# Design of a ticketing system amid Covid-19 pandemic

Federico Pappani - 298223

August 2021

Project for Course of Databases and the Web

Bachelor's Degree in Computer, Electronic and Communications Engineering
University of Parma
github.com/pappani/CovidTicketingSystem

**Project Goal**

Development of a ticketing system for events, aimed at contrasting the spread of Covid-19, using technologies studied during the course. The system allows the tracking and controlled entry of participants in generic events.

**System requirements**

User-friendly interface intended for use by a non-expert user, with fields that can be easily and quickly filled in. Simple, fast and secure booking procedure: each user requests a personal ticket which is privately delivered to the user. The system is managed by an administrator user who, having access to a reserved panel, can control participants, events, and tickets. The ticketing procedure must be fast and minimally invasive on users, in order not to cause too many slowdowns in the management of events, and it must be safe and reliable.

**System development and implementation**

The system consists of a web server that manages client requests, and the information necessary for the operation of the platform is saved in an SQL database. The frontend and GUI are built with the Bootstrap package. The backend is made with PHP and some special libraries for PostgreSQL, STMP mail and QR Code generation. The communication between frontend and backend takes place via HTTP REST API. I have published a demo of this system, hosted on Heroku hosting service, using ElephantSQL (PostgreSQL) for database management. The Elastic Email service is used to deliver the tickets, via the SMTP protocol, connected to a Gmail account.

**Use case example**

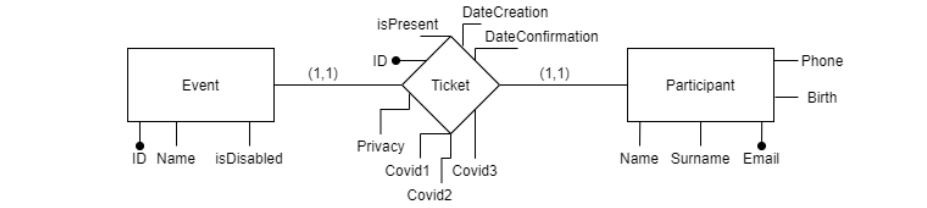
- Participant: The user connects to the web page of the service, completes the form by entering his/her personal data; if the user's profile is compatible with health requirements (and if the user is not already registered for the event) a personal ticket is created for the user, which is delivered via email. The ticket contains the user's first and last name, the event name, and a unique QR Code. Upon arrival at the event the user will show the ticket to an employee, who will scan it, and the user will be automatically marked as present at the event.

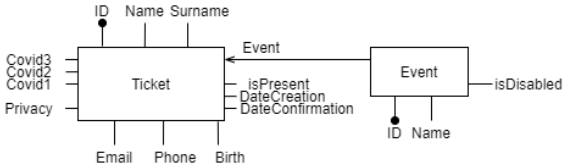
- Administrator/Employee: The administrator or employee user authenticates with a username and password and accesses a reserved panel. From the latter he/she has access to the complete list of participants, and can manage upcoming and current events, but above all, by logging in, the browser in use is enabled to register the attendance of the participants. Upon arrival of the participant at the event, the employee scans the ticket and immediately has access to the participant's data. The administrator can, through the reserved panel, create new events and open/close event registrations based on the number of available tickets. Lastly, the administrator can download the participant list as a CSV file.

**How does the QR Code ticket work?**

The QR Code in the ticket contains a unique URL, containing the user data necessary for recognition (registration email and event name), and that URL is "protected": if an unauthorized user opens the link contained in the QR Code, nothing happens, and he/she is simply redirected to the homepage. If, on the other hand, the link is opened by an authorized user, the system returns the information of the user relating to that ticket, and simultaneously carries out some security check operations; for example, it notifies the employee if the ticket has already been used before, or if the participant is a minor. If everything is correct, the participant is marked as present at the event and is allowed to enter.

### Diagrams




Ticket(<u>ID</u>, Name, Surname, Email, Phone, Birth, <u>Event</u>, Privacy, Covid1, Covid2, Covid3, isPresent, DateCreation, DateConfirmation)
Event(<u>ID</u>, Name, isDisabled)

Ticket.Event → Event.Name

### SQL Queries

CREATE TABLE event(id SERIAL PRIMARY KEY, name VARCHAR(255), isDisabled BOOLEAN)

CREATE TABLE ticket(id SERIAL PRIMARY KEY, name VARCHAR(255), surname VARCHAR(255), email VARCHAR(255), phone VARCHAR(255), dateCreation TIMESTAMP, dateConfirmation TIMESTAMP, isPresent BOOLEAN, covid3 BOOLEAN, covid2 BOOLEAN, covid1 BOOLEAN, privacy BOOLEAN, event VARCHAR(255), birth VARCHAR(10))

CREATE VIEW ticketsCurrentEvent AS SELECT * FROM ticket WHERE event = (SELECT name FROM event ORDER BY id DESC LIMIT 1)

### Demo

A live demo of the ticketing system is available at fpevents.herokuapp.com. Source code is available at github.com/pappani/CovidTicketingSystem.

### Bibliography


- PHP
- ElephantSQL, used for DB management.
- Bootstrap, used for GUI.
- PHPMailer, used for SMTP.

- Heroku, used for platform hosting.
- PHPQRCode, used for QR Codes.
- Elastic Email, used as SMTP server.