



Buzz User Manual

Compiled By

Nkosinathi Mothoa - u12077420

Nkosenhle Ncube - u13247914

Nathan Ngobale - u15110045

Kamogelo Tsipa - u13010931

Contents

1	General Information	2
2	Getting started	2
3	Using the system	3
4	Troubleshooting	9

1 General Information

System Overview

The platform is a discussion forum aimed at providing a space where users can collaborate on topics of interest. Users can pose questions to the community. Through collaboration the quality of questions and answers on the platform will improve and as a result the user benefits from the community and is able to help others. To promote collaboration and continued interest, features of gamification are applied to the platform to encourage users to contribute and participate in forum activities.

System Configuration

Installation

In order to create the project, the following steps need to be taken:

- Download and install Node.js.
- Download and install Mongo database.
- In order to install Angular CLI the following needs to be undertaken.
 1. Open up the command line and type in the following commands
 2. `npm install -g @angular/cli`
 3. `cd "to the project directory"`
 4. `npm install`
 5. `cd "to the public directory"`
 6. `npm install`
 7. `ng build`
 8. `cd "to the project directory"`
 9. `npm start`

2 Getting started

Login

In order to login to the system, a correct combination of userID and password has to be entered into the form. The userID is your student number and the password is the same password used to login into the Computer Science website. Upon a successful login, you can navigate to your profile and update details such as your your pseudonym and email. See figure 1

Logout

Once logged into the system, you can navigate to the menu bar to logout of the system.

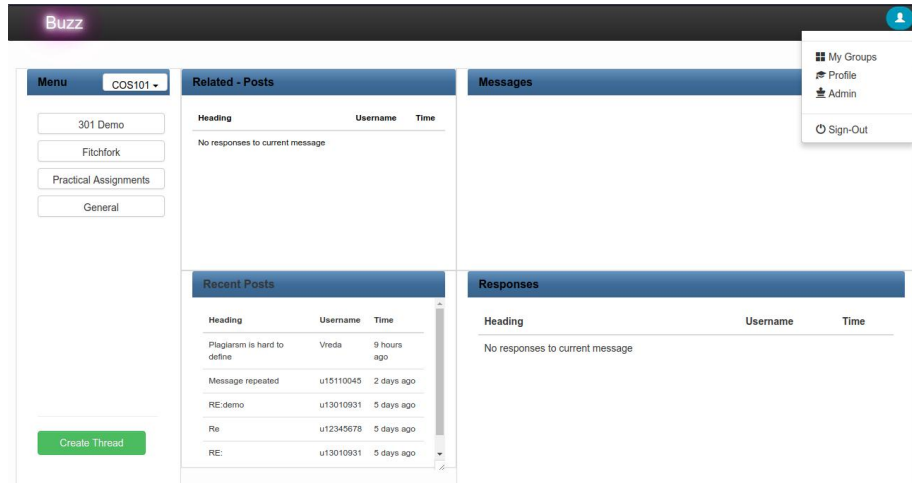


Figure 1: Logout of the system

3 Using the system

This section provides information on how to use the system. It discusses the use cases of the system, when a user would perform such a use case and why the user would want to perform this use case.

Login

This use case involves the logging in of a user in to the system. A user would need to provide a user-name and password key-value pair which will be used to authenticate this user. Figure 1 below is a view of the login screen that the user will use to login. Once the user has entered their credentials and successfully authenticated they will be taken to the system dashboard view where the remaining uses cases of the system will be relating to.

