**ТЗ для Сервиса Личных Сообщений**

**Привет, Everyone!)**

Вкратце по сервису личных сообщений (сокращенно СЛС по русски или PMS at English - Private Message Service). Как я себе представляю его работу:

1) Каждый пользователь, зарегистрированный в Genelove, имеет доступ к сервису личных сообщений независимо от его роли/ролей.

2) Каждый пользователь может отправить сообщение любому другому пользователю. Вопрос многоадесной рассылки думаю обсудить попозже, для начала сделаем серсис типа “от одного к одному”, т.е. наш сервис на начальной стадии сможет отправлять сообщение от одного пользователя только к одному адресату.

3) Каждое сообщение доступно только двум пользователям - отправителю и получателю.

4) Каждое сообщение должно иметь статус ПРОЧИТАНО/НЕ ПРОЧИТАНО.

5) Каждое сообщение должно иметь дату создания и дату прочтения. Здесь дата также включает в себя время.

6) По мере обмена сообщениями между пользователями каждому пользователю будут доступны только те сообщения в которых этот пользователь участвует либо как отправитель либо как получатель сообщения.

7) Создание сообщения происходит путем его добавления в таблицу **Message**, при этом полю статуса присваивается значение НЕ ПРОЧИТАНО, а полю даты создания - текущее время на момент создания сообщения.

8) По прочтении сообщения получателем, статус сообщения меняется на ПРОЧИТАНО. При этом также в поле даты прочтения вставляется текущее время на момент прочтения сообщения.

Как это работает:

Первоначально у пользователя нет ни одного сообщения, т.е. в таблице нет ни одной записи с идентификатором этого пользователя в одном из полей **sender\_id** или **recipient\_id**. В этом случае этому пользователю ничего не отображается, нет сообщений - нет вывода.

Как только пользователь создает сообщение кому-либо (или наоборот, кто-либо создает сообщение для нашего пользователя) в таблице появляется запись с идентификатором нашего пользовается в одном их двух полей **sender\_id** или **recipient\_id**. Теперь мы показываем нашему пользователю что у него есть сообщение другому пользователю (если наш пользователь создал сообщение) или наоборот что у нашего пользователя есть одно входящее сообщение от другого пользователя. Назовем это **Диалог** (**Диалог** - наличие хотя бы одной записи в таблице **Message** с идентификатором нашего пользователя в любом из двух полей **sender\_id** или **recipient\_id**. Или другими словами - Диалог это пара идентификаторов **sender\_id** и **recipient\_id**.)

При отображении **Диалога** нашего пользователя с другим пользователем отображаем следующую информацию: количество входящих и исходящих сообщений, статус и время САМОГО последнего (по времени) сообщения для этого **Диалога** (т.е. для этой пары полей **sender\_id** и **recipient\_id)**.

При вхождении в **Диалог** нашему пользователю отображаем все сообщения из таблицы для которых пара полей **sender\_id** и **recipient\_id** соответствуют паре **Диалога**.

При создании нашим пользователем нового сообщения другому пользователю, отличному от предыдущего собеседника создается новый **Диалог** - новая пара полей **sender\_id** и **recipient\_id.** Алгоритм работы аналогичен вышеописанному.

Дополнение: сообщение и ответ на него в одном диалоге будут иметь зеркальные пары Отправитель-Получатель. Необходимо учитывать что это один Диалог а не два разных, и расположение идентификаторов пользователей-собеседников в полях Отправитель и Получатель не имеет никакого значения.

Исходя из этого таблица **Message** должна иметь следующий набор полей (типы полей пока для упрощения напишу на Java.)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Тип и название поля | Инициализация поля | Описание поля | Требование к полю |
| 1 | int id | Инициализируется при создании сообщения | Идентификатор сообщения. В БД также первичный ключ. | Должен быть уникальным |
| 2 | int sender\_id | Инициализируется при создании сообщения | Идентификатор отправителя (связка с таблицей Users) | Должен быть в таблице Users |
| 3 | int recipient\_id | Инициализируется при создании сообщения | Идентификатор получателя(связка с таблицей Users) | Должен быть в таблице Users |
| 4 | Date creationTime | Инициализируется при создании сообщения | Дата и время создания сообщения |  |
| 5 | Date readingTime | Инициализируется при прочтении сообщения | Дата и время прочтения сообщения |  |
| 6 | Enum status | Инициализируется при создании сообщения | Статус сообщения. Во время создания сообщения его статус - НЕ ПРОЧИТАНО. По прочтении сообщения получателем статус сообщения меняется на ПРОЧИТАНО |  |
| 7 | String message | Инициализируется при создании сообщения | Тело сообщения |  |

Вроде больше ничего для нашего СЛС и не нужно. Пока у меня больше нет мыслей на счет этого.

В будущем при наличии времени и желании можно будет реализовать **мультидиалог** - отправка сообщения двум и более пользователям. Также можно будет добавить в поле статуса третье значение - **УДАЛЕНО**. В этом случае, сообщение с таким статусом больше не будет отображаться ни отправителю ни получателю. Также, надо ли добалвять дополнительный функционал в СЛС для **администраторов**, если надо то какой?. Но это сейчас не принципиально, можно потом обсудить.