

Sked-it

Software Architectural Design

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 Gabriel R.

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



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

Unique Reference:

The documents are stored in the <https://github.com/jvdelasierra/sked-it/tree/master/03-Design%20Engineering/> referenced with Sked-it - Architectural Design.pdf.

Purpose:

The purpose of this document is to review and potentially revise the sked-it Analysis Model in order to further develop the software architecture of the application.

Audience:

The target audience of this document would be the developers of sked-it and anyone who is interested in our process of building an application.

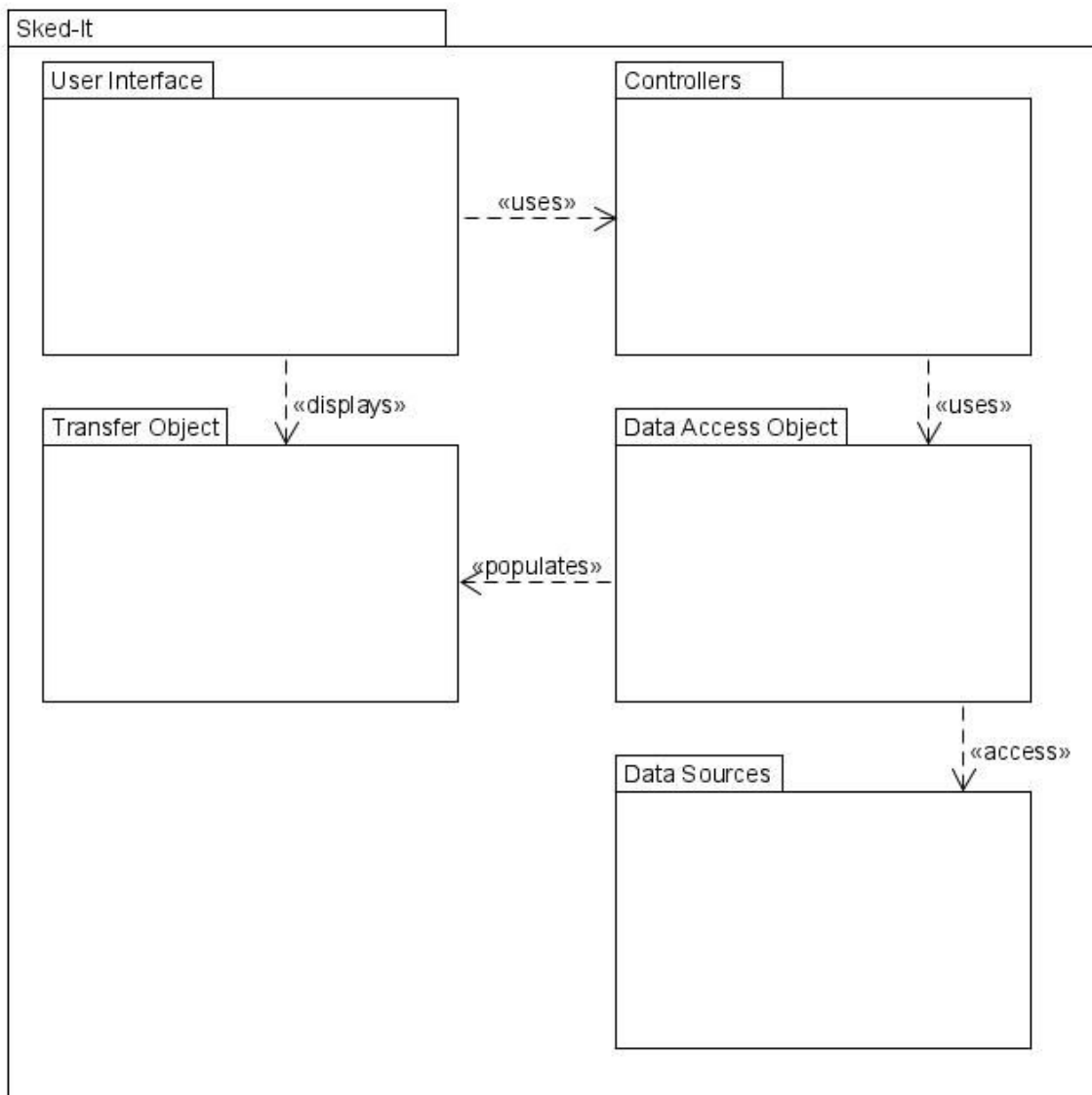
Revision Control:

<i>Revision Date</i>	<i>Person Responsible</i>	<i>Version Number</i>	<i>Contribution/Modification</i>
10/28/2019	Joshua V. Dela Sierra	1.0	Initial Document
10/30/2019	Justin Tristan Gabriel R. Granda	2.0	Added User Interface and Controller Packages
10/30/2019	Justin Tristan Gabriel R. Granda	2.5	Added DAO, Transfer Objects and Data Source Packages
10/31/2019	Joshua V. Dela Sierra	3.0	Added Software Architecture Model Diagram
11/01/2019	Patric Charles M. Garcia	4.0	Added System Name and Description. Edited the Description of the Data Sources Package Table

System Name: sked-it

Description: Sked-it is an app that we are proposing to make to help people manage their own personal schedules while being able to set-up meetings with other people. The app aims to provide users with a way to make a “routine” schedule which they could update if they have other things to do. One of the main features that we will implement is a Group Scheduling system where people in the same group will be allowed to view a portion of another person’s schedule (controlled by the owner of the schedule) within the same group so that the people involved in the group can easily set-up their meetings based on their common free time. A person will then be notified of their schedule with a reminder sent to their phones.

