## ToDo List Application Specifications Report

By: Norma Chang

January 12, 2019

Prepared for CSC 4996: Senior Project

Wayne State University

Table of Contents

**1. Requirements3**

1.1 Functional Requirements3

1.2 Non-Functional Requirements3

**2. System Architecture Diagram4**

**3. Data Flow Diagram5**

**4. Use Case Diagram6**

**5. Sequence Diagram7**

**6. Database Design8**

**7. Class Diagram9**

**8. Test Case10**

8.1 Test Case: Add Task10

8.2 Test Case: Delete Task10

8.3 Test Case: View Task11

**1. REQUIREMENTS**

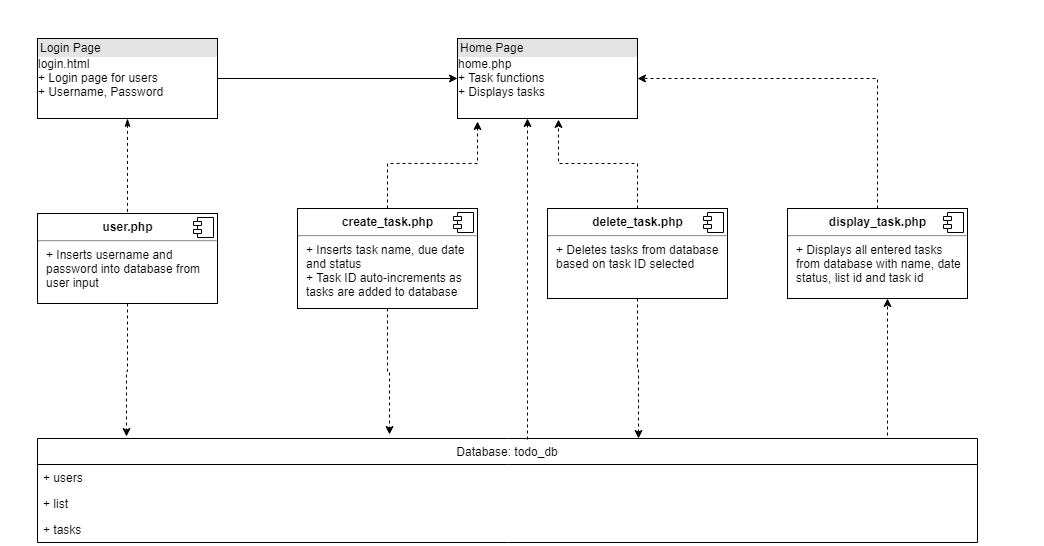
* 1. Functional Requirements

|  |  |  |
| --- | --- | --- |
| Quality Attribute | Requirement Definition | Scope/Analysis |
| Add Task | Addition of task input into the database | User must be able to add a task with its name, due date and status. |
| Delete Task | Removal of a task from the database | User must be able to delete a task that has been added to the application. |
| View Task | Display of tasks in the task list | User should be able to view all the tasks that they have added. |

