

COMS W4111 - Intro. To Databases:

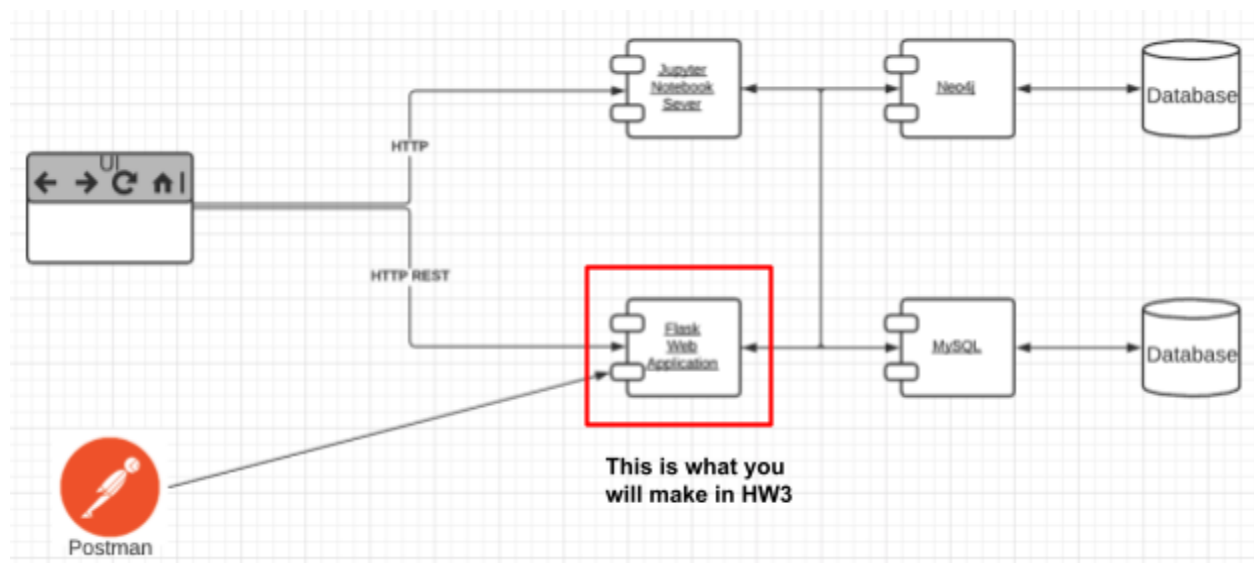
Homework 4 - Programming Track

Introduction

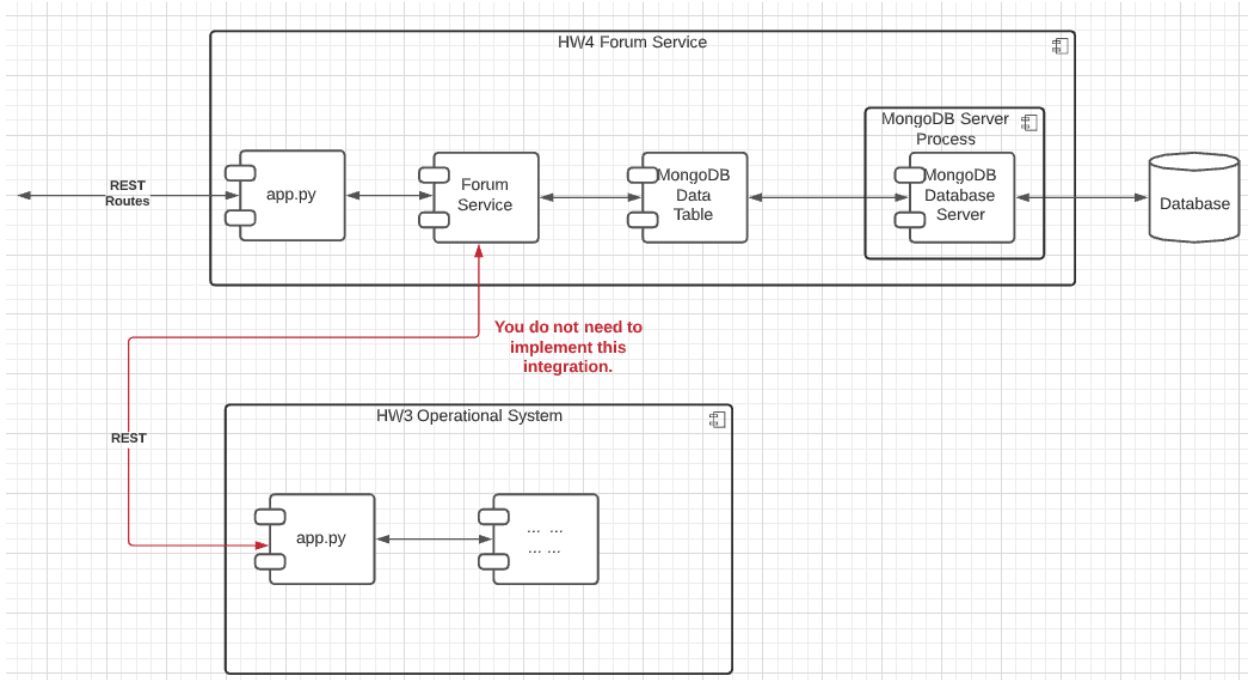
Following the concepts that [Lecture 8](#) discussed, there are two major subsystems in homework 3 and 4.

