"TagBuddy" - a safe and secure messaging app for kids
## Niall Meade Ardscoil Rís, Limerick

Tag Buddy messaging

Aims

1. Develop a safe and secure child-friendly messaging platform prototype.
2. Investigate existing messaging services and their anti-bullying protocols.
3. Investigate features to make messaging services more child-friendly.
4. Develop a quick, secure and appealing app using asynchronous design.

Current Messaging Platforms

Current messaging services offer little to no consideration for children especially when it comes to who you can talk to, anti bullying policies and also the content that can be accessed or shared. Parents who let their children access these services are faced with the worry/uncertainty regarding content, who they can talk to and the risk/threat posed. On the other hand if parents restrict usage of these messaging platforms their child may be isolated and experience peer pressure as a result. TagBuddy aims to solve these problems by creating a secure, safe (child) and transparent (parent) messaging platform.

Surveys
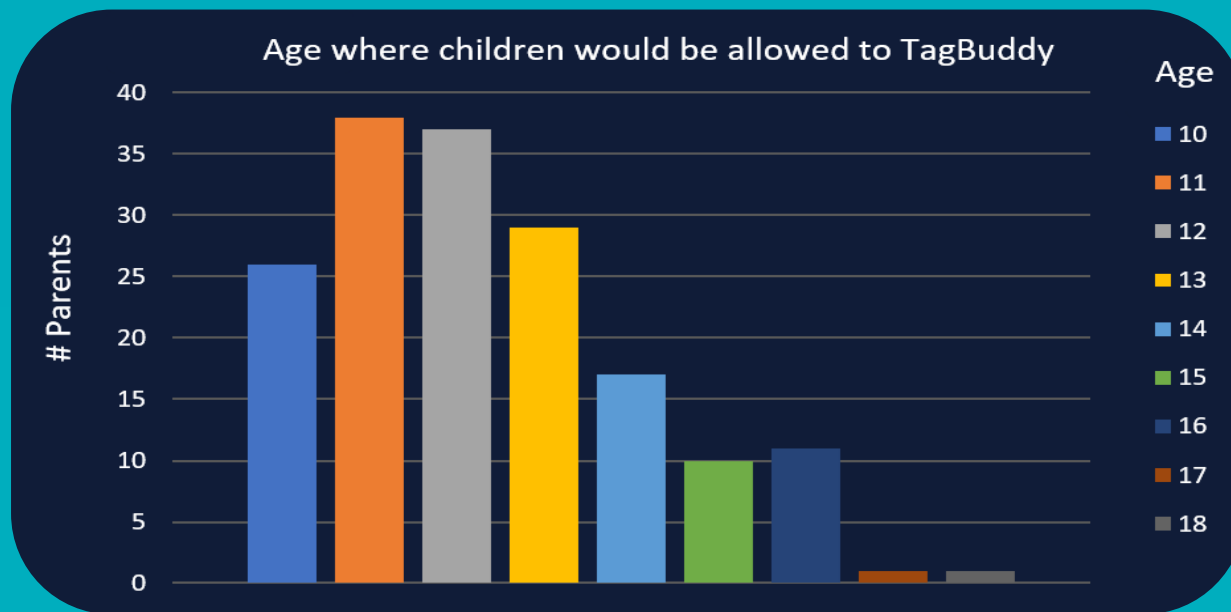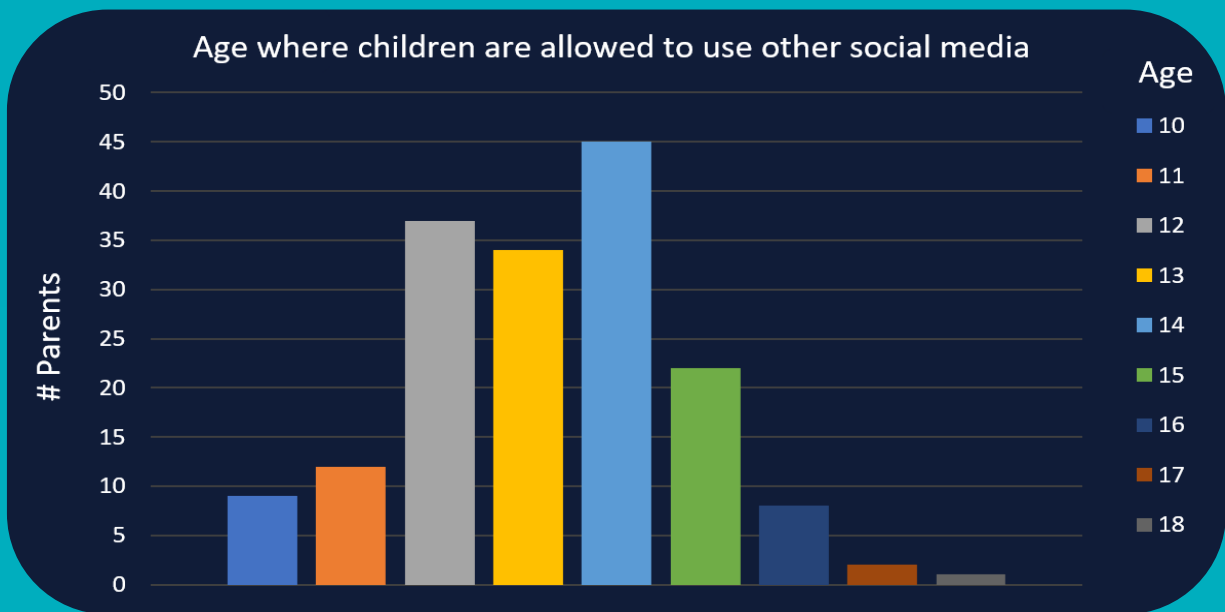
I carried out a survey in my school to get a better understanding about parents views on messaging apps. The results were very consistent. I discovered that on average children are allowed to use these apps when they are **14 yrs old**. Surprisingly when presented with the TagBuddy concept this age dropped to **11.4 yrs**. Although the aim of TagBuddy is not to get children on social media at an early age, this change in attitude shows that parents would feel a lot more comfortable with an app of its kind.

