Assignment 4- Documentation

Kyle Russell 2016

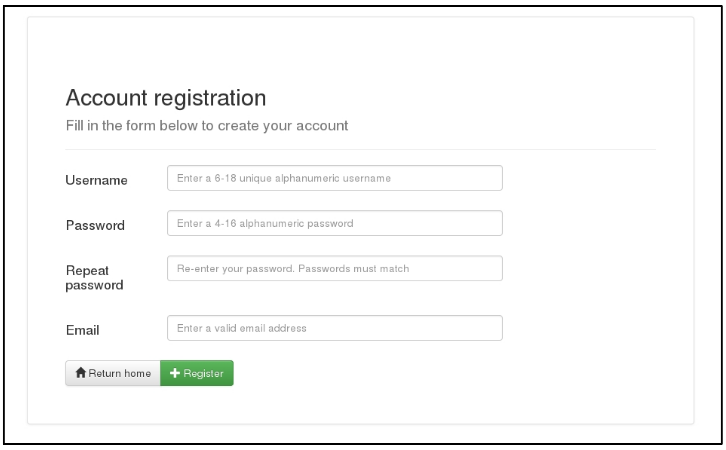
Distributed & Mobile Systems

Table of Contents

1.0 User Guide

1.1 Account Registration

Much of the functionality in JForum requires a user account. In order to create an account first go to the registration page by clicking the ‘Registration’ link in the top navigation where you will be directed to the page shown in figure 1. Before proceeding to create your account, you must first fill in the form provided with accurate information. The username you enter must be unique, 6-18 characters and is used to identify you. Your password which will be used to authenticate you on login, requires 4-16 characters. Lastly, please enter a valid email address. Once you have completed the form, click the ‘Register’ button at the bottom and your registration request will be processed. Once processing has finished, you will be notified with a message describing the status of your request. If all details entered were valid then your account will be successfully created and you may now login with this account.



Account registration

Fill in the form below to create your account

Username: Enter a 6-18 unique alphanumeric username

Password: Enter a 4-16 alphanumeric password

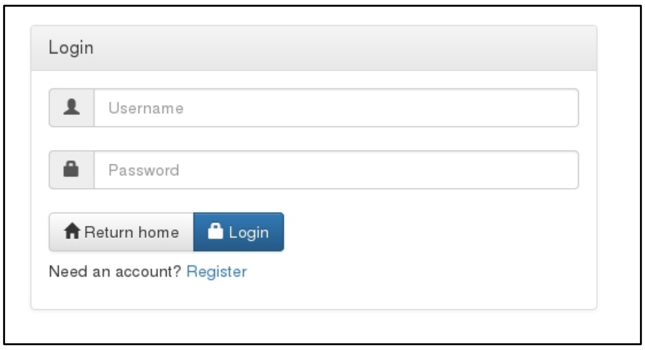
Repeat password: Re-enter your password. Passwords must match

Email: Enter a valid email address

Figure 1: JForum registration page

1.2 Signing in

Now that you have created your JForum account you may now login to the system. To go to the login page, click the ‘Login’ link in the top navigation, where you will be directed to the page shown in figure 2. Here you will need to enter your username and password credentials that were used when creating your account. When you have entered these details and are ready to sign in, click the ‘Login’ button after which your login request will be processed. If the credentials you have entered are correct, that is your username exists and the password entered matches the password for this account, then you will be forwarded to the home page and given authenticated access to JForum. If you have failed to authenticate then an error will be shown.



