**Task List Documentation**

**Pavan Policherla**

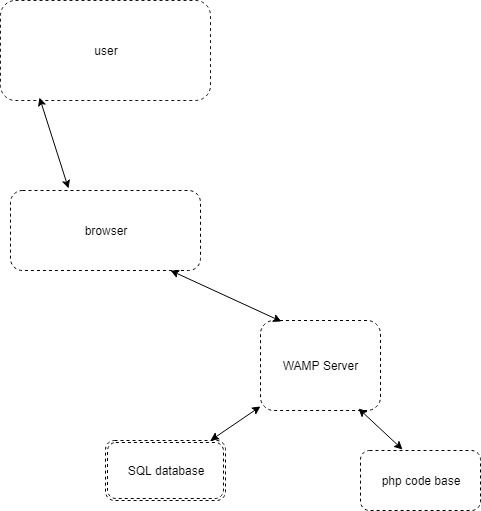
**Functional Requirements:**

* Add user input into the database table
* Pull the data from the database for the user to view
* Remove the data from the database at the user’s request.

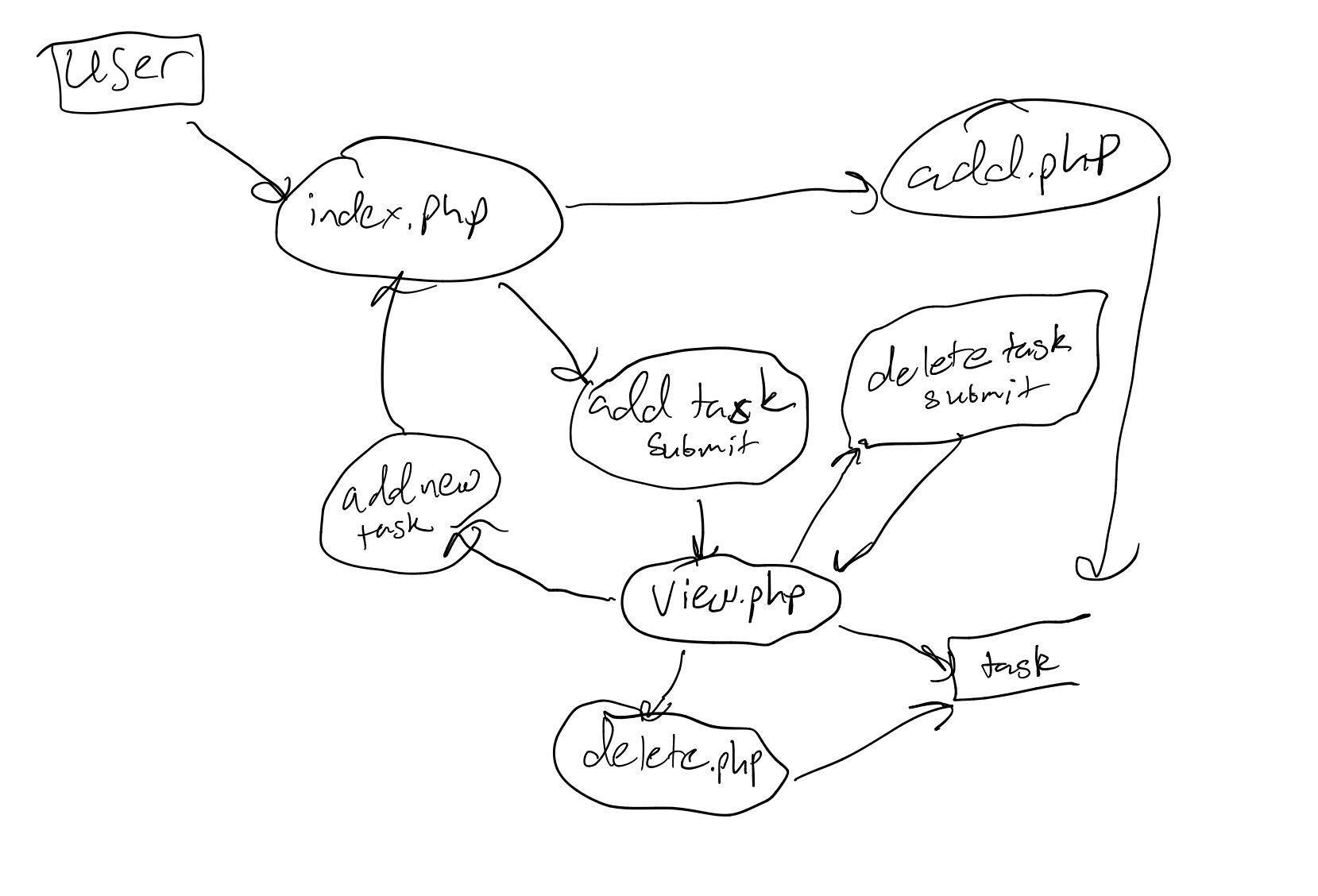
**Non-functional Requirements:**

* Updates fast
* Capable of storing over 100 tasks
* Can work with all date formats.

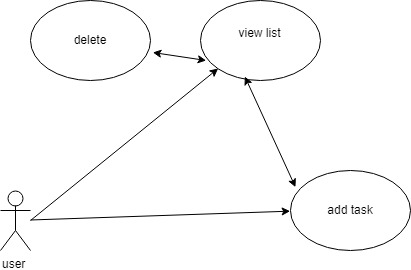
**System Architecture Diagram**

****

**Data Flow Diagram**

****

**Use cases**

****

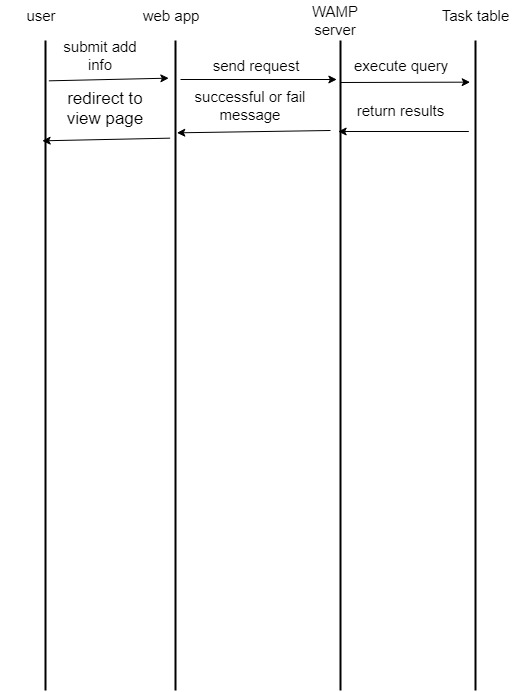
