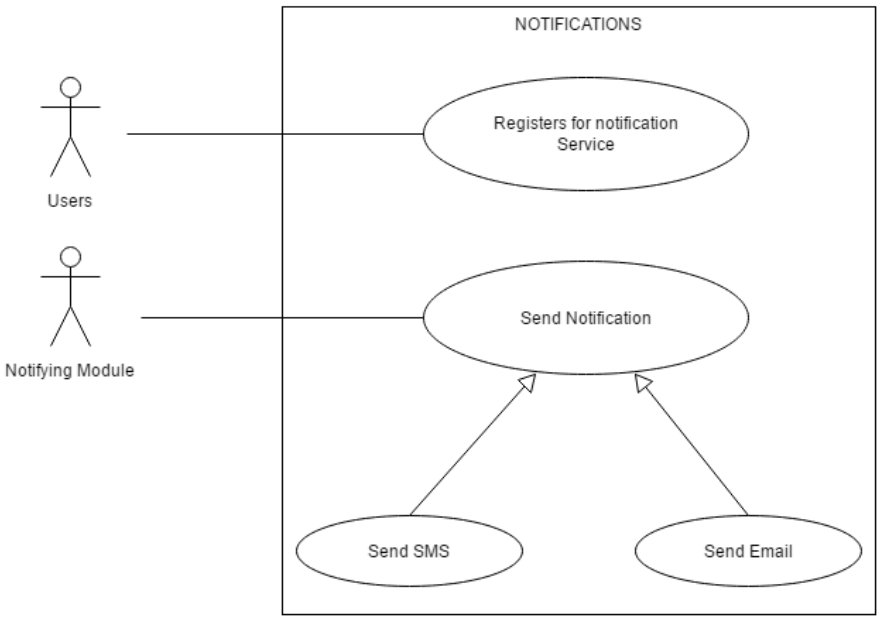
1 Subsystems

1.1 Notification Subsystem

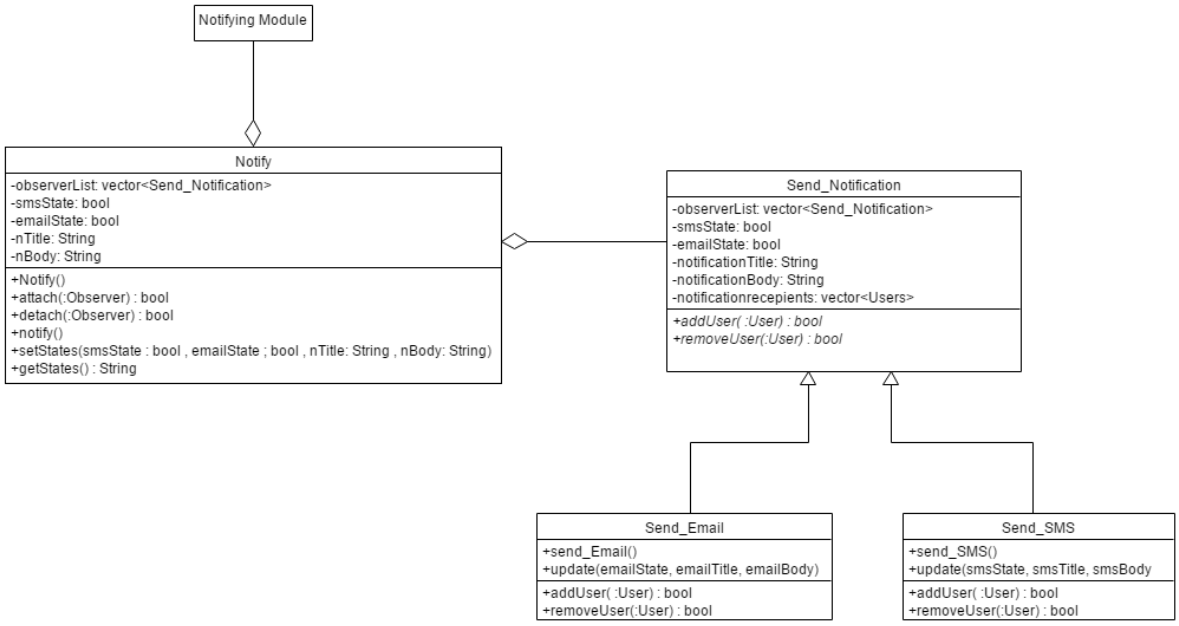
1.1.1 Overview

The notifications module provides notifications to system users regarding particular system updates that a user would like to be notified about through some medium external to the application.

