**Use Case 1**

**Use Case Name :** Access Application

**Goal:** bring up to do list

**Precondition:** XAMPP server and MySQL database are ready

**Postcondition:** User accesses application and is able to use it

**Actors:** User

**Triggering Event:** User enters http://localhost:1234/scripts/index.php

**Description:** User enters above site into any web browser, if the XAMPP server and database are running, it should bring user to page. User may need to alter XAMPP ports in order to find free ones.

**Alternative:** XAMPP server is not running, user will get an error to retry.

**Use Case 2**

**Use Case Name:** Enter task to be stored into database

**Goal:** Store a task into the database to be displayed with task name, due date, and status of task(due, past due, pending).

**Precondition:** User has application up and running, and is on field 1 to enter task name, then field 2 to enter due date.

**Postcondition:** Task name and due date are entered into the database and displayed for user.

**Actors:** User

**Triggering Event:** When user clicks add

**Description:** Task name, due date, and status are entered into the database and stored. They are then displayed as an unordered list for the user to see when they run the application.

**Alternative:** If the user does not enter a task name or due date they are unable to store them and are asked to enter one of the above. These are required fields.

**Use Case 3**

**Use Case Name:** Delete task from database

**Goal:** Remove specified task (by task name) from the database, and remove it from the unordered list displayed to user.

**Precondition:** Enter the correct task name in the appropriate field.

**Postcondition:** Task entered by user is removed from the database and the display.

**Actors:** User

**Triggering Event:** When the user clicks on the delete button after having entered an appropriate task in the appropriate field.

**Description:** The user will use the task names displayed to them to select a task they no longer want on the to do list and enter it into the field to remove it from the database.

**Alternative:** If the user does not enter a task, the application will ask the user to enter one when they click the delete button, as this is a required field.

**Use Case 4**

**Use Case Name:** Delete all tasks

**Goal:** Delete all tasks from database

**Precondition:** User clicks delete all button

**Postcondition:** All tasks are removed from database

**Actors:** User

**Triggering Event:** User clicks delete all button

**Description:** All tasks are removed from the database when use clicks delete all button

**Alternative:** No alternative, should implement a safeguard against unwanted deletion

**Use Case 5**

**Use Case Name:** Show only selected tasks

**Goal:** Show the tasks the user wants to see based on status

**Precondition:** user enters appropriate status and clicks on button

**Postcondition:** User is taken to a page to show only the tasks that have that status

**Actors:** User

**Triggering Event:** User enters appropriate task and hits appropriate button

**Description:** This will show only the tasks that the user has entered based on their status.

**Alternative:** If user does not enter a task it will ask user to enter a task, as this is required.

**Use Case 6**

**Use Case Name:** Return button

**Goal:** Return user to index.php

**Precondition:** User has selected to sort based on task status and is on separate page

**Postcondition:** User is returned to index page

**Actors:** User

**Triggering Event:** User hits return button

**Description:** User is returned to index page after seeing sorted tasks.

**Alternative:** No alternative