Technical Documentation

DegreeAudit++

November 1, 2018

Sara Caponi, Jamie Flores, Daniel Jaegers, Derek Rogers,
Erika Eckfeld, Kate Gardner, Sohaila Bakr

App.js

This component contains a JSON object for the data of our demo. It has the student information like: PawPrint, name, degree program, etc…. It also has a JSON array of all the courses that the student has taken. This component is passed to the UserInfo component so that it uses it to show the data of the student.

Header.js

The header component has the Nav bar of our web app that appears on all the pages. It has the title of the app on the left corner, home, and login button on the right corner. The login button changes dynamically to logout when the user clicks it.

Home.js

Home is the component that renders the other components that are viable on our home page. It renders the userInfo and Tab components which together comprise the majority of our system. It has an array of users as it state which it can then pass to other components to use.

LoginForm.js

This component is incharge of validating if the inputted user name and password are correct and determining what type of user the user is, student or advisor. It makes a request to the back end to check if the information matches our records and then sets the state and props of the component with the correct information. If the entered data is not correct, the user will be presented with a error. If the entered data is correct they will be directed to the next appropriate page which is determined by the type of user.

Main.js

This component displays the correct screen to the user based on the information provided. It also renders information to be displayed on the desired page. It does this by implementing a switch statement with the correct routes in the switch.

StudentSearch.js

This component is comprised of a component with a search bar and button. It allows the user, if an advisor, to search for students by entering the student number. This class will produce an error message if the number entered is not a current student in the database. It then sends this number to the user info and app class to render the correct information to the user.

Tab.js

This component composed of three different tabs: Past Classes, Current Classes, and Outstanding Classes. To navigate between these options, click the tabs. The user is able to swipe through three different tabs that correspond to the table views being shown.

Table.js

This component gathers information from the student's class history and what they need to complete their degree. The tabs relate the to view of the table which indicate whether the classes shown are past classes taken, current classes, or classes that are outstanding. The table is sortable based on certain criteria such as class name, grades, credit hours, etc.

UserInfo.js

This component renders the students information onto the homepage. It is in this class that we will implement the algorithms to calculate the students expected graduation date, and their gpa.

Below is our Domain model from report 1. Although some of the names have changed the overall model of our data flow and relationship between classes and components remain the same .

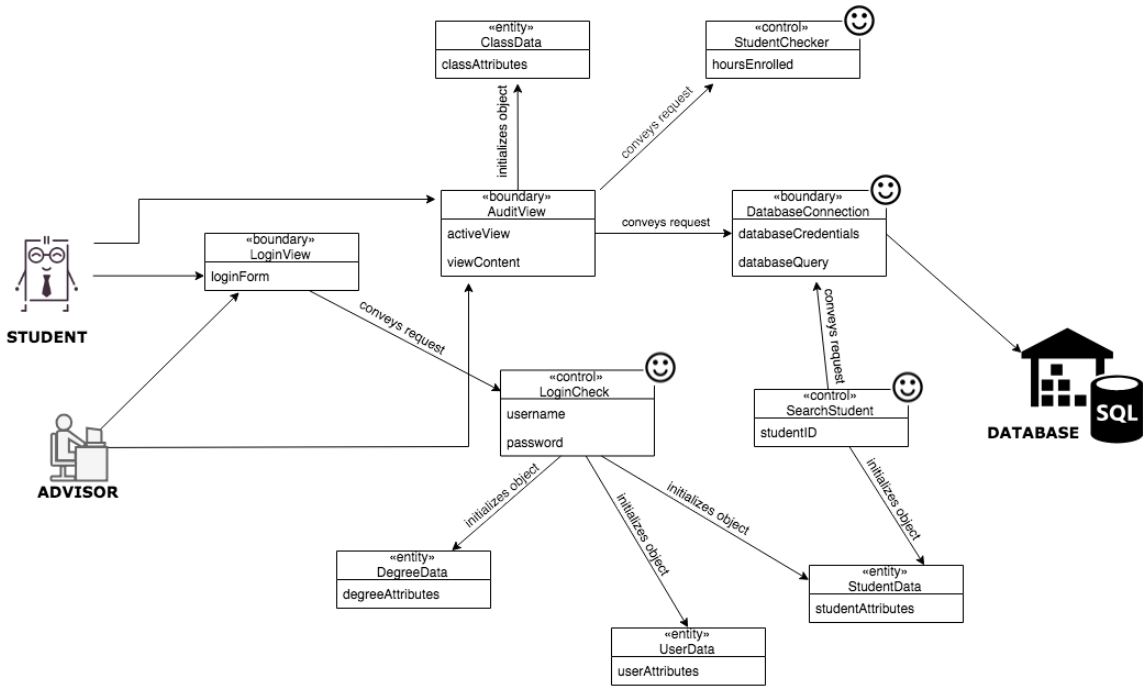

Figure 1. This chart is from our report 1 part 3. It is the Domain model that conceptualizes the model of the domain that describes the behaviours and flow of data through the system.