ID: 180917

Course: Software Testing

Date: April 18, 2019

Class Summary

In the class on the date above we were taught how to use the Cucumber testing framework to test a database. The communication to the database is accomplished using the ActiveRecord ruby gem with just a few lines of code. For the class exercise we were required to test a bank application’s database using Cucumber. The application utilized a sqlite3database and as such the ActiveRecord sqlite related module was used. By creating classes containing attributes that correlate to the column names in the database, in addition to extending the ActiveRecord class allows the ruby application to communicate with the database and the data can be referenced or even manipulated. An example is shown below:

account = Account.find\_by\_number(2214)

The Account class extends the ActiveRecord class thus acquiring all the necessary methods from its parent class to communicate with the bank database.

For the class exercise we were required to:

1. Build, run, and debug with the simple database banking account service
2. Understand how things work in Cucumber/Ruby/ActiveRecord
3. Instead of “banking/account” database, test your “user” / “tasks” database, make sure the database schema is set, db is running, db can be connected with ActiveRecord, and the returned task is as expected, write a feature/scenario in Cucumber to verify these
4. Test with other database, e.g. MySQL and Postgres