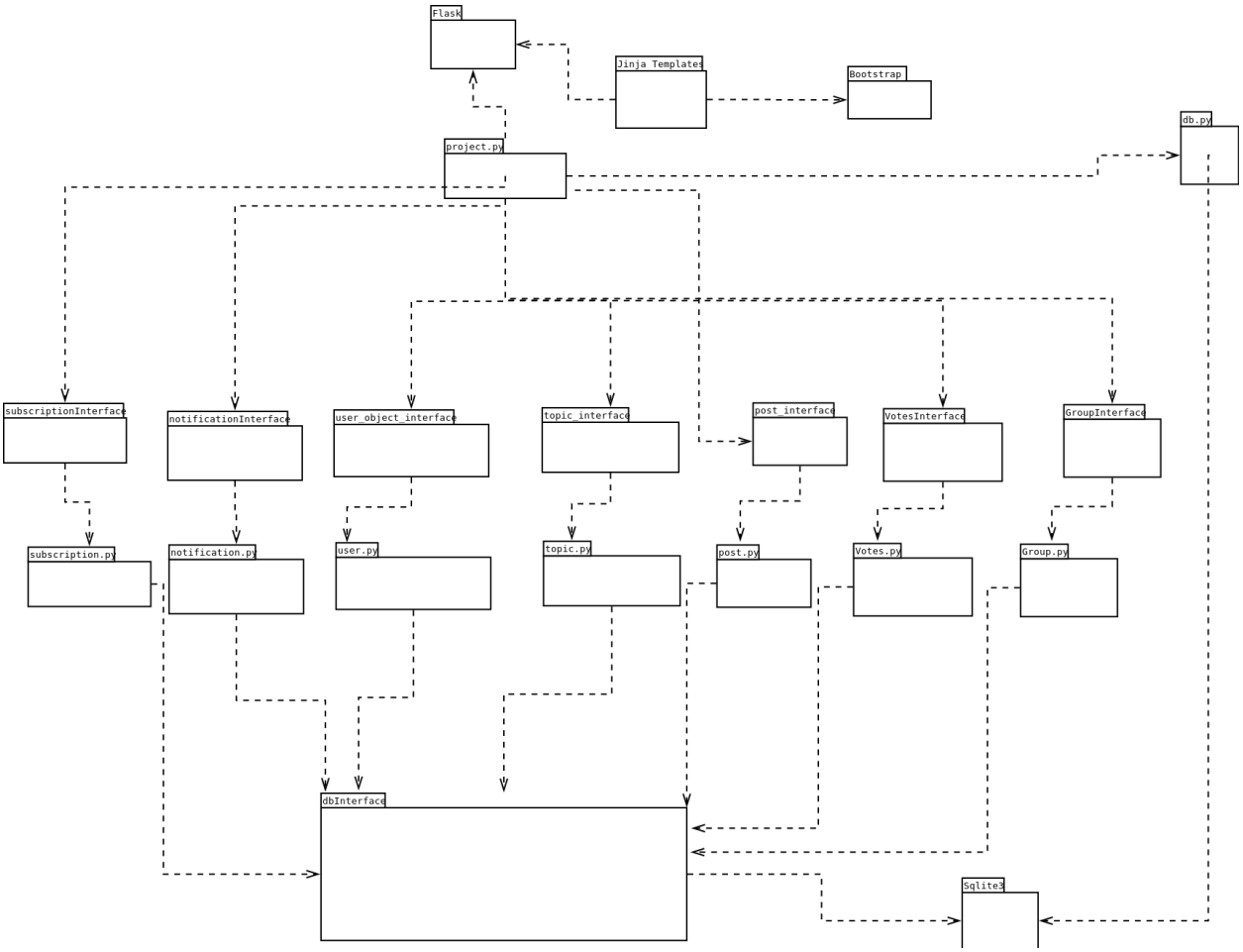
Design

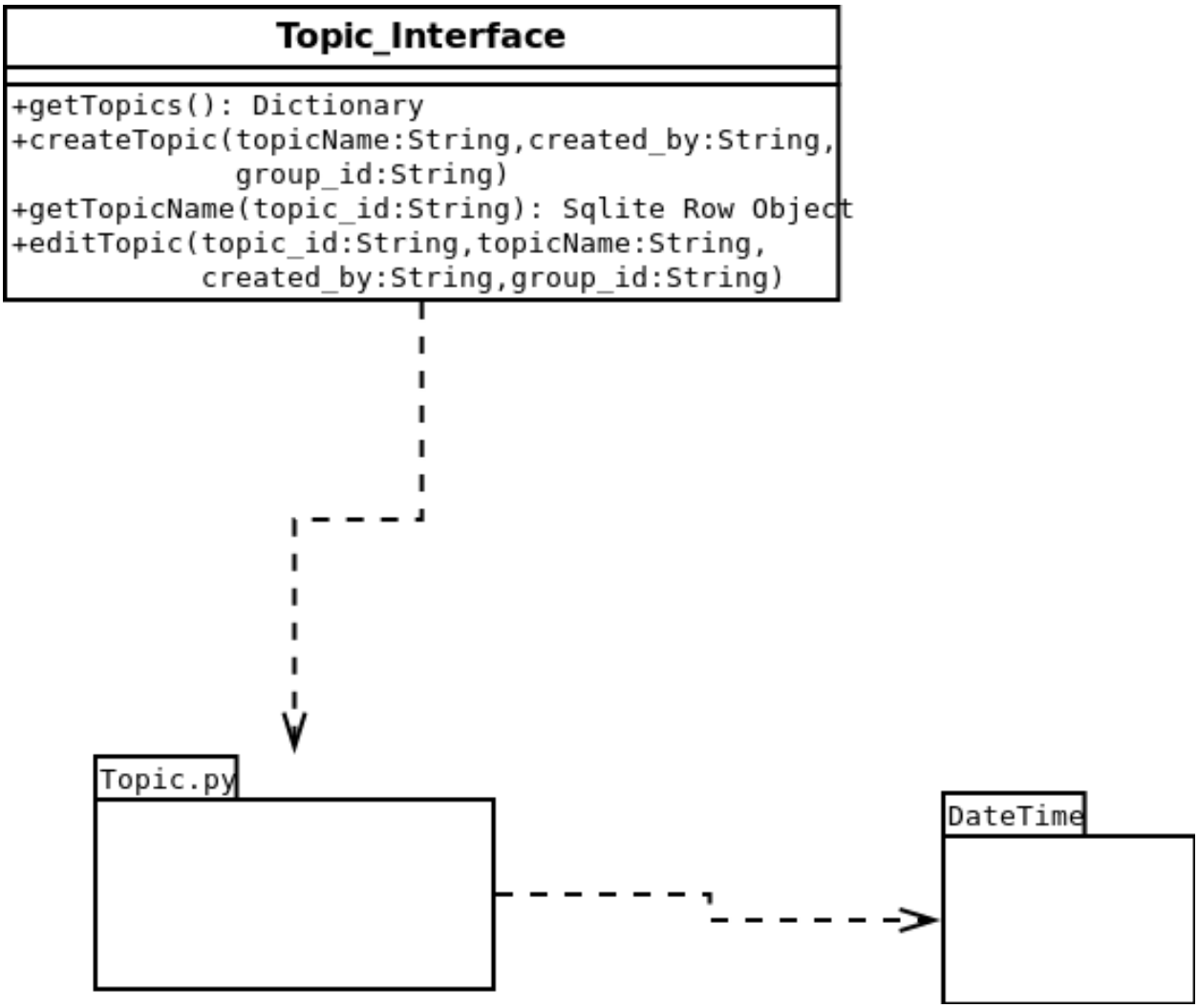
High Level Design - Current System



The above UML diagram describes the current state of the prototype. The design utilizes interfaces to provide a boundary for packages. These interfaces allow loose coupling within the project with respect to the domain packages. These domain packages are used by a main project module, which has a dependency on flask. The domain modules connect to a db Interface to provide persistence.