Application Design Diagrams

Before I started development on the application I decided to do some research into software/app design. This was to make sure I avoided any design issues I might have faced in development. I found three ways of representing my design in diagram form: flow, sequence and Jacobson use case diagrams. From these diagrams I was able to understand the asynchronous flow of my app, the functions I might need and any error catching.
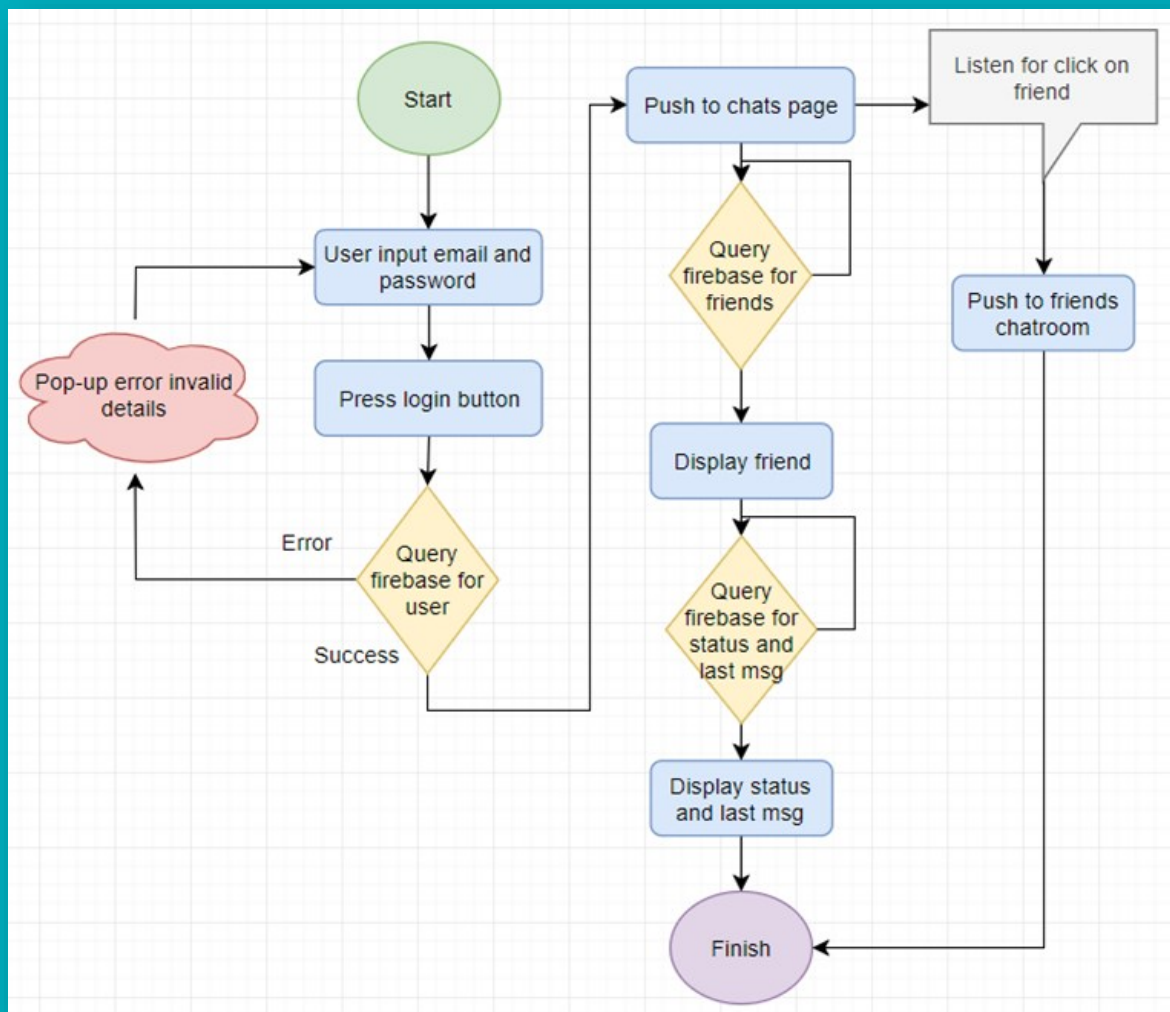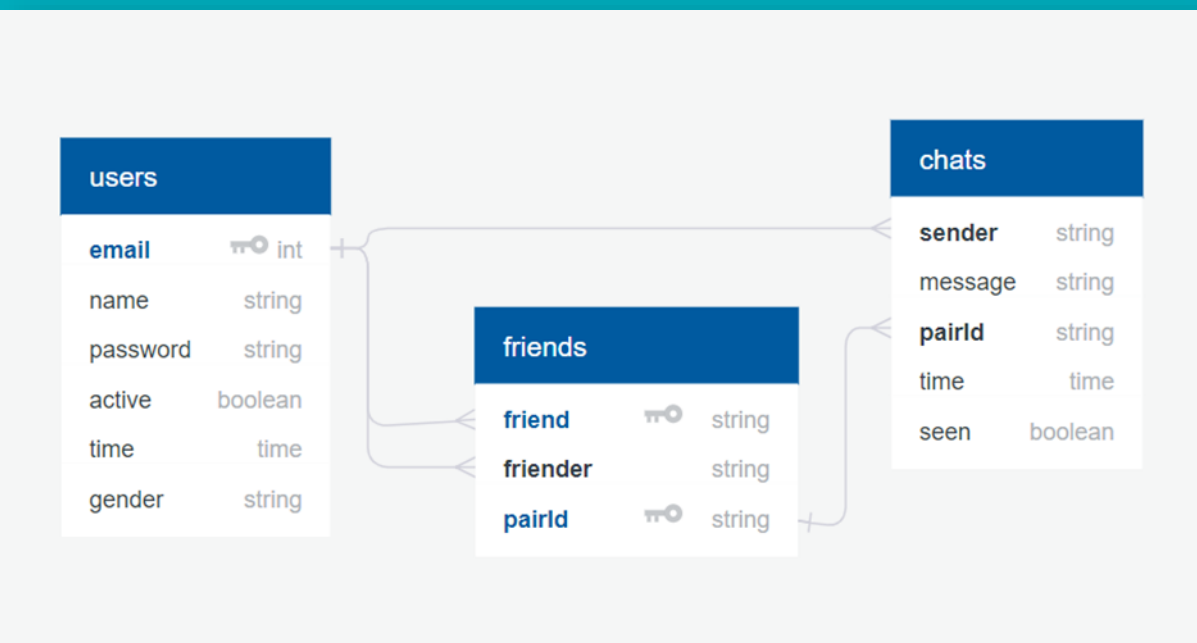
Development Platforms

