Ticketing App

You must implement the back-end system of an application used for selling tickets. The back end will expose REST endpoints with functionality, which are consumed by a separate front-end system. Also, it will store the data in an SQL database.

You must implement the following endpoints:

- **add a ticket** – allows a user to add a new ticket in the database. The following business rules are defined for this endpoint:
    - If the event is SPORT_COMPETITION and there is already a ticket with the same combination of type, event date and event venue in the database, the request will be rejected, and the user will receive the message "There is already a sport competition on the same date and venue".
    - The request to add a ticket has the following structure:

Field name	**Present on the request**	**Mandatory on request**	**Present on response**	**Other validations**
id	no	no	yes	PK, auto-generated on DB level
event title	yes	yes	yes	String with max length of 200
type	yes	yes	yes	May be only one of the following values: THEATER, CONCERT, SPORT_COMPETITION.
event date	yes	yes	yes	Date, cannot be in the past
event venue	yes	yes	yes	String with max length of 200
price	yes	yes	yes	Double positive value

- **get a list of tickets** – allows a user to get a list with tickets stored in the database, using filters. The user must filter by event date and can optionally filter by the event venue. The response will contain all the details of the tickets (all the fields for a ticket record).

Unit tests should be implemented for the service methods, with a coverage of at least 70% per line.

The type of the SQL DB is chosen by you.

Please create an archive (for example, zip) with your project and the SQL script used to create the database table. The archive file name must contain your name and your group name. Please upload the archive here: https://www.dropbox.com/request/xvCDVWQMwAiNfzS1M3bI .