



APPMETER

SCU CAPSTONE PROJECT

Amarjit Dhal | Ameya Ghatpande | Chairali M Naik | Omkar Gokhale | Shalini Gopalkrishnan
Under the guidance of: Dr. Yasin Ceran

Table of Contents

1. Introduction.....	0
1.1 Problem Statement and Business Need	0
1.2 Business Opportunities	1
1.3 Proposed Solution	1
1.4 Key Stakeholders.....	2
1.4.1 Application Owners	2
1.4.2 User.....	2
1.4.3 AppMeter Admin	2
2. Initial Research.....	3
3 Competitor Analysis	5
3.1 AppAnnie ..	5
3.2 Flurry/Mixpanel	6
3.3 Priori Data	6
4 Dependencies, Assumptions, Risks and Deliverables	7
4.1 Data dependencies	7
4.2 Technical Dependencies	9
4.4 Risks	9
4.4.1 Data Risks	9
4.4.2 Other risks	10
4.5 Deliverables	10
5. Project Planning	11
5.1 Team and Responsibilities.....	11
5.2 Timeline	13
5.3 Weekly Team Meeting Schedule	16
6 Requirement Definition	17
6.1 Functional Requirement.....	17
6.1.1 Process Oriented	17
6.1.2 Information Oriented	18
6.2 Non-Functional Requirement.....	18
6.2.1 Operational.....	18
6.2.2 Security	18
6.2.3 Performance.....	18

6.2.4 Cultural	18
7 Use Case.....	19
8. Data Flow Diagrams	26
9. Program Chart.....	31
10. Process Flow Diagram	32
11 Technology Stack.....	33
12. Test Cases	35
13. Data Dictionary	40
14. Machine Learning Algorithm researched and used	42
14.1 Linear Regression (OLS)	42
14.2 Decision Tree Regression	44
14.3 2SLS	45
14.4 Principal Component Analysis	46
15 Code Snippets for Data Cleaning	48
16 Web Application User Interface	51
16.1 AppMeter Home Page:	51
16.1.1 What we do Section	53
16.1.2 Pricing	53
16.1.3 Team	54
16.1.4 Contact Us	55
16.2 Register	55
16.3 Login	57
16.4 Find your app score	58
16.5 Predict.....	64
16.6 Analyze:	67
17 Salient Features of the Web Application UI	70
17.1 Session Management and Security	70
17.2 Responsive Website Design.....	71
18 Challenges and Learnings	71
18.1 Challenges	71
18.2 Learning	72
19. Future scope	72

ACKNOWLEDGEMENT

We would like to express our deepest appreciation to all those who provided us the support to complete this project. Firstly, we would like to convey our sincere thanks to our advisor and mentor, Professor Dr. Yasin Ceran, for his expert guidance and constant supervision throughout this design project. We are highly indebted to him for providing his invaluable inputs for the development of this challenging system. He was kind enough to spare us his valuable time and to help us without any hesitation.

We would like to thank Professor Vasu Kadambi, our capstone advisor for accepting our capstone design project proposal for SCU. His motivation created us the opportunity to explore topics of our interest and we are extremely grateful to him. The valuable insights from his rich and varied experience made us think of new dimensions in deploying the techniques for our project.

Furthermore, we would also like to express our special gratitude to AppMonsta for providing the data that we used in our system, which helped to drive this project forward.

We would like to convey our sincere thanks to all our Professors for investing their effort and time in guiding and teaching us throughout the program. Last but not the least, we are grateful to our institution, Santa Clara University Leavey School of Business for nourishing us with knowledge and for offering us a warm welcome every day.

Thank You

Team AppMeter

1. Introduction

Wireless communication is one of the most helpful technological innovations of recent times. Once considered to be a rich man's toy, now mobile phones have become a necessity. We rely on mobile phones, not just for communication but also in our day to day activities like GPS navigation, bank transactions, social connect, stock trading, weather information, news updates and many others. With technological advancements and growing user base, there is a huge rise in the number of mobile Apps. The App industry is becoming highly competitive, so it is important for organization to be on top of their game. While an app rating gives a generic standing of an app, this is often misleading and there are various other parameters which decide its success. To help answer these complex relationships, there are many app analytics platforms providing several insights, but it will be convenient for the busy executives to have a system which can combine multiple data points to give one comprehensive score.

Our team was successfully able to draw upon our collective experiences in the program to analyze, design and create a comprehensive app analytics system that provides a single uniform score defining the position of an app in the app ecosystem. Our system provides three major functionalities:

- Checking the score of an existing application.
- Predicting the success of an existing app/ to-be-launched application.
- Providing an all-inclusive analytics dashboard to give an overview about the app ecosystem.

1.1 Problem Statement and Business Need

Mobile application is one of the most competitive industry in the tech business. Being innovative and launching an App ahead of the competitors is one of the most critical challenges that any organization faces. Trends are continuously changing in the market and it is a challenge for any organization to stay put with these changes. Organizations need to take timely actions to fuel growth in sales as well as minimize loss. Product managers

and marketing team always look out for these trends and analyze how their app is faring in the market. They would like to have a platform which can predict the success of the app.

With plenty of options to choose from, the user is also not sure which is the best app for their needs. The best app is not just defined by its rating, but has multiple parameters which affects consumer behavior.

1.2 Business Opportunities

The mobile app market is growing faster. The revenue generated by the global mobile app industry has skyrocketed. According to Statista, gross annual revenue is projected to exceed \$189 billion by 2020¹. Despite increasing strong competition in the industry, developers are applying new monetization methods and creating more interesting and useful solutions for users. Organization's reliance on analytics for business growth is increasing in the recent years, so this is a perfect opportunity for a new and innovative app analytics platform. A platform which can analyze different data points of an app and combine them to give a comprehensive score is the need of the day. So, this is a good business opportunity for a solution like AppMeter.

1.3 Proposed Solution

- **Score:** As the name suggests, AppMeter generates one comprehensive score for each app, which encompasses multiple data points. It also provides a great search functionality, which help users to analyze various details of any app. They can view the AppMeter score and the rank of that app in its genre. Since different genres have different target audiences and business models, it is important to categorize apps as per their genre. After researching on multiple regression algorithms, we zeroed on Principal Component Analysis (PCA) which considers several aspects of an app, like number of downloads, average rating, number of raters, price and the age of the app.

¹ <https://www.statista.com/statistics/269025/worldwide-mobile-app-revenue-forecast/>

- **Predict:** AppMeter provides predictive analytics capabilities to app owners. They can use our system to predict the number of downloads an app will reach in the specific time frame. This is a great feature which can help teams to plan release, add functionalities or set the price. This prediction function is calculated using 2SLS algorithm that considers two endogenous variables: total number of downloads and number of raters.
- **Analyze:** AppMeter provides analytics platform to analyze data from multiple data points in one place. The app owners can leverage the insights that are helpful in developing business strategies.

1.4 Key Stakeholders

1.4.1 Application Owners

The app development team needs to understand how their app is performing, and where they stand in the market. The application owners should also be able to identify what parameters the team should focus on, to make their app successful. Our platform helps them to strategize their next release or to build a new application.

1.4.2 User

With plethora of options in the market, the end user should select the right app suitable for their needs. Although user rating is one of the important parameters to evaluate an app, this is not the sole parameter that determines its value. The user can search an app and view its score and other relevant parameters which help him decide which app to use.

1.4.3 AppMeter Admin

The admin is one of the key stakeholders of any system who controls the system infrastructure and environment. They should be able to publish new content to the webpage, update database and add/remove users from the backend. Admin should also be able to modify and dictate AppMeter business model to the customers.

2. Initial Research

Our initial research was about understanding the App store and the corresponding data available in the App store. We also researched few papers about App Analytics and what makes an app successful. Below are the details of the literature which was evaluated:

We started our research with sentiment/twitter analysis. We researched papers which explained feature and keyword analysis in depth.

Research Paper Name & Author	Description
How Do Users Like This Feature? A Fine-Grained Sentiment Analysis of App Reviews - by <i>Emitza Guzman and Walid Maalej</i> ²	This paper explains how to identify the fine-grained app features using NLTK. It also tells about extracting the user sentiments about the identified features and giving them a general score across all reviews.
AR-Miner: Mining Informative Reviews for Developers from Mobile App Marketplace - by <i>Ning Chen, Jialiu Lin</i> ³ :	This paper focuses on extracting informative user reviews by filtering noisy and irrelevant ones, and grouping the reviews automatically using machine learning.

Our scope of the project then changed from user sentiment analysis to creating a score, and predicting app success, and hence we moved our focus to machine learning algorithm analysis and its relevant research

² <http://ieeexplore.ieee.org/document/6912257/?reload=true>

³ http://ink.library.smu.edu.sg/cgi/viewcontent.cgi?article=3323&context=sis_research

Research Paper Name & Author	Description
PRINCIPAL COMPONENTS ANALYSIS (PCA) by <i>Steven M. Holland</i> ⁴	This is an important reference paper which provides the basics of PCA. It explains how to formulate the equations using R and how to interpret the results.
An Examination of the Current Rating System used in Mobile App Stores: by - <i>Israel J. Mojica Ruiz, Meiyappan Nagappan, Bram Adams, Thorsten Berger, Steffen Dienst and Ahmed E. Hassa</i> ⁵	It discusses the effect of subsequent releases of an app, disadvantages of the current app rating system and the advantages of Version based App score.

We also studied the research papers about Mobile App analytics and its success parameters.

Research Paper Name & Author	Description
Understanding the Determinants of Success in Mobile Apps Markets by <i>Gun Woong Lee</i> ⁶	<p>This paper studied mainly 3 parameters.</p> <ul style="list-style-type: none"> • App Developers portfolio - How the App developer's portfolio of Apps influence in the App downloads and its success. • App Description - Considering there are many Apps which provides similar features and usability, the App description plays an important role in attracting consumers. Also with limited character set and smaller mobile screens, strategic words play an important role in success of App • App Quality updates decision - It is observed that the success of the App is dependent upon the continuous effort of the App developers in releasing timely updates

⁴ <http://strata.uga.edu/software/pdf/pcaTutorial.pdf>

⁵ <http://ieeexplore.ieee.org/document/7045413/>

⁶ https://repository.asu.edu/attachments/150636/content/Lee_asu_0010E_14861.pdf

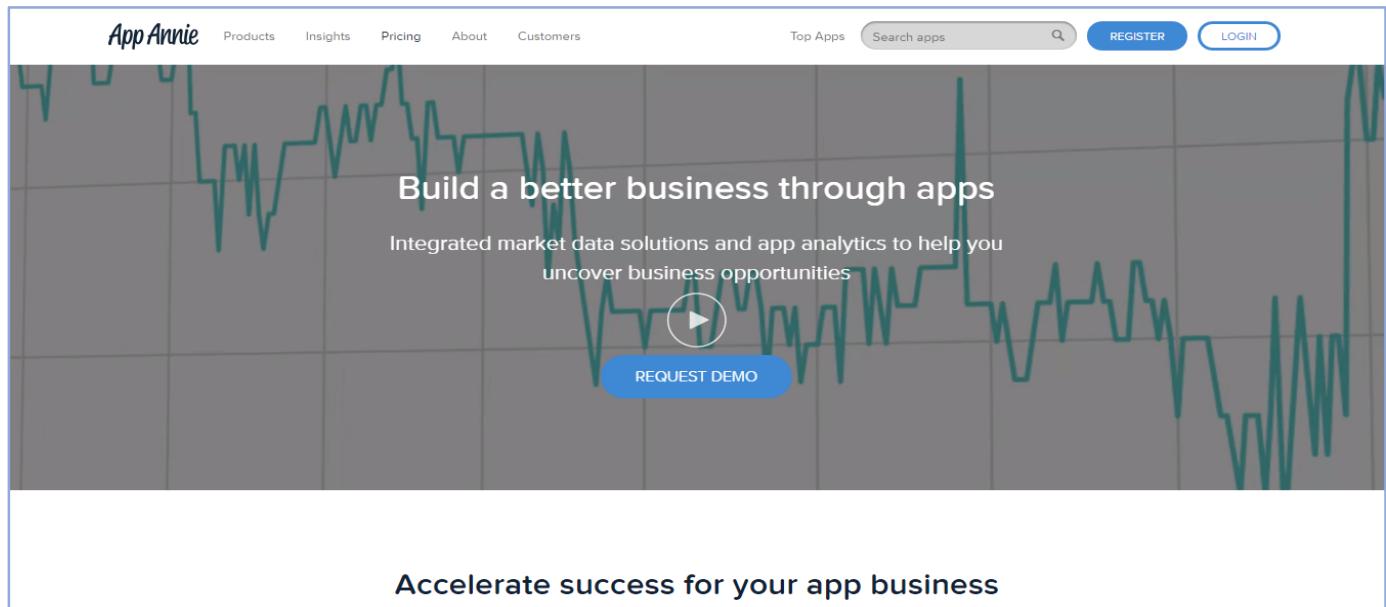
Our market research involved studying the existing mobile Apps analytical products. Even though we studied the different features and functionalities provided by these vendors, AppMeter is different and unique. Studying the existing systems also enabled us to understand its shortcomings/disadvantages; and ways to improve AppMeter.

3 Competitor Analysis

3.1 AppAnnie⁷

AppAnnie is probably the leader in the app analytics platform. It consists of three main products: Store Stats, Analytics, and Intelligence which provides the most accurate market estimates available. It provides data analytics and insights to the app owners to strategize their business

How we are different: Our platform keeps things simple. Unlike AppAnnie, which specializes in giving various insights, and focuses on multiple data points, we have considered multiple parameters to come up with one comprehensive score.

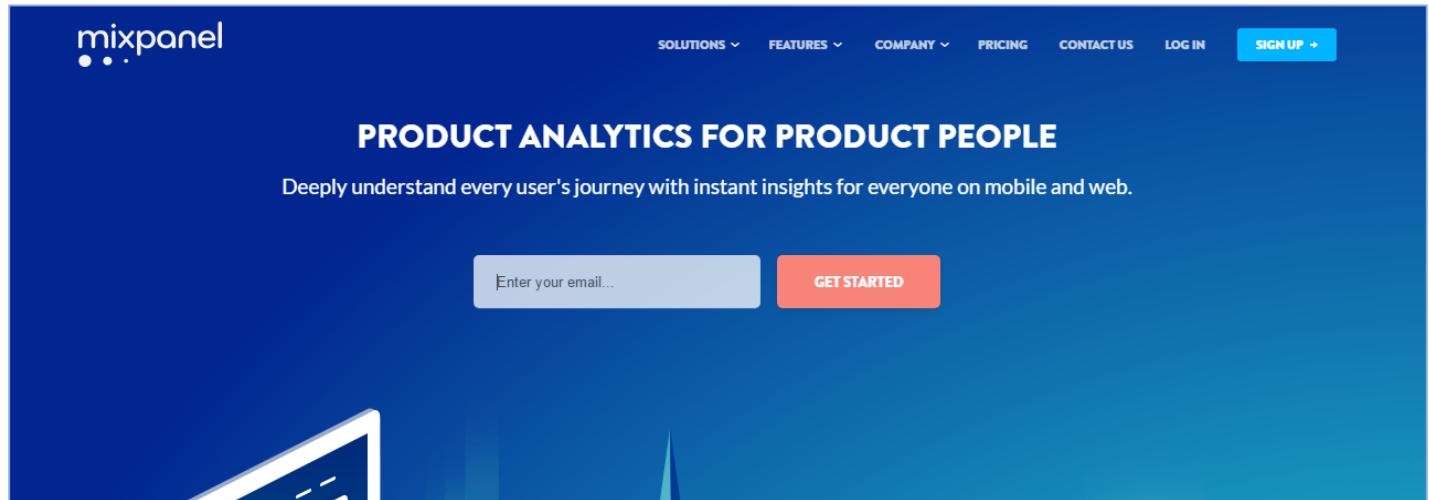


⁷ <https://www.appannie.com/en/>

3.2 Flurry/Mixpanel⁸

Flurry and Mixpanel both provides a very granular look into the key metrics surrounding user engagement for a developer's own apps. Both focus basically into user behavior and customer engagement.

How we are different: Our platform considers the word of mouth factor, which considers the ratings an app received over a period. Unlike average rating, this parameter considers the age of an app and that helps our algorithm to differentiate a *highly rated app*, and a *highly rated app over the time*.



3.3 Priori Data⁹

Companies of any size can use Priori's platform to track the performance of their mobile competitors and explore market opportunities across the globe. Prior Data also specializes on keyword search.

How are we different: We provide competitive analytics on specific genre.

⁸ <https://mixpanel.com/>

⁹ <https://prioridata.com>

The Prioridata homepage features a prominent teal wave graphic at the bottom. A search bar with the placeholder "Research any app for free" is positioned on the wave. Below the search bar, a row of popular search icons includes Instagram, Facebook, WhatsApp, and others. At the top, the Prioridata logo is on the left, followed by navigation links: Products, Solutions, About, Data Partners, Pricing, and Blog.

4 Dependencies, Assumptions, Risks and Deliverables

4.1 Data dependencies

AppMeter project requires mobile App data for existing Apps. The scope of this project is restricted to Android Apps. Each Android App has the following data/attributes associated with it.

