



**CERN : The European
Organization
for Nuclear Research
CMS : The Compact Muon
Solenoid**



SUMMER STUDENTS PROGRAM 2023

Project report

**Contact and project management
mobile application**

Student :

EL Ouni Hadhami

Supervisor :

Zein Zebib

25/06/2023 - 18/08/2023

Table of contents

Introduction	3
Project description.....	5
Technology Stack	6
1.Frontend Architecture:.....	6
2.Backend Architecture (Flask):.....	7
3.Database:	7
4.Integration and Communication:.....	8
Functional Requirements:	9
Non-Functional Requirements	10
1.Security and Privacy:	10
2.Maintainability and Updates:	10
3.User Interface:	10
Database design:.....	12
1.Class diagram:	12
2.Relational schema:	12
Development Cycle	13
Realization.....	14
Conclusion.....	16

Figures table

Figure 1: React Native logo	6
Figure 2 : Flask logo	7
Figure 3 : MySQL logo	7
Figure 4 : Architecture	8
Figure 5 : prototype of the application	11
Figure 6 : Class Diagram	12
Figure 7: Single contact page	14
Figure 8 : New contact page	14
Figure 9 : Home page	14
Figure 10 : New sub-group page	15
Figure 11 : Share group page	15
Figure 12 : reminders page	15
Figure 13: filter groups page	15

Introduction

The CMS (Compact Muon Solenoid) experiment is one of four main experiments CERN. It is also one of the largest international scientific collaborations in history, involving about 5500 particle physicists, engineers, technicians, students, and support staff that unites over 250 institutions hailing from more than 50 diverse countries. Within CMS, a global congregation of scientific minds converges, spanning the breadth of the planet, with the shared objective of advancing our comprehension of the universe and facilitating the transfer of cutting-edge technologies.

Yet, in this dynamic environment where people at CERN are always coming and going, and handovers for projects and contacts can be notoriously tricky, managing these intricate connections has become increasingly complex. The constant flux of individuals, ideas, and responsibilities demands an innovative and comprehensive approach to ensure the seamless management of this intricate tapestry of interactions.

With the continuous expansion of the collaboration within both CMS and CERN, the task of conducting outreach initiatives undergoes a gradual transformation, becoming progressively complex. This complexity arises primarily from the necessity to adeptly navigate and orchestrate a sprawling and perpetually shifting labyrinth of contacts and information. This elaborate web consists of a multitude of contact cards, each representing an essential link to diverse individuals and entities, meticulously organized into various distinct categories that delineate the multifaceted connections within this expansive network. The sheer volume of these connections and the diversity of their categorization contribute to the challenge, demanding an innovative and comprehensive approach to ensure the seamless management of this intricate tapestry of interactions.

Project description

The CMS Contact and Project Management Application is a specialized software solution aimed at optimizing contact management within the global Compact Muon Solenoid (CMS) experiment at CERN. With its extensive network of scientists, researchers, and students spanning the globe, the application provides customized functionalities for effective contact organization. It enables users to create, categorize, and track contacts, facilitating group and subgroup arrangements. The tool also enables sharing contacts among CERN members and e-groups while allowing users to set reminders for each contact. By employing an internal secure database, the application ensures data privacy and aids in handling the intricate task of managing a substantial volume of contacts, making it an asset for outreach activities, and facilitating smooth contact handovers.

Technology Stack

Our mobile application is developed using React Native in conjunction with Flask for its backend, exemplifies a dynamic blend of cutting-edge technologies, resulting in a responsive and feature-rich user experience. This symbiotic architecture is founded upon a client-server model, with React Native serving as the frontend framework and Flask facilitating the backend functionality, all orchestrated through a REST API (Representational State Transfer Application Programming Interface).

1. Frontend Architecture:



Figure 1: React Native logo

The technology landscape we are employing encompasses two pivotal components: React Native and Expo. React Native is a pioneering cross-platform framework, harnessing the power of JavaScript to sculpt genuinely native applications for both iOS and Android platforms, all consolidated within a singular codebase. This architecture showcases the synergy between code efficiency and native performance.

