University of Arkansas – CSCE Department
Capstone I – Final Proposal – Fall 2019

FoodAlert

Gabriel Priest, Nick Waterworth, Jasper Harrison, Bentley Lager, Alycia Carey, Jiamin Lin

Abstract

Food waste is a major issue in America. It is estimated that between 30–40 percent of food in the United States goes to waste. Of this, 31 percent of food loss is at the retail and consumer levels; this corresponds to 133 billion pounds and $161 billion worth of food waste [1]. We are not only wasting food that could end global hunger but also burning up Earth's resources. All of the labor and resources used to produce wasted food is wasted as well. The objective of FoodAlert is to limit the food waste of application users by planning out food expiration and usage. FoodAlert will be a platform for users to see their food usage trends and plan their most optimal food usage. FoodAlert will also allow alarms to be set for expiration deadlines to help cleanliness of the kitchen environment and prevent pests.

FoodAlert will help consumers by allowing the transfer of information from food purchases onto the FoodAlert calendar and database. In the application, users will see when certain food items expire, plan out meals, and analyze their own food waste. By facilitating user planning, FoodAlert will enable consumers to monitor how much produce they purchase and cut back at food waste in households. The food saved with FoodAlert can be left up to the retailers for distribution to the homeless or to farmers to be used as fertilizer, minimizing the food that ends up in landfills.

1.0 Problem

Every year Americans waste over 133 billion pounds of food. This equates to 218.9 pounds for every person in America. This leads to over 161 billion dollars wasted every year that could be used for other essential purchases. This food could be used to feed thousands of families and the money could be used to house and cloth those families as well. The USDA has set a goal to cut food waste by half before 2030; FoodAlert will assist in addressing this issue [3].

Rotten food leads to disease and can ruin other foods that it comes into contact with. This rot can lead to further waste that continues to compound upon itself. With the issues that the world has currently such as global warming and pollution, every little bit will help. If we

continue to waste food at the level we have been for the past several years, the world could be irreversibly harmed [1].

Food waste is an exponential loss of resources, not only in America but globally. According to FOA (Food and Agriculture Organization of the United Nations) [5], “Global quantitative food losses and waste per year are roughly 30% of cereals, 40-50% for root crops, fruits, and vegetables, 20% for oilseeds, meat, and dairy plus 35% for fish.” Despite the production of the world’s cereal crops in 2009/2010 is about 2.3 million tonnes, it is still less than the total amount of food wasted every year.

2.0 Objective

The objective of this project is to create an application that will help users to keep track of food expiration dates, and reduce the amount of food wasted. Most people will put food in the corners of their pantries or refrigerator and forget that they even have it there, letting it expire or buying more products. Cleaning up after expired food is a pain and even though it is a small problem, it is still an inconvenience that needs to be addressed. The application will allow the user to take a photo of the food items and store the information in our database. The application will include an alert system and a calendar with expiration dates which will help people easily see as food is reaching the time where they should be disposed of. This will also help when people go grocery shopping, they could check with the application on what food is still available in the house. Also, waste less food by having a physical list of their wasted food which will keep them accountable. We also hope to have a receipt parser which will make the application very easy to use, so there will be almost no excuse not to use the application.

3.0 Background

3.1 Key Concepts

ReactNative [6]: ReactNative is a platform used to create native apps using React for Android and iOS. React is a javascript library that is used as a tool to build UI Components. The basis of React is the UI components that are then rendered within the HTML DOM. This Framework is supported by Facebook and is continuously under development to streamline the development process. Animations and multiple fragments are included in this framework to make the production of applications extremely simple. Fragments allow you to group a list of children elements without adding extra nodes to the DOM. To create application elements, you must declare each one, and render the element with a render() method. Each rendered element is immutable and must be rerendered each time it is changed. Due to how components are rendered, the application can be used on any platform seamlessly. The development cycle of ReactNative applications also allows for an almost instant view of changes and allows for simple debugging as problems arise. The difference between React and ReactNative is the type of components that are used within the application. React uses web components and ReactNative using native components. There are plenty of benefits to choosing ReactNative vs native development for Android and iOS. React is known for optimal performance because it is connected to native components for both operating systems. The framework also has support for a wide range of third party plugins due to the lack of some components in the main framework.