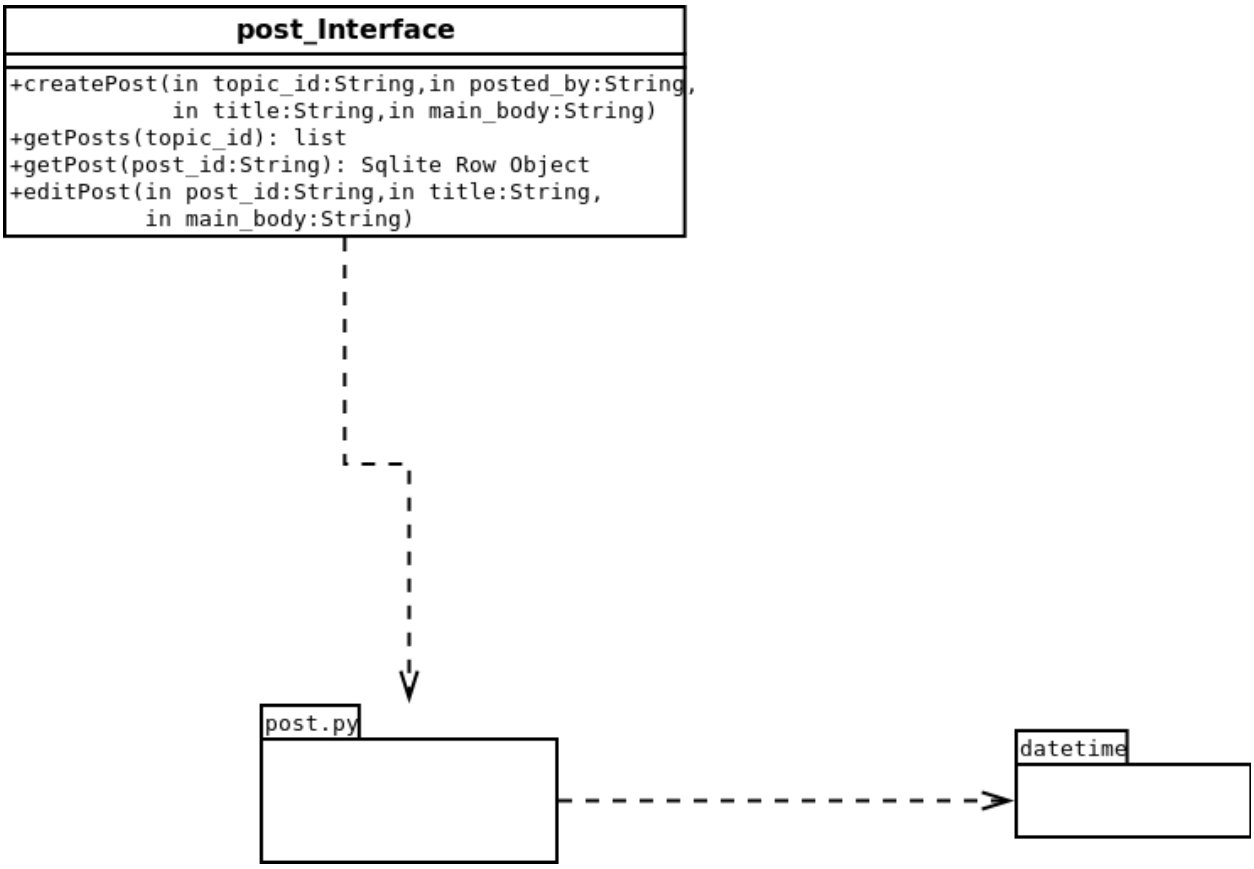
Low Level Design

Topic Interface



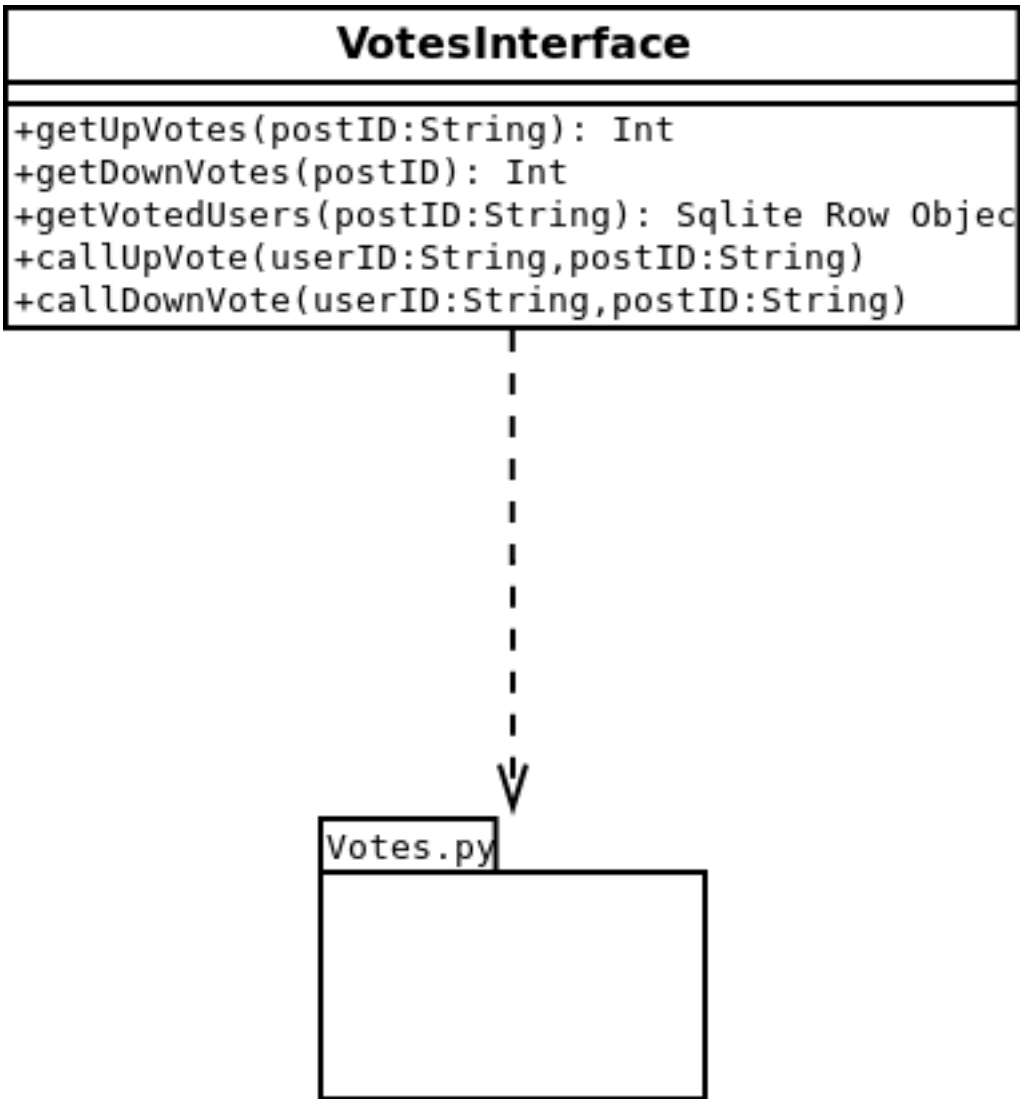
The above UML diagram describes the current state of the topic interface and its relationships within the package and domain module. The topic interface is a core part of the project is allows operates on the bounds of the related functional features surrounding creating and editing topics.

