# DomoticApp



## Project Idea

**DomoticApp** is a platform that offer users the convenience of managing various aspects of their home environment remotely or automatically.

## Functional Requirements

The users of the application will be divided into two categories: **unregistered users** and **registered users**.

Registered users are allowed to use the main functionalities of the application.

Access to the application is guaranteed only to registered users. It's provided a login system using username and password, through which the user will be correctly identified.

A registration form will allow new users to register within the application as registered users.

### User Management:

- Allow users to register, log in, and manage their accounts.

### Device Integration:

- Support for adding and configuring various smart home devices.
- Ability to discover and connect to devices using different protocols.
- Provide a dashboard to view and control connected devices.

### Automation:

- Enable users to create automation rules based on triggers.
- Support for scheduling automated actions and scenes.

**Remote Access:**

- Allow users to control devices remotely from their smartphones.

**Notifications:**

- Send push notifications for events such as motion detection, door/window openings, or security alerts.

**Voice Control:**

- Integration for voice control of devices.
- Support for voice commands to execute actions and scenes.

# Non-Functional Requirements:

## Reliability:

- Ensure high availability and uptime for the app and connected devices.
- Implement error handling and recovery mechanisms to handle failures gracefully.

## Performance:

- Optimize app performance for fast response times and minimal latency.
- Efficient handling of device communication and automation execution.

## Usability:

- Design an intuitive and user-friendly interface with easy navigation and clear feedback.

## Compatibility:

- Ensure compatibility with a wide range of Android devices.
- Test the app on different screen sizes and resolutions to ensure responsiveness and consistency.

## Scalability:

- Design scalable architecture to accommodate growth in user base and device connections.
- Performance monitoring and optimization to handle increased load and demand.