For example, if the application needs to have any Map, ReactNative allows the developer to connect to a plugin with a native or third-party module.

Firebase [2,10]: Firebase is Google's mobile application development platform that houses a variety of analytic, authentication, database, configuration, file storage, and push messaging tools. All of the services are hosted in the cloud and scale easily without much hassle from the developer. Client SDKs provided by Firebase allow developers to interact directly with the backend Google services with no need for middleware between the app and service. Firebase focuses on three main mission points: “Build Your App”, “Improve App Quality”, and “Grow Your Business”, that couple together allow for the creation of a steady and powerful application.

Build Your App	
Name	Usage
Authentication	User secure login and identity management, restrict access to per-user data. Combined with Realtime Database, Cloud Firestore, and Cloud Storage can be used to set security rules and access control to data at the source. When the user signs in, they are assigned an identification token that can be used in rules to protect who can read and write which items of data.
Realtime Database	realtime, cloud-hosted, NoSQL database
Cloud Firestore	realtime, cloud-hosted, NoSQL database
Cloud Storage	scalable file storage, able to upload and download files directly to and from the Cloud Storage bucket
Cloud Functions	serverless event-driven backend. Can write and deploy code that automatically responds to events coming from other Firebase products. Firebase products emit events when data changes within the product and the code used in the Cloud Functions can be triggered in response to those events.
Firebase Hosting	A secure, global web-hosted content delivery network suitable for delivering static content such as HTML, CSS, JS, and images.
ML Kit	SDK for common machine learning tasks such as recognizing products in image captures such as text, faces, and landmarks.

Improve App Quality	
Name	Usage
Test Lab	Provides scalable and automated app testing on cloud-hosted iOS and Android devices.
Crashlytics	Provides clear, actionable insight into the app's crashes. It is also integrated with Analytics so you can measure how crashes are affecting the way users use the app.
Performance Monitoring	Provides insight into the app's performance issues from the user's point of view by measuring HTTP requests, startup time, and other code using its API.

Grow Your App	
Name	Usage
Analytics	Understand the "audiences" (a group of users who have taken some predefined action in the app, share some properties, or have common device characteristics) and how they use the app. Firebase analytics analyzes the stream of events these audiences take and target those audiences with other Firebase products in the Grow Your App category.
Predictions	Apply machine learning to analytics to predict user behavior such as which users are likely to not open the app, which will spend money, and who will use the app daily.
Cloud Messaging	Allows you to deliver push messages to indicate something of interest in the app to the users of the app. This can be done with code written on the backend to ping the app when something gets updated, or by composing the message in the Firebase console to pin the users with information of interest.
Remote Config	Allows you to customize the app without deploying a new version and to monitor the changes.
A/B Testing	Allows you to run marketing and usability experiments on the app to see what works best.
Dynamic Links	Enable native app conversions, user sharing, and marketing campaigns. When a client clicks on a dynamic link and the app that is needed is not installed, they are directed to the appropriate app marketplace to install it.

App Indexing	Helps re-engage users with Google Search integration.
In-App Messaging	Helps to show targeted, customizable messages to the users to engage with key features in the app. Based on the user's behavior measured by Analytics and Predictions, the message is delivered at the best possible time to increase client engagement.

For FoodAlert, a combination of all the toolkits above will be utilized, but in the beginning, the main focus will be placed on Authentication and the Cloud Firestore. Once the main framework of the application is created and has secure user login and data storage, focus will shift to using the ML kit to parse text from the receipts and Analytics and Predictions to enable the user to better manage their food wastage, coupled with the In-App messaging to make sure users are notified in a timely manner before their food expires. As the app is being developed, the Test Lab will be utilized to test functionality and make sure that the app is usable and reliable from the user's point of view.

Text Image Recognition and parsing AI[8]: Using Firebases ML-kit we can create local neural network models within our react code to handle text recognition through and image. First, we get a wrapper so that we can use their ML-kit library within the react code. From there their library allows us to build models locally on our app, such that the user can scan images with their camera and get instant text feedback for their purchases. By using the Firebases library, it prevents us from having to implement our own model, train it and pass it onto a mobile model or the cloud. This saves the burden of serving a neural net and needing a connection to get results for queries. React only comes in the form of an Android or iOS SDK and not an official web-SDK but there are open source wrappers and libraries that are compatible with react. ML-kit also allows for the use of custom TensorFlow-Lite models that lets users easily port them onto the mobile platform, meaning if normal text recognition doesn't cut it a custom model can be made.