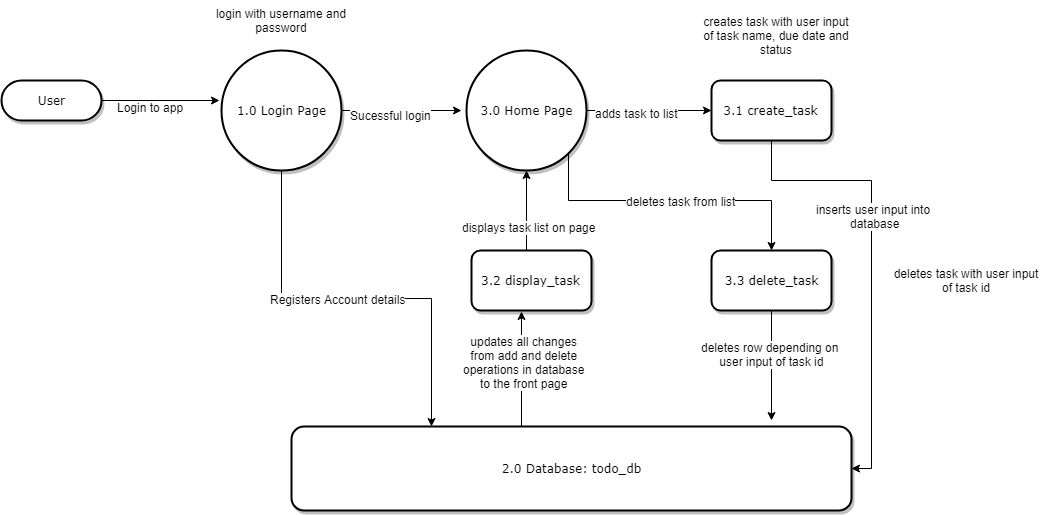
* 1. Non-Functional Requirements

|  |  |  |
| --- | --- | --- |
| Quality Attribute | Requirement Definition | Scope/Analysis |
| Scalability | Ease of adding data to the database from user input | Table size in the MySQL databse is constrained by the operating system constraints on file size. |
| Recoverability | In the event of hardware or security failures, the possibility of recovering the database and application data. | Unless proper backups are configured, in the event of a hardware/security failure, the code and the data in the database would need to be manually recreated. |
| Integrity | Consistency of the methods, values, inputs, results and data. | Avoid using methods like POST to prevent security issues like SQL injection. Limiting how the data in the database is access to prevent unauthorized modifications to the database data. |

