ScanWAN

Group 6 : Mobile Application registering IoT nodes

Overview

Thanks to 5G expansion lately, IoT will be able to unleash its full potential which is naturally reflected by the need of efficient tools in order to register, manage and keep track of IoT devices.

Based on the LoraAlliance formalization of QR Code, we created a mobile application that allows one to register an IoT node by simply scanning its code.

Students

Elias El Yandouzi

Malone Julienne

Quentin Cambus

Referent teacher

Donsez Didier

Table of contents

Introduction

During this semester, INFO4 students had to realize a group project. Among the list of projects provided by the teachers, we had to create a mobile application to register IoT nodes in different networks. This said application should ease the laborious process of registering a node thanks to QR code.

Initially, long verbose strings (such as JoinEUI, DevEUI) are mandatory when one wants to register a node in a network. In 2020, the Lora Alliance decided to formalize QR code containing those required informations as the following :

Given:

SchemaID of D0
JoinEUI of 11-22-33-44-55-66-77-88
DevEUI of AA-BB-CC-DD-EE-FF-00-11
ProfileID of AABB-1122
OwnerToken of AABBCCDDEEFF
SerNum of YYWWNNNNNN
Proprietary of FOOBAR
CheckSum of AF2C

This requires size 4 and can only have ECC=Medium.

Here are the 88 bytes of data:

LW:D0:1122334455667788:AABBCCDDEEFF0011:AABB1122:OAABBCCDDEEFF:
SYYWWNNNNNN:PFOOBAR:CAF2C

And the QR code:

Excerpt of Lora Alliance TR005

Mobile application… how do we do?

Polytech'Grenoble doesn't provide any lecture about mobile application and our knowledge about this subject was roughly zero. Thus, the project began with discovering this vast world, picking a technology and learning how to use it.

The possibilities are countless however our choice had to be consistent and coherent with our needs. To resume, an easy and fast to learn technology, deployable on both Android and iOS and that has a big community.

Therefore, our choice was between Flutter and React Native, but the fact that we had to learn Dart to use Flutter was a hindrance. Hence, we started our project with the React Native framework and Javascript (that all of us knew a bit).

Native development wasn't an option since the very beginning. It would mean two different code projects, and rhymes with more problems. Moreover, such a choice doesn't bring any benefit for our project. We need to create a simple interface to let the user provide mandatory informations that we send to a network in order to register a node.

So during the first couple of weeks, we tried to design and think about our application and learnt React Native. It was a key moment, since a good conception leads to an easy and efficient development.

Thus, we tried to follow as much as possible what was presented during the GL lecture. Based on the key features we identified, we have striven to create a simple and detailed task that we referenced on. This way, we could work independently in a flexible manner.

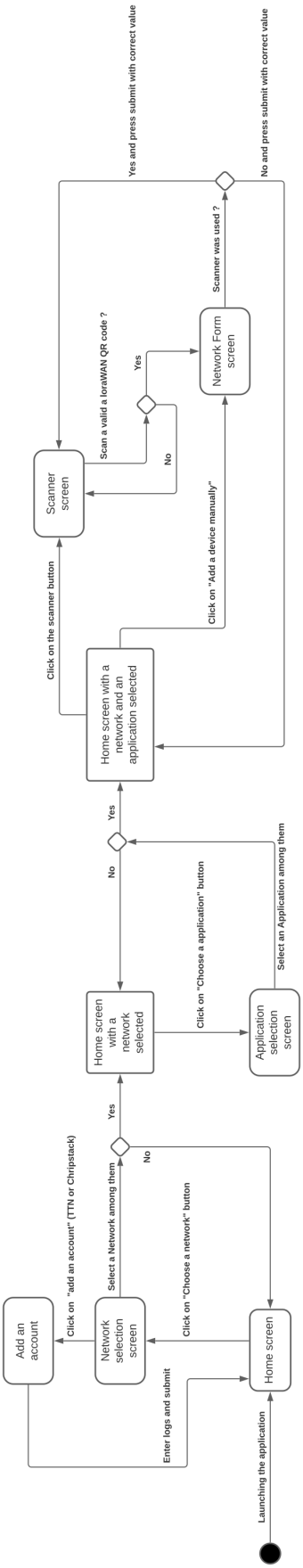


Diagram - Registering a node

The development, what did we do ?