3.2 Related Work

There are many applications out there that are related to food, and it is either a mobile application or a website. Such as FoodKeeper APP [9]. This app was developed by the USDA's Food Safety and Inspection Service, with Cornell University and the Food Marketing Institute. The main function of FoodKeeper is to provide the user with information on how to store the food that will have a maximum expiration date and maximize freshness and quality. This app is available in three platforms: iOS, Android and Web application.

No Waste [3] is an app that has similar functionality, developed by KH Creations IVS. The app includes food inventories, barcode scanning, receipt scanning, and recipe suggestions. This app does have many aspects that work well. However, during the testing of this app, we discovered that the receipt scanner has a bug where it will enter deadlock and scan indefinitely. When the app enters this state there is no way to break the deadlock and the app must be exited entirely. Once a receipt is scanned all expiration must be entered manually. Our app will calculate the expiration dates and auto-input the information. This app is also exclusively

available on Apple platforms and our app will be built to work across platforms. This app was originally developed in Sweden and there are some parts of the app where some of the writing is still in Swedish and some of the monetary notations are still in the Swedish Krona.

There is another mobile application called Fooducate [7]. This mobile app was developed by Fooducate, Ltd. It is available on iOS and Android. The main function of this application is when the user scans a food item, the system will assign a grade for a food item based on its nutritional value and with a breakdown of what's actually in the food. It will also give the user alternative options and tracks daily calories on the side.

However, the FoodKeeper tells the user how long will the food stay fresh, and the Fooducate tells the user nutritional values. None of them will keep track of the expiration date of the food that the user purchases. Our FoodAlert app will allow the user to take a picture of the food items and keep track of food expiration dates. It will also notify the user the day before a food item expires, the notification day would be calculated and scheduled. In the end, FoodAlert will provide a solution that would help the user keep track of their food and food expiration.

4.0 Design

4.1 Requirements, Use Cases, Design Goals

Functional Requirements

- Scan receipts and store parsed food information into firebase.
- Provide the user with the ability to manually enter an item and date into the calendar.
- Allow the user to choose a product and manually remove the item from the food item list.
- Calculate the expiration date of items from Firebase and output to the calendar and item list.
- Predict item purchases and suggest shopping lists.

Nonfunctional Requirements

- A user interface should be easy to navigate and conform to android and iOS norms.
- Startup should complete within 5 seconds.
- Receipt scanning and parsing should complete within 15 seconds
- The calendar should display the expiration dates of food.
- The items list should display both the purchase date and expiration date.
- The suggested shopping lists should have the ability to manually add and remove items.
- A notification should appear for an item on its day of expiration.

4.2 High-Level Architecture

FoodAlert will have an easy to use interface that the user will be able to intuitively navigate. The first page that will load is the Login/ Create Account screen (see UI Designs a). On this screen, the user will be able to select login, which will bring them to a separate page to type in their credentials that will be passed to Firebase for authentication or Create an Account, which will allow them to submit credentials to Firebase for a new user. Once authenticated, the user will be brought to the Calendar view (see UI Designs b) as the default home screen. The

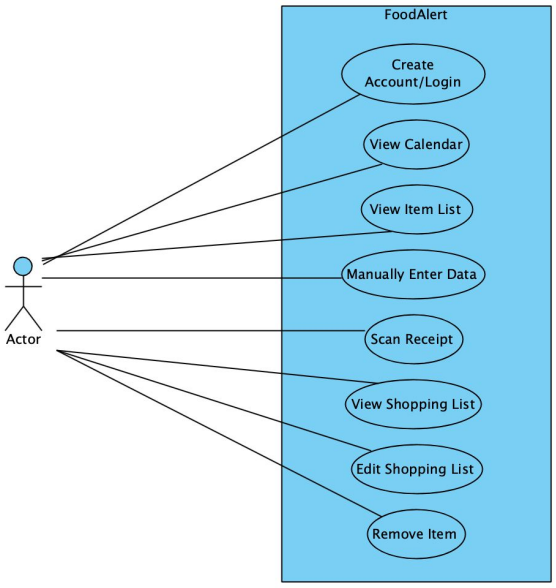
calendar view will allow the user to see an overview of their food expiration dates. On each date that food is set to expire, a bar will appear under the number. The user can then click on the number, and what food is set to expire will appear on the bottom half of the screen. This page is continuous scrolling, meaning that the user can just swipe up to go to the next month. In addition, there is the ability to choose the month by clicking on the drop down arrow next to the month name.