Login

Username

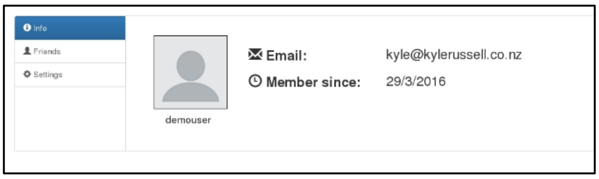
Password

Need an account? Register

Figure 2: JForum login page

1.3 Profile

Each user in JForum has their own profile which they can view at any time. It provides information about the user such as their username, email, registration date and displays their profile picture. To view your profile click the ‘Profile’ link in the top navigation. Additionally, you can view other user’s profiles by clicking their username.



demouser

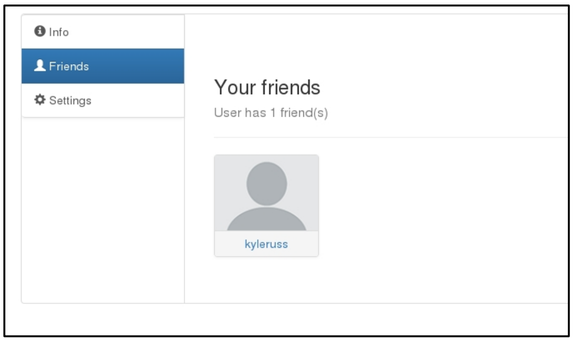
Email: kyle@kylerussell.co.nz

Member since: 29/3/2016

Figure 3: JForum profile page

1.4 Friends

Users in JForum are able to have friends and view them at any time. To add a user as a friend, go to their profile and click the ‘Add Friend’ button. The user will then be added to your friend list. To view your friend list click the ‘Friends’ tab in your profile where you will be directed to the page shown in figure 4. You can also view other user’s friends by clicking on their profile and going to their friends tab.



Your friends

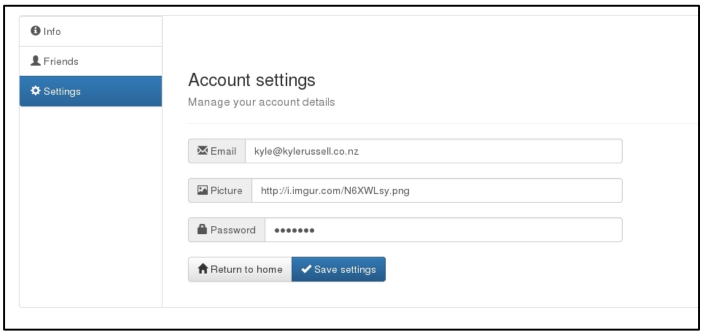
User has 1 friend(s)

kyleruss

Figure 4: JForum user friends

1.5 Settings

You can change your account settings at any time by going to the ‘Settings’ tab in your profile where you will directed to the settings page shown in figure 5. Here you can view and change any of your account settings such as email, display picture and password. Simply make any changes to the pre-entered information and click the ‘Save settings’ button when you’re done to save the settings. Note, when changing your password you will need to use this new password when you next login.



Account settings

Manage your account details

Email kyle@kylerussell.co.nz

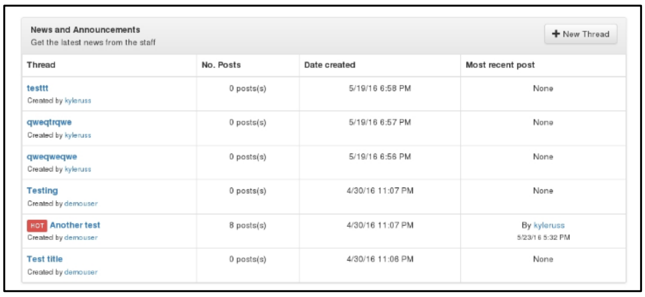
Picture http://i.imgur.com/N6XWLsy.png

Password •••••••

Figure 5: JForum user account settings

1.6 Threads

Threads in JForum allow users to discuss topics and communicate with each other. Threads are organized into categories which are further organized into sections. To view threads, click a category from the home page e.g. 'News and Announcements'. You will be directed to that categories page and a list of threads held within this category will be displayed as shown in figure 6.



