Documentation of Gnokky

Sommario

[Introduction 3](#_Toc140676613)

[Analysis & planning 3](#_Toc140676614)

[Setup Environment 4](#_Toc140676615)

[Installation 4](#_Toc140676616)

[Expo CLI 4](#_Toc140676617)

[Use Expo CLI 5](#_Toc140676618)

[Install & Setup Android Studio 5](#_Toc140676619)

[Create your first project 5](#_Toc140676620)

[Implementation 6](#_Toc140676621)

[Folder structure 6](#_Toc140676622)

[Admin 6](#_Toc140676623)

[Auth 7](#_Toc140676624)

[Login Page 7](#_Toc140676625)

[Chat 8](#_Toc140676626)

[ChatPage 8](#_Toc140676627)

[ListResearchChat 9](#_Toc140676628)

[ChatTemplate 9](#_Toc140676629)

[Message 10](#_Toc140676630)

[Home 10](#_Toc140676631)

[Models 10](#_Toc140676632)

[Utils 10](#_Toc140676633)

[SpotifyUtils 10](#_Toc140676634)

[Firebase 10](#_Toc140676635)

[FirebaseErrorManagement 11](#_Toc140676636)

[Globals 11](#_Toc140676637)

[User 11](#_Toc140676638)

[Navigations 11](#_Toc140676639)

[Notifications 11](#_Toc140676640)

[Post 11](#_Toc140676641)

[Profile 11](#_Toc140676642)

[Repost 11](#_Toc140676643)

[Search 11](#_Toc140676644)

[Stories 11](#_Toc140676645)

# Introduction

Our goal was to create a social media (like Instagram) with the framework “expo”. Expo is framework of react-native, that is a language based on JavaScript and is mainly used for creating mobile application. We chose to work with expo because we would like to learn to use this framework.

For the database we used “firebase” that is a cloud database offered by google and we chose it because it can offer some useful function (for example the email confirmation when you create an account).

# Analysis & planning

Before we start with the implementation, we dedicated a few days to analyse our project. We made the assignment book, that contains all the functionality that our app must have, some mock-up, a use-case diagram, the test-cases, and a Trello board.

Trello is a website and we used it for arrange our project. In the board there are lists of activity and every morning we make a little meeting where we chose which activity we want to do during the day.

Immagine che contiene testo, schermata, software, Software multimediale

Descrizione generata automaticamente

Figure 1 Trello board

# Setup Environment

There is a little guide for setup the environment on Windows 10/11.

## Installation

To develop applications with Expo, you will want to start with two tools:

* Expo CLI
* Expo GO (Auto installed when you start your first project)

### Expo CLI

To use Expo CLI, you need to have the following tools installed:

* NodeJS
* Git

#### NodeJS

Download the installer here → [https://nodejs.org/en](https://nodejs.org/en%20)

Only Node.js LTS releases (even-numbered) are recommended.

Once the download has finished, run the installer, and click next leaving everything default except for this one where **I recommend you tick the checkbox:**

Immagine che contiene testo, schermata, software, Pagina Web

Descrizione generata automaticamente

Figure 2 NodeJS download

Next, the terminal will be opened, follow the instructions written on it and wait until the installation is finished.

#### Git

To install Git, click here → [https://git-scm.com/download/win](https://git-scm.com/download/win%20) and choose:

*“64-bit Git for Windows Setup”*

Simply run the exe and let everything default.

## Use Expo CLI

To check if Expo CLI is it working type in terminal:

**npx expo -h**

It will just show you useful information about npx expo command.

## Install & Setup Android Studio

## Create your first project

Open CMD and type

**npx create-expo-app my-app**

and once it has done

**cd my-app**

Install dependencies (only if you need to develop this application also for web)

**npx expo install react-dom react-native-web @expo/webpack-config**

To start the project type

**npx expo start**

# Implementation

## Folder structure

Four our application we tried to keep an ordered structure for the folders of the project. In fact, we decided to divide all in many sub-folders, one for each functionality of the application.

Immagine che contiene testo, schermata, design

Descrizione generata automaticamente

Figure 3 Folders Structure

## Admin

## Auth

In the Auth Folder there are the three GUI (Graphical User Interface) used for the auth part that are, the Login page, the Register page, and the Waiting page. Over these three files there is the last file (AuthUtils) that contains some function needed for check if the registration and the login worked correctly.

