Technical Report - **Product specification**

SmartHomes#

|  |  |
| --- | --- |
| Course: | IES - Introdução à Engenharia de Software |
| Date: | Aveiro, 19/10/2023 |
| Students: | 68264: Bruno Lopes  108712: Diogo Falcão  108011: Fábio Matias  107927: Rúben Garrido |
| Project abstract: | An app for monitoring the client's home resources, such as electricity, water (and possibly gas - ? ). |

Table of contents:

[1 Introduction](#_35sslkb8z3)

[2 Product concept](#_tfjhd25drpey)

[Vision statement](#_gc3ga3y7km9t)

[Personas](#_2geau0v6yvx1)

[Main scenarios](#_5iyqdln4iw44)

[3 Architecture notebook](#_ht05adz0s1wb)

[Key requirements and constrains](#_hwblkhip049d)

[Architetural view](#_flrr17v8574m)

[Module interactions](#_32p2sma1plje)

[4 Information perspetive](#_pcyv88xodbd9)

[5 References and resources](#_yc3oy6awb07y)

# 1 Introduction

In this first iteration of the IES final project, we’ll firstly present our product, then the architecture used and finally the core user cases and usage scenarios. With this deliverable, we will also explain the organization of the software project, build the software as a team and use corporate solutions and tools related to the already presented SmartHomes theme.

# 2 Product concept

## Vision statement

The SmartHomes application is design to address the problem of efficient and sustainable home resources management. Therefor, it provides a solution for homeowners to monitor and control their electricity and water.(for now).

<functional (black-box) description of the application: what will you system be used for? Which is the **high-level/business problem being solved by your system**?>

<if needed, clarify what was planned/expected to be included but was changed to a different approach/concept >

<optional: how is your system different or similar to other well-known products?>

<optional: you may include a UML Use Case diagram to support the explanation>

<optional: additional details on the process for the requirements gathering and selection (how did we developed the concept? Who helped us with the requirements? etc)>

## Personas and Scenarios

<Uma Persona é uma personagem utilizada para contar histórias representativas da futura utilização do sistema. Uma Persona é um Actor instanciado, à qual se dá um conjunto de caraterísticas para a humanizar e definir o contexto em que usará o sistema e as suas motivações.

“Personas are fictional people. They have names, likenesses, clothes, occupations, families, friends, pets, possessions, and so forth. They have age, gender, ethnicity, educational achievement, and socioeconomic status. They have life stories, goals and tasks. Scenarios can be constructed around personas, but the personas come first. They are not ‘agents’ or ‘actors’ in a script, they are people. Photographs of the personas and their workplaces are created and displayed. […] It is to obtain a more powerful level of identification and engagement that enable design, development, and testing to move forward more effectively”. Adapted from Grudin, J. and Pruitt, J., 2002, June. Personas, participatory design and product development: An infrastructure for engagement. In Proc. PDC (Vol. 2).

Exemplo: ver [secção 4.1 + 4.2 neste artigo](http://www.mdpi.com/1424-8220/18/4/1285) (open access)] >

## Product requirements (User stories)

<Keep in mind **main scenarios, related to the core value** of the system, and the methods for [agile project management](https://www.atlassian.com/agile/project-management)>

<Define the epics to be covered in the project.>

<present the user stories identified for the selected epics.>

Core stories:

→ Register several houses and their electrical system in one app.

→ Control every bit of water used in a drought in its region.

→ Oversee electronic devices connected to the house grid.

→ Administer the maximum percentage of the grid energy that can supply the house.

→ Turn off electronic devices with just a few taps trough the app.

→

# 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>