RETENTION SCHEDULER
Project Description

Submitted to:

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

Submitted by:
Joaquin, Neil Jonathan A.
Relao, David Paul A.
Simpao, Aldrin

In partial fulfillment of Academic Requirements
for the course
CS 191 Software Engineering I
of the
1st Semester, AY 2014-2015

Unique Reference:

You can find our documents stored at https://github.com/bluenavy/requirements.

Document Purpose:

This document serves as a blueprint for the project. It aims to be a guide or a model for the development process of the actual program and its prototype.

Target Audience:

The target audience of this project consists mainly of the clients, however since the project is useful not only for academics purposes, – it can be used for office work, or simply for daily use – we'd like to focus on the project as a useful tool for students.
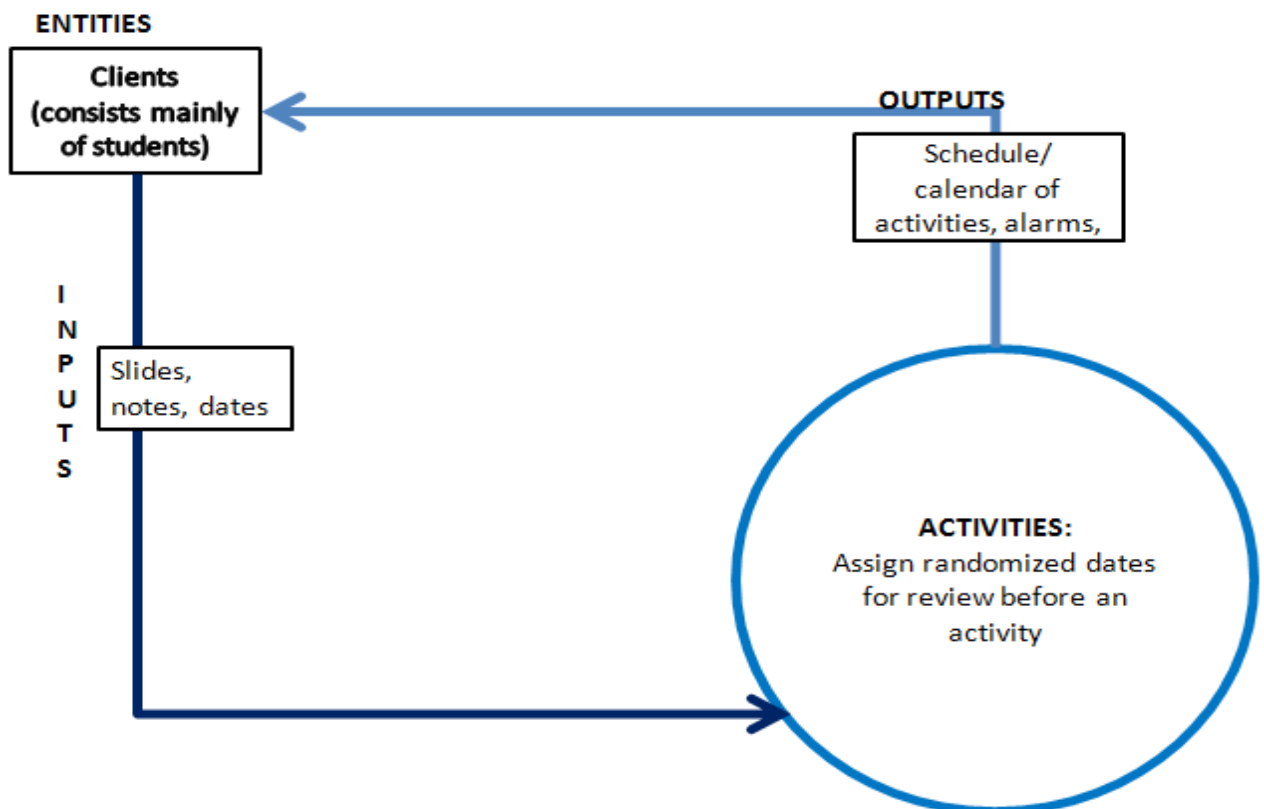
Revision Control:

Revision Date	Person Responsible	Version Number	Modification
8/26/14	Yno, Neil, David	1.0	Initial Document;
8/27/14	Yno	2.0	Added context diagram, and the gist of the document purpose, and a few minor edits.
8/27/14	Yno	3.0	Edit: Document purpose, Target Audience, Entities, Inputs, Outputs, and Functionalities.
9/9/2014	David	4.0	Edit: Description, Inputs, Outputs, and Functionalities

Project Title: Retention Scheduler

Description: Every time that the student learns something from class, he can jot down notes directly to this app by noting them in a separate feature like notepad. The dates and time for this are noted as well. Along with this is a student calendar synced with a school year. This can provide the student a planner in which he can note his exam dates and other stuff like quizzes, seat-works, and recitations, with certain topics/lessons that the student needs to focus on for a specific pre-scheduled activity. The student will be reminded of the upcoming activity through pop-up alarms for around 3 times before the activity day on pre-defined hour of each day. More to that, the alarms contain randomized lesson parts (cut by sub-topics like for a certain day of the lesson; moreover, this only serves as a little-by-little lesson feeding) or need-to-remember stuff (noted by students) that would be gotten from the notepad for student's retention as a preparation for the activity. Furthermore, the students can pile up slides and educational stuff on an archive, for example, for better compilation of his stuff to study.

Context Diagram:

Entities:
1. User – the target audience, an entity who uses/interacts with the program

Major Inputs:
1. Slides – external file inputs from the user
2. Notes – directly noted by students in a separate feature, notepad
3. Important dates (eg. Examination days, Quiz days, etc…)

Major Outputs:
1. Create a calendar of activities for the user.
2. Create randomized but relevant pop-up alarms.

Major Functionalities:
1. Manage the user's inputs and dates to remember.
2. Generate pop-up alarms on randomized dates.