

# **PLANIMAL**

## Use Case Specification

Submitted to:

Asst. Prof. Ma. Rowena C. Solamo  
Faculty Member  
Department of Computer Science  
College of Engineering  
University of the Philippines, Diliman

Submitted by:  
Bautista, Yohannah  
Bedia, Nicle Vynique  
Castillo, Algina

In partial fulfillment of academic requirements  
for the course  
CS 191 Software Engineering I  
of the  
1<sup>st</sup> Semester, AY 2014-2015

## **Revision Control**

### *History Revision:*

<b><i>Revision Date</i></b>	<b><i>Person Responsible</i></b>	<b><i>Version Number</i></b>	<b><i>Modification</i></b>
10/01/14	Yohannah Bautista Nicle Vynique Bedia Algina Castillo	1.0	Initial Document.
10/27/14	Nicle Vynique Bedia	1.1	Added clarifications in the Add Task use case about the application's calendar
12/08/14	Algina Castillo	1.2	Added "description" and replaced "time" and "date" with "deadline" in Use-Case 1.0 Scenario 1, i.e. Use-Case 1.1 ;Changed scenarios 3 and 4 of Use-Case 1.1; Added scenario 5 in use-case 1.2;Changed the activity diagrams of Use-Cases 1.1 and 1.2;

**Use-Case Name:** Use-Case 1.0 Maintain Schedule

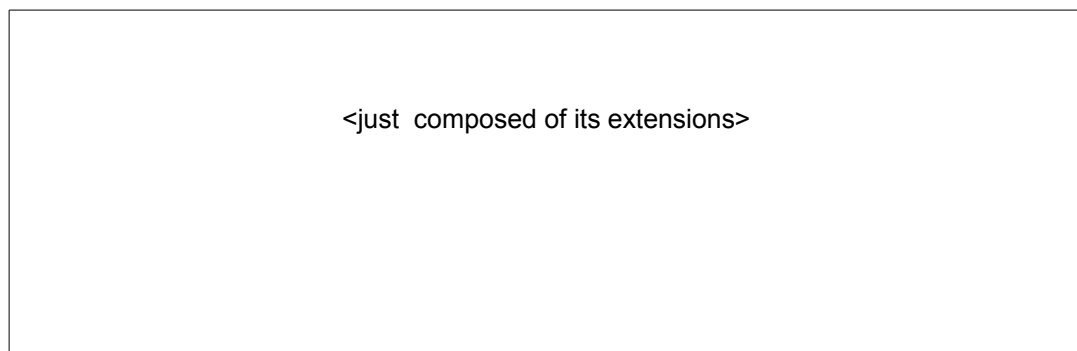
**Description:** The student must be able to keep track of the daily schedules in the planner, this means monitoring the tasks that were (or will *be*) *added, deleted, and edited*.

**Preconditions:** None

**Flow of Events:**

<b>Scenario Name</b>	<b>Description</b>
Scenario 1 (Basic Flow) Student adds task.	The student inputs a task in the planner. The task should have the following details: name, description, deadline, venue, and person-in-charge. (See Use-Case 1.1 for detailed how-to)
Scenario 2 Student edits task	In the instances where there are changes in a certain task, such as a change in venue or deadline, the student can edit the information in the task except the person-in-charge. (See Use-Case 1.2 for detailed how-to)
Scenario 3 Student deletes task	In the instances that an event related to a task is canceled, or perhaps the person-in-charge deems the task unnecessary for some reason, then the student can simply delete the task. (See Use-Case 1.3 for detailed how-to)