Attribute Name	Description
App Name	Name given to the App by the owners
Bundle ID	Unique ID to identify the App
App Description	A brief description about the functionality of the App
Genre	Which category does the App fall into
Release date	The date when the first version of the App was released
File Size	Size of the App
Price	Free/Paid (If paid then its price is in \$)
Downloads	Number of downloads till date
Rating	Average rating till date (0- low, 5 - maximum)
Raters	Number of users who rated the App

Why this data?

The metadata about an app is very important in deriving the KPIs for its performance or it can be a set of indicators for the success of the app. In the world of mobile applications, app download is considered as sales. So, if we consider downloads as a metric, the following can be good indicators to consider.

- **Average Rating:** It is general tendency of user to check the average rating of an app before downloading it. A highly rated App attracts more users as compared to a low rated app. So, the average rating plays an important role in determining the downloads of an app.
- **Price:** Considering 80% of all mobile Apps are free, price becomes an important factor for user. Users tend to think twice before downloading a paid app. Expectations can rise significantly for a high-priced app. So, price becomes an important parameter while considering App success.
- **Number of raters:** Average rating depends upon the number of raters and their corresponding rating. If there are less number of raters, then the average rating does not accurately reflect the quality of App. So, number of raters is also an important parameter
- **File Size:** If an App has a bigger file size, the user will be reluctant to download it. Whereas a smaller file size app will be of less concern. Though not very important, this is also a deciding factor in the success of the app.
- **Genre:** Every app falls under a specific category/genre, which has its target audience. Some genre tends to have more downloads as compared to others. Apps for 'Social Networking' and 'Games' genre have maximum downloads whereas 'News' or 'Business' genre have less than average downloads. So, Genre of the app is also one of the major indicators of app downloads.
- **Age:** Age of an app is an important parameter which quantifies the app.

4.2 Technical Dependencies

- Connectivity to the internet as the AppMeter system is web hosted.
- Availability of the 3rd party web server (AwardSpace).
- Availability of Power BI analytics thin client.
- Availability of MySQL database hosted on AwardSpace.

4.3 Project Assumptions

- The AppMeter system will be implemented by June 10th, 2017
- The AppMeter system is a capstone project and not a commercial software
- The AppMeter system algorithms are run on data for the day of **07 January 2013**. These algorithms will hold true for any other timeline.
- The AppMeter system considers the price for paid Apps in dollars
- The AppMeter system scores all existing Apps out of 10
- The AppMeter system combines similar genres together and reducing the number of genres to 20

4.4 Risks

4.4.1 Data Risks

The data for the Apps has been received from AppMonsta commercial App data provider. As per the SLA, snapshot of all the available Android Apps data for the given date of "**January 07, 2013**" has been provided for this project. All the attributes for any given Android App will be for this date. The scope of the project is restricted to

- The Apps data supplied by AppMonsta vendor, therefore the reliability of the data is solely based on the discretion of the vendor.
- The accuracy cannot be fully verified for all the data and is assumed to be correct
- Prediction model and scoring criteria will be based on data as of 07 January 2013

4.4.2 Other risks

- Currently we are using free version of Power BI, whose capabilities and support are a risk.
- MySQL database, web server load and response time are dependent on the SLA of AwardSpace.
- Since this web application is not a commercial website, we have not integrated SSL encryption feature in our system.

4.5 Deliverables

- GitHub repository link pointing to the entire source code.
- The original Android App Data as received from AppMonsta.
- A detailed Project Report.
- A detailed project presentation and demo.

5. Project Planning

5.1 Team and Responsibilities

Name	Role	Responsibilities
Ameya Ghatpande	Team member and Project lead	<ul style="list-style-type: none">Formed the idea for the project and contributed in developing machine learning algorithms for prediction.Led the project coordination and task allocation.Requirement specification, Data Exploration, Transformation and Project monitoring and management.Research on data models. Code management.
Amarjit Dhal	Team member and backend engineer	<ul style="list-style-type: none">Formed the idea for the project and contributed in developing machine learning algorithms for scoring an App.Requirement definition, Data gathering.Data research, Data cleaning.Execution of Data model for AppMeter score functionality.
Chaitrali Naik	Team member and front-end engineer	<ul style="list-style-type: none">Designed and developed the UI functionality for Prediction webpage.Integration of Prediction charts into AppMeter web UI using external JavaScript LibrariesData exploration, Data cleaning and transformations.Executing data models like regression, Principle component Analysis (PCA) on the cleaned data.

Omkar Gokhale	Team member and Analytics engineer	<ul style="list-style-type: none"> ● Designed and Developed the Analytics Dashboard in Microsoft Power BI. ● Integration of Analytics Dashboard hosted by Power BI in AppMeter System. ● Designed and developed the web application UI including Home page and User Login functionality. ● Database Server Setup and Analytics dashboard web UI.
Shalini Gopalkrishnan	Team member and front-end engineer	<ul style="list-style-type: none"> ● Designed and developed the UI functionality for User Registration, App Search and Score page. ● Management of System Security using Access Control and Session Management techniques. ● Data Exploration, use-case detailing and Research on Data models. ● Data loading on to database server.
Yasin Ceran	Project Advisor	<ul style="list-style-type: none"> ● Gave appropriate guidance for the direction of the project and frequent feedback on the progress made. ● Provided research papers and resources for further study of concept and encouraged to learn, try and implement the techniques. ● Helped the team understand implementation and interpretation of the data models; specially 2SLS data algorithm.

5.2 Timeline

Once the scope of the project was finalized, we created a detailed project timeline. The team followed the realistic timeline under the guidance of our advisor, Professor Ceran and adhered to the schedule for the important milestones. This allowed the team to make good timely progress and be on schedule for the project delivery.

Phase	Activities	Duration	Actual	Duration (in days)
Initiation	<ul style="list-style-type: none">● Brainstorm the AppMeter Idea● Study existing market/Competitors	Start date: 12/12/2016 End date: 12/23/2016	Start date: 12/12/2016 End date: 12/23/2016	10
Team formation	<ul style="list-style-type: none">● First team meeting● Ice breaking session	Start date: 12/26/2016 End date: 12/26/2016	Start date: 12/26/2016 End date: 12/26/2016	1
AppMeter Concept creation	<ul style="list-style-type: none">● Identify the Scope● Define the Assumptions● System understanding	Start date: 01/02/2017 End date: 01/13/2017	Start date: 01/02/2017 End date: 01/17/2017	12

AppMeter Data Search	<ul style="list-style-type: none"> • Identify the Data source • Identify the parameters required for each App • Identify the platform of Apps 	Start date: 01/16/2017 End date: 01/30/2017	Start date: 01/18/2017 End date: 01/30/2017	11
Requirement Analysis	<ul style="list-style-type: none"> • User requirements • System requirements • Functional requirements • Performance requirements 	Start date: 01/31/2017 End date: 02/17/2017	Start date: 01/31/2017 End date: 02/20/2017	15
Design	<ul style="list-style-type: none"> • Create Use cases • Create Context diagram • Create Data flow diagrams • Create program chart • Create data dictionaries 	Start date: 02/21/2017 End date: 03/15/2017	Start date: 01/21/2017 End date: 03/15/2017	17
Data preparation	<ul style="list-style-type: none"> • Data cleaning • Data wrangling • Data conversion • Data upload 	Start date: 03/16/2017 End date: 03/24/2017	Start date: 03/16/2017 End date: 03/24/2017	07

Code upload and version control	<ul style="list-style-type: none"> ● GitHub Repository creation ● Code upload on GitHub 	Start date: 03/25/2017 End date: 04/27/2017	Start date: 03/25/2017 End date: 04/27/2017	02
Implementation	<ul style="list-style-type: none"> ● Run the machine learning algorithms for creating the AppMeter scoring page, AppMeter success prediction and Analytics Dashboard. ● Design and Implement the web pages for Important modules of the project 	Start date: 03/28/2017 End date: 04/22/2017	Start date: 03/28/2017 End date: 04/22/2017	24
System verification	<ul style="list-style-type: none"> ● Create test plan (test cases) ● Unit testing ● Integration testing ● System testing 	Start date: 04/23/2017 End date: 05/18/2017	Start date: 04/28/2017 End date: 05/18/2017	20
System deployment	<ul style="list-style-type: none"> ● Deploy the website on a commercial domain ● Test the Application for availability and integration 	Start date: 05/19/2017 End date: 05/23/2017	Start date: 05/19/2017 End date: 05/23/2017	04

Project documentation	<ul style="list-style-type: none"> ● Project report ● Create a Demo ● Upload code on GitHub 	Start date: 05/23/2017 End date: 06/06/2017	Start date: 05/23/2017 End date: 06/06/2017	13
Project Delivery	<ul style="list-style-type: none"> ● Final Demo ● Submission of report and Github 	Start date: 06/08/2017 End date: 06/08/2017	Start date: 06/08/2017 End date: 06/08/2017	01

5.3 Weekly Team Meeting Schedule

Day	Timings	Platform
Monday	7 pm to 10 pm	Lucas Hall
Tuesday	3 pm to 5 pm	Lucas Hall
Thursday	2 pm to 3 pm 3 pm to 4 pm (Meeting professor Yasin Ceran)	Lucas Hall
Friday	6 pm to 8 pm	Lucas Hall
Sunday	11 am to 12 pm	Google hangout/ conference call

6 Requirement Definition .

6.1 Functional Requirement

6.1.1 Process Oriented

a) User Login

- The system should allow the existing users to login using their username and password.
- The system should allow new users to sign up or register by entering their details.
- The system should prompt the new user to enter the details such as first name, last name, username, email address and password.
- The system should throw an error if the username is already registered or the password requirements are not met.

b) Search

- The system should allow guest users and registered users to access the search functionality
- The system should allow the users to filter results based on genre, price and average rating
- The system should allow users to view the score and rank
- The system should display the metadata for the searched app like - rating, downloads, price etc.
- The system should show the listing of top ten apps in the selected app genre

c) Analytics

- The dashboard should be dynamic and should reflect visualization as per user filters
- The system should allow registered users to view generic analytics of all Apps
- The system should allow registered users to view filtered analytics as per selected Genre
- The system should allow registered users to view filtered analytics as per Paid/Free Apps
- The system should allow registered users to view filtered analytics as per rating buckets

d) Prediction

- The system should allow registered users to enter values for price, file size, average rating, genre and age.

- The system should validate all the user inputs for prediction
- The system should allow registered users to predict the success (downloads) for given inputs
- The system should allow registered users to compare the success between two sets of inputs
- The system should allow registered to view the success prediction chart

6.1.2 Information Oriented

- The system should contain App data
- The system should contain registered user's data who will have higher privileges.

6.2 Non-Functional Requirement

6.2.1 Operational

- The system should be a web based application
- The system should be supported by multiple web browsers
- The system should be viewed on Mobile devices

6.2.2 Security

- The system should keep all the App and user data private and confidential
- The system should provide proper access control to different types of users.

6.2.3 Performance

- The system should be up and running 24 hours a day
- The system should be able to search any App less than 3 seconds
- The system should support at least 300 users at any given time

6.2.4 Cultural

- The system should support multilingual App names and App description
- The system should not be biased among individuals based on their race/religion/gender

7 Use Case

Use Case Name: New user Registration	ID: UC 1	Importance Level: High	
Primary Actor: Guest user			
Short Description: This describes how a user will register inside the system			
Trigger: Guest user enters the system and wants to register himself (or) Guest user tries to access the premium content			
Type: <i>External</i>			
Preconditions: 1. System is available and online 2. User data store is up and running			
Major Inputs:		Major Outputs:	
Description	Source	Description	Destination
Register New User Page	System	Confirmation page	Guest User
Username	Guest User		
Password	Guest User		

Major Steps Performed	Information for Steps
Normal course: 1.0	
1. User is provided with the Registration page 2. User enters the registration details and clicks submit button 3. The system displays registration confirmation message	Register New User Page Username, Password Confirmation page
Alternative course 2.0: User registration is unsuccessful (Branch at step 3)	
1. User is prompted to provide the registration information again 2. User re-enters the info and submits. 3. The system displays registration confirmation message	Register New User Page Username, Password Confirmation page

Use Case Name: User login	ID: UC 2	Importance Level: High	
Primary Actor: Registered user			
Short Description: This describes how a user logs into the system OR Guest user tries to access the premium content			
Trigger: Registered user tries to log in to his account,			
Type: External			
Preconditions:			
1. System is available and online 2. User data store is up and running			
Major Inputs:	Major Outputs:		
Description	Source	Description	Destination

Login page	System	Registered user home page	Registered User
Username	Registered User		
Password	Registered User		
Major Steps Performed			Information for Steps
<p><i>Normal course: 1.0</i></p> <p>1. User is provided with the Login page</p> <p>2. User enters the login details and clicks submit button</p> <p>3. The system authenticates the user and displays the registered user home page.</p>			<p>Login page</p> <p>Username, Password</p> <p>Registered user home page</p>
<p><i>Alternative course 2.0:</i></p> <p>User login is unsuccessful (Branch at step 3)</p> <p>1. User is prompted to provide the correct login credentials</p> <p>2. User re-enters the info and submits.</p> <p>3. The system authenticates the user and displays the registered user home page.</p>			<p>Login page</p> <p>Username, Password</p> <p>Registered user home page</p>

Use Case Name: Find App Score	ID: UC 3	Importance Level: High	
Primary Actor: Guest user / Registered user			
Short Description: This describes how a user can view the App Score			
Trigger: User searches for an app by entering an App Name			
Type: <i>External</i>			
Preconditions			
1. System is available and online 2. The Android data store is up and running			
Major Inputs:		Major Outputs:	
Description	Source	Description	Destination
Search page Search keyword and filters Selected app	System Guest/ Registered User	Search results Score page	Guest/ Registered User
Major Steps Performed		Information for Steps Search page Search keyword and filters Search results Selected app Score page	
1. User enters an App Name and/or selects the filters for genre, price and rating. 2. System displays the list of apps matching the search criteria entered 3. The user selects the desired App from the list of apps 4. The system displays the score and other information related to the App like File Size, Average rating, Price etc. along with the top ten apps in the selected app's genre			

Use Case Name: Predict App downloads	ID: UC 4	Importance Level: High	
Primary Actor: Registered user			
Short Description: This describes how a registered user can predict downloads for a new App			
Trigger: Registered User enters the necessary information about his App.			
Type: <i>External</i>			
Preconditions:			
Registered user should be logged into the system			
Major Inputs:		Major Outputs:	
Description	Source	Description	Destination
App Genre	Registered User	Predicted downloads	Registered User
App Price	Registered User	Prediction chart	Registered User
App File size	Registered User		
App Average Rating	Registered User		
App Age	Registered User		

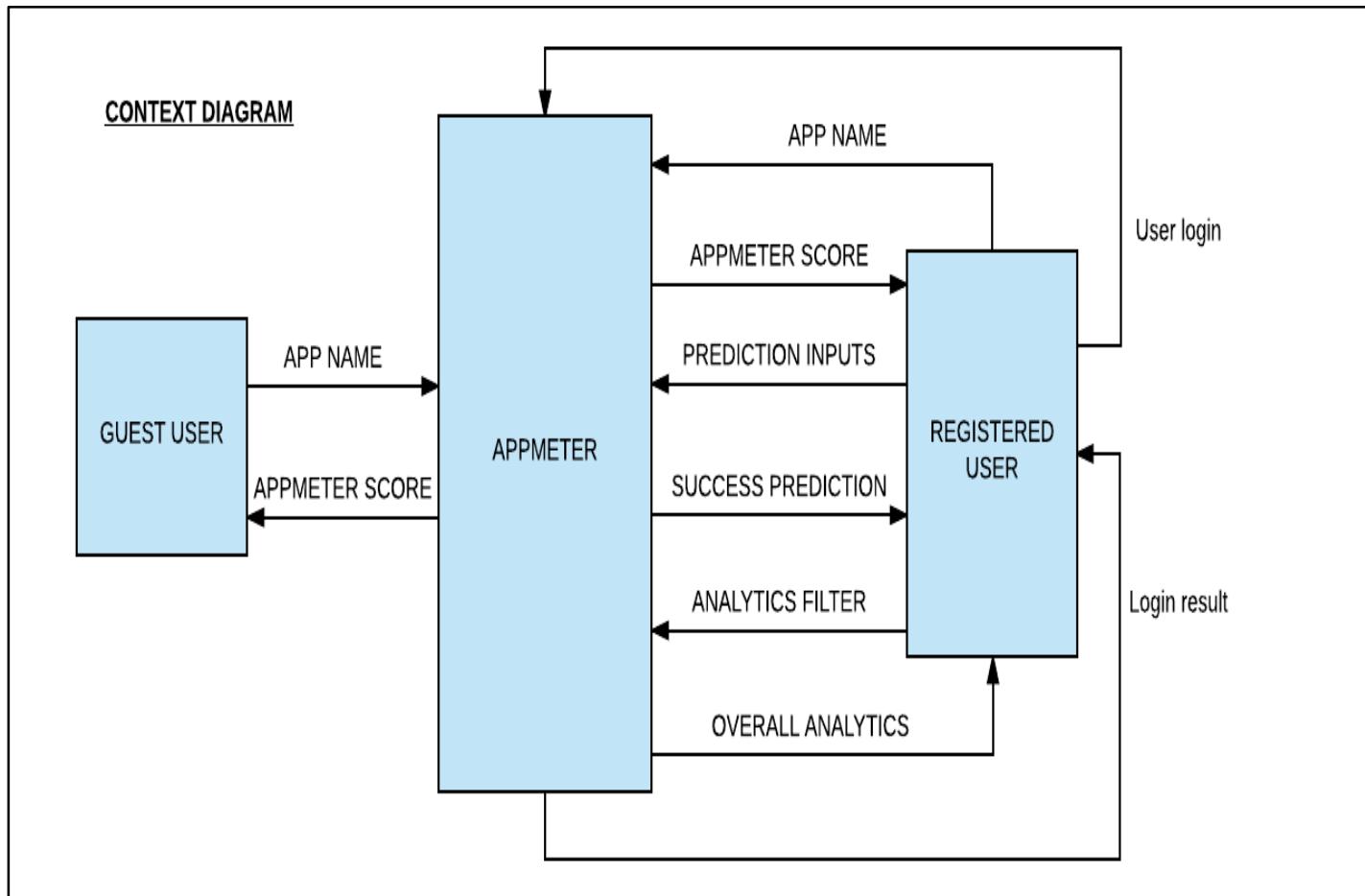
Major Steps Performed	Information for Steps
1. User is provided with the predict page.	Predict Page
2. User selects the Genre.	App Genre
3. User enters App Price.	App Price
4. User enters App File size.	App File size
5. User enters Average rating.	App Average rating
6. User enters App Age. This is the time after the release of the app at which user wants to predict the number of downloads.	App Age
7. The system will display the predicted downloads.	Predict page
8. The system will display the prediction chart.	Predict Page
9. The User has the option of resetting the parameters.	Predict Page