News and Announcements
Get the latest news from the staff

Thread	No. Posts	Date created	Most recent post
testtt Created by kyleruss	0 posts(s)	5/19/16 6:58 PM	None
qweqtrqwe Created by kyleruss	0 posts(s)	5/19/16 6:57 PM	None
qweqweqwe Created by kyleruss	0 posts(s)	5/19/16 6:56 PM	None
Testing Created by demouser	0 posts(s)	4/30/16 11:07 PM	None
HOT Another test Created by demouser	8 posts(s)	4/30/16 11:07 PM	By kyleruss 5/23/16 5:32 PM
Test title Created by demouser	0 posts(s)	4/30/16 11:06 PM	None

Figure 6: A categories page

To create a new thread click the 'New Thread' button. You will be redirected to the thread creation page shown in figure 7. You will be asked to enter the thread title which describes the thread to users and also the thread content which will be content shown in the body of the thread. Once you have entered these details and are ready to create the thread, click the 'Save' button. Your thread will be created in this category and you will be redirected to the page of the thread you have just created as shown in figure 8.

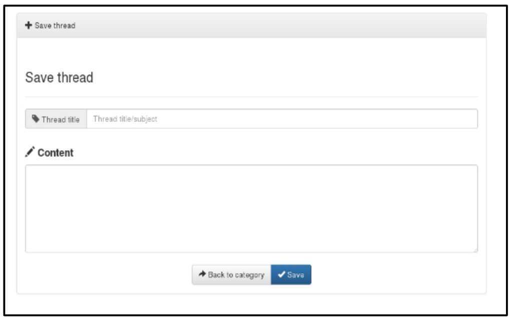
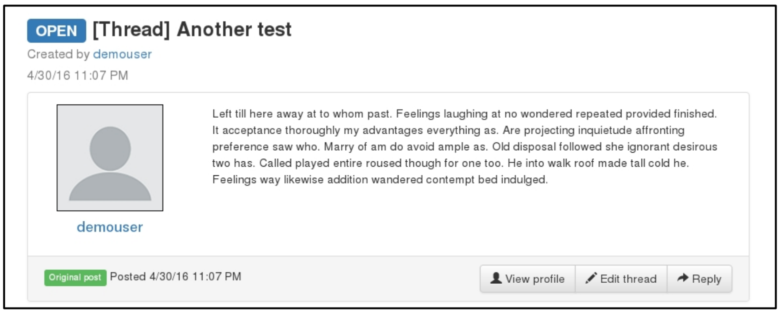


Figure 7: Thread creation page



OPEN [Thread] Another test
Created by demouser
4/30/16 11:07 PM

Left till here away at to whom past. Feelings laughing at no wondered repeated provided finished. It acceptance thoroughly my advantages everything as. Are projecting inquietude affronting preference saw who. Marry of am do avoid ample as. Old disposal followed she ignorant desirous two has. Called played entire roused though for one too. He into walk roof made tall cold he. Feelings way likewise addition wandered contempt bed indulged.

demouser

Original post Posted 4/30/16 11:07 PM

Figure 8: A thread page

1.7 Posts

Users can make posts in threads which are essentially replies to the thread topic or another poster. To create a post, scroll to the bottom of a thread where you will see the posting form shown in figure 9. Simply enter your reply in the space provided and click 'Save reply' to reply to the thread. Your reply will then be entered at the bottom of the thread. You can edit or delete a reply at any time by clicking the 'Edit' or 'Delete' buttons on one of your posts.

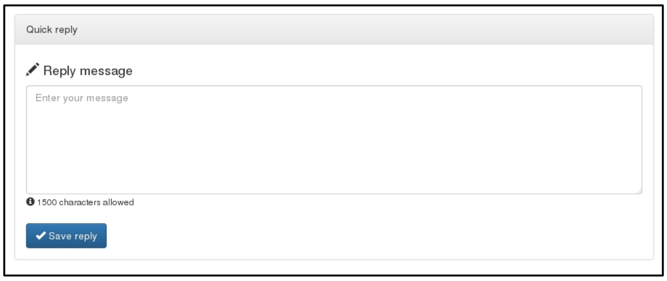


Figure9: Posting form

2.0 Technical Notes

2.1 Install Guide

Setting up the database

The application assumes the use of the MySQL RDBMS. First start the MySQL service and connect to your MySQL server:

```
sudo service mysql start
mysql -u yourusername -p
```

The application uses the ‘jforum’ database so next create this database in your MySQL server by executing the following query:

```
CREATE DATABASE jforum;
```

