi V Bharadwaj  
USN-1PI11CS205, Yogesh Chellappa C**

**Guide**

**Asst Prof. Trupti V G  
PESIT CSE  
Bangalore-560085**

**August-December 2014**

Department of Computer Science & Engineering  
**PES INSTITUTE OF TECHNOLOGY**  
(An Autonomous Institute under VTU Belgaum)  
**100 Feet Road, BSK III Stage, Hosakerehalli, Bengaluru -  
560 085**



**PES INSTITUTE OF TECHNOLOGY**  
(An Autonomous Institute under VTU Belgaum)  
**100 Feet Road, BSK III Stage, Hosakerehalli, Bengaluru -  
560 085**

**Department of Computer Science & Engineering**

**CERTIFICATE**

Certified that the Special Topics: Mini Project work entitled **App Market Analysis** is a bonafied work carried out by **Vaishnavi V Bharadwaj (1PI11CS188)** and **Yogesh Chellappa C (1PI11CS205)** in partial fulfilment for the award of degree of Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belgaum during the academic semester August 2014 to December 2014.

**Signature of the Guide**

**Asst Prof. Trupti V G**

**Signature of the HOD**

**Prof. Nitin V Pujari**

**Vaishnavi V Bharadwaj, 1PI11CS188  
Yogesh Chellappa C, 1PI11CS205**

## ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible, and whose constant guidance and encouragement helped us in completing the project successfully. We consider it a privilege to express gratitude and respect to all those who guided us throughout the course of the completion of the project.

We would like to express our heartfelt thanks to **Prof. M. R. Doreswamy**, PESIT founder, **Prof. D. Jawahar**, CEO and **Dr. K. S. Sridhar**, Principal, for providing us with a congenial environment for carrying out the seminar.

We express our gratitude to **Prof. Nitin V. Pujari**, Head of the Department, Computer Science, PESIT, whose guidance and support has been invaluable and for including the project as part of the course.

We extend our sincere thanks to **Asst Prof. Trupti V G** and the entire laboratory faculty, for her invaluable suggestions, constant guidance, encouragement, support and invaluable advice without which this project would not have been completed.

Last, but not the least, we would like to thank our friends whose invaluable feedback helped us to improve the application by leaps and bounds, and our parents for their unending encouragement and support.

## **ABSTRACT**

Android and Apple app developers are provided with a tool to measure their app performance in comparison with the competitor apps in the market. They are provided with options to search for their respective apps and run analytics on them.

A positive score is given if the algorithm judges the competing app to be a better performer according to the statistics obtained. A negative score, if the competing app is found lacking. The input app serves as the base score.

The results from the analysis are displayed in the form a graph which makes them easily comprehensible.

# CONTENTS

Chapter title	Page Number
1. Introduction	6
2. Problem Definition	7
3. Literature Survey	8
4. Project Requirement Definition	9
5. System Requirement specification	10
6. System Design	11
7. Pseudo Code	12
8. Results Discussion	15
9. Conclusion	19
10. Further Enhancements	20
11. Bibliography	21

## **1. Introduction**

The developer of a given app is enabled to view the performance of his app in comparison with the competitor apps available on the Google Play store and the App Store. Key factors such as Rating, Downloads etc., which determine the performance, are considered for the given app and its competitors, and an algorithm is run which assigns a score to each of them. Based on this score, the results are shown to the developer in the form of a vertical bar graph.

The simple User interface and the graphical representations, make the results easily comprehensible by the users.

## **2. Problem Definition**

App market simulator and simplification - Enabling a developer to compare the performance of his apps on the Google Play Store and the Apple App Store against his direct competitors in the same app space.

### **2.1 Description**

The user is presented with a simple web user interface where the name of the app is taken as input. A web crawler is then run and the broad category under which the app is listed on the Play Store or the App Store is obtained. A list of competitor apps from the same category are pulled and an extended classification algorithm is run on this data set.

The algorithm takes in various parameters such as rating, number of user ratings, etc. of all the apps. Sentiment analysis is done on the reviews and appropriate weight-age is given to it.