Use Case Name: App Analytics	ID: UC 5	Importance Level: High
Primary Actor: Registered user		
Short Description: This describes how a registered user can look at analytics on the whole app market		
Trigger: User provides different conditions and filtering criteria on the analytics		
Type : <i>Internal</i>		
Preconditions:		
Registered user should be logged into the system		
Major Inputs:	Major Outputs:	

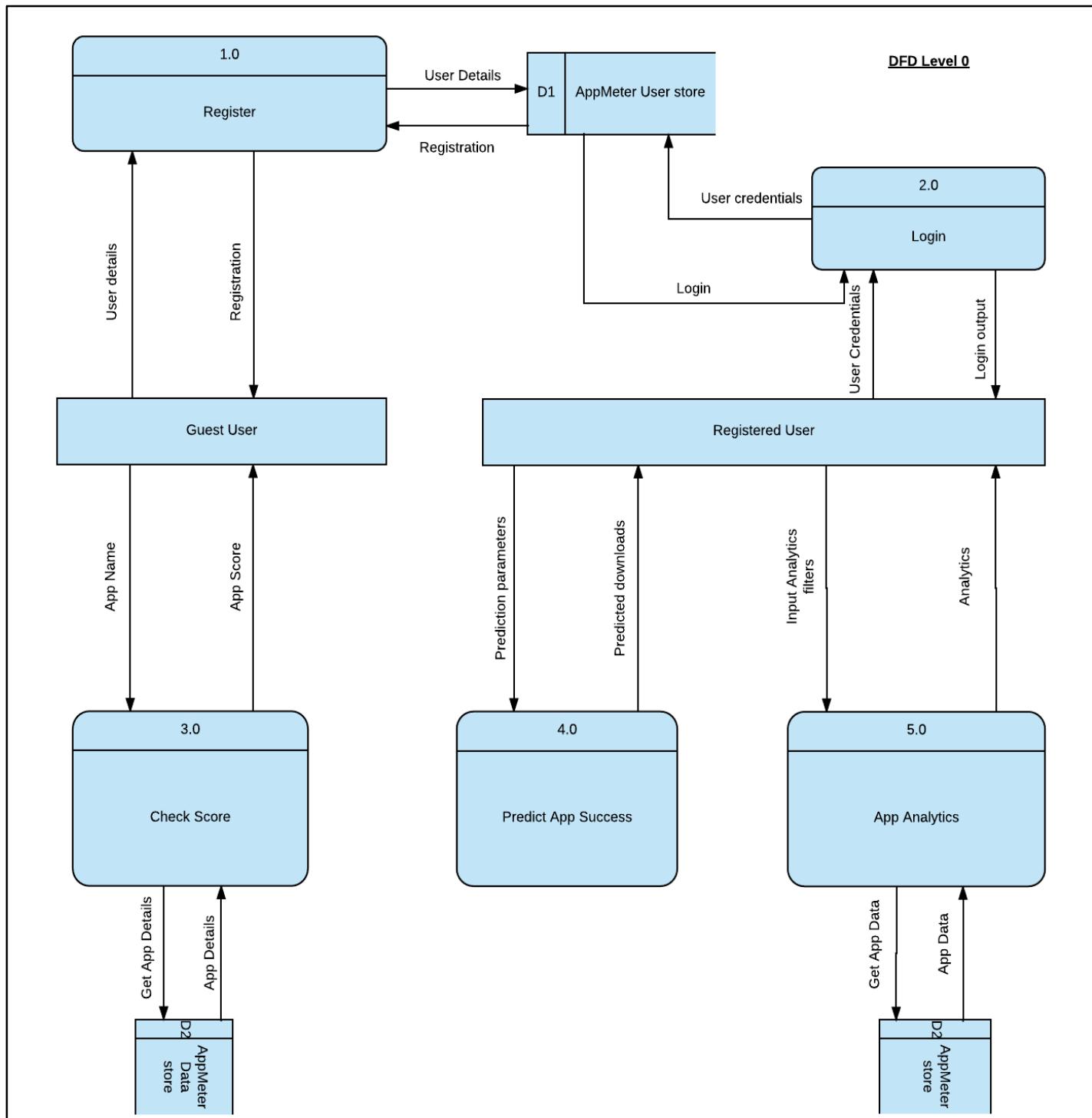
Description	Source	Description	Destination
Genre of App (optional) Select Paid or Free App (optional) Select App Rating Bucket (optional)	Registered User Registered User Registered User	Total number of Apps Total Downloads Average File Size Bucket wise App count Bucket wise Downloads Percentage and count of Paid/Free Apps Year-wise Apps launched	Registered User Registered User Registered User Registered User Registered User Registered User
Major Steps Performed 1. User enters Analytics page to see the overall analytics of App market 2. User selects the relevant genre 3. User selects if App is Paid or Free 4. User select the Rating Bucket			Information for Steps Overall App Trends/Analytics Filtered Analytics specific to that genre only Further filtered result

