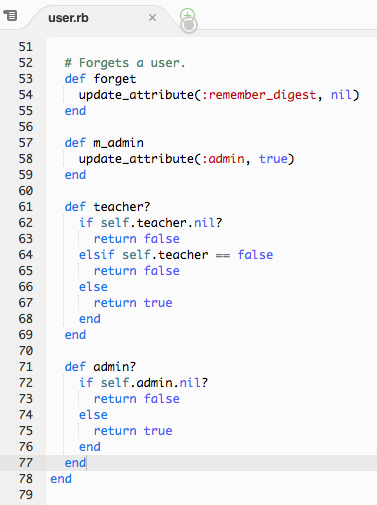
**Model:**

|  |  |
| --- | --- |
| Support_Files/Model_1.png  **Figure 1. Model Code part 1** | Support_Files/Model_2.png  **Figure 2. Model Code part 2** |



**Figure 3. Model Code Part 3**

The above model was implemented to allow the controller to achieve specific actions such as determine if a user is an admin or a teacher. Validations were put in place to validate all user input from the forms however as can be seen in Figure 1, some validation was not implemented as responses from the client team were pending. For these our team wanted to confirm if passwords needed validation and if parent information should be included in the first release. The model was continuously subject to change until the very end of release 1 and thus some changes may have not been fully implemented. Required changes are to be added in release 2.

**Controller:**

|  |  |
| --- | --- |
| **Support_Files/Controller_1.png**  **Figure 4. User Controller part 1** | **Support_Files/Controller_2.png**  **Figure 5. User Controller part 2**  **Support_Files/Controller_3.png**  **Figure 6. User Controller part 3** |

The user controller handles the creation, viewing, updating and deletion of the users in the website. Two functions are used to display the users of the website (one for students and another for students) as seen in Figure 4. Originally it was planned for the index teacher to only return users that are teachers however after some difficulty implementing this, the index\_teacher function returns all users to the page and then the view separates the teachers from all other users. This is an inefficient solution that should be changed in future sprints. The create function as seen in Figure 4, creates the user with the inputted form parameters and adds them to the database. On successful save to the database, a flash card is displayed to the user. Update works in a similar fashion.

The biggest design decision for the controllers was the choice of mas assignable user parameters (ass seen in Figure 6). This decision was based on what the client team believed the users should have the ability to assign to themselves. Originally, teachers were unable to assign themselves as teachers and it was up to the admin to go through and promote the teacher from a student to a teacher. This was later changed as reflected above and now anyone could elect themselves as a teacher. Future releases will fix this by sending a confirmation email to the admins on attempted teacher registration.