Use case 1: the add task use case is the default page where the user is asked to input the name of a task and the date that the task will happen. The user will then be redirected to the view page automatically where they can view all the tasks in the list.

Use case 2: The view button is also on the default page where the user can click the view button and they will be taken to the view page where they are able to look at the tasks that need to be completed.

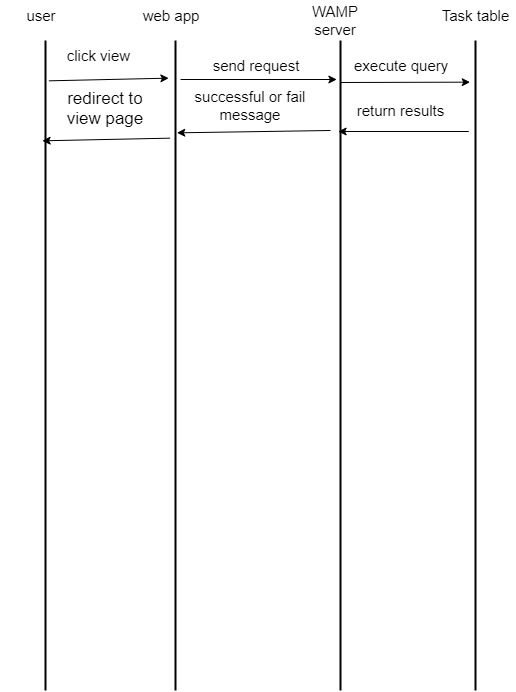
Use case 3: the user while on the view page will be able to do book keeping and delete any tasks that have been completed or no longer need to be tracked.