8. Data Flow Diagrams

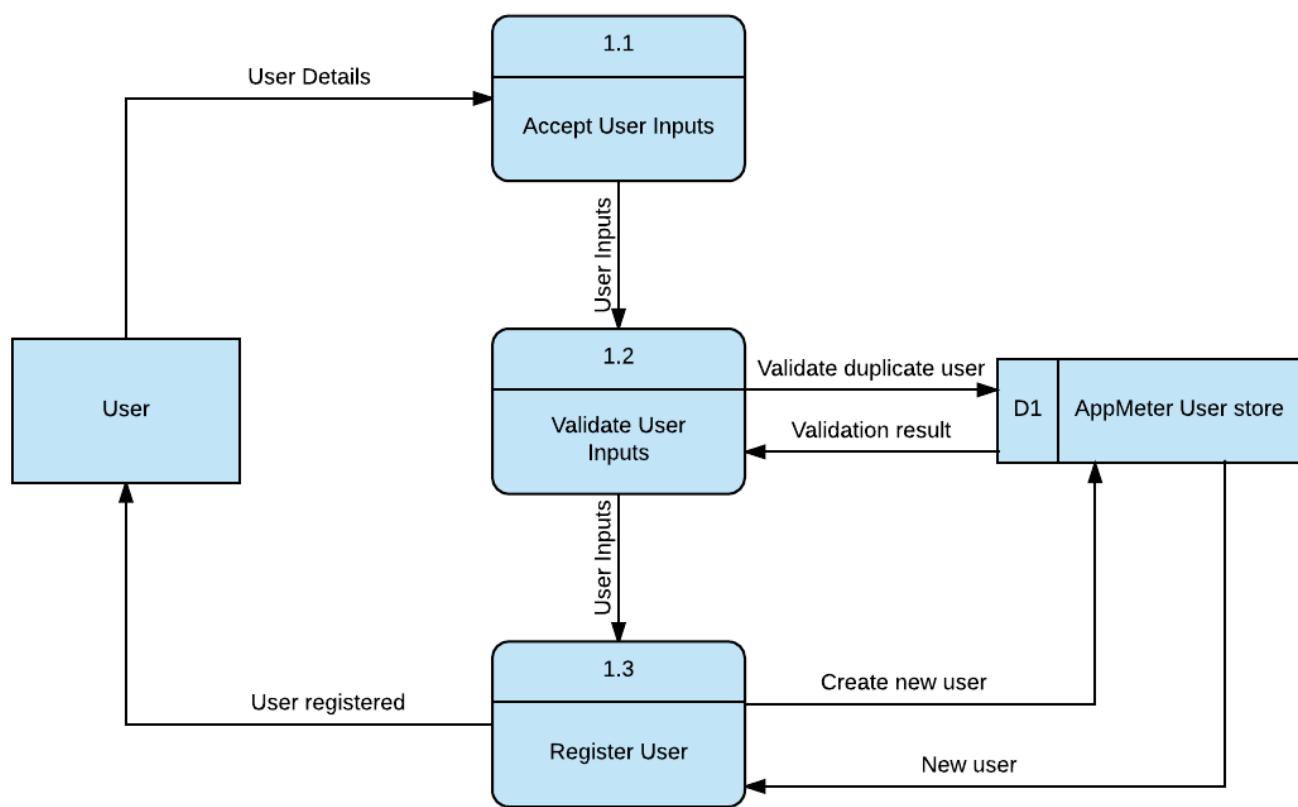
Context Diagram



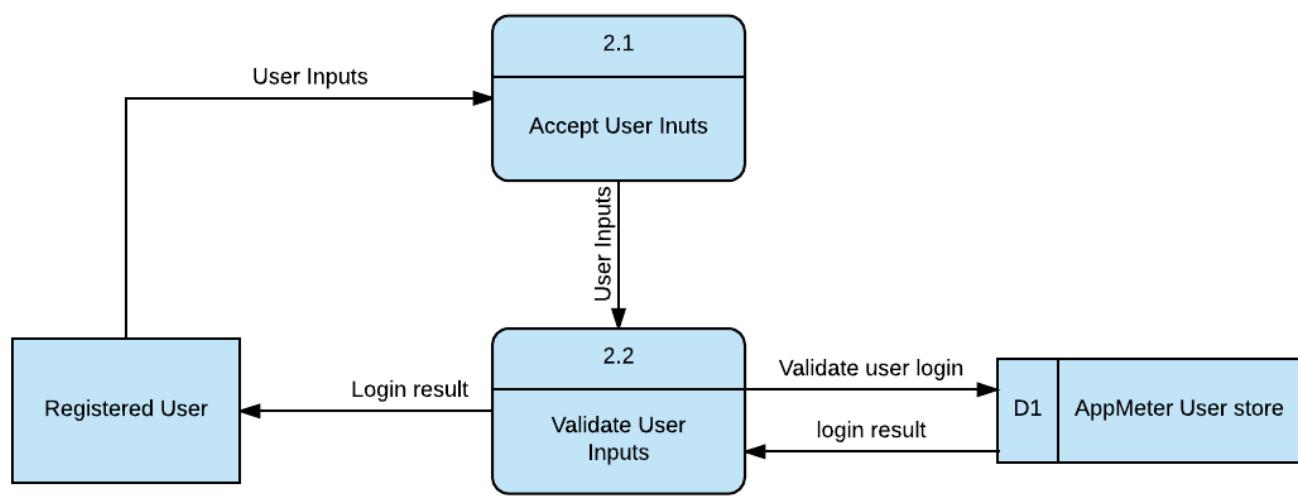
DFD Level 0



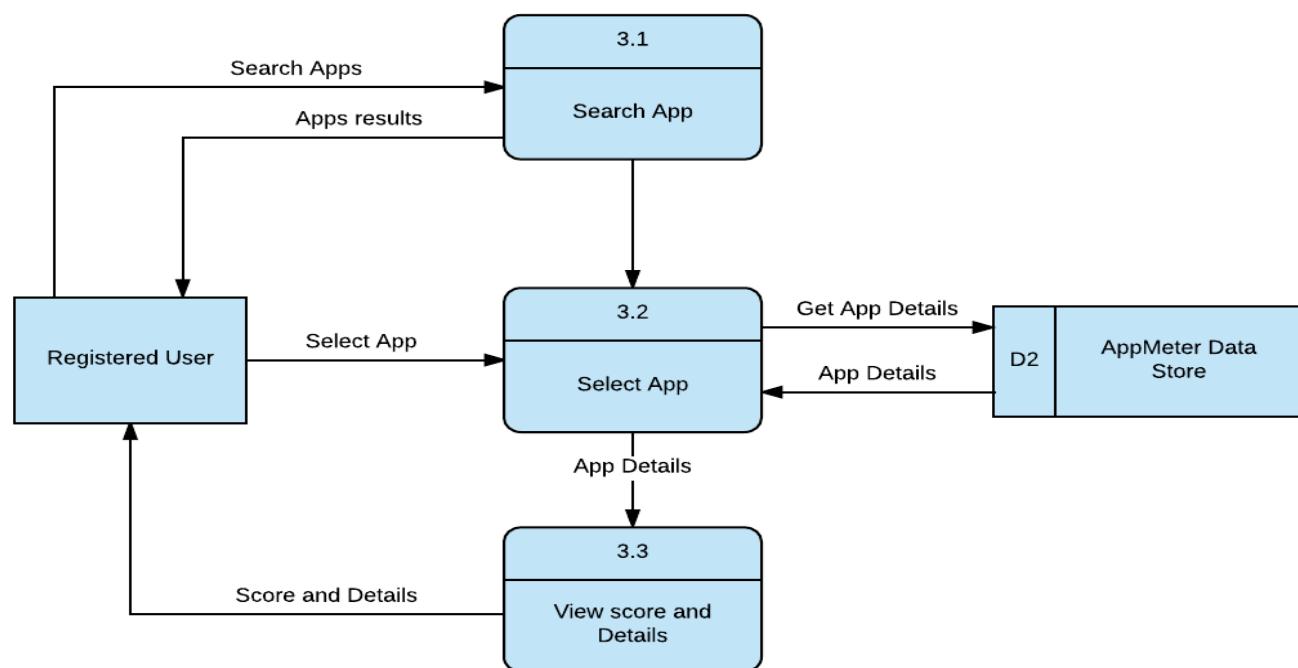
PROCESS 1 - REGISTER
DFD Level 1



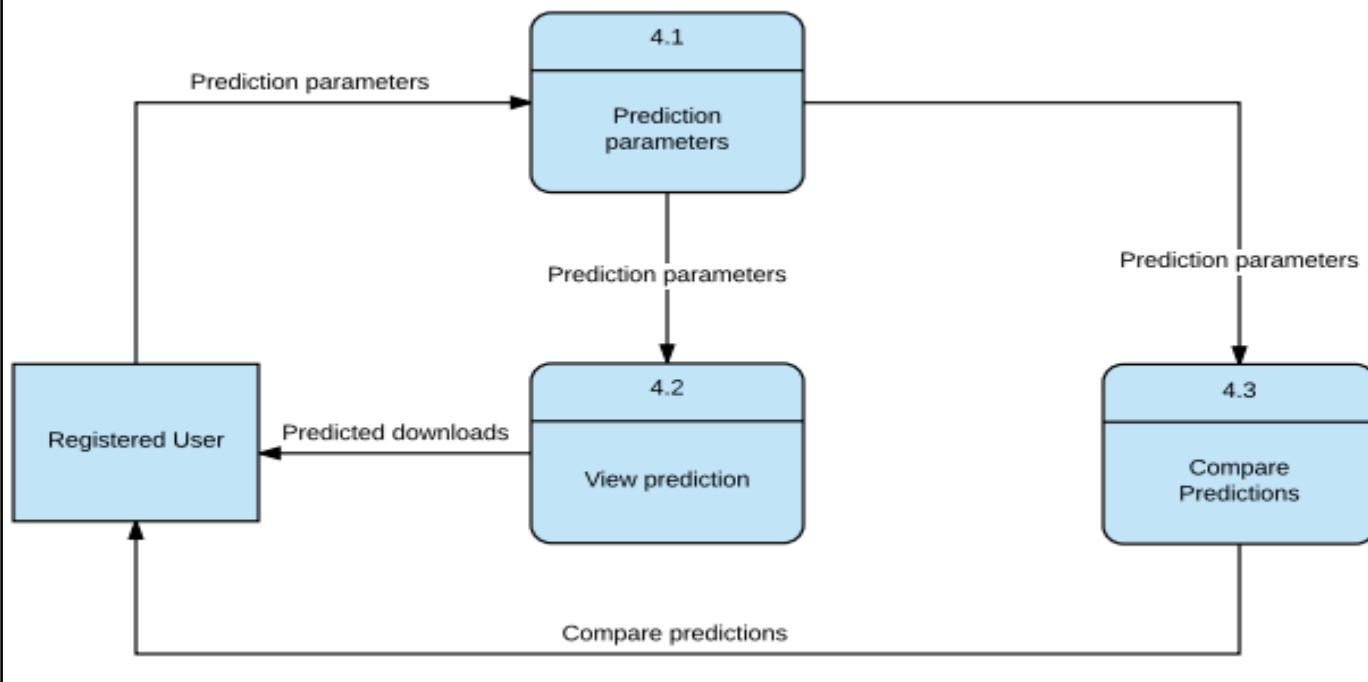
PROCESS 2 - LOGIN
DFD Level 1



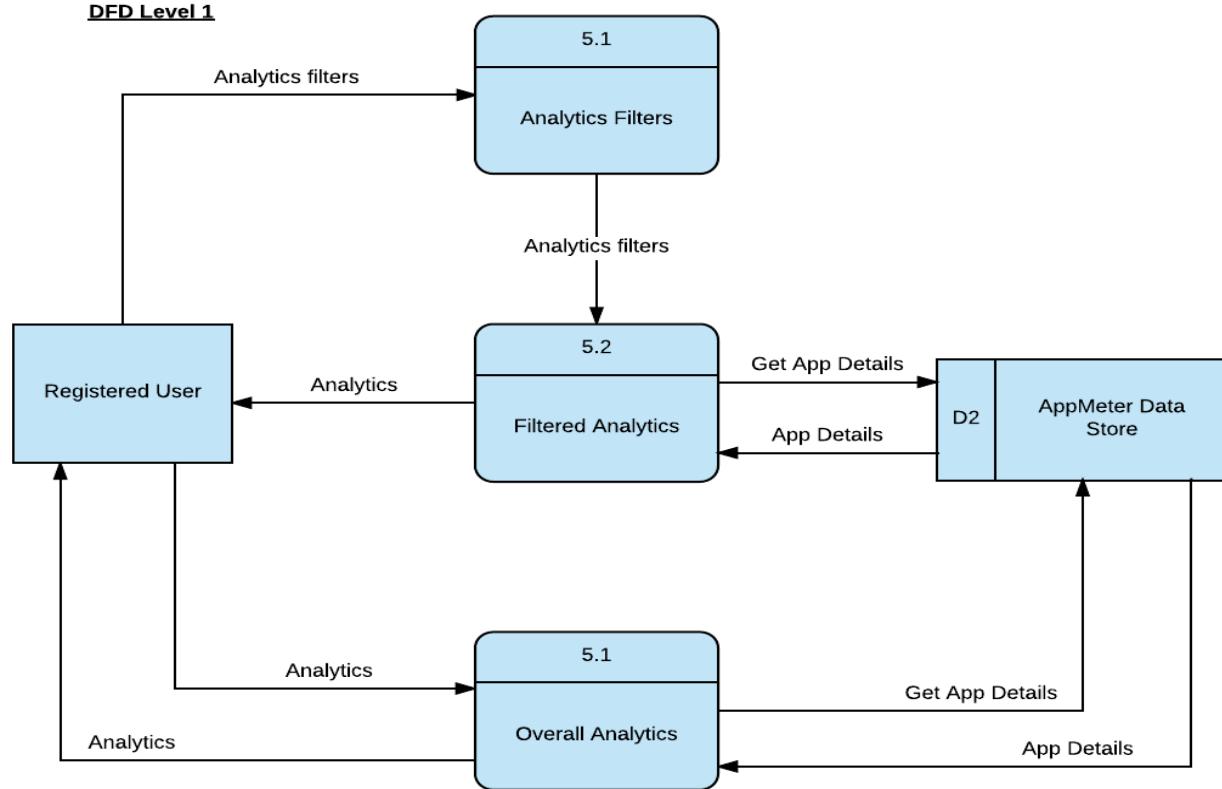
PROCESS 3 - Check Score
DFD Level 1



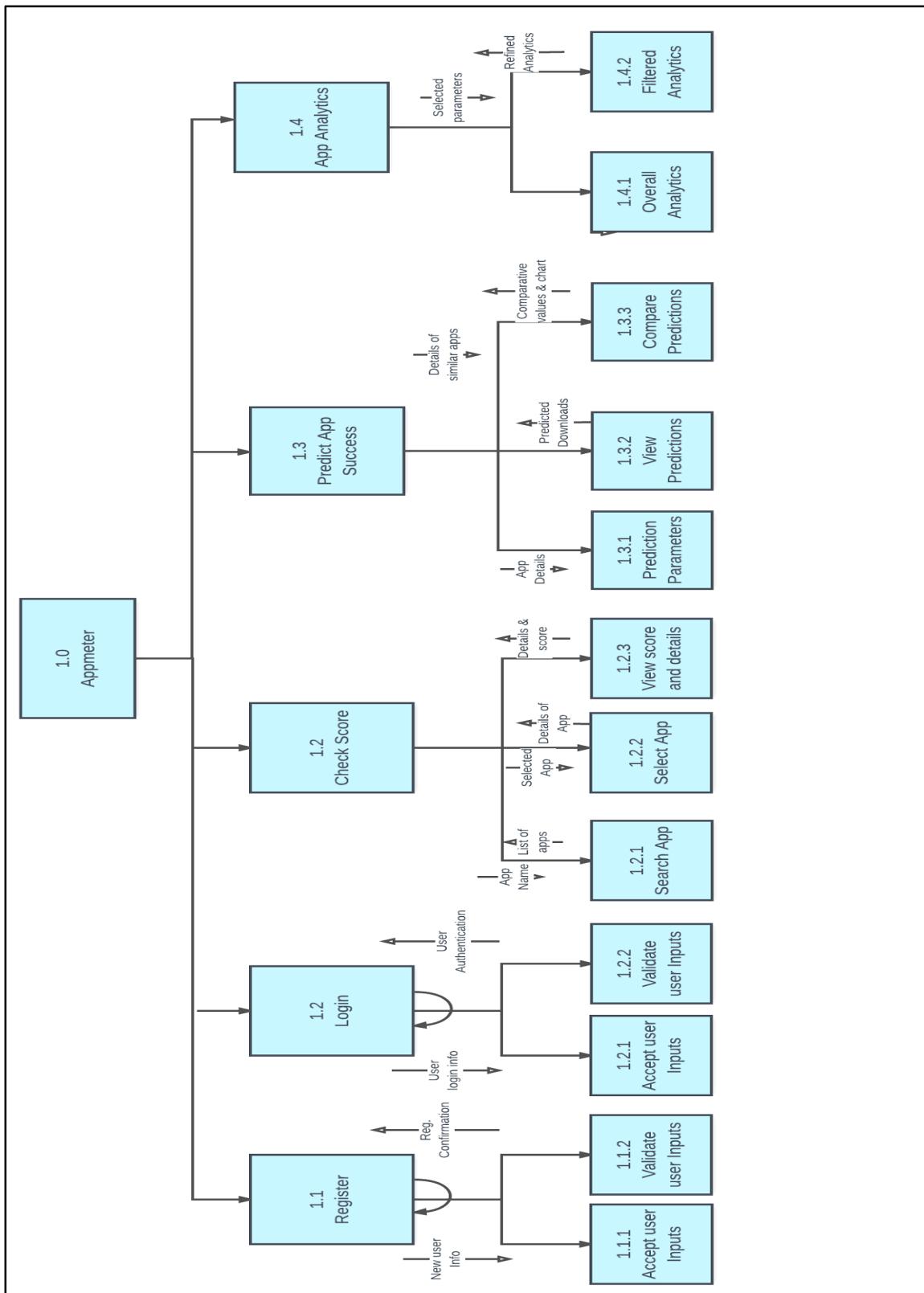
PROCESS 4 - Predict App success
DFD Level 1



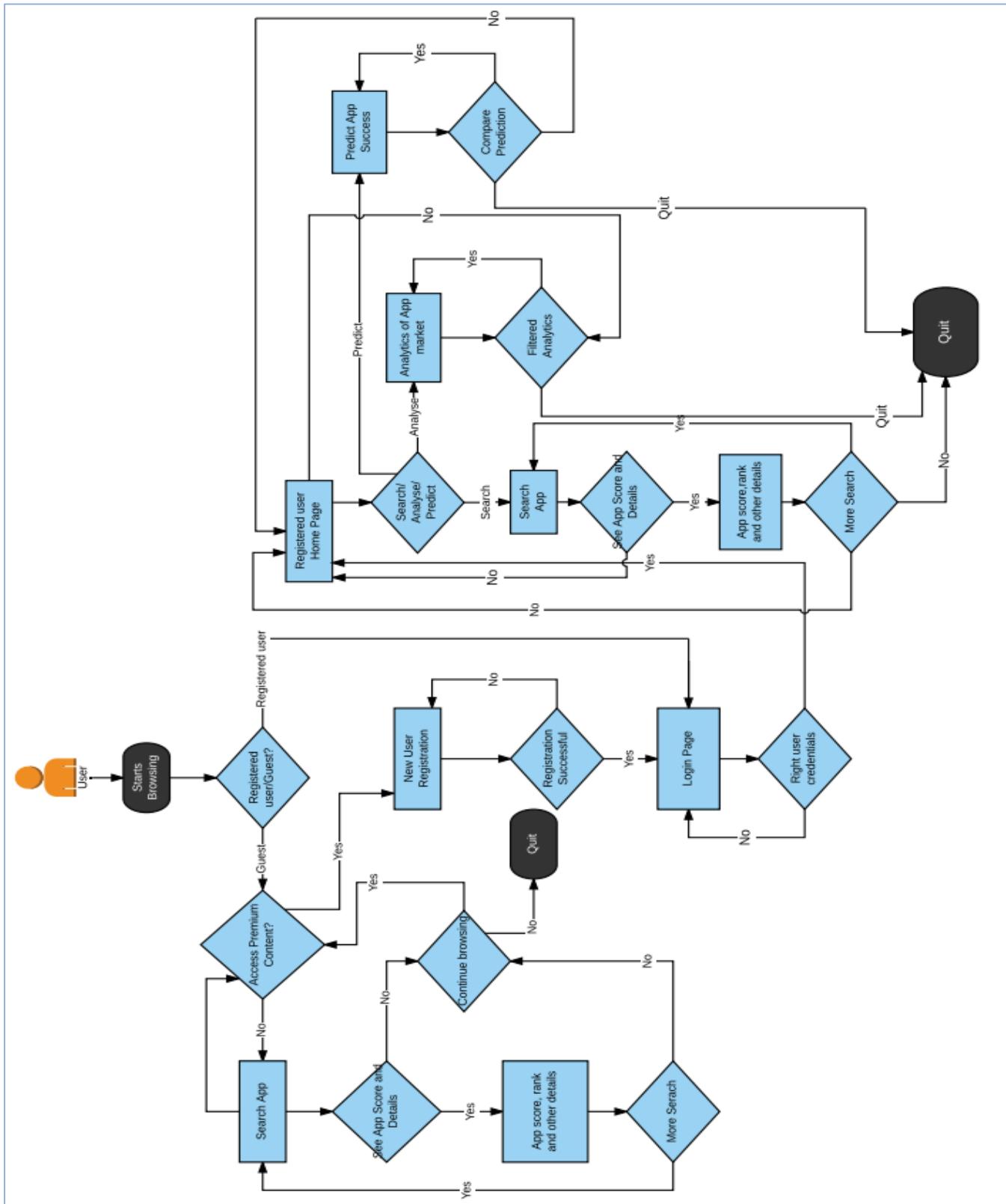
PROCESS 5 - App Analytics
DFD Level 1



9. Program Chart



10. Process Flow Diagram



11 Technology Stack

Data Analytics Python has been used primarily to explore, clean and validate the data

Following python libraries have been extensively used

- Numpy (<http://www.numpy.org/>)
- Pandas (<http://pandas.pydata.org/>)

Machine learning: Multiple Machine learning algorithms have been used for deriving the AppMeter score.

Following machine learning Python libraries have been used

- Statsmodel (<http://www.statsmodels.org/stable/index.html>)
- Scikit-learn (<http://scikit-learn.org/stable/>)
- Matplotlib (<https://matplotlib.org/>)
- Graphviz (<https://pypi.python.org/pypi/graphviz>)

Statistical package R has been used to derive the AppMeter prediction formula. Both the data cleaning/wrangling and running the 2SLS model have been implemented using the following libraries in R

<https://cran.r-project.org/web/packages/systemfit/systemfit.pdf>

<https://cran.rstudio.com/web/packages/dplyr/vignettes/introduction.html>

Analytics: The huge data (more than 450000 Apps) was visualized using Microsoft Power BI package. An interactive dashboard has been created and embedded into AppMeter website.

- Microsoft Power BI (<https://powerbi.microsoft.com/en-us/>)

Front End: Our front end predominantly consists of HTML, CSS, Javascript and PHP. Our front end is a combination of HTML and CSS which is supported by ajax script for fast processing of the data. This helps the website to be faster and functioning till the time the data is delivered from the server. PHP is used for database connection.

- HTML (<https://www.w3.org/TR/html51/>)
- CSS (<https://www.w3.org/Style/CSS/Overview.en.html>)
- Javascript (<https://www.javascript.com/>)
- PHP 5 (<http://www.php5-tutorial.com/>)

Web server: AppMeter website has been hosted on Awardspace (commercial web hosting service). The hostname for AppMeter is (<http://appmeter.atwebpages.com/home.php>). This ensures high availability (24*7) of AppMeter website

- Web hosting service - Awardspace (<https://www.awardspace.com/>)

Database: MySQL database is used for storing the App related data. Data definition and manipulation (CRUD) operations are done through SQL queries

- MySQL V.5 (<https://www.mysql.com/>)

Management tools: Microsoft Project is used for creating the project plan and tracking the project progress.

Diagram and documentation tool: Microsoft Visio is used for creating the UML diagrams.

Version Control: We used GitHub as our tool for version control and rollback during development.

12. Test Cases

Test Case ID:	P01	Test Designed By:	Ameya Ghatpande					
Test Priority (Low/Medium/ High)	High	Test Designed Date:	05/10/2017					
Module Name	Display AppMeter Score	Test Executed By:	Amarjit Dhal					
Test Title	Scoring Functionality	Test Execution Date:	05/15/2017					
Description	Verify if an AppMeter generated unique score is displayed for any App searched by user							
Dependencies	No							
Step	Test Steps	Expected Result	Actual Result	Status (Pass or Fail)	Notes			
1	User opens the Score page	Score page should be opened displaying the search App box	Score page with search box opens	Pass				
2	User inputs any keyword for searching the App And hits “Enter”	All the Apps matching the user provided keywords should be displayed	Same and Relevant results came	Pass	Several combinations were used			
3	User filters the result by selecting different criteria like Genre/price	Search results must be filtered showing only the filtered Apps	Search results are filtered	Pass				

4	User clicks on an App to view the score	A new page should be displayed with following information about the App 1] AppMeter Score 2] AppMeter Rank 3] App Name 4] App Description 5] App Icon 6] Downloads 7] Rating 8] Number of raters 9] Price 10] Release date	The selected app opens, and have all the required data as specified in the test case	Pass	All test cases passed on 3 rd Iteration on date: 05/25/2017
5	User clicks the back button on browser	User should be taken back on the “Score” page	Goes back to search page	Pass	

Test Case ID:	P02	Test Designed By:	Ameya Ghatpande					
Test Priority (Low/Medium/ High)	High	Test Designed Date:	05/11/2017					
Module Name	Predict the App Downloads	Test Executed By:	Amarjit Dhal					
Test Title		Test Execution Date:	5/15/2017					
Description	Predict the success(downloads) of App for the given set of user parameters							
Dependencies	User is registered in AppMeter system and has logged in							
Step	Test Steps	Expected Result	Actual Result	Status (Pass or Fail)	Notes			
1	User opens the predict page	Predict page should be opened displaying the different fields for user to enter	Opens the predict price	Pass				
2	User inputs the App parameters	AppMeter should validate all the user input parameters (not NULL) like 1] App rating 2] App price 3] App file size 4] App Age	User input validations are made	Pass	Passed on 2 nd iteration on date: 05/22/2017			
3	User hits the “predict” button	AppMeter should predict the number of downloads for the user entered parameters	Predicted downloads	Pass				
4	User clicks the back button on browser	User is taken back to the AppMeter home page	Goes to predict page	Pass				

Test Case ID:	P03	Test Designed By:	Ameya Ghatpande					
Test Priority (Low/Medium/High)	High	Test Designed Date:	05/19/2017					
Module Name	Display the Analytics page	Test Executed By:	Amarjit Dhal					
Test Title		Test Execution Date:	05/25/2017					
Description	Show the Apps related dashboard to user							
Dependencies	User is registered in AppMeter system and has logged in							
Step	Test Steps	Expected Result	Actual Result	Status (Pass or Fail)	Notes			
1	User opens the Analytics page	Analytics page should be opened and Genre list should be displayed to the user	Opens the page and display the default analytics	Pass				
2	User selects a Genre/multiple Genre's	AppMeter should display following information of all the Apps belonging to that selected Genre's 1] Number of downloads 2] Average rating 3] Average file size 4] Number of free/paid Apps	When user selects any filter, then displays all related values.	Pass	Different filter conditions were satisfied			
4	User clicks the back button on browser	User is taken back to the AppMeter home page	Goes to home page	Pass				

Test Case ID:	P04	Test Designed By:	Ameya Ghatpande					
Test Priority (Low/Medium/ High)	Medium	Test Designed Date:	05/11/2017					
Module Name	New user is registered in AppMeter	Test Executed By:	Amarjit Dhal					
Test Title		Test Execution Date:						
Description	Register a new user in AppMeter system							
Dependencies	No							
Step	Test Steps	Expected Result	Actual Result	Status (Pass or Fail)	Notes			
1	User opens the Register page	Register page should be opened displaying the different fields for user to enter	Registration page opens	Pass				
2	User inputs self-information	AppMeter should validate all the user input parameters (not NULL) like 1] User name 2] User Email 3] User Password 4] User company name	All Validation is done	Pass	Email validation is not done			
3	User hits the “Register” button	AppMeter system should validate if the user is already registered in system	Registration done for new and valid information.	Pass				

		(by email). If yes, then take the user to Login page. If not registered, then the new user should be entered the system	Old user is altered that he is in system already.		
4	User clicks the back button on browser	User is taken back to the AppMeter home page	Goes back to home page	Pass	

13. Data Dictionary

A data dictionary is a collection of descriptions of the data objects or items describing what type of data is collected within a database, its format, structure, and how the data is used. A data dictionary can be thought of as the rules which all the data within the system follows.

