Todo Web Application Design Documentation

Created by:

Ryan Cibasek

Problem Statement:
To make a small web base application that can be used as a simple todo list. It should be able to add tasks, view tasks, and delete tasks.

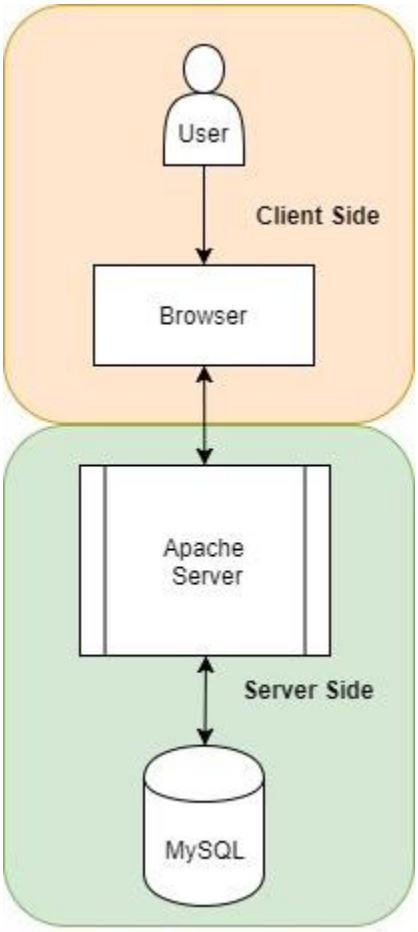
Functional Requirements:

- This web application shall be able to add new tasks to a database.
- This web application shall be able to view all tasks that are stored in a database.
- This web application shall be able to delete tasks from the database.

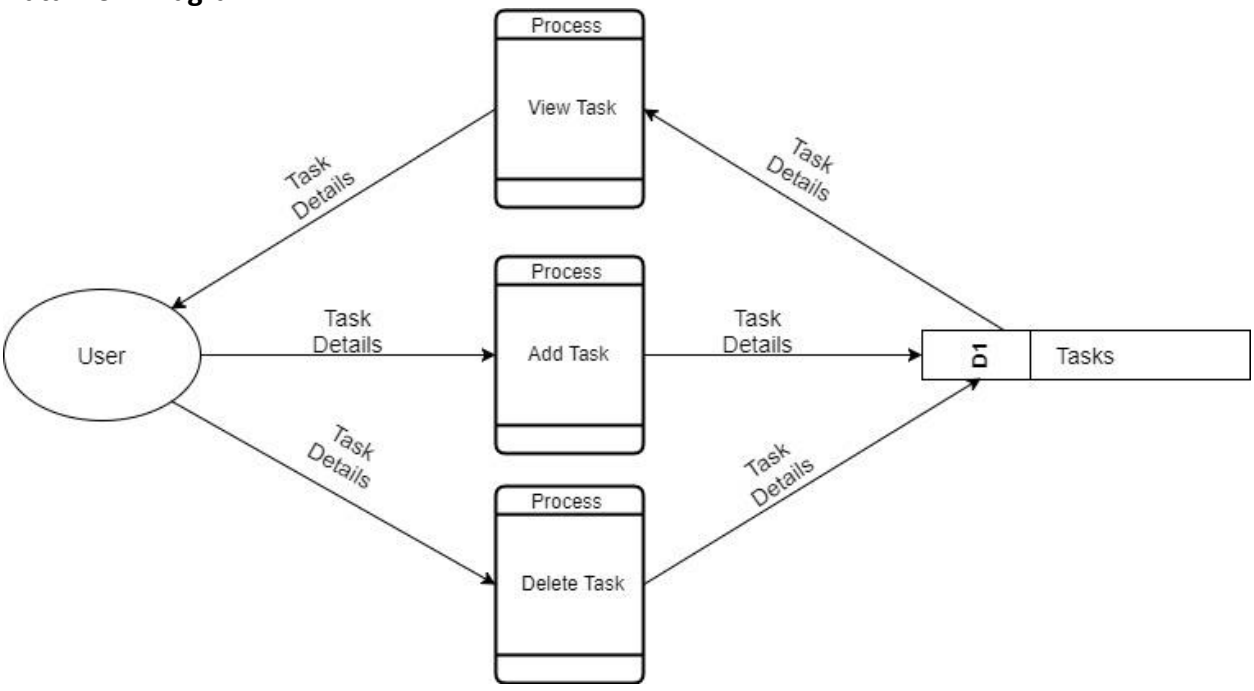
Non-Function Requirements:

- This web application shall instantly display confirmation of adding a task, then redirect to the main page in two seconds.
- This web application shall instantly display confirmation of deleting a task, then redirect to the main page in two seconds.
- This web application shall display all tasks within five seconds.