**Activity Diagram of the Flow of Events:**



**Postcondition:** None

**Relationships:** None

**Special Requirements:** None

**Use-Case Name:** Use-Case 1.1 Add task

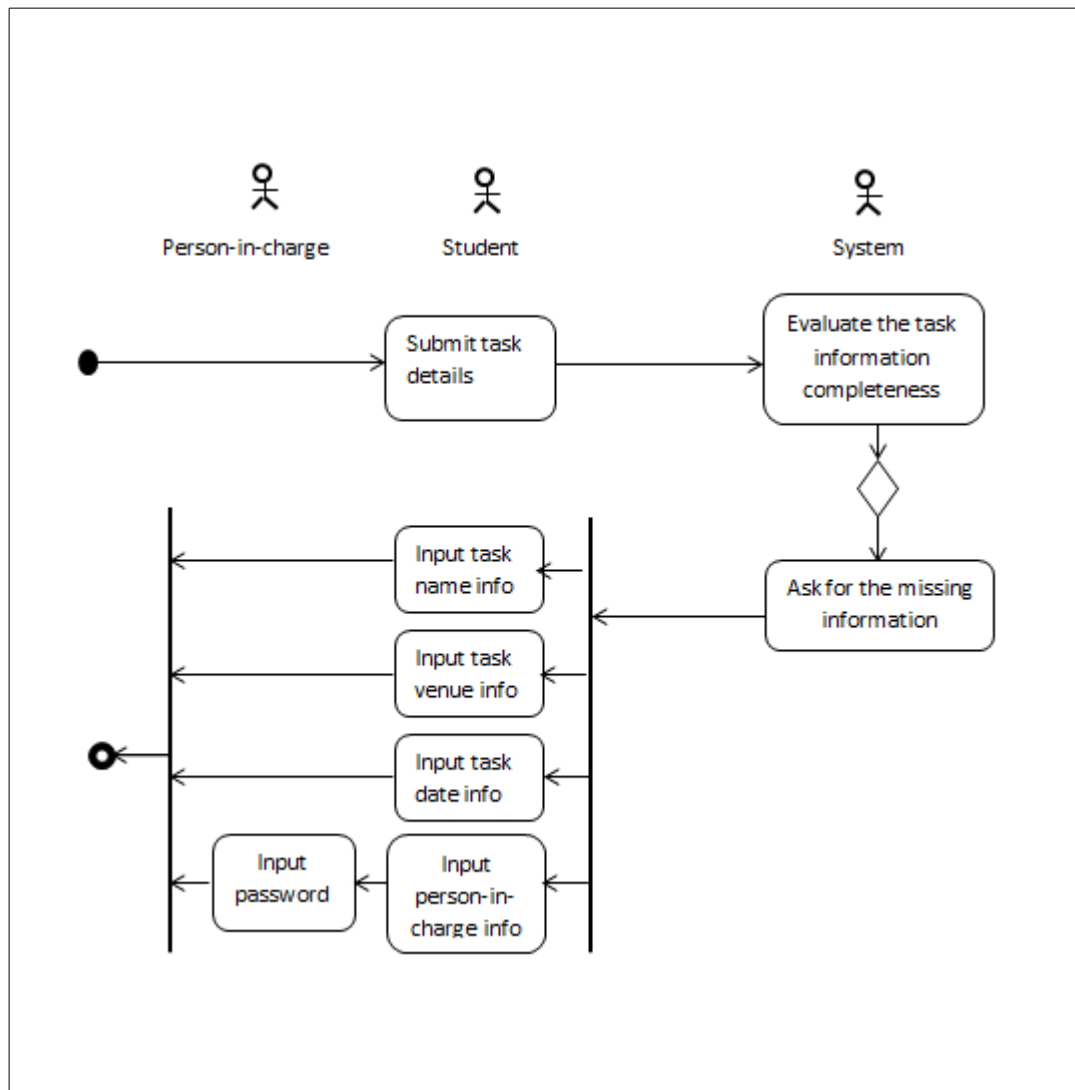
**Description:** The student inputs a task in the planner. The task should have a name, description, deadline, venue, and person-in-charge.

**Preconditions:** Schedule is not full

**Flow of Events:**

<b>Scenario Name</b>	<b>Description</b>
Scenario 1 (Basic Flow) Student submits complete information to the system.	The task is added to the schedule. The student is notified that the task has been successfully added to the schedule and in the application's calendar.
Scenario 2 Student did not specify the name of the task	The student will be asked to either continue adding the particular task or not. If he or she wishes to to continue, the system will ask the student to input the name of the task. Names are not required to be unique. After giving the information, the student is notified that the task has been successfully added to the schedule and in the application's calendar.
Scenario 3 Student did not put any description	The task is added to the schedule. The student is notified that the task has been successfully added to the schedule and in the application's calendar.  This behaves like the first scenario since description can be left blank. This is as good as complete.
Scenario 4 Student did not specify the deadline of the task	The student will be asked to either continue adding the particular task or not. If he or she wishes to to continue, the system will ask the student to input the deadline of the task. After giving the information, the student is notified that the task has been successfully added to the schedule and in the application's calendar.
Scenario 5 Student did not specify the venue of the task	The student will be asked to either continue adding the particular task or not. If he or she wishes to to continue, the system will ask the student to input the venue of the task. After giving the information, the student is notified that the task has been successfully added to his or her schedule and in the application's calendar.
Scenario 6 Student has no task's person-in-charge information	The student will be asked to either continue adding the particular task or not. If he or she wishes to to continue, the system will ask the student to input the name of the person-in-charge of the task. Once that's done, the person-in-charge needs to input a password that will be used to confirm the completion of the task. If the person-in-charge does not input a password, the task will not be added. Otherwise, the student is notified that the task has been successfully added to his or her schedule and in the application's calendar.

