Ty Austin
FL5861
CSC4996

To-Do Application

Functional Requirements

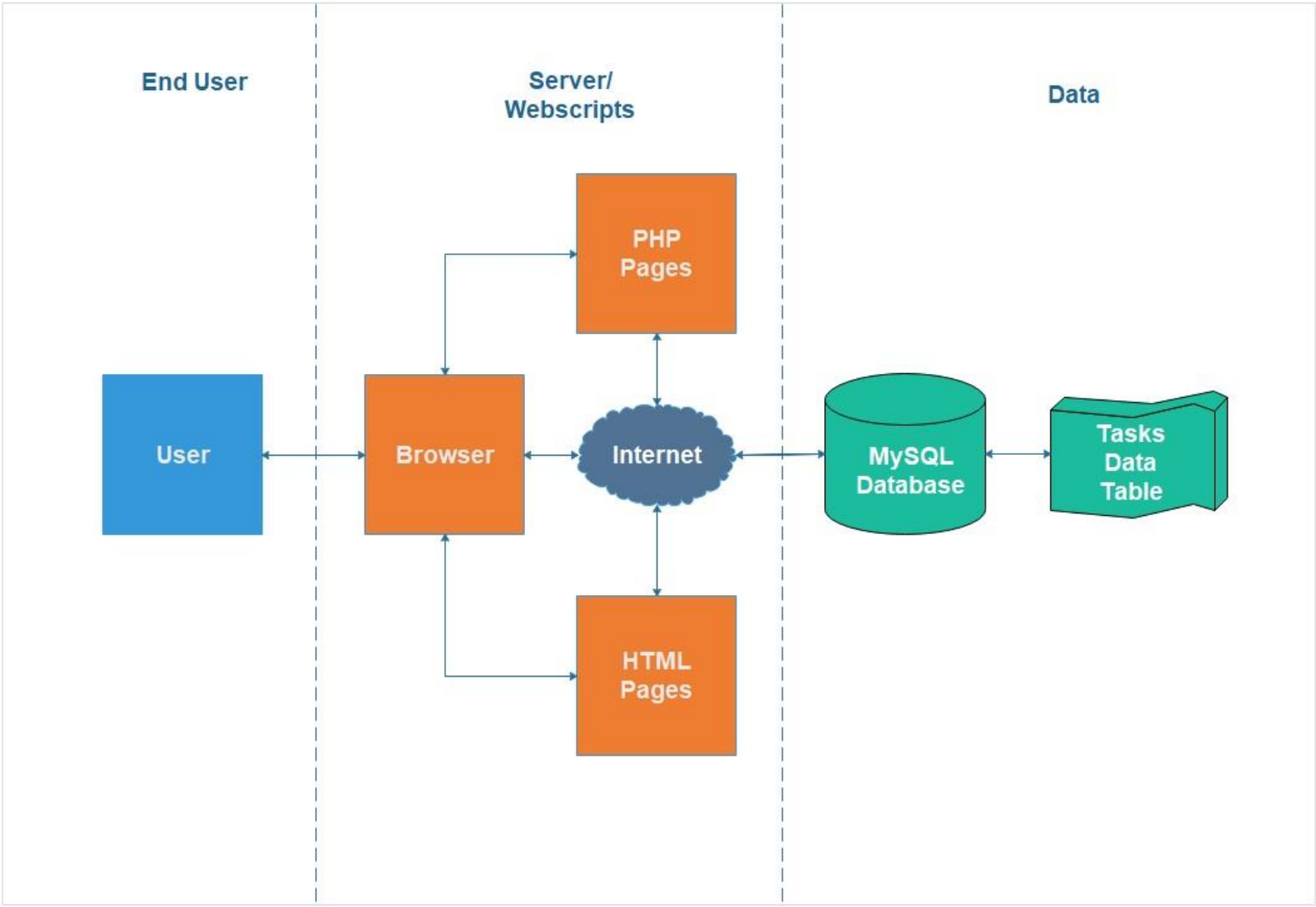
Requirement ID	Requirement Statement	Priority	Comments
FR001	Tasks are assigned name, type, due date	Required	Task types: Pending, Started, Complete, late
FR002	User can add tasks	Required	
FR003	User can delete tasks	Required	
FR004	User can view tasks	Required	
FR005	List must keep track of number of total tasks	Required	
FR006	List must keep track of different types of tasks	Required	Types of tasks: Pending, Started, Completed, Late
FR007	Clicking type count filters list by that type	Required	EX: Click pending task count filters list showing only pending tasks
FR008	Tasks added with due date after system date are automatically marked late	Optional	

