Tampere University of Applied Sciences

GeoQRNavigator Application

Final assignment for native Android

Ej Sobrepeña

FINAL ASSIGNMENT FOR NATIVE ANDROID
October 2023

Degree Programme
5G00DL99-3004 Mobile App Development 2

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Bachelor's Degree Programme in Software Engineering

AUTHOR : Ej Sobrepeña
GeoQRNavigator Application

Final Assignment For Native Android 15 pages, appendices 4 pages
October 2023

GeoQRNavigator is an Android application that leverages the power of QR codes and location services to provide a unique geographical navigation experience. This report presents the design, implementation, and testing of the application, along with the challenges encountered and future improvements.

CONTENTS

1 INTRODUCTION

The advent of QR codes has revolutionized data sharing, providing a simple and efficient method of encoding information. GeoQRNavigator harnesses this potential to encode geographical coordinates in a QR code, offering a novel approach to navigation.

2 SYSTEM REQUIREMENTS

The application is compatible with Android 7.0 (Nougat) or higher. It utilizes Google’s FusedLocationProviderClient for location services and the ZXing library for QR code scanning.

3 SYSTEM DESIGN AND IMPLEMENTATION

The application comprises several components, including a QR code scanner, location services, and user interface elements. The primary activities include checking for necessary permissions, scanning QR codes, parsing geographical coordinates from the QR codes, calculating distances between locations, and displaying information to the user.

3.1 Location Services

The application uses FusedLocationProviderClient to access the device's current location. It checks for location permissions at launch and before each scan operation. If permissions are not granted, it requests them from the user.

3.2 QR Code Scanning

The application uses the ZXing library to scan QR codes and barcodes. The scan operation is initiated by the user through a button click. When the scan operation is initiated, the camera automatically tries to detect QR codes and barcodes. The application handles various scenarios such as scan cancellation and scanning of non-geo-encoded QR codes. It also includes additional features like beeping on successful scans and flashlight functionality for low light conditions. The flashlight can be turned on by pressing the volume up button and turned off with the volume down button during scanning. This feature enhances the usability of the application in low light conditions.

3.3 User Interface

The user interface displays the current location of the device, the location encoded in the scanned QR code, and the distance between these two locations. It also provides feedback to the user in case of errors or exceptions.

3.4 Saving and Restoring State

The application includes functionality to save its state when it's stopped (e.g., during a screen rotation) and restore it when it's restarted. This ensures that data like the current location, qr code location, and distance are not lost during such transitions.

4 TESTING

The application has been rigorously tested on various Android devices with different OS versions and with different types of QR codes.

5 CHALLENGES AND LEARNINGS

The development process presented several challenges related to error handling, such as handling denial of location permissions and failure to get current location. Overcoming these challenges provided valuable insights into robust error handling in Android applications.

6 CONCLUSION AND FUTURE WORK

GeoQRNavigator demonstrates an innovative use of QR codes for geographical navigation. Future enhancements could include map visualization of locations and turn-by-turn navigation instructions.

REFERENCES

Cambo Tutorial. 2022. Implement Barcode QR Scanner in Android Studio Barcode Reader | Cambo Tutorial. YouTube. https://www.youtube.com/watch?v=jtT60yFPeII&t=144s

APPENDICES

Figure A1: App after installation

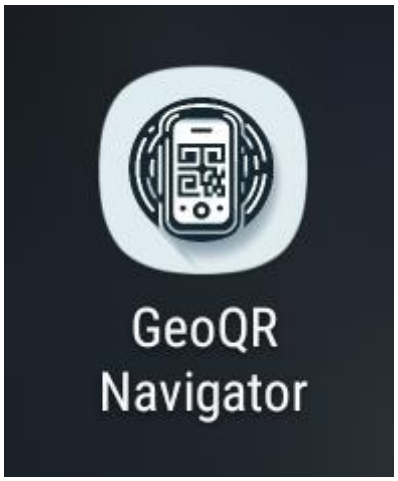

Figure A2: App after opening without location permission.

