



— CSUSB — LIBRARY STUDY SPACE

CSUSB LIBRARY STUDY SPACE
MOBILE APPLICATION

MAINTENANCE MANUAL

Prepared by:

MICHAEL GARAYSI (*Project Manager*)

MARK MARTINEZ (*Assistance Project Manager*)

Advisor:

DR. ARTURO I. CONCEPCION

Section 1: File Structures

Due to the presence of IDEA, Gradle, Android Studio, and Git files in the repository, the file structure contains a lot of miniscule files that don't need to be considered when developing for the app due to their self-explanatory nature.

The following are the core folders and files within necessary for the functioning of the app:

[FOLDER] app/src/main/

(contains most source files and the AndroidManifest.xml)

AndroidManifest.xml

The manifest for the app. Specifies the activities included in the app.

Has a permission request to use the network on a device. Specifies that the default activity is SplashScreen.

[FOLDER] app/src/main/java/edu/csusb/libraryspace

(contains all .java class files)

BookingActivity.java

The Booking Details screen that lets the user input their name, email, and public booking label. Sends a POST message to the servers when the input is valid. Receives information from ConditionsActivity.

ConditionsActivity.java

The Conditions of Use screen that displays the terms of use.

Hands-down the simplest of the activities. Passes information to BookingActivity from a previous calendar Activity (Group, Individual, Multimedia).

GetRequest.java

A network class that can do HTTP GET statements.

GroupActivity.java

The Group Study Room screen. Allows users to select a day, room, and hour. It gets information from the CSUSB Library servers by a POST and updates the relevant items. Sends information to ConditionsActivity.

IndividualActivity.java

The Individual Study Carrel screen. Allows users to select a day, carrel, and hour. It gets information from the CSUSB Library servers by a POST and updates the relevant items. Sends information to ConditionsActivity.

MainActivity.java

The home screen of the app. Allows users to select a room to reserve. Loads after SplashScreen. Allows users to download a .pdf map of the library, visit a browser link to the CSUSB website, and view a help/guide for the app.

MultimediaActivity.java

The Multimedia Collaboration Room screen. Allows users to select a day, room, and hour. It gets information from the CSUSB Library servers by a POST and updates the relevant items. Sends information to ConditionsActivity.

PostRequest.java

A network class that can do HTTP POST requests on behalf of other objects. Has two modes: one for the calendars (to get calendar data) and one for booking (to submit data).

SplashScreen.java

A launcher class that serves as the default activity of the app. It displays the logo of the app for 2 seconds and opens MainActivity afterwards.

[FOLDER] app/src/main/res/drawable

(contains the drawable assets of the app)

[FOLDER] app/src/main/res/layout

(contains the .xml layouts for the app)

activity_booking.xml

Layout for the Booking Details screen

activity_conditions.xml

Layout for the Conditions of Use screen

activity_group.xml

Layout for the Group Study Room screen

activity_individual.xml

Layout for the Individual Study Carrel screen

activity_main.xml

Layout for the Main Screen

activity_multimedia.xml

Layout for the Multimedia Collaboration Room

activity_splash_screen.xml

Layout for the Splash Screen

[FOLDER] app/src/main/res/values

(contains the strings.xml, dims.xml, and styles.xml files)

[FOLDER] apps/srs/main/assets/fonts

(contains the custom font files for the app)

dosis-medium.ttf

The "bold" font for headers and titles

dosis-regular.ttf

The "light" font for standard text and bodies

Section 2: Instructions

Here is how to acquire, compile, build, and deploy the app:

- 1) Acquire the source files (from repository or other means)
- 2) Run Android Studio 1.1 and do the following:
 - File > Import Project
 - Open the main project directory
- 3) To compile and build, do the following:
 - Run > Debug 'app'
 - The app will compile and build
 - In addition, Android Studio will ask you to choose to deploy to an emulator or a real connected device. (NOTE: The app does not function well on emulators due to network issues, so only test or deploy to real devices.)
- 4) To deploy the app, make sure debugging/developer mode is enabled on the target device. Follow Step 3 and choose your target device (ideally a real physical device and not an emulator).

Section 3: Good Implementations

The following are some of the positive and novel implementations of the app:

- The app is speedy and interacts nicely with the current configuration of the CSUSB Library website and database without imposing a large toll on the server.
- The app has a well-designed and easy to use user interface thanks to the guidance of our artist, Deanna Sulli.
- The app is simple to use and allows the user to do exactly what they want in the shortest amount of time.
- Several frequent users of the reserve system were asked to test the app, and they preferred using the app's interface over the web interface on a computer.

Section 4: Needs Improvements

The following are weak points of the current implementation:

- The app is incredibly sensitive to changes. If the CSUSB Library website changes or is offline, the app will most likely stop working. Careful and constant maintenance will be required for upkeep.
- The app does not respond well to the absence of an internet connection.
- If the team had more time, we would've added additional security measures to ensure proper use of the app. No one on the team had experience with network security protocols, so the app is not as secure as it could be.

These weak points will need to be carefully considered by future teams if they wish to continue development.

Section 5: Overall Recommendations

Regarding future development: the app can become more than what it is now. The following are possible additions that future teams can implement:

- Show public booking labels
- An iOS version
- Reserve multiple hours at once
- Reserve books and other library reservation items
- Further expand to make an entire CSUSB Library app