Complementing React Native's prowess, Expo stands as an innovative framework that extends its capabilities. Serving as a catalyst for streamlined app development and launches, Expo provides an array of tools and services meticulously tailored for React Native projects. Its significance lies in simplifying project launches, facilitating in-production bug rectification, bolstering team collaboration, and enabling client engagement. By fostering this symbiotic relationship, we leverage React Native's innate potential and amplify it with Expo's streamlined development paradigms, allowing us to create and launch applications with heightened efficiency and collaboration.

Diving into the realm of styling and theming, our approach is significantly enriched by the integration of React Native Paper, an impactful library that elevates the aesthetics and user interface (UI) dynamics of our applications. React Native Paper emerges as a powerful toolkit, meticulously crafted to provide a seamless and visually appealing user experience.

This library introduces a structured methodology for creating consistent and engaging UI elements across our applications. With React Native Paper, we gain access to a spectrum of pre-designed components, each thoughtfully tailored to enhance the visual appeal while maintaining a harmonious user interface. This eradicates the need to construct UI components from scratch, saving valuable development time and resources.

In essence, the integration of React Native Paper empowers our development endeavors with a library that not only accelerates UI component creation but also imbues our applications with a visual allure that resonates with our users. By harnessing its features, we can ensure a delightful user experience while maintaining a consistent and elegant design throughout our applications.

2. Backend Architecture (Flask):



Figure 2 : Flask logo

Server Logic: Flask, a micro web framework for Python, forms the foundation of the backend. It handles incoming HTTP requests, processes them, and returns appropriate responses.

REST API: Flask's RESTful capabilities are harnessed to design a comprehensive set of API endpoints. These endpoints are designed to adhere to REST principles, enabling the frontend to communicate effectively with the backend.

Database Integration: Flask interacts with multiple type of databases, MySQL in our case. The data models are defined using an Object-Relational Mapping (ORM) library like SQLAlchemy.

3. Database:



Figure 3 : MySQL logo

MySQL, a robust relational database management system, will serve as our data repository. The designated repository will reside within the highly secure enclave known as the Database on Demand Service (DBOD) at CERN. Furthermore, it's pivotal to underline that the database intricately intertwines with the CERN internal private CMS database. This amalgamation of MySQL and the internal private CMS database underscores the meticulous attention to data management, security, and integration, ensuring a cohesive and robust environment for housing and accessing critical information.

4. Integration and Communication:

REST API: The communication between the frontend and the backend occurs through REST API endpoints. The API endpoints are responsible for receiving requests, processing them, and sending back appropriate responses in a structured format, often in JSON.

Data Flow: The React Native frontend sends HTTP requests to specific endpoints on the Flask backend using libraries which is fetch in our case. The backend processes these requests, retrieves, or stores data from/to the database, and returns the response to the frontend.

