Communication interfaces / Database / User interface - UNIT TESTS REPORT - IOT PROJECT XXXXXXXX

|  | **interface name and purpose** | **Test results** | **Test code** |
| --- | --- | --- | --- |
| 1 | **Wi-Fi – HTTP server** – We can communicate with the esp32 on local network. | The espe32 played the role of a server , and started listening on a local server , on the laptop we connected to the server and sent commands to the esp32 using http .  We connected to the esp a light emulating a door lock , and a display to show messages. sending commands via http changed these hardware. | [**light\_from\_webServer**](https://drive.google.com/drive/folders/1qGuuX6NEl7SJz5cmbFMHMQM3AjY2PA3_) |
| 2 | **Wi-Fi –HTTP server with the app** – communicate between the app and the esp32 on local network. | The esp32 played the role of a server , and started listening on a local server , the app played the role of the client , and sent commands to the esp via http . | [light\_from\_the\_app](https://drive.google.com/drive/folders/1NLvjwy5n0-4zGQKdIWu3SsOVnUPH2tlt) |
| 3 | **Wi-Fi –HTTP server with the app (camera) -**  communicate between the app and the esp32-cam on local network. | Successfully paired ESP32-cam and Flutter app tested a widget that can let us view a web page that the esp32-cam hosts locally and see a live video on the app . | [light\_from\_the\_app](https://drive.google.com/drive/folders/1NLvjwy5n0-4zGQKdIWu3SsOVnUPH2tlt) |
| 4 | **Firebase real-time database** - create / change values from ESP32, get update on changed values | Successfully created and updated database values; changes reflected instantly in the Firebase console.   used the open source FirebaseClient library :  https://github.com/mobizt/FirebaseClient/tree/main | [RealtimeDatabaseNoAuth](https://drive.google.com/drive/folders/1xTbYcS49lntiO_SlrsYMyNjteIMnRwtE)  based on : https://github.com/mobizt/FirebaseClient/blob/main/examples/RealtimeDatabase/Simple/SimpleNoAuth/SimpleNoAuth.ino |
| 5 | **Firebase real-time database** -create / change / read values from Flutter app | Successfully performed all operations; tested with multiple data entries; instant database sync |  |
| 6 | **Firebase Authentication - authenticate users using firebase API and auth services.** | succeeded to add new users to the system using Firebase authentication services. |  |
| 7 | **Connection between Firebase and ESP32-CAM** | succeeded to encode the esp32-camera output to a string using <Base64.h> library . in a loop , the esp32-cam captures images and writes them after encoding to the firebase realtime database .  In the app we use the same library to decode the image . | [CameraToDatabase](https://drive.google.com/drive/folders/1iAJwdGWc9IB93Ie6FYufiLOFETxUvsTt) |
| 8 | **Fetching images from firebase to the app**  **PROBLEM** | Our current solution which is we encode an image to a string and push it to the firebase then the app fetches it and shows it to the user has a small issue , which is the smoothness of the live stream .  We tried another approach , which is to save a buffer in the database , and the esp pushes images to the buffer in a cyclic way and saves the last pushed index also in the database.Then the app fetches from the buffer also in a cyclic way but we have to make sure that we pick the right index .  But unfortunately it is not working as expected . |  |

**Notes :**

Decided to use firebase realtime database and not firestore . because it works a little faster , simple to use , and suits our needs .

tips :

* esp32 and firebase , use FirebaseClient [library](https://github.com/mobizt/FirebaseClient/tree/main) .
* make sure to define the correct database\_URL

**For the future :**

for now the esp32-cam works locally only .

thinking about sending images frequently to the realtime database and fetching it from the app . use **<Base64.h>** to encode/decode images to strings and then store/read them to/from the database.

**From the future** : we did it so it is working remotely and not only locally 😀.