*Activity Diagram of the Flow of Events:*



*Postcondition:* NONE

*Relationships:* extends use case 1.0 Maintain Schedule

*Special Requirements:* NONE

**Use-Case Name:** Use-Case 1.2 Edit task

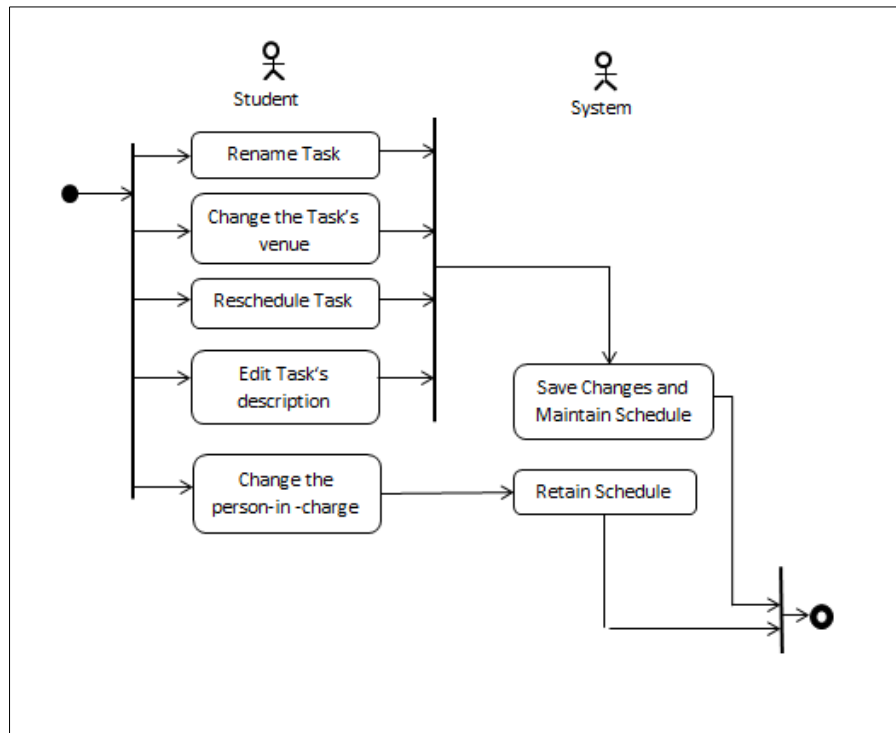
**Description:** In the instances where there are changes in a certain task, such as a change in venue or deadline, the student can edit the information in the task.  
The person-in-charge cannot change.

**Precondition:** The task exists

**Flow of events:**

<b>Scenario Name</b>	<b>Description</b>
Scenario 1 (Basic Flow) The student wants to rename the task	The system displays the task's details in editable fields. The student edits the name and then selects 'Save Changes'. The system saves the changes and displays the task's updated details.
Scenario 2 The student wants to edit the task's venue or location	The system displays the task's details in editable fields. The student edits the venue and then selects 'Save Changes'. The system saves the changes and displays the task's updated details.
Scenario 3 The student wants to reschedule the task	The system displays the task's details in editable fields. The student can edit the date of the task, or the time of the task, or both and select 'Save Changes'. After selecting 'Save Changes', the system saves the changes made and displays the updated task's details.
Scenario 4 The student wants to change the person-in-charge	The system notifies the student that the person-in-charge can't be changed. The task details remains the same.
Scenario 5 The student wants to add or modify the task's description	The system displays the task's details in editable fields. The student edits the description and then selects 'Save Changes'. The system saves the changes and displays the task's updated details.

*Activity Diagram of the Flow of Events:*



*Postcondition:* NONE

*Relationships:* extends use case 1.0 Maintain Schedule

*Special Requirements:* NONE

**Use-Case Name:** Use-Case 1.3 Delete task

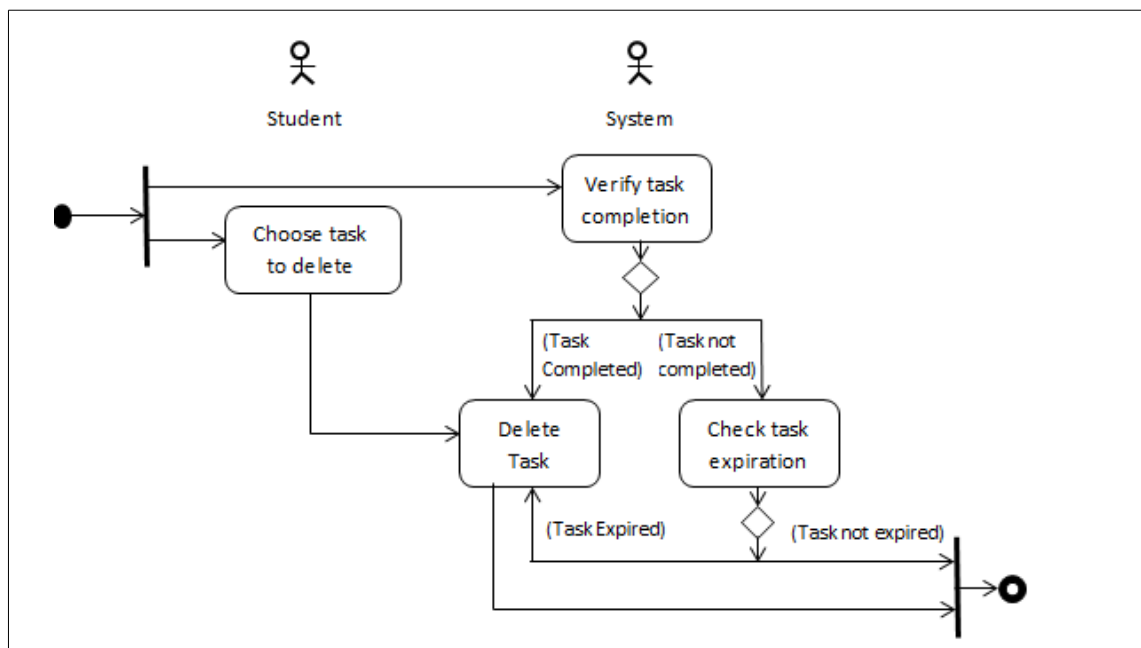
**Description:** In the instances that an event related to a task is canceled, or perhaps the person-in-charge deems the task unnecessary for some reason, then the student can simply delete the task.

