Technical Report - **Product specification**

SensorSafe

Course:	IES - Introdução à Engenharia de Software
Date:	Aveiro, <date of submission>
Students:	108317: Miguel Aido Miragaia 107572: Gonçalo Rafael Correia Moreira Lopes 108536: Cristiano Antunes Nicolau
Project abstract:	Interactive system to manage Humidity/Temperature/Smoke Sensors

Table of contents:

1 Introduction

The creation of a system for tracking temperature, humidity, and smoke seems to be an ideal project. It's valuable in logistics, healthcare, security, and manufacturing. We will be able to implement some of the learned concepts in the IES course, and learn about new technologies, the different types of Architectures. We will also have the opportunity to see the behind the scenes of building a project with the the collaboration of a team where every member as his own role, the efficiency of working with GitHub to control the workflows, among using other for other functionalities.

Team Roles

Team Manager: Miguel Miragaia;

Architect: Gonçalo lopes;

Product Owner: Cristiano Nicolau;

DevOps Master: Miguel/ Gonçalo/ Cristiano;

2 Product concept

Vision statement

SensorSafe is a system with the objective of managing all the Sensors in a building. This system was design to help different types of users, from house owners to the Building Manager, having control of the temperatures/ humidity of his own divisions/apartments or even for a security context by having control of a smoke sensor to prevent a fire.

This system offers a full control of every type of sensors, allows users to have an app to manage them all, to prevent the need of the users to own multiple apps to control each one.

Personas and Scenarios

Personas:

- **Name:** John Smith
- **Age:** 47
- **Gender:** Male
- **Role:** Building Manager
- **Background:** John has several years of experience managing residential and commercial buildings. He is responsible for ensuring the safety and comfort of all residents and occupants.

- **Name:** Peter Williams
- **Age:** 37
- **Gender:** Male
- **Role:** Resident on the third floor of the building
- **Background:** Peter is a resident on the third floor of the building. He is deeply concerned about the safety and comfort of his family. Peter is interested in installing sensors in his apartment to enhance fire safety, monitor humidity in the bathrooms, and control the temperature to create a comfortable living environment for his family. He is actively seeking technological solutions to achieve these objectives within his apartment.

Product requirements (User stories)

User Story 1: As the Building Manager, John requires the ability to configure customized alerts for specific divisions within the building. For example, he should receive immediate notifications if the temperature in the boiler room exceeds a predefined threshold. This feature empowers John to proactively prevent potential equipment damage.

System: Using SafeSensor John can define specific temperature and humidity thresholds for each division.

User Story 2: In his role as Building Manager, John seeks access to historical data regarding smoke sensor events. This access allows him to analyze trends and identify potential areas of concern within the building.

System: The data is presented in a clear and visual format, helping John in identifying trends or patterns.

User Story 3: John intends to automate the generation of reports on sensor status, events, and maintenance schedules for management purposes."

System: There are some types of reports (sensor status, maintenance schedule). Reports are automatically scheduled.

User Story 4: As a resident on the third floor, Peter wants to install smoke detectors in multiple rooms of his apartment to ensure the safety of his family. He would like to receive immediate notifications on his smartphone in case of smoke detection in any area of his apartment.

System: The system allows for the installation of smoke detectors in specific rooms with real-time notifications to ensure Peter and his family's safety.

User Story 5: Furthermore, Peter wishes to install temperature sensors in various areas of his apartment to ensure the thermal comfort of his family. He wants to be able to monitor temperatures in real-time.

System: The system allows the installation of temperature sensors in specific areas of the apartment, providing real-time readings.

3 Architecture notebook

Key requirements and constrains

<Identify issues that will drive the choices for the architecture such as: Will the system be driven by complex deployment concerns, adapting to legacy systems, or performance issues? Does it need to be robust for long-term maintenance?

Identify critical issues that must be addressed by the architecture, such as: Are there hardware dependencies that should be isolated from the rest of the system? Does the system need to function efficiently under unusual conditions? Are there integrations with external systems? Is the system to be offered in different user-interfacing platforms (web, mobile devices, big screens,...)?

E.g.: (the references cited in [XX] would be hypothetical links to previous specification documents/deliverables)

There are some key requirements and system constraints that have a significant bearing on the architecture. They are:

- The existing legacy Course Catalog System at Wylie College must be accessed to retrieve all course information for the current semester. The C-Registration System must support the data formats and DBMS of the legacy Course Catalog System [E2].
- The existing legacy Billing System at Wylie College must be interfaced with to support billing of students. This interface is defined in the Course Billing Interface Specification [E1].
- All student, professor, and Registrar functionality must be available from both local campus PCs and remote PCs with internet dial up connections.
- The C-Registration System must ensure complete protection of data from unauthorized access. All remote accesses are subject to user identification and password control.
- The C-Registration System will be implemented as a client-server system. The client portion resides on PCs and the server portion must operate on the Wylie College UNIX Server. [E2]
- All performance and loading requirements, as stipulated in the Vision Document [E2] and the Supplementary Specification [15], must be taken into consideration as the architecture is being developed.>

Architetural view

→ Discuss architecture planned for the software solution.

→ include a diagram

Module interactions

→ explain how the identified modules will interact. Use sequence diagrams to clarify the interactions along time, when needed

→ dicuss more advanced app design issues: integration with Internet-based external services, data synchronization strategy, distributed workflows, push notifications mechanism, distribution of updates to distributed devices, etc.>

4 Information perspetive

<which concepts will be managed in this domain? How are they related?>

<use a logical model (UML classes) to explain the concepts of the domain and their attributes>

5 References and resources

<document the key components (e.g.: libraries, web services) or key references (e.g.: blog post) used that were really helpful and certainly would help other students pursuing a similar work>