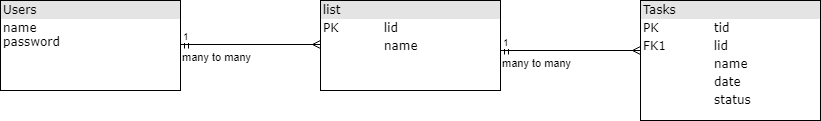
**2. Systems Architecture Diagram**



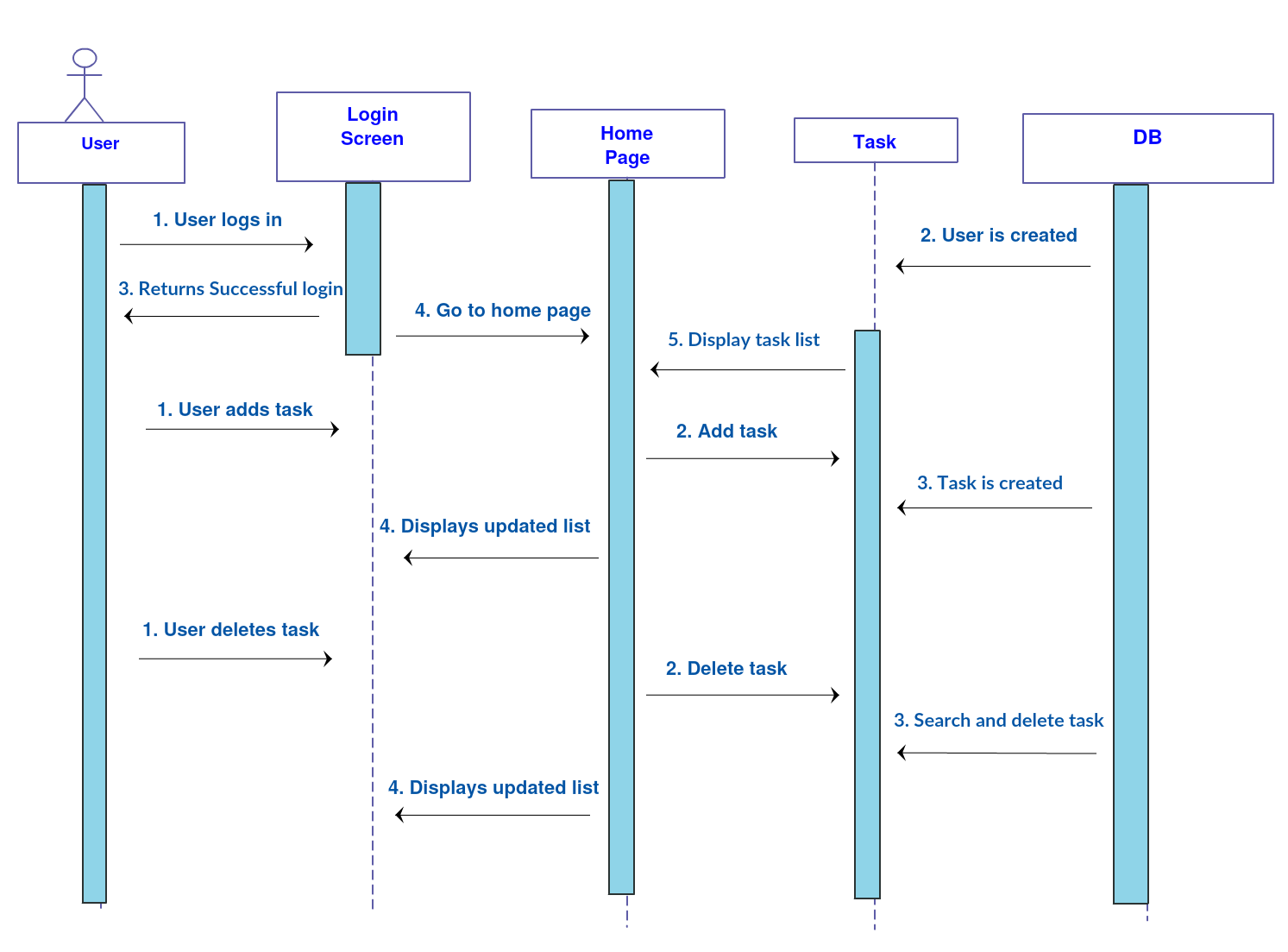
**3. Data Flow Diagram**



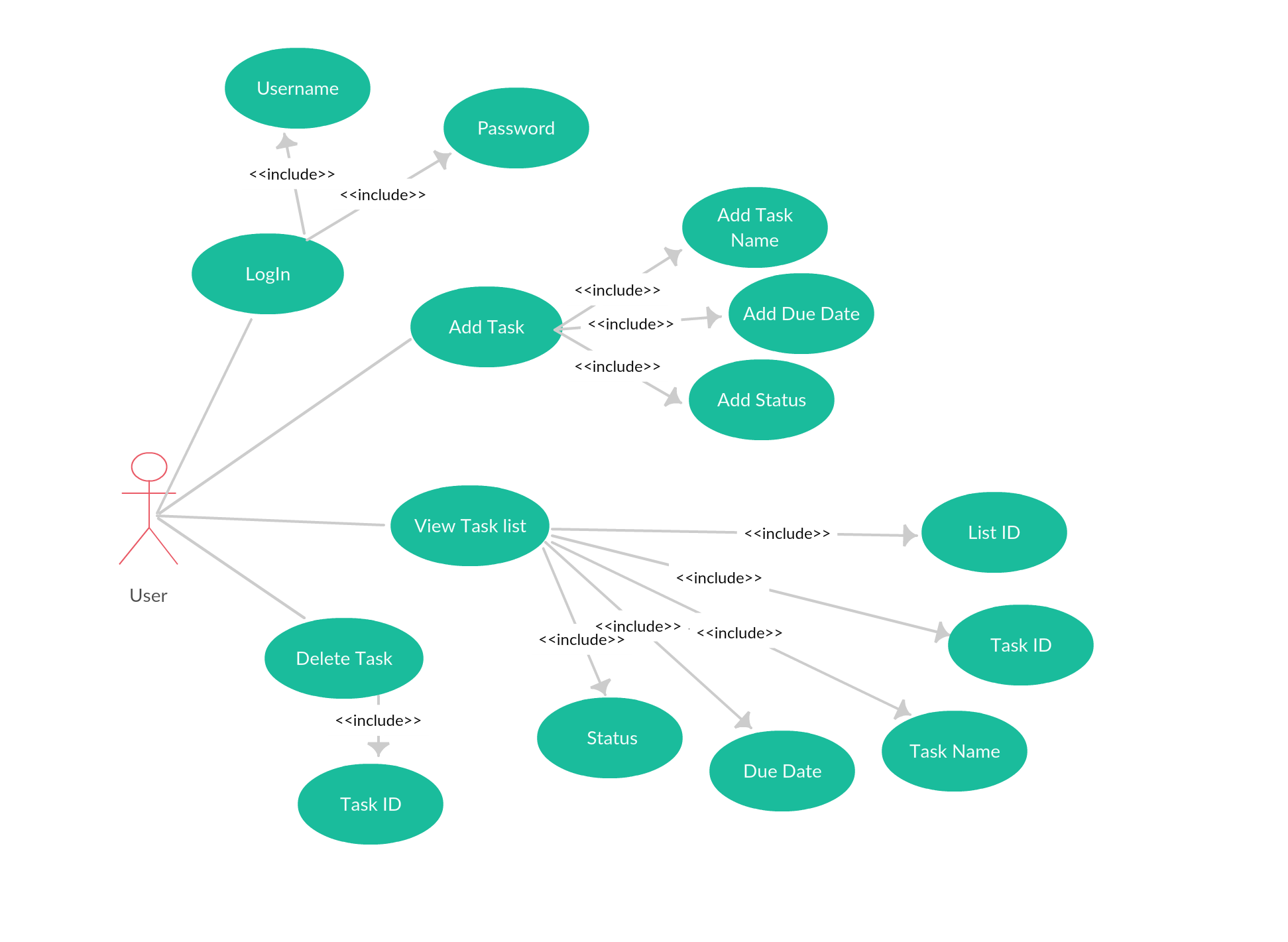
**4. Use Case Diagram**



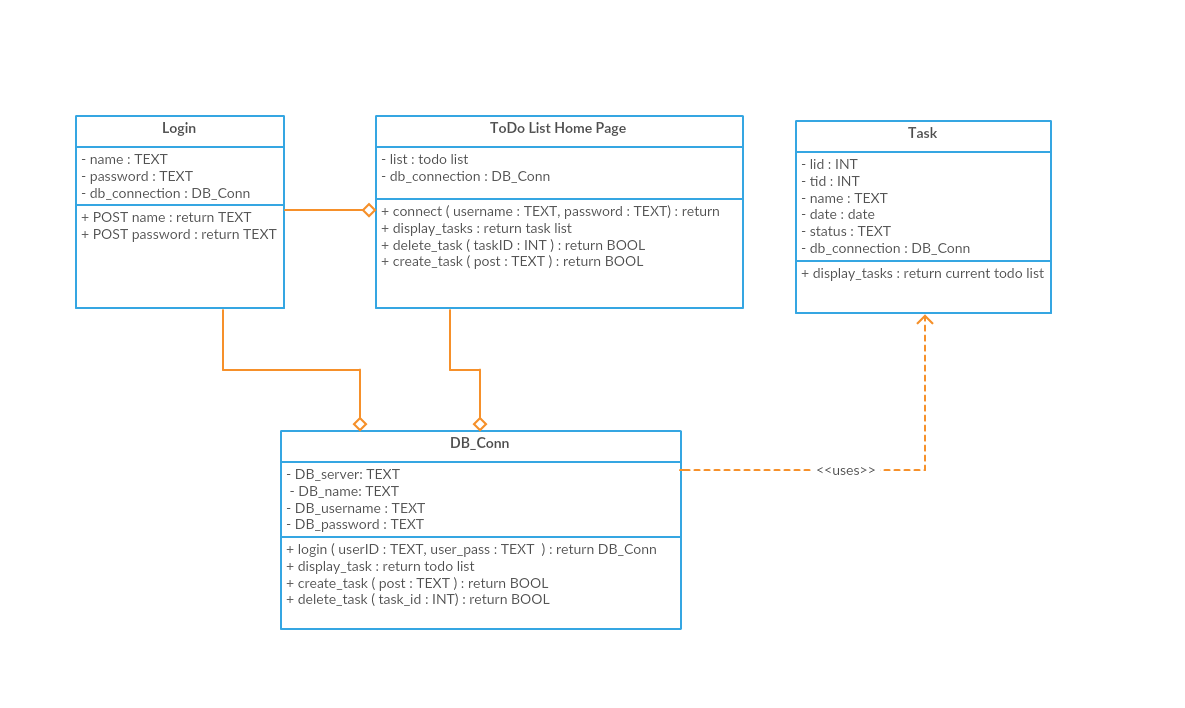
**5. Sequence Diagram**



**6. Database Design Diagram**



**7. Class Diagram**



**8. Test Case**

**1. Login**

1.1 Login - Successful

|  |  |
| --- | --- |
| Test Case Name | Login – Login Successful |
| Test Case ID | TC-1 |
| Preconditions | User account from registration |
| Postconditions | User can log in and view Todo List Dashboard |
| Test Steps | 1. Open Application  2. Enter Username and Password (account will be automatically registered if it does not exist)  3. Click Login |
| Expected Results | Application displays Todo List Home Page |

**2. Add Task**

2.1 Add Task - Due

|  |  |
| --- | --- |
| Test Case Name | Add Task – Add Task Due |
| Test Case ID | TC-2 |
| Preconditions | Successful completion of login |
| Postconditions | Task is added to the database and displayed in the Todo List view |
| Test Steps | 1. Under Add Task, Enter “Due Task” in the Task Name field 2. Click the drop down in the calendar field and select “01/14/2019” 3. Type “Due” in the Task Status field 4. Click Add Task |
| Expected Results | “Due Task” will display under Todo List |

2.2 Add Task - Started

|  |  |
| --- | --- |
| Test Case Name | Add Task – Add Task Started |
| Test Case ID | TC-3 |
| Preconditions | Successful completion of login |
| Postconditions | Task is added to the database and displayed in the Todo List view |
| Test Steps | 1. Under Add Task, Enter “Started Task” in the Task Name field 2. Click the drop down in the calendar field and select “01/10/2019” 3. Select “Started” in the Task Status field 4. Click Add Task |
| Expected Results | “Started Task” will display under Todo List |

**3. Delete Task**

3.1 Delete Task - Started

|  |  |
| --- | --- |
| Test Case Name | Delete Task – Delete Task Started |
| Test Case ID | TC-4 |
| Preconditions | Successful completion of login and TC-2 |
| Postconditions | Task is deleted from database and list view will be updated |
| Test Steps | 1. Under Delete Task, enter the task id of the “Started Task” 2. Click Delete |
| Expected Results | “Started Task” will be removed under the task list |