1.1.1 Notifications Module

Scope

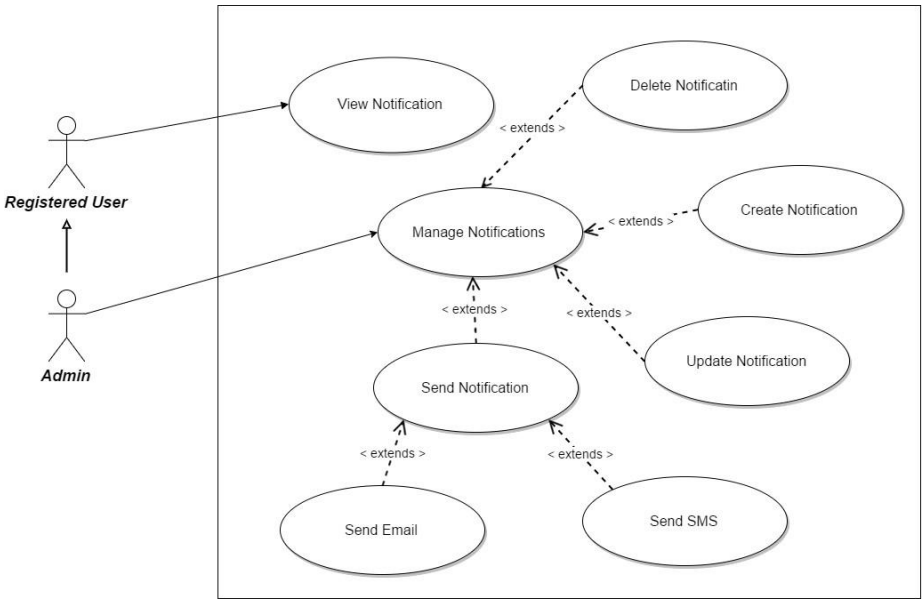


Figure 5: Notification Module Use-Case Diagram

The notification Module will provide end-users notifications with regards to the different aspects of the system, such as system updates or events taking place. Both subsystems as well as administrative users will use the notification system. The notification subsystem will send notifications either via email or SMS, depending on the end-user's noted preference.

Requirements, Interfaces and Constraints

The system is required to be light weight and must be easily interfaced with to allow any subsystem that needs the ability to send a user a notification the ability to do so. It thus needs to be extremely modular

