Special Event Registration System 260844113

In this deliverable, the Event Registration System provided as an initial code contribution needs to be extended individually. In addition to the existing features developed in tutorials, the *Specialized Event Registration System* shall support the following features.

1. The system shall support managing **promoter**s. Each **promoter** can **promote** events. All **promoter**s count as persons, that is, they can still be registered to events.
2. In addition to generic events, specific events can be registered as **circus**. Each **circus** has one **company** attribute.
3. Users of the system shall be able to register payments for existing registrations. Participants of an event pay for a registration via **Bitcoin**. A payment shall record the **Bitcoin user** ID in the format `LLLL-DDDD` (where `D` denotes a digit and `L` denotes a letter of the English alphabet) and the payment amount in CAD.

Backend updates

You need to implement all features mentioned above as business methods and make them available as RESTful services using Java Spring Technology. The services need to persist data by using a database. The backend application is provided to you as a gradle project in the *$REPO_ROOT/EventRegistration-Backend/* folder, where *$REPO_ROOT* denotes the root of your assignment repository.

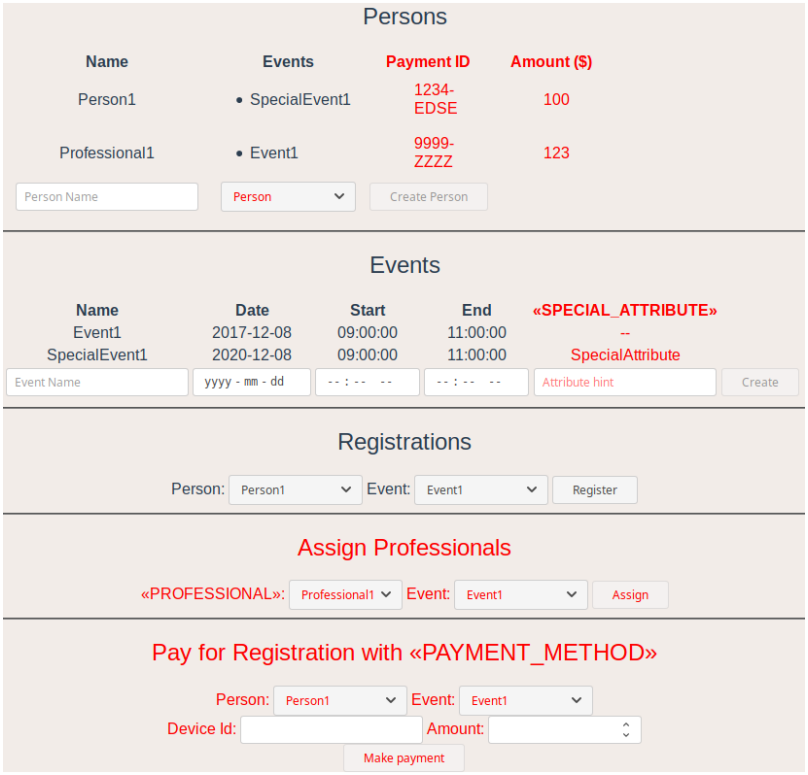
To check the correctness of your application, we published some test cases for the backend, where the source code is located in the *$REPO_ROOT/EventRegistration-Backend/src/test/java/ca/mcgill/ecse321/eventregistration/service*. These tests contain compile errors that you can fix by adding the required missing classes and methods to the project.

Running the bakend tests. To run the backend tests cases we made available for you, issue `./gradlew test -i` in the *$REPO_ROOT/EventRegistration-Backend/* folder.

NOTE	Running backend tests require a database connection, that is, you need to configure the `SPRING_DATASOURCE_URL` *or* the corresponding variables in the *application.properties* file.

Frontend updates

You need to update the initial web UI to support the specialized features. Below is a screenshot as an illustration of the specialized front end page, where red font indicates newly added elements to the existing UI (substitute «SPECIAL_ATTRIBUTE» with **company**, «PROFESSIONAL» with **promoter**, and «PAYMENT_METHOD» with **Bitcoin**).



The source code for the end-to-end test cases provided for the project is located in the folder *$REPO_ROOT/EventRegistration-Frontend/test/e2e/specs*, where *$REPO_ROOT* denotes the root of your assignment repository. **The test cases rely on assigning proper identifiers for the UI elements.** The used identifiers are in the `config.js` and `config.json` files within the frontend test folder *$REPO_ROOT/EventRegistration-Frontend/test/e2e/specs*.

Running the frontend tests. To run the end-to-end (e2e) tests cases we made available for you, you need to issue `npm run e2e` in the *$REPO_ROOT/EventRegistration-Frontend/* folder.

NOTE The successful execution of end-to-end tests requires that you have executed `npm install` in the frontend folder *and* the backend is already running (and it is connected to a database).

NOTE When running `npm run e2e` locally, some end-to-end tests may fail if your computer is operating with a higher load - just re-run the tests if this happens to verify correct behavior. The performance will not be an issue in the CI evaluation environment because there will not be many other processes running there and the database will also be locally available.