**TODO App Documentation**

James Demery

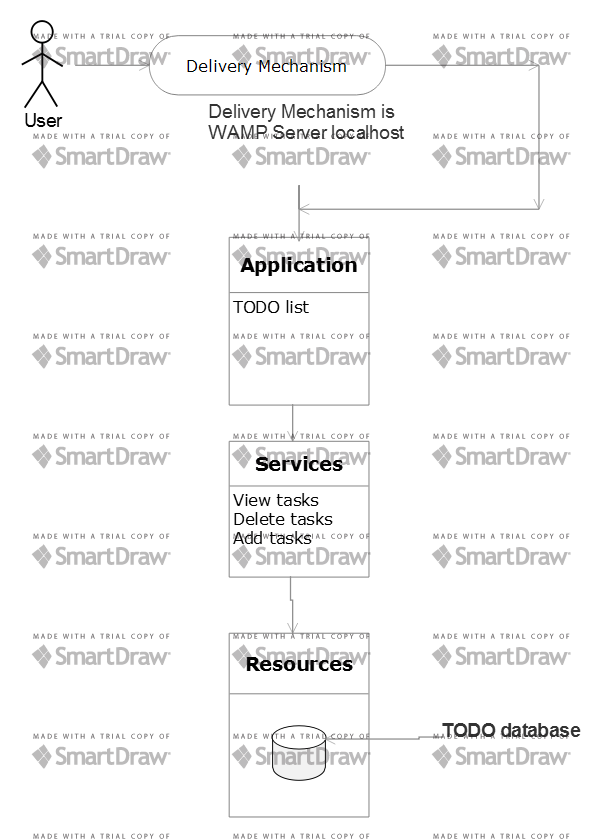
**Functional Requirements**

1. The user shall be able to view the total number of tasks in the system, as well as how many of each status type (pending, started, late, completed) there are.
2. The user shall be able to add a new task to the todo list, including a name for the task, its status, and its due date.
3. The user shall be able to view all the tasks in the system and be able to delete any of them that they choose to.
4. If the user clicks on one of the status counts (such as completed) they will be taken to a new page which lists all of those tasks, whichever category they may be.
5. When the user adds or deletes a task that is reflected in the database used for this app titled “todo”.
6. If the user enters a task that does not fall under one of the four pre-defined statuses it is filtered into an “Other” section that is also displayed with the other four statuses.