Post Interface



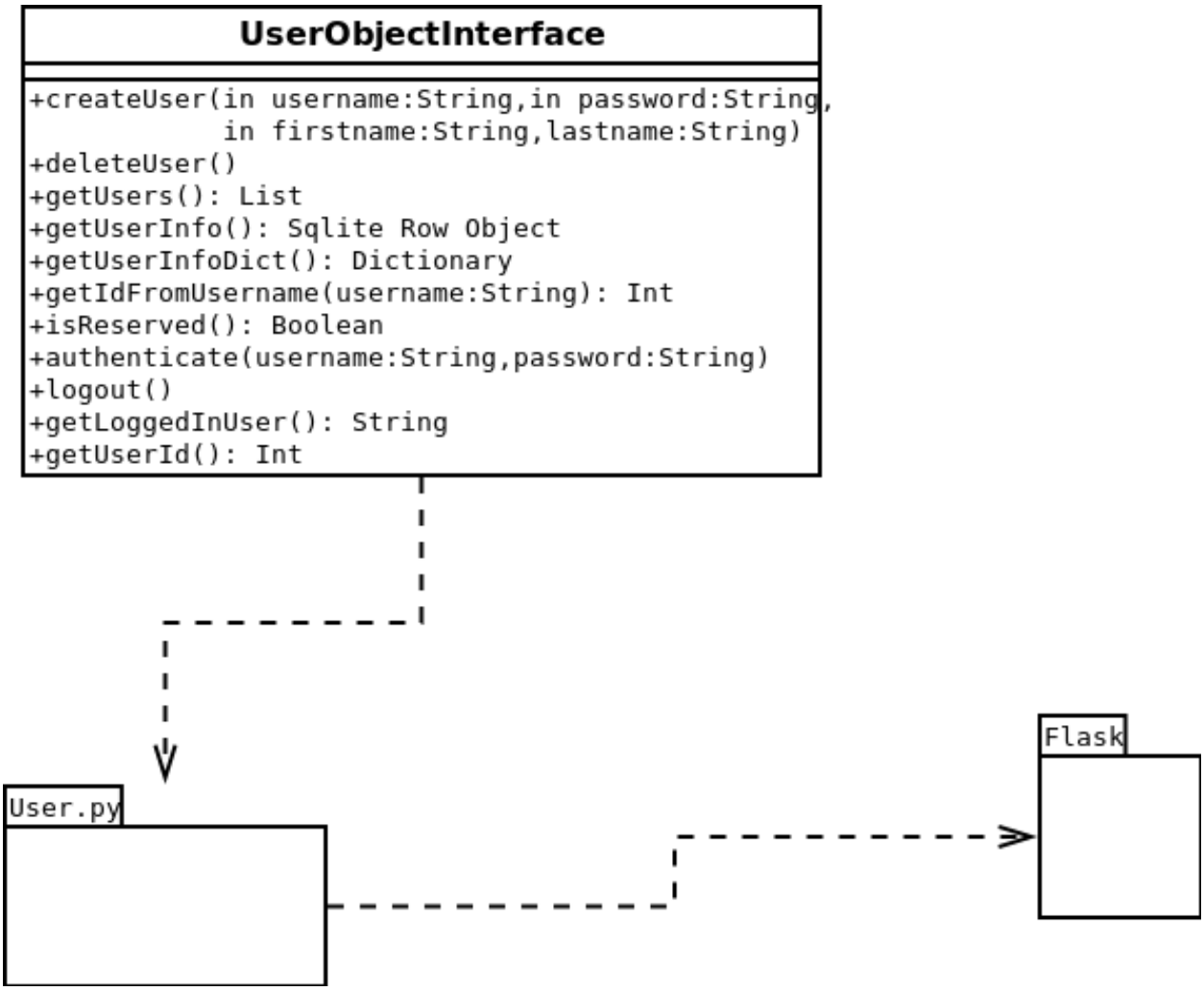
The above UML diagram describes the current state of the post interface and its relationships within the package and domain module. The post interface is a core part of the project is allows operates on the bounds of the related functional features surrounding creating and editing posts.