Based on these, a score is assigned to the apps and the comparison is performed. The performance is represented as a vertical bar.

### **3. Literature Survey**

The following were performed as a part of literature survey

- Searched the Google Play store and the Apple App Store to note the parameters common to all apps.
- Look for appropriate ways for pulling app data from the stores.
- Studied how sentiment analysis is made on a given review.
- Decide margins on basis for pulling the initial apps for comparison with the base app.
- Devising an IPC mechanism for communicating between PHP and Python.
- Deciding the weightage given to each factor during calculation of overall score.



## **4. Project Requirement Definition**

The system should be able to do the following:

1. The system is required have access to the internet because the APIs used in the implementation pull data from the app stores.
2. The system should have access to Google charts.

## **5. System Requirement Specification**

1. Provide an easy-to-use interface for the user to enter and search for a given app.
2. Pull and display data pertaining to a given app from the appropriate app stores.
3. Run analytics for the given app by considering all parameters of the competitor apps and apply the algorithm to arrive at the scores.
4. Display the performance results in the form of a graph.

## 6. System Design

### 6.1 Database design

The schema in the database design is as follows:

Table name: apple\_mapping

<u>cat_name</u>	cat_int

This table is used to provide a mapping between the category name and the integer assigned to that category by Apple so that it can be fed to the API endpoint.

### 6.2 System design

The following were made use of for the system design:

#### 6.2.1 Front End

- Javascript
- HTML5
- CSS3
- Twitter Bootstrap
- jQuery

#### 6.2.2 Back End

- PHP
- Python
- MySQL

#### 6.2.3 Tools

- Google charts
- Sentiment Analyzer

#### 6.2.4 APIs

- 42matters.com
- playstoreapi.com
- text-processing.com/sentiment

## **7. Pseudo Code**

### **7.1 Control Flow**

The control flow of the application proceeds as follows:

1. The user selects the platform on which he wants to compare his app and the competitors.
2. The user enters his app name and the system fetches and displays the details of the app he enters.
3. The user then clicks the "Run analytics" button.
4. The system fetches competing apps, assigns each of them a score.
5. A progress bar on the results page informs the user about the various stages during calculation of the score.
6. The final results are displayed in a graphical format.

### **7.2 Graphical user interface**

The following are list of files which the user interacts with and views the results.

#### **7.2.1 search\_android.html**

This page enables the user to key in the name of the base app in Android and displays the particulars of the app entered.

#### **7.2.2 results\_android.html**

This page initially displays a progress bar and then the results in a graphical format.

#### **7.2.3 search\_apple.html**

This page enables the user to key in the name of the base app in Apple and displays the particulars of the app entered.

#### **7.2.4 results\_apple.html**

This page initially displays a progress bar and then the results in a graphical format.

## **7.3 Server-side scripts**

The following are list of files which are called during calculation of the score at the backend.

### **7.3.1 search\_my\_android\_app.php**

This page feeds the app name entered by the user to the API endpoint and parses the data returned and formats it.

### **7.3.2 run\_android\_analytics.php**

This page finds the apps in the same category as the base app and pulls recommendations from the Play Store in order to obtain the competing apps and feeds the package names to the function to calculate score and returns the results.

### **7.3.3 functions\_android.php**

This is used to calculate score for each app. The weightage for each score and overall score is determined. Sentiment analysis is also performed for each app by making a system call.

### **7.3.4 android.py**

This Python script finds the sentiment about the reviews and returns the overall sentiment from all the reviews of that app.

### **7.3.5 package.txt**

This file is used as an IPC mechanism to communicate between Python and PHP.

### **7.3.6 search\_my\_apple\_app.php**

This page feeds the app name entered by the user to the API endpoint and parses the data returned and formats it.

### **7.3.7 run\_apple\_analytics.php**

This page finds the apps in the same category as the base app and pulls recommendations from the Apple App Store in order to obtain the competing apps and feeds the package names to the function to calculate score and returns the results.

### **7.3.8 functions\_apple.php**

This is used to calculate score for each app. The weightage for each score and overall score is determined. Sentiment analysis is also performed for each app by making a system call.