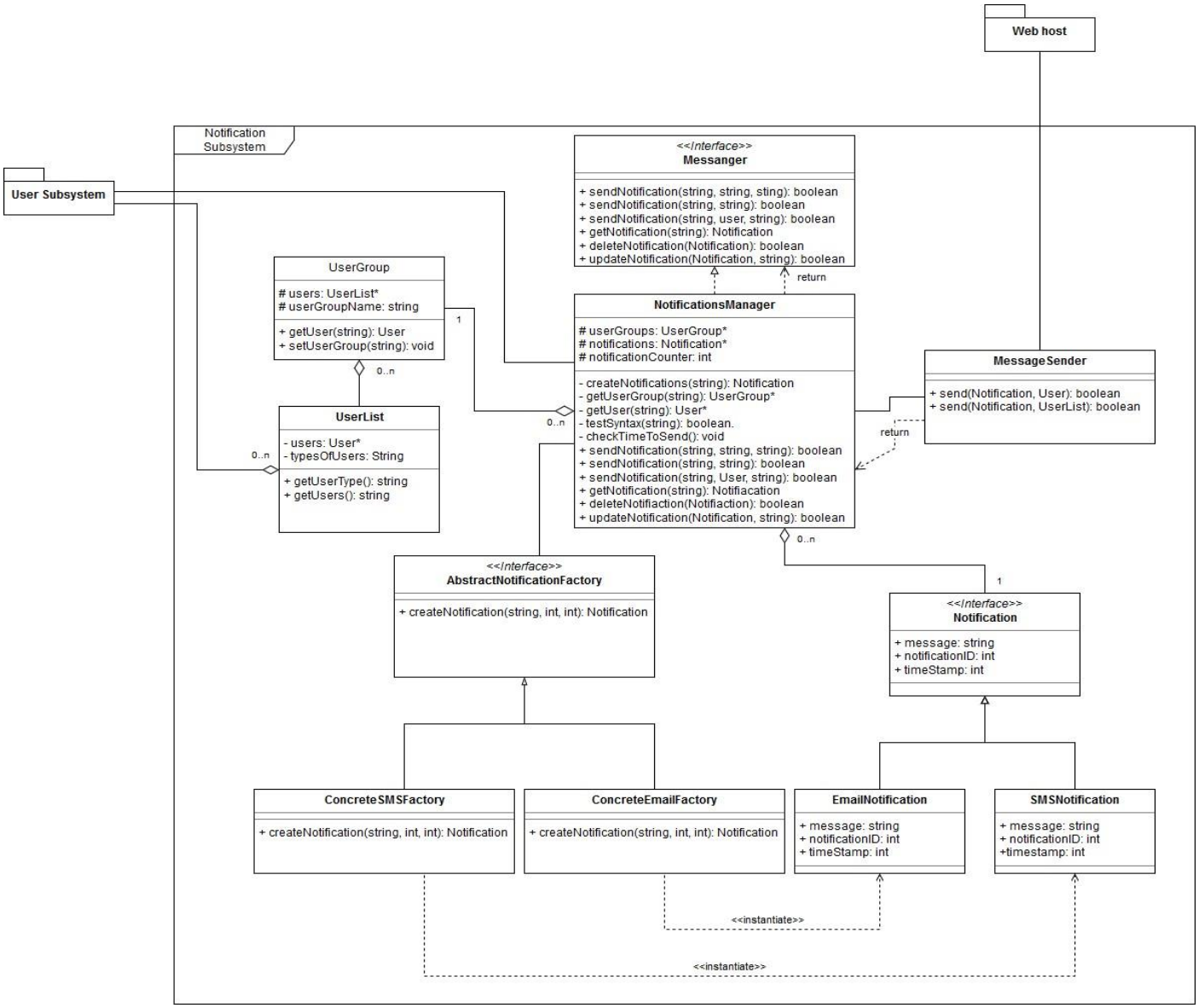
The Notification Module will need to interface with the user module, as it will need access to different users and their contact details as well as their preference methods of notification. The module will also need to provide a simple interface, as mentioned above, for simple messaging through the system. The Notification subsystem will also need to interface with an external browser or email client in order to send email to the UP email server so that those emails may in turn be

forwarded to the end-user and/or an email to SMS gateway that will SMS the end-user.

The Notification Module will need to be able to built message all users, certain groups of users and specific users. The Notification Module will also need to be able to send time scheduled messages and as such be able to update said notifications or update them. Thus notifications must be retrievable.

Since the Notification Module does not host its own in application messaging system, the subsystem is constrained to interfacing with web based methods of communication through the use of email and gateways. This means that their is no way for the notification subsystem to know if the end-user has indeed reviewed their notification and thus can not create or send reminder notifications. There is also lot less control over the presentation and user experience of the notifications as they will have to conform to a standardized means of test presentation. However this means the system is easier to implement as a lot of leg work has already been done with regards to sending emails from software packages and frees developers up from creating a in application based notification system from scratch.

Notification Module Structure



The Notification Module will provide a simple interface through which most communication and interaction with the subsystem will be done. This ensures a highly modular design. This interface allows for easy message creation by creating

notification with various overloaded functions. It also allows for notification retrieval, updating and deletion.

The NotificationsManager class implements the interface and provides the administrative functionality required to send notifications. It interfaces with the User Module to get users for their details by either getting users or lists of users, called user groups. Updates notifications and retrieves them from a stored list. It also ensures that notification are sent when they need to go out. Furthermore it tests the syntax structure of the message it is required to send before handing it off to the abstract factory class AbstractNotifcationFactory.

The AbstractNotificationFactory class is an interface that allows access to two concrete factories that implement a different concrete type of the Notification class. one being a email factory and the other being a SMS factory. Once these different kinds of concrete Notification types have been created they are handed back to the NotificationsManger, which stores them for when they need to be sent.

Using dependency injection, the NotificationsManager hands off the required notifications and users to the MessageSender class who interfaces with the required web technologies and handles the sending of notifications to each or groups of end-user.