Non-Functional Requirements

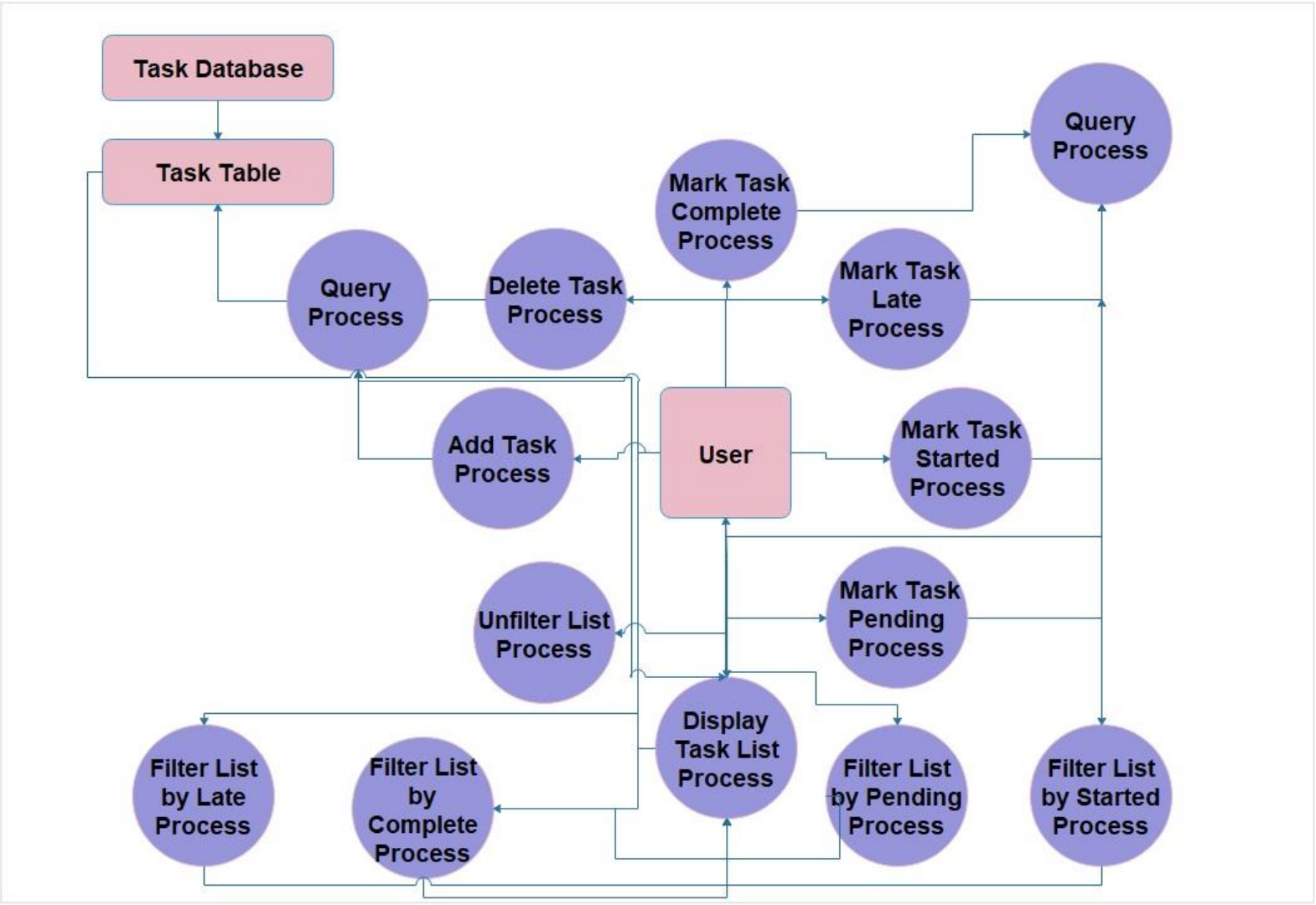
Requirement ID	Requirement Statement	Priority	Comments
NF001	To-do list stored on database back end	Required	Utilize MySQL , WAMP/XAMP stack
NF002	To-do list front end written in html/php	Required	Utilize WAMP/XAMP stack
NF003	To-do list has main page	Required	

NF004	Application due 1/15/2018 by 10:00am	Required	
NF005	To-do Utilize WAMP/XAMP Stack	Optional	List utilizes XAMP Stack