I used the Ionic platform to build this app. It allows you to develop using common web frameworks and compile a your package into a single native apk (cross platform development). It includes a large set of native APIs to interface with platform hardware and software such as NFC and the GPS.
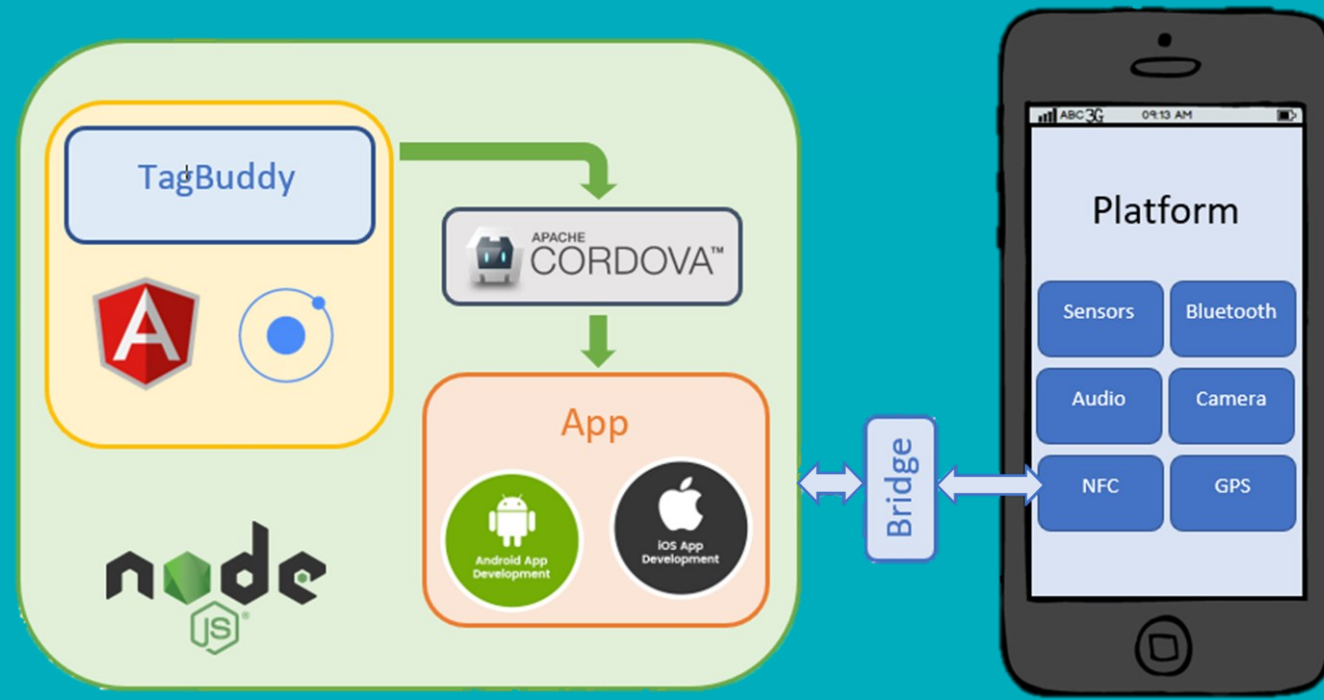
Firebase (Google) is a mobile and web application development platform. Instead of having any server side code I could simply interface with the platform with my unique API key using it as a simple document storage database. The firebase can be queried/ updated asynchronously using it's own node (Angular JS) API.

Caesar Shift Cipher Encryption

To make sure the data held in firebase is secure I decided to encrypt it. To do this I used the Caesar Cipher, a very simple, easy to understand cipher scheme. It includes a key which is just a number (pos or neg) and uses it to shift the input string by that amount in the alphabet. If you don't have the key the strings held in the firebase document look like gibberish!

