Kalie Humbarger

Ev4295@wayne.edu

To Do application

Software Documentation

Table of Contents

[Functional Requirements 2](#_Toc503727202)

[Non-Functional Requirements 2](#_Toc503727203)

[System Architecture Diagram 2](#_Toc503727204)

[Data Flow Diagram 3](#_Toc503727205)

[Use Cases 3](#_Toc503727206)

[Sequence Diagrams 4](#_Toc503727207)

[Database Diagram 7](#_Toc503727208)

[Class Diagram 8](#_Toc503727209)

[Test Cases 8](#_Toc503727210)

[Work Breakdown 11](#_Toc503727211)

# Functional Requirements

**FR-1** Application must be able to display the ID, name, description, priority, status, and due date of each task in the system.

**FR-2** Application must allow the user to display the tasks according to status.

**FR-3** Application must allow the user to delete a task from the system.

**FR-4** Application must allow the user to update the following attributes for any task in the system: name, description, priority, status, due date.

**FR-5** Application must allow the user to add new tasks to the system by providing the following attributes: name, description, priority, status, due date.

**FR-6** Application update a tasks status to late if the task remains incomplete beyond its due date.

# Non-Functional Requirements

**NFR-1** The application must add a new task to the system in under 1 second.

**NFR-2** The application must delete a task from the system in under 1 second.

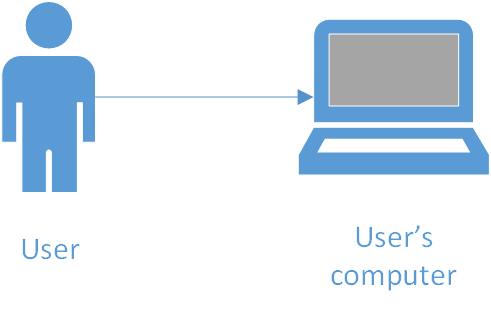
**NFR-3** The application must update a task in the system in under 1 second.

**NFR-4** The system must protect database credentials from being viewed within the browser.

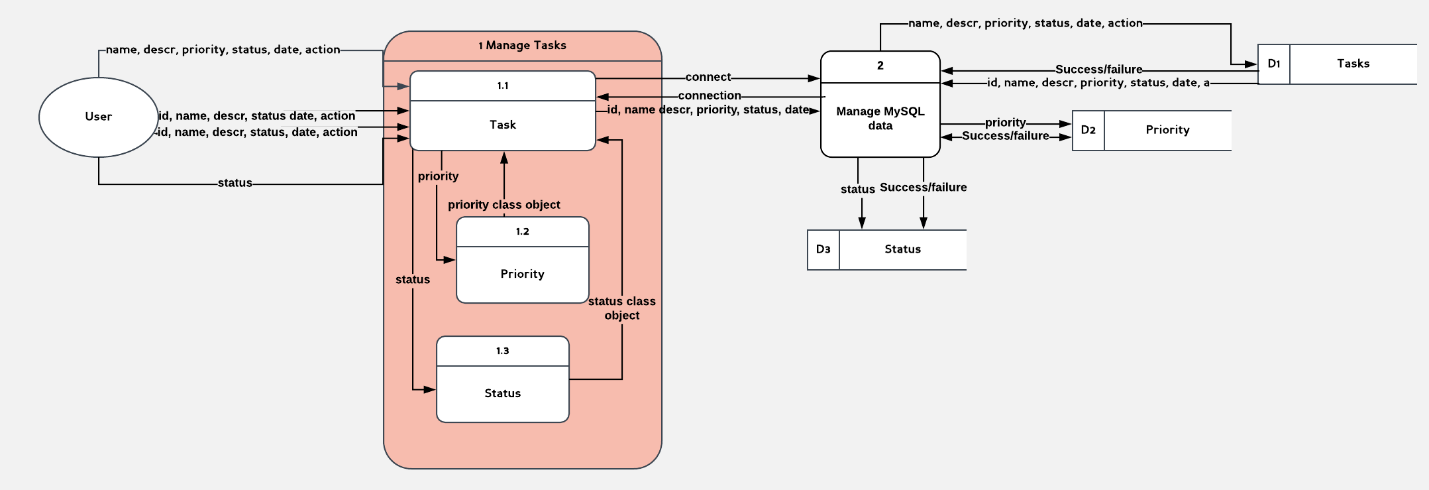
**NFR-5** The system must recover from errors without crashing.

