|  |  |
| --- | --- |
| **Name** | **1.3.3.3.4 RateMyProfessor Data Flow View** |
| **Purpose** | Illustrate the data flow between RateMyProfessor and our own database |
| **Description** | The federator is getting a request from the logic, which then sends it to the proxy manager. The proxy manager then determines where to get the data from, which could either be our own proxy database or from RateMyProfessor itself. The data is then either brought back to the federator or it’s brought to our proxy database. |
| **Requirements** | Requirement 11 |
| **Elements** | **1.3.1 Federator:** Converts requests from the logic to SQL and returns a data response back to the logic |
| **1.3.3.4 RateMyProfessor Proxy:** Handles everything related to RateMyProfessor |
| **1.3.3.4.1 Proxy Manager:** Depending on the request, either sends it to the proxy storage or to the external storage |
| **1.3.3.4.2 RateMyProfessor Proxy Storage:** Stores relevant data from RateMyProfessor so we don’t have to send a request to the external source every single time |
| **1.3.3.4.4 External Request Manager:** When we do need to talk to RateMyProfessor externally, this takes the request from the proxy manager and sends it to the database |
| **1.3.7 Rate My Professor:** Keeps track of the ratings for each professor |
| **Request:** Exactly what it sounds like: a request for certain data |
|  | **Route:** A route to the requested data |
| **Referenced By** | 1.3.3.3 |
| **Viewpoint** | Data Flow Diagram |