Votes Interface



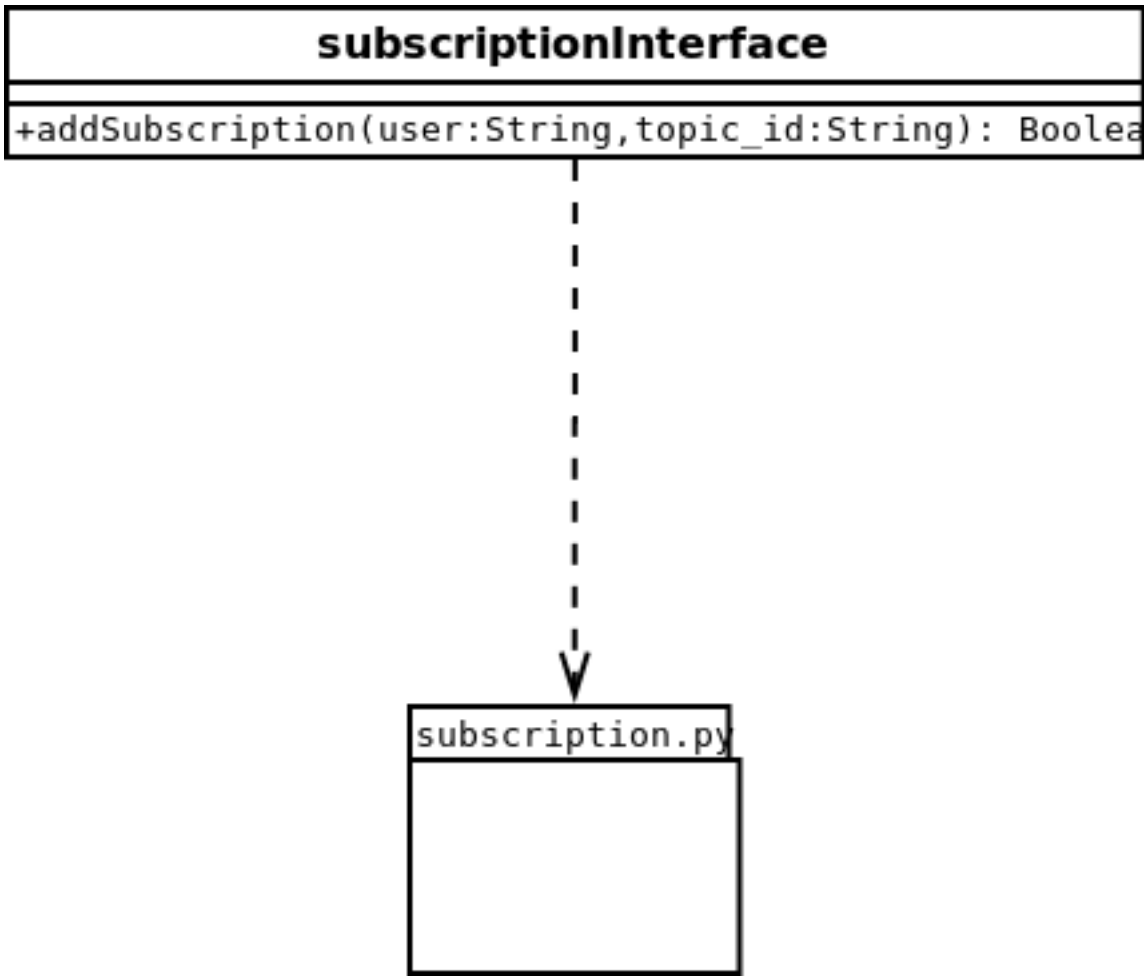
The votes interface has a dependency on a votes module, which carries out the required logic for implementing the required functional feature. Votes Interface is used only by the main project module.

User Object Interface



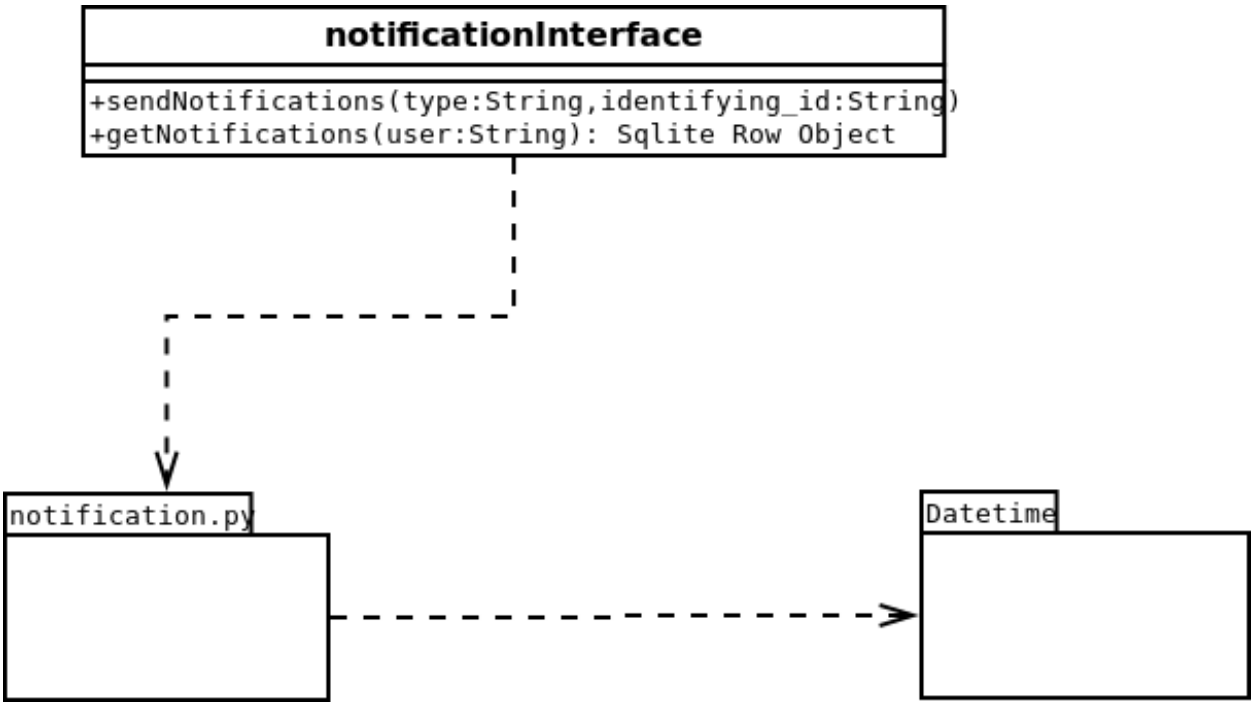
The User object Interface is used to provide access to the user module, which implements the login for creating accounts and the required authentication features. This interface is utilized by the main project module, which using the flask module calls the desired User Object Interface to interact with this portion of the project.