On the home screen, there will be a hovering plus button on the bottom left-hand corner. If clicked, the user will be brought to a new page where they can scan their receipts to populate the expiration dates of purchased food. This process will be done through using the phone's camera interface and processing through Firebase ML toolkit.

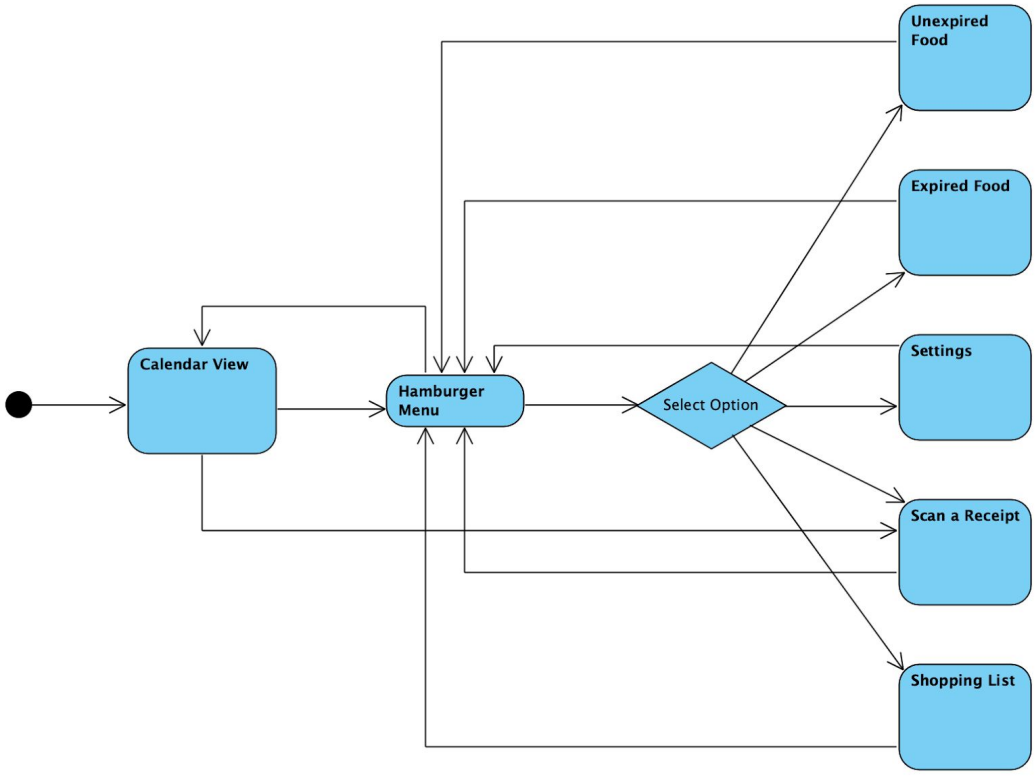
The hamburger menu on the top left corner gives the user several options once clicked (see UI design c). Within this menu, the user will be able to go back to the calendar view if they have navigated away, navigate to a page to view all their unexpired food, view their expired food, create and edit their shopping list, or manually add a food expiration entry. If the user chooses to go to the unexpired food page, they will be brought to a screen that shows all of the unexpired food they have, when it was purchased, how it is being stored, and when it is set to expire. The expired food page will show much of the same information as the unexpired food page, but will only show foods past their expiration date. It will also give the user an option to remove the food item from the app once they have disposed of the expired food and add the food item to their shopping list.

The shopping list will allow a user to keep track of what items they need to purchase at the store next time they go. It will be a manual entry page, but from the expired food page, the user will have an option to automatically add items based on what food items have expired. The users will also have the ability to manually add food expiration dates. This allows the user to set expiration dates for leftovers that are not able to be scanned through a receipt. A user can also modify the settings of their account to change the email, phone number, name, and the password associated with the app along with the default settings of the app such as notification frequencies.