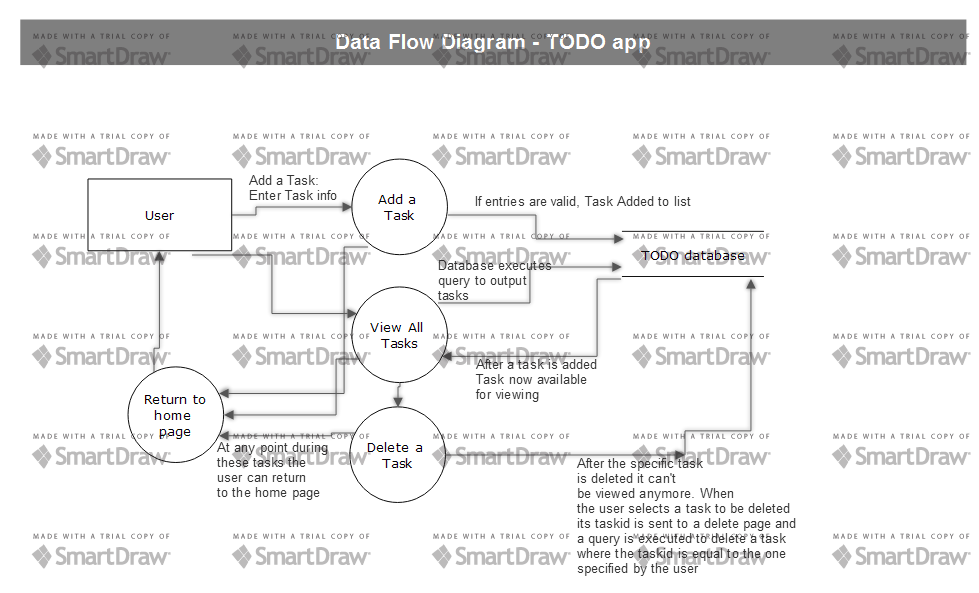
**Non-Functional Requirements**

1. Privacy Requirements: Users will not want other people to be able to see or manipulate their tasks. Specifically when using the POST (for adding a task) or GET (for deleting a task) methods to pass a value the inputs need to be “sanitized” to prevent SQL Injection attacks.
2. Usability: Because it is such a straightforward and simple app the usability is fairly high. The app is pretty intuitive to use, there’s a link to add a task and the user is then prompted to enter the three necessary values or cancel and return to the homepage. The user can then view all tasks and in the right most entry or each row there is a delete button that if the user presses they are redirected back to homepage and the selected task was deleted.
3. Robustness: I am currently testing the app and trying to make it more robust. One way of doing that was if a user entered a task that was not one of the four pre-defined statuses it filters it to an “Other” section so that the tasks existence is still reflected in the app. Also trying to achieve some modicum of attractiveness for the app so it looks like something people may actually use, not just plaintext and blue, underlined hyperlinks.
4. Cost: As long as the user already has a PC the cost should be $0. Downloading WAMP Server (the tech stack for this app) is free, and of course using the application is free as well. There is no cost of installing the app or deleting it besides a small amount of the users time.
5. Reliability: Thus far the app has passed all the tests I have put it through. Take that with a small grain of salt though as I am inexperienced in robustly testing things (something I need to improve on and fast). However, all the required functionalities are performing properly.
6. Modifiability: The app is easily modifiable, I am hoping if I have enough time I can add a couple of functionalities that were not required, but that I believe would make it a more robust app. Plus all the source code is publicly available on my github repository so really anyone could study the code and make additions/changes if they so wished.

**System Architecture Design**



**Data Flow Diagram**



**Use Cases**

I based the format of the text part of these use cases from these websites: <ftp://ftp.software.ibm.com/software/rational/web/whitepapers/RAW14023-USEN-00.pdf>

James Heumann

<https://blog.testlodge.com/how-to-write-test-cases-for-software-with-sample/>

Jake Bartlett

1. **Title:** Add a Task

**Summary:** The user is able to add a task to their todo list.

**Rationale:** Unfortunately, we have to add tasks to our todo lists not just cross them off or delete them. Users have to be able to add more tasks or the app is pointless.

**Users:** All users

**Primary Actor:** User

**Secondary Actor:** TODO app

**Preconditions:** WAMP Server installed. Navigate to localhost within the WAMP Server, go to localhost/todo/index.php to get to the homepage.

**Basic Events:**

1. User clicks the Add a Task button right below the header
2. The user fills out the form, all fields are required, and presses submit or enter to add the task to the todo list.
3. The user can also at any time while filling out the form press the Home/Cancel button to return to the homepage without adding a task.
4. If the user chooses to fill out all 3 fields and click submit, they will be redirected to the homepage and the task will have been added to the overall count, as well as filtered to its status count. The user could then see all the details of the task by clicking View/Delete tasks.
5. **Title:** View Tasks

**Summary:** User can view all tasks on the todo list.

**Rationale:** So users can see what tasks are on their todo list.

**Users:** All users

**Primary Actor:** User

**Secondary Actor:** TODO app

**Preconditions:** WAMP Server installed. Navigate to localhost within the WAMP Server, go to localhost/todo/index.php to get to the homepage.

**Basic Events:**

1. User clicks button View/Delete Tasks
2. If there are any tasks in the database they are displayed in a table that has the task name, status, and due date. There is also a delete button in the rightmost column of each table. Pressing this button will delete that specific task and take the user back to the homepage.
3. If there are no tasks in the database then there is a message saying that on the page and a button to return home.
4. If there are tasks in the database there will be a Delete All button below the table. The user can press that button to delete all tasks in the database.
5. **Title:** Delete Tasks

**Summary:** User can delete a task on the todo list.

**Rationale:** User can delete a task that they have completed or if it is dated somehow.

**Users:** All users

**Primary Actor:** User

**Secondary Actor:** TODO app

**Preconditions:** WAMP Server installed. Navigate to localhost within the WAMP Server, go to localhost/todo/index.php to get to the homepage. At least one task entered in the todo list.

**Basic Events:**

1. User presses the button the says View/Delete tasks.
2. The user will then see a table that displays all the tasks currently in the database. In the rightmost column of each table is a Delete button that, if the user clicks it, will delete that specific task and return the user to the home page.
3. The user could also choose to click the Delete All button below the table to delete all tasks on the todo list.
4. If there are no tasks on the todo list then they of course can’t be deleted. A task would have to be added first, instructions of how to do that are above.
5. **Title:** Delete All Tasks

**Summary:** User can delete all tasks on the todo list at once.

**Rationale:** Sometimes you’re really productive and you get all the things on your todo list done, then you can delete them all at once if you so choose.

