RV COLLEGE OF ENGINEERING®

Bengaluru – 560059

(Autonomous Institution Affiliated to Visvesvaraya Technological University, Belagavi)

Assignment Report

On

"Student Database using Firebase"

Submitted by

Lakshitha Karthik(1RV18SIT09)
Meghana V (1RV18SIT12)
Shilpa V Swamy (1RV18SIT14)

Under the Guidance of

Prof. Anisha B S
Assistant Professor
Department of ISE, RVCE

In partial fulfilment for the award of degree of Master of Technology

in

Information Technology

Information Science and Engineering

2019

RV COLLEGE OF ENGINEERING®

Bengaluru - 560059

(Autonomous Institution Affiliated to Visvesvaraya Technological University, Belagavi)

CERTIFICATE

Certified that the project work titled **'Student Database using Firebase'** is carried out by **Lakshitha Karthik N (1RV18SIT09), Meghana V (1RV18SIT12) and Shilpa V Swamy (1RV18SIT14)** a bonafide students of RV College of Engineering®, Bangalore, in partial fulfilment for the award of degree of **MTech** in Information Technology, Information Science and Engineering, during the year **2019-2020.** It is certified that all corrections/suggestions indicated for the internal Assessment have been incorporated in the report deposited in the departmental library. The report has been approved as it satisfies the academic requirements in respect of assignment work prescribed for the course by the department for the said degree.

Maximum marks	Marks obtained

Signature of Course In-charge

ABSTRACT

Android Studio is the official IDE for android platform development and is freely available under the apache license. It is designed specifically for Android development and I am using it to develop the app, as thus far the app is exclusive to android.

Google Firebase is a mobile platform for helping to develop app and increase your user base. It's made up of features that can be mixed and matched to fit the needs of the user. Most of the features on Firebase are free to use.

Our app "Student Database Using Firebase" helps in educational institutions, where the mandatory processes with student details like - collection, management, and maintenance are done for the student record. Since the management and maintenance part of the data is cumbersome for any educational institution there are many built software and applications. Student Database Using Firebase is an android application developed to enable ease of work in such managing and maintaining processes. Our project helps in creating the database of the student which maintains the record of student Name, USN, Specialization and Department Name.

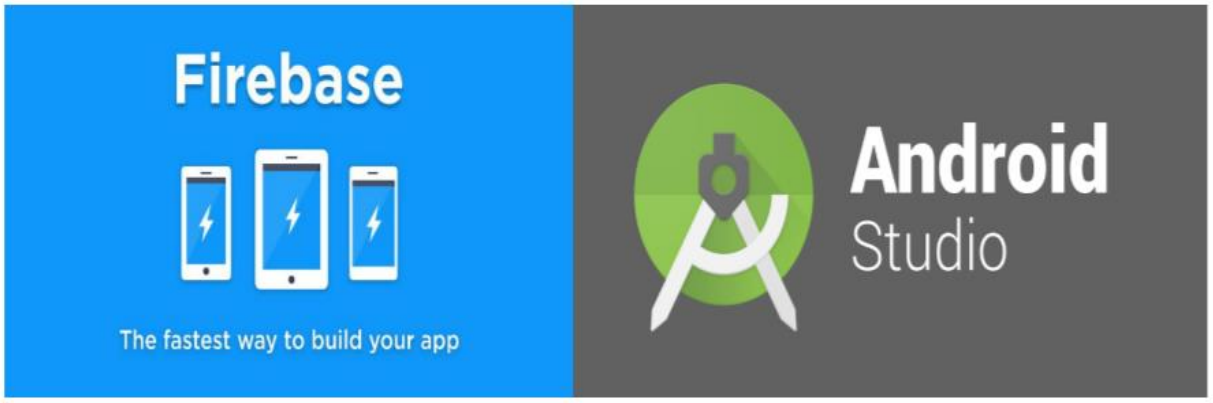
INTRODUCTION

Android Studio and Firebase

Android Studio is the official IDE for android platform development and is freely available under the apache license. It is designed specifically for Android development and I am using it to develop the app, as thus far the app is exclusive to android.

Google Firebase is a mobile platform for helping to develop app and increase your user base. It's made up of features that can be mixed and matched to fit the needs of the user. Most of the features on Firebase are free to use.