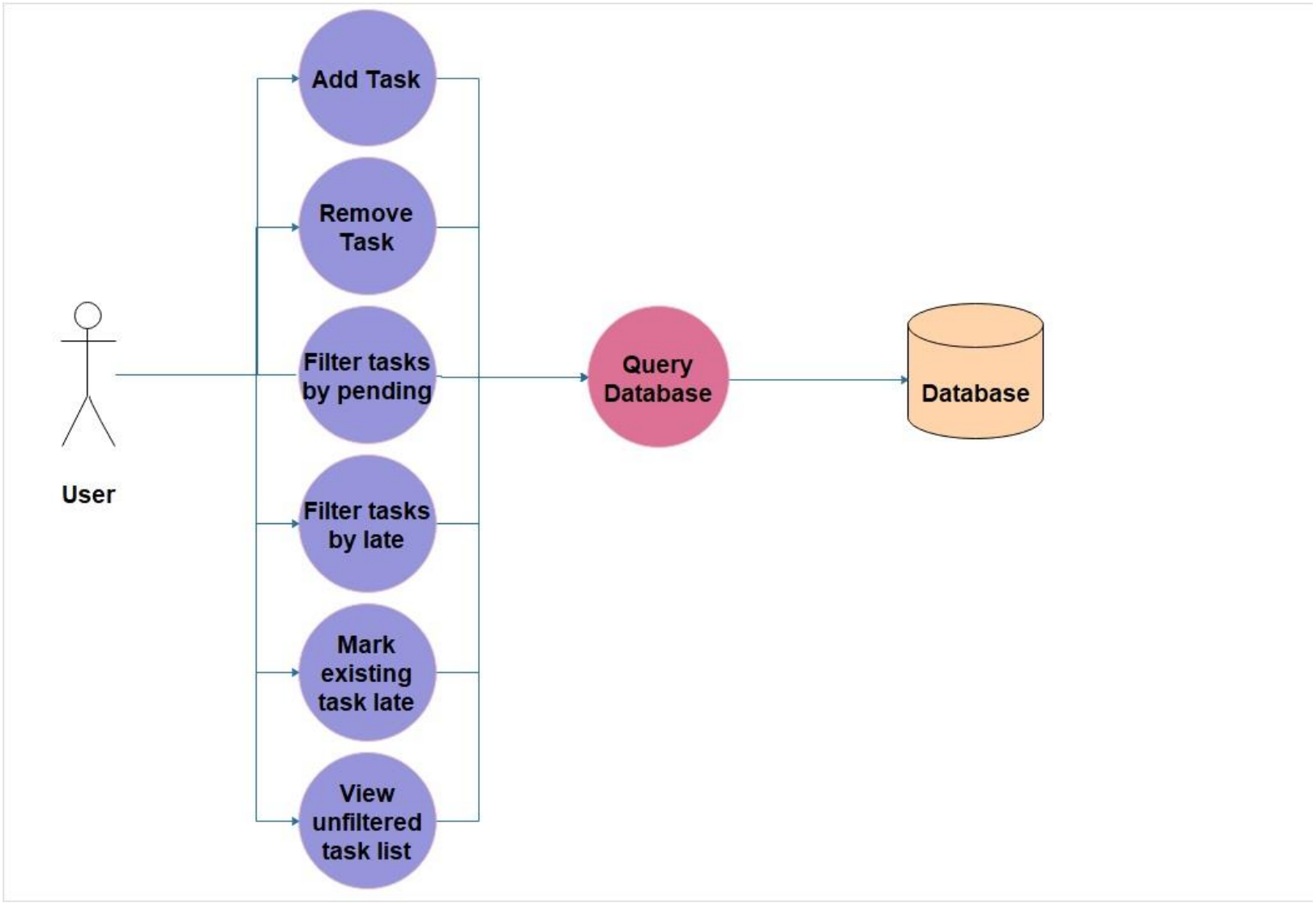
System Architecture Diagram



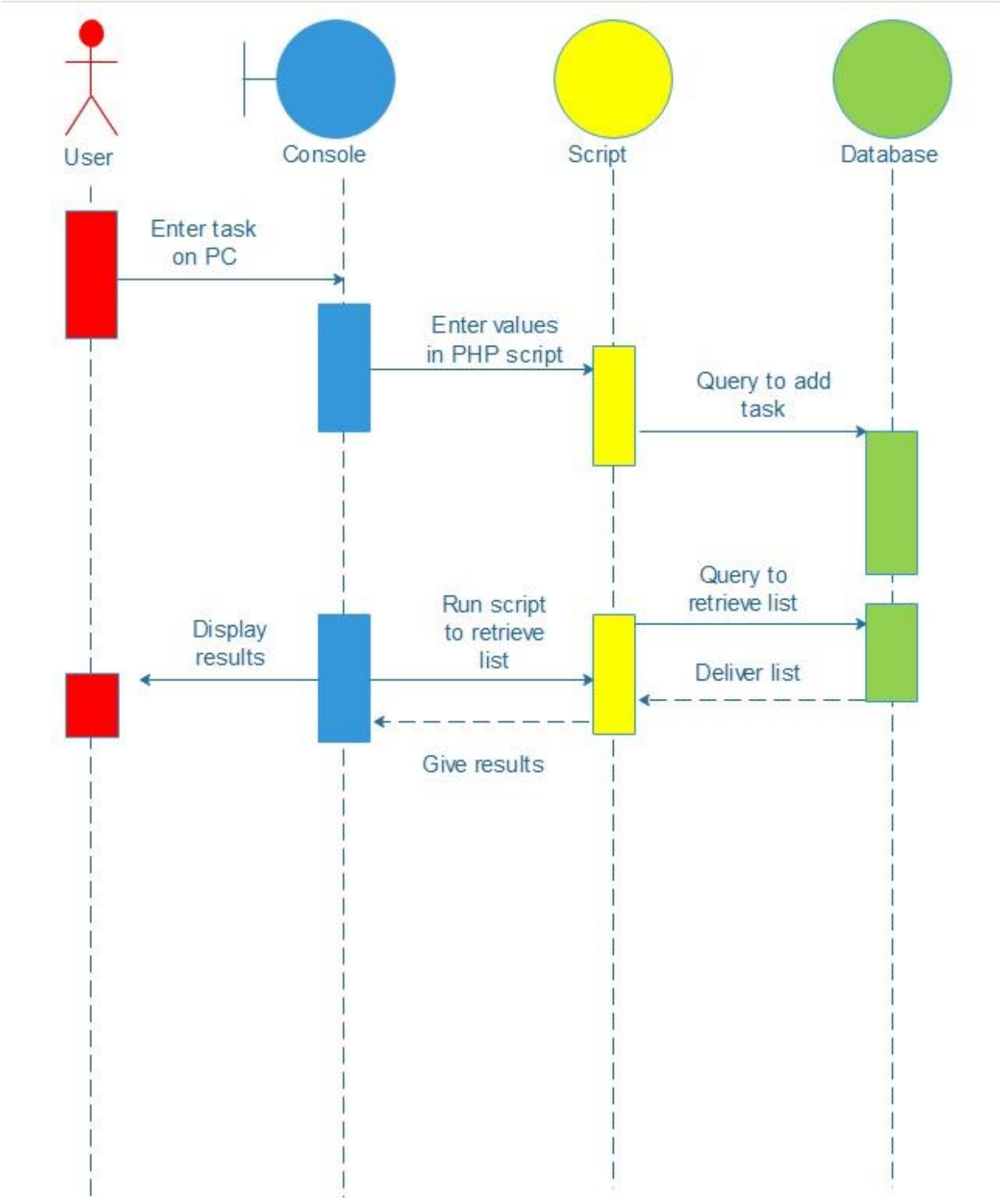
Data Flow Diagram



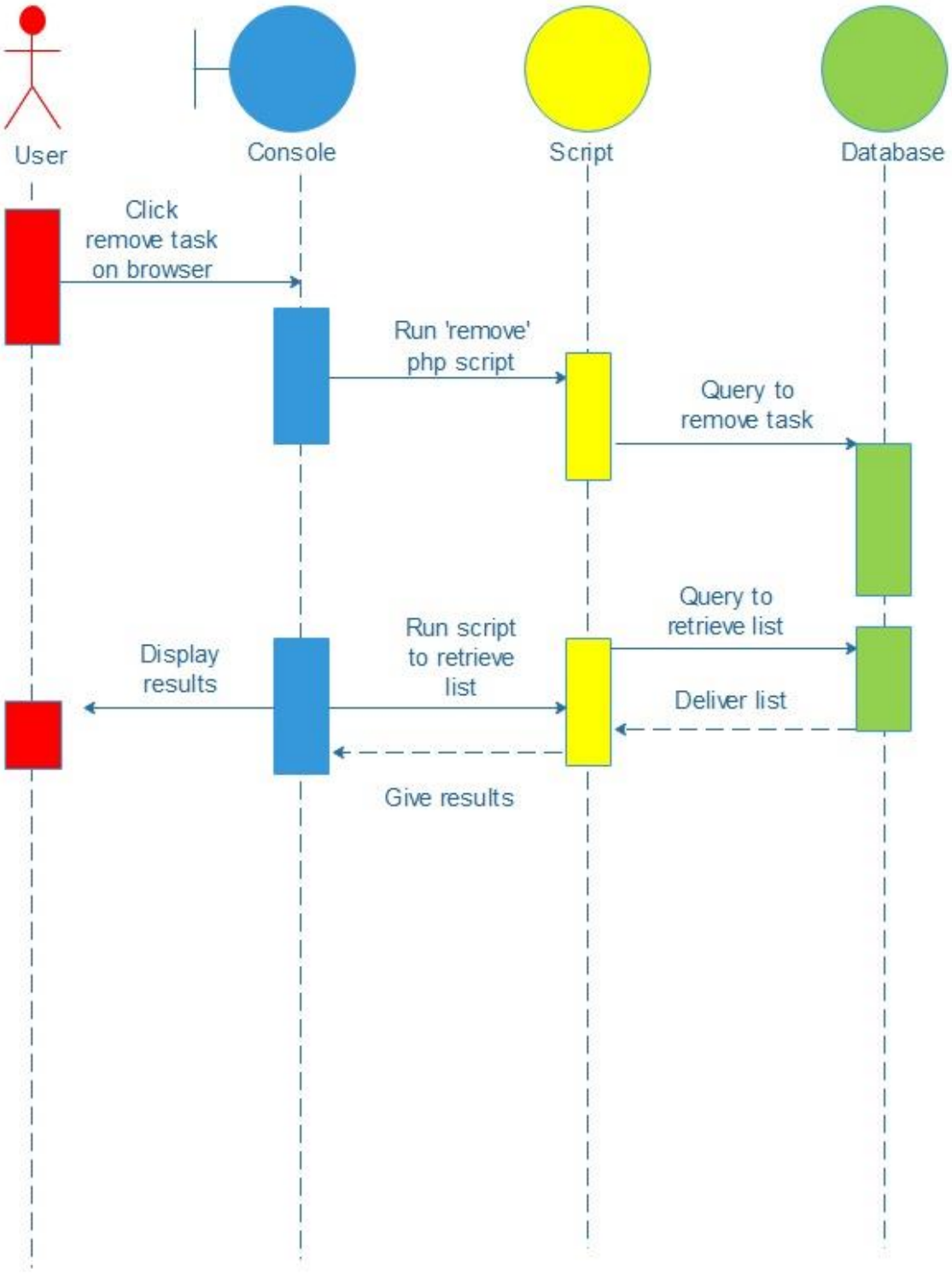
Use Case Diagram



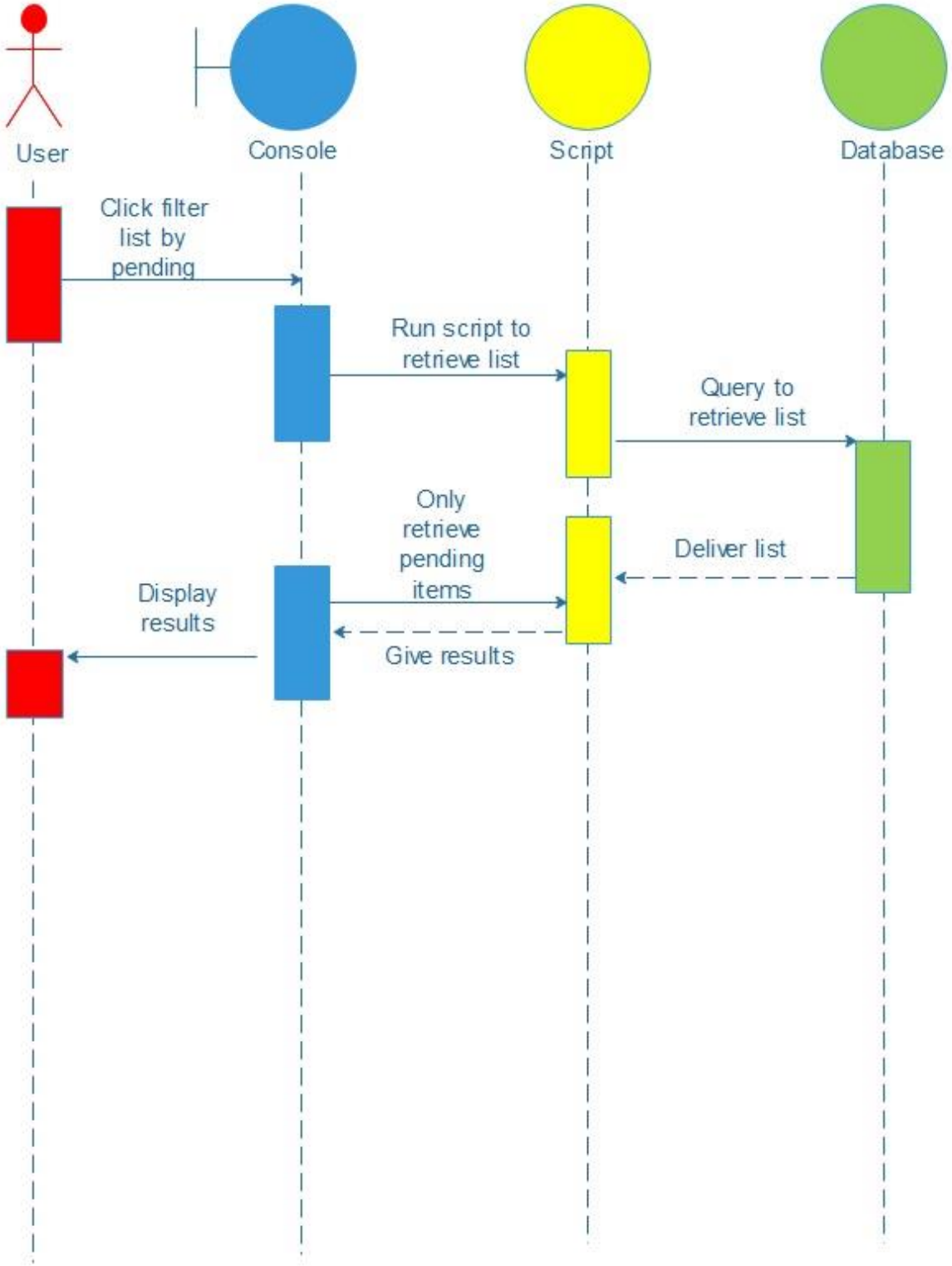
Sequence Diagram #1 (User enters task into list)



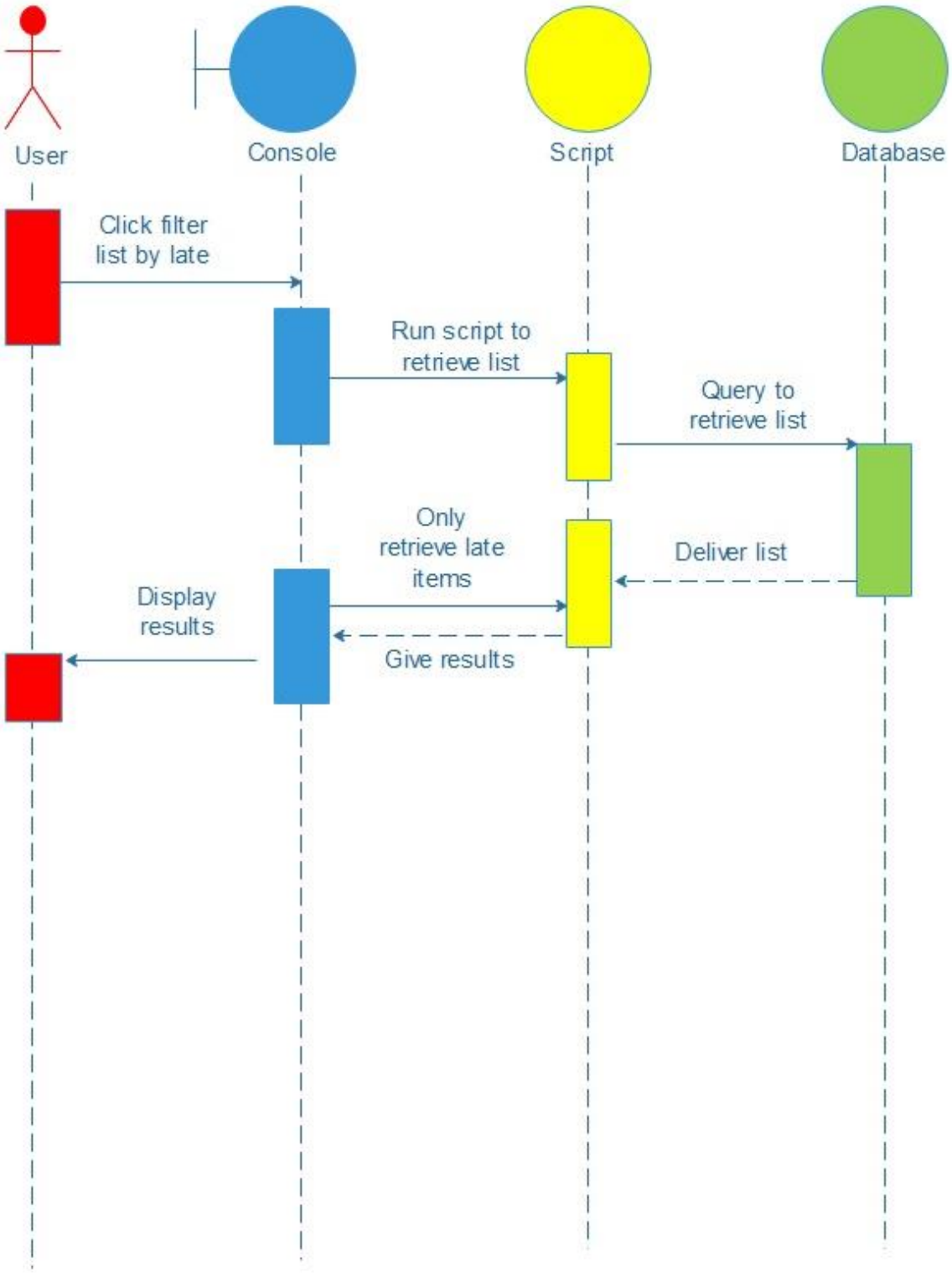
Sequence Diagram #2 (User removes task from list)



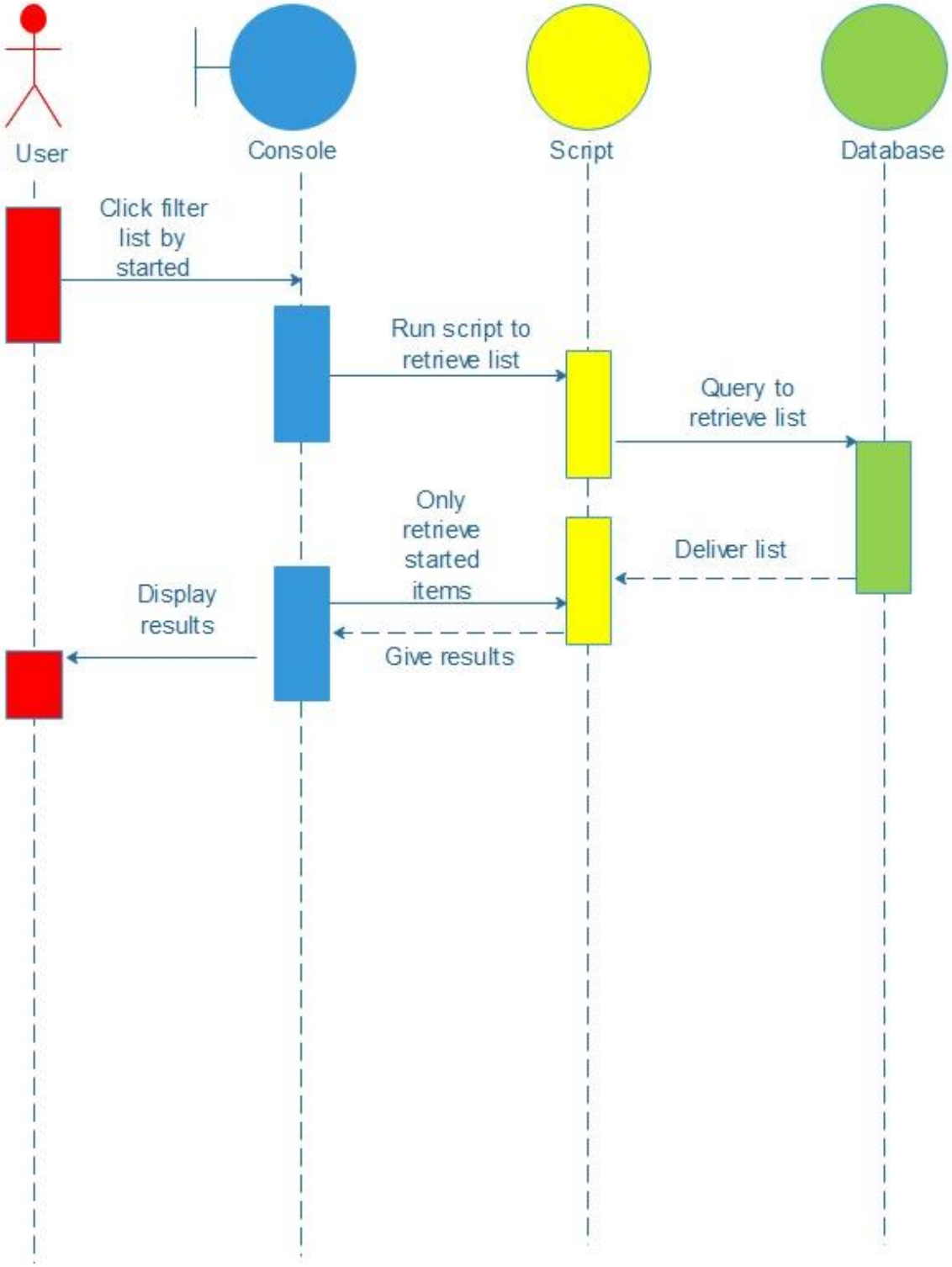
Sequence Diagram #3 (User filters list by pending)