System Architecture Diagram:



Data Flow Diagram:



Use Cases:

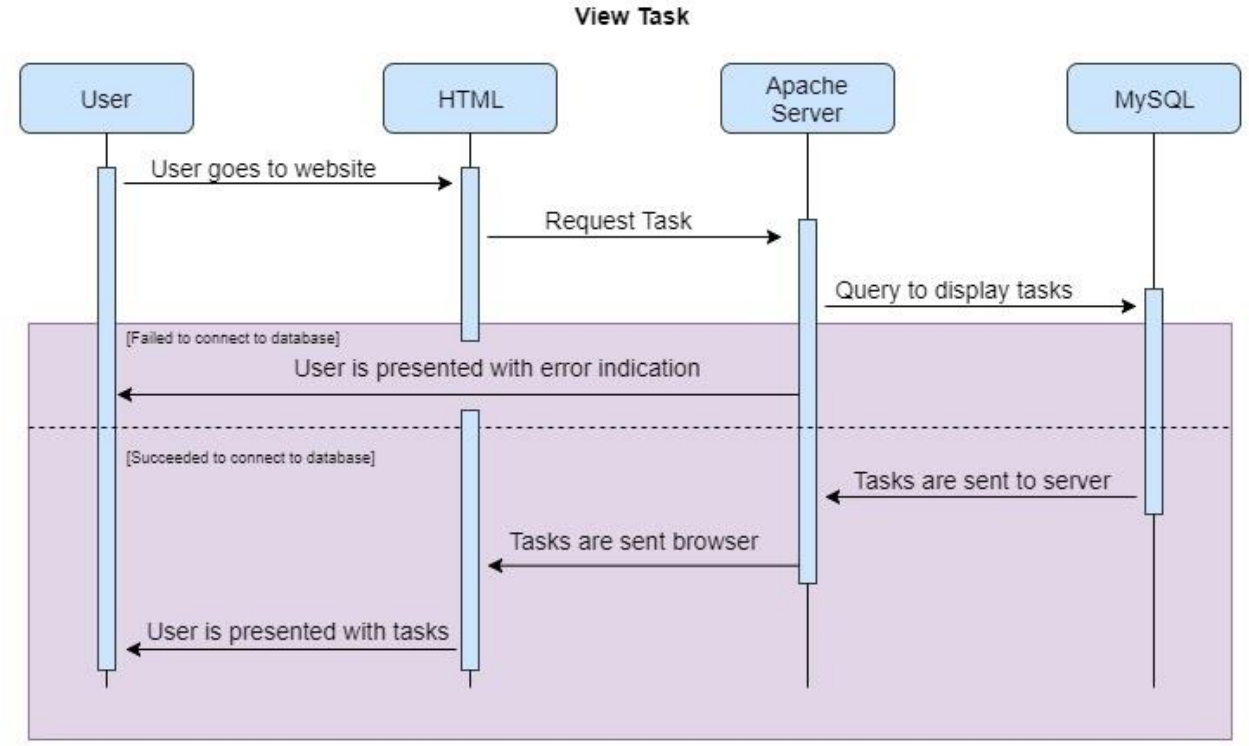
Use Case ID:	UC1
Use Case Name:	View Tasks
Actors:	User
Description:	This is the normal path that a user will go through to see their todo list with all of the tasks they have added.
Preconditions:	A user has a browser that is connected to the internet.
Trigger:	Navigating their browser to the index page of the todo web application.
Basic Flow:	1. User opens browser. 2. User navigates to the todo application page. 3. The user then will be given all the tasks that are currently in the database.
Alternate Flow:	1. User opens browser. 2. User navigates to the todo application page. 3. A error is displayed indicating that connection to database could not be made.
Postcondition:	A browser displaying all tasks that are stored within the database.

