universidade de aveiro deti departamento de eletrónica, telecomunicações e informática

CoworkClub

5 Jun 2022

*Final project for the subject of **System Analysis**, class **P6**.*

Team

Daniel Capitão	75943
David Araújo	93444
Filipe Costa	77548
Samuel Teixeira	103325

1. Introduction

1.1 Executive Summary

In this report we focus on the results of the *Elaboration* and *Development* stages. This is divided into two sections: Architecture validation and Functionality implementation.

When validating the architecture we focused on picking the most important components for a user to accomplish a satisfactory simple interaction like booking a space. Since the system will be constructed with Vue JS, component reusability is also very efficient which leads to easy recognizance by the user. It also contributes to a very high level of compatibility across devices. This is important as **CoworkClub** is expected to be accessible globally and is not dependent on end-user device specificities.

Implementing further functionalities will consist in implementing capabilities and tools that increase the simplicity and speed of processes, on the user side; and increase robustness and reliability on the back office side of the processes, this includes the creation of shortcuts, more control over data by the user and increased quality in the visual aspect in order to improve appeal and UX.

1.2 Version Control

Date	Editor	Changes
7 Jun 2022	David Araújo	Introduction
7 Jun 2022	Samuel Teixeira	System Architecture and Use Cases
9 Jun 2022	Daniel Capitão	System Architecture and Use Cases
10 Jun 2022	David Araújo	Software architecture diagram Physical architecture diagram Strategy and Estate of implementation
19 Jun 2022	Samuel Teixeira	Use cases of the Second Increment
23 Jun 2022	Daniel Capitão	Update Decisions and Justifications Use cases of the Second Increment Implementation Stage
24 Jun 2022	David Araújo	Implementation state Use cases

1.3 References & Supplementary Resources

- Refine the Architecture
- CoworkClub

2. System Architecture

2.1 General Objectives

Our architecture will be done by prioritizing the user's characteristics. **CoworkClub** will have the following specifications:

- The System shall give quick notifications if something happens to a Booking. For example, if the Coworking space gets damaged to the point it's simply not usable, **CoworkClub** should warn the user about this change and provide help for repayment.
- Since **CoworkClub** is going to be a PWA, it will work in very similar ways in Desktop or Mobile, which makes the usage of the service very convenient.
- The system must be capable of making payments with various methods of online payment, such as MBWay, Paypal, etc.
- The System must not accept more than one booking for the same time frame. If that is the case, every Booking after the first one in these circumstances will not be allowed.
- Reviews of a certain Coworking space will only be allowed if the user has made at least one Booking of that specific space

2.2 Requirements with impact on the architecture

Requirements	Description
RD-1	User-friendly interface with the integrated payment platform.
RD-2	The system will be able to support up to 250 users and will have cross-platform support.
RS-1	Secure storage and payment platform.
RU-1	Results based on user location integrable with calendar applications.
RS-1	Login operations should rely on Firebase services and social login
RU-2	Re-use of components should be preferred in order for the user to recollect the various components' functions.
RU-3	The user can search without any input or with incomplete input, the system, in this case, must assume default values. This is done so a user that does not know what is his/her ideal space, can still browse through the available ones
RD-3	A minimalistic design, and resorting to industry-standard keywords and

Requirements	Description
	icons must be used to accomplish a more user-friendly environment.

2.3 Decisions and Justifications

- TeleportHQ is our choice for front-end websites and UI elements developments. It comes with a visual builder and allows us to create our own custom functionalities with integrated front-end development tools.
- VueJS (JavaScript framework) for building user interfaces and single-page applications is ideal for modularization and fast development.
- Inicialy ExpressJS, changed to Firebase, for easier implementation, authentication, automated build, database, faster integration and free hosting.
- Jira for project management purposes allows us to track user stories' stage of development.

2.4 Software Architecture

Our implementation will be subdivided into three main domains, these being: User Interfaces, Logical Domains, and Infrastructure. A package diagram can be found in the Annexe section.

In the UI section, we highlight the two main packages: website and *progressive web application*. Our solutions intend to implement both in order to increase compatibility across devices and encourage “persistence” with the user by enabling it to keep an application of our system locally.

