Technical Documentation

## **System Architecture**

The site was largely built around the Angular and Ionic frameworks. In hiring a full-time engineer, we suggest you look for someone with a background building web and mobile apps using them.

### **Technology Stack**

* Key Programming Languages
  + Javascript/Typescript
* Front End
  + Angular
  + Angular Material
  + Bootstrap
  + Ionic
    - Capacitor/Cordova Plugins
  + Luxon
* Back End
  + NestJS
  + Mongoose
  + Passport
  + Handlebars
* Third Party Services
  + Postmark (Email Delivery)
  + Twilio (SMS Delivery)
  + Google Places API (Place Lookup)
  + Google Tag Manager & Google Analytics
  + Google Fonts

### **Servers**

* Amazon AWS EC2 instance
  + PM2 running versions of staging and live app on that server
* MongoDB Atlas (Cloud Database)
  + Databases
    - cohear-staging DB
    - cohear-live DB
  + Collections:
    - Cities
    - Focus Groups
    - Users
    - Clients
    - Issues
    - Neighborhoods
    - Communities

# **DevOps Accounts**

* **AWS Hosting**
  + Current Status: Cohear managed
* **MongoDB Atlas**
  + Current Status: Cohear managed
* **Postmark**
  + Current Status: Cohear managed
  + Next Steps: Cohear need to add a payment method and plan
* **Google Analytics**
  + Current Status: Cohear added as Admin
* **Google Places API**
  + Current Status: Cohear API Key Created
* **Twilio**
  + Current Status: Gitwit created an account for Cohear
  + Next Steps: Gitwit to transfer account and set up new 2FA when final invoice is paid
* **Github**
  + Current Status: Gitwit controls repo
  + Next Steps: Gitwit to transfer repo ownership when final invoice is paid

# **Push to Staging Steps**

* In local environment:
  + On master branch:
    - Ensure environment.prod.ts httpUrl is set to ‘<https://staging.app.wecohear.com>’
    - Inc version numbers in ionic/package.json and server/package.json
    - Push all changes
  + git checkout dist
  + git pull
  + git merge master
  + cd ionic
  + ng build –configuration production
  + git add -f www/
  + cd ../server
  + nest build
  + git add -f dist
  + git commit -m “Dist build of v\_.\_.\_ for staging”
  + git push origin dist
* In AWS Server (login as ‘ubuntu’ user via ssh):
  + cd /apps/cohear-staging
  + git pull origin dist
  + pm2 restart cohear-staging

# **Push to Live Steps**

* In local environment:
  + On master branch:
    - Ensure environment.prod.ts httpUrl is set to ‘<https://app.wecohear.com>’
    - Push all changes
  + git checkout dist
  + git pull
  + git merge master
  + cd ionic
  + ng build –configuration production
  + git add -f www/
  + cd ../server
  + nest build
  + git add -f dist
  + git commit -m “Dist build of v\_.\_.\_ for live”
  + git push origin dist
* In AWS Server (login as ‘ubuntu’ user via ssh):
  + cd /apps/cohear-live
  + git pull origin dist
  + pm2 restart cohear-live

# **Build and push new iOS App to App Store Connect for testing**

* In local environment:
  + Ensure xCode is installed
  + Ensure developer has a signing certificate setup that’s tied to Cohear Developer account
  + On master branch:
    - Ensure environment.prod.ts httpUrl is set to ‘<https://app.wecohear.com>’
    - Inc versionCode and versionName in cohear/ionic/ios/App/App/Info.plist and cohear/ionic/android/app/build.gradle
    - Push all changes to master
    - cd ionic
    - npm run build-ios
      * When finished, this should open xCode
      * Edit ionic/ios/App/App/config.xml variable from “$ANALYTICS\_COLLECTION\_ENABLED” to “true”
      * Make sure App Scheme exists and ‘Any iOS Device (arm64)’ is selected
      * Click Product > Build
        + Wait to finish
      * Click Product > Archive
        + Once finished, a pop up will come up
        + Make sure you’ve selected the newest archive that was just created
        + Click Validate App

No need to change anything here, just click Next until the Validate button shows up

Click Validate

* + - * + Assuming validation succeeds:

Click Distribute App

Choose App Store Connect as the destination

Choose Upload as the option

Click Next until the Upload button shows up

Click Upload

* + - * + Once finished, you should get a success message that the archive was uploaded to App Store Connect
    - You can then manage the rest of the process in the App Store Connect website

# **Build and push new Android App to Google Play Store for testing**

* In local environment:
  + Ensure Android Studio is installed
  + On master branch:
    - Ensure environment.prod.ts httpUrl is set to ‘<https://app.wecohear.com>’
    - Inc versionCode and versionName in cohear/ionic/ios/App/App/Info.plist and cohear/ionic/android/app/build.gradle
      * No need to inc this twice, ensure this matches the versionCode and versionName for the iOS Build
    - Push all changes to master
    - cd ionic
    - npm run build-android
      * When finished, this should open Android Studio
      * Edit ionic/android/app/src/main/res/xml/config.xml variable from “$ANALYTICS\_COLLECTION\_ENABLED” to “true”
      * Find all references to platform”(‘com.google.firebase:firebase-bom:30.3.1’)”
        + Update to remove the “” from around the ( )
        + Should be updated to: platform('com.google.firebase:firebase-bom:30.3.1')

The only quotes are single quotes *inside* the parentheses

* + - * Click Build > Generate Signed Bundle / APK…
        + A pop up will show up
        + Selected Android App Bundle
        + Click Next
        + If this is the first time building, on “Key store path”, click Create New

Create this key store, and then save it somewhere you can access it on your computer

Add this information to the current pop up window

Click Next

* + - * + Select “Release” build variant
      * Once build finishes, click the little pop up in the bottom right corner, and select “Locate”
        + This will show you the .aab file stored on your computer

Rename this file to cohearv\_.\_.\_.aab

Replace \_.\_.\_ with current version number of mobile build

Move this file somewhere easily accessible on your computer

* + - * Navigate to the Google Play Dashboard
        + Create a new release depending on use case (Testing or Production)

Upload this new cohearv\_.\_.\_.aab file to this new release

* + - * You can then manage the rest of the process in the Google Play Dashboard

Logins:

* [Google Play](https://play.google.com/console/u/0/developers/7851662366377461444/app-list)
  + Info@wecohear gmail
* [Apple Store](https://developer.apple.com/account)
  + [Info@wecohear.com](mailto:Info@wecohear.com)
  + Obamallama22!
* [GitHub](https://github.com)
  + [Info@wecohear.com](mailto:Info@wecohear.com)
  + Obamallama2022!
* Heroku
* [AWS](https://signin.aws.amazon.com/signin?redirect_uri=https%3A%2F%2Fconsole.aws.amazon.com%2Fconsole%2Fhome%3FhashArgs%3D%2523%26isauthcode%3Dtrue%26nc2%3Dh_ct%26src%3Dheader-signin%26state%3DhashArgsFromTB_us-east-2_e898b32dc4087034&client_id=arn%3Aaws%3Asignin%3A%3A%3Aconsole%2Fcanvas&forceMobileApp=0&code_challenge=M1nufNp5Sw0k9t79hKH9M0lWyzoVcXesw9sMKGnLg2k&code_challenge_method=SHA-256)
  + [Info@wecohear.com](mailto:Info@wecohear.com)
  + Obamallama2022!
* [Postmark](https://account.postmarkapp.com/servers)
  + Cohear
  + Obamallama2022!
* MongoDB
  + [Info@wecohear.com](mailto:Info@wecohear.com)
  + Obamallama2022!