#### **7.3.9 apple.py**

This Python script finds the sentiment about the reviews and returns the overall sentiment from all the reviews of that app.

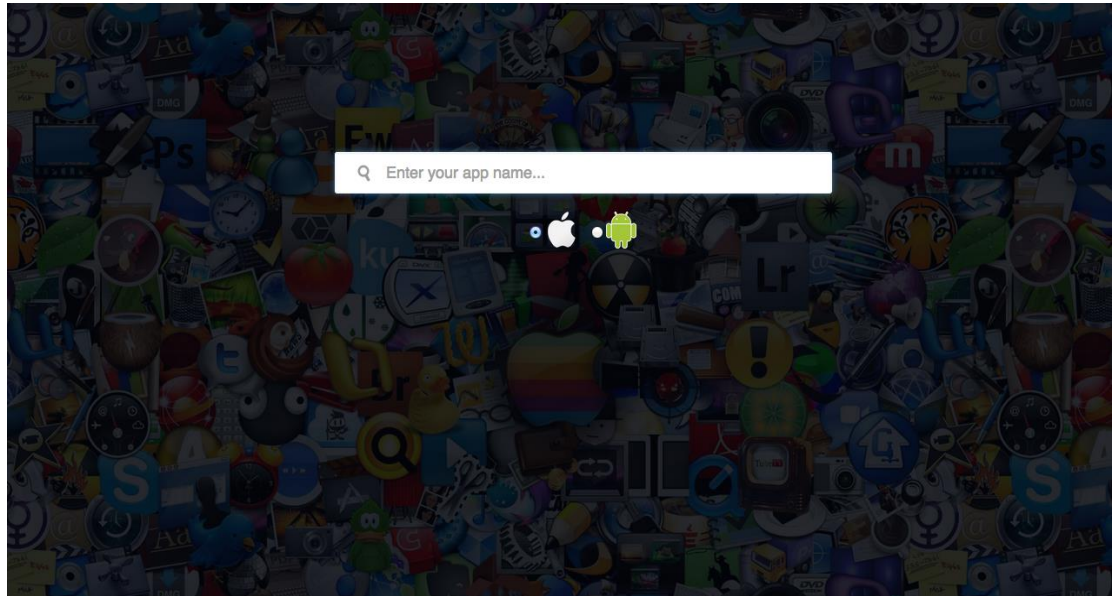
#### **7.3.10 id.txt**

This file is used as an IPC mechanism to communicate between Python and PHP.