Dashboard Navigation

This section introduces the user to the layout of the system dashboard. The dashboard is partitioned into panels and each panel holds unique content that is delivered to the user. The dashboard has a dynamic design modeling the Mediator design pattern where each panel is independent in the sense that it does not communicate directly with all other panels, instead the current post

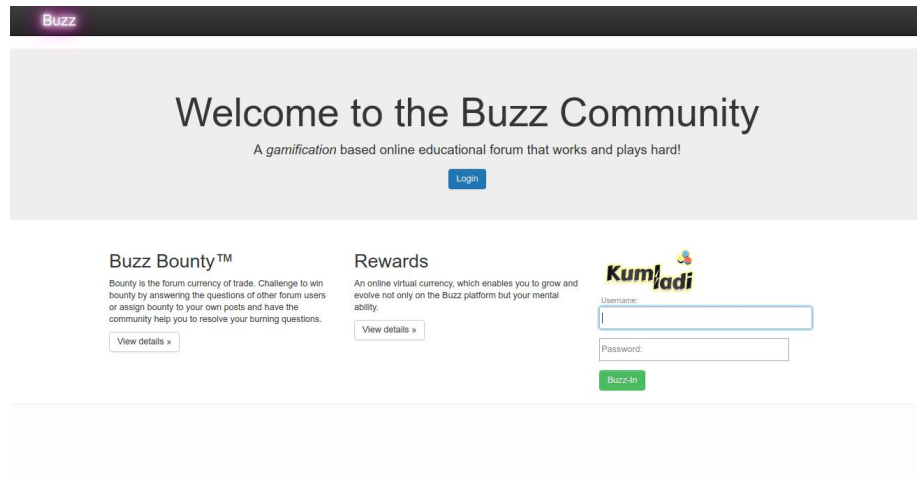


Figure 2: Login/landing page

the user is interacting with (through clicking on it) acts as a "delegator" and notifies all other panels of the content they need to display at that given time. The dashboard has in total five panels discussed below, figure 2 shows each panel and its contents labeled:

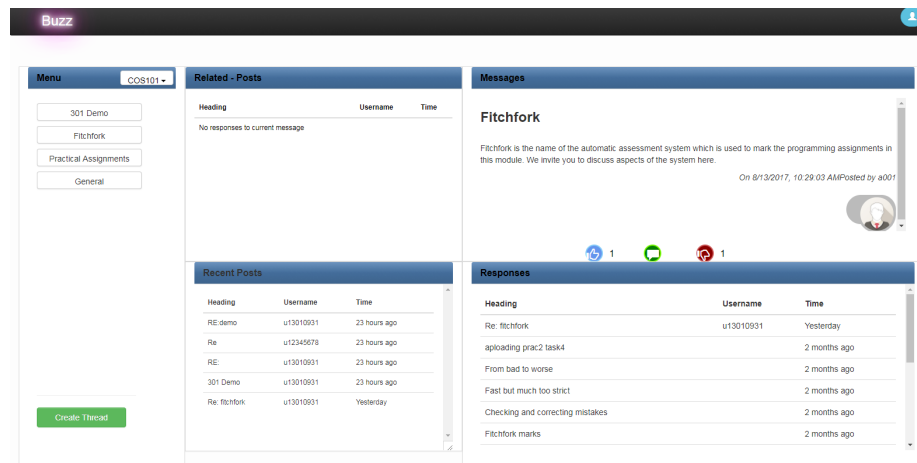


Figure 3: Dashboard

- Menu Panel: All root posts (level-zero posts) are listed in this panel, this is labeled in figure 2 below. When a new level-zero post is made it will be piled on top of the already existing level-zero posts in this panel. At the bottom of this panel we see an interactive button. This is the "Create

Thread” button and is used to start the use case of creating a new level-zero post.

- **Messages Panel:** This panel holds the content and details of the currently selected message. Selecting a level-zero post in the Menu panel will load the contents of that level-zero post into the Messages Panel. The same goes for when selecting a post in the Related, Recent or Responses Panels, the contents of the selected post will be loaded into the Messages Panel. The content displayed in the Messages Panel includes: post-heading, post-content (body of the post), the user who made the post (in the form of a user icon), the time passed since the post was made, a like and dislike button that is used to give the post a thumbs up/down. Lastly there is a reply button in the Messages panel, this button is used to make a response to a post (responding to the currently selected message). When this button is clicked, a modal form will appear that will require the use to provide the response post details before posting.
- **Response Panel:** This panel holds the replies to the currently selected message. Thus the replies to the post visible in the Messages Panel will be listed in this panel, ordered by timestamp.
- **Related-Posts Panel:** This panel holds all posts related to the post current visible in the Messages panel. This panel will not hold replies to the post currently visible in the Messages panel since we already have a panel dedicated for replies (Responses-Panel).

