Mobeye’s Mobile Application

Software Design Specification

By S3_CB05_04

Table of Contents

1. INTRODUCTION 3

1.1. DOCUMENT DESCRIPTION 3

1.1.1. Introduction 3

1.1.2. System Overview 3

2. DESIGN CONSIDERATIONS 4

2.1. ASSUMPTIONS AND DEPENDENCIES 4

2.2. GENERAL CONSTRAINTS 4

2.3. GOALS AND GUIDELINES 4

3. ARCHITECTURAL STRATEGIES 5

3.1. FRONT-END 5

3.2. BACK-END 5

4. SYSTEM ARCHITECTURE 6

5. DETAILED SYSTEM DESIGN 7

5.1. GUI 7

6. GLOSSARY 8

7. BIBLIOGRAPHY 8

1. Introduction

This project is created for the Dutch company Mobeye which specialize in innovative alarm and telemetry technology. The company was founded in 2008 with the idea to help people and organizations to make their life safer and easier by providing them with a meaningful way to secure, control and monitor their property and devices remotely.

Mobeye products have received multiple awards and are also used worldwide in various sectors, including police, the agricultural and medical industry, in building management and industry, by both professional users and consumers. Their in-house Research & Development team designs and develops customer-specific products on request as well.

The goal of this project is to design and develop a mobile application that would contribute to the Mobeye existing notification system.

1.1. Document Description

1.1.1. Introduction

The design document gives and overview of the system specifications of the mobile application that would be developed. It illustrates the software architecture of the application, explaining each of the components and their correlation. Furthermore, the document presents an initial design prototype of the GUI and comments on the specific design choices. The design document would also contain a research analysis on the technologies chosen for the implementation of the application.

1.1.2. System Overview

Due to increasing demand, Mobeye has decided to create a mobile application that their customers can use as another way to receive any important alarm system messages.

The mobile application would also offer a more convenient way for the customers to access some basic control functions.

By logging into their personal account, the users would be able to control and monitor their devices. They would receive a push notification in case there is an emergency and they would also have the possibility to view relevant data concerning the alarm that has been forwarded to their device. The users would also have the possibility to access some basic control functions as for example to arm or disarm a device. Furthermore, there would be the option to go to the Mobeye's web portal from where their customers could acquire full control. The mobile application that would be developed for the Dutch company Mobeye would not only elevate their performance and answer their customers' needs, but also would offer a new modern way for their client to overview and control their devices.

2. Design Considerations

2.1. Assumptions and Dependencies

Some of the dependencies that could occur when designing the application could be the possibility of changes in functionalities. It is probable that during the development phase a feature could not be implemented or it does not make sense to be in the application, then the user requirements that have been set at the beginning would have to change, thus the design of the application could change as well. Another dependency is the operating system the users have on their devices. To mitigate this dependency, assuming that users use different OS, the application would be cross-platform, meaning that it could be used on various operating systems. Furthermore, for the use of the application, the users would have to have access to the Internet.

2.2. General Constraints

One of the main constraints that the project could face is the deviation from the original project plan. Since it is an individual project and due to the agile approach, at some point in time the developers could change the original plan of the application and either decide to add some functionality that is not needed or it could complicate the already existing architecture of the project.

The connection between the frontend and backend of the application could be a major constraint that would have to be resolved. Since it is vital for the application to be secure and follow the best software development practices, special attention would be given to securing the application and delivering a quality product for the end-users.

A universal constraint would be the time. Additional functionality and deviations from the original plan could result in need of more workhours for the project. However, with a strict and thorough planning, this constraint might not be an issue.

2.3. Goals and Guidelines

The main goal of the project is to design and develop a user-friendly mobile application that would be easy to use and would realize the needs of the end-users. The application should be implemented following the best software development practices and UX principles.

