To Do List Web Application

Nick Napoletano

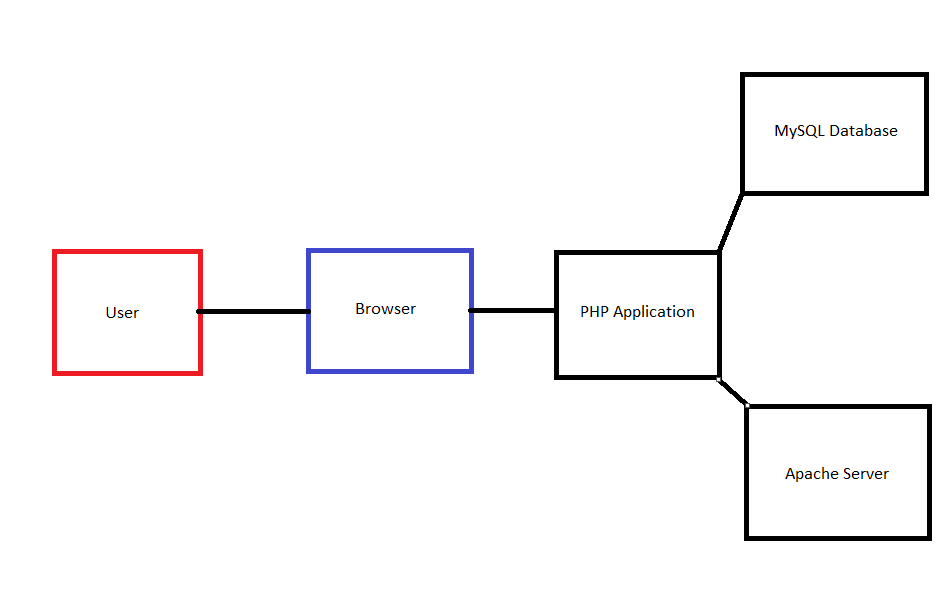
Functional requirements:

1. Add new items to the list
2. View Items
3. Remove Items
4. Must use WAMP stack (on windows)

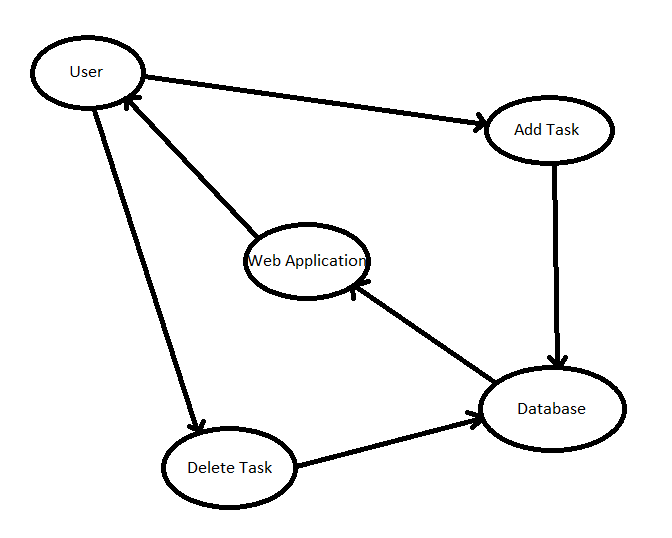
Non-Functional Requirements:

1. Easily Testable
2. Simple in UI
3. Code easy to understand

System Architecture:



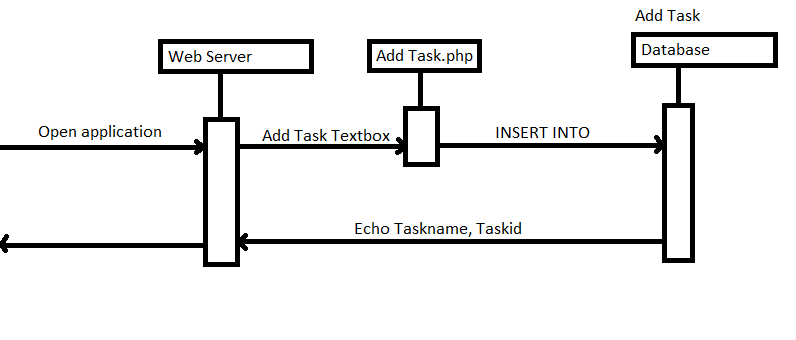
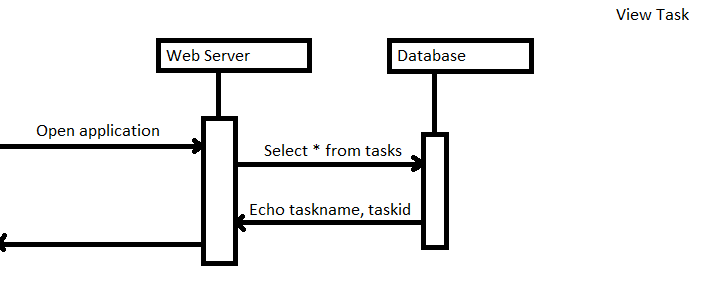
Data flow diagram:

