

# DCS Codex Notification System

## 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:

Galano, Anica Clarice Antonella P.  
Isidro, Rogiella Beatrice N.  
Tabagan, Ken Aaron F.

In partial fulfillment of Academic Requirements  
for the course  
CS 191 Software Engineering I  
of the  
1<sup>st</sup> 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 following link:

<https://github.com/theswinghobo/DCSCodexApp/tree/master/03-Design%20Engineering>

With the following filename:

**DCS Codex Notification System - Architectural Design.pdf**

### ***Purpose:***

The purpose of this document is to summarize the architecture of our system.

### ***Audience:***

The target audience is our Professor in CS 191, Ma'am Solamo and for our reference, in creation of our project.

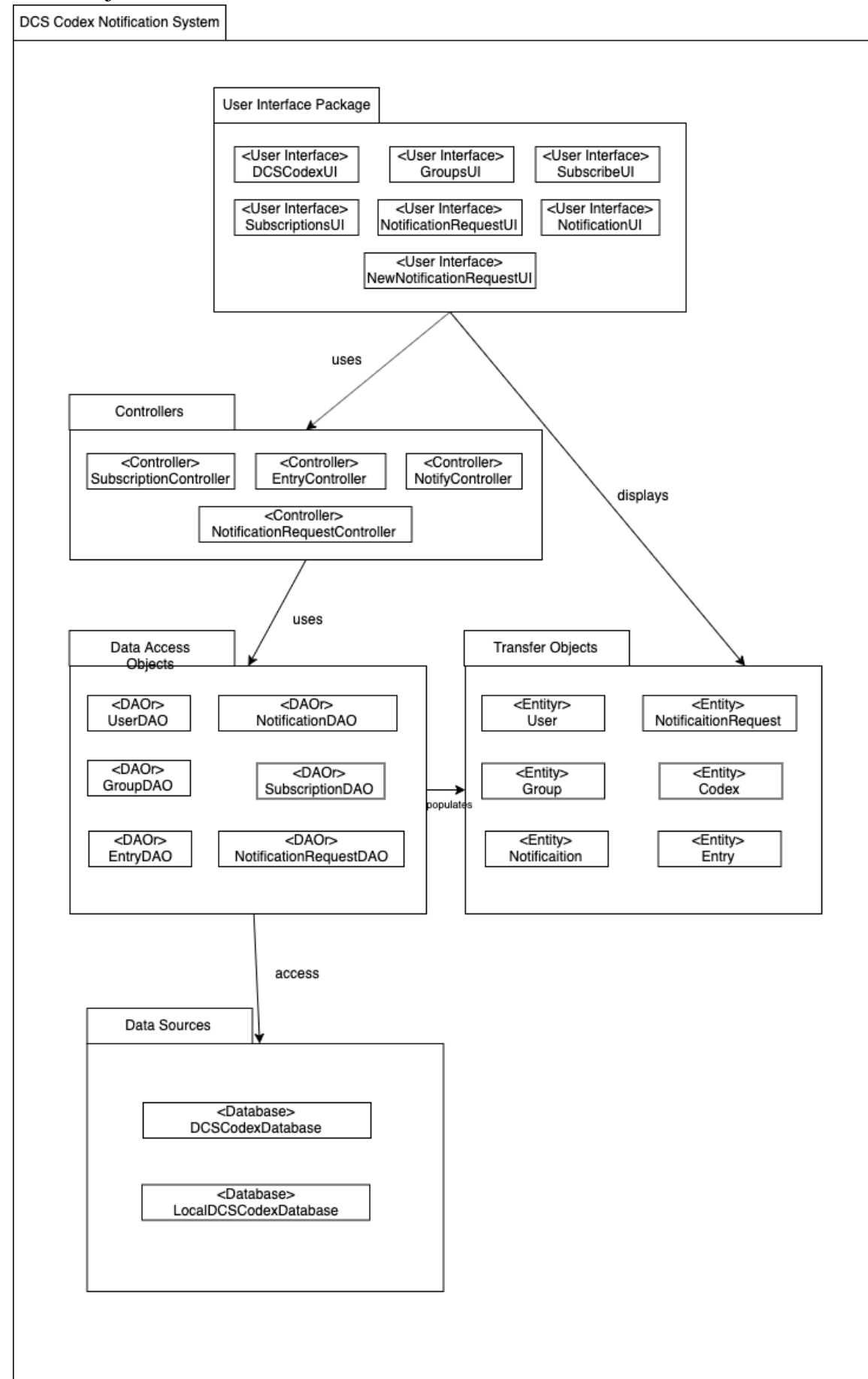
### ***Revision Control:***

<b><i>Revision Date</i></b>	<b><i>Person Responsible</i></b>	<b><i>Version Number</i></b>	<b><i>Contribution/Modification</i></b>
10/30/19	Anica Galano	1.0	Initial Document, system description, initial all User Interface Packages
10/31/19	Ken Tabagan	1.1	Initial Data Sources Package and Transfer Objects Package
11/1/19	Anica Galano	2.0	Initial Controllers Package, initial DAO packages, edited Data Sources Package edited Transfer Objects Package, added diagram

***System Name:*** DCS Codex Notification System

***Description:*** The DCS Codex is a calendar containing entries of events, deadlines and other important reminders related to the university, courses and CS organizations. Extending from that, the system implements notifications that serves as reminders for different entries via a mobile application. The system also introduces the idea of “groups”, which pertain to a certain class or organization a user can subscribe to in order to receive notifications in relation to the group. Special notification can also be requested by users to be shared with other people subscribed to a particular group.

## Revised Software Architecture Model:



### *User Interface Package:*

Screen Name	Description
