Software Architecture Document

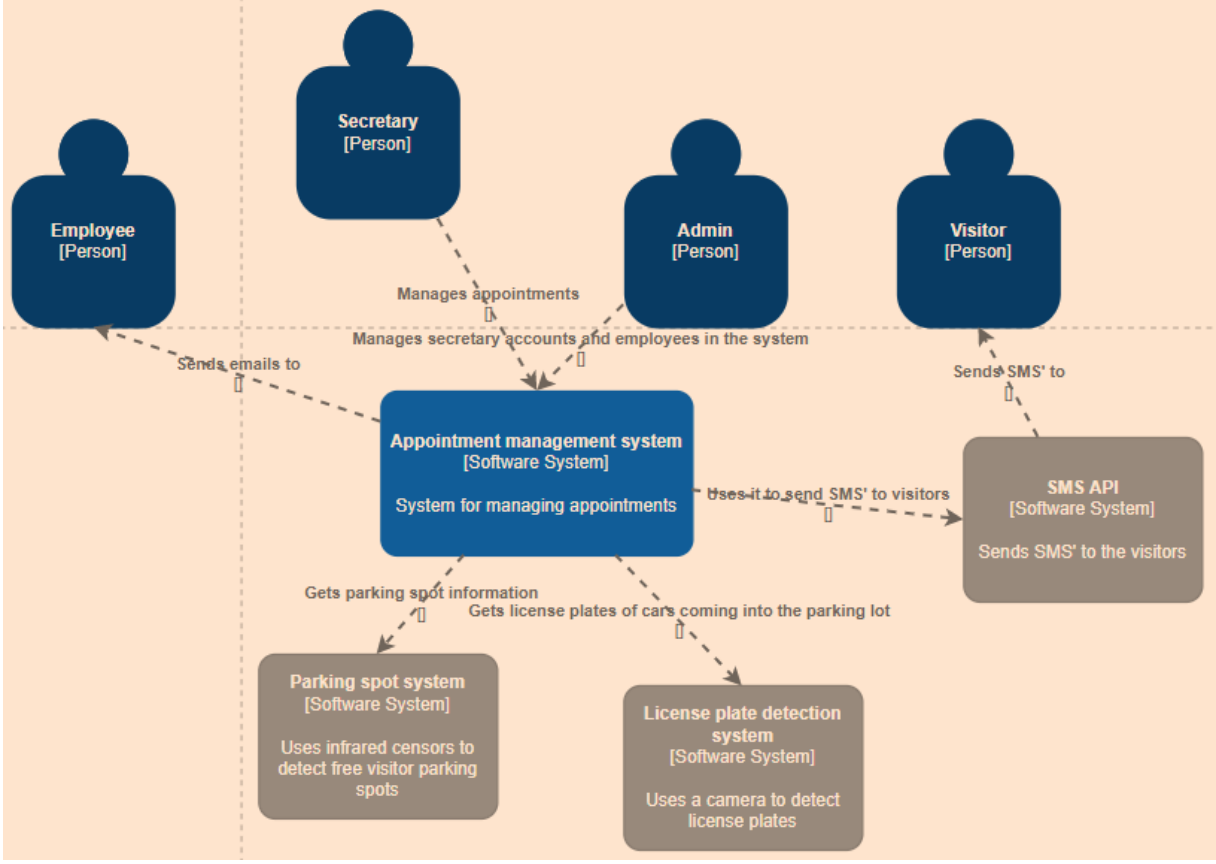
OnlyDevs;