3. Architectural Strategies

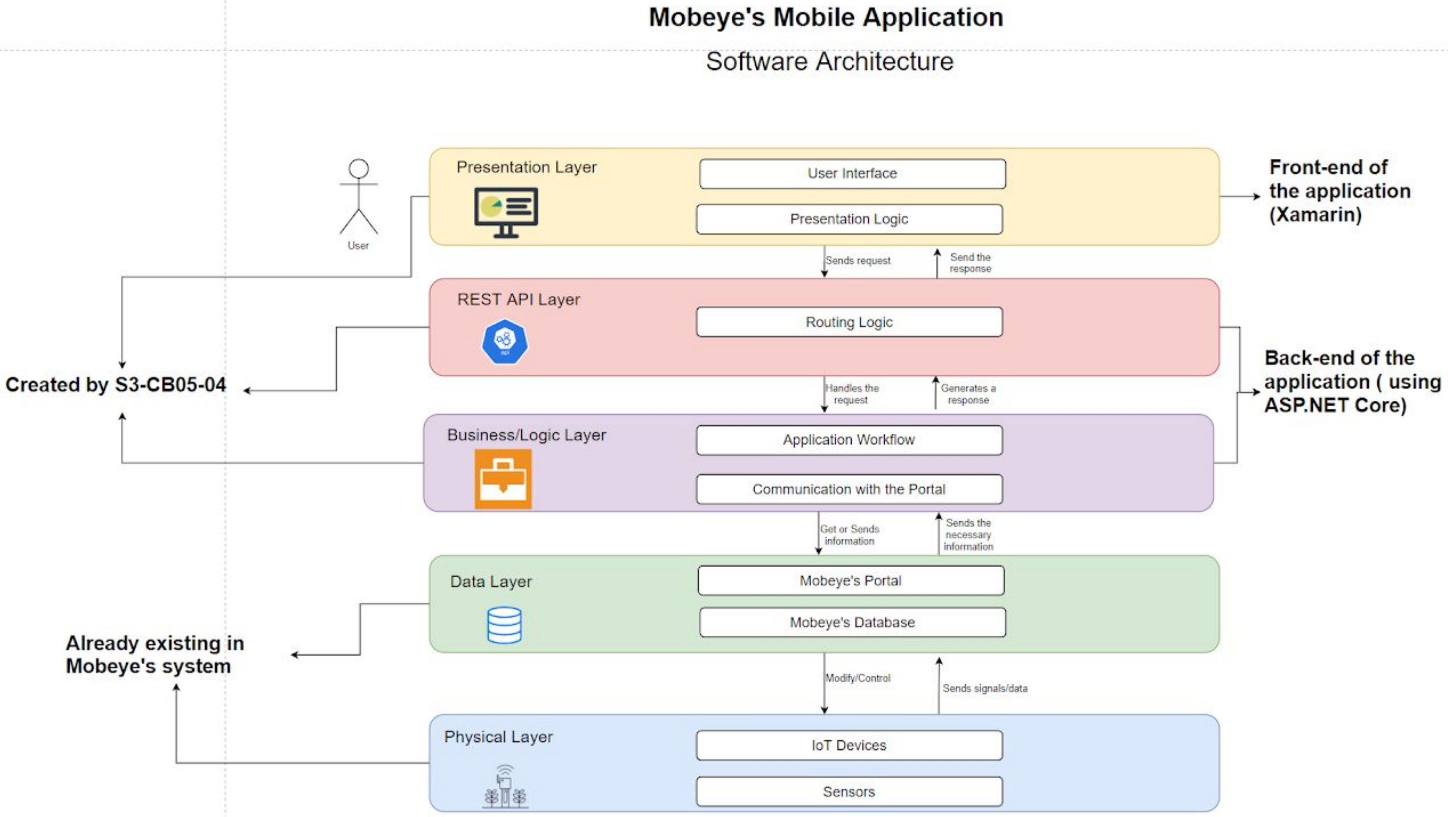
3.1. Front-end

For the creation of the front-end the project group is going to use Xamarin.Forms, which uses xml for the design and development of the GUI components. Xamarin is a tool used for cross-platform mobile app development that allows engineers to share about 90 percent of code across major platforms. It means that the application could be created to be used by both iOS and Android users. Xamarin uses a single language, C#, to create apps for all mobile platforms.

3.2. Back-end

For the backend the team is going to create an API using ASP.NET Core. The reason for choosing ASP.NET Core as a backend framework for the mobile application is that ASP.Net Core is one of the best performing frameworks (compared with for example Java Scarlet or Node.js). Furthermore, the technology stack used by Mobeye's engineers further influenced the decision of choosing ASP.Net Core, so that if needed, the developers could further improve the mobile application or handle the support for it.

4. System Architecture



The diagram above illustrates the software architecture of Mobeye's mobile application. Even though, the physical and data layer would not be created and/or modified by the project developers, they are still included in the diagram, so that an overview of the whole system could be created.

The developers would be concerned with the creation of the business/logic layer, the API layer that would handle the communication with the data layer (previously developed by Mobeye) and the presentation layer, that consists of the GUI.

