

Alex Krebiehl
Mark Frye
Khalid Alsuwaiyel

In the CrimeStoppers project, our group relied heavily on the controller GRASP pattern. All user interactions pass through a controller object, which interfaces with the models representing the data.

One of the use cases we implemented was the ability to reset the user's password. If the user is unable to login, they provide their email address, which is passed to an instance of a UsersController. The UsersController handles all interactions between the end user and the users data model in the application. The UsersController will find the appropriate users object and begin the reset password workflow. This consists of instructing the user object to generate a random password, as well as setting a boolean flag to force the user to change their password on their next login. The UsersController will then send out an email to the registered email address with the user's temporary password, then forward a notice to the view layer, notifying the end user that their password has been reset.

Another use case that was implemented was the ability to pull CSV data directly from OpenDataCincy. Again, we used the Controller GRASP pattern and handled the interaction through the CrimeRecordsController. The CrimeRecordsController handles all interfacing with crime records. It is responsible for pulling crime records from the database, as well as importing new crime data into the database.

When an administrator clicks 'import data', the message is forwarded to the CrimeRecordsController. The records controller opens an http request to the OpenDataCincy website and begins the transfer of the data. After the data has been downloaded, it is unpacked and the records controller is ready to start inflating CrimeRecords. The records controller iterates through each line of the CSV file, and inflates a CrimeRecord object for each. After the process is complete, the records controller signals to the view layer that the operation is complete and a message is displayed to the user.

One of our use cases is to present an overview of crime data to the user on the home screen. Since this process is interacting with crime records, the CrimeRecordsController is again used. When the user arrives on the home screen, a message is sent to the records controller to get an overview of the crime data. The records controller will search through the CrimeRecords and figure out what percentage of crimes were a certain type. Since this can be a lengthy process, the results are cached so subsequent requests are faster. After the data is retrieved, the controller passes the data to a Google Pie Chart instance for display on the home screen. This pie chart is passed to the view layer for display to the user.

Our final use cases again followed the Controller GRASP pattern. Editing an incident description, whether it be the name or the severity rating, is passed from the user to the

CrimeTypesController. The controller fetches the relevant CrimeType object and performs the modifications. The controller then forwards the success message back up to the view layer for the user to see. When performing a search for crimes of a given severity or within a time period, the request is again sent to a controller object. The controller fetches the relevant objects from the database and returns them to the view layer for displaying.