DCSCodexUI	<p>This is the screen the <b>users</b> interact with to view the DCS Codex. It contains the Codex Entries. It also serves as the home screen.</p> <p><u>Attributes:</u></p> <pre>private Button nextMonthB; private Button prevMonthB; private Label monthLB; private Label dayLB; //Additional Labels for each day (Mon - Sun) private Label dateLB; // Additional Labels for each date (1-28/30/31) private Label entryLB; //Additional Labels for each entry  private Label notificationsLB; private Label notifHeaderLB; //Additional Labels for each notif private Label notifMessageLB; //Additional Labels for each notif</pre> <p><u>Responsibilities:</u></p> <pre>public Codex viewCodex() public Entry viewEntry(ActionEvent onClick, Date dateToday) public ArrayList&lt;Notification&gt; viewNotifications(int userID)</pre>
GroupsUI	<p>This is the screen the <b>admin</b> uses to manage the different groups.</p> <p><u>Responsibilities:</u></p> <pre>public void addGroup(String groupName); public void removeGroup(String groupName);</pre>
SubscribeUI	<p>This is the screen the <b>users</b> interact with to subscribe to different groups.</p> <p><u>Attributes:</u></p> <pre>private Label groupNameLB; //Additional Labels for each group private Checkbox groupNameCB; //Additional Checkboxes for each group private Button saveSubsBTN; private Button cancelSubsBTN;</pre> <p><u>Responsibilities:</u></p> <pre>public ArrayList&lt;String&gt; addToSubscribed(String group); public void subscribeToGroups(ArrayList&lt;String&gt; groups); public void cancelAction();</pre>
SubscriptionsUI	<p>This is the screen where <b>users</b> can view and interact with all of their current subscriptions.</p> <p><u>Attributes:</u></p> <pre>private Label mySubscriptionsLB;</pre>