Revised Software Architecture Model:



User Interface Package:

Screen Name	Description
MonthCalendarUI	<p>This is the interface of the end user to the system whenever he/she needs to manage his/her monthly calendar.</p> <p>Responsibilities:</p> <pre>public void addEventCalendar(boolean addOrDelete, String date, String time, String description, String groupPrivacy) public void updateEventCalendar(String eventID, String date, String time, String description, String groupPrivacy) public void deleteEventCalendar(String eventID, boolean addOrDelete)</pre>
WeeklyTemplateCalendarUI	<p>This is the interface of the end user to the system whenever he/she needs to manage his/her weekly template calendar.</p> <p>Responsibilities:</p> <pre>public void addEventCalendar(boolean addOrDelete, String date, String time, String description, String groupPrivacy) public void updateEventCalendar(String eventID, String date, String time, String description, String groupPrivacy) public void deleteEventCalendar(String eventID, boolean addOrDelete)</pre>
WeeklyCalendarUI	<p>This is the interface of the end user to the system whenever he/she needs to manage his/her weekly calendar.</p> <p>Responsibilities:</p> <pre>public void addEventCalendar(boolean addOrDelete, String date, String time, String description, String groupPrivacy) public void updateEventCalendar(String eventID, String date, String time, String description, String groupPrivacy) public void deleteEventCalendar(String eventID, boolean addOrDelete)</pre>
DailyCalendarUI	<p>This is the interface of the end user to the system whenever he/she needs to manage his/her daily calendar.</p> <p>Responsibilities:</p> <pre>public void addEventCalendar(boolean addOrDelete, String date, String time, String description, String groupPrivacy) public void updateEventCalendar(String eventID, String date, String time, String description, String groupPrivacy) public void deleteEventCalendar(String eventID, boolean addOrDelete)</pre>
LoginUI	<p>This is the interface of the end user to the system whenever he/she needs to enter the system.</p> <p>Responsibility:</p> <pre>public String enterLoginDetails(String email, String password)</pre>
CreateProfileUI	<p>This is the interface of the end user to the system whenever he/she needs to create an account for the system.</p> <p>Responsibility:</p> <pre>public String enterProfileDetails(String email, String password, boolean agreeOnTerms)</pre>

ForgetPasswordUI	<p>This is the interface of the end user to the system whenever he/she forgot the password for the system.</p> <p>Responsibility:</p> <pre>public String enterDetails(String email)</pre>
LinkScheduleUI	<p>This is the interface of the end user to the system whenever he/she needs to link a schedule to a group.</p> <p>Responsibility:</p> <pre>public String enterDetails(String groupCalendarID, string eventID)</pre>
ConfirmScheduleIO.	<p>This is the interface of the end user to the system whenever he/she needs to link a schedule to a group.</p> <p>Responsibility:</p> <pre>public String confirmDetails(String groupCalendarID, string eventID) public String cancelDetails(String groupCalendarID, string eventID)</pre>

Controllers Package:

Controller Name	Description
ManageCalendarController	<p>This controller handles creating, updating and deleting calendars for the user.</p> <p>Responsibilities: public void updateCalendar(String eventID, String date, String time, String description, String groupPrivacy)</p>
AccountController	<p>This controller is responsible for maintaining existing user accounts, including logging in and managing forgotten passwords.</p> <p>Responsibilities: public String loginSystem(String email, String password) public void RenewPassword(String email)</p>
LinkScheduleController	<p>This controller handles schedule linking (i.e. finding common free time between schedules).</p> <p>Responsibilities: public Schedule linkSchedules(String eventID_1, String eventID_2) public boolean isCommonTimeFound(String eventID)</p>
ConfirmationController	<p>This controller handles cases when users need to confirm their schedule with the group.</p> <p>Responsibilities: public String setConfirmedSched(Schedule a) public void cancelConfirmedSched(Schedule a)</p>

Data Access Objects Packages:

DAO Name	Description
DBIndividual	This data access object is responsible for getting Individual data from a database.
DBGroup	This data access object is responsible for getting Group data from a database.
DBSchedule	This data access object is responsible for getting Schedule data from a database.

Transfer Objects Package:

Class Name	Description
Individual	This persistent class is responsible for holding a single athlete record
Schedule	This persistent class is responsible for holding a schedule record
Group	This persistent class is responsible for holding a single group record
PCLIndividual	This persistent class list is responsible for holding a list of Individual records.
PCLSchedule	This persistent class list is responsible for holding a list of Schedule records.
PCLGroup	This persistent class list is responsible for holding a list of Group records.

Data Sources Package:

Filename or Database Name	Descriptions
Individual	It contains the details of all the individual users in the sked-it app found in a relational database system.
Group	It contains the details of all the groups in the sked-it app found in a relational database system.
Schedule	It contains the schedules of all the users in the sked-it app found in a relational database system.