Sked-it
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:
Dela Sierra, Joshua Joseph Riki V.
Garcia, Patric Charles M.
Granda, Justin Tristan

In partial fulfillment of academic requirements
for the course
CS 191 Software Engineering I
of the
1st Semester, AY 2019-2020

Unique Reference:

The documents are stored in the https://github.com/jvdelasierra/sked-it/tree/master/02-Requirements%20Engineering/Project%20Deliverables referenced with “sked-it-2.2-Confirmation of Common Schedules.pdf" .

Document Purpose:

To be able to specify the requirements and other details of the Confirmation of Common Schedule , Use Case 2.2, of the sked-it application.

Target Audience:

The target audience of this document are programmers and designers of the sked-it. Also, anyone who is interested in the application and also acquainted to most, if not all, jargons in this document.

Revision Control:

Revision Date	*Person Responsible*	*Version Number*	*Modification*
09/22/19	Joshua V. Dela Sierra	1.0	Preparation of initial document
09/22/19	Joshua V. Dela Sierra	2.0	Added Swimlanes
09/23/19	Joshua V. Dela Sierra	3.0	Added ER Diagram

***Use-Case Name*:** Confirmation of Common Schedule

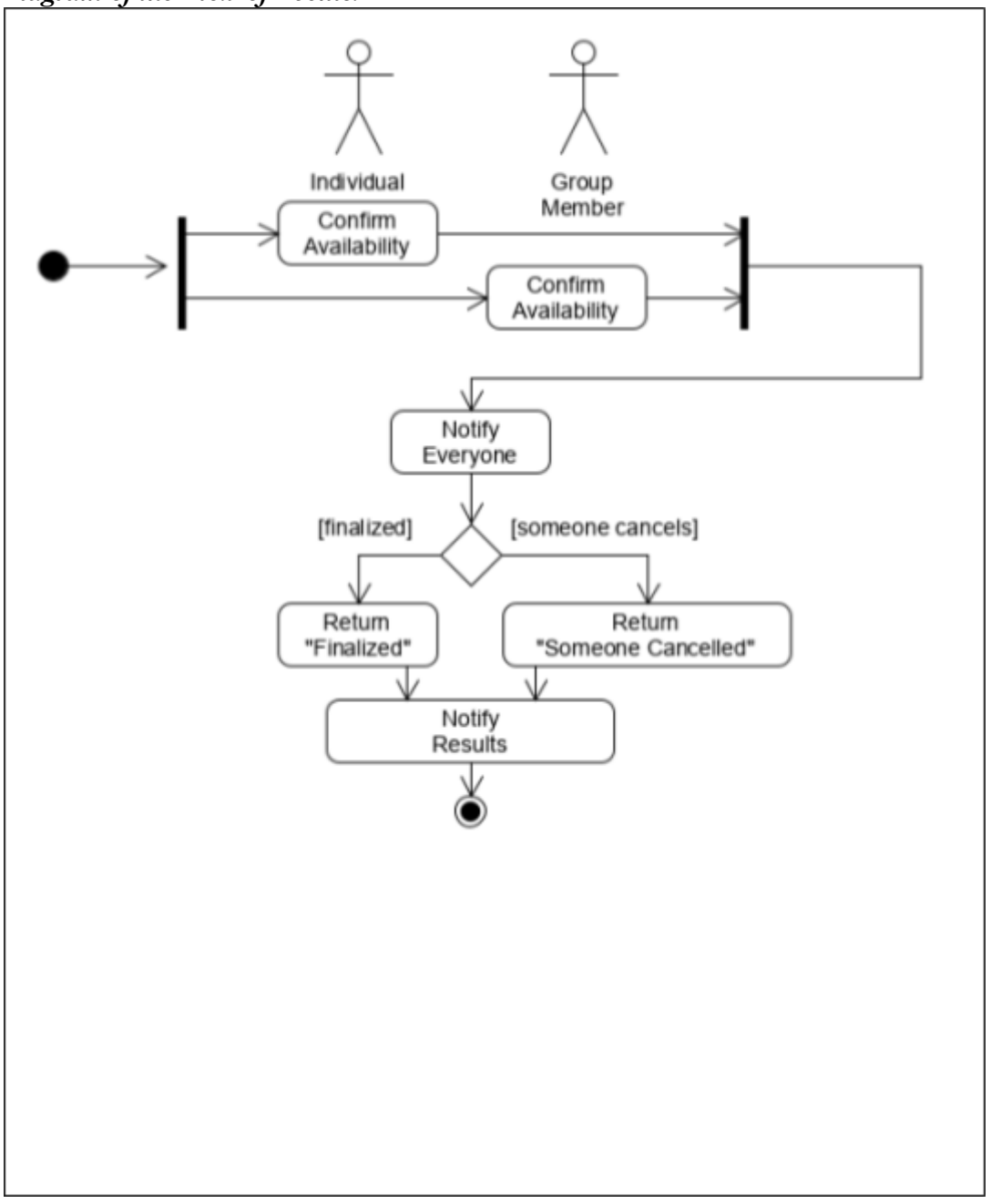
Description: The group can confirm their common schedule for a certain activity/event. This requires the Link Schedule functionality of the system. An automatic notification (notify group) will be queued shortly.

