Technical Project Report - Flutter Module

PanicNet

Subject:	Universidade de Aveiro - Computação Móvel
Date:	June 8th 2025
Students:	113655: Abel Teixeira (35%) 114429: José Mendes (65%)

Index

Introduction

In the class of ICM (Introduction to Mobile Computing), we're asked to work on a second project, which consists in developing a fully functional mobile application, using the framework Flutter. The framework Flutter is a versatile and dynamic framework developed by Google, that emerged as a game-changer in the universe of mobile app development. Offering a robust set of tools and an intuitive programming environment, Flutter empowers developers to create stunning cross-platform applications with ease.

Project Context

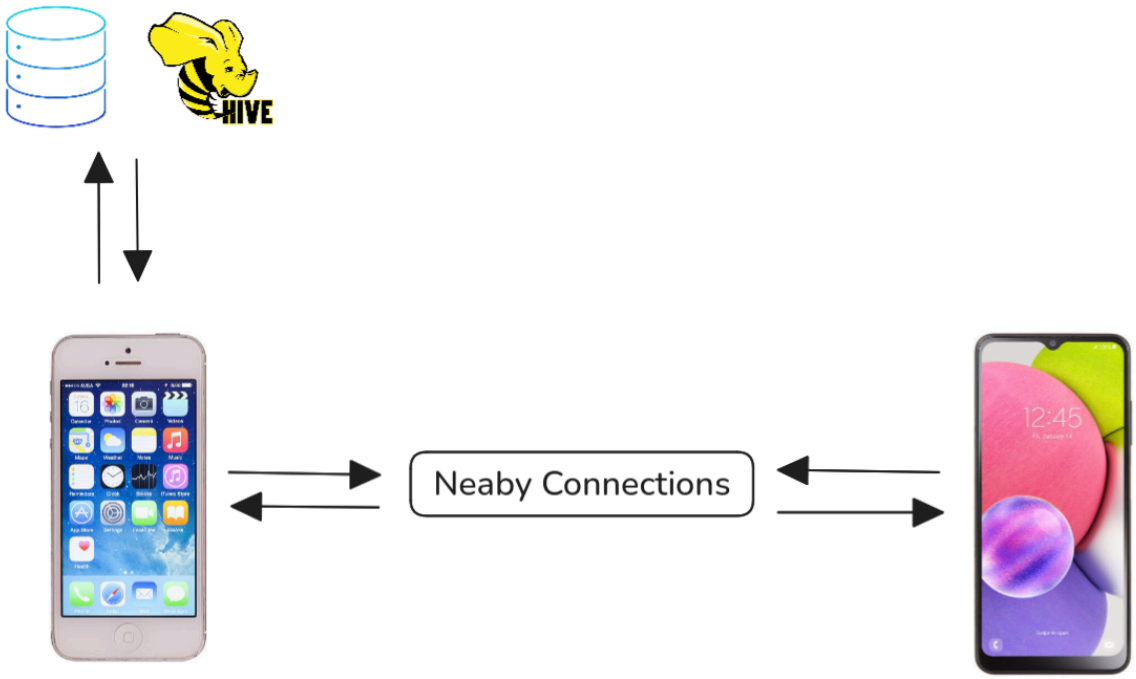
For this project, under the guidelines given by our professor, our group created PanicNet. This App is designed to assist individuals in emergency situations. It provides users with a simple and effective way to send SOS alerts to the nearby phones, allowing them to quickly seek help when needed. When the user is in an emergency situation, it shakes the phone and takes a picture of its surroundings and sends it to a nearby device that is connected.

Requirements

Our group defined the following requirements for us to make our system fully functional:

- Device interaction and communication:
  - Enable explicit communication and interaction between mobile devices.
  - Establish a connection using *Nearby Connections*.
  - Implement device discovery and a pairing mechanism.
- Display real time information:
  - Develop an user interface that provides real-time feed.
  - Create a "chat like" page to send pictures.
- Using various sensors:
  - Give permission to use the device sensors.
  - Ensure that the device can use the sensors and retrieve data from them.
  - Use Accelerometer to check when the device is moving.

Architecture Diagram



The architecture of our system is based on 3 essential elements: two devices with sensors (Accelerometer, gyro, etc) and Hive database for each device. Each element is crucial for the app to work properly.

