**To-Do List Setup**

1. Install Bitnami WAMP Stack using WAMP Installer found here: <https://bitnami.com/stack/wamp>
2. Navigate to folder location:

[C:\Bitnami\wampstack-7.1.12-0\apache2\htdocs](file:///C:\Bitnami\wampstack-7.1.12-0\apache2\htdocs)

1. Copy files (todolist.php, todolist.css, sqlCon.php, sqlCommands.php, textRtn.php) to folder
2. Launch Bitnami WAMP Stack Manager Tool
3. On the ‘Welcome’ tab click ‘Open phpMyAdmin’
4. Username = ‘root’, Password is the password set during MySQL installation through WAMP Installer
5. Click SQL tab to open a SQL query box.
6. Enter the following SQL commands to set up database:

CREATE DATABASE todo;

-- Table structure for table duedates

--

CREATE TABLE duedates (

date\_id int(11) NOT NULL,

due\_date date NOT NULL

)

-- Table structure for table statuses

--

CREATE TABLE statuses (

status\_id int(11) NOT NULL,

status varchar(25) DEFAULT NULL

)

-- Dumping data for table statuses

--

INSERT INTO statuses (status\_id, status) VALUES

(1, 'Pending'),

(2, 'Started'),

(3, 'Completed'),

(4, 'Late');

-- Table structure for table tasks

--

CREATE TABLE tasks (

id int(11) NOT NULL,

task varchar(255) DEFAULT NULL,

status\_id int(11) NOT NULL DEFAULT '1'

)

-- Indexes for table duedates

--

ALTER TABLE duedates

ADD PRIMARY KEY (date\_id);

--

-- Indexes for table statuses

--

ALTER TABLE statuses

ADD PRIMARY KEY (status\_id);

--

-- Indexes for table tasks

--

ALTER TABLE tasks

ADD PRIMARY KEY (id);

-- AUTO\_INCREMENT for table tasks

--

ALTER TABLE tasks

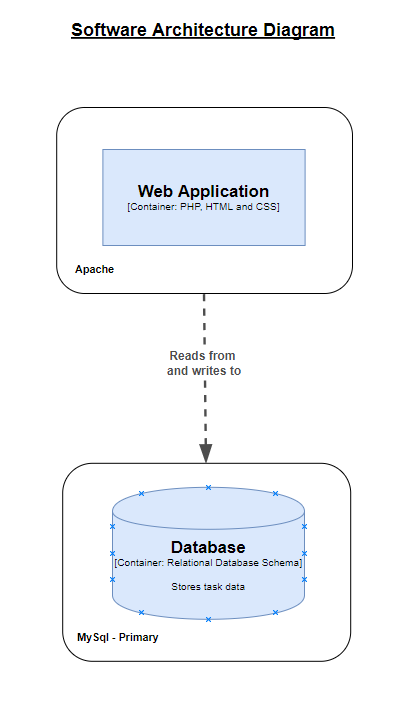
MODIFY id int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=106;

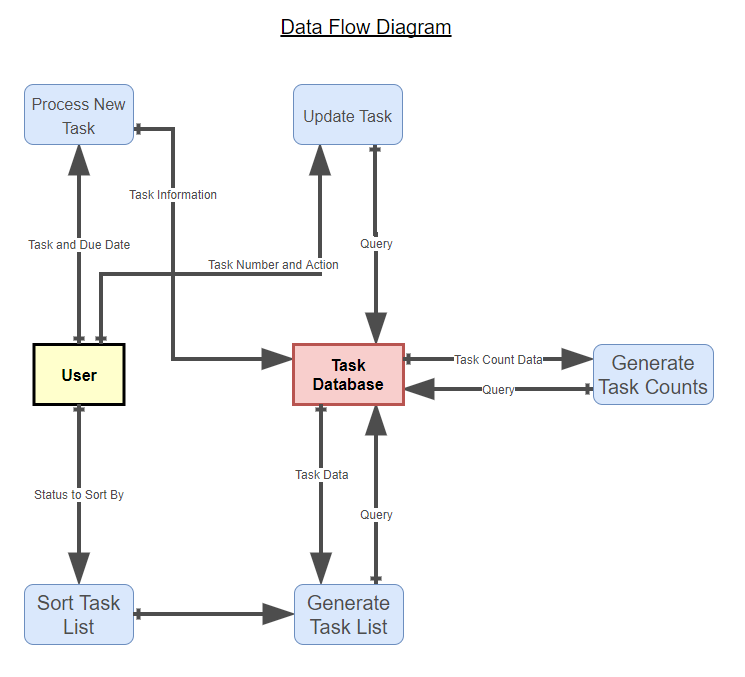
1. Modify sqlCon.php and set variable ‘pass’ to your MySQL server password.
2. Open a browser and navigate to: <http://localhost/todolist.php>