Table of contents

Version table

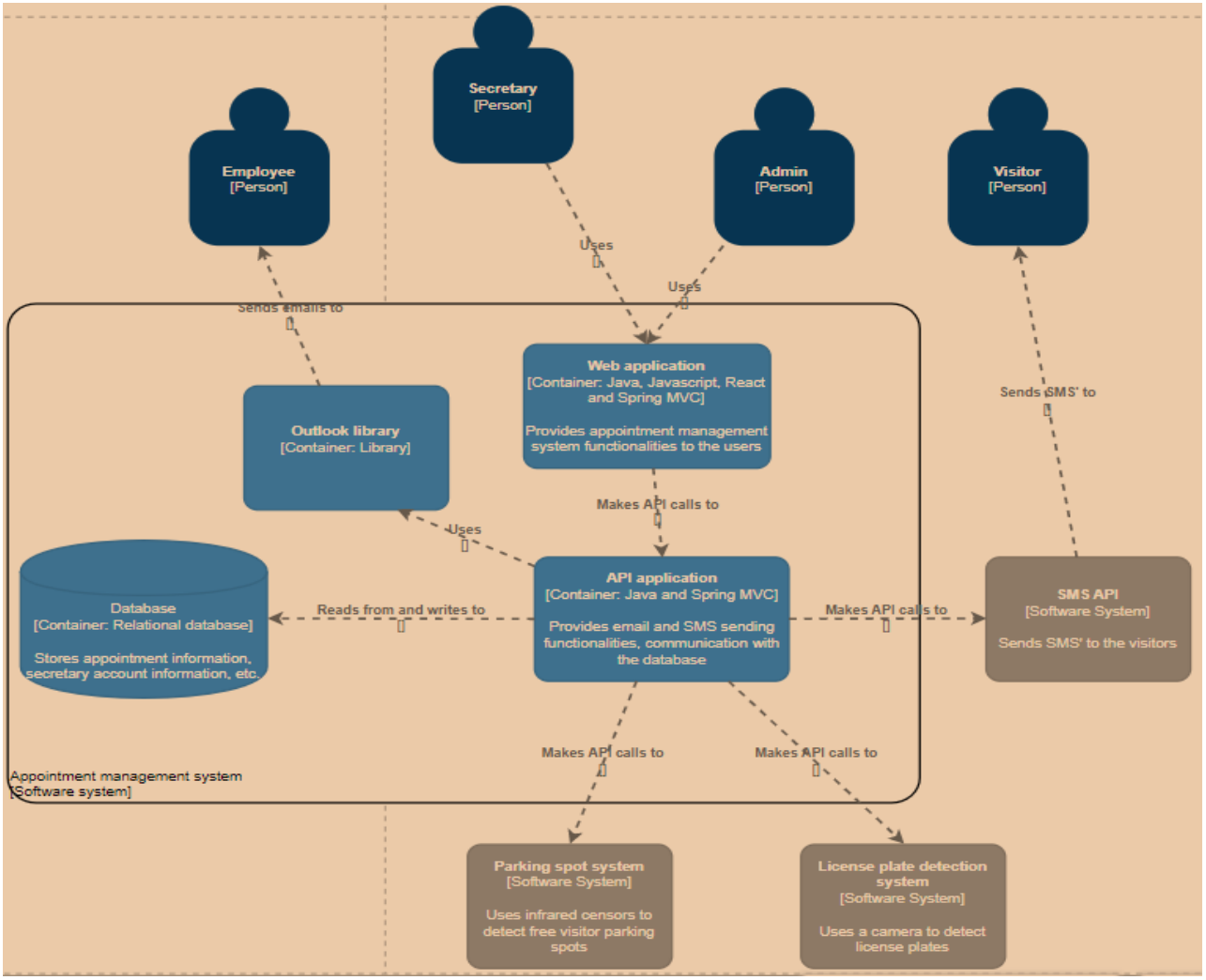
Version	Date	Author	Description
1.0	17.10.2022	Everyone	Document Created