**Preconditions:** The task exists

**Flow of Events:**

Scenario Name	Description
Scenario 1 (Basic Flow) The student deletes a task	The student selects the task that he or she wants to delete. The system permanently removes the task from the student's schedule.
Scenario 2 The system deletes the task	This could be result of the task's completion or expiration (the student has missed its deadline).  The task is deemed complete upon the submission of password by the person-in-charge.  When the task exceeds its deadline, its date and time, it is automatically deleted from the schedule.

**Activity Diagram of the Flow of Events:**



**Postcondition:** NONE

**Relationships:** extends use case 1.0 Maintain Schedule

**Special Requirements:** NONE



**Use-Case Name:** Use-Case 2.0 Buy pet's necessities

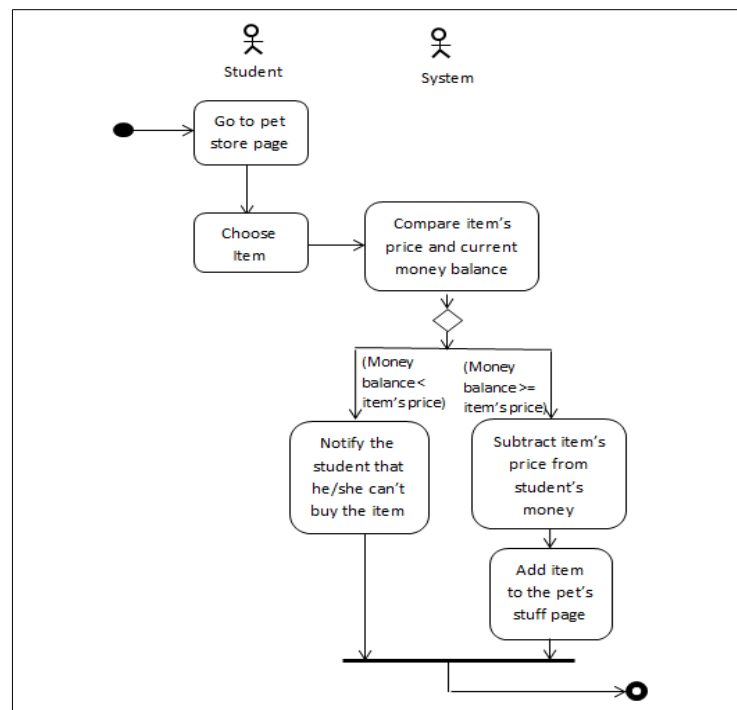
**Description:** Caring for the pet means purchasing items needed for the pet's continued survival and happiness. Purchasing items require money, something that the student can only get through completing tasks.

**Preconditions:** Student already owns a pet and the student has enough money to buy an item in pet store.

**Flow of Events:**

Scenario Name	Description
Scenario 1 (Basic Flow) Student having sufficient money on hand buys an item for his or her pet.	The student checks the items in the pet shop and chooses a specific pet item. Upon selecting the item he or she wants, the system, considering its price is less than or equal to the student's current money balance, updates the money balance left and the item now belongs to the student (or the pet, technically). The student can buy as many items as he or she wants as long as he or she can afford it.
Scenario 2 Student wants to buy an item for his or her pet but is short of money	Upon selecting the item that he or she currently cannot afford, the student is notified that the item cannot be purchased.  The student is then advised to complete tasks within a certain time in order to buy that item.
Scenario 3 Student has considerable amount of money to buy an item for his or her pet but has not bought any item yet	If the pet needs an item, the student is reminded that the pet is in need of something and therefore must purchase it since he or she can now afford it; otherwise, the student can save the money for future needs.