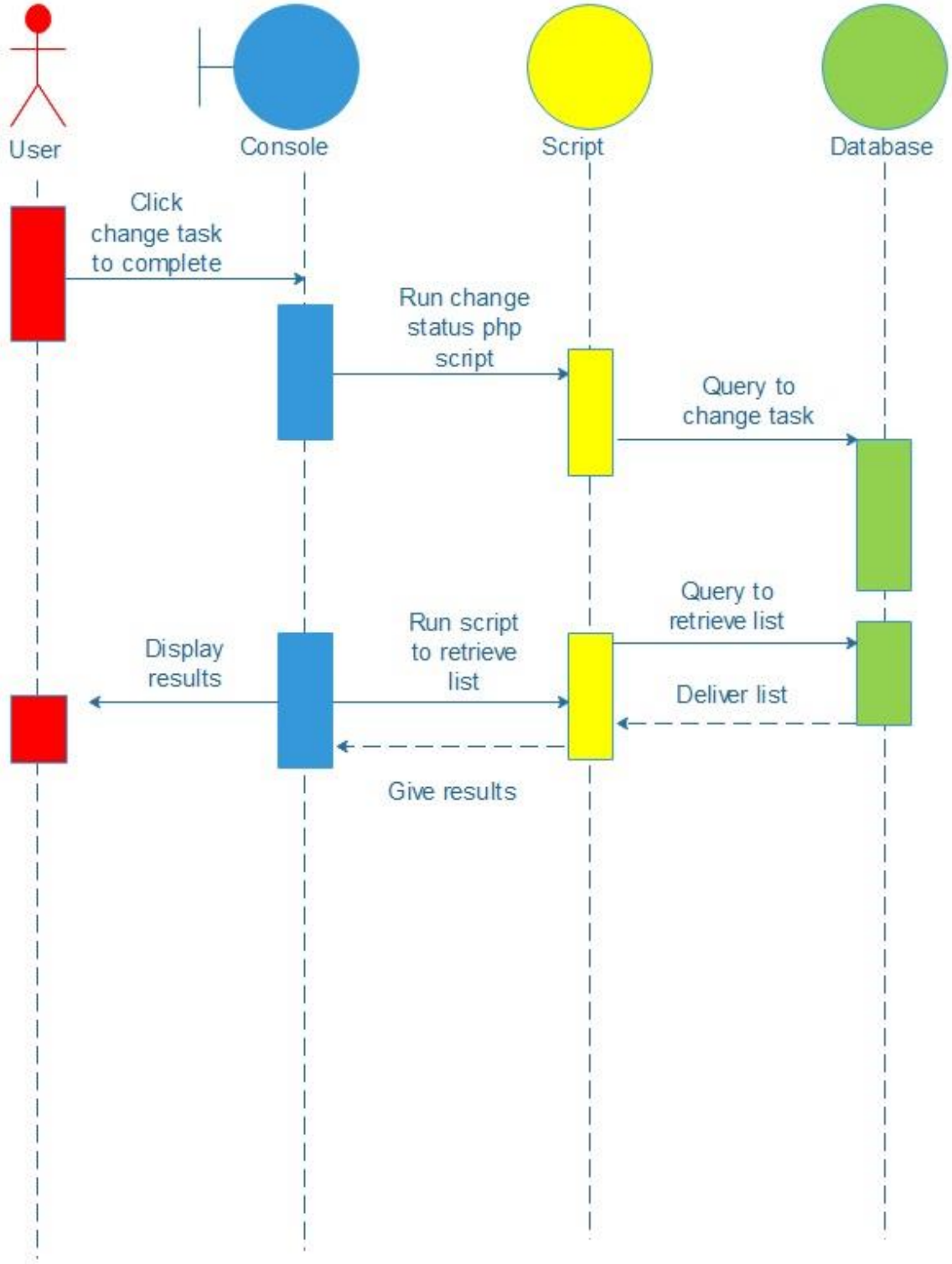
Sequence Diagram #4 (User filters list by late)



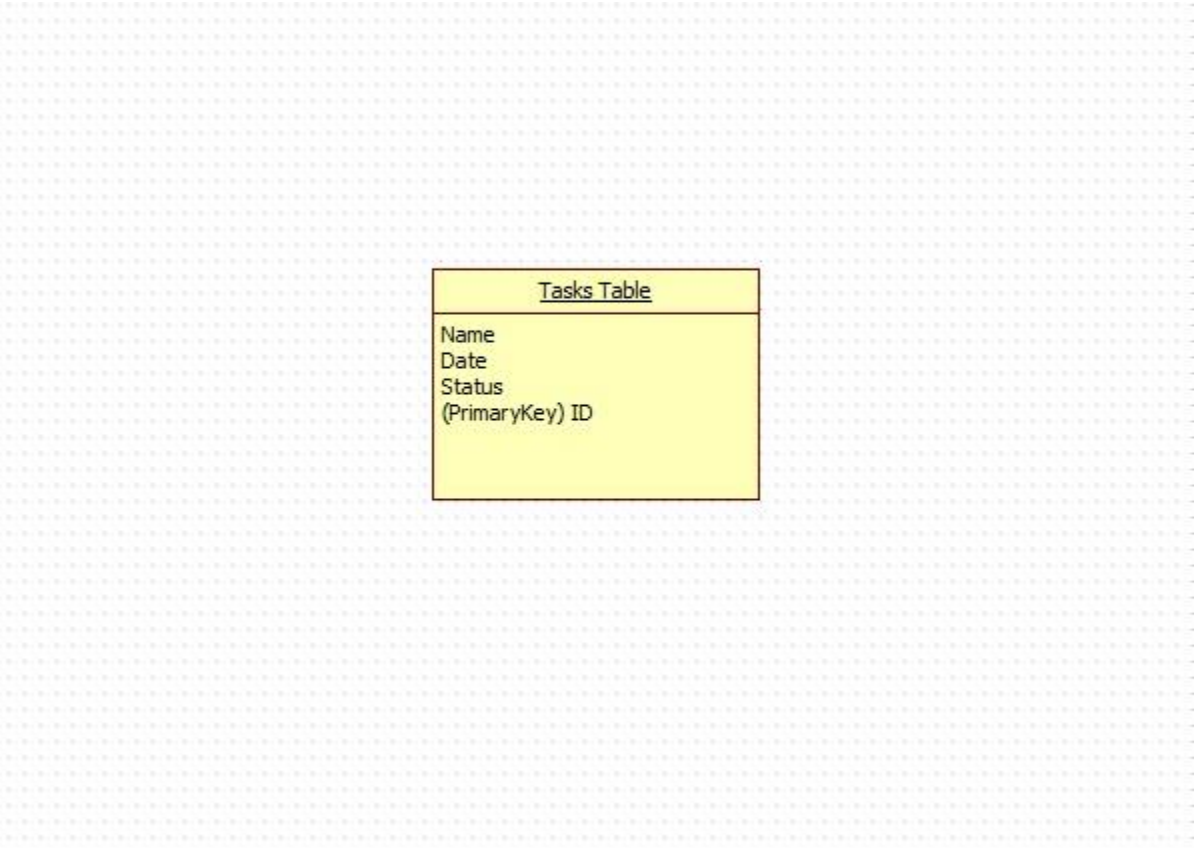
Sequence Diagram #5 (User filters list by started)