Table	ANDROID			
Table Description	The table contains the list of over 450,000 apps from the Android store			
Column Name	Description	Type	Mandatory	Unique
bundle_id	Unique Identifier for each App	VARCHAR (255)	YES	YES
app_name	Name of the App	VARCHAR (255)	YES	NO
app_description	Brief description of the App	VARCHAR (1000)	NO	NO
age	Number of days since the app release till Jan 7, 2013	FLOAT	YES	NO
all_rating	Average rating	FLOAT	YES	NO
all_rating_count	Number of raters	FLOAT	YES	NO

release_date	App release date	VARCHAR (20)	YES	NO
downloads	Number of downloads of an app	FLOAT	YES	NO
file_size	Size of the app in mb	FLOAT	YES	NO
genre	App Category	VARCHAR (255)	YES	NO
mod_genre	Modified Genre	VARCHAR (200)	YES	NO
icon_url	App Icon/Logo	VARCHAR (255)	NO	NO
price	Price of the app	FLOAT	YES	NO
score	Comprehensive score of an App (AppMeter proprietary)	FLOAT	YES	NO
rank	Rank of an App within their genre (AppMeter proprietary)	FLOAT	YES	NO
version	App version	VARCHAR (50)	NO	NO

Table	USER			
Table Description	The table contains information about the registered users.			
Column Name	Description	Type	Mandatory	Unique
email	Email address of the user. Unique ID.	VARCHAR (20)	YES	YES
fname	First Name of the user	TEXT	YES	NO
lname	Last Name of the user	TEXT	YES	NO
password	User password	VARCHAR (20)	YES	NO
company	Name of the company	VARCHAR (20)	YES	YES

14. Machine Learning Algorithm researched and used

14.1 Linear Regression (OLS)

In linear regression, we predict index/coefficient of the variables. Ordinary least squares (OLS) or linear least squares is a method for estimating the unknown parameters in a linear regression model, with the goal of minimizing the sum of the squares of the differences between the observed responses.

How this can be applied to our project: In our project, the dependent variable is number of downloads, while all other parameters are independent variable. This method is used to find the coefficient of each of the independent parameters

Model:

Number of downloads ~ Number of Raters + Average Rating + Age of the App + File size + Price

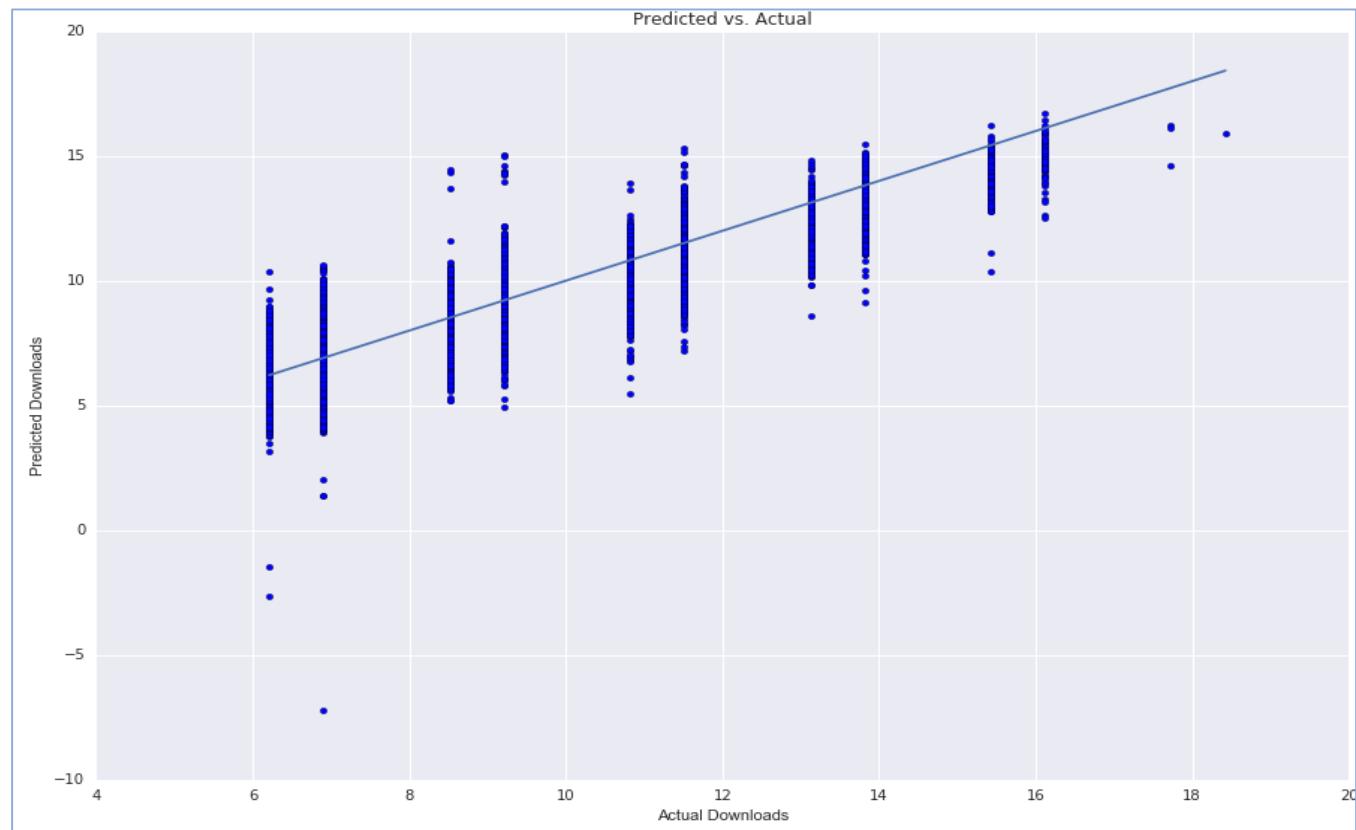
Usage of the model: There was no linear relationship between the number of downloads and other parameters. There were many outliers in our data, and hence this model was not accurate. Even though this model was used for initial research, it was not used as the final model

```
In [4]: linearRegressionData = pd.read_csv("/Users/test/Downloads/android-for-amarjit.csv")
In [7]: linearRegressionData["Downloads"] = linearRegressionData.downloads.apply(toLog)
         linearRegressionData["Age"] = linearRegressionData.age.apply(toLog)
         linearRegressionData["Raters"] = linearRegressionData.all_rating_count.apply(toLog)

In [13]: import statsmodels.formula.api as smf
          linearRegressionResult = smf.ols(formula='Downloads ~ price + all_rating + Raters + file_size + Age', data = linearRegressionData)
          linearRegressionResult.params
          print(linearRegressionResult.summary())
Out[13]: Intercept      5.022925
          price        -0.077404
          all_rating   -0.207137
          Raters       1.127823
          file_size    -0.005356
          Age          0.101184
          dtype: float64
```

```

OLS Regression Results
=====
Dep. Variable: Downloads R-squared: 0.750
Model: OLS Adj. R-squared: 0.750
Method: Least Squares F-statistic: 2.886e+05
Date: Sun, 28 May 2017 Prob (F-statistic): 0.00
Time: 19:38:07 Log-Likelihood: -8.2075e+05
No. Observations: 481151 AIC: 1.642e+06
Df Residuals: 481145 BIC: 1.642e+06
Df Model: 5
Covariance Type: nonrobust
=====
      coef  std err      t   P>|t|    [ 0.025   0.975 ]
-----
Intercept  5.0229  0.015  332.380  0.000    4.993  5.053
price     -0.0774  0.001 -126.179  0.000   -0.079 -0.076
all_rating -0.2071  0.002 -100.117  0.000   -0.211 -0.203
Raters     1.1278  0.001 1146.920  0.000    1.126  1.130
file_size  -0.0054  0.000 -40.925  0.000   -0.006 -0.005
Age        0.1012  0.002   49.940  0.000    0.097  0.105
-----
Omnibus: 13101.056 Durbin-Watson: 1.730
Prob(Omnibus): 0.000 Jarque-Bera (JB): 21935.885
Skew: -0.250 Prob(JB): 0.00
Kurtosis: 3.919 Cond. No. 124.
=====
```



14.2 Decision Tree Regression

It is one of the predictive modelling approaches, where the observations about an item is represented in the branches. It predicts the value of the target variable by using set of conditions and path. The leaf of a decision tree represents a value of the target variable given the values of the input variables represented by the path from the root to the leaf.

How this can be applied to our project: This method is used to determine the possible value of dependent variable based on independent variables. We need to specify the depth of the tree and similarly the number of parameters will be considered.

Usage of the model: The Decision Tree Regression algorithm was not used in our model, as we did not have any control over the path that was selected in determining the number of downloads. Without increasing the depth of the tree, we were unable to cover all the variables in the equation, but with increased depth, the tree got more convoluted and tedious to understand.

```
In [6]: import sklearn.tree as tree
import sklearn as sk
from sklearn.model_selection import train_test_split

Y=AllCleanedApps['Downloads']
X=AllCleanedApps.drop(['Downloads', 'Name', 'Unnamed: 0', 'Random'], axis=1)

X_train, X_test, y_train, y_test = train_test_split(X,Y,test_size=0.3,random_state=2)
```

```
In [8]: import numpy as np
dt = tree.DecisionTreeRegressor(max_depth=7)
dt.fit(X = X_train, y= y_train)

Out[8]: DecisionTreeRegressor(criterion='mse', max_depth=7, max_features=None,
                               max_leaf_nodes=None, min_impurity_split=1e-07,
                               min_samples_leaf=1, min_samples_split=2,
                               min_weight_fraction_leaf=0.0, presort=False, random_state=None,
                               splitter='best')
```

14.3 2SLS

Two-Stage least squares regression technique is often used when the dependent variable's error terms are correlated with the independent variables. Additionally, it is useful when there are feedback loops in the model. This technique is the extension of the OLS method. A 2-stage method is followed

1. In the first stage, a new variable is created using the instrument variable.
2. In the second stage, the model-estimated values from stage one is then used in place of the actual values of the problematic predictors to compute an OLS model for the response of interest.

How can this be applied to our project?

Our project had 2 endogenous variables; Downloads and Word of Mouth(WOM) where

Word of mouth (WOM) = Total number of Stars an app has received till date / Age of the App.

We created the 2 Stage least squares equations as-

Downloads = Wom + Age + Price + Rating + File_size + Genre + dummy_price

Where,

Word of mouth (WOM) = Total number of Stars an app has received till date / Age of the App.

Total Stars = Average Rating of an app * Total number of Raters

Model

```
eqDown <- ldownloads ~ lwom + age + price + all_rating + mod_genre + dummy_price1 + dummy_price2 + dummy_price3 + dummy_price4
eqWOM<-lwom~ldownloads + file_size + all_rating
inst<-~price + mod_genre + all_rating + age + file_size + dummy_price1 + dummy_price2 + dummy_price3 + dummy_price4
system <- list(DL = eqDown, WOM = eqWOM)

fitI2sls <- systemfit(system, method = "2SLS", inst = inst, data =train)
summary(fitI2sls)    #ask prof (NAs produced by interger overflow)
```

14.4 Principal Component Analysis

It is often useful to measure data in terms of its principal components rather than on a normal X-Y axis.

Principal components are the directions where there is the most variance in the data. PCA can be used to calculate the score of an app

How can this be applied to our project: We passed the data to the PCA algorithm to determine which component carries the highest coverage of information. We determined that the quickly an app is downloaded, the better it is for an app, and hence a better score. For Paid Apps, score also have a direct relationship with the price of the app as compared with the highest priced apps in that genre. We also observed that the score is also dependent on “word of mouth”. So irrespective of an app having high price, a better word of mouth can elevate its score.

Word of mouth (WOM) = Total number of Stars an app has received till date / Age of the App.

Total Stars = Average Rating of an app * Total number of Raters

Model:

We created our model per genre of apps, as every genre have a different acceptance among the users.

Below are the different parameters we calculated for our model. We used log values to normalize the data.

```
In [8]: df["Downloads"] = df.downloads.apply(toLog)
df["Size"] = df.file_size.apply(toLog)
df["Price"] = df.price1.apply(toLog)
df["AvgR"] = df.all_rating.apply(toLog)
df["Raters"] = df.all_rating_count.apply(toLog)

In [10]: df["stars"]=(df["all_rating_count"]*df["all_rating"])

In [11]: df["TotalStars"] = df.stars.apply(toLog)
df["Age"] = df.age.apply(toLog)

In [12]: df["AgePerYear"]=df["age"]/365
df["wom"]= df["Stars"]/df["Age"]
df["DownloadsPerYear"]=df["Downloads"]/df["Age"]
```

We ran our model on specific genre (*Game* in the screen capture below)

```
In [25]: dfpaid = df[df.price > 0]

Genre : Games

In [26]: dfGame = df[df.mod_genre=='Weather']

In [27]: dfw["price_factor"] = dfw.price/dfw.price.max()

...
Score

In [28]: dfGame["score"] = 0.7 * (dfGame['DownloadsPerYear'] * dfGame['price_factor']) + (0.3 * dfGame['wom'])
```

Using PCA:

```
In [36]: from sklearn.preprocessing import StandardScaler
X_std = StandardScaler().fit_transform(dfgpca)

In [37]: from sklearn.decomposition import PCA
pca = PCA(n_components=1, whiten=False).fit(X_std)

In [38]: print pca.components_
[[ 0.70710678  0.70710678]]

In [39]: print pca.explained_variance_ratio_
print sum(pca.explained_variance_ratio_)

[ 0.92525638]
0.925256380965

In [40]: pca.components_.T * math.sqrt(pca.explained_variance_)

Out[40]: array([[ 0.96190248],
   [ 0.96190248]])
```

$$\text{Score} = 0.7(\text{Download Factor} + \text{Price Factor}) + 0.3(\text{Popularity Factor})$$

Where: Score is calculated per genre.

Download Factor is Number of Downloads of the App/Age of the App

Price Factor is Price of the App/Highest Price of that genre

Popularity Factor is number of stars the app has got till date/Age of the App

** All the data has been normalized using logarithm, as the data had lot of variance in their range.

Usage of the model: PCA is used in our project, since we can determine how the parameters are able to affect the scores.