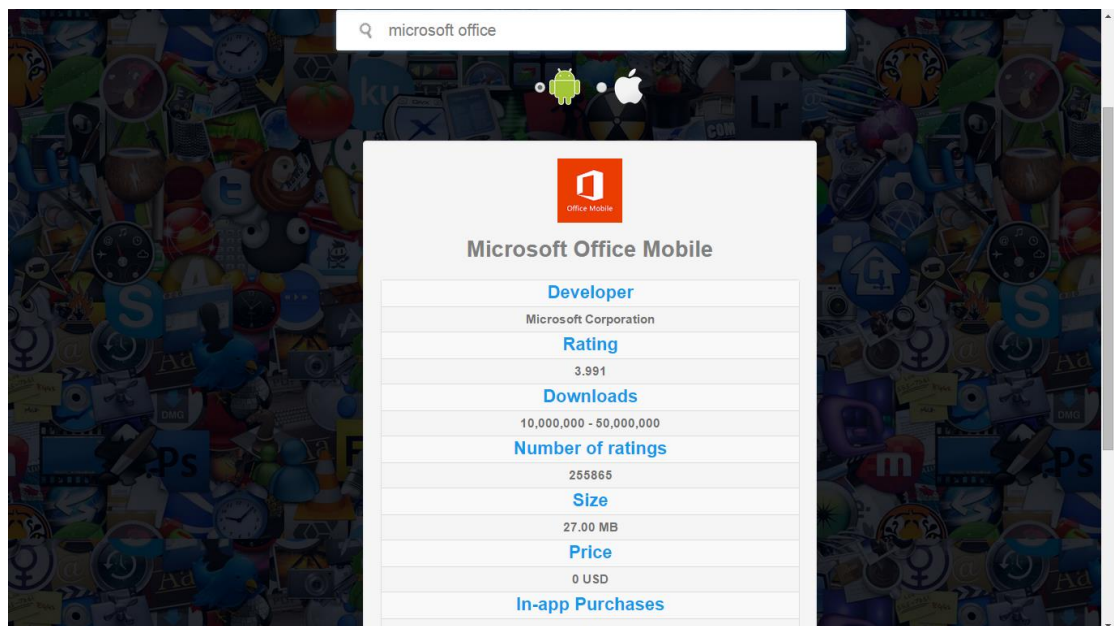
## 8. Results Discussion

### 8.1 Screenshots

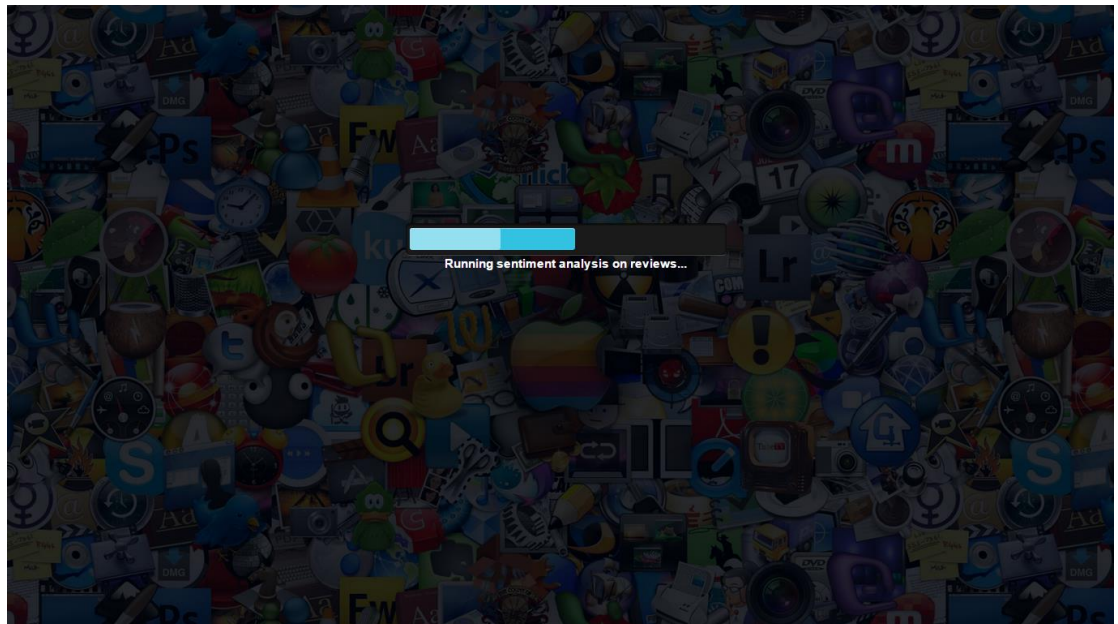
#### 8.1.1 Search based on platforms



#### 8.1.2 Search results, Android

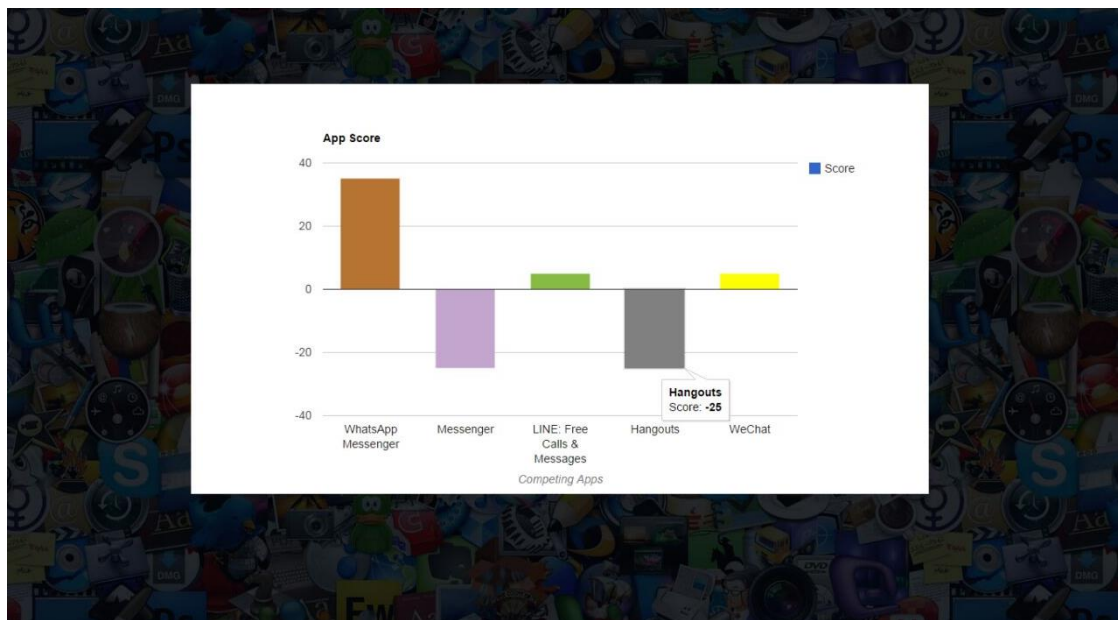


### 8.1.3 Sentiment analysis on reviews



### 8.1.4 Apple app Results

Base App: BlackBerry Messenger





### 8.1.5 Android App results

Base App: Microsoft Office



## 8.2 Weightage in calculating score

### 8.2.1 Android

Factors	Weightage
Ratings	25%
Sentiment analysis on reviews	25%
Age vs. downloads	15%
In-app-purchases	10%
Size	10%
Price	10%
Content rating	5%

### 8.2.2 Apple

Factors	Weightage
Ratings	25%
Sentiment analysis on reviews	25%
Game center support	10%
Number of languages	10%
Size	10%
Price	10%
Tablet support	5%
Content rating	5%

### 8.3 Score Calculation

- Every app in the competing space starts with an initial score of 0.
- The base app and the competing app are then compared on each factor.
- If the competing app trumps the base app in a factor, it is awarded a score based on the weightage given to that factor.
- If the competing app loses to the base app in a factor, it is deducted a score based on the weightage given to that factor.
- Thus, the final score obtained can be positive or negative or zero.
- The score is positive if the competing app outperforms the base app overall and is negative if it is outperformed.

## **9. Conclusion**

- The application fulfills all the requirements as specified in the SRS and once an app name is entered, obtains apps in the competing app space and assigns each app a score based on a number of factors.
- Performs sentiment analysis on the app reviews so as to accurately gauge user reaction and ranks the results in an easily comprehensible graph.
- App Developers can successfully know how their app is performing in the market (Play store or App store).
- They are made known of the competitor developers and the stand of their apps with respect to others.
- An insight is provided as to which kind of apps the customers prefer based on the reviews.
- Based on these results, they can further take measures in order to popularize their own apps.

## **10. Further Enhancements**

Some of the enhancements that can be provided to the current functionality are:

- Provide recommendations to the user in areas where the application can be optimized so as to obtain a higher score in the algorithm.
- Extend the present analysis to support Windows app store.
- Improve performance of the system by making it faster and more efficient.
- Enabling apps to be compared across platforms and analysing the forthcoming of the app in across platforms.

## **11. Bibliography**

- [1] [www.google.com/charts](http://www.google.com/charts)  
For data visualization and graph rendering.
- [2] [www.42matters.com](http://www.42matters.com)  
Providing APIs to pull data from the Google Play Store.
- [3] [www.playstoreapi.com](http://www.playstoreapi.com)  
Providing app recommendations to obtain competing apps.
- [4] <http://text-processing.com/api/sentiment/>  
Running sentiment analysis on reviews.