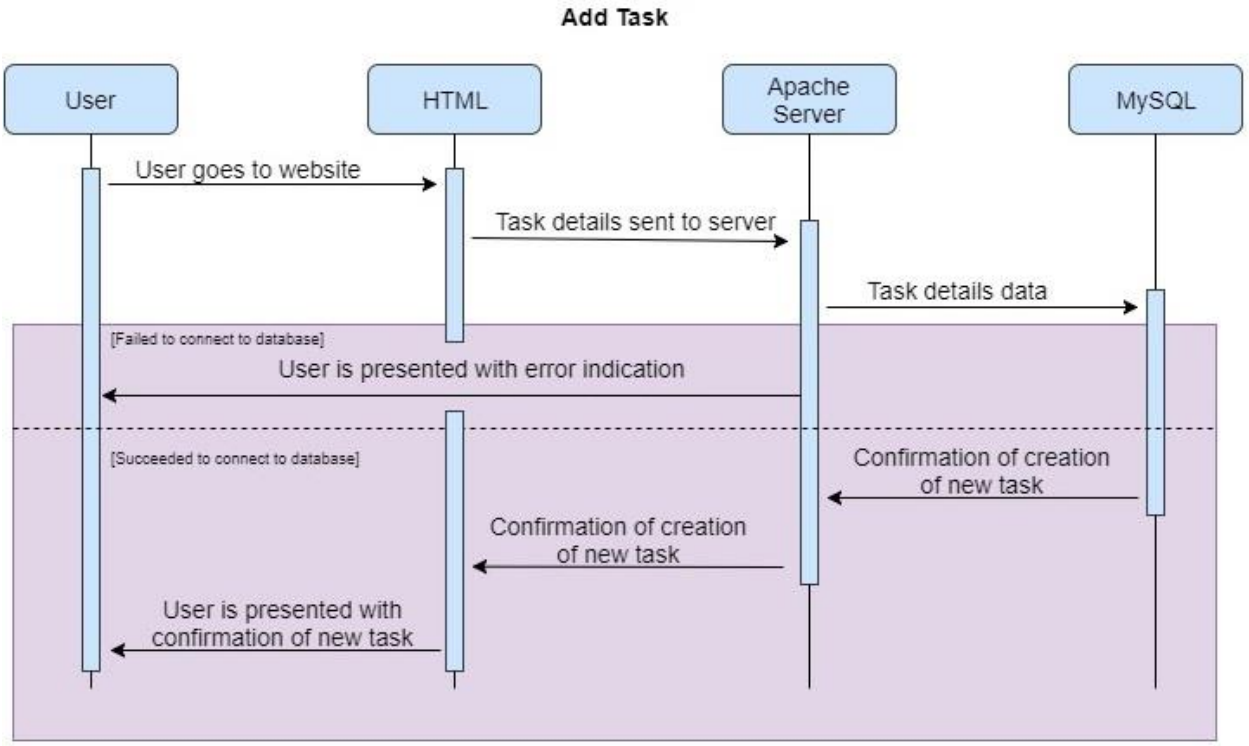
Use Case ID:	UC2
Use Case Name:	Add Task
Actors:	User
Description:	This is the normal path that a user will go through to add a task.
Preconditions:	A user has a browser that is connected to the internet.
Trigger:	Clicking the button “Submit”.
Basic Flow:	1. User opens browser. 2. User navigates to the todo application page. 3. The user then clicks “Add Item” button. 4. The user fills in the fields, then clicks “Submit”.
Alternate Flow:	1. User opens browser.

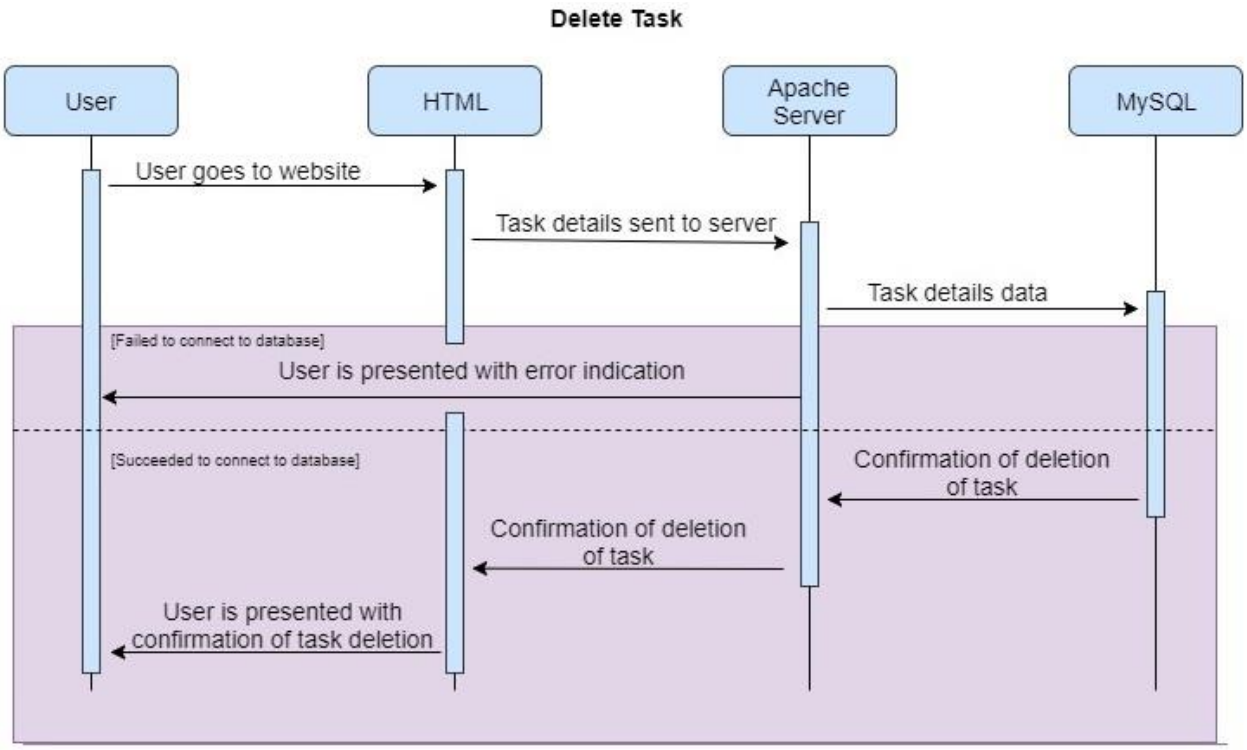
	2. User navigates to the todo application page. 3. The user then clicks “Add Item” button. 4. The user fills in the fields, then clicks “Submit”. 5. An indication that posting the data to the database failed.
Postcondition:	A confirmation message indicating successful post to the database.

