Description

Our project is a Cassandra database for storing and managing concert tickets. The project is available on github: https://github.com/WojciechMajewski/Cassandra-ticket-store
The database is handled through a bash interface, with options which correspond to actions and, by an extension, queries, inserts and updates to the database. These options currently include:

1. List all scenes and concerts
   This option firstly lists all scenes and how many tickets have been bought for them, then it lists all concerts, with the name of the artist, the scene on which the concert will take place, the number of available tickets and the date of the event.
2. Buy a ticket
   This option guides the user through a ticket reservation process, first by asking which artist would the user like to see, then which ticket number the user would like to buy and lastly, the name and email address of the customer.
   By completing this process this ticket will now be occupied until the user decides to sell the ticket.
3. See ticket
   This option asks the user about the email address to which there is a ticket assigned. It will show the tickets artist, scene, ticket number and the name of the ticket owner.
4. Sell ticket
   This option allows the user to sell (delete) a ticket assigned to an email address, for which it asks.
5. Update ticket holder
   This option shows information about the ticket like in option 3, and then asks for a new name the ticket is supposed to be assigned to.
6. Exit the application
   As the name suggests, this option quits the application.

We have also performed stress tests, which are included on our github repository, both .yaml files used with cassandra-stress, as well as the graphs that were created.

Database schema

The schema below shows elements of all tables used, with Primary Keys marked.

Database Schema

Table Name	Columns
scene	• PK scene_id uuid • name text • tickets list<int>
concert	• PK concert_id uuid • artist text • scene_id uuid • available_tickets list<int> • concert_date timestamp
ticket	• PK ticket_id uuid • ticket_nr int • concert_id uuid • name text • email text

Problems encountered

Our first issue was setting up the environment for the cassandra database. It turns out that WSL1 and Windows10 are not well suited for running docker and cassandra, so after a long and arduous journey some of us had to upgrade their PC operating systems.
After setting up Ubuntu 20.04.6 LTS, installing docker and cassandra, we still were not free from errors and hijinx.
Setting up cassandra containers is not an easy task for the impatient, as all 3 containers need some time to be set up correctly and running them in quick succession can lead to errors.
Similarly, running main.py to start the application can fail if the user does not wait for about a minute for everything to start up the right way. In case of a container error, fresh start is advised, tha is, to stop all containers and start them correctly again. For the main.py error, it is best to just wait for a minute and run the command again.
In regards to stress testing, we used cassandra-stress which is truly lacking in terms of online documentation, and thus our testing process proved to be quite challenging.