**NFR-6** The system must be able to support more than 20 tasks at a time.

# System Architecture Diagram



# Data Flow Diagram



# Use Cases

**UC-1** Adds a Task

**Description:** A user adds a new task to the application.

**Primary Actor:** User.

**Basic Flow:** The user enters valid attributes into the name, description, priority, status, and due date fields, then clicks the submit button located to the right of the fields. The task is added to the system.

**Alternate Flows:** The user enters invalid attributes into one or more of the fields described above. The task is not added to the system.

**UC-2** Deletes a Task

**Description:** A user deletes an existing task from the system.

**Primary Actor:** User.

**Basic Flow:** The user clicks the delete button located to the right of the task in the task table. The task is deleted from the system.

**Alternate Flows:** None.

**UC-3** Updates a Task

**Description:** A user updates an existing task in the system.

**Primary Actor:** User.

**Basic Flow:** The user enters valid attributes into one or more fields (name, description, priority, status, or due date fields), then clicks the submit button located to the right of the fields. The task is updated in the system.

**Alternate Flows:** The user enters invalid attributes into one or more of the fields described above. The task is not updated in the system.

**UC-4** Displays All Tasks

**Description:** The application will display to the user all tasks in the system.

**Primary Actor:** User.

**Basic Flow:** A user navigates to the homepage of the application or clicks on a link associated with “total tasks”. The application will display to the user all tasks in the system.

**Alternate Flows:** None.

**UC-5** Displays All Pending Tasks

**Description:** The application will display to the user all pending tasks in the system.

**Primary Actor:** User.

**Basic Flow:** A user navigates to the homepage and the clicks a link associated with pending tasks. The application will then display to the user all pending tasks in the system and only those tasks.

**Alternate Flows:** None.

**UC-6** Displays All Completed Tasks

**Description:** The application will display to the user all completed tasks in the system.

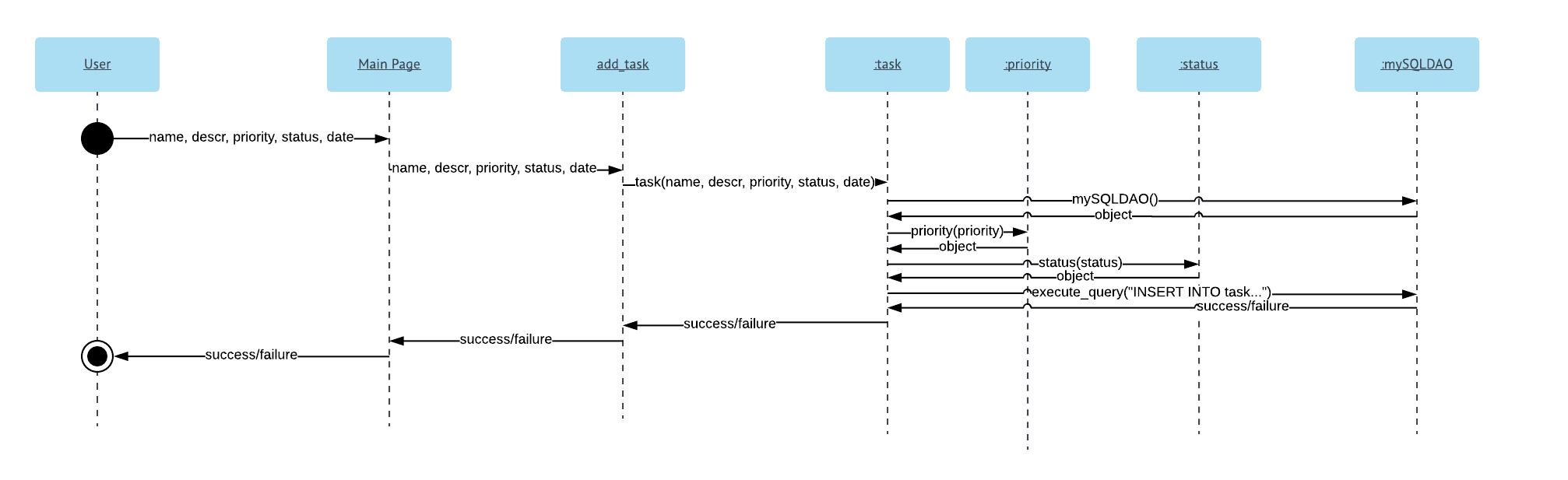
**Primary Actor:** User.

