**Database,Backend and CI/CD choices are shared by MobileApp and WebApp,
**Current implementation of Mobile app is not connected to a backend.

Report for the Checkout Mobile App

Frontend Technology: React Native , Java , Flutter

1) Ease of development:

With some experience programming in Java and Javascript, React Native and Java for Android are my first two choices. Flutter is written in Dart so it would take me more time to get used to the platform. A pros for both React native and Flutter is that we only need to write one code that can run on both IOS and Android devices since they are both for cross-platform developments. On the other hand, writing an Android app using Java is more challenging since we will first need to understand the activity lifecycle, callback, and how to handle events. Therefore, we would argue that it is easier and faster to write a mobile app on React Native.

2) Maturity of the language/technology and the libraries available for integrations/UI:

Android SDK released in 2009, React-native released in 2015 and Flutter in 2018. Native Android apps come with bigger libraries and it can get direct access to the device without the need of a third party or extra modules seen in hybrid app development. UI libraries are rich for all 3 development platforms.

3) The domains covered by the technology/language:

React-native is also very similar to React for web dev, so one can maintain and scale both platforms relatively easily. Both Java and Javascript can be used for both front-end and back-end development. Dart can be used for web dev, mobile dev or compiled to JavaScript.

4) Popularity of the technology:

According to Stackoverflow 2019 Survey, in the most popular programming language is JavaScript (with 67.8%), Java comes shortly after with (41.1%), Dart (1.9%). In the most popular frameworks voting, React-native has 10.5% while Flutter has 3.4%. So React-native overall is more popular among developer

5) Performance, scale and speed of the solutions built with these technologies.

Native apps have higher performance and scalability since they are optimized for a particular platform. However, with good architecture, users won't be able to notice the difference in speed and performance among the 3 frameworks.

6) Any other criteria you think is important and needs to be considered

Since we also choose React for our web app, react native would be a good choice to maintain code, API for both platforms at the same time. Combine it with redux - a library for managing application state, the app can meet the requirements with a clean code architecture.

Report for the Checkout Web App

Frontend: React.js vs Vue.js vs Angular

1. Ease of development for you

   In case we have experience working with React.js and Node.js framework, React is the first choice as he has enough knowledge to work with it. Vue and Angular are using much different syntax, libraries and frameworks, which is time consuming to learn and implement. Also React.js is also the core language of React Native used for mobile app development. It would be easier to maintain Apps with same/similar framework than different frameworks. Also Vue and React do not require developers to be very good at typescript, which have a quick learning curve.

` 2. Maturity of the language/technology and the libraries available for integrations/UI

   Angular was released in 2010 by Google, and is used for Google Adwords applications. This adds to the authenticity of Angular which provides it strong community support since its launch.

   React was released in 2013 by Facebook. A team of engineers at Facebook are still working on its improvement. Facebook has coded its several products to React and invested in it heavily. Highly reliable.

   Vue is released in 2014 by an ex-engineer of Google, Evan You, and it is not backed by any top-level corporate, however this framework is very popular

   Angular and React has a strong community base with backing from top companies like Facebook and Google. Vue is popular in the open-source community but has no support from large companies. I think Angular and React are more stable and reliable based on they're backed up by google/facebook. Also React, Angular are both released earlier than Vue which has more open source libraries/shared repos available compared to Vue.

3. The domains covered by the technology/language

   All three frameworks are somewhat javascript based, and all three of them can be used for both mobile app frontend, and web app frontend. React has React.js for web app and React Native for mobile app; Angular itself adapts both; and Vue has Vue.js for web app and Vue Native for mobile app. The three cannot really distinguish each other by their language coverage.

4. Popularity of the technology

   React is ranked first in Most Loved, and Wanted Web Frameworks in Stackoverflow 2019 Survey, Followed by Vue.js. Also according to https://hackernoon.com/, React is the most popular in search and download trends.

5. Performance, scale and speed of the solutions built with these technologies.

   Angular uses real DOM, It is time-consuming, and risky to attract a lot of bugs.Also it results in slow performance

   React uses virtual DOM, It is not browser specific and is lightweight.the issues of slow performance of real DOM is eliminated

   Vue takes all the good attributes of frameworks launched before it, uses virtual DOM and ensures faster and bug-free performance.

In case of performance speed and efficiency, Vue is probably the best framework to work with, followed by React, and Angular.

Also worth paying attention is that our web app is a really small, toy app. Vue and react are more suited for light-weight applications, offer better performance and flexibility compared to angular, and angular is the best for large UI applications as it offers everything from routing, templates to testing utilities in its package.

Summary: In summary, consider React.js's easiness of development, great support by facebook and community, its high efficiency, low learning curve and our small app size. React would be the best choice.

Report for sections shared by Webapp and Mobileapp

Backend: Node.js vs PHP vs Python

1. Ease of development for you

   In case we have experience working with React.js and Node.js framework, Node.js is the first choice, it is javascript based, which keeps the frontend and backend in the same language. Neither of us know PHP well so it's the last choice. Python seems to be a good choice, but also requires a lot of time to read documents and learn the libraries.The advantage of node.js which is using same language for both frontend and backend make it easier to learn and adapt.

` 2. Maturity of the language/technology and the libraries available for integrations/UI

   PHP was developed in 1994 by Rasmus Lerdorf. Over the last 20 years its usage and support are fully extended and still one of the most popular language with great support.