We will be using Firebase for app sign in options and authentication and to maintain a student database



Detailed Design and Architecture

The following is the system architecture representation, the application itself will be hosted locally on a device that runs Android OS. The application will use Google Firebases infrastructure and implement Firebase code to use their secure database for storing user information and handling user messages. For a project of such a low scale compared to real life companies (Vodafone, 3, Meteor) this was the best approach to provide secure real-time messaging. I as Firebase admin to this project have sole access to user information, passwords will be encrypted and not visible to anyone including myself.

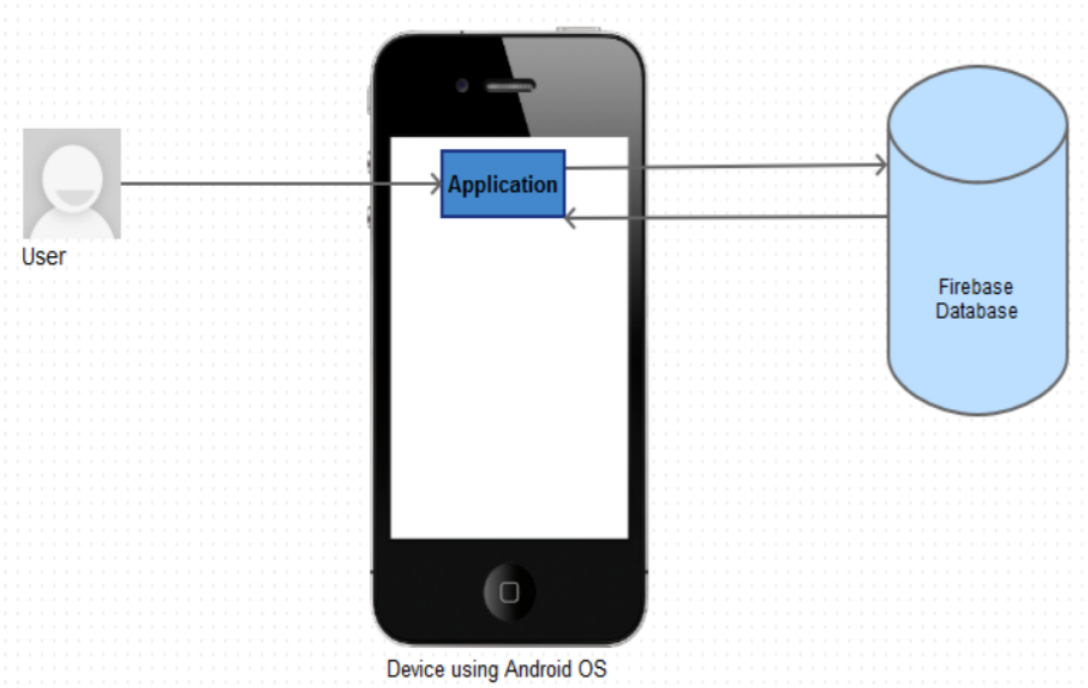


Figure 1.1: Design Architecture

The entire lifetime of an activity happens between onCreate() and onDestroy(). The activity is visible between onStart() and onStop(). The foreground lifetime of the activity happening when the user interacts with the application is between onResume() and onPause().