Reading/viewing Posts

This section describes how the user can read and/or view a post on the dashboard. The Message panel is the main viewing area of a post. This panel details the selected post giving the user better context on the selected post. All other panels mainly shows post headings, the user who made the post and the time-stamp of the post.

Create Thread (level-zero post)

To create a new thread you use the interactive button at the bottom of the Menu panel. Clicking this button will load a modal view of a form. The user must then complete the form and enter the message content before making the post. All new level-zero posts are piled up in the Menu panel.

Create Response (reply to a post)

To respond to a message, you first need to ensure that the message is the currently selected message (visible in the Message-Panel). Once the specific post is selected you need to use the ”make response” button located in the Message-Panel to load up the response modal. The contents of the modal then need to be completed before the response is made (figure 5).

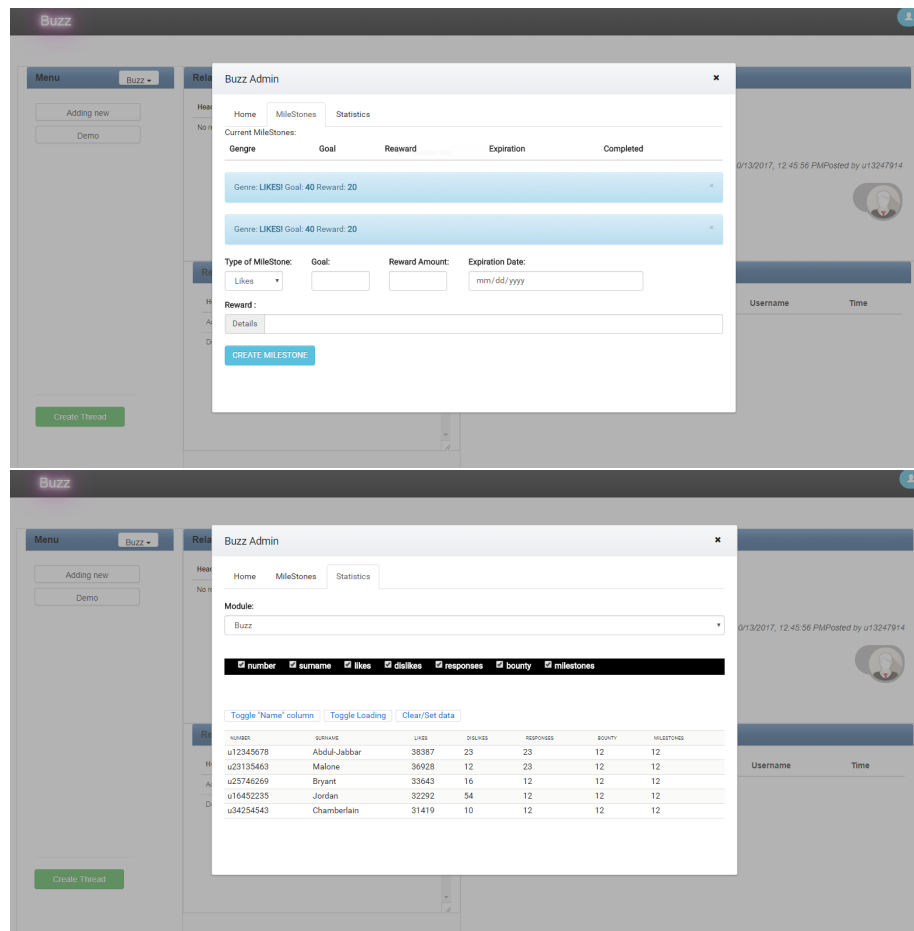


Figure 4: Milestones and stats views for an admin user

Like/dislike a post

The Message-panel has two buttons (red and green) with thumbs up/down icons. These buttons are used to give the currently selected post a thumbs up or thumbs down. To negate either action, you simply need to perform the action twice (meaning two likes/dislikes of the same message by the same user equate to zero likes/dislikes.)