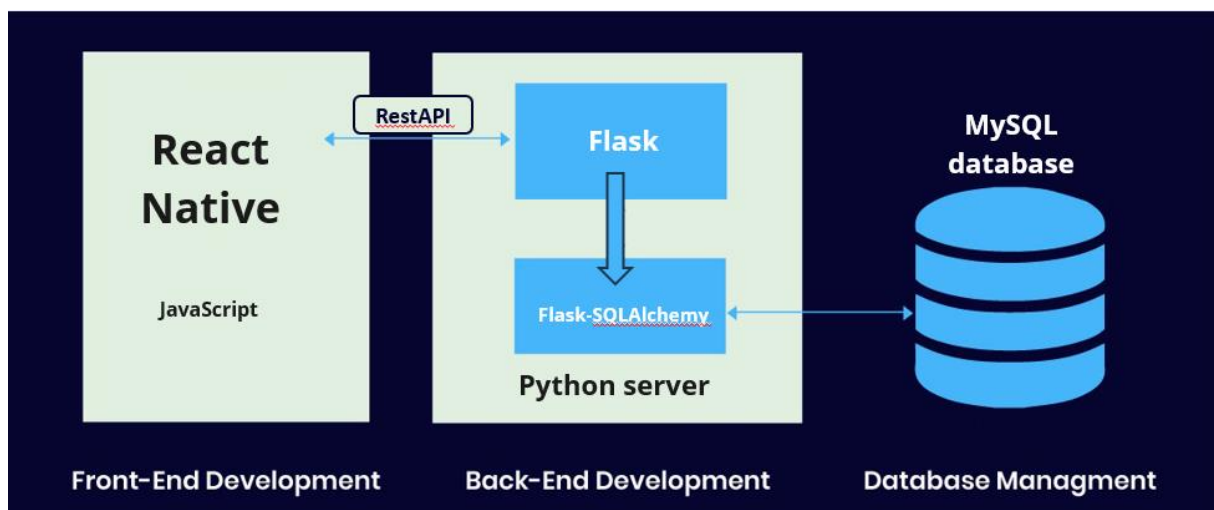


Figure 4 : Architecture

Functional Requirements:

The Functional Requirements encapsulate the pivotal features that are imperative for integration within the application. These requirements delineate the specific functionalities that the application must possess to comprehensively address the needs and expectations of its users. These features serve as the building blocks that collectively define the application's capabilities and purpose.

Features of the application:

- Create, read, update, delete contact groups.
- Create, read, update, delete contacts.
- Create, read, update, delete reminder of each contact.
- Sort and filter contacts by name, organization, and country
- Search contact groups by name
- Share contacts
- Share contact groups with CERN members and e-groups
- Organize contacts within groups and subgroups

Non-Functional Requirements

1. Security and Privacy:

Our solution must uphold the confidentiality of personal data belonging to users of the platform. Similarly, it is essential to ensure the security of authentication and the online payment procedure, which remains one of the most significant constraints in the web environment.

2. Maintainability and Updates:

The source code should be readable and comprehensible while adhering to established standards and accompanied by documentation. The application must be simple, extensible, and readily interactable with other functions. It should adhere to a standardized and clear architecture, enabling cost-effective maintenance and ease of reuse.

3. User Interface:

- Simple and User-Friendly:

The purpose of the application is to simplify course management within an e-learning platform for both learners and instructors.

It should cater to the user, ensuring effortless navigation between various pages, utilizing intuitive color schemes, and employing clear text formatting. The interfaces, therefore, need to be uncomplicated, user-friendly, and easily comprehensible.

- Consistent and Coherent:

Design choices for the interface should remain consistent across similar contexts and distinct for different contexts. This consistency aims to establish a predictable system, enhance manageability of learning, and minimize error rates. The application must adhere to a graphical charter to ensure visual elements within the interface maintain coherence.

- Reliable and Efficient:

The application must address business challenges, operate without errors, and satisfy the stakeholders of the system.

To fulfill these criteria, we have designed our interfaces using mockups before embarking on their actual development.

