



deti

universidade de aveiro
departamento de eletrónica,
telecomunicações e informática



CoworkClub

Work **where**, **when** and **how** you want.

*Architecture overview of a coworking spaces aggregator for the subject of **System Analysis**.*

Presented on: 25th May 2022, at P6.

Group: David Araújo 93444, Daniel Capitão 75943, Filipe Costa 77548 and Samuel Teixeira 103325.

Architecture



1. **Global** reach with **maximum compatibility**.
2. **Low end-user** requirements.
3. **Reliable** user experience.

Architecture Requirements

- ❑ 250 **concurrent** sessions.
- ❑ Secure **storage**.
- ❑ System must **adapt results** based on **user location**.
- ❑ **Cross-platform** support.
- ❑ Integrate with **calendar applications**.
- ❑ System must **keep and history** of **at least 1 year** for each transaction.
- ❑ **Recognisance** of UI should be preferred with **re-use** of simple **components**.
- ❑ Should integrate a **payment platform**.

Software Architecture

Can include **internal** modules as well as **external** calls and **embedded** elements.

Partitioned in **3 main areas**:

- ❑ **Space** management;
- ❑ **Search** and browsing;
- ❑ **Booking** management.

Layered approach correlated with **depth of interaction** from the user with the system.





Vue JS

- ❑ **Client-side** render;
 - ❑ **Modularization**;
 - ❑ High **scalability** potential.
 - ❑ High **adaptability** for **mobile** applications.
 - ❑ **Open-source** with a **large** and growing **community**.
 - ❑ Option to easily turn into a **PWA**.
-

Space Card Component

Capability of creating **multiple instances** of a component **easily**.

Full **parameterizable** components.

Thank to the **Single-File Component** approach, **clusters of files** for one component are **non existing**.



```
SpaceCard.vue

<template>
  <div class="space-card-container" v-bind:class="rootClassName">
    
    <div class="space-card-container1">
      <div class="space-card-container2">
        <h3 class="space-card-text">{{ title }}</h3>
        <div class="space-card-container3">
          <h3 class="space-card-text1">{{ Rating }}</h3>
          <svg viewBox="0 0 950.8571428571428 1024" class="space-card-icon">
            </svg>
          </div>
        </div>
        <div>
          <span class="space-card-text2">{{ description }}</span>
          <div class="space-card-container4">
            <next-button button="Read more"></next-button>
            <h3 id="price" class="space-card-text3">{{ heading }}</h3>
          </div>
        </div>
      </div>
    </template>

<script>
import NextButton from './next-button'

export default {
  name: 'SpaceCard',
  props: {
    Rating: {},
    image_alt: {},
    rootClassName: String,
    description: {},
    heading: {},
    title: {},
    image_src: {},
  },
  components: {
    NextButton,
  },
}
</script>

<style scoped>
...
</style>
```

Express JS

ex

- ❑ **Node JS** web framework.
 - ❑ **Minimalist** and **reliable**.
 - ❑ Development **speed**.
 - ❑ **Advanced routing** mechanisms.
 - ❑ **Flexible middleware** modules.
 - ❑ Wide **HTTP focused** libraries.
-

Aveiro Hub Space

<https://coworkclub.com/spaces/AveiroHub>

Express JS is able to **access elements** of the **path** directly.

Every URL path is **specialty crafted** to provide all **necessary attribute**.

All the data is returned in **JSON** format.

```
JS api.js

app.get("/spaces/:name", (req, res) => {
  /* Route path

  /spaces/AveiroHub

  :name      → Each space has a unique name id to retrieve info from databases

  This will be fetch every time a space's page is opened.
  */
  var name = req.params.name;
  /*
  Retrieving matches from database with routing parameters...
  */
  var response = {
    id: "AveiroHub",
    name: "Aveiro Hub",
    rating: "4.5",
    description: "Lorem ipsum dolor sit amet...",
    priceTable: [
      {
        id: "hotdesk",
        day: "15",
        week: "60",
      },
      {
        id: "office",
        day: "25",
        week: "100",
      },
    ],
  };
  res.send(response);
});
```


Space Search With Filters

<https://coworkclub.com/spaces?search=Aveiro&type=hotdesk&amenities=coffee+fax>

More complex than the last, yet **query parameters** still easily accessible.

No hardcoded filters, **flexibility** and **specificity** is paramount

```
JS api.js

app.get("/spaces", (req, res) => {
  /* QUERY types

      /spaces?search=Aveiro&type=hotdesk&amenities=coffee+fax

      search      → specifies the text inputted in search box
      type        → type of item being search
      amenities   → list of amenities selected
  */
  var r = req.query;
  /*
      Retrieving matches from database with query parameters...
  */
  var response = [
    {
      id: "AveiroHub",
      name: "Aveiro Hub",
      rating: "4.5",
      description:
        "Lorem ipsum dolor sit amet...",
      price: "30",
    },
    {
      id: "WorkSmart",
      name: "Work Smart",
      rating: "4",
      description:
        "Lorem ipsum dolor sit amet...",
      price: "27",
    },
  ];
  res.send(response);
});
```

Key Abstraction



User Account

Space Profile

Availability

Target Audience