Navigating between forum modules

The content on the dashboard at any given time is related to one forum module (ie: COS333 or WTW389). To switch between modules the user would need to navigate to the drop-down menu on the panel heading of the Menu panel (figure

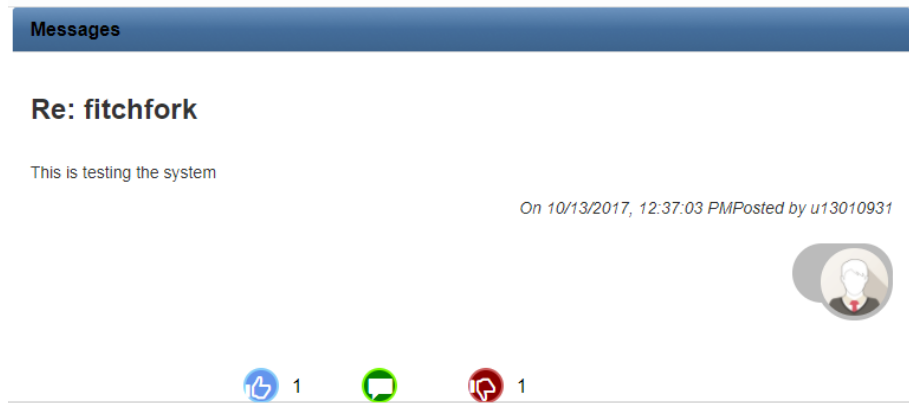


Figure 5: Messages-Panel

6). Upon clicking this drop-down, a list of the users modules will be seen and clicking on one will reload the entire dashboard and posts related to the selected module will appear in the Menu panel.

User information

The user is able to view their user information and account details by navigating to the top-right hand corner of the dashboard and clicking on the "profile icon" which will load a drop-down menu. This drop-down menu will have the following list items (figure 8):

- Profile: This option will display user details which are described below:
 - Profile Information: User information will be displayed here. This includes names, surnames, forum pseudo-name and contact details.
 - Statistics: This view will show a bar-graph representation of the users forum activity, this includes number of likes/dislikes, posts and responses.
 - Gamification: This view displays gamification information relating to the current user. This includes the rank of the user, the number of points the user has and their progress.
 - Milestones: This view will be used by admin users to create and view new milestones. Figure 3 shows both the milestones and statistics views for an admin user.
- Groups: This item will load up the Private-Group chats view where a user is able to create groups and invite other users to these groups. The user is able to create and open already existing groups in this view. Figure 8 shows the "groups" view. When a user creates a new group they are

