**Metropolitan Convention Center**

**Attendee Event Registration Application Design & Work Plan**

Diagram

Description automatically generated

**Application Design:** The application will be split into 3 microservices –

* Client Application (React)
* Authentication Service
* Data Service (HSQLDB)

**Application Internal Functionality:**

* The client application uses JWT tokens provided by authentication service for attendee login and HSQLDB API interactions
* The client application accesses Customers, Events, and Registration records via HSQLDB API
* The client application adds JWT token to all HSQLDB API requests
* The authentication service will interact with HSQLDB API to verify attendee username/passwords during token requests
* The authentication service will interact with HSQLDB API to create attendee accounts during register requests

**Authentication Service:**

* Service Information:
  + context path = \account
  + port = 8081
* Endpoints:
  + \account\ (**Root**)
  + \account\token
  + \account\register
* Token Endpoint:
  + Responds to requests for JWT token
  + Takes username and password as parameters
  + Checks that username exists in Customers
  + Checks that password matches the password stored for Customer found
  + Returns valid JWT token for connection to HSQLDB
* Register Endpoint:
  + Takes name, email address, and password – Adds to Customer data store via Customer endpoint of HSQLDB
* Security:
  + Token **not** required to access Authentication Service endpoints

**Data Service (HSQLDB):**

* Service Information:
  + context path = \api\ **(Root)**
  + port = 8080
* Endpoints:
  + \api\customers
  + \api\events
  + \api\registration
* Customers Endpoint:
  + Supports CRUD management of Customer records
  + Supports GET, PUT, POST, & DELETE requests
  + JWT token required for all requests
* Events Endpoint:
  + Supports CRUD management of Event records
  + Supports GET, PUT, POST, & DELETE requests
  + JWT token required for all requests
* Registration Endpoint:
  + Supports CRUD management of Registration records
  + Supports GET, PUT, POST, & DELETE requests
  + JWT token required for all requests
  + Will allow customers to register for events
* Backend Data:
  + Embedded HSQLDB
* Security:
  + Servlet filter will be used to intercept calls to API – checking requests for JWT tokens

**Work Plan (Task Backlog):**

* Create local Data Service (HSQLDB) API repository
* Create local Authentication Service API repository
* Implement HSQLDB API root endpoint
* Implement HSQLDB API Customer GET endpoint
* Test root and Customer GET endpoints using Postman
* Implement remaining PUT, POST, & DELETE Customer API endpoints
* Test Customer PUT, POST, & DELETE endpoint using Postman
* Test Customer GET, PUT, POST, & DELETE endpoint using a React client
* Implement Events & Registrations endpoints
* Test Events & Registrations endpoints using Postman
* Test Events & Registrations endpoint using a React client
* Create Authentication Service API
* Implement token endpoint
* Test token endpoint using Postman
* Test token endpoint using a React client (Client login will utilize the token endpoint)
* Add authentication filter to HSQLDB API – Require JWT token to access services
* Access HSQLDB ***with*** authentication filter using Postman
* Test login function using a React client
* Test Customers record accessibility using a React client
* Test Events record accessibility using a React client
* Test Registrations record accessibility using a React client