1. The *Operational System* is a web application that allows the GoT characters to define teams, leagues, pick players, etc.
2. The *Analysis System* performs data engineering and analysis to determine which players to pick when choosing players for teams.

Students on the Programming Track implemented elements of the *Operational System* focusing on resources backed by services with data in MySQL and Neo4j. The basic structure of the HW3 systems is:



The programming track for HW4 adds support for a *Forum and Discussion Service* that maintains the information in MongoDB. For simplicity, we are not going to integrate the *ForumService* with the other services, and we will develop and test in isolation. The following is an overview of the structure:



Installation and Setup

In preparation for HW4, we will release tonight or tomorrow morning, please install:

1. MongoDB: <https://docs.mongodb.com/manual/installation/>
2. Compass: <https://docs.mongodb.com/compass/current/install/>

MongoDB is a database management system, analogous to MySQL and Neo4j. Compass is a development and admin tool analogous to DataGrip.

There is a starter project that can help you with your implementation. Please clone the project. The link is: <https://github.com/donald-f-ferguson/w4111-s21-forum.git>

The project's files contain code to help get started:

- `app.py` is the REST Flask application that you will use to implement routes.
- `Services/DataServices/MongoDBTable.py` provides functions that access MongoDB databases and collections. *You may have to add additional methods to complete the assignment.*
- `Services/ForumService` is the file where you will implement methods that `app.py` calls to implement routes. You will have to add methods to this file.
- `unit_tests/t_forums.py` contains some functions you can use to become familiar with the code and test your installation and setup.

- *dump/new_forum* contains files you can use to load sample data using [mongorestore](#).
 - Please note that if you are on Windows you will need to download mongorestore separately. Follow the instructions to do so and be sure to restart after updating the Path environment variable.

Implementation Tasks

You must implement the following paths:

1. */api/forums*:
 - a. POST: Creates a new post in the forum.
 - b. GET: Retrieves posts matching a query string.
2. */api/forums/<post_id>*:
 - a. GET: Returns a post with the specified *post_id*
3. */api/forums/<post_id>/comments*
 - a. GET: Returns the comments relative to the post matching a query string.
 - b. POST: Creates a new comment on the post.
4. */api/forums/<post_id>/comments/<comment_id>*:
 - a. GET: Returns the specified comment.

The query processing for GET on */api/forums* must support the following query complexities:

- hosts have a *tags* property, which is a list. The query string parameter “tags=xxx” matches a post if any one of the tags values is xxx.
- The query string parameters *children.email* matches posts for which any comment’s author’s email is the specified value.

Submission Format and Instructions

This homework is due **Monday, April 15th @ 11:59pm ET**.

1. PDF Submission: Create a Jupyter notebook testing each path & method listed above like in HW3. In the notebook include your screenshots of Postman/MongoDB showing that your tests were successful. Export this Jupyter notebook to PDF and submit to Gradescope under the appropriate assignment.
2. ZIP Submission: Submit a zip file of **the files you changed**, the screenshots you took, and the jupyter notebook you created. **You will lose points for extra files and hidden files so please make sure to remove them** when uploading to Gradescope. Even if they are hidden on your Mac, Gradescope will give you the opportunity to remove them from your submission.