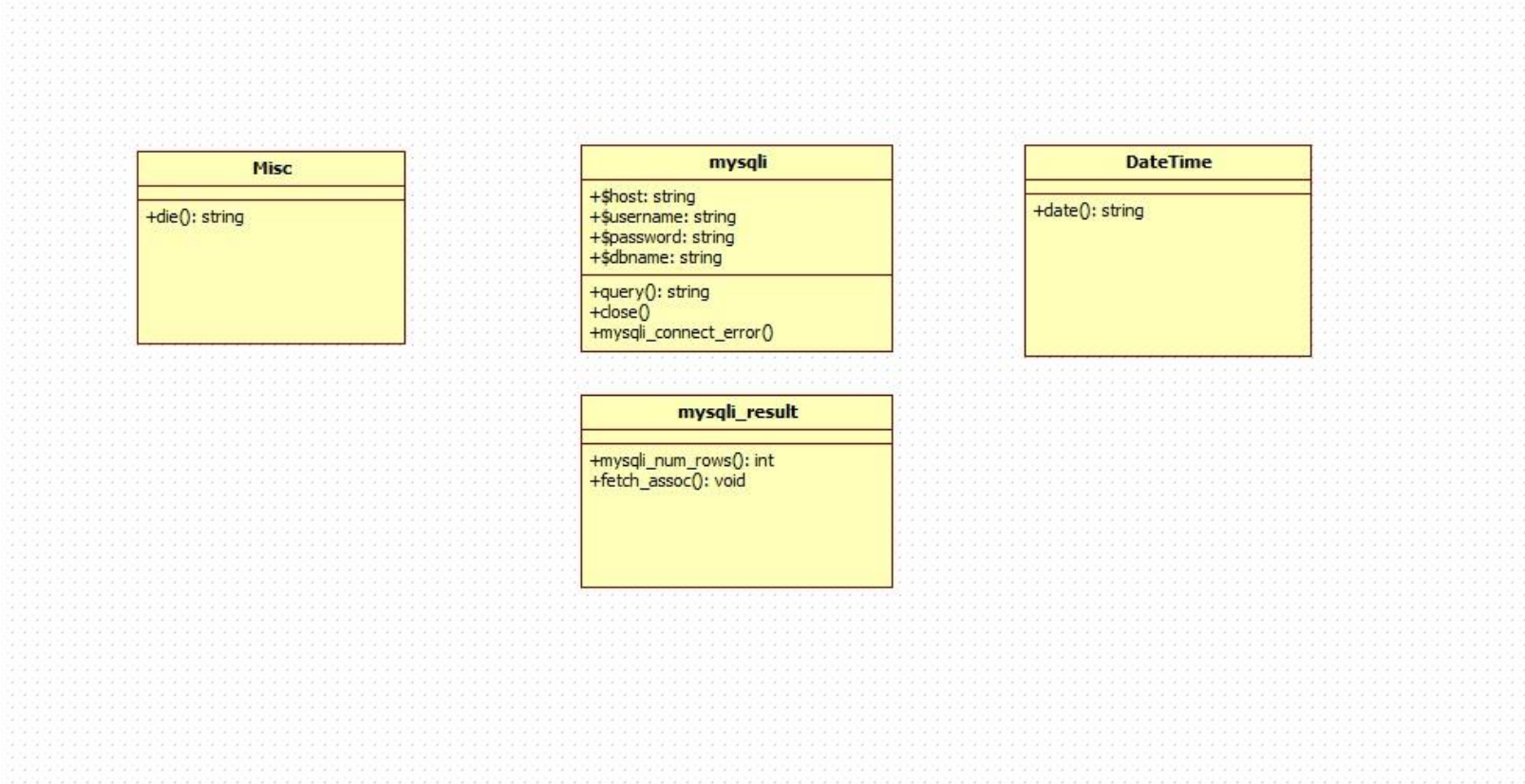
Sequence Diagram #6 (User changes existing task to complete status)



Database Diagram (For Tasks Database)



Class Diagram



Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
TU01	Check form submission with valid data	1. Load to-do list 2. Add Task Name to form 3. Select Started in status drop box 4. Add Date in date box 5. Click submit button	$taskname = “Vacuum” $taskdate = “2018-01-28” $taskstatus = “Started”	Task should be added to list as expected	As expected	Pass
TU02	Check form submission when fields are empty	1. Load to-do list 2. Leave fields blank 3. Click Submit	$taskname = “” $taskdate = “” $taskstatus = “Pending”	Entry will be added with blank task name, status will be pending, date will be represented by all zeroes ‘0’	As expected	Pass
TU03	Test task removal function	1. Load to-do list 2. Add task to list 3. Click “Remove” link next to task in list	$taskname = “Vacuum” $taskdate = “2018-01-28” $taskstatus = “Started”	Task will be removed from list, count will decrement by 1	As expected	Pass
TU04	Test Filter by Pending Function	1. Load to-do list 2. Add tasks to list with multiple status 3.Click “Filter by pending” link	Task 1 $taskname = “Vacuum” $taskdate = “2018-01-28” $taskstatus = “Started” Task 2 $taskname = “Clean” $taskdate = “2018-01-26” $taskstatus = “Pending”	List will be filtered to only display pending (showing task #2)	As expected	Pass
TU05	Add task with due date before current due date, verify it is marked late	1. Load to-do list 2. Add task to list with due date set before current date. Set status to “Started”	$taskname = “Vacuum” $taskdate = “2018-01-01” $taskstatus = “Started”	Task will be added to list, status will be overwritten from “Started” to “Late”	As expected	Pass
TU06	Filter empty list	1. Load to-do list 2. Click Filter by “Late” link	N/A	Page will filter but with nothing in list	As expected	Pass