The following diagrams and design sketches show the main features of the app as well as the app-use flow.

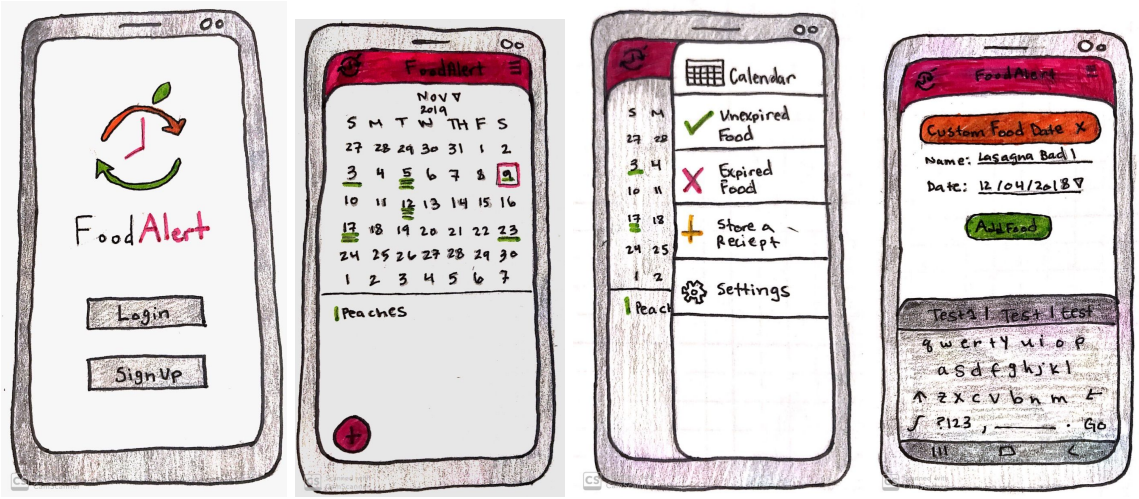


Use Case Diagram



Activity Diagram

UI Design



a b c d

UI Design

4.3 Risks

Risk	Risk Reduction
React has problems with libraries come with many dependencies.	Minimize the amount of unnecessary web-app dependencies that could have security faults in their open-source codes.
AI receipt scanner does not work	Collaborate with national supermarket chains for database access. Allow manual entry as a last resort for entering produce into the app.
Lost database additions	Firebase has an offline local database so users can store data locally and sync with our servers when connected again.

4.4 Tasks

NOTE: Unit testing and documentation will be performed on each step where applicable such that each part of a whole will be working. Also integration testing will be done when combining two different systems.

1. Design Flow-Chart for application design, sketches and
2. Begin creating React skeleton with necessary pages, make sure interpage-communication is handled. Basic GUI elements.
3. Create Firebase Database Tables along with specific query calls within our API to handle our specific populations. Implement online and offline functionality.
4. Implement database on React side and create page population.
5. Create User-Authentication so the app could be accessed cross-platform on users different devices.

6. Create Tensorflow-Lite model to handle receipt text recognition.
7. Integrate Firebase Machine Learning Kit, add in-camera functionality with neural network scanning.
8. functional testing of each part of the whole of the application
9. Polish the application front end, reinforce documentation

Stretch Goals:

1. User acceptance testing for application performance and aesthetic.
2. Smart fridge functionality/optimization.

4.5 Schedule

Tasks	Dates
1. Basic Planning(flow charts and more)	11/14-11/28
2. Basic React app skeleton implementation (menus, empty pages, hard food entries)	11/28-1/13
3. Create Database schema	12/9-1/20
4. Implement React calendar page	1/13-1/27
5. Implement Notifications	1/27-2/2
6. Implement DB API with queries	1/20-2/10
7. Integrate database with React Web-App(ie: local database population, queries)	2/10-2/24
8. Create user roles and authentication Firebase side	2/10-2/24
9. Integrate user-authentication logic app side	2/24-3/5
10. Create Tensorflow Lite model for text recognition	1/20-2/10
11. Implement Ml-kit with Tensorflow-Lite model load	2/10-2/24
12. Functional Test	3/5-3/19
13. Format UI, create platform-based scaling other than android	3/19-3/25
14. Acceptance Testing	STRETCH
15. Fridge Testing	STRETCH