**Basic Flow:** A user navigates to the homepage and the clicks a link associated with completed tasks. The application will then display to the user all completed tasks in the system and only those tasks.

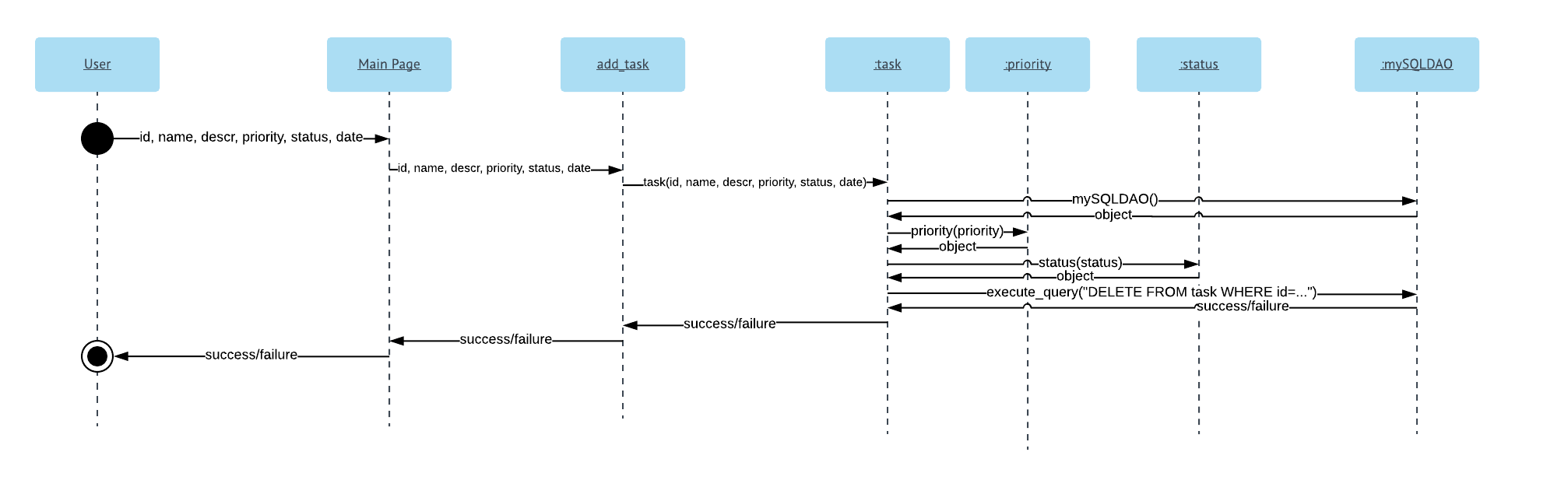
**Alternate Flows:** None.