Subscription Interface



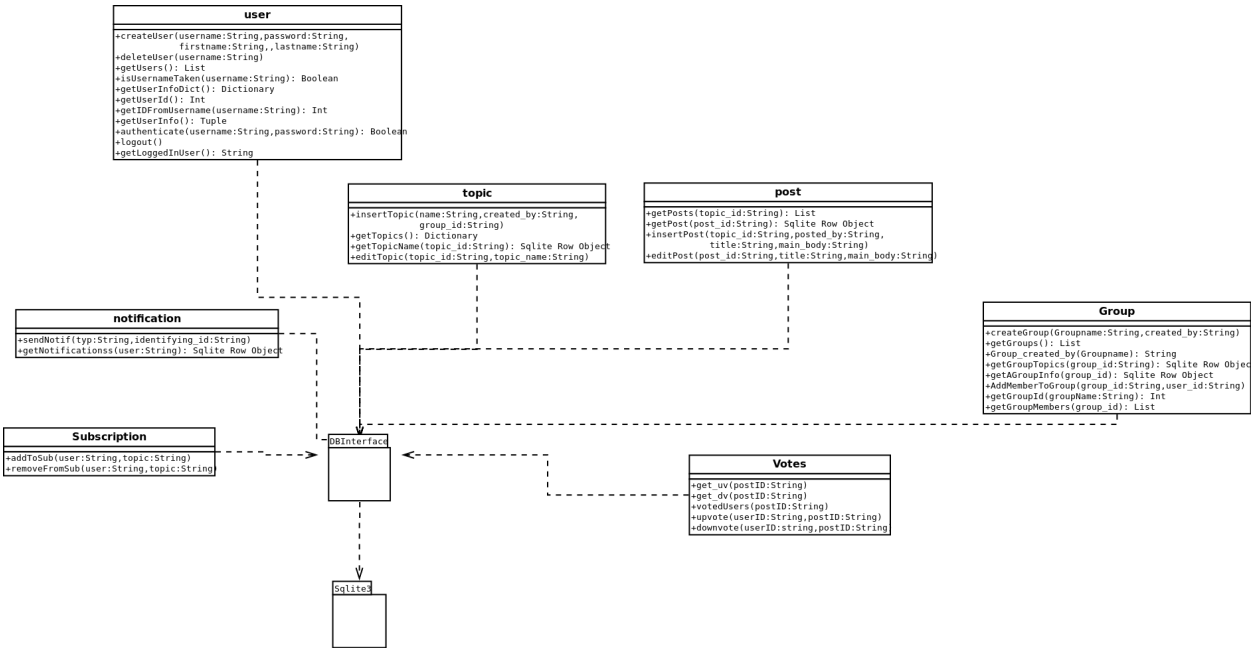
The subscription interface is an integral portion to the project as it provides the method for uses to subscribe to a topic. This is important as this is used later by the project module to determine the subscribed users to send the appropriate notification to. This particular interface in the current state of the prototype may require a observable or listener to operate freely with more loose coupling.

Notification Interface



The notification interface is used to provide the required functionality for sending notification to uses and also for retrieving notifications to display to the appropriate user. This module works by being interacted by with the main project module. This runs the appropriate logic to send the notification to the appropriate subscribed users.