1.1.2 UML Diagrams



Notification Use Case Diagram

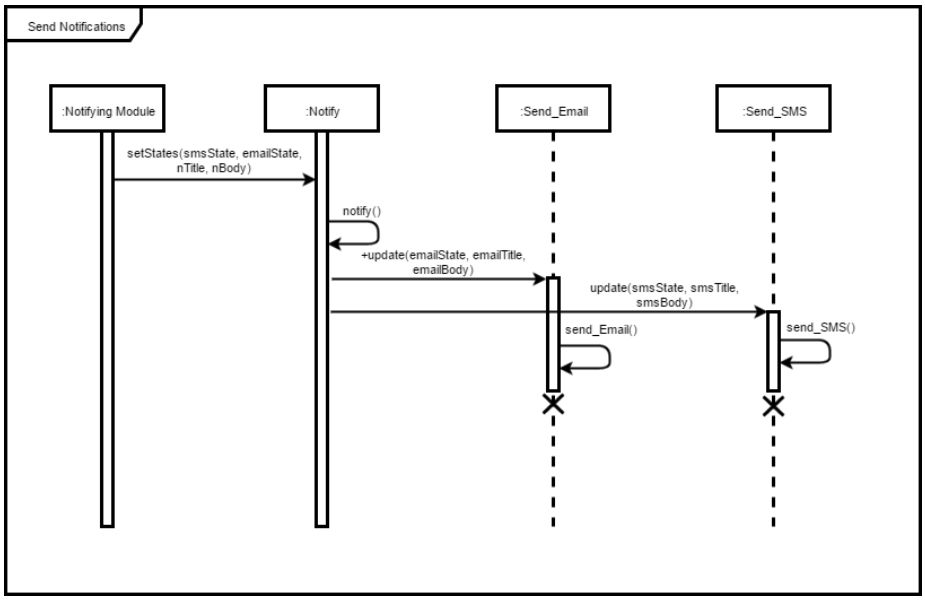


Notification Class Diagram

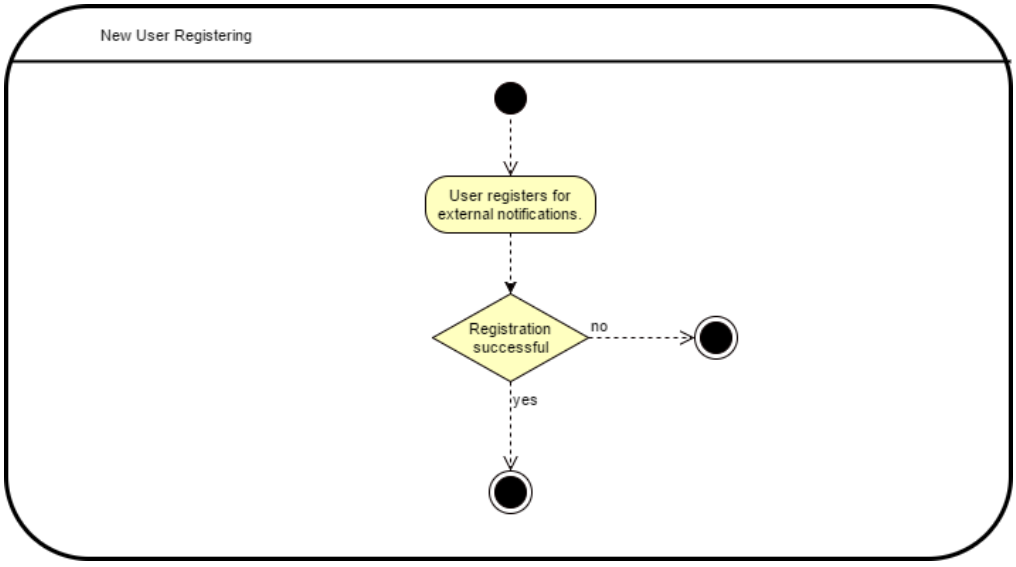
For the notifications subsystem the observer design pattern came to light, as this pattern allows objects to change state and all of its dependents follow suit. This allows a notification state to be used that can update whether or not a notification needs to be sent out. In this case the push method of observer was used. Also evident is the template design pattern that uses the concrete observers to redefine the way in which the notifications are sent. Notifying modules are also given the flexibility to decide which medium/s are used for the notification.