At the logical domain level, we focus on what are the core domains of our platform. Users are of two types, workers and providers, which can overlap. Also we have spaces, and spaces can have their own attributes, but most importantly, a collection of offices, that can be of different types. It's these spaces that can be subject to booking.

2.5 Physical Architecture

For our physical infrastructure, we envision a simple implementation. Since the application is mostly rendered on the side of the user, there is no need for a highly capable server system.

For it to give location based results, we would need access to the devices gps capabilities, or if in a non mobile system, access to the WiFi information in order to estimate a location. On the

other end, the server will only need to run an API and a database, probably a JSON base one. A physical diagram can be found in the Annexe section.

3. First Increment

3.1 Use Cases on the First Increment

For this increment, we thought about the most important functionalities of our Project (also called Core Functionalities). Those are the ones that make our Service work the way it was meant to be. In order to achieve that goal, a user needs to able to Search for Coworking Spaces, look at their features and informations and do the Booking itself. On this increment we focused more about the Coworker's functionalities rather than the Space Provider's.

3.2 Selected Stories of Use

Story/Use Case Slice	Acceptance Criteria
Joaquim Searches for a Coworking Space Joaquim needs a Coworking Space for his online meetings, so he decides to use CoworkClub to search for it	**1º Scenario: Searches for an existing name** Joaquim opens the app and searches for a Coworking Space he knows. When Joaquim presses the search button, the System finds out that there is a Coworking Space with that name and gives that result to him **2º Scenario: Searches for a non-existing name** Joaquim opens the app and, by mistake, searches "askfdskz". As he presses the search button of mere curiosity, the System doesn't find any Coworking Spaces/Geographical regions, and gives no results to him **3º Scenario: Searches with filtering** Joaquim doesn't know a specific Coworking Space, but he knows what he wants in one. So, he uses the System's filters to filter by region and private room with free Wi-Fi and Coffee. After pressing

Story/Use Case Slice	Acceptance Criteria
	the search button, the System finds out 2 options for Joaquim and gives to him as a result.
Ana wants to make a booking for a Coworking Space After finding a Coworking Space that fits her, Ana wants to	Financial institution. Its goal at **CoworkClub** is to help the process of payment of Coworking Spaces by the Coworkers. The Bank can either accept or decline the transaction
Cátia wants to compare cowork spaces to find the best price offer based on her preferences Cátia uses **CoworkClub** to filter high prices coworking spaces	The platform has a search bar that can help the user find the best booking offer based on his preferences such as price, location, type of service, amenities, etc. The user chooses the filter parameters and hit search, the spaces are shown in relevance order.
José is a digital nomad and loves to travel, he needs a professional working space close to his current location José searched Coworking Space on **CoworkClub** based on his current location and availability	**CoworkClub** offers numerous Coworking Spaces based on location. José is in a new city he doesn't know about, so he trusts our platform to help him find a solution. He uses the search filter to find a professional working space based on his preferences, books for a week, and searches for coworking spaces in the next city he will stay.
Pedro puts his Coworking Space available at CoworkClub Pedro owns a potential Coworking Space, thus, he wants to get more clients and finds out a solution for that at **CoworkClub**	The System asks for the space's features, and, after that, as he presses the add button, the System makes Pedro's space available for searching/making bookings at **CoworkClub**.
Rita's Coworking space had a problem Rita has a coworking space at **CoworkClub** and that space had an	After reporting the situation to **CoworkClub**, the System will send a notification to every Coworker affected by this event. The Coworking space will be temporarily unavailable until the problems are solved. Also, **CoworkClub** will need to help the Coworkers by providing repayment to them.

Story/Use Case Slice	Acceptance Criteria
unpredicted incident that severely affected the space, making it impossible for Bookings to occur. She wants to inform the Coworkers.	
Office/Coworking Space	The space owned by the Space provider that was paid by the Coworker with the primary goal of using that space to work for a determined period of time
Calendar	An agenda with all the Bookings of an Office/Coworking Space of a Space Provider for all time frames

3.3 Strategy and Implementation Stage

As of this stage the implementation is still on the design of the user interface, using a Teleport HQ, the team is designing a interface so later it can be exported in Vue Js code. Home and search pages are already complete, and now follows the spaces' pages, profile and booking pages too.