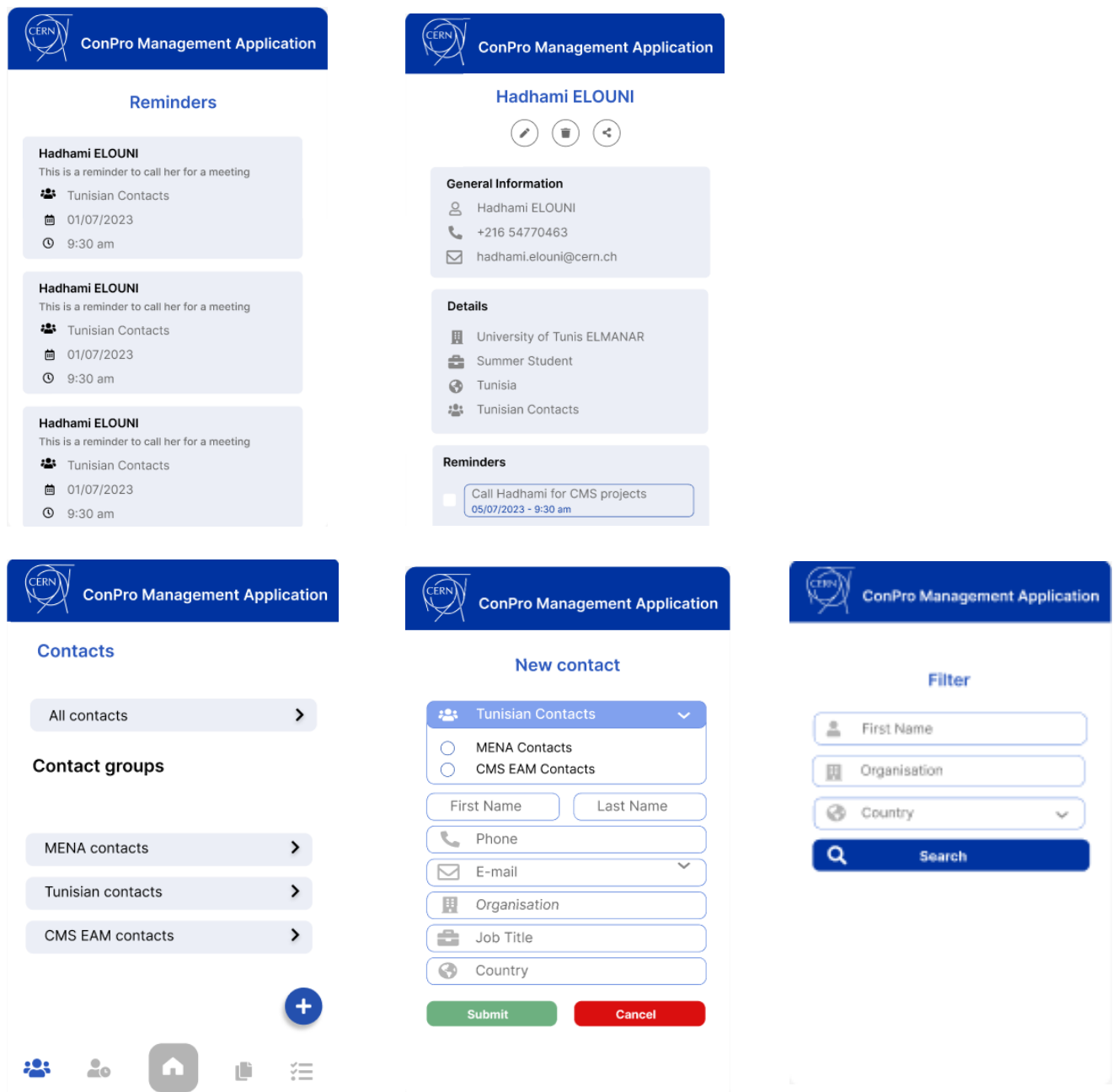


Figure 5 : prototype of the application

Database design:

1. Class diagram:

The class diagram functions as a static representation that portrays the unchanging perspective of an application. Its purpose extends beyond just visualization, description, and documentation of various elements within a system.

