Charlie McClatchy
mcclatc4
CSC301 A1 mobile app report

In this assignment our goal was to create a checkout price calculator with some basic functionality. This calculator would correct for tax and any possible deals as well as allow for the users to add items themselves for a fully customizable checkout experience. For this section our task was to develop a mobile app version of the assignment. While this was similar in effect to the web app version, producing this checkout calculator for mobile results in its own particular series of challenges and choices to make.

This project includes technologies in the following tech stack. Examples of various solutions as well as our assignment selection are given for each. Finally in each category a considerations section is added to explain the reasoning for picking our particular technologies.

Frontend:

Possibilities: Angular, React native, Vue

Chosen: **React Native**

Considerations:

Many considerations went into choosing which frontend framework and language to go with. When creating a mobile app you have to start with the choice of what operating system you would like to run on. We decided it would be best for our app to be able to run on both ios and android, as this would be the most convenient for any customers looking to use our product. This is a positive consideration for all of our possibilities because they work on both platforms, but react native in particular is especially renowned for cross platform support, giving it the edge. With this is mind we also considered the domains covered by these languages. All three of these options have mobile app frontend development in their domains, with react being the most specific and narrowed in focus. As a last consideration we thought about the ease of development for me and my partner. For the two of us we have more experience in react than the other two frameworks. These considerations lead us to conclude that react native and javascript would be the ideal framework and language for our frontend. (popularity)

Backend:

Possibilities:Parse, Express, Django

Chosen: **Parse**

Considerations:

Choosing the language and framework for the backend is a very important task which comes hand in hand with choosing the frontend. As you will see many of these decisions come from the knowledge that we will be using React Native and javascript as our actual frontend technology. In this category there are conflicting considerations for using the different options and it will be shown that there is no true right answer. For the first consideration we can say that Django will be the easiest for me to use by far, as I have lots of Python experience but little javascript experience. This is a big advantage because time spent learning a language is not time spent working on the app and we have a limited timeframe. On the other hand we have the domain of the language and the libraries available. While Python can be used for backend app development, that is not in its primary domain whereas javascript is the go to language for this kind of thing. Django is primarily a framework for web apps, so using it to develop our mobile app would be out of its normal domain. Similarly, Express is made as a web application framework and is not designed for mobile, whereas Parse integrates seamlessly as a mobile backend framework. Moreover, having the whole frontend and backend all in javascript makes things much cleaner and easier for development. With this all in mind, we decided to stick with Parse as a reliable framework suited to the domain of our problem.

Testing infrastructure:

Possibilities: Jest, Mocha, Jasmine

Chosen: **Jest**

Considerations:

A good testing framework is vital in the production of good code, as time spent on inefficient bug testing and error reports might as well be time wasted if you don't have proper unit tests. Jest, Mocha, and Jasmine are all very common javascript testing frameworks and are all quite useful. We chose only javascript testing frameworks because both our frontend and backend are chosen in javascript at this point. Our first consideration for this section is popularity of the software. This is because with something straightforward like this it makes sense that the common and frequently used frameworks will be quite useful for general purposes. Jest is the default unit testing framework for React Native, so it was chosen by us as the popular choice. Using the default framework makes things easy, standard, and ensures minimal troubles in setting up effective testing procedures. This puts Jest as the best choice as far as the ease of development consideration is concerned as well, making it the best choice for our project.

Database:

Possibilities: SQLite, MongoDB, MySQL

Chosen: **MongoDB**

Considerations:

For our last group of considerations we chose our database once we had already made all of the prior choices. This left us with less room for the choice and for the proper database option to functionally fall into place. Because we chose Parse as our backend we were required to go with MongoDB, as is stated by the Parse website. However, there are still possible considerations to be made for the database in our assignment. While these are all well respected and highly effective database options, sqlite is most considered a choice for mobile development and local data storage. However, MongoDB has another plus side in that the scalability and ease of use that comes from storing objects as json-like data is very useful compared to the standard practice of using sql tables. This makes it easier to do whatever we want to our Item objects and have no worries of storing it in the flexible schema of the MongoDB database. This is why we went with MongoDB as our database technology option.