Can you tell me my key? YflGziid

Plain text
Cipher text
Key = 19

Promises and Observables

Requests to the firebase and chatbot are made asynchronously in my app. The two concepts used in angular are Promises and Observables. A promise sends out a request, the main thread continues but the .then() function isn't trigger until data comes back. An observable is something that can .subscribe() to data so is basically multiple promises with some other functionality.
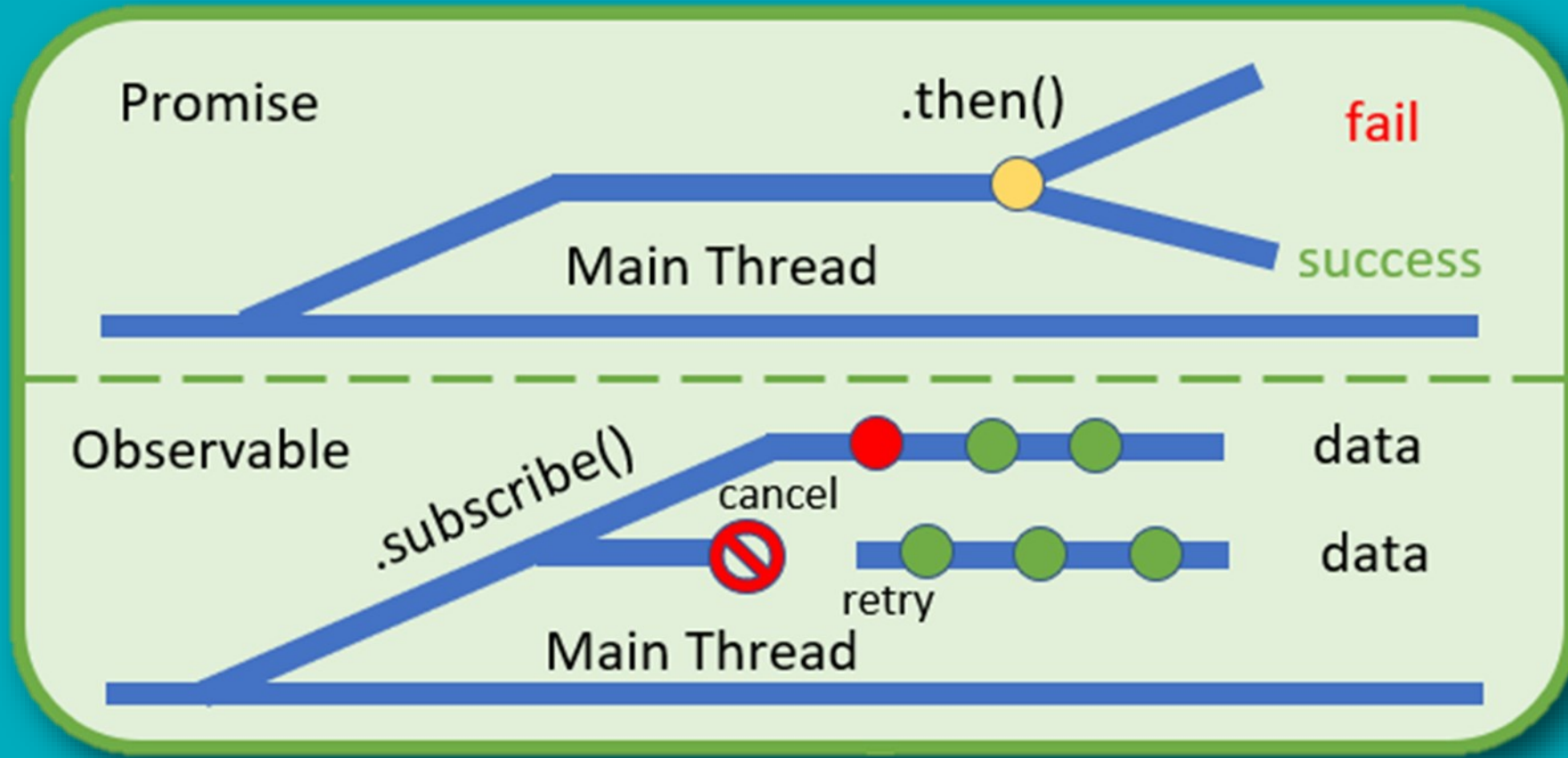
AI Chat Bot