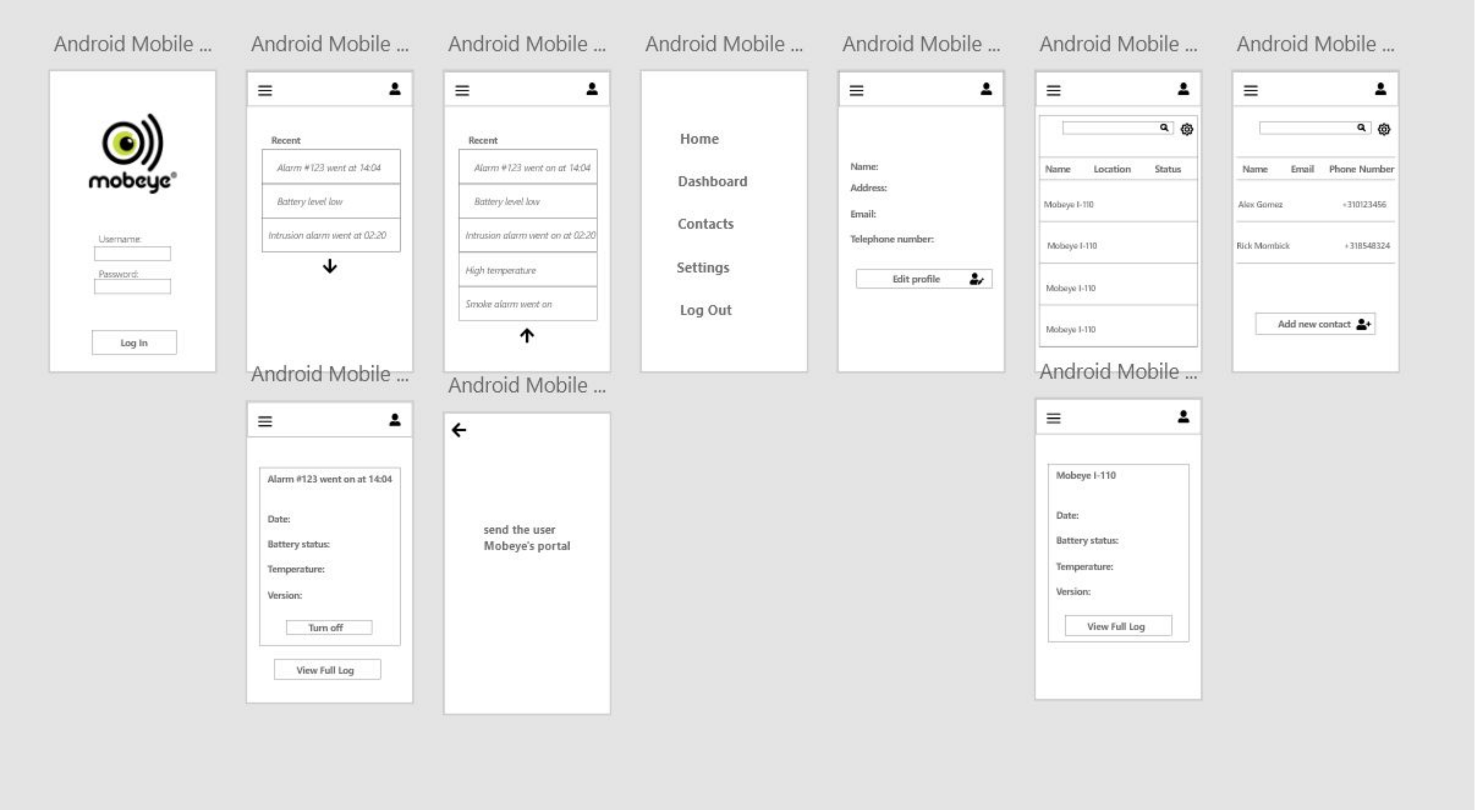
As mentioned in Section 3.1 the presentation layer of the application (the front-end) would be created using Xamarin. Mobeye's clients would interact with the user-friendly UI and would be able to monitor and access their IoT devices at any time. The backend, that would consist of the API layer, that handles the communication with the portal, and the business/logic layer.

5. Detailed System Design

5.1. GUI

An initial design prototype could be demonstrated here:

https://youtu.be/Xckc4Km_xw8



6. Glossary

- GUI – Graphical User Interface
- UX – User Experience
- API – Application Programming Interface

7. Bibliography

A list of referenced and/or related publications:

https://www.altexsoft.com/blog/mobile/pros-and-cons-of-xamarin-vs-native/
https://dotnet.microsoft.com/learn/aspnet/what-is-aspnet-core