**Sequence Diagrams**

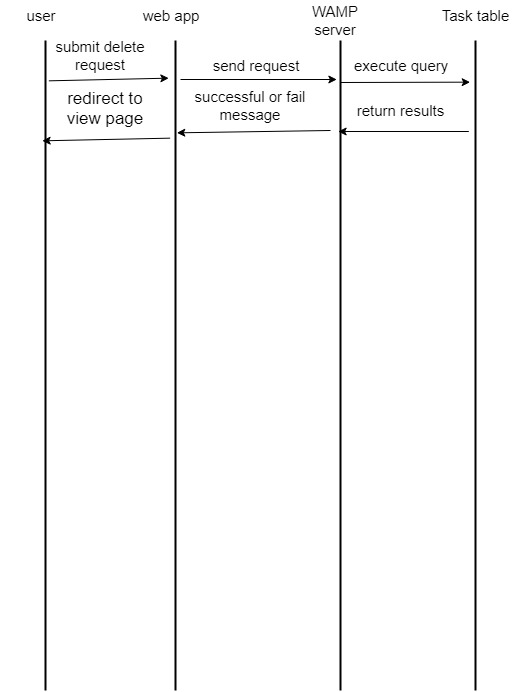
Add task:



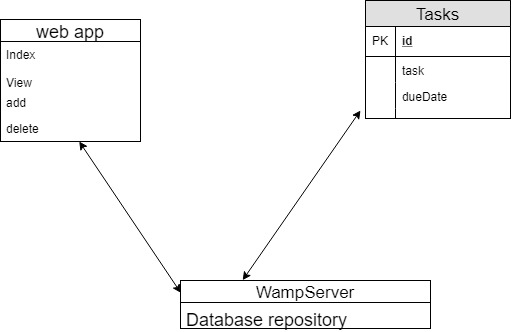
View list:



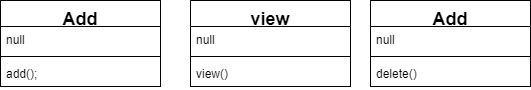
Delete task:



**Database Design**

****

**Class Diagram**

****

**Test Cases:**

Test case 1: test the add operation by adding a new task to the list on the index page.

Test case 2: test the view function by either submitting a task to the database from the index page or by clicking the view button on the index page.

Test case 3: test the delete operation by viewing the list and checking the delete box on any of the tasks and then submitting. First test by deleting a single task then by deleting multiple at the same time.

**Work Break Down:**

Name of the task: build a simple server based web app that allows a user to keep track of tasks and stores them on a table in a database.

Estimated time spent:

Learning new language and software: 6 hours

Creating app: 12 hours

Documentation: 2 hours

Actual time spent:

Learning new language and technologies: 12 hours

Creating app: 20 hours

Documentation: 5 hours

Notes and issues:

This project seemed simple at first but what I underestimated what how vast php really was which caused me to greatly underestimate the time it would take me to learn what I needed to know to get the project up and running. Another issue I ran into was working with html, I have never worked with html, php, or databases before so it was a challenge learning the formatting and communications between the files and the database. Having WAMP to help me a long greatly expedited the whole process with its extensive database and SQL functionality making the creation, linking and checking of the database while building and testing the app very easy and straightforward. Despite hickups and some headaches this was an enjoyable project.