The API in Express JS is already prepared and designed, requests and responses are already constructed, and are being implemented into the code.

What is lacking as of this stage is, besides the missing pages, the connection between front-end and API.

4. Second Increment

4.1 Use Cases on the Second Increment

In this Increment, we focused on the core functionalities that were not made on the first increment and the remaining functionalities. The first ones are Booking tracking and helping Space Providers transform their spaces according to the Coworkers needs. The second ones are everything that is common in applications, such as login, registering, and so on.

4.2 Selected Stories of Use

ID	Story/Use Case Slice	Acceptance Criteria
USC-1	**Gonçalo wants to register at the CoworkClub platform** Gonçalo that just discovered the advantages of being a **CoworkClub** user. Because of that, he wants to create an account.	After pressing the register button at the index page, the System will redirect the User to the registering page, where Gonçalo will insert its credentials such as his name, email, phone number, etc. After agreeing with **CoworkClub's** Terms and Conditions, Gonçalo will successfully have an account created and, after that, the "get started" will help him understand **CoworkClub**.
USC-2	**Patrícia wants to do login at CoworkClub** After buying a new smartphone, Patrícia downloads the **CoworkClub** app in order to enjoy the benefits of our services. But, for that, she has to log into her account.	**1º Scenario: Tries to log with the wrong credentials.** As Patrícia clicks on the login button, she will be redirected to the login page. As she writes her credentials, she misspelled her password by mistake and the System reacts accordingly by denying the access and saying what went wrong. **2º Scenario: Logs with the right credentials.** Patrícia clicks on the login button, gets redirected to the login page and inserts the right credentials. After that, the System will recognize Patrícia's account, giving her access to the account.

ID	Story/Use Case Slice	Acceptance Criteria
USC-3	**Joana register in the CoworkClub platform, now she want to update and modify her profile information**	**CoworkClub** offers each user a profile page where the personal information can be edited. Everything since login information to profiling preferences.
USC-4	**Maria needs to make sure that her Booking in a Coworking Space will be possible** As a CEO who couldn't be physically present at a very important reunion of the company she works at, Maria a coworking space is a proper place for such an event. But, due to the level of demand for her job, her Booking needs to happen.	**CoworkClub** offers tracking of Bookings that tells the user the current state of a certain Booking. If for some reason the Booking gets canceled due to internal problems, the System will not only send a notification to the user and help with the repayment(as mentioned earlier), but also will try to give the best options of Booking possible to the user for that time frame, according to the features of the canceled Booking.
USC-5	**Diogo wants to keep he's expenses to a minimum, soo he needs to search for a space that meets is price demands**. For this, he needs to be able to specify a price limit, maximum or minimum that narrow his search.	**CoworkClub** offers a search toolbar with results filters that allow any user to adapt the search results according to a set of specified requirements. For this, the filters must work in conjunction with each other and not be exclusive. Filters must work even with information that is not visible in spaces cards.
USC-6	**Mariana has found a place she likes, but she need to know what kind of office solutions they offer** in order to view the prices associated with each offering and cross compare them, selecting the one thar suits her needs.	**CoworkClub** enable space providers to specify a pricing table for user consultation. This table is divided into sections, per office type, and each one of this sections is subdivided by time frames of: per dium, per week, per month.
USC-7	**José has made is choice of space and office, and he confirm the specificities of his booking, but in the confirmation page he noticed he inputted the incorrect date**, he needs to go back and correct it before confirming	**CoworkClub** prompts the user to confirm all of the chosen options before proceeding with payment and registration on booking process. The platform displays a summary of all the booking details, payment options, final calculated price and a notice for the user to know that the booking will now follow in the process of awaiting for the space providers approval.