	<p>private Label groupLB; //Additional Labels for each subscription</p> <p>private Button bellBTN; //Additional Buttons for each subscription</p> <p>private Button unsubscribeBTN //Additional Buttons for each subscription</p> <p>private Button addSubscriptionBTN;</p> <p>private Button removeAllBTN;</p> <p><u>Responsibilities</u></p> <p>private ArrayList&lt;String&gt; removeSubscription(String group);</p> <p>private void removeAllSubscriptions();</p> <p>private void addSubscription();</p>
NotificationRequestsUI	<p>This is the screen the <b>users</b> interact with to view the notification requests they have sent and their statuses.</p> <p><u>Attributes</u>:</p> <p>private Label notificationRequestsLB;</p> <p>private Label targetRecipientLB; //Additional Labels for each notification request</p> <p>private Label messageSummaryLB; //Additional Labels for each notification request</p> <p>private Tag statusTag; //Additional Tags for each notification request</p> <p>private Button addNewRequestBTN;</p> <p>private Button clearRequestsBTN;</p> <p><u>Responsibilities</u></p> <p>public void addRequest();</p> <p>public void clearRequests();</p>
NewNotificationRequestUI	<p>This is the screen the <b>users</b> interact with to make notification requests and send them to the admin.</p> <p><u>Attributes</u>:</p> <p>private Label NewRequestLB;</p> <p>private Label targetGroupLB;</p> <p>private Label messageLB;</p> <p>private Label purposeLB;</p> <p>private Label targetDateTimeLB;</p> <p>private DropDownList targetGroupDDL;</p> <p>private List&lt;String&gt; targetGroupList;</p> <p>private TextField messageTF;</p> <p>private TextField purposeTF;</p> <p>private DateTimePicker targetDateTimePicker;</p> <p>private Button submitBTN;</p> <p>private Button cancelBTN;</p>

	<u>Responsibilities:</u>  private submitRequest();
NotificationUI	This is the screen <b>users</b> interacts with when viewing a popup notification.  <u>Attributes:</u>  private Label headerLB; private Label messageLB; private Button dismissBTN; private Button remindBTN; private Image logoIMG;  <u>Responsibilities:</u>  public void dismissNotif(); public void remindLater(Int timeAfter);

### ***Controllers Package:***

Controller Name	Description
SubscriptionController	<p>This is the control that maintains all subscriptions, specifically all user-group pairs.</p> <p><u>Attributes:</u></p> <pre>private int userID; private int groupID;</pre> <p><u>Responsibilities:</u></p> <pre>public addSubscription(int userID, int groupID); public removeSubscription(int userID, int groupID);</pre>
NotificationRequestController	<p>This is the control that handles notification requests from users.</p> <p><u>Responsibilities:</u></p> <pre>public void sendNotifRequest(NotificationRequest n) public void notifResponse(NotificationRequest n, String status, String message)</pre>
NotifyController	<p>This is the control that handles notifications sent to users from the admin.</p> <p><u>Responsibilities:</u></p> <pre>public void sendNotification(Notification n, User user) public void resendNotification(Notification n, Date datetime, User user)</pre>
EntryController	<p>This is the control that handles entries in the DCS Codex as maintained by the admin.</p> <p><u>Responsibilities:</u></p> <pre>public void addEntry(Entry e)</pre>



### *Data Access Objects Packages:*

DAO Name	Description
UserDAO	<p>This is the entity class User, which contains the data about the User.</p> <p><u>Attributes:</u></p> <p>private User u;</p> <p><u>Methods:</u></p> <p>private void insertUser(User user); private void updateUser(User user); private void deleteUser(User user);</p>
GroupDAO	<p>This is the entity class Group, which contains the data about a Group.</p> <p><u>Attributes:</u></p> <p>private Group group;</p> <p><u>Methods:</u></p> <p>private void insertGroup(Group group); private void updateGroup(Group group); private void deleteGroup(Group group);</p>
SubscriptionDAO	<p>This is the entity class Subscription, which contains the data about aSubscription.</p> <p><u>Attributes:</u></p> <p>private int userID; private int groupID;</p> <p><u>Methods:</u></p> <p>private void insertSubscription(int userID, int groupID); private void updateSubscription(int userID, int groupID); private void deleteSubscription(int userID, int groupID);</p>
NotificationRequestDAO	<p>This is the entity class NotificationRequest, which contains the data about a NotificationRequest.</p> <p><u>Attributes:</u></p> <p>private NotificationRequest n;</p> <p><u>Methods:</u></p> <p>private void insertNR(NotificationRequest n); private void updateNR(NotificationRequest n); private void deleteNR(NotificationRequest n);</p>
NotificationDAO	<p>This is the entity class Notification, which contains the data about a Notification.</p> <p><u>Attributes:</u></p> <p>private Notification n;</p> <p><u>Responsibilities:</u></p> <p>private void insertNotification(Notification n); private void deleteNotification(Notification n);</p>
EntryDAO	<p>This is the entity class Entry, which contains the data about an Entry.</p> <p><u>Attributes:</u></p>