15 Code Snippets for Data Cleaning

Since AppMeter project is heavily data centric, the data cleaning part was extensive and was done very carefully. Following steps were followed to make the data ready for machine learning algorithms

1. The large data file (. json) of 2 Gb was broken into smaller chunks of 500 Mb each so that it can be processed by the Python Jupyter notebook

```
In [ ]: class Jsonloads:  
    def read_in_chunks(self, file_object, chunk_size=500000000):  
        while True:  
            data = file_object.read(chunk_size)  
            if not data:  
                break  
            yield data  
  
In [ ]: count = 1;  
# <ghatpande> on 14-April-2017 for AppMeter  
# Give the path of the large input JSON file  
f = open('Original_Data_from_Luka.json')  
for piece in Jsonloads().read_in_chunks(f):  
    pathname = "part" + str(count) + ".csv"  
    text_file = open(pathname, "w")  
    text_file.write(piece)  
    text_file.close()  
    count = count + 1;  
    print("\n =====The new chunk is ===== \n")  
  
print("Count is %d" %count);
```

2. Remove all the outliers (those Apps which have Nulls/0 values) for columns like

- Downloads
- Raters
- App Name
- File Size

```
In [25]: # Remove all the entries with Nulls/0 values
dfcombined = dfcombined[dfcombined.all_rating_count != 0]
dfcombined = dfcombined[dfcombined.downloads.isnull() != True]
dfcombined = dfcombined[dfcombined.app_name.isnull() != True]
dfcombined = dfcombined[dfcombined.file_size.isnull() != True]
len(dfcombined)
```

3. Mobile Apps can be free or paid. In our data, free Apps had value “Free” for Price column. This was a String field. Also paid apps had a Dollar (\$) symbol before the price. So, we removed the \$ symbol and for Free apps assigned a value 0 to it. By converting the data type to float, we made this column useful for the mathematical calculations in predicting the success of App.

```
In [27]: #Change price column

def removedollar(x):
    tempstring=str(x)
    if str(x)=="Free":
        tempstring=float("0")
    elif str(x).startswith("$"):
        tempstring1=str(x).lstrip("$")
        tempstring=float(tempstring1)
    return float(tempstring)
```

```
In [28]: dfcombined["pricel"] = dfcombined["price"].apply(lambda x: removedollar(str(x)))
```

4. The file Size of the Apps was a combination of both MB and KB. Also, this was a string field and had “Mb” or “KB” attached to it. We converted all the Apps in Mb's and made this a float field so that it is useful for the mathematical calculations in predicting the success of App.

```
In [49]: # Change the file_size column

def changeFileSize(x):
    tempstring = str(x)
    tempstring = tempstring.replace(",","")
    if str(x).endswith("K"):
        tempstring = tempstring.rstrip("K")
        tempfloat = float(tempstring)
        return float(tempfloat/1000)
    if str(x).endswith("M"):
        tempstring = tempstring.rstrip("M")
        return float(tempstring)
    else:
        return int(0.0)
```

```
In [50]: dfcombined["file_size1"] = dfcombined["file_size"].apply(lambda x: changeFileSize(str(x)))
```

5. Every App has a Genre/Category associated with it. There are many Genre's which are closely associated with each other. (Cards and Casino, Brain and puzzles both can be categorized as Games). By combining 30 minor Genre's into 20 Genre's, we could reduce the number of genres and find genre specific insights more accurately.

```
In [58]: # Combine the genre's into logical group of reduced number of genre's
dfcombined[ "mod_genre"] = dfcombined[ "genre"]
dfcombined[ "mod_genre"].replace('Social','Social Networking',inplace=True)
dfcombined[ "mod_genre"].replace('Brain & Puzzle','Games',inplace=True)
dfcombined[ "mod_genre"].replace('Media & Video','Photo & Video',inplace=True)
dfcombined[ "mod_genre"].replace('Personalization','Entertainment',inplace=True)
dfcombined[ "mod_genre"].replace('Photography','Photo & Video',inplace=True)
dfcombined[ "mod_genre"].replace('Arcade & Action','Games',inplace=True)
dfcombined[ "mod_genre"].replace('Communication','Social Networking',inplace=True)
dfcombined[ "mod_genre"].replace('Tools','Utilities',inplace=True)
dfcombined[ "mod_genre"].replace('Comics','Entertainment',inplace=True)
dfcombined[ "mod_genre"].replace('News & Magazines','News',inplace=True)
dfcombined[ "mod_genre"].replace('Travel & Local','Travel',inplace=True)
dfcombined[ "mod_genre"].replace('Music & Audio','Music',inplace=True)
dfcombined[ "mod_genre"].replace('Racing','Games',inplace=True)
dfcombined[ "mod_genre"].replace('Casual','Games',inplace=True)
dfcombined[ "mod_genre"].replace('Transportation','Navigation',inplace=True)
dfcombined[ "mod_genre"].replace('Libraries & Demo','Entertainment',inplace=True)
dfcombined[ "mod_genre"].replace('Cards & Casino','Games',inplace=True)
dfcombined[ "mod_genre"].replace('Sports Games','Games',inplace=True)
```

6. There was some data which had Non-Ascii characters as well as special characters in the App Name field. This was creating problem in searching that Apps in the database. We replaced such Non- Ascii and special characters by space from the App Names. Now the App Names are in English and can be read easily

```
In [93]: # Remove the special characters from app_name column
dfcombined[ "app_name"].replace(to_replace = '[\!\\.\+<>,\\#\\$\\^\\&\\*()\\-=\\+\\~]+',value = ' ', regex = True, inplace=True)

# Remove Non Ascii codes from the app_name column
dfcombined[ "app_name"].replace(to_replace = '[^\x00-\x7F]+',value = ' ', regex = True, inplace=True)
```

16 Web Application User Interface

16.1 AppMeter Home Page:

<http://appmeter.atwebpages.com/home.php>

The purpose of any home page of a website is to help the user/visitor understand what the website/company is all about. The home page should have a conversation with the user, it should understand what type of user is visiting and direct the visitor to the intended use cases or functions. The user might want to know about the application, features, functions, the team and contact information. A home page should be a one-stop-shop for a user to know the product.

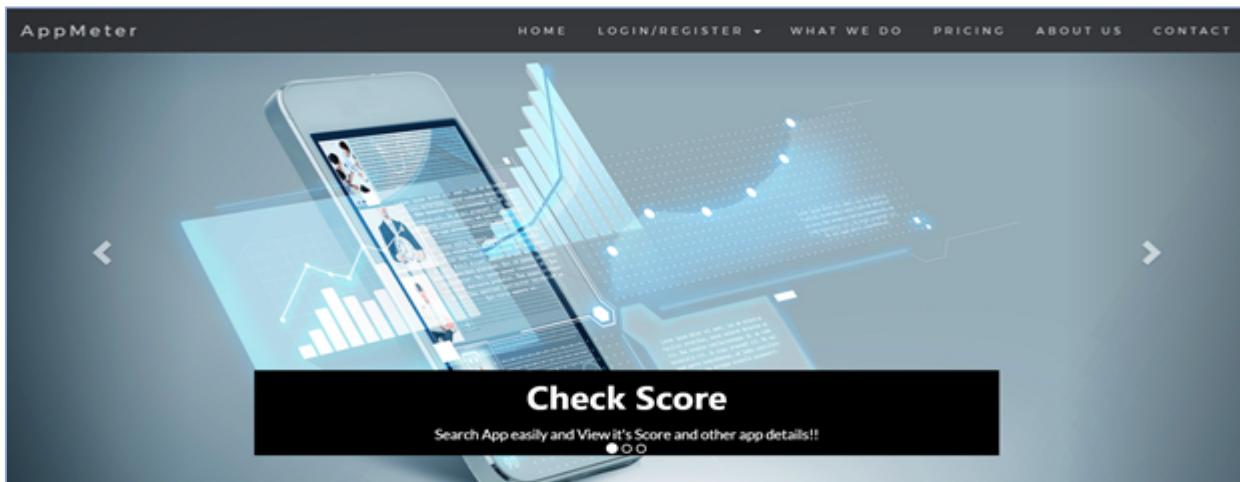
The AppMeter web application home page is user friendly and intuitive. The home page caters to every type of user; viz. a first-time visitor, the guest (unregistered) user, and the premium(registered) user.

The user interface is self-explanatory so that new visitor understands what AppMeter does, the functionalities it performs, its pricing model, the developer team and contact information. Basically, the user is walked through the features and functions of AppMeter.

Design Considerations:

- The home page functions like a single page website where each division is accessible with smoothness. This is currently a hot trend in the UI industry; as user does not have to navigate to other pages to access some rudimentary content.
- We have used carousal and images for displaying a quick view of AppMeter features at the first sight to catch user's attention.
- Intuitive and expressive icons make the homepage attractive. The intention is to keep the user excited and engaged.

Snapshots:



16.1.1 What we do Section

What we do section contains a short description of functions AppMeter has to offer. It describes the process for each of the AppMeter function viz. Score, Predict and Analyze.

The screenshot shows the 'WHAT WE DO' section of the AppMeter website. At the top, there's a navigation bar with links for HOME, LOGIN/REGISTER, WHAT WE DO (which is the active page), PRICING, ABOUT US, and CONTACT. Below the navigation, the title 'WHAT WE DO' is centered, followed by a subtext: 'Mobile App Analytics? We have got everything for you!'. Three main service sections are displayed: 'CHECKSCORE' with an icon of a gauge, 'PREDICT' with an icon of a hand holding a crystal ball over a graph, and 'ANALYZE' with an icon of a dashboard displaying various charts and graphs. Each service has a brief description below it.

Service	Description
CHECKSCORE	AppMeter Score - Our algorithm gives you a comprehensive score for a mobile application which will judge the performance/acceptance of the mobile application in the app market. Also see how that application ranks in its genre!
PREDICT	Predictive Analytics - Are you planning to launch a new mobile app? Our download predictor will help you estimate future downloads! You just give in details of your app and that's it! Our algorithm will give you your download prediction for your new app!
ANALYZE	App Analytics - Curious about the current app market scenario? What genre has most downloads? How many apps are rated 4+? What's the genre-wise rating-wise Average AppMeter Score? Try our all-inclusive 'App Analytics' dashboard!

16.1.2 Pricing

As some functionalities of AppMeter are premium, AppMeter has a pricing model associated with the services it provides. The homepage pricing section comprises of quick information about the pricing model of AppMeter.

The screenshot shows the 'PRICING' section of the AppMeter website. The navigation bar at the top includes links for HOME, LOGIN/REGISTER, WHAT WE DO, PRICING (active), ABOUT US, and CONTACT. The 'PRICING' section features two main plan comparisons: 'Basic' and 'Pro'. Both plans include 'Score Check Included'. The 'Basic' plan offers 'Predictions Not Included', 'App Analytics Not Included', and 'No Support', with a price of '\$ 0 Per User Per Year'. The 'Pro' plan offers 'Predictions Included', 'App Analytics Included', and 'Full Support', with a price of '\$ 499 Per User Per Year'.

Plan	Score Check Included	Predictions Included	App Analytics Included	Support	Price
Basic	Score Check Included	No	No	No	\$ 0 Per User Per Year
Pro	Score Check Included	Yes	Yes	Full	\$ 499 Per User Per Year

16.1.3 Team

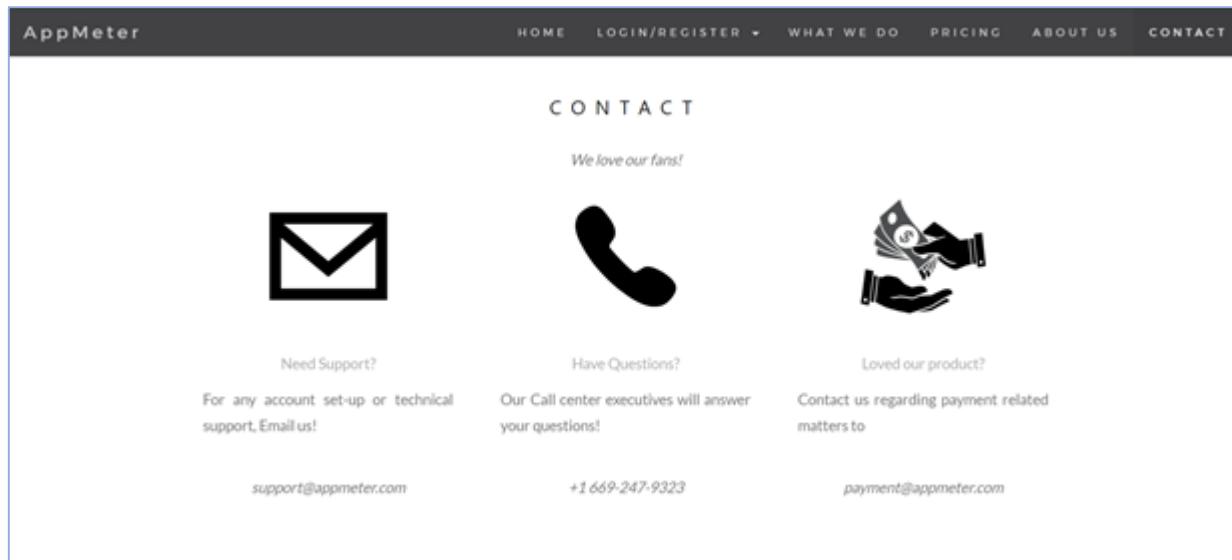
The developer and advisory team description and their profiles are mentioned in the team section. The user might want to know who developed the system and their interests and areas of expertise

The screenshot shows the 'TEAM' section of the AppMeter website. At the top, there's a navigation bar with links for HOME, LOGIN/REGISTER, WHAT WE DO, PRICING, ABOUT US, and CONTACT. Below the navigation, the word 'TEAM' is centered above the subtitle 'THE BRAINS BEHIND THE MAGIC'. There are five circular profile pictures, each with a name below it and a detailed description of their role and background.

- Prof. Yasin Ceran:** Professor Ceran's research interests include incentive issues in software development and demand estimation using information from online social networks. His teaching interests include telecommunications and business networks, business intelligence, web analytics, data mining, management information systems, data management, and SAS. Professor Yasin's extremely valuable guidance has helped AppMeter initiative immensely!
- Ameya Ghatpande:** Ameya has extensive experience of developing enterprise level software. He is one of the founding members of AppMeter team. He led the project and contributed in designing the machine learning algorithms to develop the trademark AppMeter Score and Ranking system. He has keen interest in big data analytics and cloud computing.
- Amarjit:** Amarjit is a software engineer by training and an analyst by passion, after completing his Bachelor's in Computer Science Engineering, he has worked for more than 6 years in IT industry in various roles before joining the MSIS course. He is passionate about how data can be used to create meaningful insights that can be beneficial to a business.
- Shalini Gopalakrishnan:** Shalini is a meticulous, proactive and focused individual with 4 years of experience as a software developer. During this tenure, she has designed and developed critical web-based applications from inception to launch. She aspires to become a data analyst as she is passionate about interpreting data, analyzing results using statistical techniques and providing reports using visualization tools. She has been working towards her aspiration through MSIS-coursework and also by learning the relevant skills on her own.
- Chaitrali:** Chaitrali has previous work experience in software development involving requirements gathering, analysis, design and testing of software systems. Technology and its applications gave a chance to work with tons of data with variety. During her internship, she got the opportunity to study data from diverse views and perceptions which helped the management of the organization in decision making. This required data processing, transformations, storage and visualizations which strengthened her competency in the field of analytics.
- Omkar Gokhale:** Omkar is passionate about Business Intelligence and has background in Information Technology. Working in BI domain has given him exposure to working with large data sets and experience with various tools for data manipulations, cleansing and visualizations. Omkar is proficient with data related programming and has good understanding of dimensional data modeling and complete BI life-cycle. Omkar's quick grasp on web development technologies has helped him gain momentum in his career.

16.1.4 Contact Us

The user might have any support issues or technical difficulties or simply, a feedback about AppMeter. The contact section provides information to contact the developer/customer support/payment teams. The user might want to get help or assistance regarding account set-up or payments.



16.2 Register

The premium features of the application require the user to be registered in the system. When the user tries to access the premium content, he is directed to the login page. If he is a new user and wants to continue to explore the premium features, he should register by providing the details of name, email, company and setting a password.

AppMeter

HOME LOGIN WHAT WE DO PRICING ABOUT US CONTACT

REGISTER

First Name : *

Last Name : *

Email : *

Company : *

Password : *



After the user has entered all the details and submitted it, the page will display a “successful login” message with the link to be directed to the login page to access the premium content.

AppMeter

HOME LOGIN WHAT WE DO PRICING ABOUT US CONTACT

REGISTER

You have Successfully Registered!