**Activity Diagram of the Flow of Events:**



**Postcondition:** NONE

**Relationships:** includes use case 5.1 Subtract Amount

**Special Requirements:** NONE

**Use-Case Name:** Use-Case 3.0 View Schedule

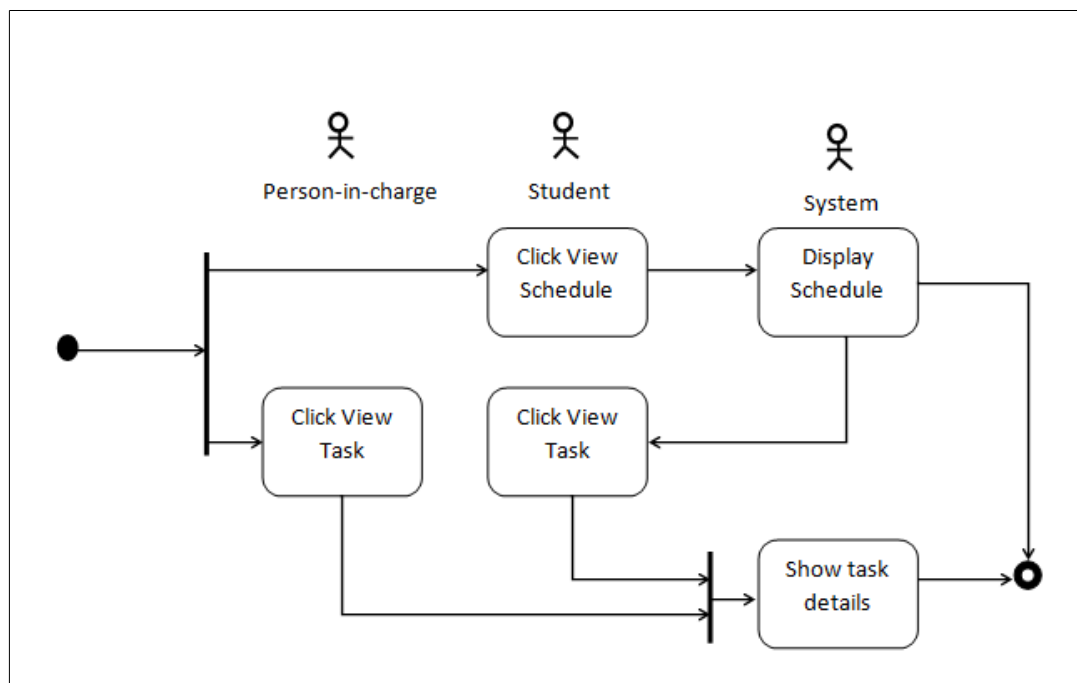
**Description:** The student, and only the student, can view his or her complete schedule. The person-in-charge will only see the task that he or she assigned to the student.

**Preconditions:** None

**Flow of Events:**

Scenario Name	Description
Scenario 1 (Basic Flow) Student views the schedule	The student is shown a schedule for the week. The schedule consists of the tasks he or she has saved. The tasks would only show some basic information such as its name, description(if any) and deadline.

**Activity Diagram of the Flow of Events:**



**Postcondition:** NONE

**Relationships:** NONE

**Special Requirements:** This feature is linked to the phone's date and time.

**Use-Case Name:** Use-Case 3.1 View Task

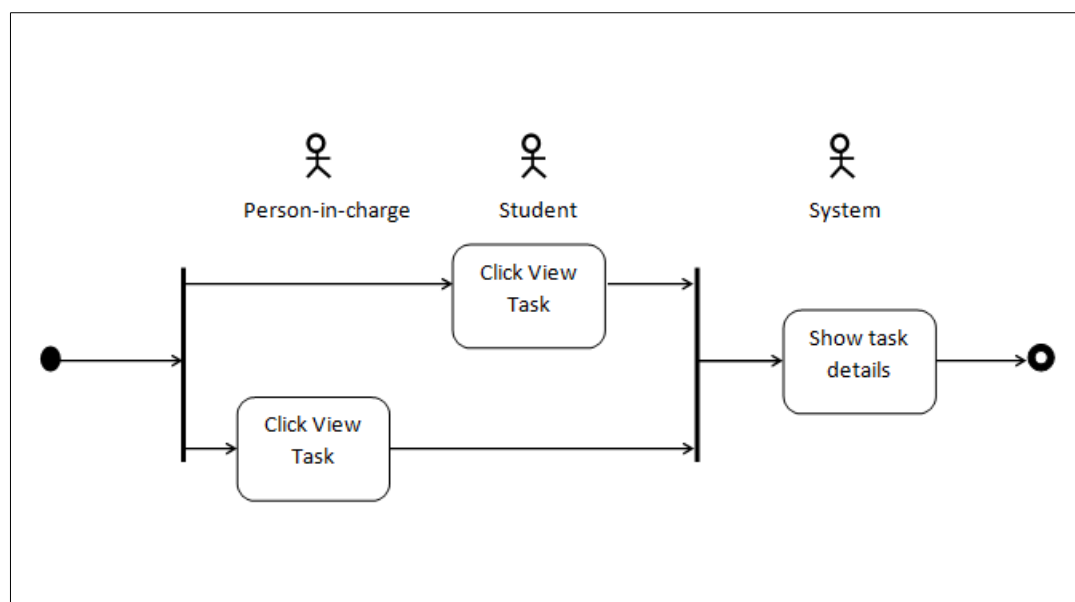
**Description:** The student can view any task in his or her schedule. The person-in-charge, on the other hand, will only see the task that he or she assigned to the student.