Workflow

The workflow represented by our architecture unfolds as follows:

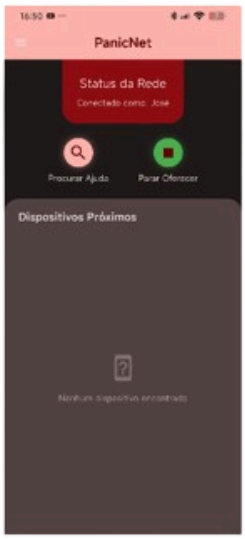
- Login
  - Initially, the user inputs an user name, for data storage and displaying to others.
- Device connection and configuration
  - When logged in, the user can choose a device to connect and pair.
  - After the pairing the two devices connect and can start sharing data.
  - One device is the sender and the other the receiver.
- Sensor measurements
  - The app is depending on gestures to send an alert, so the accelerometer needs to read the movement of the device.
  - After detecting movement, the phone needs camera access to take a picture.
- Data storage and visualization

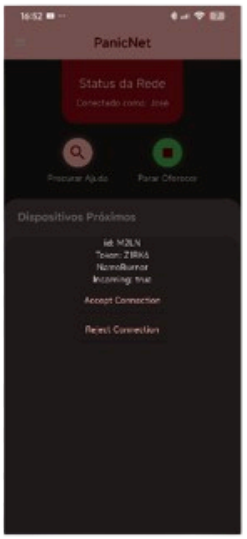
- The Hive database in each device serves as an alert repository, storing pictures of other devices alerts received previously.
- The app accesses the database to check the alerts and show them to the user.

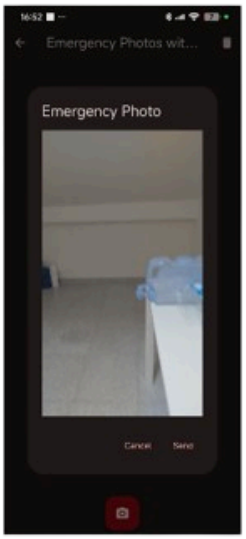
Hive was chosen as the local storage solution due to its simplicity, efficiency, and excellent performance on mobile devices. As a NoSQL database, Hive is optimized for fast read and write operations, which is crucial for an application that needs to store and access data quickly, such as user settings and alert history.

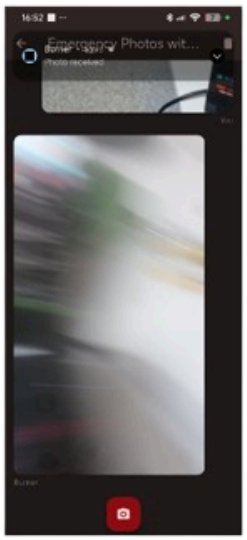
Nearby Connections was chosen as the solution for communication between devices due to its ability to create high-performance peer-to-peer networks without relying on an internet connection.

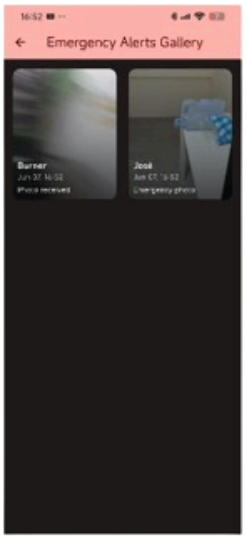
Screens

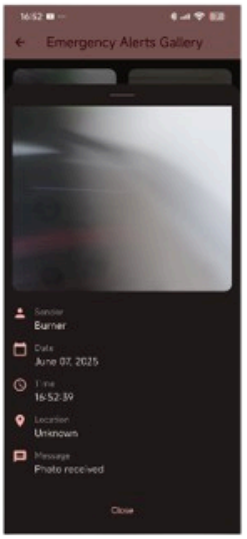












Dependencies

1. cupertino_icons
2. sensors_plus
3. provider
4. nearby_connections
5. hive
6. hive_flutter
7. flutter_animate
8. lottie
9. permission_handler
10. flutter_local_notifications
11. camera
12. geolocator
13. flutter_map
14. latlong2
15. image_picker
16. path_provider
17. intl
18. flutter_image_compress

Conclusion

To conclude our project in the Introduction to Mobile Computing (ICM) class, flutter, by Google, turned out to be a real game-changer for us. It's user-friendly and packed with features that let us make an app that not only works across platforms but also looks and feels great.

Project Resources

Resource:	**Available from:**
Code Repository	https://github.com/ttabelhaxd/Projeto2-ICM