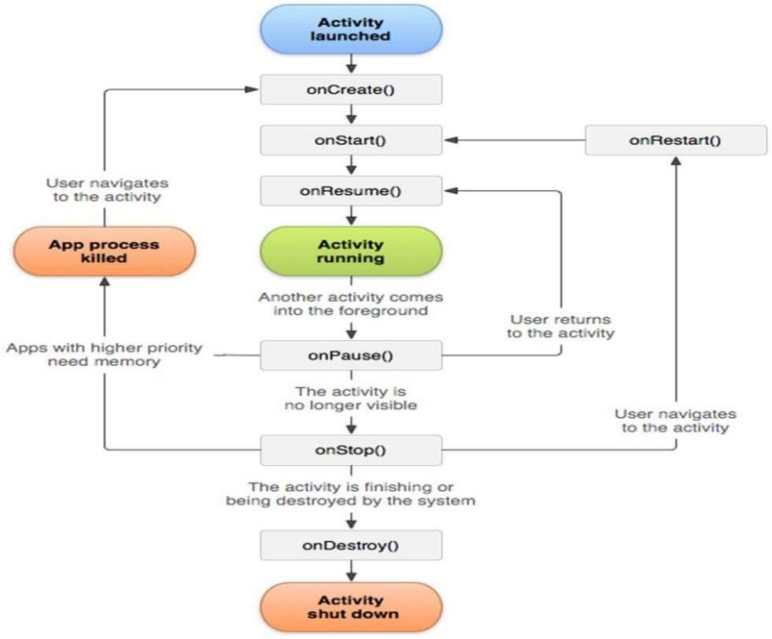


Figure 1.2: Activity Lifecycle

LITERATURE SURVEY

In this paper [1], the author Introduces the Android platform and the features of Android applications and gave a detailed description of Android application framework from the prospective of developers. A simple music player is provided as instance to illustrate the basic working processes of Android application components. This paper could provide guidance to understanding the operation mechanism of Android applications and to developing applications on Android platform.

In this paper [2], the author studied that the learning and building an Android application can be both easy and hard at the same time due to all the figures and facts mentioned throughout the thesis. It takes time and requires the developers to deeply understand what they are dealing with. The main purpose of this thesis is to develop a simple REST API Android application. All the information leads to the result that the application that searches venues near users' current location is built. Throughout the process of making an application, the author has acquired more knowledge in Java programming, Android development as well as how to publish an application to Google Play store.

In this paper [3], the author describes about how the android application provides a full overview on how to save the data in the application, along with other operations as such sorting, searching, changing themes, etc. The UI of the application is made attractive to provide the user a good soothing experience. Widgets and other features as such voice navigation and searching for the user is available.

Chapter 3

System Requirement

Functional Requirement

Navigation:

- The application must make seamless transitions between activities or pages.
- Different pages must be defined clearly for the user.
- Features must be clearly visible to users.
- Available options on each page should be clearly visible to users.
- Instructions for Auto Message should be available within the pages.

Database Connection:

- Database must be constantly available for new student registration.
- Database must be updated upon new student registration.

Profile:

- The user shall be able to create a table for each student with student name, USN, Specialization and Department name.

User Requirements

- A user shall be required to have WI-FI enabled on their smartphone/device
- A new user shall be required to register or use a provided login method to gain access to the key features of the app.
- Users shall be able to use the app from any smartphone/device that uses Android Operating System 2.3 Gingerbread or higher.
- A user may be able to set a location to be associated with a WI-FI network that the user has connected to.

Environmental Requirements

- A WI-FI network must be available.
- WI-FI network signal strength must be at a level where a device can make an authenticated connection.

Requirement 1: Start Application

Description & Priority

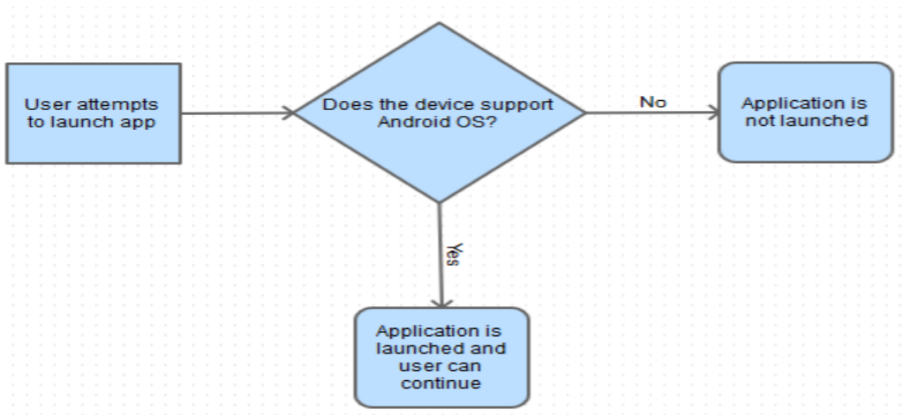
This use case is first in the priority list and describes the key functions that are offered to the user on starting up the application. If the user does not have the correct hardware they cannot launch the application.

Use Case

Scope

The scope of this use case is to describe how a user can launch the application and access the primary functions that are displayed to the user.

Use Case Diagram



Precondition

The application was downloaded and installed on a smart device that supports and runs Android OS.

Activation

This use case is activated when the user clicks on and starts up the application while in its idle state.

Main Flow

- Actor Action: User clicks on the application launcher icon in an Android device menu.

- System Response: Application is launched.

Alternate Flow

- Actor Action: User clicks on the application launcher icon in an Android device menu.
- System Response: The application does not launch due to a system crash or the OS is not compatible.

Termination

This flow is terminated when the application has booted on the device.

Post Condition

The user is brought to the main menu of the application where the system enters a state of waiting

Requirement 2: Student details entry

Description & Priority

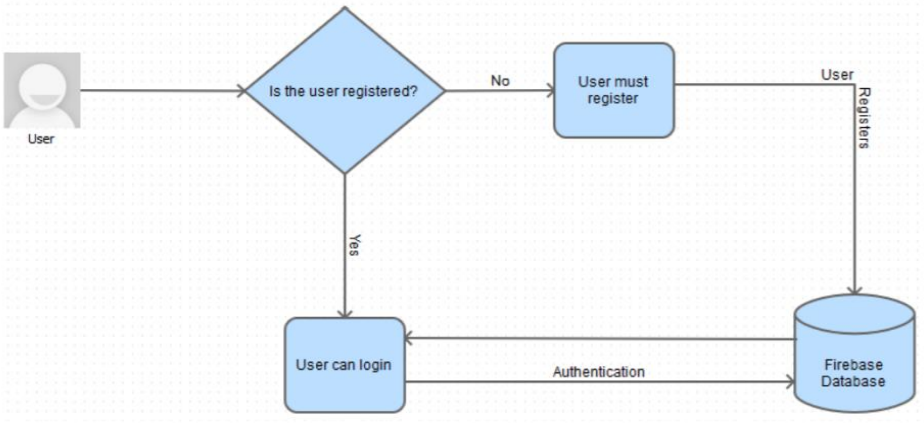
This use case is second in the priority list and describes the process of student registering, once the application has booted up the user must be able to register.

Use Case

Scope

The scope of this use case is to describe how the user can begin registering the student details.

Use Case Diagram



Precondition

The application has been booted up and is running in a waiting state.

Activation

This use case will start when the user has launched the application.

Main Flow

- Actor Action: User clicks on the Enter student details
- System Response: Student detail page (name, USN, specialization, department) displayed for user.

Alternate Flow

- Actor Action: The user is not registered and clicks on login.
- System Response: The user is prompted to create an account.
- Actor Action: The user does NOT click on login.
- System Response: Login no register tasks will be launched.

Termination

This use case is terminated when the user enters the student details successfully.

Implementation

Implementation languages that will be used in this project are mainly java which is the language used when developing in Android Studio. Google Firebase for the database to store the student information. We can use Firebases authentication code to enable users to sign in using their Google account, to me this made the most sense because the app is Android specific. For a user to download an Android app they need to access the Google play store, for this, a user needs a Google account which would guarantee that if a user can download the app to a device then they can sign in. Anything else at this point is unforeseen but there may be other languages used for implementation of various aspects of the project that can be added in the later stages of development.

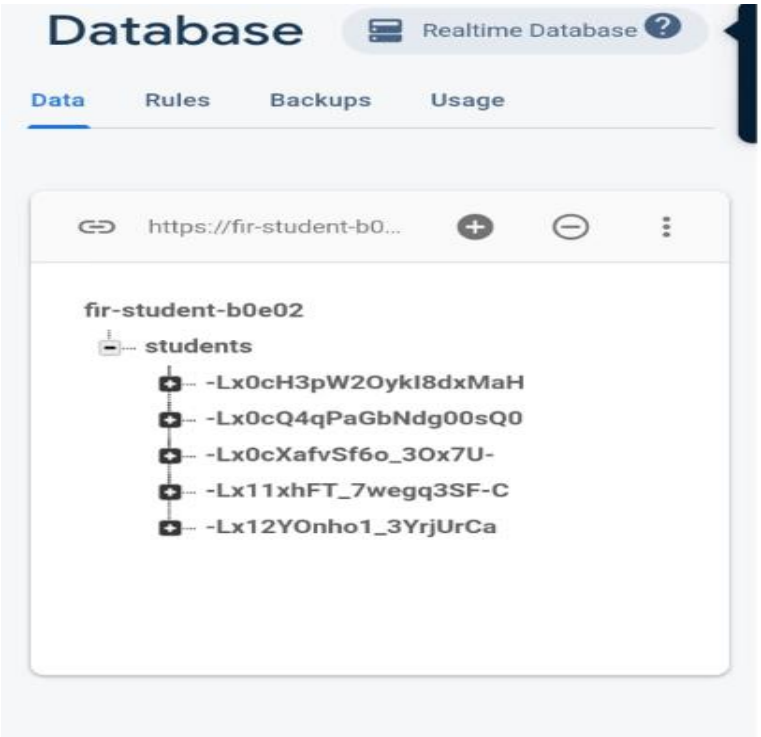
Firebase Authentication Firebase authentication uses SDK’s and ready-made UI libraries to authenticate users to my app.

Sign Up Code for Firebase

```
if(FirebaseAuth.getInstance().getCurrentUser() == null) {
    // Start sign in/sign up activity
    startActivityForResult(
            AuthUI.getInstance()
                    .createSignInIntentBuilder()
                    .build(),
            SIGN_IN_REQUEST_CODE
    );
```

Firebase Database

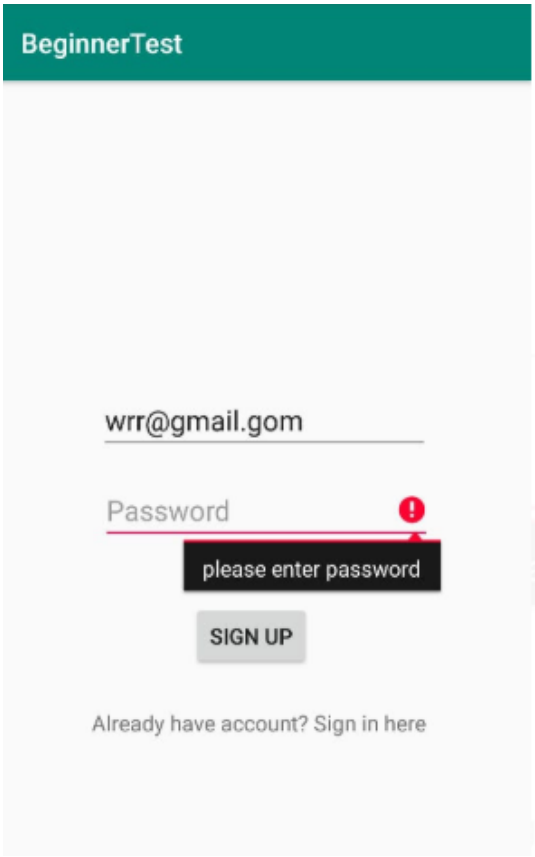
Not only is the database being used to store user information, but it is also being used to store user text messages and send them back to the application for all users to see. Below is an example of the message and user structure as seen on Firebase and further below that is a screenshot of the same conversation within the app.



Create an Account

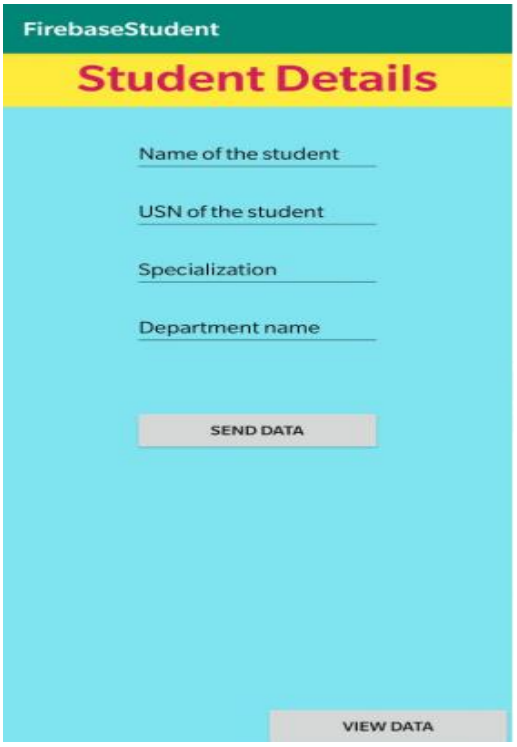
The creation of a new account on the app is done by giving the email and password. The app ran smoothly and worked as it should, the tester had no difficulty during the performance of this task. This leads us to believe that interface testing has been successful.

Step 1: User enters email address to be used in account setup, if the email is not recognized in the database, the tester is redirected to create an account.