At this point we had a clear idea on how to create interfaces, use a redux store to share variables between components and so on. The only missing part was the use of the API from the different networks. Didier Donsez, our referent, gave us access for Chirpstack and The Things Network API.

The application implies two major parts : the frontend and the backend. We tried as much as possible to explore and share the different tasks to ensure that we all get decent knowledge of the project in its entirety.

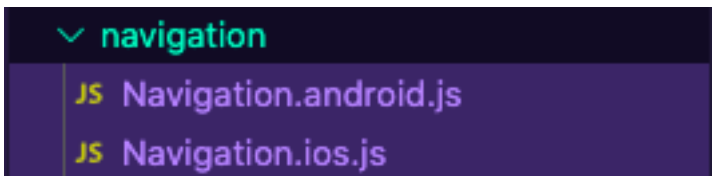
Frontend

It is the interface upon which the user will interact. The project, as described by the teachers, is intended for mass registration of nodes thus it has to be straight-forward.

The design is basic but functional and very clear. Regrettably, the group has any notion of design, so to improve it (at least slightly) we used UI Kitten. This is an open source and free library which provides enhanced design components. We feel it is important to remind that the design part is usually apart from the coding process and made by someone who studied the field.

Once each view were created, we had to add navigation. Fortunately, React Native comes with a huge diversity of module, so we could use and install it easily. Even though we said we chose React Native to avoid platform specific development, the navigation impose some changes. Android devices have a native button, to go back, always displayed on the screen contrary to iOS. Hence we added a top bar on iOS version of our application.

The creators of React Native were certainly aware of this kind of problems and eased the developer life allowing to rename files as the following to be considered on the proper platform :



Navigation platform specific files

Backend

In other terms, the biggest part of this project. The key feature was to scan a QR Code and being able to parse the information it contains. In addition, we had to find a way to store and share variables between the different views and finally be able to make a request to an API.

Thanks to a special module, provided by the community, we can easily scan and parse the content of a QR Code. However it was quite a challenge to make it works since the module was not adapted to our usage. Initially it was trying to scan forever even when the user left the dedicated view. This problem has been solved by using state variables.

Here again, there is a specific module to store and share variables : Redux. For a given state, it can store a bunch of variables that can be restored anywhere in the application. This tool was required to save the network and application selected by a user for the registration process.

The credentials for each network should have been initially stored in the keystore. However, it was quite complicated and prone to bug. Hence, we decided to use a much less secure but effective way : RNFS. The API key and the couple login password are stored in plain text file on the device storage.

Finally, It turns out that API was the most complicated part due to a lack of information, especially with TTN. Indeed, they are at this time migrating their API from V2 to V3. So we struggled to find the base URL for our API request and it was quite an experiment for all our requests.

The bugs, the terrible bugs

What is a development process without having bugs to solve ? Obviously, a lot of bugs occurred, here comes a list of the main ones :

QR Code Scanner

Problem: Once the user started to scan a QR Code, the application wouldn't stop scanning in the background.

Solution: Created state variable to turn on/off the camera component.

The Things Network API

Problem: Unable to find where to send our request and how to format them.

Solution: After many hours seeking for a proper documentation to know how to use the TTN API. Lot of experimentation on our side too.

Certificate of Authority

Problem: Couldn't make any API request due to untrusted CA.

Solution: The user has to accept all the certificate. In a certain way, we corrupt the trust chain to make the API call possible.

iOS Compilation

Problem: Each time a module was added, the compilation failed even though the modules were installed and the Podfile updated.

Solution: At the time, there is no generic solution. Spent so many hours to try to understand the error message and solve the corresponding problem.

Conclusion

This project was a great opportunity for us to discover new technologies and new field of application. It was for us the first time in a real mobile application project and we are really proud to have achieved this project.

Moreover, we learn a lot about Javascript and the React Native framework, which is used a lot in mobile application and web development. We strongly believe that it will be a strong asset for us in our future career.

Because we were starting from scratch and totally free, we had to think about everything to create a working application. It was a great experience and a perfect occasion to apply our GL lecture knowledge.

Given the time we had, the application is basic but working. However, if we were asked what would we do to improve the application, here are some ideas:

- A proper design to enhance the user experience
- The use of OCR to read long strings such as API keys or AppEui
- Use tags and variables to add more options on forms
- Take charge of more networks (private or public)
- Deployment on application marketplace