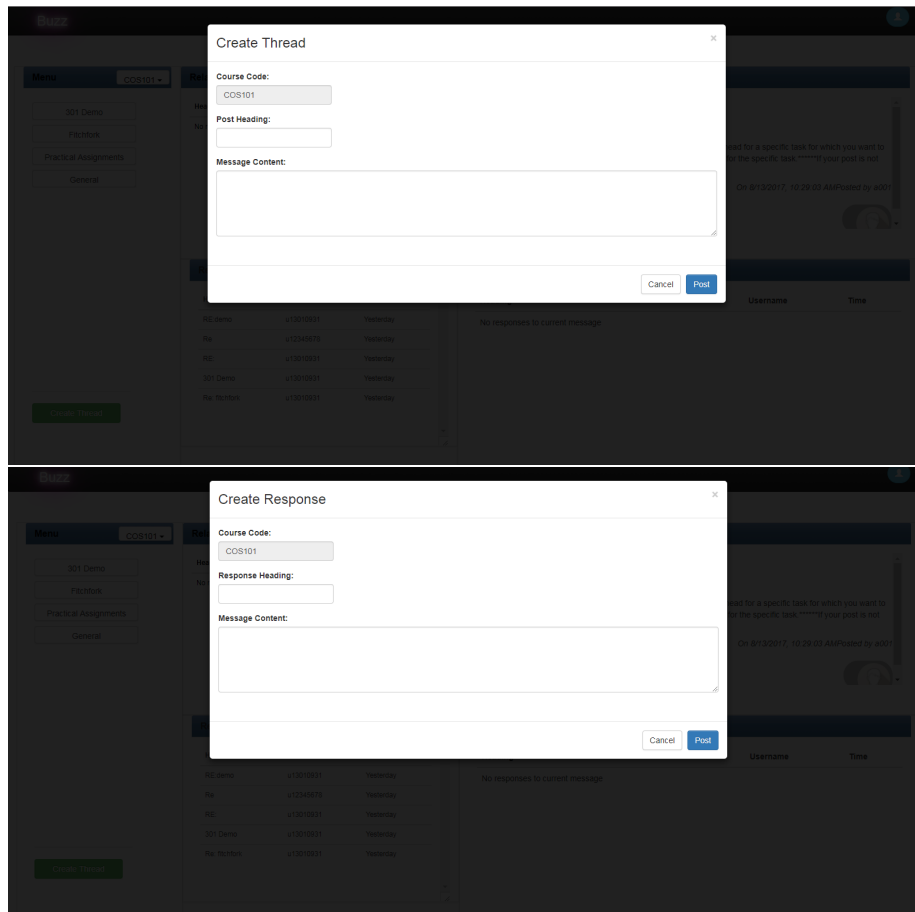


Figure 6: Create thread/response views

charged (points) thus the users points balance has to be at least that of the charge amount for the creation of a group in order for the user to successfully create the group.

- Admin: This item is visible only to admin users accounts. This loads up a modal view where the admin-user is able to see the performance and activity of specified forum users. The admin-user is also able to create forum Milestones which are achievements that the users receive rewards on when they reach these milestones. Figure 4 details the Milestones view.
- Sign-Out: This item is used to log-out of the system.

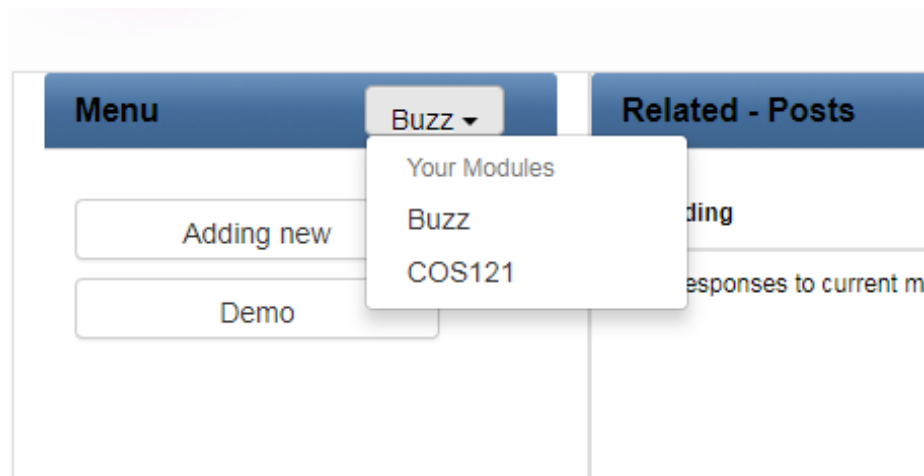


Figure 7: Module drop-down

4 Troubleshooting

Missing Parameters

This is a problem that may occur when a certain directory on the server is called but has some missing parameter when it is called. Make sure to read the doxygen of all functions within the server file to make sure that the function receives everything it needs for it to do all its operations properly. If the problem still persists, then check if the initial if-then-else block require anything else that may not have been specified accordingly.