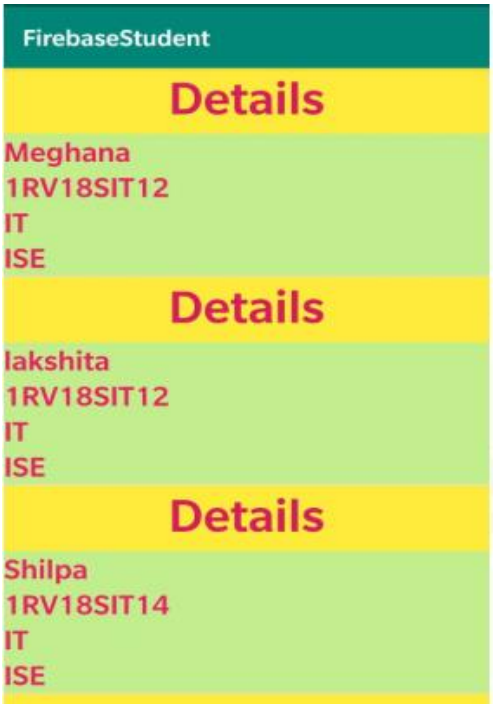


Step 2: The User is redirected to create an account page as the email address is not recognized as an existing user

Step 3: The User enters the Student details in the Firebase. Where the User can send the data to database or view the data from the database.



Step 4: If the User choose to View the data, the View Data button has to be clicked, which intern redirects the view page.



Step 5: If the User choose to Send the data, the Send Data button has to be clicked, which send the entered data to the firebase.

FUTURE ENHANCEMENT

The App can be used in the educational institutions with additional features adding to it like storing the details of student's marks, grades and other details related to the department. The institute can fetch the details of the students by entering their unique USN. The integration with the student login application can be done.

CONCLUSION

Learning and building an Android application can be both easy and hard at the same time due to all the figures and facts mentioned throughout the thesis. It takes time and requires the developers to deeply understand what they are dealing with. The main purpose of this thesis is to develop a simple REST API Android application. All the information leads to the result that the application that searches venues near users' current location is built. Throughout the process of making an application, the author has acquired more knowledge in Java programming, Android development as well as how to publish an application to Google Play store. However, while developing an application for mobile devices, the problems lie in the mobile device itself.

To be more precise, bandwidth, hard drive space and memory, or battery power are all the drawbacks. What is more, there are too many Android OS versions existing. This brings out the situation where almost modern Android phones work much smoothly compared to the old versioned ones. These problems are general issues that no one could fix them completely, but developers do have the ability to restrict some parts of them such as coding for the mobile devices to use less batteries, or to spend less hard drive spaces and memories. Long story short, the principle objectives of this thesis, which are to develop an Android application with REST API, have all been accomplished.

Chapter 7

REFERENCES

[1] www.idc.com. (2016). IDC: Smartphone OS Market Share. [online] Available at:

[2] http://www.idc.com/prodserv/smartphone-os-market-share.jsp [Accessed 18 Nov. 2016].

[3] Hathibelagal, A. (2017). How to Create an Android Chat App Using Firebase. [online] Code Envato Tuts+. Available at: https://code.tutsplus.com/tutorials/how-to-create-an-android-chat-app-usingfirebase--cms-27397 [Accessed 24 Jan. 2017].

[4] www.tutorialspoint.com. (2017). Android Broadcast Receivers. [online] Available at:

[5] https://www.tutorialspoint.com//android/android_broadcast_receivers.htm [Accessed 10 Feb 2017].

[6] Developer.android.com. (2017). ScanResult | Android Developers. [online] Available at:

https://developer.android.com/reference/android/net/wifi/ScanResult.html [Accessed 20 Mar 2017].

[7] Programcreek.com. (2017). Java Code Example android.net.wifi.ScanResult. [online] Available at:

http://www.programcreek.com/java-examples/index.php?api=android.net.wifi.ScanResult[Accessed 25 Apr. 2017].

[8] Developer.android.com. (2017). Android Developers. [online] Available at: https://developer.android.com/index.html [Accessed 16 Oct. 2016].

[9] YouTube. (2017). TVAC Studio. [online] Available at: https://www.youtube.com/user/akshayejh [Accessed 6 Nov. 2016].