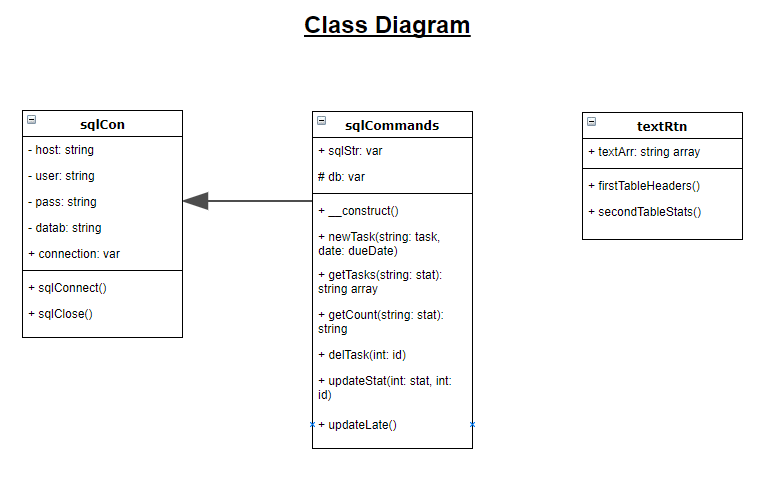
|  |  |
| --- | --- |
| Functional Requirements | |
| 1 | Must allow the user to add a task |
| 2 | Must allow the user to assign a due date to a task |
| 3 | Must allow the user to view filtered status lists |
| 4 | Must allow the user to mark a task as “started” |
| 5 | Must allow the user to mark a task as “completed” |
| 6 | Must allow the user to delete a task |

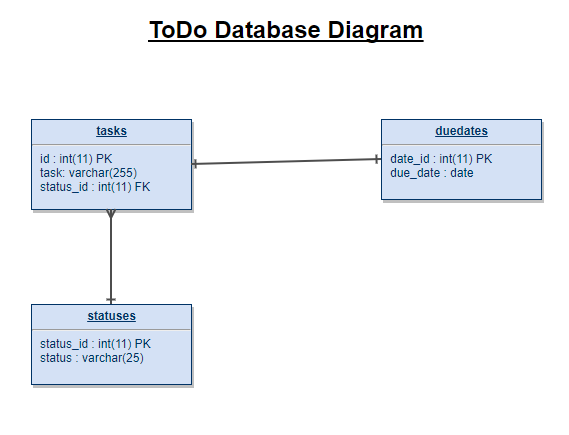
|  |  |
| --- | --- |
| Non-functional Requirements | |
| 1 | The application provides a user friendly interface |
| 2 | Must be able to communicate with a MySQL database to insert and extract information |
| 3 | The application must provide certain functionalities in the user interface |
| 4 | Must not have a limit of tasks to add |
| 5 | Must have a consistent style of text |
| 6 | Queries must have a fast response time |