[Login Here](#)

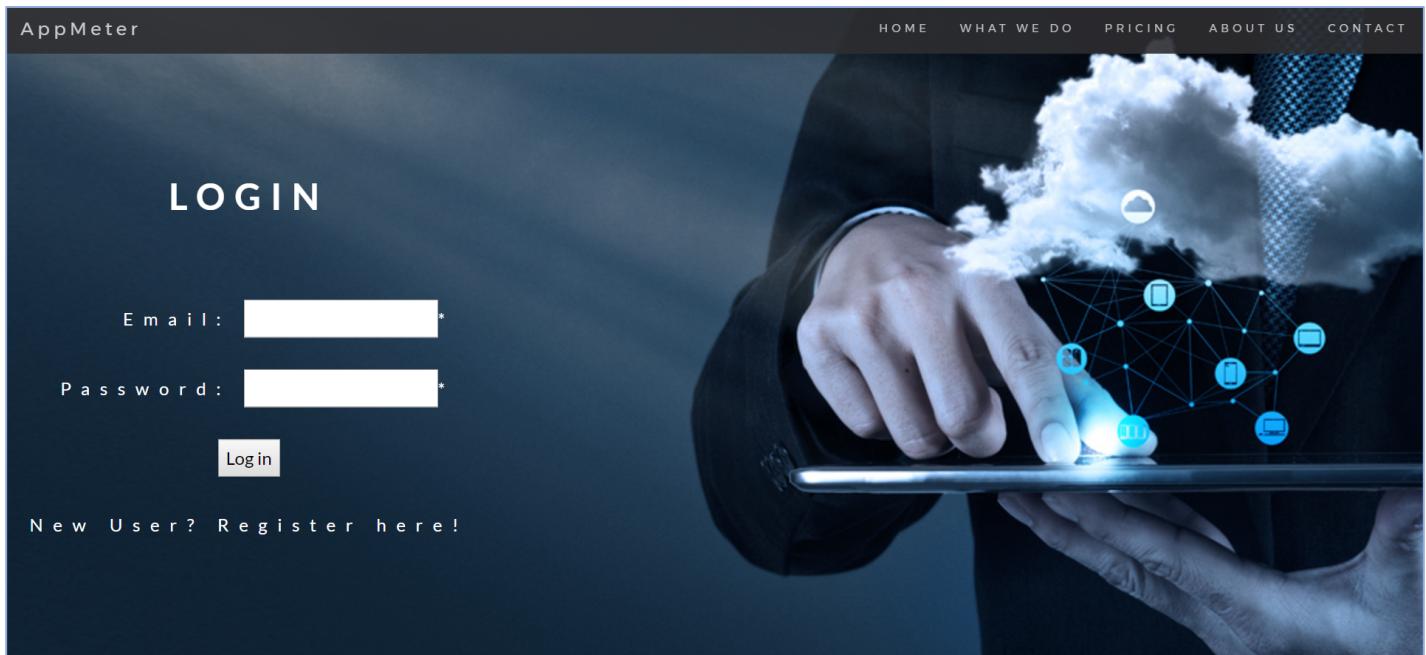


Design Considerations:

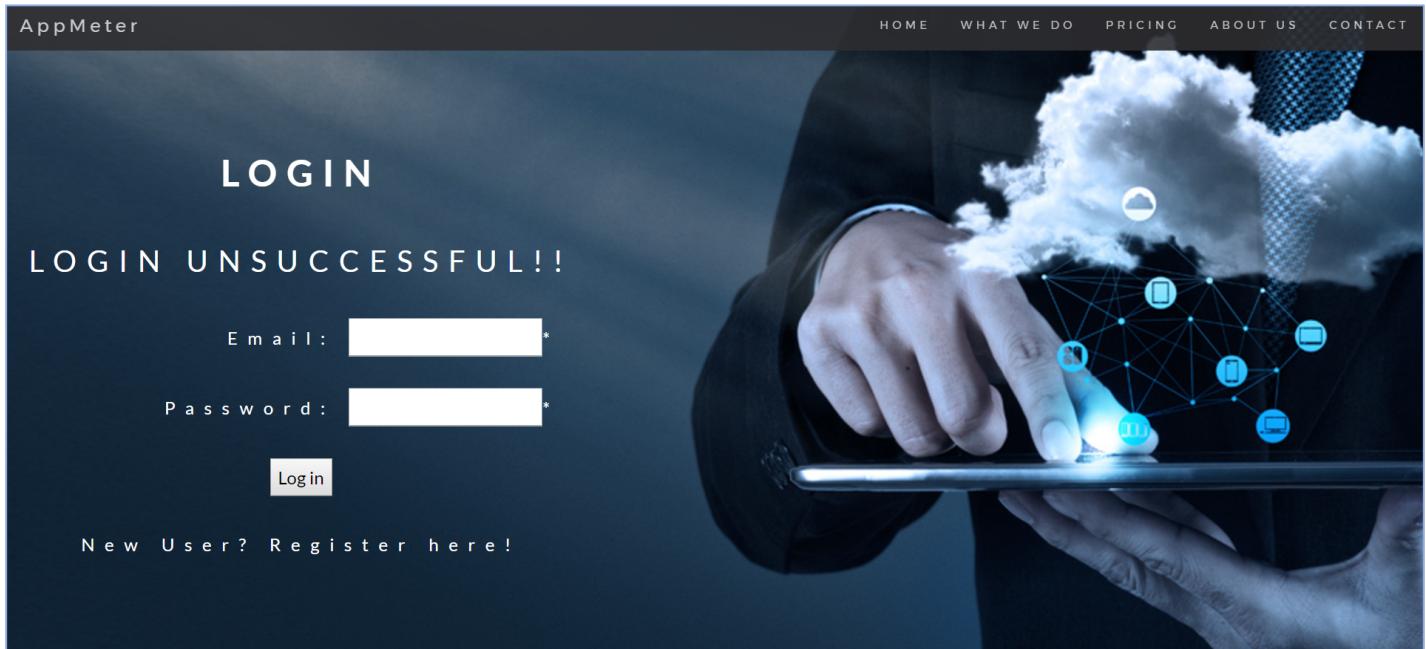
- The User is prompted if a field is left empty while submitting the form.
- Validations are applied to the textboxes to avoid incorrect user details.

16.3 Login

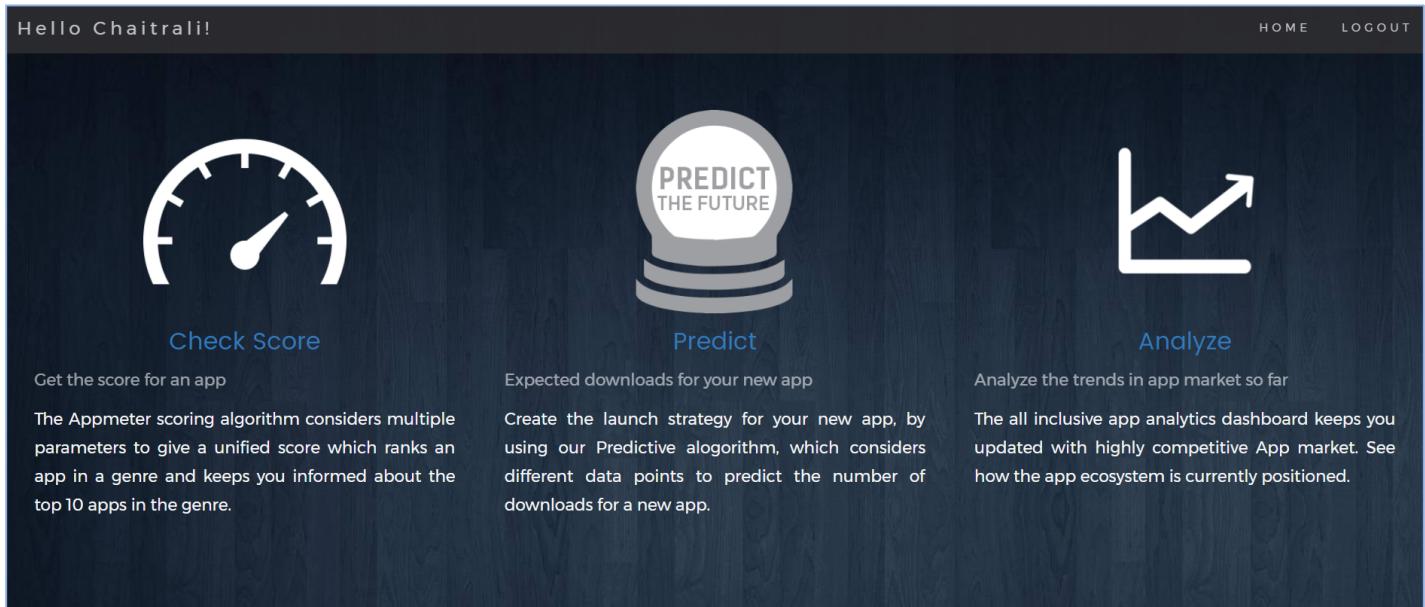
To access the premium features of predict and analyze, the user should be registered to the AppMeter system and be logged in. The system requires the user to enter his email address and password to login. If the user is new to the application, he has the option to register.



If the user enters invalid credentials, “Login Unsuccessful” message is displayed.



After successful login, the User is redirected to the “Features” page which displays the main functionality of the application. The User name is displayed at the top of the page to show the start of the session of the user.



Design Considerations:

- The User credentials are matched with the user information in the system database and message is prompted if they are incorrect.
- Navigation to register page is provided, so that new users can register effortlessly.

16.4 Find your app score

URL: <http://appmeter.atwebpages.com/searchscore.php>

This is a prime functionality of our web application where:

- Users can search an app from 450K+ apps by providing search keywords and applying filters for genre, price and average rating
- After getting suggestions based on the search keyword, user can get score by selecting the app.

The purpose of this functionality is to find out where an android app stands in comparison to the other apps in its own genre. To compute the score, the parameters - downloads, average rating, number of raters, age of the app have been considered. Also, users are provided with a list of top ten apps in the selected genre to help compare their app with the top apps and derive insights on the differing parameters that contribute to the popularity and success of these apps.

Search:

The search functionality allows the users to quickly find an app by simply entering the app name. The users can also perform advanced search by applying the filters available on the search page for the following - App genre, price range and Average Rating.

Design considerations:

Some of the design features that have been considered while developing this page are as follows:

Input features:

- The search box eases the search by displaying results that match the keyword entered and does not require the user to enter the entire/exact app name.

Control features:

- Pagination for search results is provided which reduces response time and enables easy browsing through the results
- Users can control the number of results displayed in a page
- Advanced search options using filters for genre, price and average rating is provided
- To help with a quick re-search, 'Reset search' feature is provided to clear the previous search

Snapshots:

Simple search:

Hello Shalinil

Search by filters

Category

- Books & Reference
- Social Networking
- Music
- Games
- Utilities

Price

- Free
- >0\$ - 10\$
- >10\$ - 50\$
- >50 - 100\$
- >100\$

Rating

- 0
- >0 - 2
- >2 - 3
- >3 - 4
- >4

Start typing a name in the input field below and then choose the filters on the left:

App name: Search

Search Results: 14

Rows per page: 10 ▾ Jump to page: [1] 2 > >>

 Instagram  Instagram Quotes PinQuotes

 Instagram World  Instagram Popular Tile

 Instagram Tips Tricks  Instagram Twitter followers

HOME PREDICT ANALYZE LOGOUT

Results for partial keyword entered:

Hello Shalinil

Search by filters

Category

- Books & Reference
- Social Networking
- Music
- Games
- Utilities

Price

- Free
- >0\$ - 10\$
- >10\$ - 50\$
- >50 - 100\$
- >100\$

Rating

- 0
- >0 - 2
- >2 - 3
- >3 - 4

Start typing a name in the input field below and then choose the filters on the left:

App name: Search

Search Results: 94

Rows per page: 10 ▾ Jump to page: [1] 2 3 4 > >>

 WhatsApp Messenger  WhatsApp Wallpaper

 WhatsApp Smilies Free  WhatsApp Offline

 WhatsFun WhatsApp smileys  WhatsApp Connect

HOME PREDICT ANALYZE LOGOUT

Advanced search:

For the keyword ‘business’, the filters applied are:

Genre - Travel, **Price** - Free and **Rating** - >2-3, >3-4, >4

The screenshot shows a web application interface titled "Hello Shalinil". On the left, there is a sidebar titled "Search by filters" with sections for Category (Games, Utilities, News, Travel), Price (Free, >0\$ - 10\$, >10\$ - 50\$, >50 - 100\$, >100\$), and Rating (0, >0 - 2, >2 - 3, >3 - 4, >4). The main area has a search bar with "App name: business" and a "Search" button. Below the search bar, it says "Start typing a name in the input field below and then choose the filters on the left:". The search results show "Search Results: 2" with "Rows per page: 10" and "Jump to page: [1]". There are two results displayed: "Business Travel Connection" with an icon of a briefcase and "BusinessChauffeur Oberding" with an icon of a stylized 'B'. Both results have a "Jump to page: [1]" link below them.

View score:

The view score page displays the score of the app with other metadata about the app.

This page displays the following:

- App score and rank in comparison with other apps in the same genre.
- App icon and name of the app
- Other metadata: App description, number of downloads, number of raters, average rating, genre, price and release date

- A graphical representation of the status of the app based on its score. Each app is categorized into one of the three zones - Red (Score <= 4), Yellow (Score > 4 to 7) and Green (Score > 7). This feature gives the user a quick comparison of the searched app with the other apps in the same genre.
- Top ten apps in the selected genre displayed as a slider and hyperlinks to Score page for these apps

Design considerations:

- The score displayed has been made intuitive by using color codes
- The score, rank and meta data can be viewed without scrolling the page
- Images have been used for displaying metadata which is quite intuitive
- Tooltips for metadata have been included to show which detail they refer to rather than displaying as plain text
- Top ten apps have been displayed as a tiny carousel making it easy for users to browse through the list
- The top ten apps are displayed as hyperlinks with a tooltip showing the rank and app name and take the user to the score page of the selected app
- A 'Back to Search' button has been included to facilitate navigation to the search page to continue search.
- The header provides links to other pages in the website for easy navigation.

Snapshots:



Hello Shalini!

2/9/2012 HOME PREDICT ANALYZE LOGOUT

App Description

Get WhatsApp Messenger and say goodbye to SMS!

WhatsApp Messenger is a smartphone messenger available for Android, BlackBerry, iPhone, Windows Phone and Nokia phones. WhatsApp uses your 3G or WiFi (when available) to message with friends and family. Switch from SMS to WhatsApp to send and receive messages, pictures, audio notes, and video messages. First year FREE! (\$0.99/year after)

WHY USE WHATSAPP:

- ? NO HIDDEN COST: Once you and your friends download the application, you can use it to chat as much as you want. Send a million messages a day to your friends for free! WhatsApp uses your Internet connection: 3G/EDGE or Wi-Fi when available.
- ? MULTIMEDIA: Send Video, Images, and Voice notes to your friends and contacts.
- ? GROUP CHAT: Enjoy group conversations with your contacts.
- ? NO INTERNATIONAL CHARGES: Just like there is no added cost to send an international email, there is no cost to send WhatsApp messages internationally. Chat with your friends

Check out the top ten apps in this genre!

Icon	Name	Score
	WhatsApp	10
	Instagram	9.82
	Skype	9.8
	Twitter	9.79
	Talk	9.45
	Phone	9.36
	Text	9.22
	Dolphin	9.16
	Go	8.99

This section displays a list of the top ten apps in the same genre as WhatsApp. The apps shown are WhatsApp, Instagram, Skype, Twitter, Talk, Phone, Text, Dolphin, and Go. Each app is accompanied by its icon and a numerical score.

16.5 Predict

This feature is particularly targeted towards the marketing-distribution team of the application.

This is one of the premium features of AppMeter offered to users who are interested in the prediction of the number of downloads.

The application developers will see how the file size, the expected average-rating and the days since the release affect the number of downloads of the App and how these parameters can be adapted to increase the downloads.

The marketing and sales team will be more inclined towards knowing the revenue produced by the app which is primarily related to the number of downloads of the app. The focus for them will be how the price and the genre of the application affect the downloads and can frame strategies to boost them.

The significance of the various parameters can be seen while predicting the number of downloads and can answer some vital questions about app development:

- How the downloads vary if the app is free or paid? Does paying price for an App guarantee that it will have features that are unique and beneficial to the user. Or making the app free will make it more accessible to the masses and increase the downloads.
- What should be the range in which the file size of the app be so that users are more likely to download it? Heavy Apps may be loaded with good features but sometimes the large file size discourages users to download it, reflecting the storage space and the RAM it will occupy on the phone.
- How the genre affects the downloads? Does the “Games” genre attract more number of users than the genre “Entertainment”?
- Does the number of downloads rise consistently with time? When can the downloads cease to increase and may be a new version or update is required to give downloads the needed boost?
- What is the impact of average ratings on the number of downloads? Does high average rating show that the app is successful or indicate less number of raters?

Taking all the above questions into account, the predict functionality gives you an estimate of the number of downloads for definite parameters.

Snapshots:

The screenshot shows the 'PREDICT - COMPARE' section of the AppMeter application. At the top, there is a dropdown menu labeled 'Genre' with 'Social Networking' selected. Below this, there are two columns of input fields for comparing two apps. Each column has four rows: 'Price' (with a dropdown placeholder 'Price in \$'), 'Size' (with a dropdown placeholder 'File Size in Mb'), 'Average Rating' (with a dropdown placeholder 'Average rating out of 5'), and 'Age' (with a dropdown placeholder 'Days since the release of the App'). At the bottom of each column are two buttons: a blue 'Predict' button and a white 'Reset' button.

The User will enter the parameters for which he requires the predicted number of downloads. By clicking on the Predict Button, the estimated number of downloads and the chart showing the trend in the downloads over 4 years after the release of the application is displayed.

The Reset button clears the entered parameters, the result and the chart to allow the User to utilize predict functionality again.

Extending this functionality is the option to compare: where the user can specify two sets of parameters and compare and contrast the effects of them on the number of downloads. The Second Column is used to enter the other set of parameters for comparison of the estimated number of downloads.