4.6 Deliverables

- Markdown Document: explaining each part of our code and why we used certain coding methods versus others.
- Design Documentation: A website entailing all our Structures and dependencies and getting down into deep documentation over our code exactly. We would use auto documentation to turn our comments into the documentation.
- Design Flow-Charts, Diagrams and UI drawings
- Database scheme and initial data
- React Native code
- Code and model code for the AI model
- Final Report

5.0 Key Personnel

Alycia Carey - Carey is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. She has completed relevant courses such as Software Engineering, Database Management Systems, and Computer Networks. She has worked for the University of Arkansas Network Engineering Team as a Network Security Technician as well as participated in several cryptography research projects. She will be responsible for front-end development as well as authentication.

Jasper Harrison - Harrison is a senior Computer Engineering major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed relevant courses like Data Mining, Artificial Intelligence, Software Engineering, Mobile Programming, and Database Management Systems. He has worked for a social media startup, a healthcare company, and for one of the big banks as a developer. He will be responsible for unit testing and integration testing as well as connecting the different tiers of the app together.

Bentley Lager - Lager is a senior Computer Science and Mathematics major. She has completed relevant courses including Database Management, Information Security, Software Engineering, Computer Networks, and Multimedia Compression and Delivery. She has worked for a social media start up, a healthcare software company, and a management and technology consulting firm. She will be responsible for the database development.

Jiamin Lin – Lin is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. She has completed relevant courses like Software Engineering, Wearable & Ubiquitous, and Database Management. She has worked as a software developer and IT support for private and public companies. She will be responsible for database development as well as contributing to back-end development.

Gabriel Priest - Priest is a senior Computer Engineering major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed relevant courses like Artificial Intelligence, Mobile Programming, and Data Mining. He has worked as a

Java developer at Cerner. He will be responsible for the camera-based functionality of the application.

Nicholas Waterworth – Waterworth is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed relevant courses like Artificial Intelligence, Software Engineering, and Algorithms. He has worked for a social media app startup and a Department of Defense contractor. He will create the AI network for text character identification in application to shopping receipts and contribute to UI design.

6.0 Facilities and Equipment

Android Phone: For testing and development of the FoodAlert app. Multiple members of the team have android phones with different operating system versions that can be used for testing purposes.

iPhones: For testing and development of the FoodAlert app. Multiple members of the team have iPhones with different operating system versions that can be used for testing purposes.

Smart Fridge: For testing and development of the FoodAlert app for smart fridge integration. Dr. Frank Liu at the University of Arkansas has a smart fridge in his lab that can be used for testing.

7.0 References

[1] Center for Food Safety and Applied Nutrition. “Food Waste and Loss.” *U.S. Food and Drug Administration*, FDA, www.fda.gov/food/consumers/food-waste-and-loss.

[2] “Firebase Products | Firebase.” *Google*, Google, firebase.google.com/products.

[3] “Food Waste FAQs.” *USDA*, www.usda.gov/foodwaste/faqs.

[4] “Home.” *Nowasteapp*, https://www.nowasteapp.com/.

[5] “Key Facts on Food Loss and Waste You Should Know!” *Food and Agriculture Organization of the United Nations*, www.fao.org/save-food/resources/keyfindings/en/?fbclid=IwAR1ZgQw9MCzbHgqRdt6OQSJvpAaCGBnsSWX3ruoArvcGoYf7bRpawiVjtV0.

[6] “Learn the Basics · React Native.” *React Native Blog ATOM*, facebook.github.io/react-native/docs/tutorial.

[7] “Lose Weight & Improve Your Health with a Real Food Diet.” *Fooducate*, www.fooducate.com/.

[8] “ML Kit for Firebase | Firebase.” *Google*, Google, firebase.google.com/docs/ml-kit.

[9] Public Affairs. “FoodKeeper App.” *FoodSafety.gov*, 12 June 2019, www.foodsafety.gov/keep-food-safe/foodkeeper-app.

[10] Stevenson, Doug. “What Is Firebase? The Complete Story, Abridged.” *Medium*, Firebase Developers, 25 Oct. 2018, medium.com/firebase-developers/what-is-firebase-the-complete-story-abridged-bcc730c5f2c0.