**TODO Application**

**Functional Requirements:**

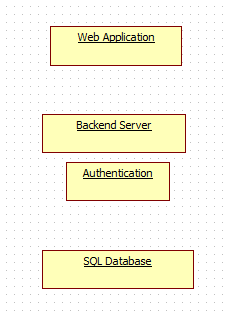
1. The application shall allow the user to add a task to the list
2. The application shall allow the user to delete a task from the list
3. The application shall allow the user to view all tasks in the list

**Non-Functional Requirements:**

1. The application shall be able to store at least 200 tasks
2. The application shall load each page within two seconds
3. The application shall allow each task to be up to 100 characters

**System Architecture Diagram:**

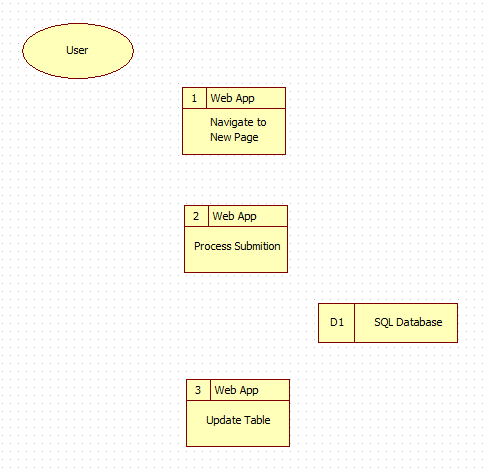
Interface/Local Application Logic

****

Server Storage

Server Logic

**Data Flow Diagram:**

****

Send confirmation or error

Approve or query error

New task information/query

Make task change

Choose option

**Use Case 1:**

Name: Add Task  
Brief Description: Allows user to add task to their to-do list.  
Actors:

* Common User/Person
* Application/System
* SQL Database

Triggers: The user has pressed the “Add Task” button.  
Preconditions: The user has all the technology to start the app.  
Post-conditions: The user has successfully added a new task to their to-do list and it is stored in the database ready to view.  
Normal Flow:

1. The user will indicate they would like to add a task by pressing the “Add Task” button.
2. The system will prompt the user to enter a task.
3. The user will enter a task and hit the submit button.
4. The system will send this information to the database and store the new task along with a unique task ID.
5. The system will navigate to a new page which offers the option of returning to the main menu.
6. The user exits the system.

Alternative Flow:

6A1. The user goes back to the main menu.  
 6A2. The user chooses a new option.

**Use Case 2:**

Name: Delete Task  
Brief Description: Allows user to delete a task from their to-do list.  
Actors:

* Common User/Person
* Application/System
* SQL Database

Triggers: The user has pressed the “Delete Task” button.  
Preconditions: The user has all the technology to start the app.  
Post-conditions: The user has successfully deleted a previous task from their to-do list and the new table is ready to be viewed.  
Normal Flow:

1. The user will indicate they would like to delete a task by pressing the “Delete Task” button.
2. The system will prompt the user to enter the task ID of the task they would like to delete.
3. The user will enter a task ID and hit the submit button.
4. The system will send this information to the database and remove the task ID along with the associated task.
5. The system will navigate to a new page which offers the option of returning to the main menu.
6. The user exits the system.

Alternative Flow:

6A1. The user goes back to the main menu.  
 6A2. The user chooses a new option.

**Use Case 3:**

Name: View Tasks  
Brief Description: Allows user to view all the tasks in their to-do list.  
Actors:

* Common User/Person
* Application/System
* SQL Database

Triggers: The user has pressed the “View Tasks” button.  
Preconditions: The user has all the technology to start the app.  
Post-conditions: The user has successfully been provided with a table of all the tasks in their to-do list.

Normal Flow:

1. The user will indicate they would like to view all the tasks by pressing the “View Tasks” button.
2. The system will retrieve the data from the SQL database.
3. The system will present the table of tasks along with their corresponding task ID and offer the option of returning to the main menu.
4. The user exits the system.

Alternative Flow:

4A1. The user goes back to the main menu.  
 4A2. The user chooses a new option.

**Sequence Diagram:**

Table

Computer

Server

Add Task

Submit new task

Send INSERT query

Table

Computer

Server

Delete Task

Submit task ID

Send DELETE query

Table

Computer

Server

View Tasks

Send SELECT query

Send SELECT query

**Database Design:**

**Status Table**

**taskID** int primary key **taskComplete** bool

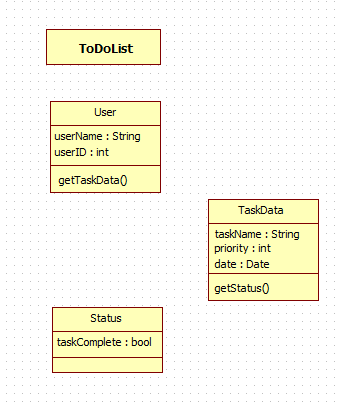
**User Table**

**firstName** varchar(10)  
**lastName** varchar(10)  
**username** varchar(15) primary key

**Tasks Table**

**task\_num** int auto\_increment primary key  
**task** varchar(100)

**Class Diagram:**

****

**Test Case:**

|  |  |  |
| --- | --- | --- |
| **Test Case Scenario** | **Test Steps** | **Expected Results** |
| Add Task | 1. Start app 2. Click “Add Task” 3. Type in a task in string form 4. Click the “Submit” button | The new task should be added to the SQL table. If “View Tasks” was clicked, the new task should now be visible. |
| Delete Task | 1. Start app 2. Click “Delete Task” 3. Type in the task ID of the task you would like to delete 4. Click the “Submit” button | The new task should be deleted from the SQL table. If “View Tasks” was clicked, the new task should now be gone. |
| View Tasks | 1. Start app 2. Click “View Tasks” | A table of all the tasks should be visible. |