

Project Documentation

Team Members

Ciarán Sweeny

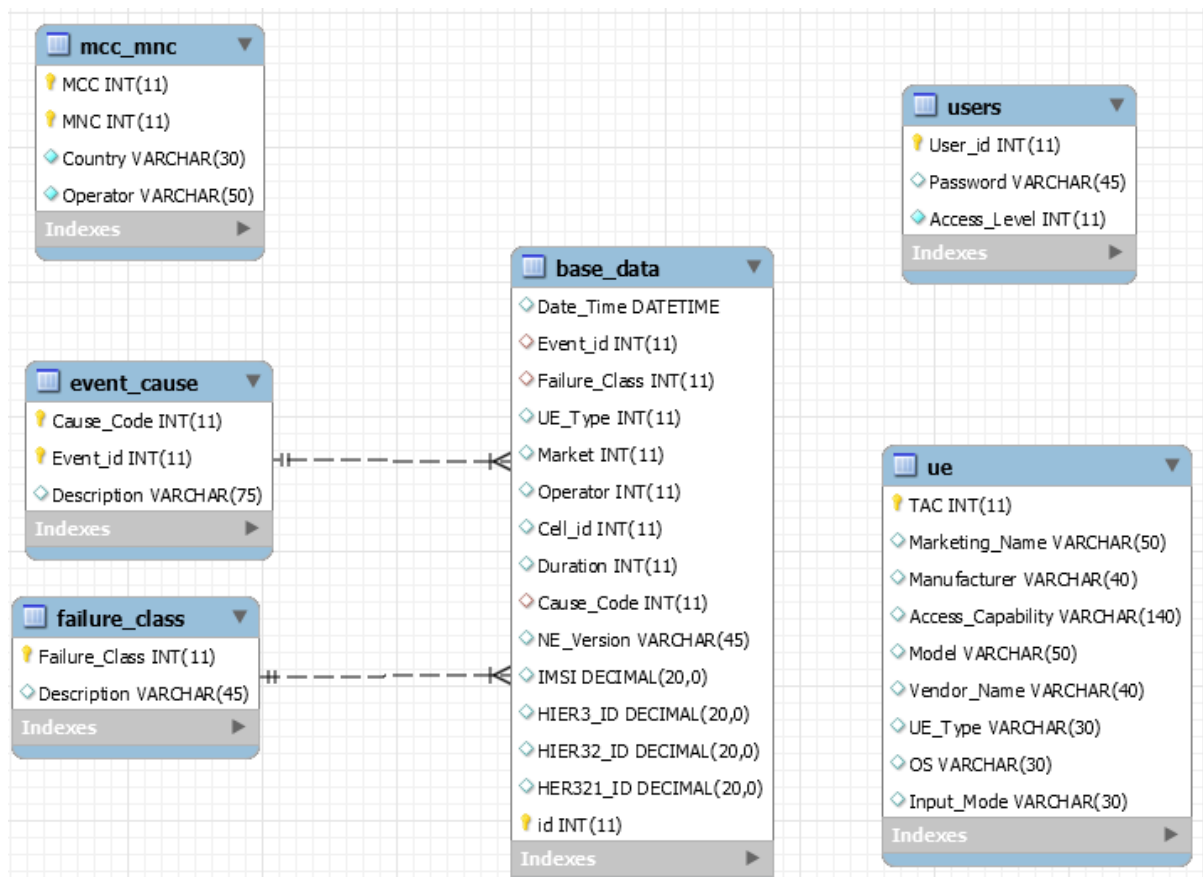
David Leonard

Dimitar Tsvetkov

Sahar Muhamad Ali

Carl McCann

Database



The database model for this project can be seen above. The design of the tables and the relationships between were made based on the data of the Xml file that was given to us. We created six tables based on the requirements that were given to us. The tables that we made are as follows basedate, event_cause, failure_class, mcc_mnc, ue and user table. For the base_date table we added in a primary key called id and made it an integer that increments by one once a new row is entered into it. The base_date table has two relationship, one with the event_cause table and one with the failure_class. The failure_class table primary key is Failure_class and this table has one relationship with the base_data table. The event_cause table has a composite primary key which is made up of the fields Cause_Code and Event_id. This table again has a relationship with the base_date table. The mcc_mnc table has

again a composite primary key which is made up of the integer fields of MCC and MNC. The ue table primary key is an integer field called Tac, this table has no relationships. The user table has a primary key called User_id. The user table is used to store all the users and their password which allows them to log into our system. The Access_Level field is an integer which determines the amount of access the user has that logged into the system. If the Access_level was 4 then the user would have network management engineer rights, if it was 3 then the user would have support engineer rights, if it was 2 the user would have service rep rights and if it was 1 then the user would have customer rights.

Problems with Database

The first issue we had with the data base was when we first tried to load the tables with data from the Xml file on Dropbox. The formatting of the Date initially caused a problem, this was solved by changing the Xml file to csv and with code we changed the format of the date in a way that would satisfy the database. This was done in java with the DataCleaner class. Another issue with loading in data was an empty integer which was read into the database as an empty string rather than as Null. This was solved with code by the DataCleaner class in java which changed the empty Strings in the data with /N. /N is read into the database as null. There is still a small issue with the database with carriage returns.

Full Slice

We were able to perform a full slice with our system. The server was able to request data from the database then receive data from the database. An example of a full slice in our system would be user log in. On the log in page of our system, the user can enter in their user id (Integer) and their password (String). The user id and password is send to log_in.js which from there is passed down to the java class UserRest(Rest). From the UserRest class a @Get request is sent and the user id and password is passed down to UserServiceEJB which these variables are then again passed from UserServiceEJB to UserDaoImp. In UserDaoImp we make a query which allows us to find any user in the user table that has the same user id and password that was passed down. If a match is found then the UserEntity is returned all the way back to the log.js. This will then allow the user to log in. This process is repeated for all entity classes in the system. However, there is a full CRUD setup on the other entity classes such as, create, read, update and delete. There are extra queries such searchById and deleteById included.

Acquillian Testing

At the moment we have a temp class and temp test class that returns true after we run the arquillian test. We are currently writing tests for our real classes but are having difficulties setting up the tests.

File Upload

File uploading, changing, and cleaning works separately from our webapp. We are working on deploying this code on the server through a HttpServlet. This work is ongoing as of yet.

Devices and Systems Used

For this project the IDE used for the java programming was IntelliJ and the server used was Wildfly version 8.2.1.Final. For the database MySql workbench version 6.3 CE was used. This project was made on Linux using Redhat. For working on code separately and combining code together we used Git.