ID	Story/Use Case Slice	Acceptance Criteria
	the booking.	
USC-8	**André wants to know user preferences so that he can customize his Coworking Space** André owns a Coworking Space that is available at **CoworkClub** and he is not happy with the amount of Bookings that are made and wants to change that fact. It would be great if there was a way to know what to add in order to get more revenue.	**CoworkClub** will ask the users occasionally about what they like/prioritize in a Coworking space. According to the location of those users, public charts/information with those features will be sent to coworking space providers that are located close to the user. Besides that, when a Booking is canceled by the user, the system will kindly ask for motives for that cancellation (it is not mandatory) and those motives will be sent to the wpace provider. Finally, reviews after a booking are a direct way to interact with the users about how good/bad the experience was and what it was lacking.
USC-9	**Marco wants to know some metrics about his use of the CoworkClub** Marco is an enthusiastic user of coworking spaces, he wants to keep track of his booking historic and personal favorite spaces.	**CoworkClub** keeps track of your booking record and favorite coworking spaces associated with your account. Marco will be able to book his next cowork space based on his preferences without the worry of not remembering or taking into account favorite spaces booked before.

4.3 Quality Insurance

For testing, Cypress was used for fast development thanks to a graphical interface during the creation of tests. Attempts were made to integrate it with GitHub action in the already existing workflows for deployment with Firebase, but these were unsuccessful.

Non the less, testing was done in local machines before committing, ensuring some kind of testing control.

4.4 Implementation State

As it stands, **CoworkClub** platform implements, from the previous table:

- USC-1;
- USC-2;
- USC-3;
- USC-5;
- USC-6;
- USC-7

USC-1 & USC-2

With Firebase services, **CoworkClub** is able to provide users with te capability of registration with the platform. This is accomplished with Authentication Services for credential management and Sign up and Login operation, and Firestore for storage and profiling of users preferences.

USC-3

With some of the services mentioned before, mainly Firestore, we are able to create a users profile containing various information like:

- Job title;
- Last visited space;
- Favorite space;
- Favorite office type;

Not all of this preferences are implemented has of this stage (only Job title is), but is all relevant information for a later stage, which would be usefull to construct user metrics and behaviourall studies.

USC-5

The filter toolbar is displayed to the user when visiting the space seach result page, in order to filter the results given. Has of now, price, parking are functional with the rest of the options constructed but still not plugged to the filtration process. This is a functionality that can be easily expanded in one iteration on a later stage.

USC-6

Fully implemented; each space loads *on render* it's price table which is read and processed on the client side and the price table display adapted to the received information.

USC-7

Also fully implemented, upon selecting booking preferences and push "Book", the user is transported to a confirmation pages, with all the imputed information already processed.
If confirmed, only then the data is pushed into the permanent database, to be latter used for user and space provider consultation.

Not accomplished

For this stage, the team intended on also implementing booking states, relating to the stage in the booking process, the book was in. This would provide to the user with information letting him/her know with the book was either, awaiting approval, approved or revoked

5. Use Cases Specification

5.1 Package: Booking

5.1.1 Search for a space

Name:	USC-5: Search for a space
Created by:	David Araújo
Actor:	**Primary**: User
Brief Description:	A user that access **CoworkClub** whanta to find a space, and to do that can just select a set of spaces or can search for name or location and filter the search if spaces parameters.
Trigger:	A user is looking for a space with specific requirements.
Preconditions:	PRE-1: User filter information by providing parameters
Postconditions:	POST-1: Specify a name or location for the database to search for data. POST-2: Platform only makes visible information that meets the parameters.
Normal Flow:	1. User specifies a location or name in the search box. 2. User gets transported to a page with all the data retrieved by the platform that meets the searc query parameters. 3. The user can review this results 4. The user can then filter then by providing a price range or selecting with wants a spaces that offers parking space.

Priority:	High
Frequency of Use:	Approximately all of the users in the platform, expected 100 user daily in first month

5.1.2 Consultation of a price table

Name:	USC-6: Consult price table
Created by:	David Araújo
Actor:	**Primary**: User **Seconday**: Space provider
Brief Description:	A user that access **CoworkClub** and find a space that suits his/her needs, now requires to consult the available price options.
Trigger:	A user need to know the price of booking space and its correlation in price.
Preconditions:	PRE-1: Visit of a specific space's profile
Postconditions:	POST-1: The platform displays information in the views table according with the selected office type
Normal Flow:	1. The user searches for a space 2. Access the space's profile 3. Consults all the relevante information, like description, images, 4. **Consults the price table**, selecting different office type to view these options prices.
Priority:	High
Frequency of Use:	Will be highly volatil, depending on the specific space performance and relevance between the users.

