**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ**

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФГБОУ ВО**

**«КУРГАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

Кафедра «Программное обеспечение автоматизированных систем»

**Автоматизированная система**

**«Интернет форум»**

**текст программы**

**РФ КГУ 090303.65.КП.101301107 ТП**

Листов 3

2016

## Сохраненная функция добавления пользователя в базу (регистрация).

1. db.system.js.save({
2. \_id: "addUser",
3. value: **function**(username, password, email){
4. **if** (db.users.insert(
5. {
6. username: username,
7. pass\_hash: hex\_md5(username+password),
8. email: email
9. }
10. )) { **return** **true**; } **else** { **return** **false**; }
11. }
12. })

Выводом работы данной процедуры будет true/false, что будет значит, добавлен пользователь или нет.

## Сохраненная функция добавления темы в базу.

1. db.system.js.save({
2. \_id: "addTheme",
3. value: **function**(username, subsection\_id, title, message){
4. **var** user = db.users.findOne({username: username});
5. **var** subsection = db.subsections.findOne({\_id: subsection\_id});
7. **if**(subsecection.encrypt) {
8. message = encrypt\_aes256(message, `${username}+${user.password\_hash}`);
9. }
11. **if**(db.themes.insert({
12. subsection\_id: subsection.\_id,
13. user\_id: user.\_id,
14. title: title,
15. message: message, *// may be encrypted*
16. })) { **return** **true**; } **else** { **return** **false**; }
17. }
18. })

Учитывается зашифрованность подраздела, если это так – то шифруется с помощью aes256 ключом (username+password\_hash). Результат стандартный – добавлена тема или нет.

## Сохраненная функция добавления сообщения в тему с учетом пользователя.

1. db.system.js.save({
2. \_id: "addMessage",
3. value: **function**(username, theme\_id, message){
4. **var** user = db.users.findOne({username: username});
5. **var** theme = db.themes.findOne({\_id: theme\_id});
6. **var** subsection = db.subsections.findOne({\_id: theme.subsection\_id});
8. **if**(subsection.encrypted) {
9. message = encrypt\_aes256(message, "${username}+${user.password\_hash");
10. }
12. **if**(db.messages.insert({
13. theme\_id: theme.\_id,
14. user\_id: user.\_id,
15. message: message,
16. })) { **return** **true**; } **else** { **return** **false**; }
17. }
18. })

Добираемся до пользователя, темы и подраздела с помощью исходных данных. Проверяем шифрование раздела, если оно есть – шифруем сообщение, как и тему. Записываем сообщение с привязкой к теме и пользователю.

## Сохраненная функция полной выборки темы со всеми её сообщениями.

1. db.system.js.save({
2. \_id: "getTheme",
3. value: **function**(theme\_id){
4. **var** theme = db.themes.findOne({\_id: theme\_id});
5. **var** subsection = db.subsections.findOne({\_id: theme.subsection\_id});
7. **var** messages = db.messages.find({theme\_id: theme.\_id})
9. **if**(subsection.encrypted) {
10. **var** theme\_user = db.users.findOne({\_id: theme.user\_id});
11. theme.message = decrypt\_aes256(message, "${theme\_user.username}+${theme\_user.password\_hash}");
13. **for**(**var** i=0; i<messages.count(); i++) {
14. **var** message\_user = db.users.findOne({\_id: message.user\_id});
15. messages.message = decrypt\_aes256(message, "${message\_user.username}+${message\_user.password\_hash}");
16. }
17. }
19. **return** {theme, messages};
20. })

Все сообщения, включая главное в теме, расшифровываются, если раздел помечен зашифрованным. Выходные параметры – поля темы, и массив сообщений, которые связаны с данной темой. В последствии, в приложении, есть возможность получить данные и о подразделе (с помощью сущности темы), и о пользователях написавших как тему, так и сообщения в ней (с помощью связи с пользователями).