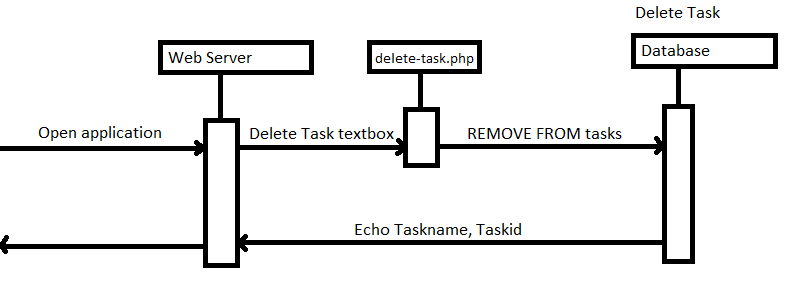


Use case:

1. User wants to add a task.
   1. The user opens the application, and types in the description of the task in the “Add” text box. He then hits submit and the task is added.
2. User wants to view the tasks
   1. The user opens the application. The tasks are right there for viewing.
3. User wants to remove a task.
   1. The user opens the application. He finds the taskID number corresponding to the task he wants to remove. The user types that number into the “Delete” textbox and submits. The task is removed.

Sequence diagrams





Database Table

|  |  |
| --- | --- |
| TaskID (primary key)(autoincrement) | Task(varchar 1000) |
| 1 | Wash the car |

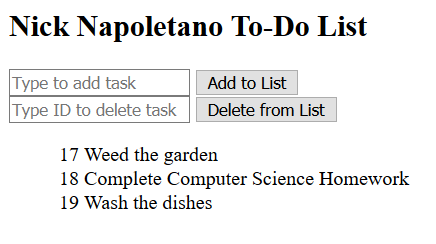
Class diagram

N/A, as I didn’t use any classes or functions.

Test Cases

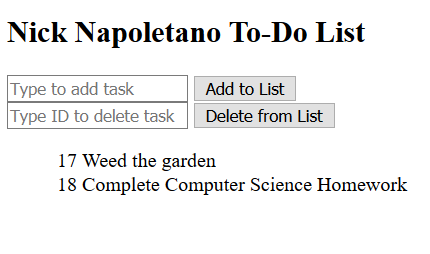
1. Add a task.
   1. This test will be testing the functionality of the “add task” textbox.
   2. I will input “Wash the dishes” as an input into the textbox, and will expect the output to be the task list with the added task “Wash the dishes”

Outcome: Success



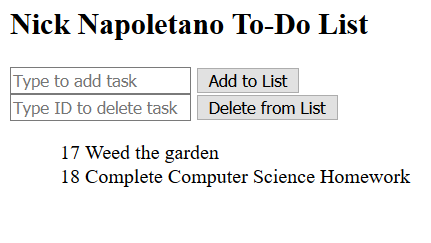
1. Remove a task.
   1. This test will be testing the functionality of the “delete task” textbox.
   2. I will input “19” as an input into the delete textbox, and will expect the output to be the task list without the task “Wash the dishes”

Outcome: Success



1. Remove a task that doesn’t exist.
   1. This test will be testing the robustness of the “delete task” textbox.
   2. I will input “19” as an input into the delete textbox, and will expect the output to be the same table as before without any errors.

Outcome: Success



Time spent per task

Name: Install software and configure Apache

Estimated time: 1 hour

Actual time: 1 hour

Notes: Used Wampserver. Configuring apache took most of this time, since ive used phpmyadmin before

Name: View Task

Estimated time: 2 hours

Actual time: 1 hour

Notes:

This was the first and easiest thing to do. Its just a SELECT with a loop to show the database.

Name: Add Task

Estimated time: 3 hours

Actual time: 5 hours

Notes: this took the most time because the methods I found online used outdated code. I ended up using snippets of code I used in my SQL class in the winter semester to make a working textbox.

Name: Remove Task

Estimated time: 3 hours

Actual time: 4 hours

Notes: This took longer because I was attempting to use images as a button to delete the corresponding task, but nothing I tried worked. So I instead used the same method as the Add function but with the TaskID so the user doesn’t have to type out an entire task when they delete something.