

# Introduction

---



## A3.3 Learning Activity

Software architecture design using C4 model.



### Instruction

- Based on research and the document provided by the advisor, perform the project documentation using the scrum framework for the case study.
- The activity must be done using a platform such as **Notion**, or **Confluence**, and must be sent in PDF style, naming it with the nomenclature **A3.3\_NombredelaActividad\_NombreAlumno.pdf**.
- Your repository plus it must have a file **readme.md** within your root directory, with information such as student data, work team, subject, career, advisor data, and even logo or images, you must have a section of contents or index.

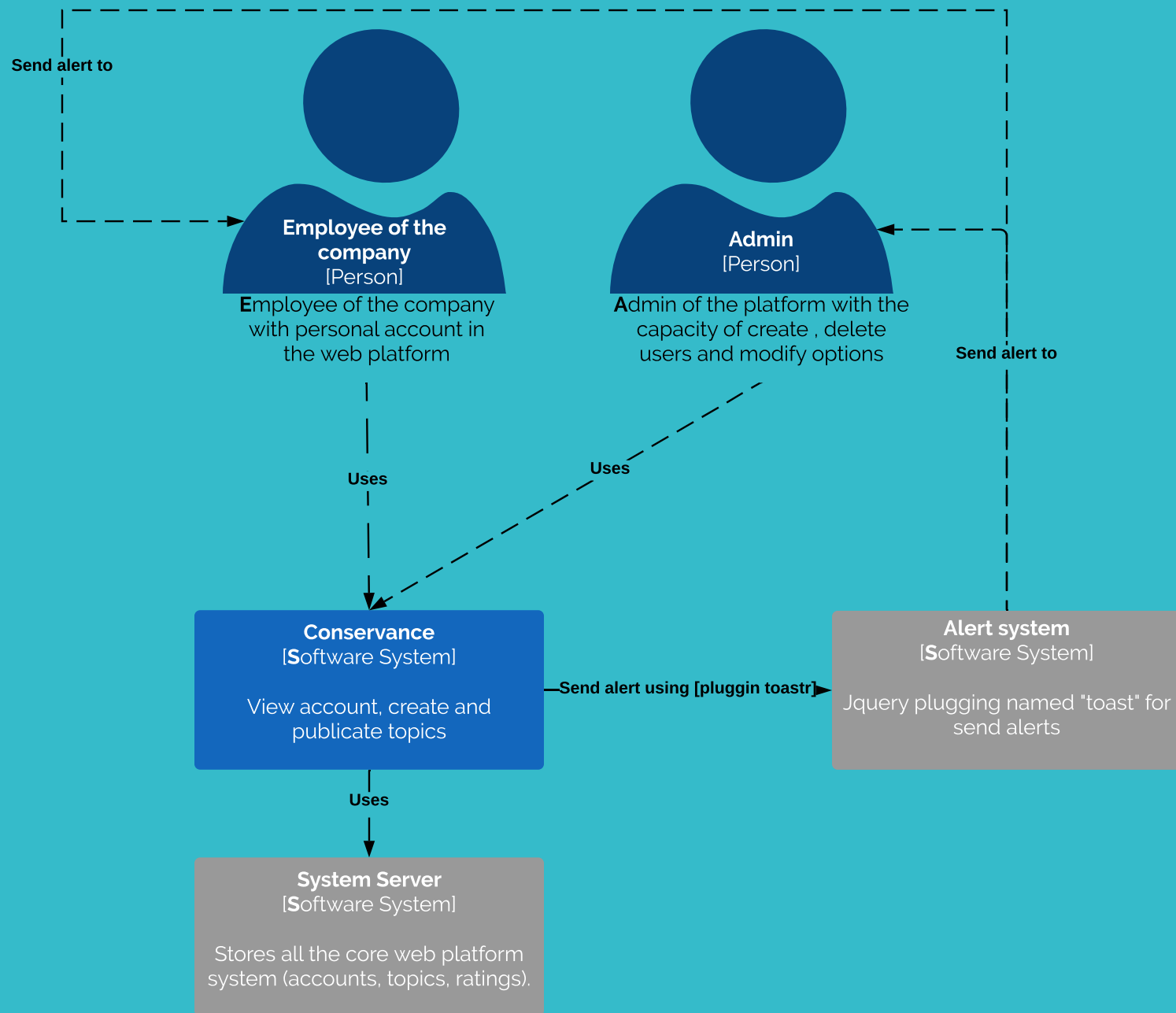


### Development

1. Create the following diagrams for the case study:

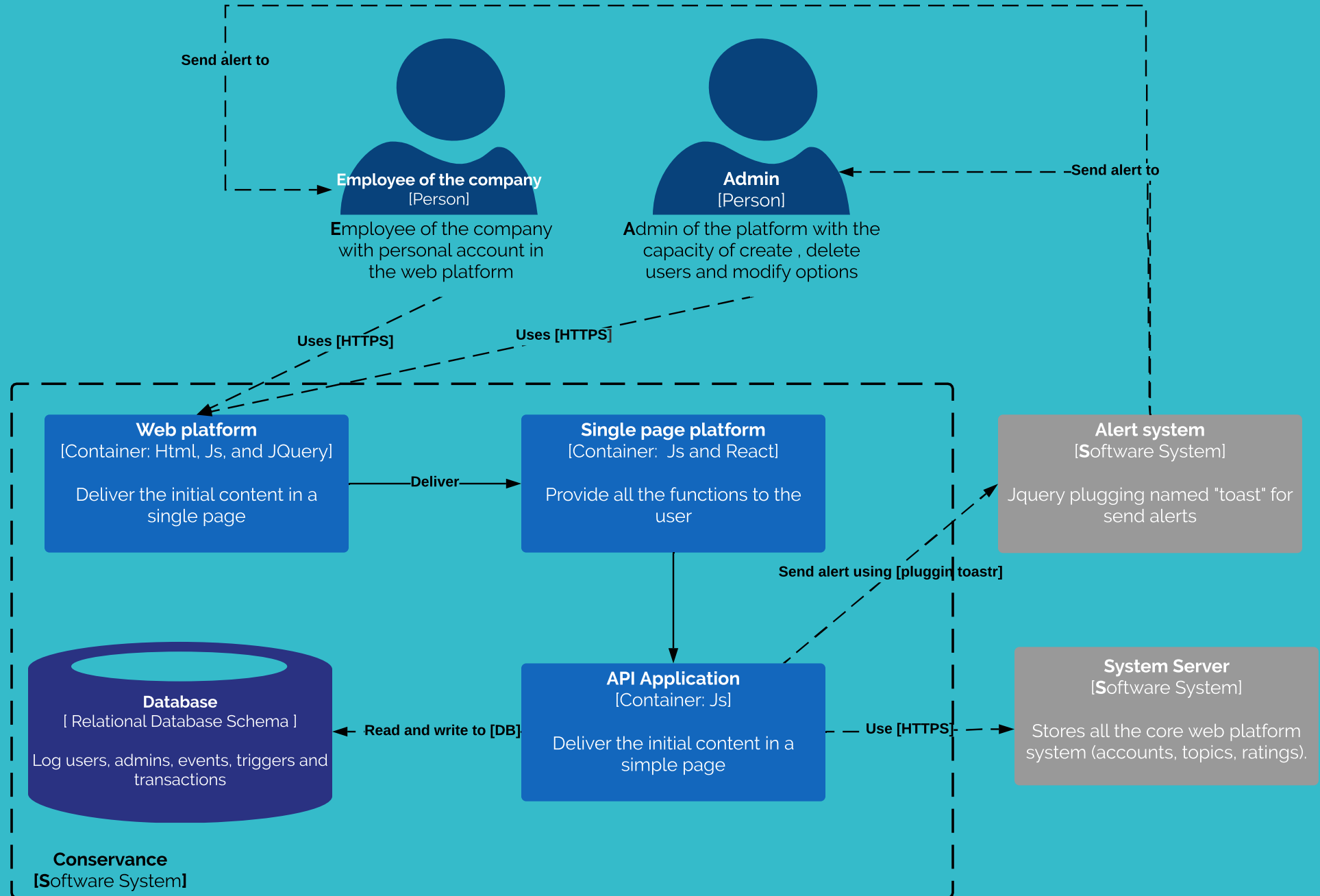
- ☒ 1. Context diagram.
- ☒ 2. Container diagram.
- ☒ 3. Component diagram.
- ☒ 4. Class diagram.