ERRINUSE Error

This error occurs when the current port number, to which the server is meant to be listening on, has already been taken by another program. Attempt a trial and error method by assigning various port numbers to the "app.listen()" function in the server.js file. Another method is to close other existing programs that are not in use, or wait a while and try to connect to the Mongo database later on.

Connection Timed Out Error

If the server tries to connect to the MLab database(s) and it takes too long to connect it will timeout and display the "Err Connection Timed Out" error message. Make sure that the MLab site is still operational. If it is still operational, attempt to connect again, as your internet connection might have cut off while you tried to connect. If you fail to go onto the internet to check if the site is still operational, then the internet connection is the problem. If the site is no

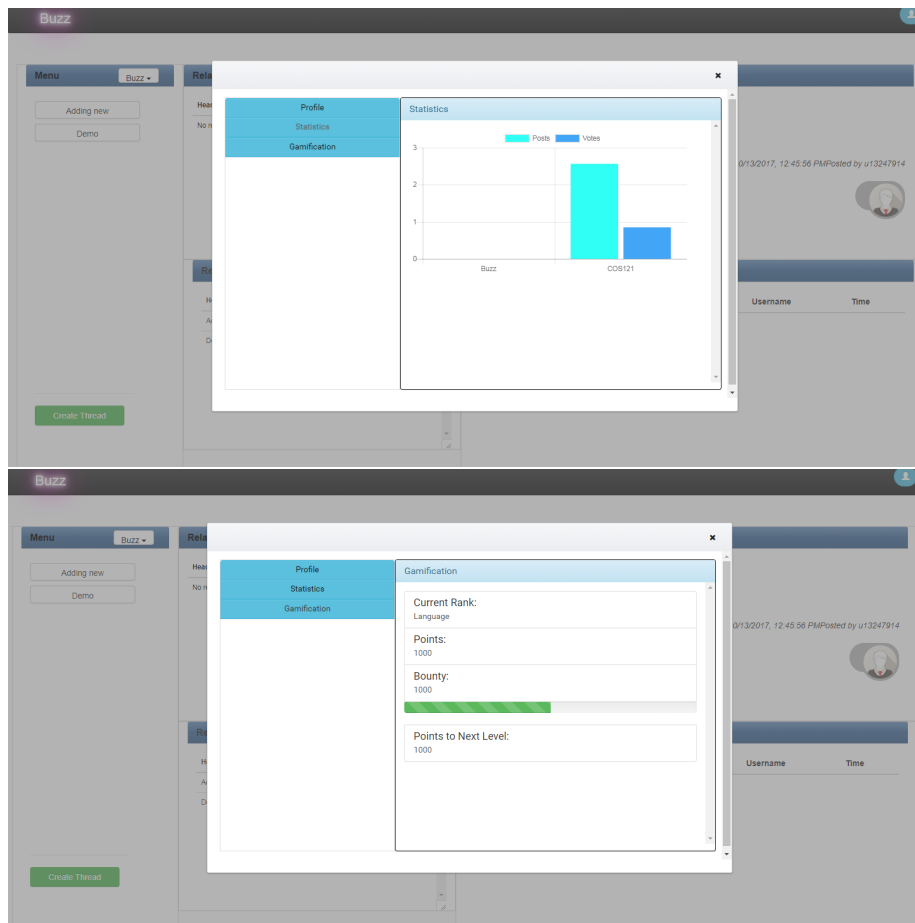


Figure 8: User stats and gamification views

longer operational then you will need to find another Mongo database hosting site and change the database reference on the server.js file.

