Choice for Android backend development

1. Java
2. Kotlin
3. React Native

Ease of development
I have learned Java a year ago and keep practicing with it, especially for the Android app development. However, I am not familiar with Kotlin and React Native and it going to take me a considerable amount of time to develop an Android with one of them.

Maturity and UI library
For Kotlin, EverNote, Uber, Coursera and Square are developed by it. Then for Java, Twitter, Cash App and Signal are examples. For React Native, it is used for the Facebook, Walmart and Instagram mobile app development. Meanwhile, Java and Kotlin are the two official language for the Android app development and Android provides a variety of pre-built UI components and modules for these two language. Though React Native has an amount of incredible UI library, in my opinion, they are not as convenient as others'.

Time cost for Development
React Native can build app faster as it saves time for recompiling the code, which can not be done in Java and Kotlin. Also, the mobile app programmed in React Native can perform in both IOS and Android platform, however, mobile app in Java or Kotlin can only be launched in Android mobile.

Complexity of the App
As the native language, Java and Kotlin allow developer to build app that requires in-depth internal API's call and develop unique and complex apps.They can also access and use the build-in hardware and software such as location, storage and GPS. While, developers using Reactive Native do not have such direct access on the device and limit the complexity of the mobile app.

Reaction to the system update
In my opinion, the biggest drawback of React Native is its uncertainty on the Android update as it is hard to guarantee the mobile app will work on the next Android version. If React Native does not perform properly after the update, the developers have to pause and wait until Facebook fix such problem. As the native language, Java and Kotlin do not have such disadvantage.

Community support and online recourses
As the oldest programming language among these three, Java has the biggest community and most developers. Such resource enables Java developer to get help from others quicker and find online resources easier. Kotlin and React Native have a relatively smaller communities thus the help may be limited.

Overall, I choose Java for the above reasons. Though it has certain drawbacks, it is the most suitable language among these three for me.

Choices for Android frontend development

1. XML
2. Java
3. React Native

Ease of development

I learned XML a year ago and am familiar with how to make graphical interface with them. However, react native is quite new to me and is going to take time to learn.

Efficiency for creating graphical interface

Android provides a straightforward XML vocabulary and allows developers to use layout, then developers such as me can modify the interface while looking at the visual interface editor directly instead of code. Thus, developers can create interface efficiently and conveniently as they do not need to check with the tedious code. However, Java and React Native do not have this feature.

Responsive interface

FlexBox provided by React Native gives all the essential instruments for creating a responsive interface for the developer. Similarly, the Android's constraint layout enables XML to do the same. Android also proves an amount of libraries and components for its official language — Java — to accomplish the responsive interface.

Complexity of the graphical interface

React Native does not provide enough amount of interface power for the highly complex UI or sophisticated animation. Similarly, XML does not have such capacity as well. Then Java will be the only available choice for the complex graphical interface and animation.

Bugs in the technology

Really Native is designed by Facebook and it does not fix bugs regularly unless they are relevant to the Facebook app. Then if an irrelevant bug occurs, developers will either fix by themselves or wait for the Facebook's response. However, Java and XML are the supposed language by Android and they are updated regularly for the bugs.

In conclusion, I choose XML as the frontend language. Since we not are implementing complicated project in this assignment, then XML is the best choice.

CI/CD Choices
1. Jenkins
2. TravisCI
3. Github Action

Support
Jenkins has no official support available or SLAs, however Github Action and TravisCI have community support or dedicated online interface.

Parallelism
Jenkins allows build to be run in parallel but all builds share the same environments and can be issues from shared resources. Github action allows concurrent jobs even multi-platform and TravisCI allows developers to split build into different stages which are then run in parallel.

Container support
Jenkins runs all builds in the same environment by default as build server itself and leads to issues. GitHub Actions allows Linux, macOS, Windows and container run differently in a VM. TravisCI runs each build in isolated virtual machine.

Build pipelines
Jenkins offers extensive support for custom pipelines, either through the Jenkins Pipeline DSL or web UI. Github Action uses GitHub Action Workflows to Let developers to build pipelines. However, CircleCI specifically built around GitHub pull requests.

Ecosystem
There is a large community use Jenkins and Github Actions but not CircleCI.

Integration
They all support for the common tools like Slack notifications. Various VCS platforms etc.

I choose Github Action, since we are using GitHub as the repository and I am a Mac user. Also, comparing to others, it the most solid one for the above reasons.

DataBase Choices
1. SQLite
2. Txt file in internal Storage
3. MongoDB

Dataformat
MongoDB uses JSON documents, SQLite uses Tables and Txt file in internal storage follows text style.

Ease of Development
Since Android studio supports SQLite, SQLite is as easy as storing Txt file in the internal storage, While. MongoDB takes more effort to implement.

Speed
All three are quite fast, but the SQLIte is the fastest one in my opinion.

Open Source
SQLite and MongoDB have open sources online, while storing using internal storage requires developers to search by themselves.

Server
MongoDB has MongoDB server, and SQLite and internal storage do not have any servers.

Since we are not going to connect to the server in this assignment, then SQLite is the most efficient one since it has Android's support and performs ideally in Android studio.

Website Development

Technology decided to use:
Frontend React
Backend Node.js

Front End

Technology put into consideration:

React
Angular
Vue

My Solution: React

Comparison:

Ease of use:

React: Generally easy to learn as there is huge community support considered more newcomer friendly

Angular: Have a high learning curve. Need to master typescript and MVC

Vue: Considered easy to learn but poor coding might lead to a lot of bugs.

Ranking Stackoverflow 2019 Survey:

React: 31%
Angular: 30%
Vue: 15%

Maturity of the technology and the libraries available for integrations/UI

React: Has a huge library and available UI API that can be used as components. The technology is backed up by big community members which contribute to creating large usable libraries.

Angular: Has the most mature one out of all of them and can customly download libraries.

Vue: Has good integration as integration with other technologies like bootstrap

The domains covered by the technology:

React, Angular, and Vue are frontend technologies that connect other backend technologies to create databases. All three frameworks provide CLI tech which helps to build projects.

Performance:

React: Generally performs the fastest coming next is Vue

Angular: The slowest. This is the result of Angular being more frequent with updates and dynamic component design (so a lot of components work simultaneously)

Vue: Second fastest but slightly better in memory allocation and startup than react

Conclusion: REACT

Since we are building a basic app, we will choose a framework that is backed by the large community such as React as React is most popular, easy to find tutorials and fastest of all those three apps, it is a perfect framework to work on for this app.

Backend

Technology put into consideration:

Node.js
Python(Flask)
Django

My Solution: Node.js

Ease of use:

Flask: I have personally worked on the framework. Also written in Python which is familiar with us and as python is simple in terms of

syntax easy to deal with. Also flask is developed to help fast development for smaller app functions.

Node.js: Backed by large community and tutorials. Easy to connect with different tech using JS as it is the center of several different frontend frameworks.

Django: Has a built- in bootstrap tool which hugely helps the user to connect different functions. However, the framework is known for its complexity to master. Django also uses python as a language. Compared to flask created for fast development for larger app.

Ranking Stackoverflow 2019 Survey:

Django:	13%
Flask:	12%
Node,js:	49.9%

(Note Node.js in different category compare to Django and Flask)

The domains covered by the technology:

Python: Primarily Backend
Node.js: Primarily Backend but can use JS tech for frontend
Django: Primarily Backend

Libraries available for integrations/UI
(Exceeding to much word need to be point form)

Flask: Minimalistic libraries but user can selectively download wanted libraries

Node.js: Large downloadable libraries. Large community creates very mature and vast integration libraries

Django: Has very very large built-in libraries (4000+)

Performance:

All three frameworks are known for their speed, but python is slower compared to Node.js (speed python < java < c) Node.js uses a

V-8 engine while python is single flow so python is slower than Node.js which makes both technologies, Flask and Django, slower than Node.js.

Conclusion: Node.js

Since we are using React for the frontend, Node.js, which has huge libraries to connect to Node.JS as it is JS language, is more suited for this assignment. Node.js has a lot of tutorials to learn and connect with React. Also Node.js is very fast.

Database

Technology put into consideration:

MySQL
PostgreSQL
MS SQL Server

Solution: My SQL

Ease of use:

MySQL: We used MySQL before, it should be familiar to construct a Query. Also MySQL community-driven DBMS system.

PostgreSQL: It is fast solving complex queries but generally has steep learning curve

MS SQL: Has very good GUI which guides users using the database more easily but not as simple as a result.

Ranking Stackoverflow 2019 Survey:

MySQL:	54%
PostgreSQL:	34%
MS SQL:	32.8%

(Note Node.js in different category compare to Django and Flask)

As the most common database MySQL has the highest percentage out of all databases .

The domains covered by the technology:

These are all databases.

Used System and connected Library:

MySQL: MS Linux and integrates with a lot of framework

PostgreSQL: works on both system

MS SQL: Specialized in MS limited integration (you can use this on linux by downloading app)

Performance:

MySQL: The database performs the best in smaller projects and web-based projects that need a database for straightforward data transactions.

PostgreSQL: Database is fast when used in large systems where it requires to read and write complex queries.

MS SQL: Similar performance with MySQL but lacks in

Conclusion: MySQL

Since we are building small apps we do not need complex query calculation also we are already familiar with MySQL which will increase our speed with developing the app.

CI CD

Technology put into consideration:

Jenkins

CircleCI

Gitlab

Solution: CircleCI

Ease of use:

Jenkins: Has easy installation and simple interface friendly to users

CircleCI: Easy testing and debugging. Also known for fast setup

GitLab: Known for easy setup also has built in CI

Ranking:

No ranking is provided online but

Jenkins: Free to use CI which has a huge community. It is widely used in tech companies and people. In other word, it is more used more often than other two CI

CircleCI: Fairly new app developed in 2011 but gaining huge popularity.

GitLab: Famous app widely used in Git Environment

Testing Environment:

Jenkins: Can use plugin for testing environment based on JUnit testing.

CircleCI: Has a fast testing environment which can be used for setting up testing environments.

GitLab: Can setup thorough extension

Pricing:

Jenkins: FREE

CircleCI: Basic app is free but other feature is locked with paywall

GitLab: FREE

Performance

Jenkins: Fast UI

CircleCI: Fairly fast UI

GitLab: Slow UI compared to other two tech.

Conclusion: CircleCI

Since we are using github as a deployment method CircleCI which has easy connection to github and easy testing environment, is our best option as CI CD platform.