Table of Contents

CRC Cards

PHP Components:

Class name: Medicine

Parent class (if any): None

Class name Subclasses (if any): None?

Responsibilities:

- Outputs data for medicine
- Manages requests relating to the medicine it represents.
- Passes request to Search class
- Returns answer to view
- Pulls relevant data from database
- Sends signal to Database class to update SQL database

Collaborators:

- Search
- Database

Class name: Database

Parent class (if any): None

Class name Subclasses (if any): None?

Responsibilities:

- Loads and stores data from SQL database
- Updates the SQL database whenever required
- Provides data to Medicine class as required

Collaborators:

- Medicine

Python Components:

Class name: Search

Parent class (if any): None

Class name Subclasses (if any): None?

Responsibilities:

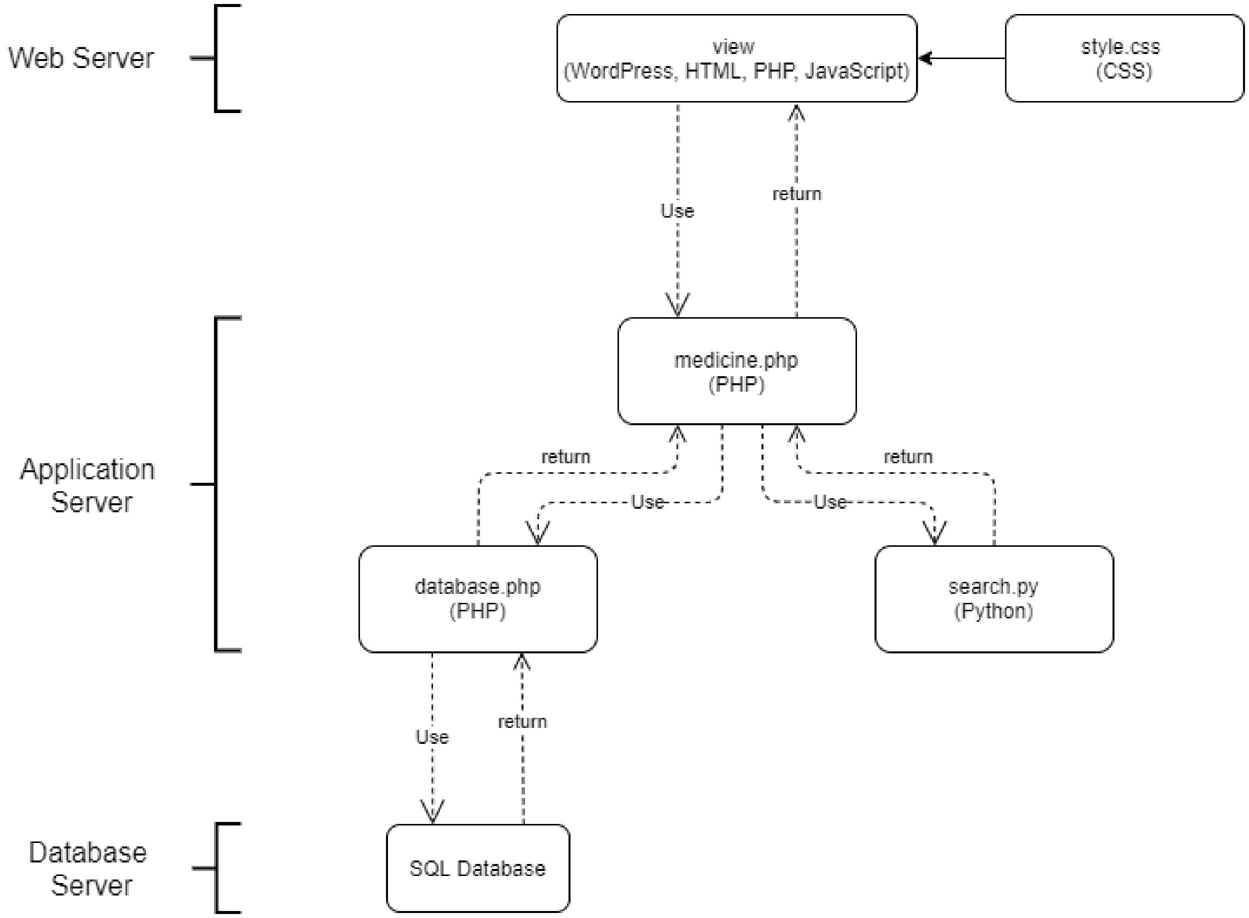
- Receives request from the Medicine class
- Breaks request into keywords
- Finds appropriate answer to request according to keywords
- Stores most frequent and recent request keywords and their corresponding answers
- Returns answer to Medicine class
- Loads data from monograph sheet

Collaborators:

- Medicine class

System Architecture:

3 Tiered Architecture:



Architecture Details:

Web Server:

This is the tier of the architecture that works directly with the user. It is what the user sees and interacts with when on the webpage. It will be done via a combination of WordPress, Javascript, PHP, HTML and CSS. Any computations/change of states do not occur here, however the results of said computations/changes of state are passed to the view, to be displayed to the user. Any request by the user is received by the user and is then passed to medicine.php (which is part of the 'Application Server').

Application Server:

This tier is where most of the backend work will go. It focuses on receiving requests from the view and providing an appropriate response. 'medecine.php' is where the view will receive the relevant data to the medicine currently being displayed. It is in charge of managing all requests pertaining to the medicine. It passes the request received from view to 'search.php' in a desirable format. Furthermore, it is in charge of issuing orders to the database (i.e. what data should be loaded, and updated – only if absolutely necessary). 'database.php' is directly in charge of acquiring the data required from the SQL database and passing it back to 'medicine.php' in the desired format. Finally, 'search.php' is where the request is analysed for keywords, and then said keywords are used to search the monograph papers and return a desirable answer.

Database Server:

In this tier, you'll find all of the actual raw data relating to the medicines covered by our project. They will all be stored in a SQL database, and will only have direct communication with 'database.php' which is in the Application Server. There is never any communication between the Web Server and the Database Server.

About the 3-Tiered Architecture:

Initially we were considering MVC, however we decided that the linear shape of the 3-Tiered Architecture is more suitable for our search engine (we also want to ensure the view never communicates with the databases). Information relating the architecture we are using can be found at https://www.ibm.com/cloud/learn/three-tier-architecture.