**Users:** All users

**Primary Actor:** User

**Secondary Actor:** TODO app

**Preconditions:** WAMP Server installed. Navigate to localhost within the WAMP Server, go to localhost/todo/index.php to get to the homepage. At least one task on the todo list.

**Basic Events:**

1. User clicks button View/Delete All tasks
2. Below the table that lists all the tasks and their details there is a button that says Delete All. The user can click that to delete all tasks on the todo list and they will be directed back to the homepage.
3. Could also delete tasks individually or simply return to the homepage by pressing Home.
4. **Title:** View Started Tasks

**Summary:** User can view all started tasks on the todo list by themselves.

**Rationale:** The user doesn’t want to see old tasks that are already done, they want the ones currently in the works.

**Users:** All users

**Primary Actor:** User

**Secondary Actor:** TODO app

**Preconditions:** WAMP Server installed. Navigate to localhost within the WAMP Server, go to localhost/todo/index.php to get to the homepage. At least one task on the todo list with status “Started”.

**Basic Events:**

1. The user clicks the blue number to the right of the sentence “Total Number of Started Tasks in the System: “
2. The page they are taken to will output all the started tasks on the todo list in a table that has the task name, status, and due date and a delete button.
3. The user can return back to the homepage, continue scrutinizing their started tasks, or delete one if they so choose.
4. If there are no started tasks on the todo list then there will be a message saying that.

1. **Title:** View Pending Tasks

**Summary:** User can view all pending tasks on the todo list.

**Rationale:** User wants to see the tasks that might be coming up soon.

**Users:** All users

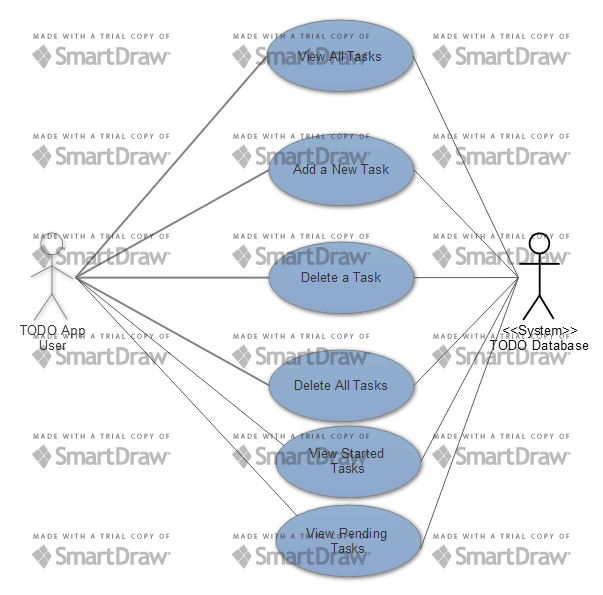
**Primary Actor:** User

**Secondary Actor:** TODO app

**Preconditions:** WAMP Server installed. Navigate to localhost within the WAMP Server, go to localhost/todo/index.php to get to the homepage. At least one task on the todo list with status pending.

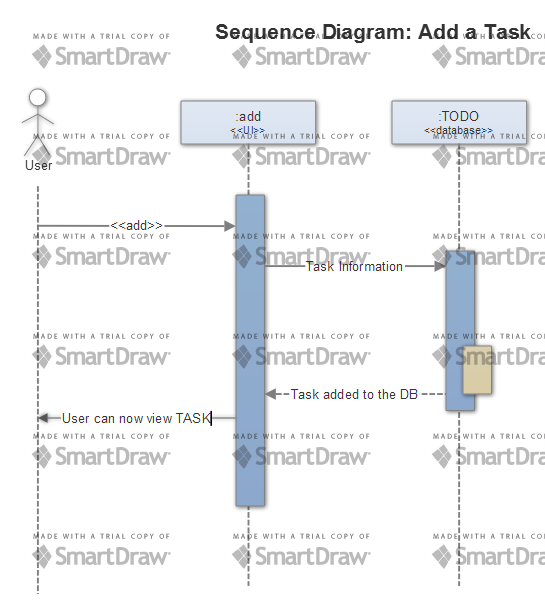
**Basic Events:**

1. The user clicks the blue number to the right of the sentence “Total Number of Pending Tasks in the System: “
2. The page they are taken to will output all the started tasks on the todo list in a table that has the task name, status, and due date and a delete button.
3. The user can return back to the homepage, continue pondering their pending tasks, or delete one if they so choose.
4. If there are no pending tasks on the todo list then there will be a message saying that.

