Ticket: Each Facility can save custom IDs for each agent – use this to generate reports of work hours

**Ticket 1: Add a Facility-AgentID Table**

*Acceptance Criteria*

* The new table contains expected fields
* The existing database does not require migration scripts to continue operation

*Time Estimate*

* 1 hour

*Implementation Details*

1. Table name: facility\_agent\_ids
2. Columns
   1. id (primary key, auto increment, unsigned int)
   2. facility id (foreign key, unsigned int)
   3. agent id (foreign key, unsigned int)
   4. custom agent id (unique, varchar(64), not null)
3. The combination of (facility id, agent id) should also be unique.
4. Write the SQL command to create this new table within the existing database

**Ticket 2: Add Methods to perform CRUD operations on the Facility-AgentID Table**

*Acceptance Criteria*

1. All exceptions are caught and handled
2. Caught exceptions should return null for expected string values
3. Caught exceptions should return -1 for expected int values
4. Single database access is required for each method (no chained db queries)
5. No unneeded dependencies are brought in from other classes

*Time Estimate*

* 10 hours

*Implementation Details* – Methods should include

1. Add new entry by taking a facility id, agent id, custom agent id
2. Delete entry by taking custom agent id or facility id and agent id
3. Get entry to return facility id and agent id from custom agent id
4. Update entry to modify custom agent id from existing custom agent id and a new custom agent id

**Ticket 3: Add Test Methods/Cases to validate CRUD operations on the Facility-AgentID Table**

*Acceptance Criteria*

* All tests pass
* All tests cover the complete range of distinct input categories (eg: null, empty, full string, matching, non-matching), including combinations of these applied to the various function parameters

*Time Estimate*

* 10 hours

*Implementation Details* – for each of the methods in ticket 2:

1. Assume facility and agent ids are unsigned integers, custom agent ids are strings
2. include unexpected inputs
   1. negative id values
   2. id values for which there are no matching entries
   3. null custom agent ids
   4. empty custom agent ids
   5. custom agent ids for which there are no matching entries
3. include expected inputs
   1. including id values for which there are matching entries
   2. custom agent id string values for which there are matching entries
4. comprehensive combinations of the above for each of the function parameters where possible
   1. Consider updating an existing entry for a custom agent id using an old custom agent id and a new custom agent id
      1. invalid id, invalid id
      2. invalid id, valid id
      3. valid id, invalid id
      4. valid id, valid id

**Ticket 4: Generate PDF reports for individual agents**

*Acceptance Criteria*

* The pdf report contains shift information for the agent of interest
* An alert is displayed when an invalid query is made (from a caught exception on the backend)

*Time Estimate*

* 4 hours

*Implementation Details*

1. Add a new getShiftsByCustomAgentId function called with the new custom agent id, returning all Shifts worked that quarter
2. Add a new overload ‘generateReport’ method called with the new custom agent id, uses the new ‘getShiftsByCustomAgentId’ method and the existing ‘generateReport’ method, returning a PDF with info for the requested custom agent id.