Cross Origin Request Not Allowed

This will occur if you are trying to connect to the server file from a different URI than the one which the server is located. To best resolve this, you will need to make sure that you call the server's functions from the right URI. If this fails, then the problem may be that you are running the user interface on a different port number than the one that the server is listening on. The server.js file is the one that is meant to present the user interface, not that user opens the interface on his/her own. If the user opens the user interface on his/her ownself then this error will occur when trying to run the server at the same time.

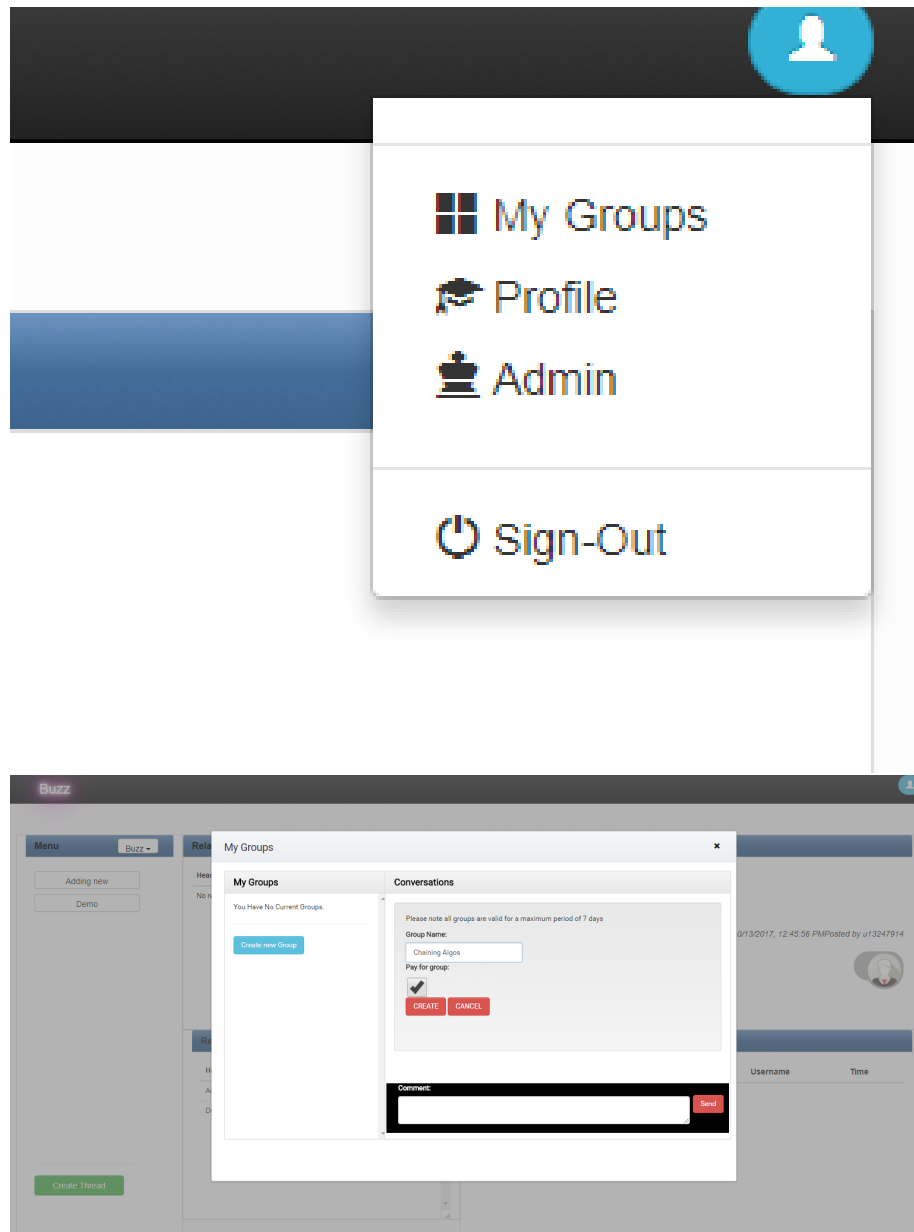


Figure 9: User options and forum groups views