Immagine che contiene testo, schermata, Carattere, Blu elettrico

Descrizione generata automaticamente

Figure 4 Auth Files

### Login Page

The login page is simple, there is just the logo f the application, two fields for the email and the password, the button for sign in and the button for go to register a new account.

Immagine che contiene testo, schermata, Cellulare, Dispositivo mobile

Descrizione generata automaticamente

Figure 5 Log in Page

When you click on the “SIGN IN” button the function “handleLogin” from the file AuthUtils. Here, we are going to validate email and password with a trim (that remove space before and after the word)

## Chat

In the chat folder there is the chat page and other component related to the chat functionality.

Immagine che contiene testo, schermata, Carattere, Blu elettrico

Descrizione generata automaticamente

Figure 6 Chat Files

### ChatPage

The ChatPage file contain the list of the existing chat. In the top there is a search bar for searching in the chat that already exists or new user for start a new chat.

Immagine che contiene schermata, testo, design

Descrizione generata automaticamente

Figure 7 Chat Page

### ListResearchChat

When you click on the search bar it opens the ListResearchChat page and filter the existing chat and other user by the keyword that you digit in the search bar.

Immagine che contiene testo, schermata

Descrizione generata automaticamente

Figure 8 ListResearchPage

### ChatTemplate

The ChatTemplate component is used for rendering the chat. In the top of the chat there is the username and the profile picture of the user you are writing to and in the bottom, there is a textbox for write a message and the button for sending it.

### Message

The Message component is used for rendering every message in the chat. The background color of the message and the position change based on if it is your message or a message from the other user.

Immagine che contiene testo, schermata, Carattere, bianco

Descrizione generata automaticamente

Figure 9 Messages example

The dark grey message is your message, and the light grey message is the message from the other user.

## Home

## Models

In the models folder we don’t any component but only utility class.

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Figure 10 Models files

### Utils

All the files that the name finish with “Utils” (apart the one called “SpotifyUtils”) are classes that communicate with the database. We decided to put all the database communication in these files because like this is more ordered.

### SpotifyUtils

The SpotifyUtils file is the only file that finish for “Utils” that doesn’t communicate with the database, but it communicates with the Spotify API.

### Firebase

This file simply contains all the data that JavaScript need for connect to the database.

### FirebaseErrorManagement

This file contains only an array with all firebase errors and their respective message to be displayed to the user. The errors then gets displayed like this.

Immagine che contiene testo, schermata, Carattere, biglietto da visita

Descrizione generata automaticamente

Figure 11 Error Pop-up

### Globals

In the Globals file there are a bunch if constant that approx. all the files need them. For example, there are declared all the colours used in the application, so if we must change a colour we can change from this file and in all the application change.

### User

The user class it is a class used for saving the data of the current user logged in.

Immagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamente

Figure 12 User class diagram

## Navigations

## Notifications

## Post

## Profile

## Repost

## Search

## Stories

The Stories folder contains all the files related to the stories function.

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Figure 13 Stories files

### NewStoryPage

The NewStoryPage component is the page for create a story, and it look like this.

Immagine che contiene testo, schermata, design

Descrizione generata automaticamente

Figure 14 NewStoryPage

In the bottom of the page there is a bar where you can select the media you want to post in your story. You can choose between open a photo in the gallery, take a photo or post a song with the Spotify API.

After you choose the media to post, the bottom bar change like this.



Figure 15 Updated bottom bar¨

With this bar you can change the media clicking on the arrows or add a text hover your media clicking on the “Aa”.

### DraggableTextInput, ColorPicker, TextEditor

When you add a text to the story it generates the component DraggableTextInput. This component adds a text to the image that you can drag wherever you want, change colour between the colours from the ColorPicker and change the font size with the TextEditor.

Immagine che contiene testo, schermata, multimediale, software

Descrizione generata automaticamente

Figure 16 DraggableTextInput and ColorPicker

In this case, the string “Hello Word” is a DraggableTextInput and in the top we can see the ColorPicker. If you want o change the font size, you can click on the icon in the bottom right and the ColorPicker will get replaced with the TextEditor.