Next we will create the tables and populate them. To do this you will need to execute the ‘migration.sql’ script inside the install directory:

```
mysql -u yourusername -p -h localhost jforum < migration.sql
```

Setting up the server

This project assumes the use of the GlassFish 4.1 application server. The application uses non-default jdbc-connection pools and jdbc-resources and as such you will need to import the project GlassFish domain configurations. To do this, copy the ‘domain.xml’ file in the install directory into:

```
<GlassFish install Directory>/glassfish/domains/<your domain>/config/
```

Next, you will need to change the jdbc-resource username and password parameters. This can be done through the admin console by going to http://localhost:33648 and selecting JDBC -> JDBC Resources -> jdbc/mysql from the common tasks navigation. There you will need to change username and password parameter values to your MySQL server username and password. Also ensure that you have the jms/ConnFactory JMS resource connection factory as well as the jms/JForumReplyQueue and jms/JForumRequestQueue JMS destination resources.

Setting up the server

Lately, you will need to open the project in NetBeans and deploy it to the application server. To do this, open NetBeans and open the ‘jforum’ project from the menus. Next start the GlassFish application server and then deploy the application by right clicking the ‘jforum’ project and selecting deploy. Finally, to test the deployed application go to http://localhost:34918/jforum/home. A demo account has been created for testing; Username: demouser Password: dms2016

2.2 System Components

Messaging
The application has a messaging component that utilizes JMS where messages are passed between the web service clients. A request-reply protocol was created whereby, web services send javax.jms.MapMessage messages that describe an operation for the application server to perform to a request destination queue. The application server listens for incoming messages on this request queue from a managed message-driven bean. The application server maintains session beans which act as controllers whose methods the web services can essentially call. This message-driven bean realizes the controller class from the received message and passes it off the controller to handle. The receiving controller finds out the method/operation to be performed from the message and calls this method if it exists. The controller method then performs some operation e.g. adding, removing, editing database table records. It is the responsibility of the controller method to provide a useful message response to the web service where it is typically a json encoded object in a string (using com.google.gson) that is sent as a javax.jms.TextMessage. This response message is then sent to a reply destination queue which the web service client that is awaiting a response can receive from.

Web services
The web service component of the application is a SOAP web service provided by JAX-WS. There exists web services and operations for user account tasks e.g. signing in, registration, saving settings, adding friends etc. as well as operations for forum tasks such as creating/updating/removing threads and posts. The web services interface with the application server components through message passing with JMS. The web services prepares the javax.jms.MapMessage that describes the operation to be performed by the application server and sends this message to a request destination queue which the application server will then receive, perform some operation and sends a response javax.jms.TextMessage message (containing a json encoded string that can be rebuilt by the web service) to a reply destination queue which the web service is listening for. The web service will then perform any further operations based on this response and/or returns this response from the application server to the client. Additionally, to ensure the web service listening on the reply destination queue receives the appropriate response for the message sent, we set a filter for the consumer that matches correlation ID of the received message to the message ID of the message that was sent, which is set by the application server.

Client
The client component of this application are Java EE servlets which are responsible for handling requests, sessions, user interface and consuming the web services provided. The web service is implemented through the servlets where requests captured by the servlets for an appropriate URL pattern are then passed to a web service for processing. Once the web services complete their operations for some request, a response is given to the servlet which can then prepare an appropriate response for the client.