I decided to add a chat bot to my app to address bullying i.e. giving the child "someone" who they feel they can talk to. The bot I designed works on a schema. In the future I hope to look at how I could design a better bot using a neural network. My bot "Buddy" can talk about school, family and problems. I send a HTTP post (promise) using the ionic natives HTTP API to the bots API end-point hosted on Snatchbot.com and get a response based on the message sent.
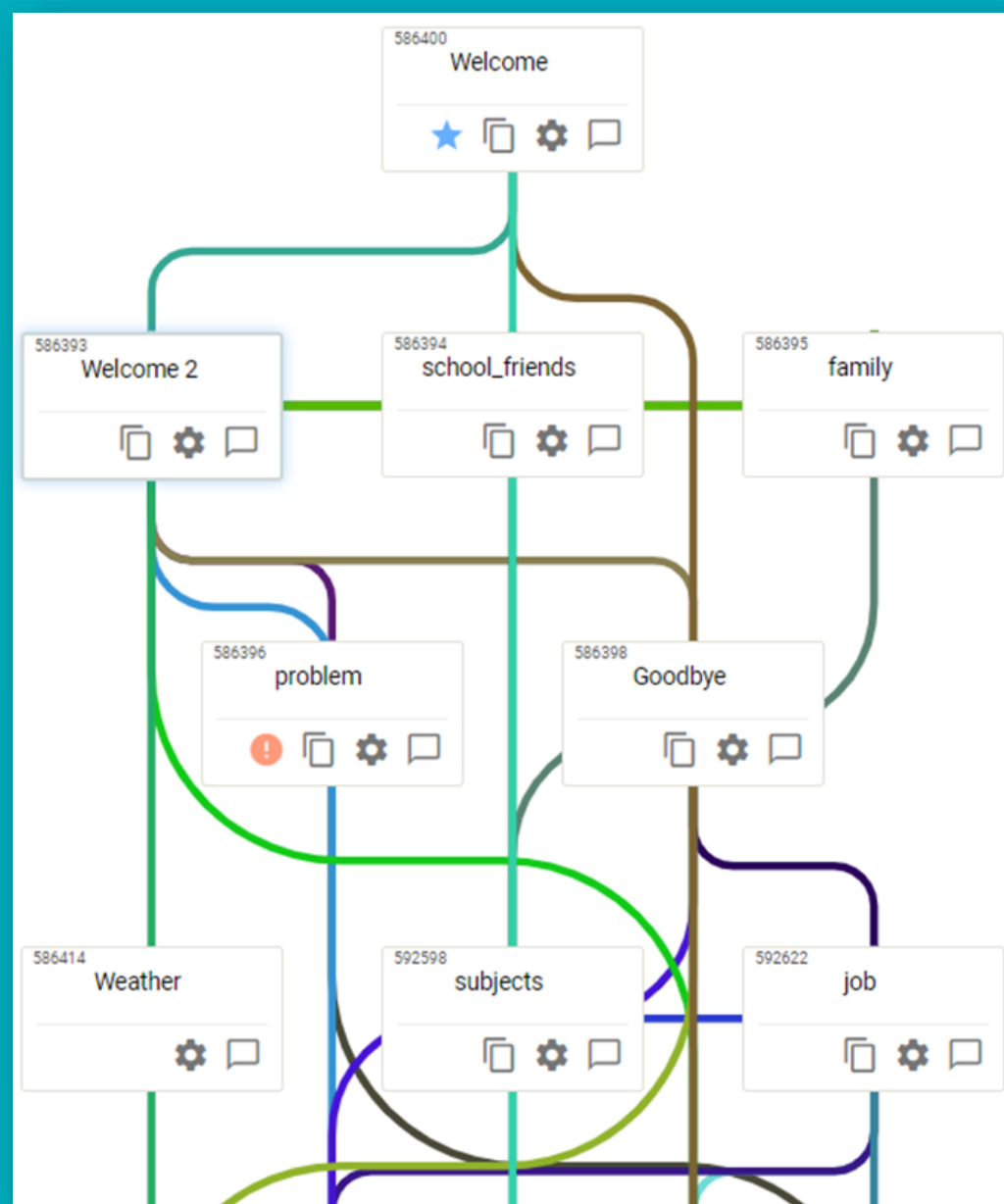
NFC

Near-field communication (NFC) is a set of communication protocols that enable two electronic devices to exchange data. In my app to add a friend you need to hold your phone back-to-back with a your "buddy". This gets rid of the risk of talking to strangers online. I used the ionic native NFC API to add an NFC listener and share a key (email) with a friend.

Testing & Conclusion

A prototype of the TagBuddy app was built and tested successfully the main features include:

. Buddy the chatbot

. Add friends via NFC

. Realtime chat

In the future I hope to build a better chatbot , investigate picture sharing as well as an app for parents to allow them monitor their child's activity on TagBuddy.