There are methods in the Send Email and Send SMS classes that allow users to be added to the user lists for the notifications to either or both of these mediums. There is also provision for more notification mediums to be added in the future and these can be attached to the Notify class. Should a notification medium be unavailable it can be detached from the Notify class until it is available. The Notify class can be called from any other module that would like to request a notification to be sent out to users which allows for modularity.

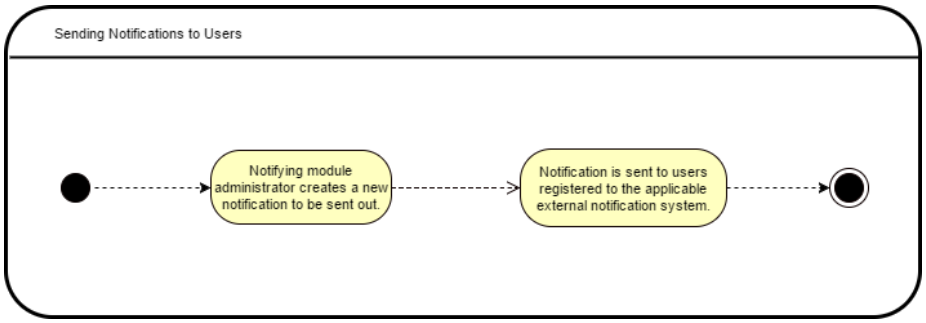
Notify: Subject
Send Notification: Observer(Observer) / Abstract Method(Template)
Send Email: Concrete Observer1(Observer) /Concrete Method(Template)
Send SMS: Concrete Observer2(Observer) / Concrete Method(Template)



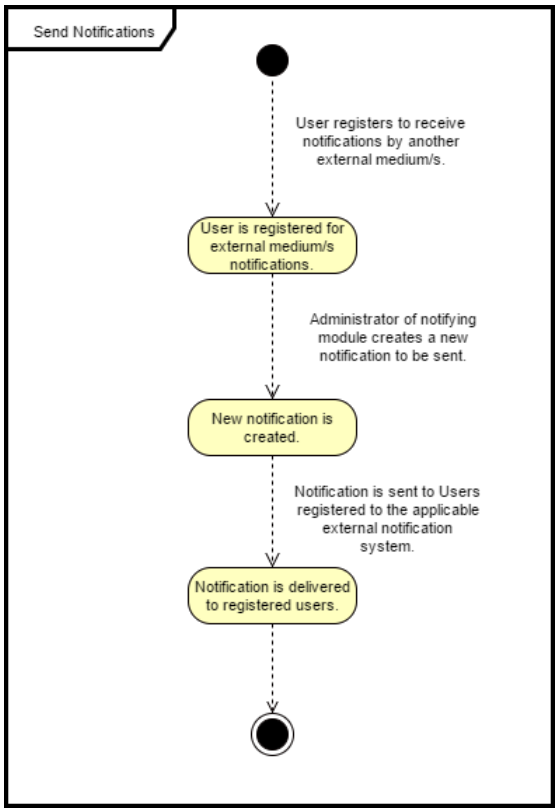
Notification Sequence Diagram



Add User Activity Diagram



Send Notification Activity Diagram



Notification State Diagram