Preconditions: All the schedules should be linked together first.

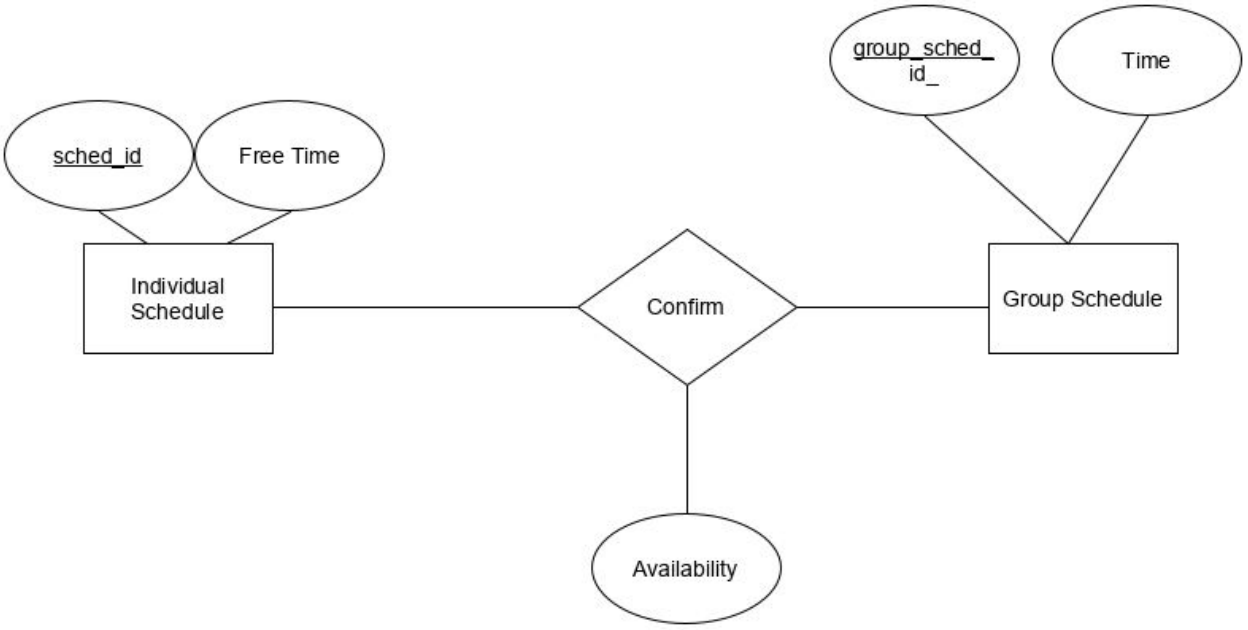
Flow of Events:

Scenario Name	*Description*
Scenario 1 (Basic Flow) Group Members have finally decided on the scheduled and everyone's schedule is free in that time interval.	1. Each member double-checks the intersection of schedules. 2. Each member confirms through the system. 3. Everyone has confirmed their schedule. 4. A notification is sent to every group member for the finalization of the schedule. 5. Everyone finalizes the schedule. 6. The schedules are confirmed.
Scenario 2 Some of the Group Members is free on that time interval while the others are not.	1. Each member double-checks the intersection of schedules. 2. Each member available will confirm availability will confirm "Available" in the system. Otherwise, the member will confirm "Unavailable" in the system. 3. Everyone has confirmed their schedule. 4. A notification is sent to every group member for the finalization of the schedule. 5. Everyone finalizes the schedule. 6. The schedules are confirmed.
Scenario 3 Some of the group members have confirmed yet needs to change their availability status.	1. Each member double-checks the intersection of schedules. 2. Each member confirms through the system. 3. Everyone has confirmed their schedule. 4. A notification is sent to every group member for the finalization of the schedule. 5. A member changes his/her availability. 6. A notification is sent to every group member for the finalization of the schedule. 7. Everyone finalizes the schedule. 8. The schedules are confirmed.

Activity Diagram of the Flow of Events:



Other Diagram:



Postcondition: NONE

Relationships: NONE

Special Requirements:
NONE