5.1 Package: Profile

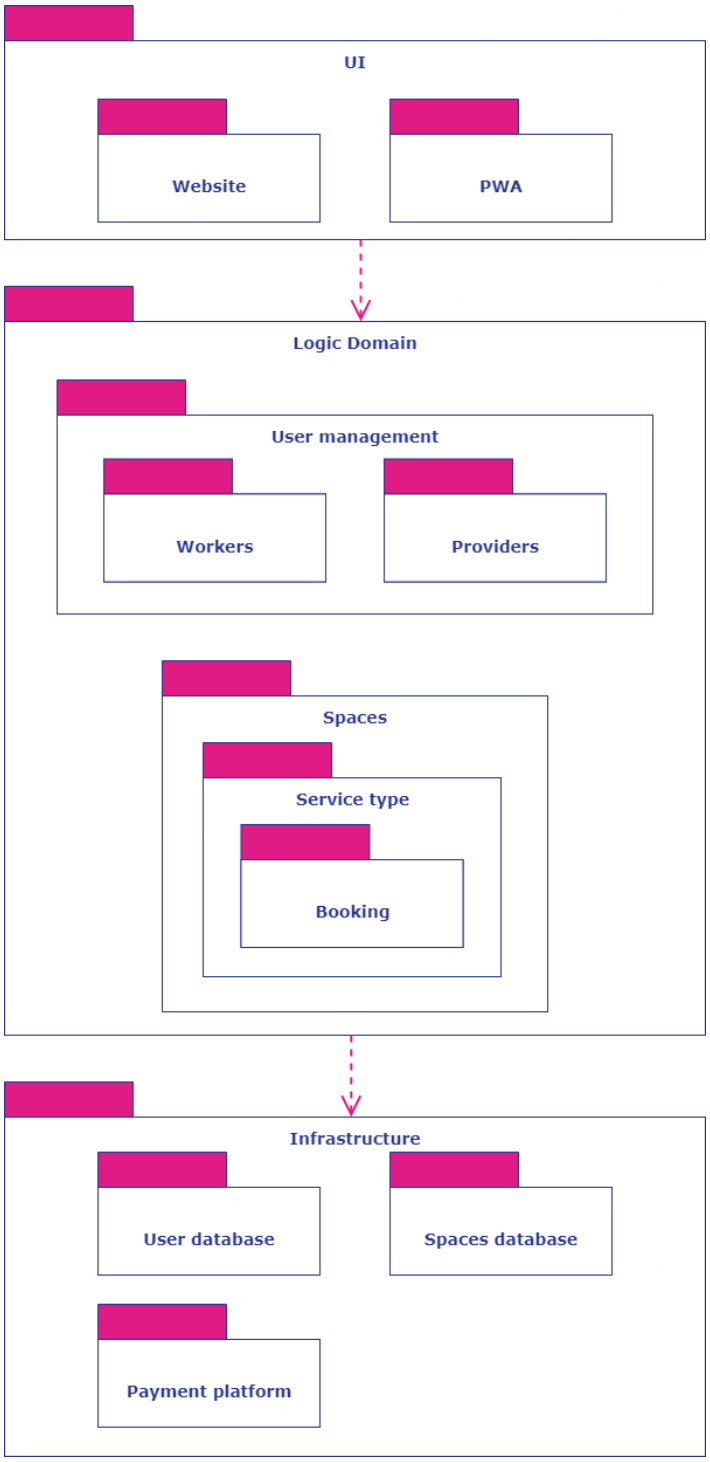
5.1.1 Edit job title

Name:	USC-3: Edit job title
Created by:	David Araújo
Actor:	**Primary**: User **Seconday**: CoworkClub

Brief Description:	A user that access **CoworkClub** may need to change his/her personal information, like email, password or job title.
Trigger:	A user need to have a flexible profile that adapts to his personal changes
Preconditions:	PRE-1: Visit of a specific user's profile PRE-2: Be logged in
Postconditions:	POST-1: The platform displays information relative to this user profile POST-2: Save the new date related to this specific user in a secure way
Normal Flow:	1. The user logged or registers in. 2. Access his/her profile page 3. Selects the option to edit his/her profile 4. Change the job title information 5. Submits the change to the platform
Priority:	High
Frequency of Use:	Will be highly volatil, depending on the specific space performance and relevance between the users.