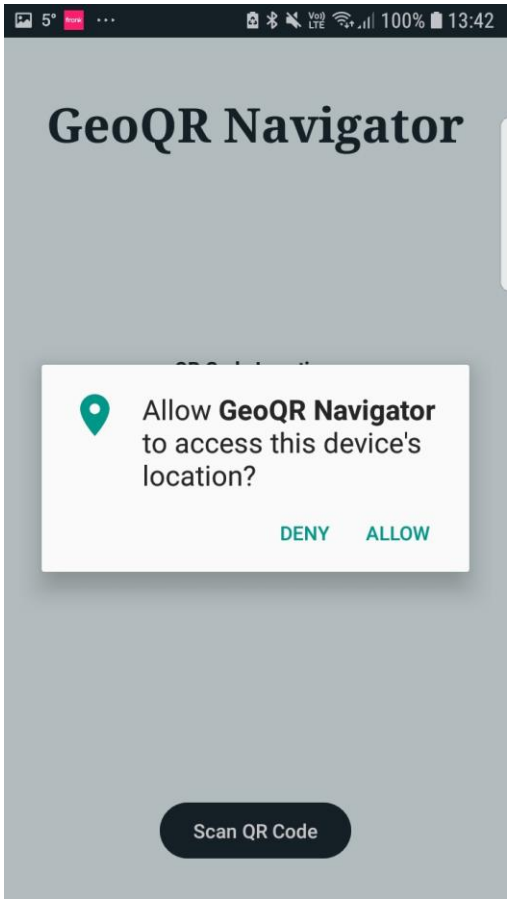

Figure A3: App requesting for camera permission

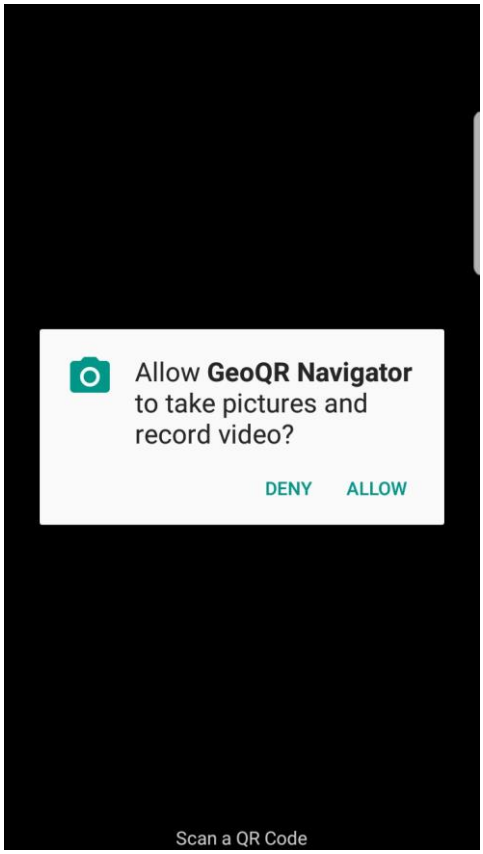

Figure A4: After successful scan

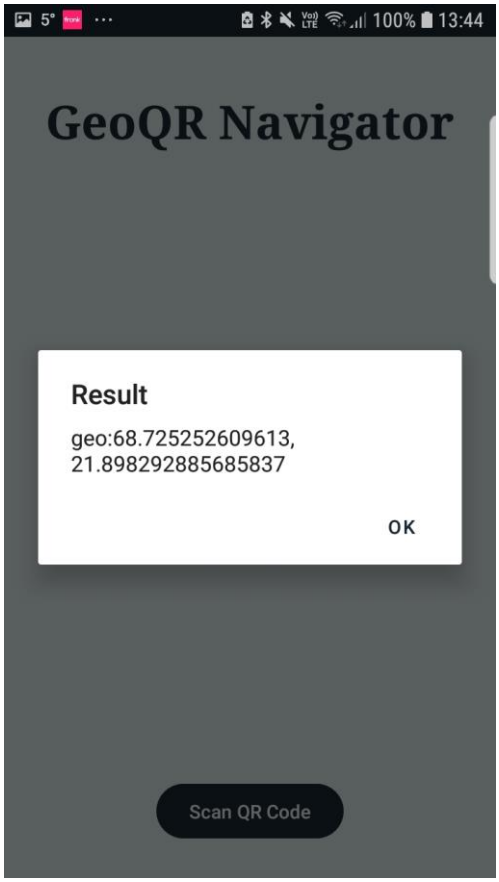

Figure A5: Showing results after successful scan scan

Error handling: