| **Tech Stack** | **Description** | **Fit to Project** | **Experience in Team** | **Maintenance** | **Risks** | **Resources** |
| --- | --- | --- | --- | --- | --- | --- |
| iOS:  Swift  Xcode  iOS SDK  SwiftUI  Android:  Kotlin  Android Studio  Android SDK  Jetpack Compose (UI)  AI:  Python  Amazon Recognition Image?  Design Tools:  Figma  Cloud AWS:  Database - RDS Aurora (MySQL/PostgreSQL) Images - S3  EC2 | This was our initial stack, which included software intended to create dedicated apps for each specified platform, as well as initial ideas for database management and design tools. | This approach would have been suited for focusing on android and iOS development. As a result, incorporating a website into the scope of the project would have been more ambitious. | Almost all of our team members have little experience with any of the mentioned software, besides Python, SQL and Android Studio | This approach might have been more intense to maintain, due to the different approaches necessary to implement an Android and iOS app. | It might have been harder for all of our team members to learn new programming languages and techniques. Moreover, managing bugs and issues across the two platforms would have been more difficult | Postman (API requests) <https://www.postman.com/downloads/>  For IOS apps developing  (Xcode, SwiftUI, and UIKit)  <https://developer.apple.com/tutorials/app-dev-training/>  Android apps  <https://www.youtube.com/playlist?list=PLEiEAq2VkUULG4j7aDpamb6b5Ym3RFQOj> |
| React PWA and Next.js  Cloud AWS:  Database SQL  AI:  Python  Amazon Recognition Image?  Design Tools:  Figma  WebSockets with Socket.IO or Kafka for any real-time features | Upon meeting with our client, we were given some feedback based on their former projects. This included utilising a web app approach, as opposed to dedicated applications. Moreover, tools for real time features were also suggested | As our client explained, the scope of the project does include a potential website with the same features as the app. As a result, this stack reflects this with a more universal approach towards this. | Some of our team members have some experience with JavaScript and SQL | Potentially, this approach is more maintenance friendly as it utilises a more universal approach via the web app implementation. | A risk of the web app approach could be that dealing with errors is now more important, as they affect the functioning of the system across all platforms. Moreover, any given roadblock could stall the progress of the project more, due to a the unified approach. |  |
| Languages**:** JavaScript, HTML, CSS  React PWA and Next.js  Cloud AWS:  Database -  PostgreSQL (with Elasticsearch)  AI:  Python  Amazon Rekognition Image?  Design Tools:  Figma  WebSockets with Socket.IO or Kafka for any real-time features  user authentication: OAuth with JWT (JSON Web Tokens) and HTTPS everywhere | This was our last iteration on the tech stack. We included additional database management suggestions from our client, as well as user authentication, to create a full tech stack for the upcoming project. | We believe this iteration of the tech stack is most fit for our project. It continues the web app approach, as well as additional security considerations, as the system will be handling a lot of private customer data. | All of our team members have some experience with HTML and Python. Some have limited knowledge of SQL and JavaScript. | Again, the advantage of this approach is the maintainability, which was one of the reason our client suggested it. |  |  |