XCRAM
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:
Agluba, Gerry Jr. P.
Go, Sharleen Joy Y.
Silverio, Robelle C.

In partial fulfillment of academic requirements
for the course
CS 191 Software Engineering I
of the
1st Semester, AY 2016-2017

Unique Reference:

The documents are stored in the https://github.com/sharleengo.

https://github.com/sharleengo/XCRAM/blob/master/01-Project-Documents/1.1%20-%20Add%20Task.pdf.

Document Purpose:

The purpose of this documentation is to give a description and explain the preconditions, flow events, postconditions, relationships with other use-cases and special requirements of Use-Case 1.1 Add Task found in the use-case model of the Task Scheduling System.

Target Audience:

Evaluators and Users

Revision Control

History Revision:

Revision Date	***Person Responsible***	***Version Number***	***Modification***
09/30/16	Gerry P. Agluba Jr.	1.0	Initial Document.
10/07/16	Gerry P. Agluba Jr.	1.1	Adds scenario 4,5 . Edit use-case diagram minimally.

Use-Case Name: 1.1 User adds task

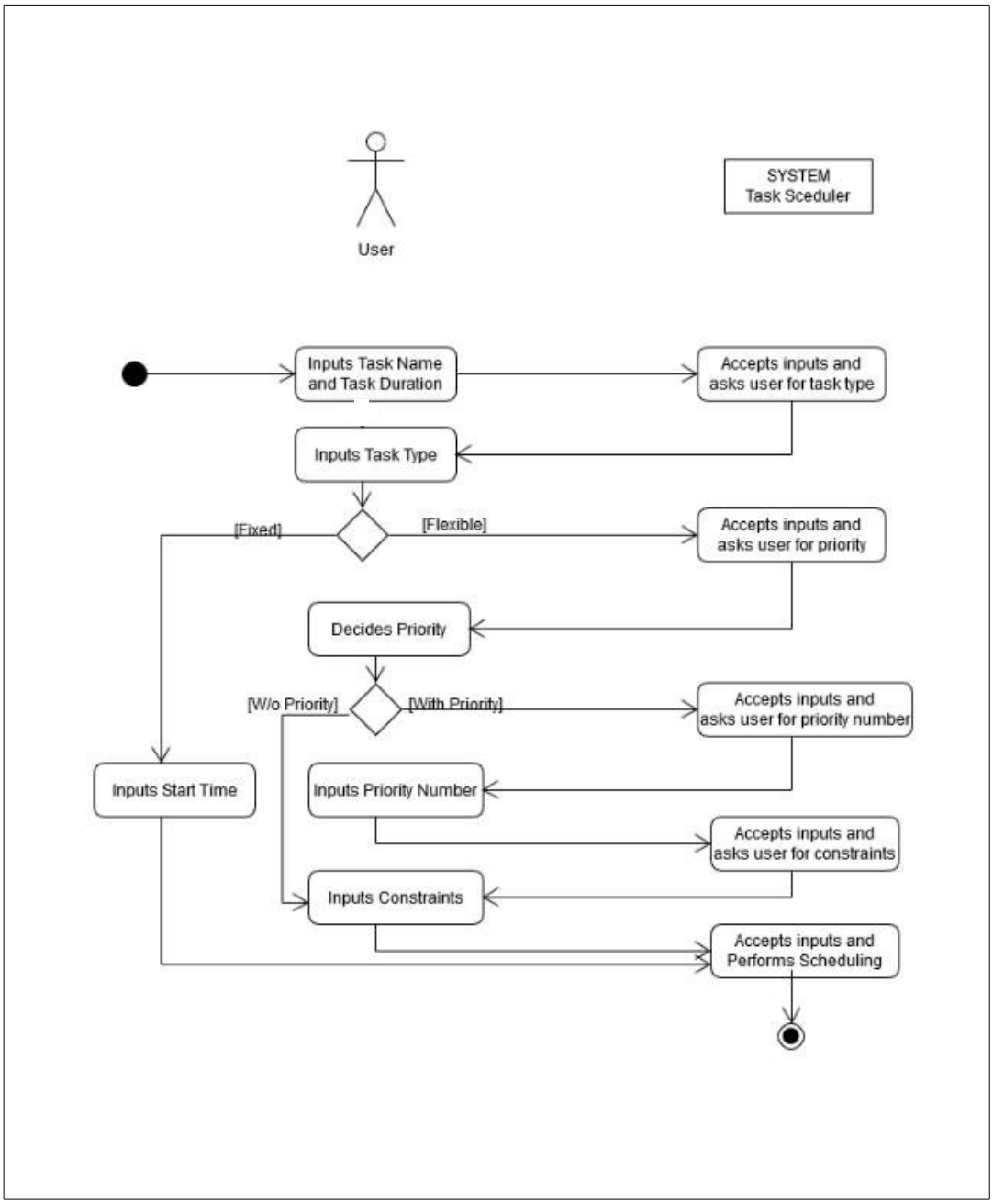
Description: The purpose of this use-case is to allow the user to add a new task to the current schedule. The user will input the task title, type (fixed or flexible: fixed tasks must be done on a specific time while flexible tasks may be done at any time during the day but may be controlled to some extent by adding constraints), and duration. If the inputted task was fixed, the user will be asked to enter the start time of the task. If the inputted task was flexible, the user will be given options to add a priority and constraints such as a preferred time range for the task to be done and the possibility of partitioning the said task.

Preconditions: NONE

Flow of Events:

Scenario Name	***Description***
Scenario 1 (Basic Flow) User enters fixed task.	1. User inputs task name and task duration. 2. Scheduler accepts the input and asks if the task is flexible or fixed. 3. User selects fixed type . 4. User inputs preferred time. 5. Scheduler accepts the input and perform the scheduling.
Scenario 2 User enters flexible task without priority	1. User inputs task name and task duration. 2. Scheduler accepts the input and asks if the task is flexible or fixed. 3. User selects flexible type . 4. Scheduler accepts input and ask if the user wants priority. 5. User decides no priority. 6. Scheduler accepts input and asks for constraints. 7. User inputs constraints if there are any. 8. System accepts the input and perform the scheduling.
Scenario 3 User enters flexible task with priority	1. User inputs task name and task duration. 2. Scheduler accepts the input and asks if the task is flexible or fixed. 3. User selects flexible type . 4. Scheduler accepts input and ask if the user wants priority. 5. User decides with priority. 6. Scheduler asks for priority number. 7. User enters priority number. 8. Scheduler accepts Priority number and asks for constraints. 9. User inputs constraints if there are any. 10. System accepts the input and perform scheduling.
Scenario 4 User fail to enter task name or task duration	1.User enter a blank task name or a task duration 2. System reject the input and returns an error message.
Scenario 5 User failed to enter start time for fixed task	1. User enter task name and task duration 2. User chooses fixed task 3. User enter a blank start time(fixed task) 4. System reject the input and return an error message

Activity Diagram of the Flow of Events:



Postcondition: If a task is added. It mus be reflected to the current schedule.

Relationships: NONE

Special Requirements: NONE