AppMeter

HOME SCORE ANALYZE LOGOUT

PREDICT-COMPARE

Genre

Social Networking

Price

0

Size

100

Average Rating

3

Age

600

Price

4

Size

150

Average Rating

3

Age

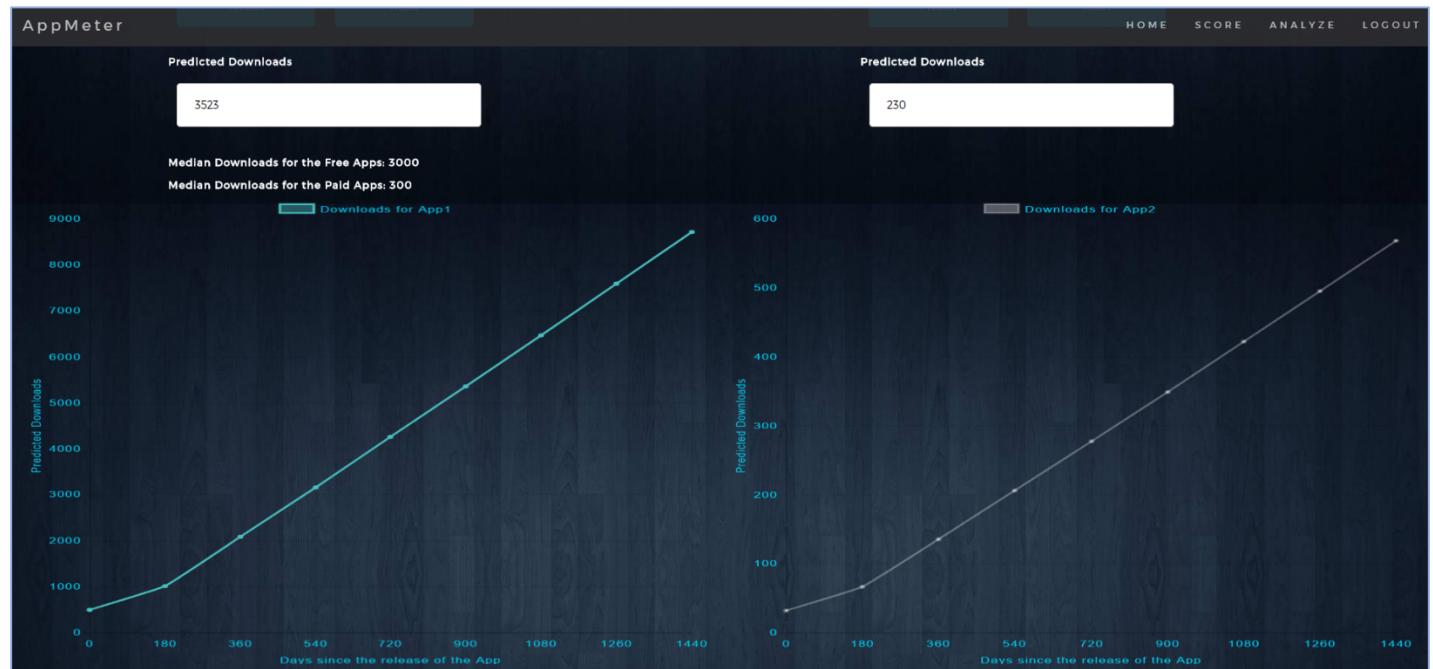
600

Predict

Reset

Predict

Reset



Design Considerations:

- Simple and User-friendly: The User should be able to enter the parameters effortlessly and efficiently. The drop-down menu for the Genre makes it easy to select the genre and the placeholders help the user understand what type of input is expected in the textbox.
- Charts have been used to show the change in the number of downloads over a period of time. This helps the user to visualize the pattern in the downloads.
- Reset Button has been provided to let the user easily reset the parameters.
- The second column of the Predict functionality is provided for the User to compare two sets of parameters for the number of downloads.
- Input Validation has been performed for the input text boxes which will prevent the User from entering invalid input and submitting an empty form.
- Both the columns perform the predict functionality independently and either or both can be used to get the predicted number of downloads.

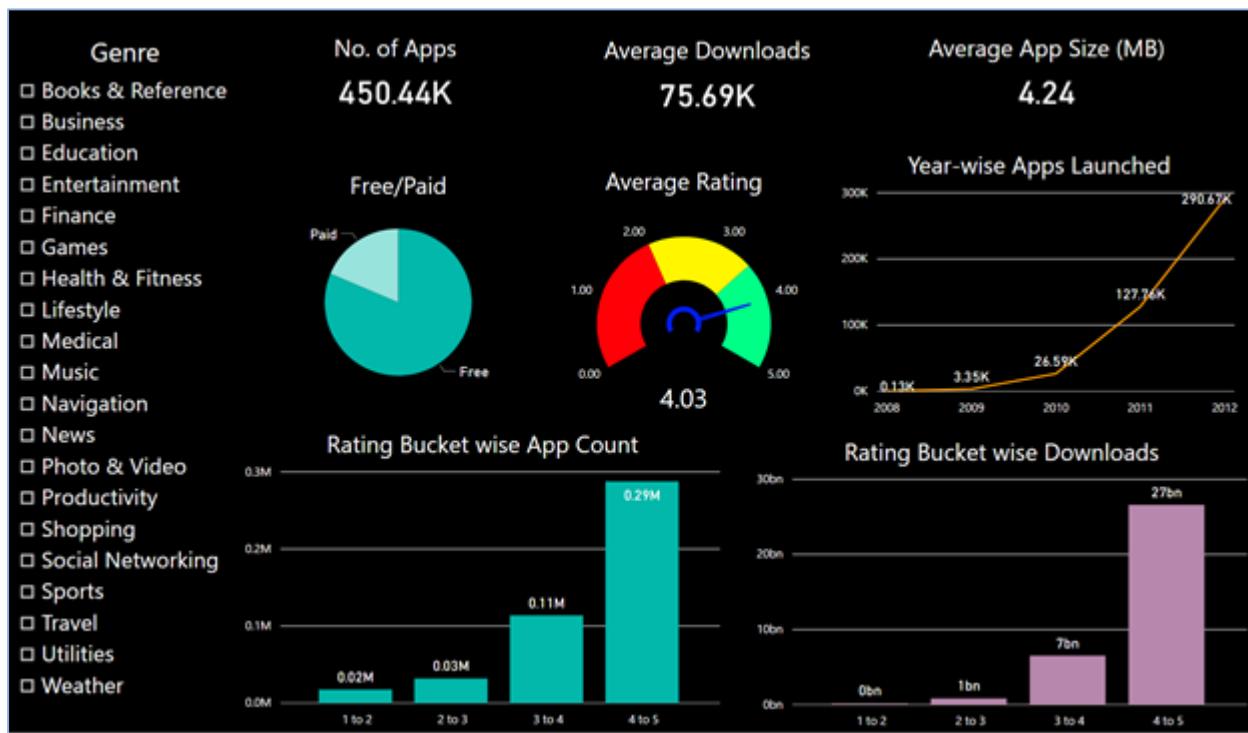
16.6 Analyze:

This is one of the premium feature of AppMeter where the user is exposed to all the analytical insights about the App ecosystem. All the 450,000+ applications and their detailed analysis is available on just one click. The user-friendly and intuitive dashboard can get the answers to App-ecosystem related questions on the fly!

For example,

Even without applying any filter, user can get answers to questions like-

- What's the average number of downloads in the App store?
- What's the average file-size of the apps?
- Out of total 450,000+ apps, how many apps are free? How many are paid? What's the percentage of Paid and Free apps?
- How many of the total apps have good quality? (Average rating 4+)
- Show me time series of apps launched over the years.

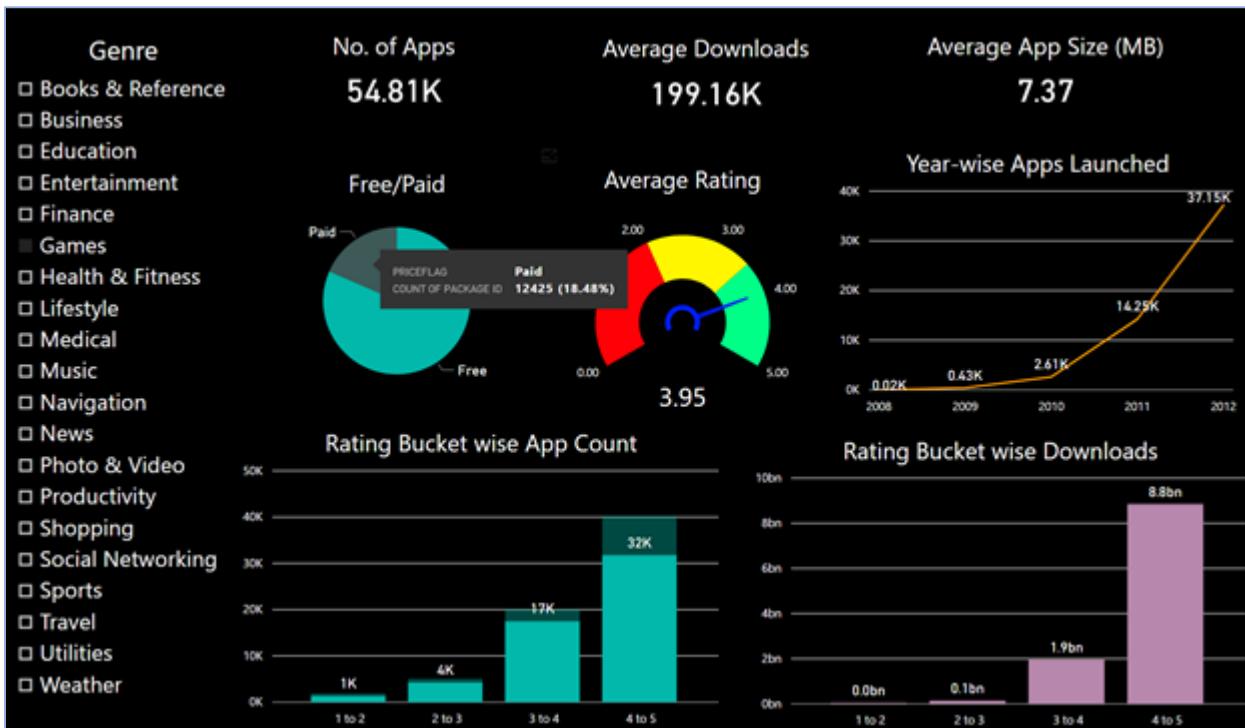


Now, we can apply filters for Genre, Free/Paid, Rating bucket, and get statistics like Average number of apps, Average downloads, Average app size, number of apps launched in recent years etc.

We can get answers to some complex questions like-

- What's the average downloads for Apps in "Social Networking" genre which are free and have average rating between 3 and 4?
- What's the average file-size for apps in "Business" Genre and which are highly rated (4+)?
- What's the average downloads for "Games" Genre and apps which are free and poorly rated (1-2)?
- What's the number of apps launched over years in certain genre and are paid?
- Does poorly rated "Games" also have high downloads?
- Apps in "Social Networking" genre have higher average downloads; but what's the average rating for that genre? Is the file-size for those apps very less? Is the average file size for social networking apps less than that of apps in all the genres combined?

For Example, Games Genre, paid apps statistics



Design considerations:

- Modification to the Data

As any executive would like some insight instead of just numbers, we made buckets for the app data in terms of Ratings and File-size. Also, price should have a flag-indicator as Paid or Free. We created the buckets using conditional columns function of the Power BI and used those fields in the dashboard. Due to data modifications, information delivery to the targeted user is made efficient.

- Ease of Use

The dashboard does not involve any repetition of information. All the complex questions can be answered by just a click. Resetting the filters is easy. The response time for the dashboard is extremely low (almost zero). The integration of the web application with Microsoft Power BI is very simple and newly added apps in the ecosystem can be maintained easily by the administrator on the backend; independent of the user.

17 Salient Features of the Web Application UI

17.1 Session Management and Security

Every web application has end-to-end session management functionality. The user activity is maintained in a session after the user logs into the system. Until the user logs out, the session is maintained. As we have two types of users and the access control is based on the user pricing model, we have incorporated session management for every page on the website. For example, the website contains premium content like Predict and Analyze; while the score functionality is accessible to all users. The session management technique allows us to know if the user is already logged in to the system or not. We start a session every time the user logs in and the session is maintained until the user logs out. The benefits of doing so is that we can restrict the guest users from accessing the premium content.

The AppMeter Score function is available to both, guest and registered users. Using session management, the menu bar item list is customized on the score page. Depending upon whether the user has logged in, user can view premium services like Analyze or predict.

Another security consideration is about the direct link access. If the user directly inputs link to any page which is a premium feature of AppMeter, the user is redirected to login page and prompted to log in or register (if not already registered!) So, direct links are protected and only premium users can access premium content.

17.2 Responsive Website Design

With an exponential increase in the use of viewing web content on mobile phones or tablets, the web developers have an added responsibility to make the website responsive to all devices. The inclusion of Bootstrap libraries for responsive web design have helped the AppMeter UI team to support any screen size. The content is now adjusted according to the screen size and the usability and user experience is not affected or sometimes, enriched. So, the user gets the same look, feel and experience while working on different devices.

18 Challenges and Learnings

18.1 Challenges

- The biggest challenge our team faced was getting the right data. We had challenges getting data from the official android store as Google didn't had any official API to get the data. We also tried using web scraping, but that didn't get us the whole data.
- The scope of the project has changed as we made progress. This included new challenges and we had to modify the data as per these changes.
- The data had more than 500K records, so we had to do a lot of data wrangling before we could derive insights from it. So, understanding what data will be used for the project and how to transform the non-refined data was an important aspect of our project.
- It was a challenge to understand the results from the statistical model analysis. Most of the team had basic knowledge on Statistics, therefore to decipher the results of the machine learning algorithm was challenging.
- It was challenging to understand the impact of the parameters on the model and to identify which statistical model could give us desirable results.

18.2 Learning

- It is important to have a clear understanding of the requirements before you start the project plan. If the requirements are not clear, it's better to ask questions.
- Learnt different machine learning models like DecisionTreeRegressor, PCA, 2SLS etc.
- Data analysis in python using Numpy, Pandas.
- Technologies like HTML, PHP, CSS and JavaScript for the front-end development.
- We implemented the SDLC standards and practices throughout our project like SCRUM and prototyping. This helped us understand the processes by practical working.
- We used GitHub for code integration and version control and thus learnt how to use the industry common practice.

19. Future scope

- Consider **feature analysis** for predicting success of App. This is important because if two Apps have comparable parameters, then the user will choose an app depending upon its features. Sometimes even if the app is poorly rated, it will have more downloads since it provides unique features. This analysis needs to be done per genre, and can be compared with the top apps in that Genre. This will require NLP and will be an advance functionality for this project.
- Consider **twitter sentiment analysis** while deciding the App rating. Twitter has become a valuable resource to collect the user sentiments. Considering the large number of active users on Twitter, this platform cannot be avoided. The challenge will be to distinguish the Tweets which are about the “App” or about the Business model itself.
- Implementing **enhanced security measure** on our system, like disallow same login from multiple platforms at the same time. This will provide additional security and avoid login from BOTS.
- Integrate payment gateway with AppMeter for registered users
- Currently AppMeter is a web based solution, we can have a **app version** for the same.
- Consider expanding AppMeter system for **iOS platform** (Apple) mobile Apps

APPENDIX

Relevant Coursework for Capstone

The courses that have been helpful in implementing the technologies that match our objectives are outlined as below:

MSIS2603 Database Management Systems

This course provided hands-on experience with database design and SQL programming. Our data store needs to have metadata on Apps. Also, it needs to be designed efficiently as our search application is linked to the database and from the usability point of view, the data retrieval should be seamless.

MSIS2802 Data Science Analysis w/Python

This course provided a hands-on experience in using Python, especially analyzing large data set. We were also exposed to the machine learning package (scikit-learn) which helped us in predictive analytics and creating the regression model.

MSIS2602 System Analysis and Design

This course taught the methodology to analyze different functional and nonfunctional requirements to create the use cases, data flow diagrams and program chart diagrams that are used for system integration and Design.

MSIS2606 Software Project Management

This course taught us the best practices followed in software project planning, requirements gathering, design, and implementation of a software project. We have used SCRUM methodology in our project that helped us to use iterative cycles to work on our project. Our Scrum meetings helped all the team members to understand

how each one is progressing and if there are any show stoppers. We also created a project timeline and adhered to that.

MSIS2630 Web Programming

This course covered concepts of HTML/CSS, PHP, JavaScript and Ajax. The knowledge about web designing techniques acquired from this class was used to build the user experience.

MSIS 2629 Dashboards, Scorecards and Visualization

This course introduced dashboards and scorecards for presenting and analyzing critical performance and market trends. By using the concepts from the class, we created the dashboard for the app owners to check the trends in the app ecosystem.

Other Skills requiring additional study

1. Although we have used SQL as the basis for database queries, however SQL also have different flavors as per the database vendors, so we had to be updated on the details specific to the backend technology we chose-MySQL.

2. We studied various literatures with regards to app analytics for predicting score and creating the coefficients for each required parameter. These literatures encompass statistics to create those models. So, we updated ourselves with the required knowledge in Math, statistics and econometrics for the project.

3. Our project has a lot of dependency on data, therefore we researched on the different APIs and commercial vendors who provide relevant data.

4. We studied the Statistical and Machine learning libraries of Python and R, and used them to create our models.
5. We used Github for code repository, and therefore we had to study how to use Github and how version control is established.
6. We used AwardSpace to host our web application and database. We also understood how the system works, how to create and deploy the infrastructure needed using its platform. Web server and MySQL database was managed using PhPMyAdmin on Awardspace Database Manager.
7. We researched on the mobile App industry, what are the parameters that affect customer behavior, and important features for the success of an app.