1. System context (C1)



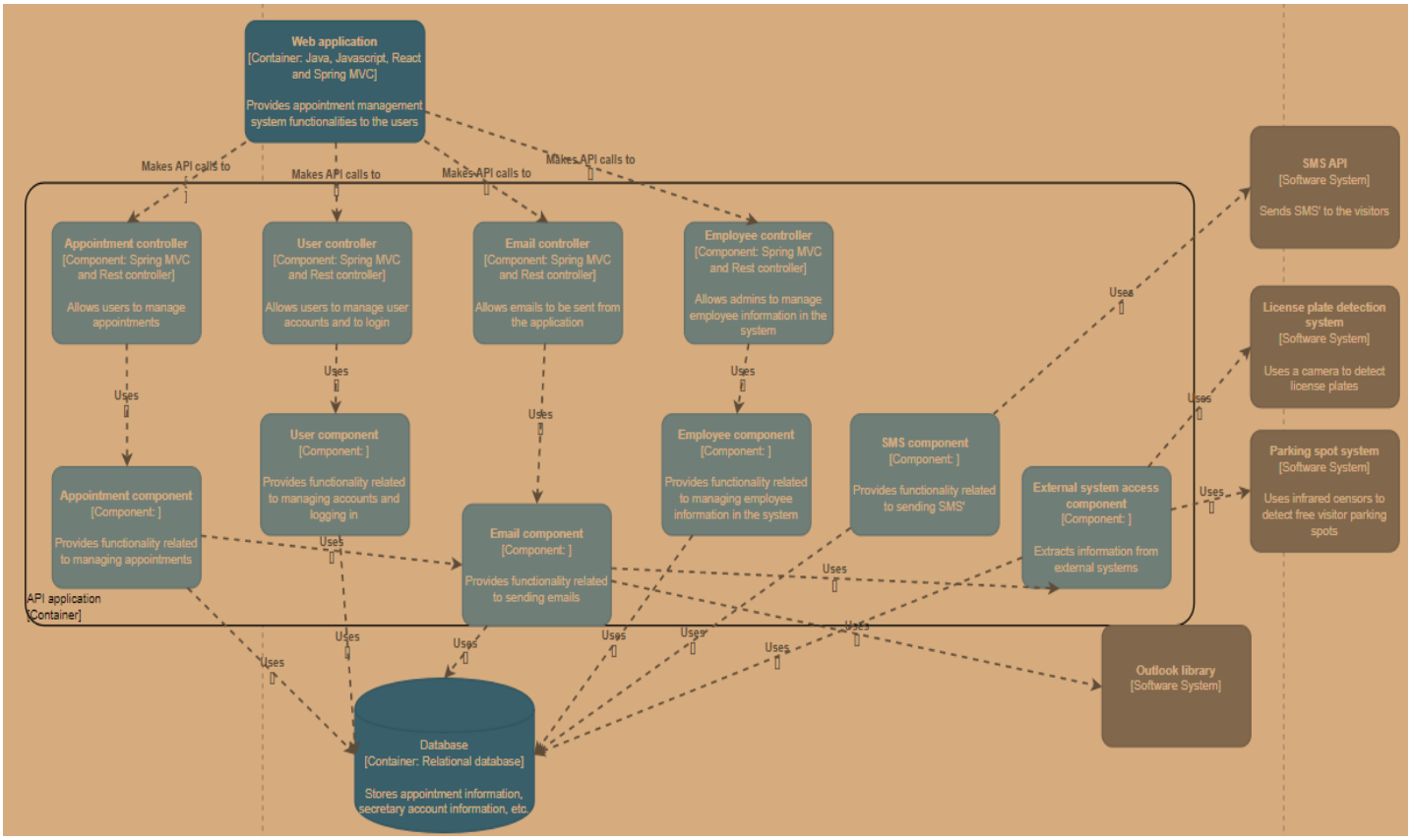
2. Containers (C2)

The secretary and the admin are using the front-end web application, which is talking to our REST API. The API talks to database, external SMS API, outlook library, parking spot system and license plate detection system.



3. Components (C3)

Our back-end application consists of controllers and components. The Controllers are going to consist of endpoints for the REST API. The Components are going to be responsible for handling business logic and connecting to the database and communication with our external systems.



4.Architecture Choices

1. Why web app?

- We chose to create a web app for the following reasons:
  - No installation is needed
  - No compatibility issues because all users can access the same version
  - Can be accessed from anywhere in the world through a web browser
  - Can run on multiple operating systems as long as the browser is compatible

2. Why React for the front-end application?

The Java Script based library - React allows us to create more complex dynamic and user-friendly user interfaces fast. First, its component–based character allows us to break up the application into small independent reusable pieces of code – components. The separation of concerns is by functionality not by technology (HTML, JavaScript, CSS). Therefore, it reduces the amount of code and speeds up the process of development. Secondly, it renders only that should be rendered and not the whole page every time something changes which makes it efficient.

3. Why Java Spring Boot for the backend?

We are making use of Spring Boot because it is a very convenient way to set up, configure and run our API. Spring boot takes care of all these things therefore it decreases our effort. Also Spring boot is using dependency injection approach which ensures the quality of product and code. The most significant Spring boot advantages are creating stand-alone Spring applications that can be started using Java -jar, no requirement for XML configuration and tests web applications easily with the help of different Embedded HTTP servers such as Tomcat, Jetty, etc. We don't need to deploy WAR files.

4. Our prototype – infrared sensors

In order to simulate the real-time parking activity, we'll build a prototype and connect it to our management system.

The system will automatically find parking spaces that are empty and available. New vehicles can enter the automated parking if the slot is free, otherwise a text message will be sent to the clients with a scheduled meeting.

For building this prototype system we will use:

- Arduino UNO
- IR sensors (for detecting the movement of vehicles)
- 16×2 LCD (to display the parking status)
- Jumper wires
- a breadboard

5. Outlook library

Since the SIOUX company uses the Outlook calendar for appointments, we decided to use the outlook library to make an appointment so the employees of SIOUX and potentially the visitor can have the appointment in their calendar.

6. SMS API

Our product owner wants the visitor to be notified to which parking they should park at. A SMS is the way to go because when the visitors are driving, it takes too much effort and time to open an email. The SMS does not require internet access and can be seen easily so it's a convenient solution.

7. Relational Database

We have chosen to use a relational database for this project firstly because we all have the more experience using it out of any other data management system and secondly because of their highly organized structure combined with flexibility offered by the relations between tables.