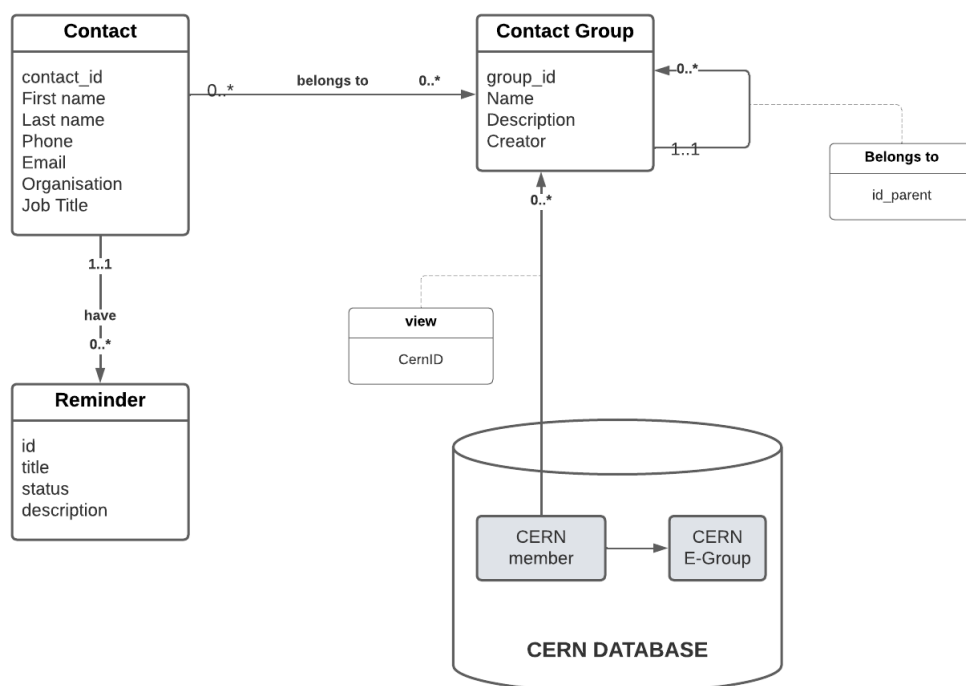


Figure 6 : Class Diagram

2. Relational schema:

Contact (contact_id, name, lastName, email, phone, jobTitle, organisation, country)

Groupcontact (group_id, name, description, id_parent#)

Groupcontactgroup(group_id#, contact_id#)

Reminder (id, title, status, description, date, contact_id#)

groupView (CernID#, group_id#)

Development Cycle

- **Human Centered Design:**

Human-centered design (HCD) is a problem-solving technique that puts real people at the center of the development process, enabling you to create products and services that resonate and are tailored to your audience's needs.

By incorporating HCD principles into the software development process, we can assure we will create an application that will tailor to the needs of different stakeholders at CERN.

The development cycle will be heavily inspired by HCD methodology, you can use the below steps for inspiration on how to move forward with development.

- **Persona Development:**

Create personas that represent different the different stakeholders, such as supervisors, newcomers, etc.

- **User Research**

Using the personas developed, conducts interviews with the different stakeholders.

Define the goals, motivations, and struggles and gather feedback on existing tools and systems used at CERN.

Use the information gathered to guide the design process and refine/add features and functionalities to the application.

- **Prototyping and Usability Testing:**

Create prototypes for the user interface using Figma to iteratively test and refine the design based on feedback.

You can use Figma to generate interactable wireframes that can then be used to test on a phone.

Once a “final” design is reached move forward to writing code for the frontend and backend to start conducting usability tests.

Realization

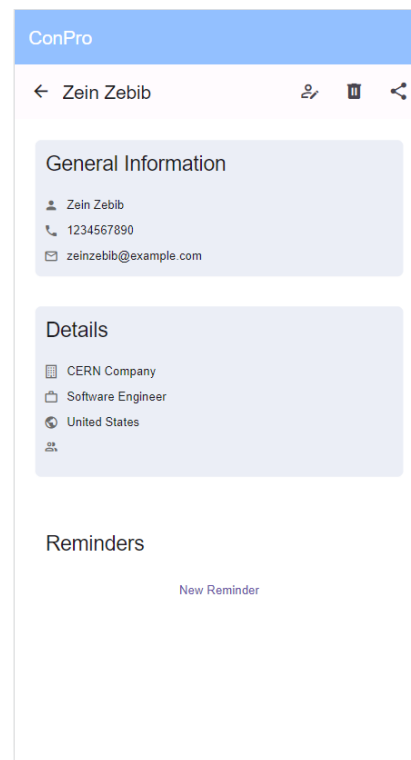
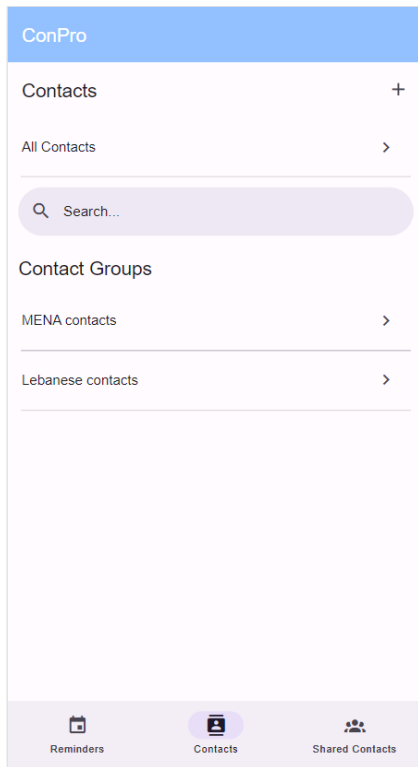