Use Case ID:	UC3
Use Case Name:	Delete Task
Actors:	User
Description:	This is the normal path that a user will go through to delete a task.
Preconditions:	A user has a browser that is connected to the internet.
Trigger:	Clicking the button “Delete”.
Basic Flow:	1. User opens browser. 2. User navigates to the todo application page. 3. The user then clicks “Delete” button.
Alternate Flow:	1. User opens browser. 2. User navigates to the todo application page. 3. The user then clicks “Delete” button. 4. An indication that deletion of the data to the database failed.
Postcondition:	A confirmation message indicating deletion of the task from the database.

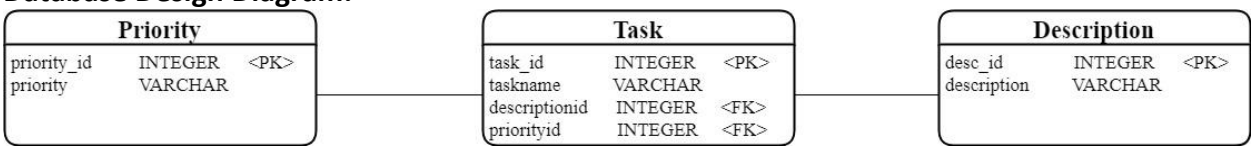
Sequence Diagram:



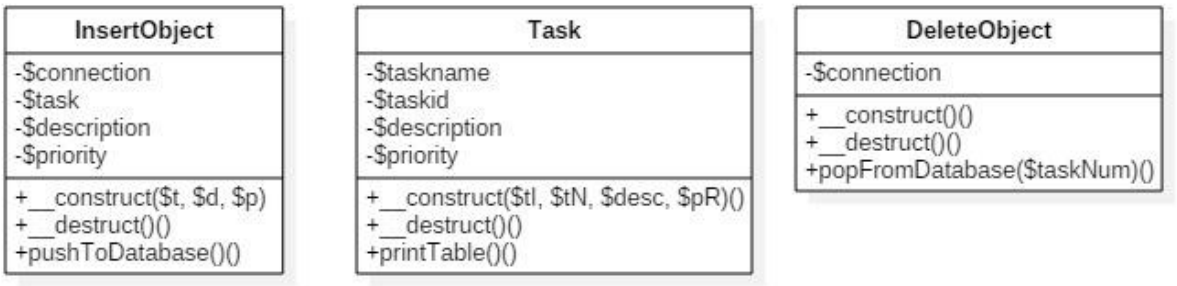




Database Design Diagram:



Class Diagram:



Test Case:

Test Case ID:	TC1
Test Name:	View Tasks
Tested Date:	9/4/17
Objective:	Verify that tasks that were added by user will all display in the landing page.
Preconditions:	Have multiple tasks saved in the data base.
Test Steps:	1. Navigate to landing page.

	2. Compare tasks displayed on landing page with tasks that exist in the database.
Expected Result:	The data displayed on the landing page should match exactly what is entered into the databases.
Actual Results:	The data matched what was in the databases.

Test Case ID:	TC2
Test Name:	Add Tasks
Tested Date:	9/3/17
Objective:	Verify that a user can add tasks.
Preconditions:	Must have all the databases setup and the webpage coded.
Test Steps:	1. Navigate to landing page. 2. Click “Add Item” button, filling in the post information. 3. Click “Submit” button.
Expected Result:	The data displayed entered and submitted to the database should match.
Actual Results:	The data matched what was in the database.

Test Case ID:	TC3
Test Name:	Delete Tasks
Tested Date:	9/3/17
Objective:	Verify that task that was deleted by a user will be removed from the databases.
Preconditions:	Must have all the databases setup and the webpage coded.
Test Steps:	1. Navigate to landing page. 2. Click “Delete” button.
Expected Result:	The task that was deleted should no longer be present in the connected database.
Actual Results:	All traces of the task have been removed from the database.