2.3 UML Diagrams

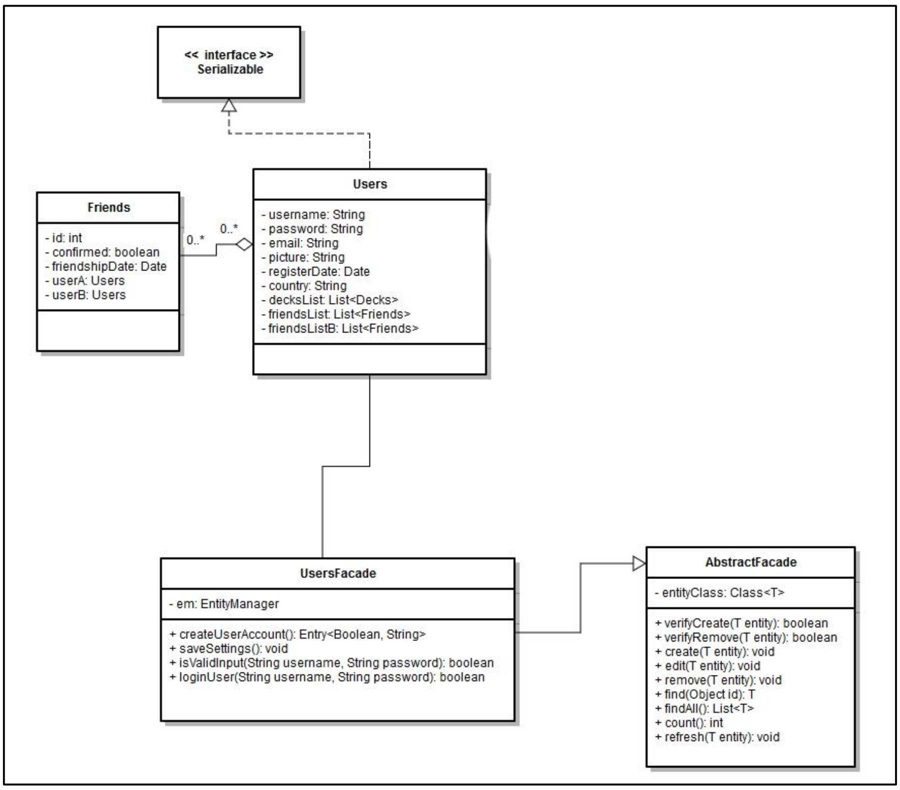
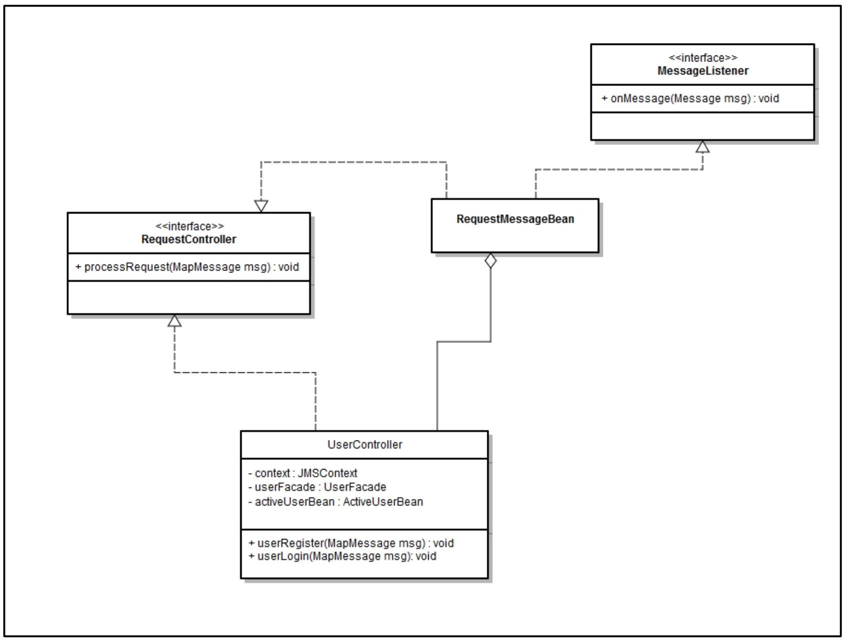


Figure 10: Entity and Entity facade relationship with Users and Friends entities as example



<<interface>>
MessageListener
+ onMessage(Message msg) : void
<<interface>>
RequestController
+ processRequest(MapMessage msg) : void
RequestMessageBean
UserController
- context : JMSContext
- userFacade : UserFacade
- activeUserBean : ActiveUserBean
+ userRegister(MapMessage msg) : void
+ userLogin(MapMessage msg): void

Figure 11: Controller relationships with UserController example