**Preconditions:** The task exists

**Flow of Events:**

Scenario Name	Description
Scenario 1 (Basic Flow) Student views a task	The student selects the task, using the task name, he or she is interested in.  The system will show the task's details such as the description, deadline, venue, and person-in-charge.
Scenario 2 Person-in-charge views a task	The person-in-charge gets to see the task when he or she inputs the password for the task or for confirming its completion. The person-in-charge can only view tasks he or she is in charge of.

**Activity Diagram of the Flow of Events:**



**Postcondition:** NONE

**Relationships:** extends use case 3.0 View Schedule

**Special Requirements:** This feature is linked to the phone's date and time.

**Use-Case Name:** Use-Case 4.0 Input password

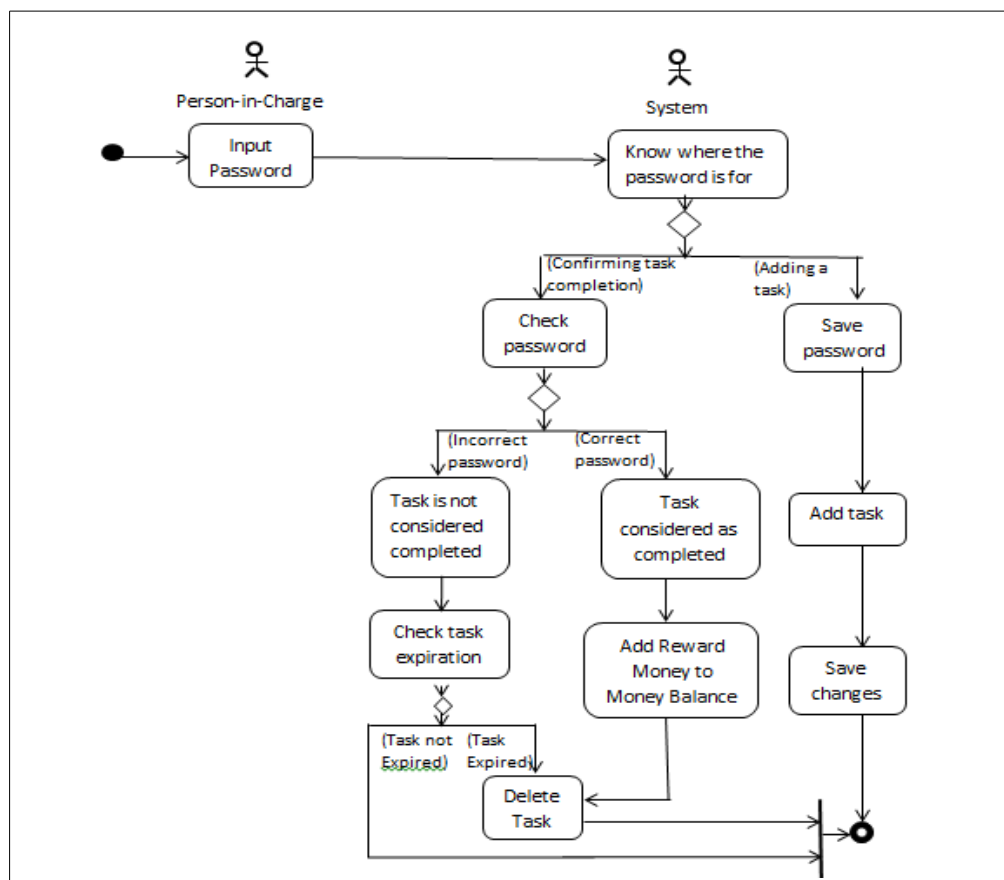
**Description:** When the student adds a task, the person-in-charge of that task needs to input a password. This will serve as the verification key later if that particular task is accomplished by the student.

**Preconditions:** If password input is for confirmation of completion, then the task should already be existing.

**Flow of Events:**

Scenario Name	Description
Scenario 1 (Basic Flow) The person-in-charge inputs the right password	If the person-in-charge is giving a password for the creation of a task, then there is no need to check the password. If the person-in-charge is giving a password for the confirmation of a task, assuming that the password is correct, the system will remove the task from the schedule and the student will be given a certain amount of money as a reward.
Scenario 2 The person-in-charge inputs a wrong password	The system will notify the person-in-charge that the password entered is incorrect. It will then ask for password until the person-in-charge inputs the correct password. The person-in-charge can opt to cancel submitting the password, this will leave the task incomplete. The student will not be given money as a reward and the task will remain in the schedule.