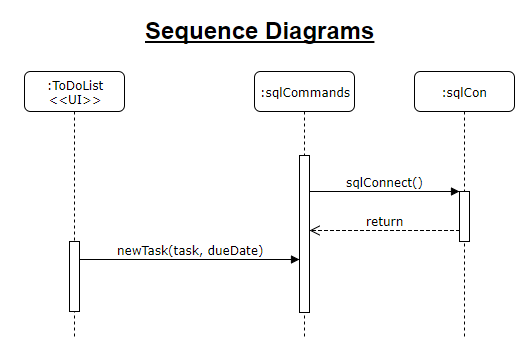
|  |  |
| --- | --- |
| Use Cases | |
| 1 | Add new task |
| 2 | Add due date to task |
| 3 | Update task to ‘Started’ status |
| 4 | Update task to ‘Completed’ status |
| 5 | Delete task |
| 6 | Filter list based on status |

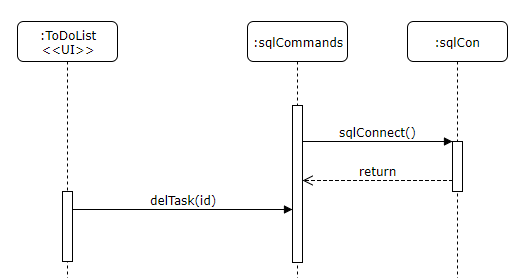


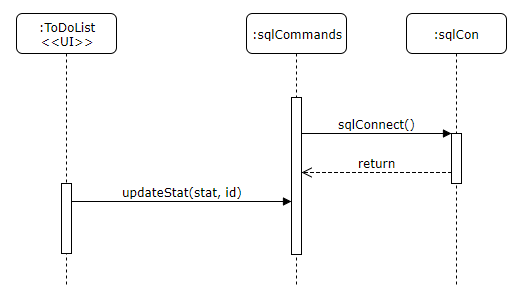


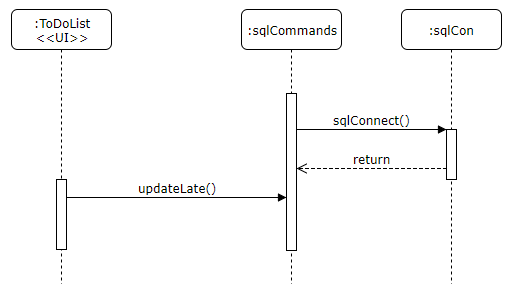
.

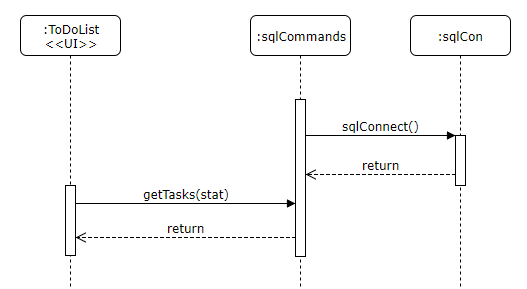


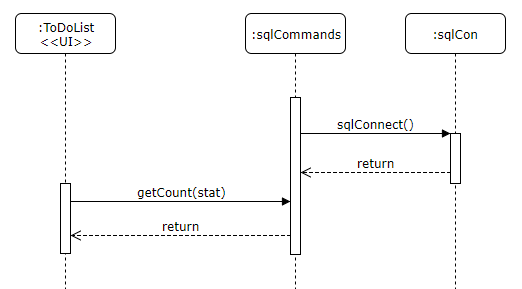












|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Work Break Down | | | | |
| Name of Task | Est. # of Hours | Actual Time Spent | Issues | Notes |
| Understanding PHP basics | 1 | 2.5 |  | Spent extra time watching tutorial videos to fully understand |
| Setting up work environment | 1.5 | 2 | Took some time to find correct installation for work environment | Spent time understanding how the installed software works |
| Building user interface | 2 | 2.5 |  |  |
| Building database | 1 | 0.5 |  | Used phpMyAdmin to create database |
| Adding PHP to work with database. Adjust UI accordingly. | 8 | 16 | Had problems with inserting into MySQL. Also struggled with passing certain parameters to class functions. |  |
| Documenting application | 5 | 7.5 |  |  |