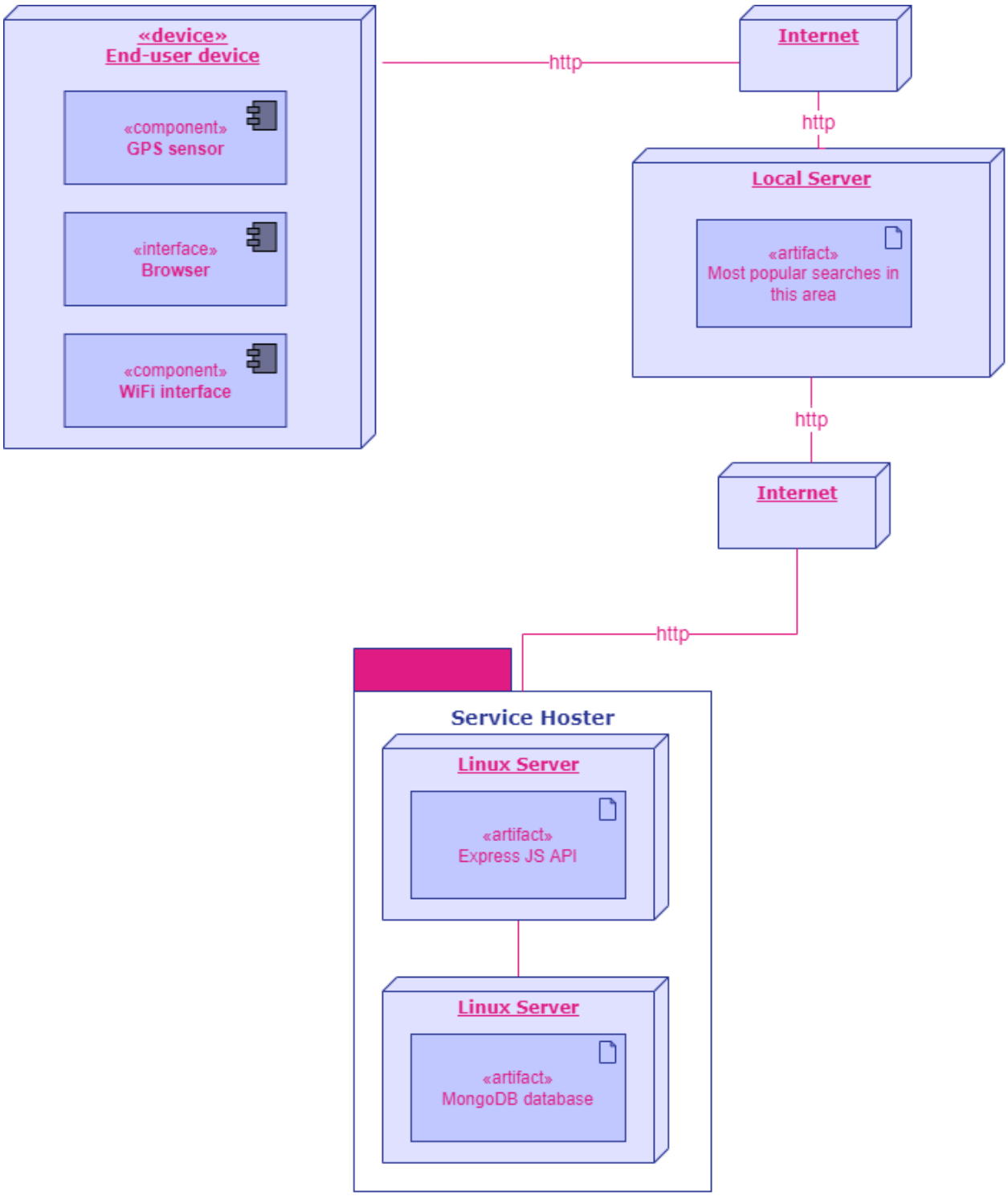
Annex

Software Architecture



Picture 1 - Package Diagram of the domains of the architecture.

Software Architecture



Picture 2 - Physical Diagram.