   Node.js was first released by Ryan Dahl in 2009, as an open source, cross platform powerful run-time environment that's built on the V8 engine. It's library is well supported and extended by the community.

   Python was developed in 1991, developers have access to high functionalities and extensive library support. Python also offers several advanced web API's for multiple backends.

3. The domains covered by the technology/language

   Python is commonly used for developing desktop GUI applications, websites and web applications, and Machine learning.

   Node.js is used for traditional web sites and back-end API services.

   PHP is used to develop Static websites or Dynamic websites or Web applications.

   In case of domains, Python has the greatest coverage and PHP has the worst cover.

4. Popularity of the technology

   Node.js uses javascript which is ranked first in Programming, Scripting, and Markup Languages in Stackoverflow 2019 Survey, python is the 4th and php is the 8th, also Node.js is ranked 1st in Other Frameworks, Libraries, and Tools.

   However, Python is ranked 2nd in Most Loved, Dreaded, and Wanted Languages over javascript, over PHP.

5. Performance, scale and speed of the solutions built with these technologies.

   Node.js is highly efficient because of its single threaded event call back mechanism which provides the opportunity to connect scripting languages with the brute force of network programming. Its speed is really fast due to the use of the V8 engine to compile javascript into machine code.

   Python is an interpreted language and is slower than compiled languages. And python is single flow which processes requests slower. It is in general less efficient and slower compare to Node.js

   PHP is also slower compared to Node.js due to the strong power of the V8 engine.

Summary: In summary, considering Node.js's easiness of development, great support from community, high speed/efficiency, and enough domain coverage Node.js is the best technology for our checkout calculator app

Database: MongoDB(NoSQL), Firebase(NoSQL), MySQL

1) Ease of development :

   In case of development, We have experience with MySQL, MongoDB, PostgreSQL, and for the purpose of developing a web/mobile app, to work with a NoSQL database which stores data in JSON objects form is easier.

   Considering our experience, and the flexibility of schema in MongoDB. MongoDB is the best for ease of development.

2) Maturity of the language/technology and the libraries available for integrations/UI:

   First release: MySQL in 1995, MongoDB in 2009, FireBase in 2009. All three databases have a big library and a big community. Cannot distinguish them based on supports and libraries. And those 3 are all well supported by Node.js

3) The domains covered by the technology/language:

   Firebase is a real-time document stored that supports Java, Javascript and Objective C. Firebase is a backend as a service which can be difficult to configure but it's easy to build with web and mobile apps.

   MongoDB is a NoSQL database that supports more programming languages than the other 2.

   MySQL is a SQL database. Both MongoDB and MySQL can be hosted on cloud service.

4) Popularity of the language/technology:

   According to the Stackoverflow 2019 report: MySQL has 54% votes, MongoDB has 25.5 % votes, Firebase has 12.8% votes. MySQL seems to be the most popular one.

5) Performance, scale and speed of the solutions built with these technologies.

   MySQL is high efficiency when working with a larger data set than other noSQL databases.

   Mongodb is faster than MySQL when handling large unstructured data. Developers also note that MySQL is quite slower in comparison to MongoDB when it comes to dealing with large databases. According to https://dzone.com, Mongodb is in general much faster at updating and inserting, but slower at selection. Which means if the data we need to implement anticipates a lot of change, a NoSQL database would be better than SQL.

Summary: For our current app requirement, there is no need to use a database. The final choice really depends on the data structure that will be stored into the database. If it's some extremely complex data structure with many relations between data entries, and requires a lot of real time updates/inserts to the database, NoSQL database is much faster and therefore is the best choice. Otherwise, for simple but large data structures, SQL databases, such as MYSQL would be the best choice. All three can work with our choice of backend, and have their own advantages. Due to our personal experience and skills MongoDB is more preferred than Firebase for a NoSQL database. It's likely the decision will be made between MySql and MongoDB depending on the requirement.

CI/CD:

Github Actions, Jenkins, Travis CI

1) Ease of development :
   Since we are using github for our code repo, it would be easier to integrate a workflow using github actions. Jenkins is also a good choice as it is Java based.

2) Maturity of the technology and the libraries available for integrations:
   Initial release: Jenkins in 2011, github actions in 2018
   Since Jenkins and Travis CI are widely used by big companies (ex: Facebook, Netflix), they provide a rich library with hundreds of plugins and organized documentation. Github Actions is still fairly new so there is not much documentation to support the tool.

3) The domains covered by the technology:
   Jenkins has control over GUI (web based), REST API. Jenkins can be integrated with 118 tools, Travis can do so with 67 tools, N/A for github Actions

4) Popularity:
   According to https://www.cuelogic.com/blog/best-continuous-integration-ci-tools, almost half of commit on github used Travis Ci, Jenkins is at 3rd most used CI/CD, N/A for Github Actions

5) Performance, scale and speed of the solutions built with these technologies:
   Speed: Jenkins needs elaborate setup so the set up time is longer than Travis CI
   Scale: Jenkins give users unlimited customize options while Travis has unlimited open source project
6) Any other criteria you think is important and needs to be considered
   Since we are looking for an easy set-up CI that supports testing and deploying React apps, we chose github Actions even though it is not as popular as the Jenkins and Travis CI. If the app plans to scale in the future, we then will consider Travis CI or Jenkins as a better option for CI/CD.