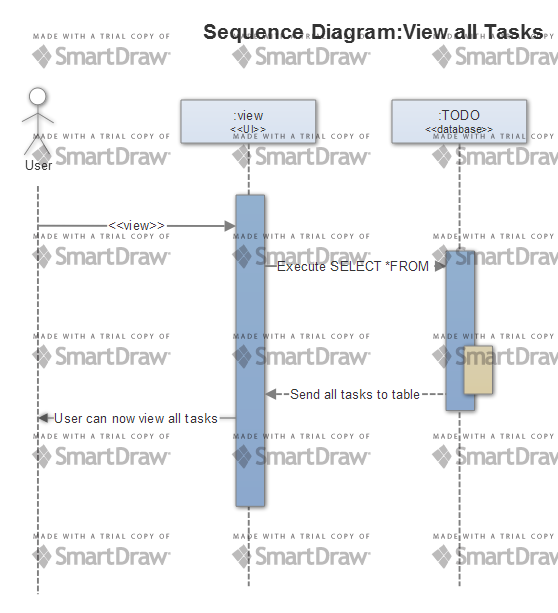


**Sequence Diagrams**

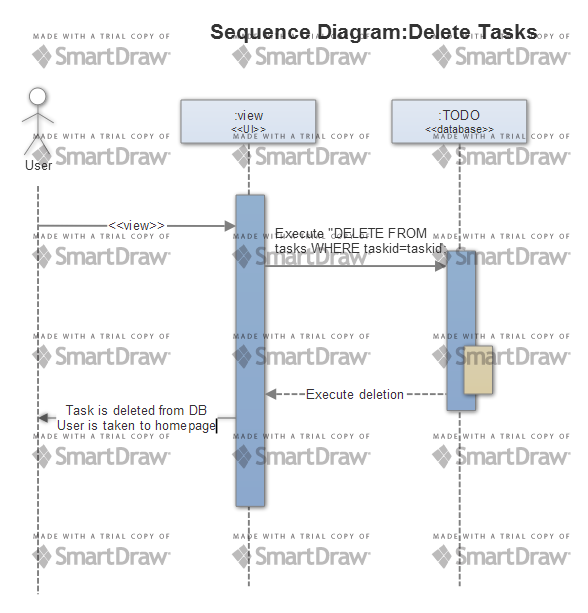
1. **Add a Task**



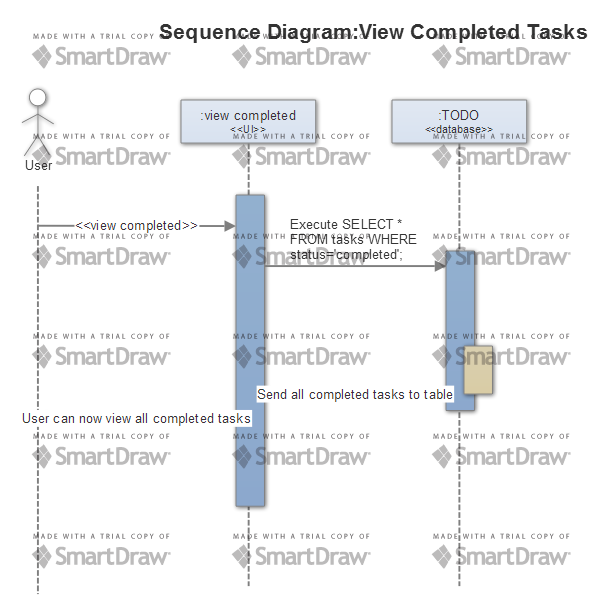
1. **View all tasks**



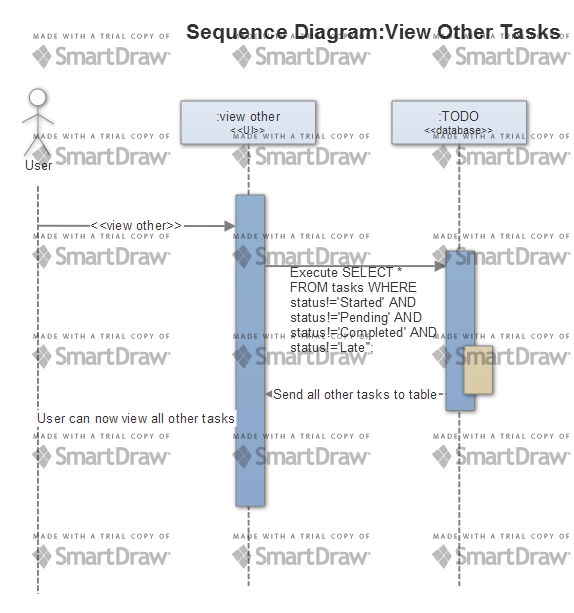
1. **Delete a task**



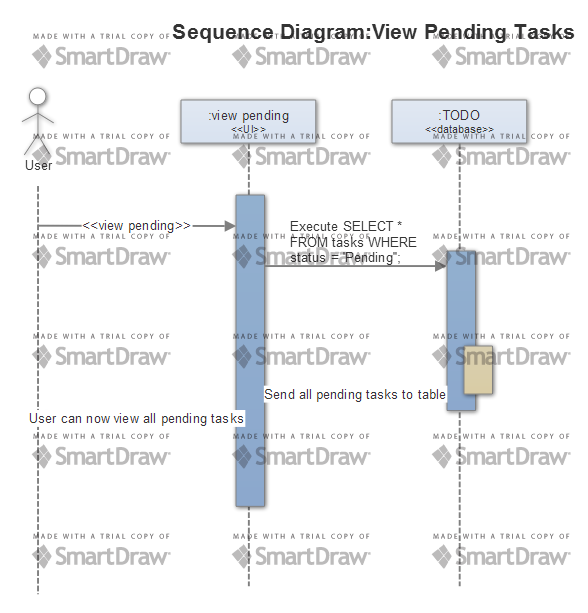
1. **View Completed Tasks**



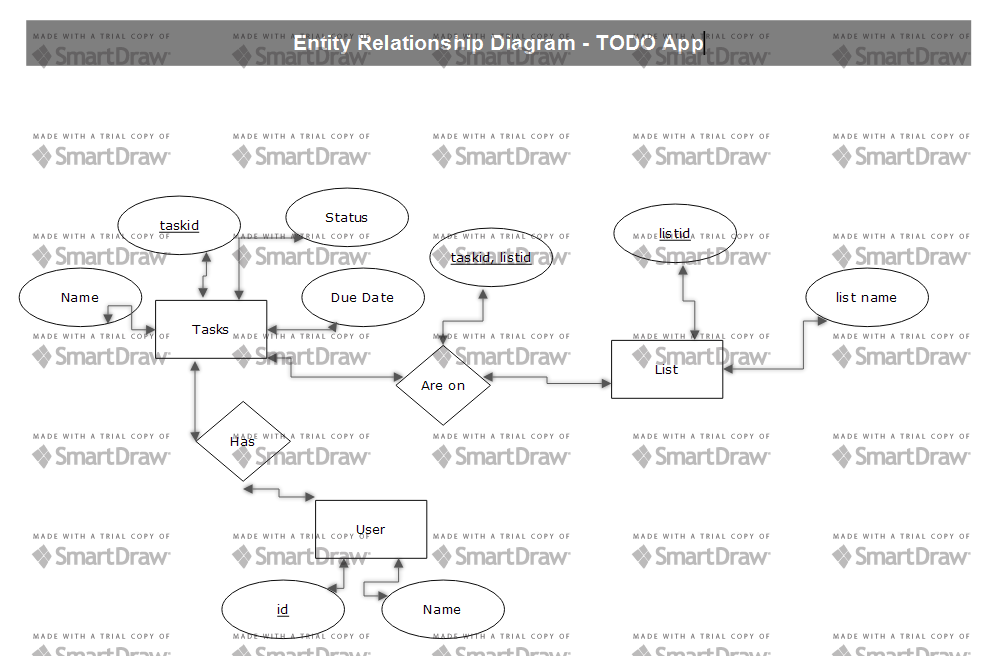
1. **View Other Tasks**



1. **View Pending Tasks**



**Database Design**



**Class Diagram**

****

**Test Cases**

I found the format for the below test cases from this website: <https://blog.testlodge.com/how-to-write-test-cases-for-software-with-sample/>. Author Jake Bartlett

1. **Title:** Add a Task

**Description:** User should be able to access todo app and add a task to the todo list. Precondition: WAMP Server must be installed and configured. After downloading the todo file with the .php files in them and saving it in the www directory in WAMP, navigate to WAMP Server localhost and then localhost/todo/index.php.