DB Interface

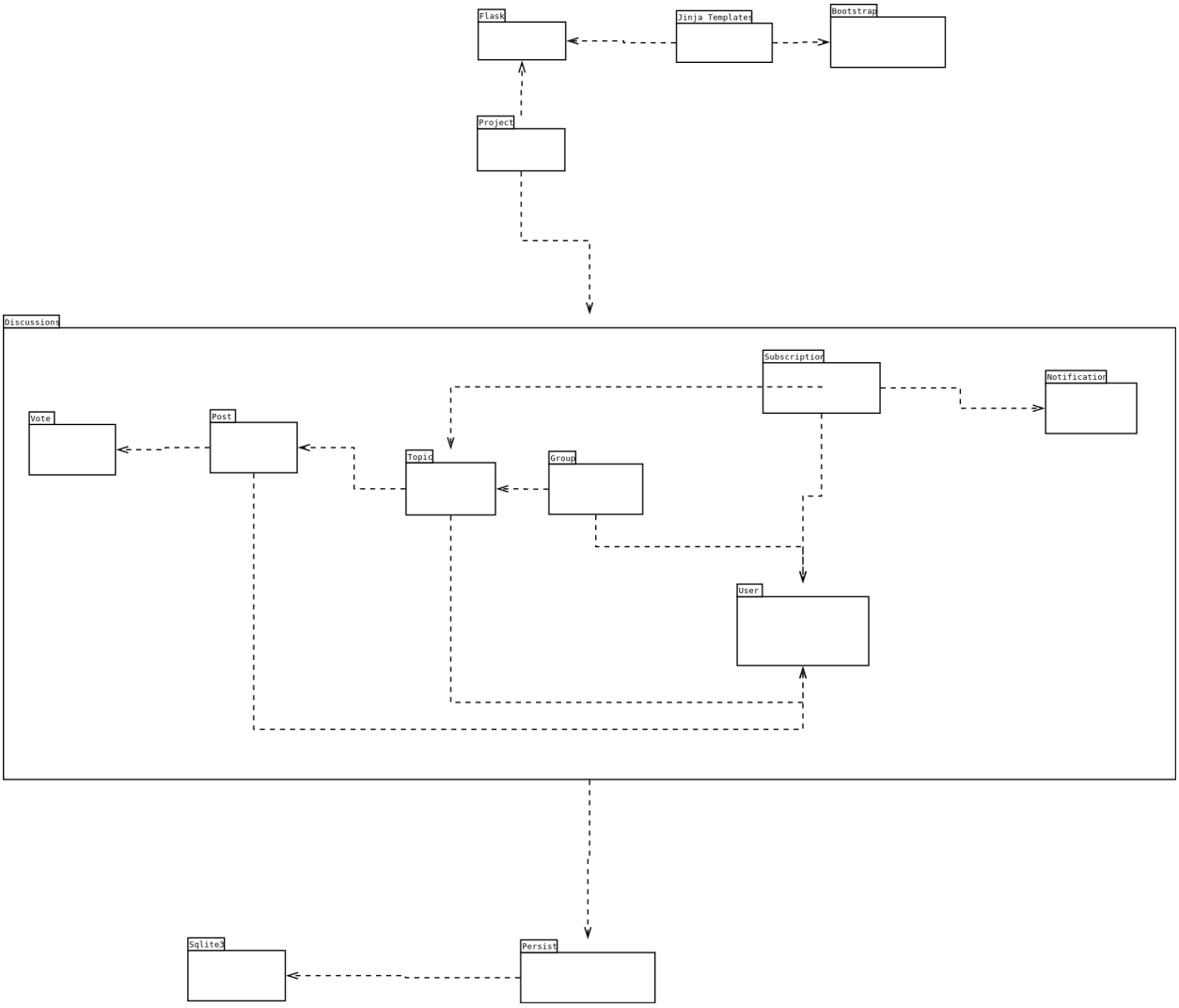


DB Interface module is used to provide persistence to the project. The important aspect is that it uses Sqlite database, which provides flexibility in having the data all stored in a .db file. This allows the database to be easily moved. The domain modules uses this persistence module throughout out the project with read and write calls that result in manipulation to the database.

Alternative Design

The team recognizes the the weaknesses and strengths of the current prototype. The current design including the high level design and low level design are not the best. The drawback with this is that is too database dependent. This can put strain on the new developers that will be continuing the project at a later date. Due to this, there are new proposed high level design and a low level design that describes the immediate domain.

Alternative High Level Design



Alternative Low Level Design