Figure 7: Single contact page

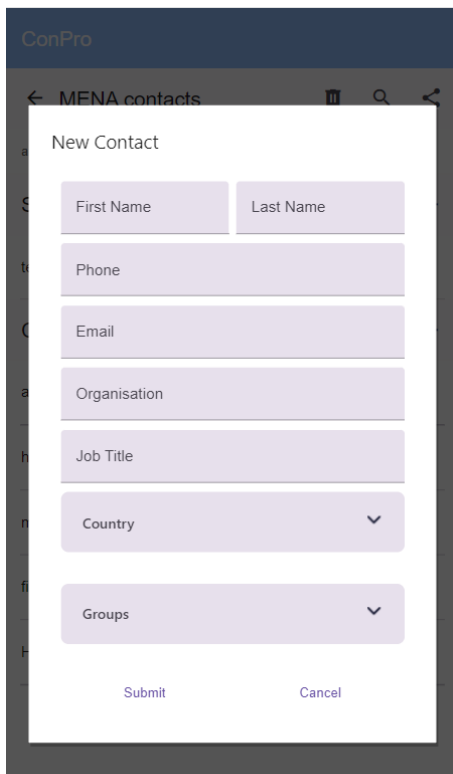


Figure 8 : New contact page

Figure 9 : Home

page

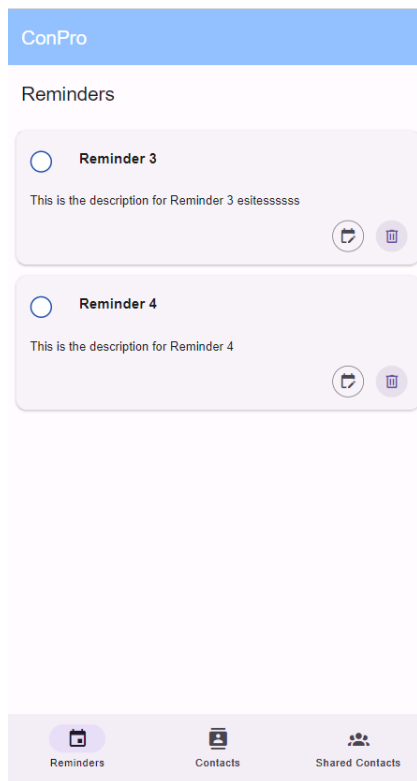


Figure 12 : reminders page

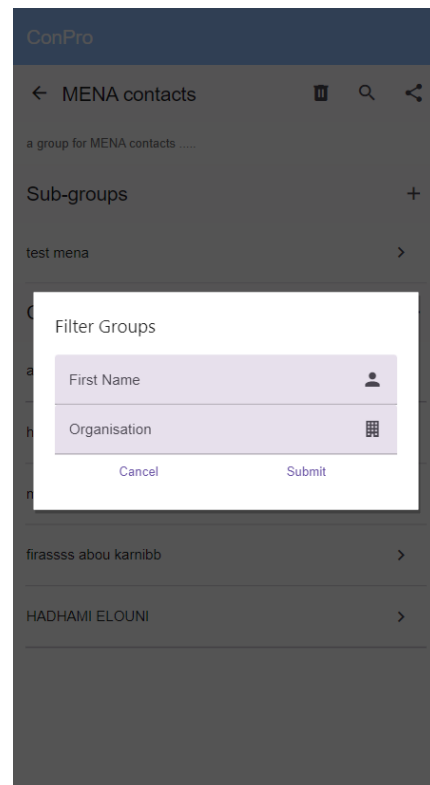


Figure 13: filter groups page

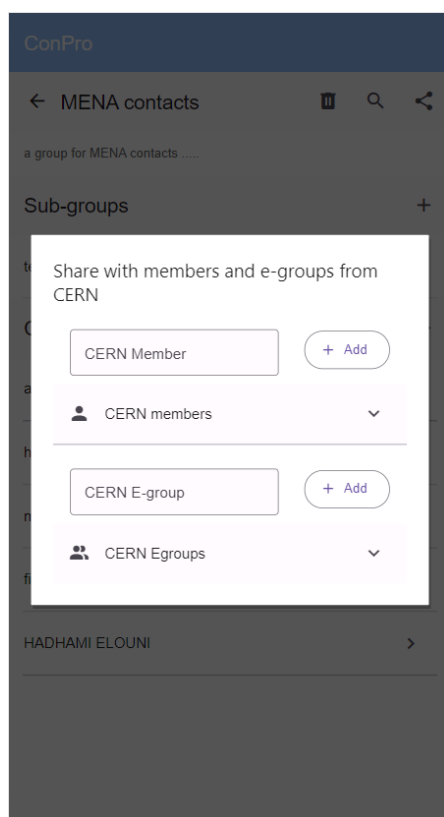


Figure 11 : Share group page

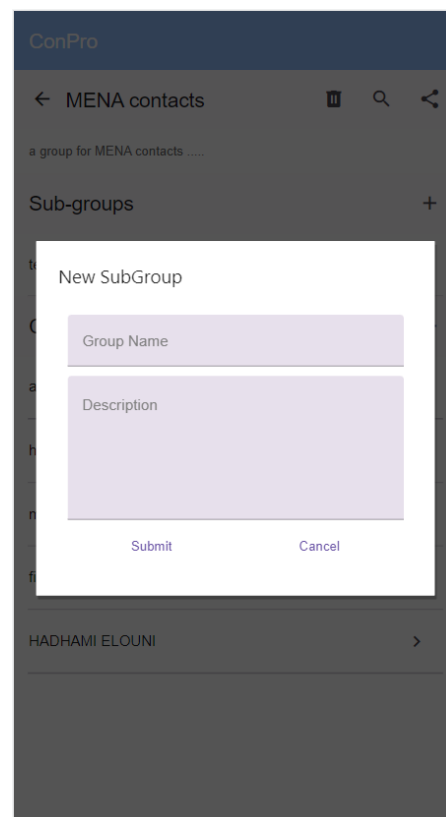


Figure 10 : New sub-group page

Conclusion

In conclusion, the CMS contact and project Management Application stands as a remarkable testament to the power of collaboration, innovation, and technological advancement. This application embodies the collective efforts of scientists, engineers, and developers from diverse corners of the world.

The application's role in optimizing contact management, streamlining interactions, and enhancing the sharing of cutting-edge knowledge among the global scientific community is undeniably significant.

The application's technology stack combines modern frameworks, like React Native and Flask, through a REST API, showcasing the harmony of frontend and backend technologies. Its functional requirements, covering contact management, reminders, sorting, filtering, and sharing, are pivotal for its success. Additionally, the focus on non-functional aspects such as security, maintainability, and a user-friendly interface strengthens the application's commitment to providing a secure, reliable, and intuitive platform. These requirements not only ensure operational excellence but also enhance user trust, long-term viability, and engagement, making the application a powerful and user-centered solution.

In essence, the CMS Project Management Application is more than just a technological achievement; it symbolizes the power of collective intellect, global collaboration, and a relentless pursuit of knowledge. It bridges geographical boundaries, facilitates seamless information exchange, and enables scientists to focus on their mission of advancing our understanding of the universe. As the CMS experiment and CERN continue to push the boundaries of scientific exploration, this application will remain an essential tool, embodying the spirit of discovery and innovation that defines humanity's quest for knowledge.