# Context Diagram



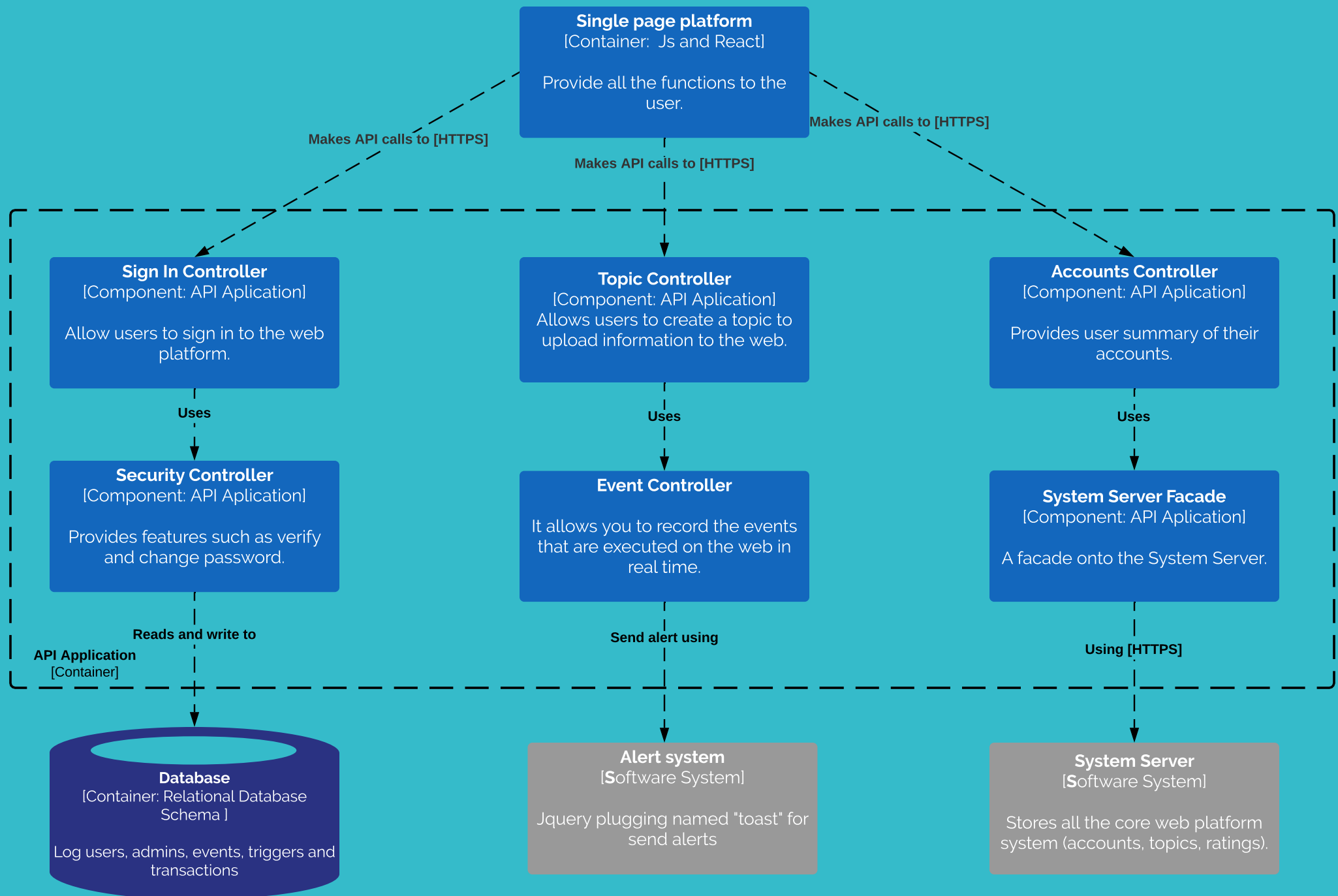
System Context Diagram for Conservance

# Container Diagram



Container Diagram for Conservance

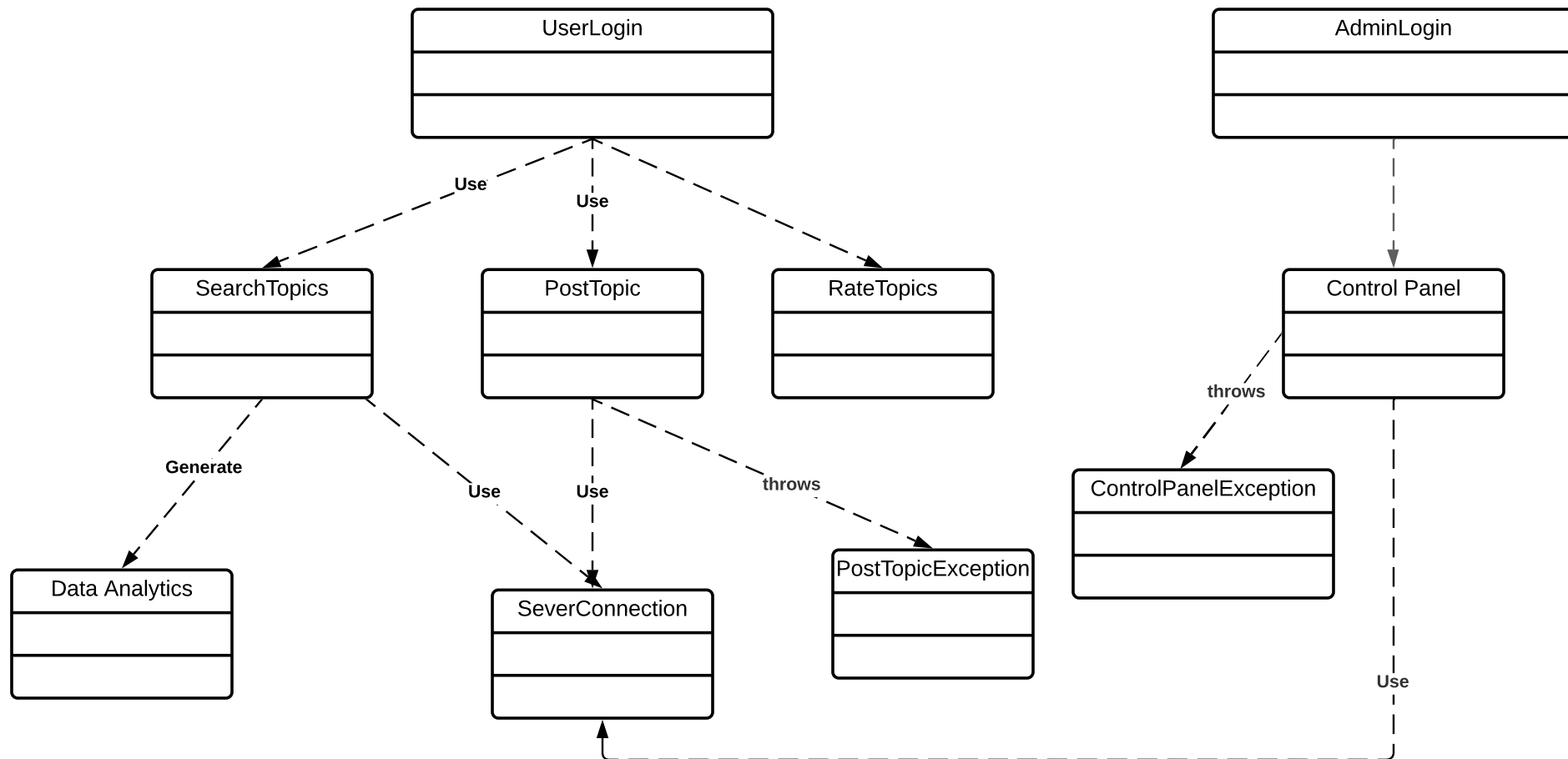
# Component Diagram



Component Diagram for Conservance

# Class Diagram

mx.com.perseverance.conservance



Class Diagram for Conservance

### **Conclution by Edson:**

By modeling the software architecture using C4 it is possible to take the main part of the system (case study) and from there focus on a point and start describing those parts, also naming the technologies to be used and a brief description of what that part represents, I consider it a good tool to detail the case study. The context diagram, containers, components and classes were used, each one is focused on different parts of the system but all together make up the case study.

### **Conclution written by Antonio:**

With the use of the architecture of the Model C4 we can see how our system works , using the diagram context has a view of the surface , subsequently diagram the container which is a view a little more specific of how they relate to the different elements, the component diagram is a little bigger where we can observe the different components of the system and their relationships , eventually the class diagram as its name indicates are the classes and how they relate to each other. It is easier to be able to appreciate all the elements that interact in the system, than functionality and what dependencies they have on each other.

### **Conclution written by Joseph:**

As far as possible, using these four diagrams helps us to have a preview of the system functions. It is important to have done the design in a somewhat abstract way as shown in the C4 model documentation and thus be able to build the software with such complements where we describe the context of the system from a general overview, the containers to be able to observe what the system will contain and a component diagram to be able to include a zoom in the container and fix the components of the same.

However, this type of practices for the documentation of a software are very important for both developers and the potential customer.



[Go to my GitHub repository](#)