**Activity Diagram of the Flow of Events:**



**Postcondition:** NONE

**Relationships:** includes use case 1.1 Add Task

*Special Requirements:* NONE

*Use-Case Name:* Use-Case 5.0 Update Money Balance

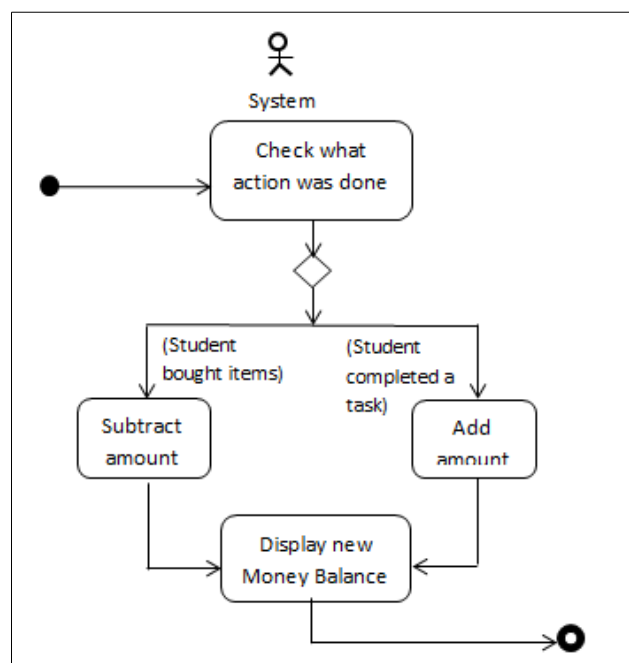
*Description:* A student gets money from completing tasks, and loses money from buying pet items. The money balance must be updated frequently.

*Preconditions:* A task has been completed or an item was bought from the in-game pet store.

*Flow of Events:*

Scenario Name	Description
Scenario 1 (Basic Flow) The system adds amount to money balance of the student	The system adds an amount of money, which is the reward money from completing a task, to the current balance. The system then displays the student's new money balance.
Scenario 2 The system subtracts amount from money balance of the student	The system subtracts an amount of money, which is based on an item's price, to the current balance. The system then displays the student's new money balance.

*Activity Diagram of the Flow of Events:*



*Postcondition:* NONE

*Relationships:* NONE

*Special Requirements:* NONE

**Use-Case Name:** Use-Case 5.1 Subtract amount

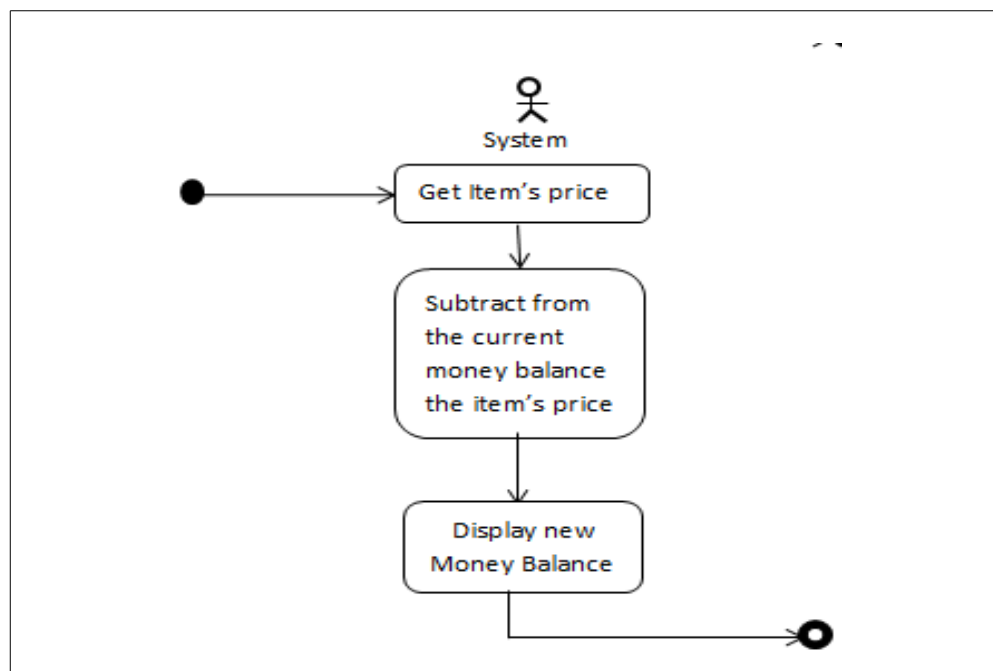
**Description:** If a student purchases items in the pet store, then the money balance will decrease.

**Preconditions:** The current money balance is greater than or equal to the amount that is to be subtracted

**Flow of Events:**

Scenario Name	Description
Scenario 1 (Basic Flow) The student buys an item from the store	The system subtracts that some amount of money, which is the item's price the student bought from pet store, to the current balance.  The system then displays the student's new money balance.