**Test Steps:**

1. Click on the WAMP Server W in the lower right of your screen, navigate to localhost.
2. Once there, navigate to localhost/todo/index.php
3. Click on Add a Task
4. Fill out all fields of the form on the add.php page. The fields include name for the task, its status, and a due date.
5. After filling out the form click submit or enter.
6. At any point in filling out the form the user can press Home/Cancel to navigate back to the homepage.

**Expected Result:** After the user presses submit or enter they will be taken back to the homepage and they will be able to see that the total number of tasks has increased by one as well as the status count for whatever they entered the status as for the task they just entered.

1. **Title:** View Tasks

**Description:** User should be able to access todo app and view all tasks to the todo list. Precondition: WAMP Server must be installed and configured. After downloading the todo file with the .php files in them and saving it in the www directory in WAMP, navigate to WAMP Server localhost and then localhost/todo/index.php.

**Test Steps:**

1. Click on the WAMP Server W in the lower right of your screen, navigate to localhost.
2. Once there, navigate to localhost/todo/index.php
3. Click on View/Delete Tasks
4. If there are any tasks in the system they will displayed in a table, otherwise a message is output saying there are no tasks in the system. The user can stay on this page or navigate back to the homepage by clicking a hyperlink that is either just below the outputted table or the message saying there are no tasks.

**Expected Result:** The user will be able to clearly see all the tasks that are currently in the system.

1. **Title:** Delete a Task

**Description:** User should be able to access todo app, view all tasks on the todo list and then be able to delete whichever one they want. Precondition: WAMP Server must be installed and configured. After downloading the todo file with the .php files in them and saving it in the www directory in WAMP, navigate to WAMP Server localhost and then localhost/todo/index.php.

**Test Steps:**

1. Click on the WAMP Server W in the lower right of your screen, navigate to localhost.
2. Once there, navigate to localhost/todo/index.php
3. Click on View/Delete Tasks
4. If there are any tasks in the system they will displayed in a table, otherwise a message is output saying there are no tasks in the system. The user can stay on this page or navigate back to the homepage by clicking a hyperlink that is either just below the outputted table or the message saying there are no tasks.
5. The user simply has to click Delete, which is the right most column of each row, to delete the task that is in that row.
6. The user is they naviagated back to the homepage where the deletion is reflected by the overall tasks being decreased by one as well as whatever status it was also decreased by one.
7. If the user then clicks View/Delete Tasks again they will see that the task they just deleted is no longer there.

**Expected Result:** The user will be able to clearly see all the tasks that are currently in the system and then choose which, if any, they want to delete.

1. **Title:** View Completed Tasks

**Description:** User should be able to access todo app, and be able to filter the current tasks to only show the completed ones. Precondition: W WAMP Server must be installed and configured. After downloading the todo file with the .php files in them and saving it in the www directory in WAMP, navigate to WAMP Server localhost and then localhost/todo/index.php.

**Test Steps:**

1. Click on the WAMP Server W in the lower right of your screen, navigate to localhost.
2. Once there, navigate to localhost/todo/index.php
3. Click the blue number to the right of the sentence “Total Number of Completed Tasks in the System: “. Clicking this will take you to a page that lists all the completed tasks.
4. After viewing the completed tasks the user can press the Home link below the completed tasks table to return to the homepage

**Expected Result:** The user will be able to clearly see all the completed tasks that are currently in the system.

1. **Title:** View Other Tasks

**Description:** User should be able to access todo app, and be able to filter the tasks to only show the ones that do *not* have a status of Pending, Started, Completed, or Late. Precondition: WAMP Server must be installed and configured. After downloading the todo file with the .php files in them and saving it in the www directory in WAMP, navigate to WAMP Server localhost and then localhost/todo/index.php.

**Test Steps:**

1. Click the blue number to the right of the sentence “Total Number of Other Tasks in the System: “. Clicking this will take you to a page that lists all the other tasks.
2. After viewing the other tasks the user can press the Home link below the other tasks table to return to the homepage

**Expected Result:** The user will be able to clearly see all the other tasks that are currently in the system.

1. **Title:** Delete All Tasks

**Description:** User can delete all tasks at once instead of individually. Precondition: WAMP Server must be installed and configured. After downloading the todo file with the .php files in them and saving it in the www directory in WAMP, navigate to WAMP Server localhost and then localhost/todo/index.php.

**Test Steps:**

1. Click on View/Delete Tasks
2. If there are any tasks in the system then there will be a button displayed that says Delete All, click the button to delete all the tasks currently in the system.
3. The user is then taken back to the homepage and there are no longer any tasks in the database.

**Expected Result:** The user will be able to delete all tasks at once by pressing a button on the view tasks page.