# Sequence Diagrams

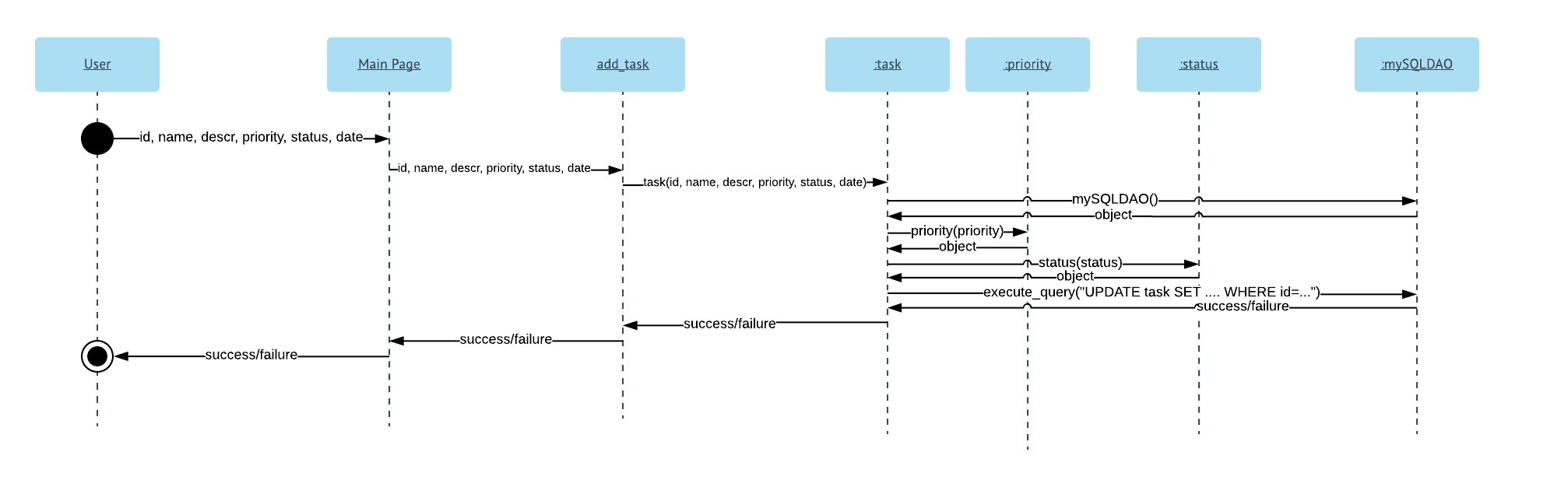
**Sequence Diagram 1 – Add New Task**



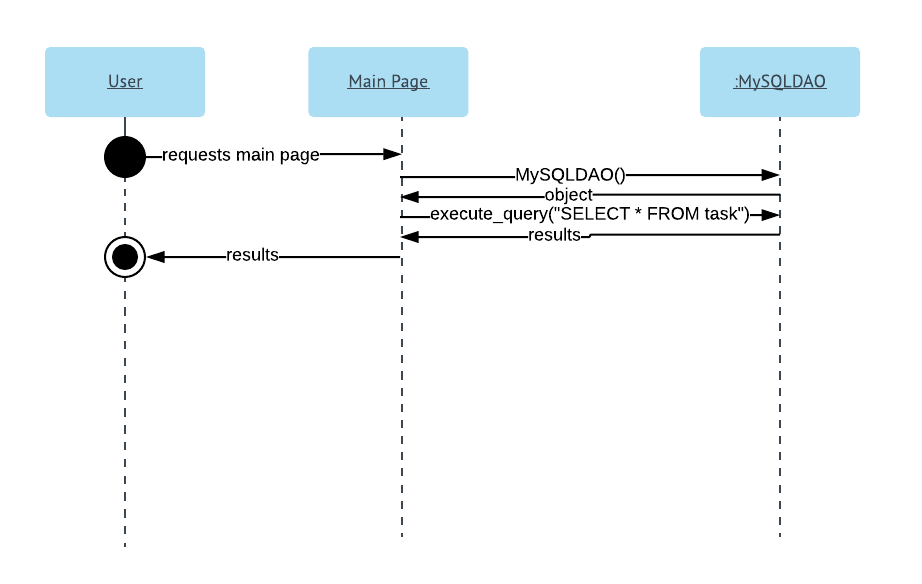
**Sequence Diagram 2 – Delete Task**



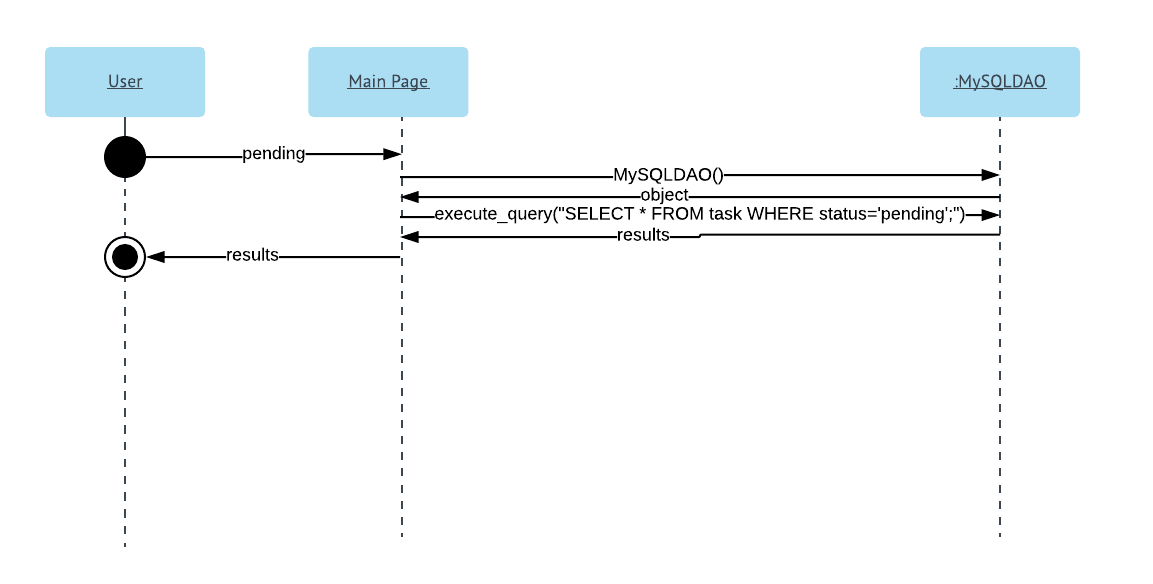
**Sequence Diagram 3 – Update Task**



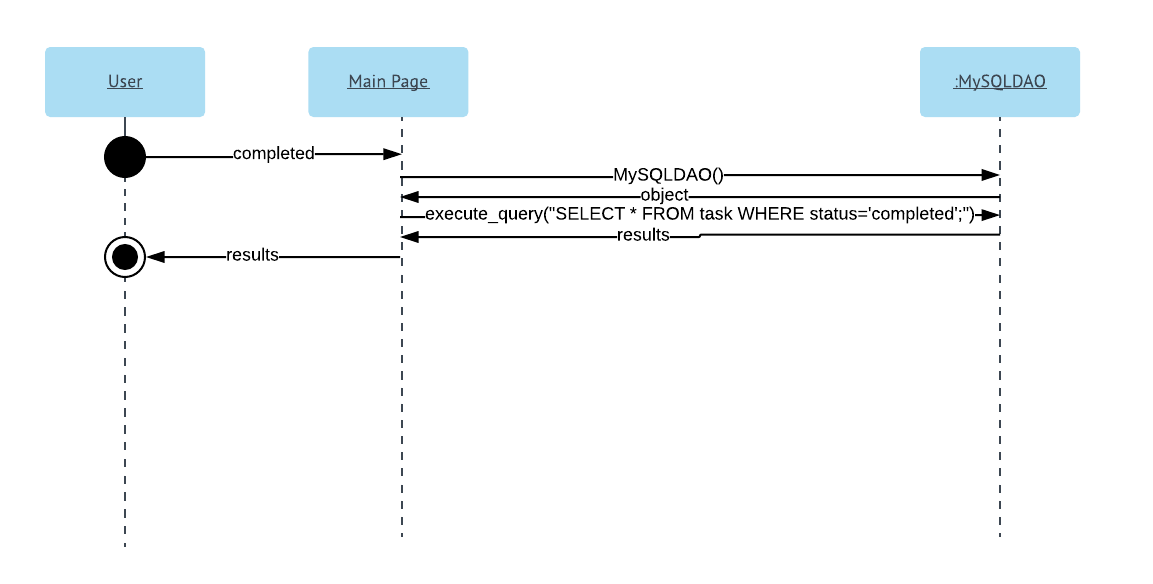
**Sequence Diagram 4 – Display All Tasks**



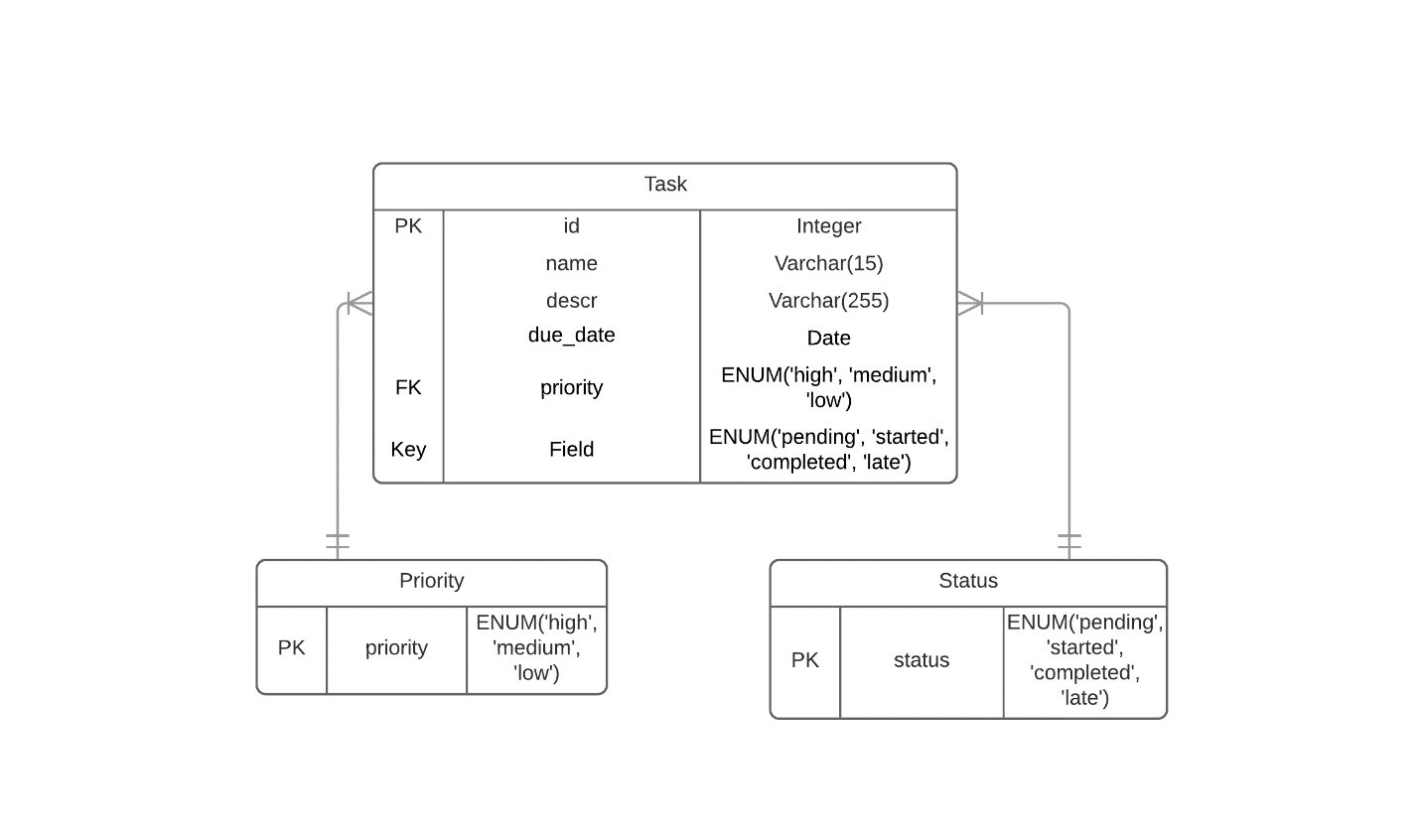
**Sequence Diagram 5 – Display All Pending Tasks**



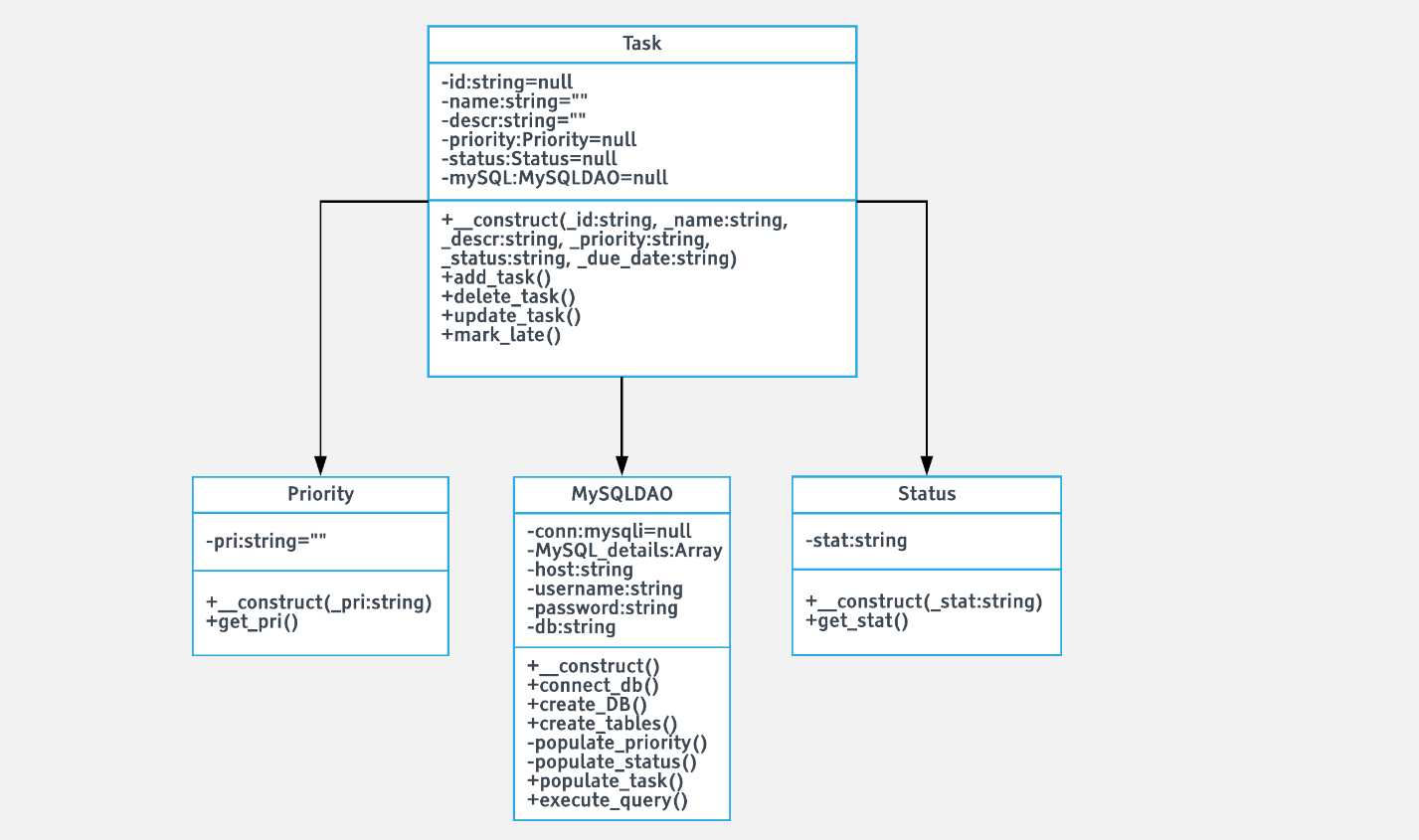
**Sequence Diagram 6 – Display All Completed Tasks**



# Database Diagram



# Class Diagram



# Test Cases

**Test Case 1**: Main Page: Add New Task with Valid Task Attributes

Description**:** User should be able to add a new task using valid data by entering the data into the provided fields and clicking “submit”. Within the context of the application, “valid data” means a title which is less than or equal to 15 characters, a description which is less than or equal to 255 characters, a priority of high, medium, or low, a status of pending, started, completed, or late, and a date in mm/dd/yyyy format which exists on the Gregorian calendar.

Assumptions: The user is using a supported browser and has been able to load the main page.

Preconditions: Upon loading the main page, the database and tables are successfully created if they do not already exist.

Test Steps:

1. Navigate to the main page in supported browser
2. Under “new task” enter a valid task name, task description, priority, status, and date into their respective fields
3. Click submit

Expected Result: The main page should reload. The new task data should be displayed, just as it was entered, in the “tasks” table on the main page.

**Test Case 2:** Main Page: Delete an Existing Task

Description: User should be able to delete an existing task by selecting the “delete” button located to the right of the desired task.

Assumptions: The user is using a supported browser and has been able to load the main page. Tasks have already been added to the system.

Preconditions: none.

Test Steps:

1. Click the delete button located to the right of the task to be deleted.

Expected Result: The main page should reload and the deleted task should no longer appear in the “tasks” table on the main page.

**Test Case 3:** Main Page: Update an Existing Task with Valid Task Attributes

Description: From the main page, a user should be able to update an existing task with new attributes, so long as those attributes are valid as defined in **Test Case 1** by editing the fields in the “tasks” table on the main page, then clicking the “update” button located to the right of the desired task.

Assumptions: The user is using a supported browser and has been able to load the main pate.

Preconditions: At least one task has been added to the system.

Test Steps:

1. Load the main page in a supported browser
2. Chose an existing task and edit the attribute fields of that task to different, but valid attributes as described in **Test Case 1**
3. Click the “update” button

Expected Results: The main page should be reloaded, and the “tasks” table on the main page should have an entry which has replaced the old entry and contains the new attributes added by the test administrator. No other entry has been affected.

**Test Case 4:** Main Page: Displays all Tasks in System

Description: The “tasks” table on the main page should display all the tasks in the system. No task stored in the data base should be left out of this table.

Assumptions: The user is using a supported browser and has been able to load the main page. Tasks have already been added to the system.

Preconditions: The administer of the test must have access to the database through MySQL Workbench. This person must be logged in and able to see the contents of the database.

Test Steps:

1. Open MySQL workbench, log in, and view the tables stored in the applications database
2. Open the main page of the application
3. Check the contents of the task table in MySQL workbench against the contents of the “tasks” table on the main page of the application

Expected Results: The content of the tasks table in MySQL workbench and on the main page of the application should be the same.

**Test Case 5:** Pending Tasks Page: Application displays Only Pending Tasks

Description: The Pending Tasks Page of the application should display only tasks whose status is stored as “pending”. No tasks with statuses of “started”, “completed”, or “late” should be displayed on this page.

Assumptions: The user is using a supported browser and has been able to load the pending tasks page.

Preconditions: Tasks with a status of “pending” have been added to the system. At least one task with a status other than pending has been added to the system.

Test Steps:

1. Load the main page of the application
2. Click on “Pending Tasks” in the navigation bar
3. View the “status” column of the tasks table on the Pending Tasks page

Expected Result: The “status” column of the tasks table on the Pending Tasks page should contain *only* entries of “pending”. There should be no entries of “started”, “completed”, or “late”.

**Test Case 6:** Main Page: Add Task Failure with Invalid Due Date

Description: Adding a new task on the main page of the application should fail if a user enters an invalid due date. Within the context of this application, an invalid due date would be a date which is not in mm/dd/yyyy format or which does not exist on the Gregorian calendar. The remainder of the task attributes which the user enters into their existing fields should be valid as defined in **Test Case 1.**

Assumptions: The user is using a supported browser and has been able to load the main page.

Preconditions: none.

Test Steps:

1. Load main page in supported browser
2. Under “add task” enter valid attributes for name, description, status, and priority as defined in **Test Case 1**
3. Under add task, enter an invalid attribute for date as defined in **Test Case 6**
4. Click “submit”

Expected Result: The main page should be reloaded and a message displayed to the user which states that adding the new task failed because the date provided was invalid, thus the user should try again.

# Work Breakdown

**Coding Tasks**

|  |  |  |
| --- | --- | --- |
| **Task** | **Estimated Time** | **Actual Time** |
| Create basic main page | 5 min | 2 min |
| Connect to MySQL | 1 min | 5 min |
| Create database/table | 10 min | 17 min |
| Display list stored in MySQL | 30 min | 32 min |
| Update tasks | 1 hour | 27 min |
| Separate into 3 tables | 1 hour | 59 min |
| Delete tasks | 30 min | 16 min |
| Improve UI | 2 hours | 1 hour 40 min |
| Write Startup script | 15 min | 20 min |
| Write classes | 1 hour | 28 min |
| Count by status | 15 min | 8 min |
| Clickable counts | 20 min | 17 min |
| Check for lateness | 30 min | 5 min |
| Alert user invalid input | 1 hour | 45 min |
| Add error checking | 1 hour | 30 min |
| Add tasks | 30 min | 65 min |

**Documentation Tasks**

|  |  |  |
| --- | --- | --- |
| **Task** | **Estimated Time** | **Actual Time** |
| Functional Requirements | 30 min | 10 min |
| Non-functional Requirements | 30 min | 10 min |
| System Architecture Diagram | 30 min | 35 min |
| Data Flow Diagram | 20 min | 37 min |
| Use Case | 30 min | 20 min |
| Sequence Diagrams | 1 hour | 1 hour 30 min |
| Database Designs | 20 min | 22 min |
| Class Diagrams | 20 min | 26 min |
| Test cases | 30 min | 44 min |