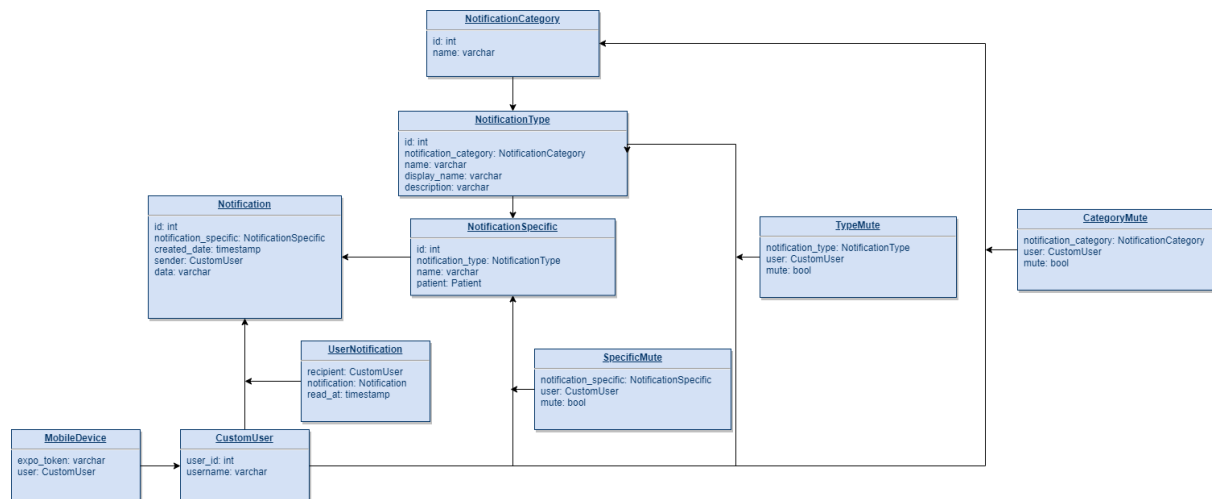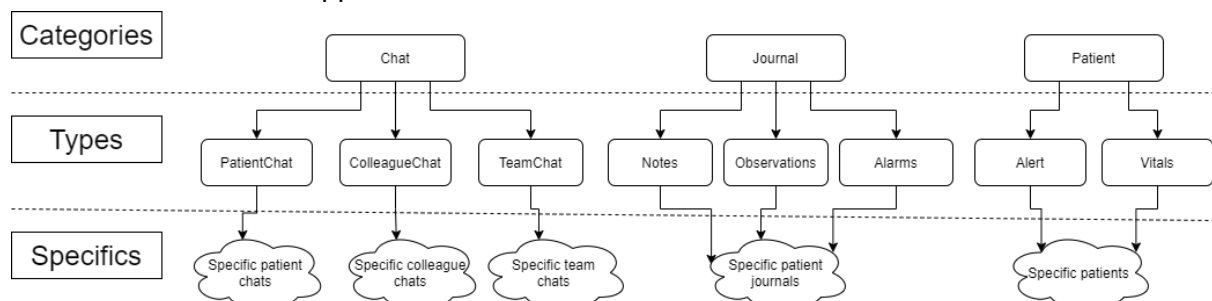# Notifications

Notifications is a django app developed by the app team. It's only used for its models which are used in the endpoints for the api. Notifications are a new system created for notifying users of changes or important news. Notifications are loosely based on Facebook's notification system. Notifications should not be confused with push notifications which are completely separate. Push notifications are sent to the device when either a notification is added or a message is received. Push notifications can also be muted, whilst notifications cannot.

This diagram shows the relationship between the models/tables for notifications. This diagram is also available in larger size. At the center there is the Notification model which has a NotificationSpecific and a many to many relationship to CustomUser as a notification can be directed to many users and users can have many notifications. The connection table UserNotifications also has a read_at field which is used to mark that the user has read the notification. Users can have zero to many mobile devices which contain expo tokens used to send push messages to the users devices. These mobile devices are automatically added when the user logs into the app from a new device, and removed when the user logs out. Notifications have a specific, a type and a category with a hierarchical structure. This is done in order to enable muting push notifications with varying degrees of scope. A user can for example mute push notifications for a specific colleague. In that case a row will be updated or appended in SpecificMute with mute equals true. These rows are lazy, meaning that they are not created until they are needed. If a user does not have a row the mute tables it means that the specific, type or category is not muted, and if a user mutes a chat they have never muted before, a row is appended.

The diagram shows the categories and types for notifications. Categories and types are added to the database in the server installation, whilst specifics are generated on the fly in a similarly lazy manner to the mute tables, meaning that they are created first when they are needed.