**Activity Diagram of the Flow of Events:**



**Postcondition:** NONE

**Relationships:** extends use case 5.0 Update Money Balance

**Special Requirements:** NONE

**Use-Case Name:** Use-Case 5.2 Add Amount

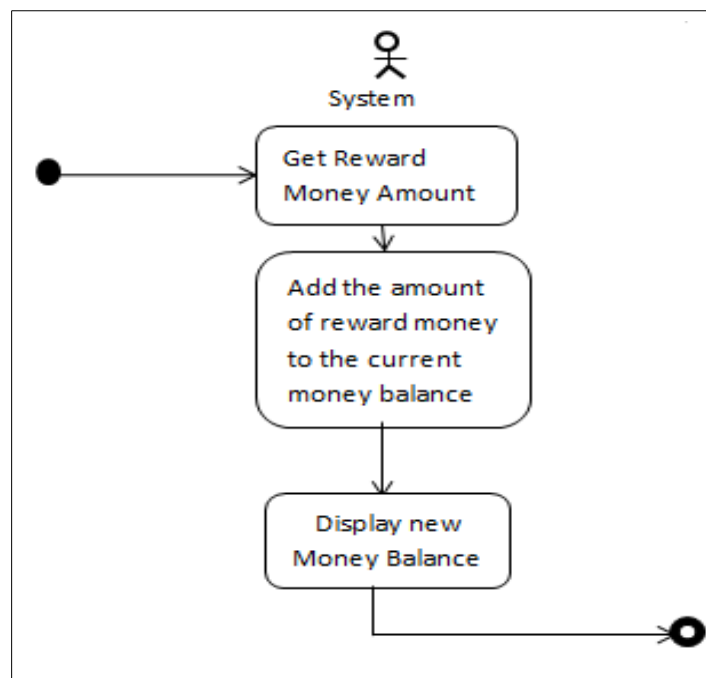
**Description:** If a student completes a task, then the money balance will increase. The student can gain more money by accepting more difficult tasks.

**Preconditions:** The task was completed and completion has been confirmed

**Flow of Events:**

Scenario Name	Description
Scenario 1 (Basic Flow) Some amount of money is added to current money balance of student	The system adds an amount of money, which is the reward money from completing a task, to the current balance. The system then displays the student's new money balance.

**Activity Diagram of the Flow of Events:**



**Postcondition:** NONE

**Relationships:** extends use case 5.0 Update Money Balance

**Special Requirements:** NONE

**Use-Case Name:** Use-Case 6.0 Confirm completion of task

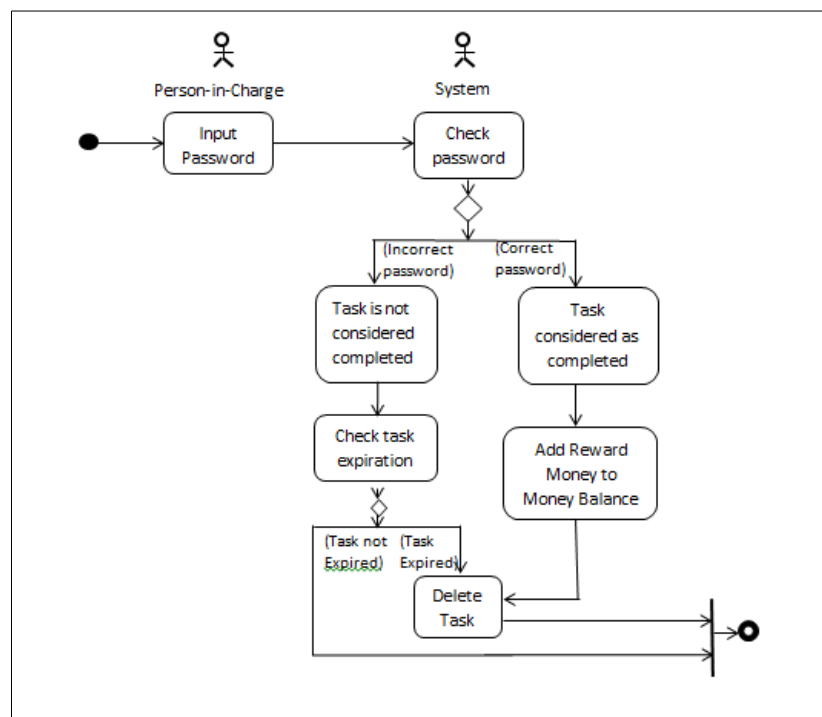
**Description:** The person-in-charge is the only one who can confirm if a task is completed. If the person does not input the password for the task, then that task will remain incomplete, and the student will be unable to reap the rewards in that task.

**Preconditions:** The task exists

**Flow of Events:**

Scenario Name	Description
Scenario 1 (Basic Flow) The task is confirmed finished	Upon the submission of the correct password by the person-in-charge, the system marks the task as finished and removes it from the schedule. The student then receives reward money for the accomplishment which results to the money balance's amount increasing due to the reward.
Scenario 2 The task is not confirmed to be done	This will most likely be due to entering incorrect password or not entering the password at all.  The system marks the task as incomplete and the student gains nothing from this.

**Activity Diagram of the Flow of Events:**



**Postcondition:** NONE

**Relationships:** includes use case 1.3 Delete Task, 3.1 View Task, 4.0 Input Password, and 5.2 Add amount

**Special Requirements:** NONE