	<pre>private Entry e;;</pre> <p><u>Responsibilities:</u></p> <pre>private void insertEntry(Entry e); private void updateEntry(Entry e); private void deleteEntry(Entry e);</pre>
--	--

### ***Transfer Objects Package:***

Class Name	Description
User	<p>This is the entity class User, which contains the data about the user.</p> <p><u>Attributes:</u></p> <pre>private int userID; private String role; private String studentNumber; private String username; private String email;</pre> <p><u>Methods:</u></p> <pre>public int getID(); public String getRole(); public String getStudentNumber(); public String getUsername(); public String getEmail();  public void setID( int ID); public void setUsername(String username); public void setEmail(String email); public void setStudentNo(String studentNumber);</pre>
Group	<p>This is the entity class Group, which contains data about subjects or organizations users can subscribe to (e.g. CS 191, DCS Servers).</p> <p><u>Attributes:</u></p> <pre>private int ID private String name</pre> <p><u>Methods:</u></p> <pre>public int getGroupID() public String getGroupName() public ArrayList&lt;Integers&gt; getSubscribers()</pre>
Notification	<p>This is the entity class Notification, which contains data about notifications that get sent out by the admin to the users.</p> <p><u>Attributes:</u></p> <pre>private int ID private int groupID private Date timestamp; private String header; private String message;</pre> <p><u>Methods:</u></p> <pre>public int getID() public int getGroupID() public void setMessage(String message) public void sendNotification(int groupID)</pre>
NotificationRequest	<p>This is the entity NotificationRequest, which contains data about notification requests sent by users to admin for approval.</p> <p><u>Attributes:</u></p> <pre>private int requesterID; private int groupID;</pre>

	<pre> private String message; private String header; private String purpose; private Date dateToSend private String status  <u>Method:</u>  public int getRequesterID(); public int getGroupID(); public String getMessage(); public String getHeader(); public String getPurpose(); public Date getDateToSend(); public String getStatus();  public void setRequesterID(int ID); public void setGroupID(int ID); public void setMessage(String message); public void setHeader(String header); public void setPurpose(String purpose); public void setDateToSend(Date date); public void setStatus(String status); </pre>
Codex	<p>This is the entity Codex, which contains data about the Codex.</p> <p><u>Attributes:</u></p> <pre> private Date = currentDate ArrayList&lt;Entry&gt; entries = new ArrayList&lt;Entry&gt;() </pre> <p><u>Method:</u></p> <pre> public ArrayList&lt;Entry&gt; getEntries() public Entry getEntry(int entryID) </pre>
Entry	<p>This is the entity class Entry, which contains data about events and other entries that are displayed in the DCS Codex.</p> <p><u>Attributes:</u></p> <pre> private int ID private Date date private String name private String info private int groupID </pre> <p><u>Method:</u></p> <pre> public int getEntryID() public Date getDate() public String getName() public String getInfo() public int getGroupID()  public void setEntryID(int ID) public void setDate(Date date) public void setName(String name) public void setInfo(String info) public void setGroupID(int groupID) </pre>

***Data Sources Package:***

<b>File Name or Database Name</b>	<b>Description</b>
DCSCodexDatabase	This is the database that contains all elements of the DCS Codex System.
LocalDCSCodexDatabase	This is the database that contains all the user-specific elements of the DCS Codex App.