

## Part I – Basic Functionality

---

### Design

#### Models

Model Name	Function name	Description
Messages_model	getMessagesByPoster(\$name)	Returns all messages posted by the user with the specified username, most recent first
Messages_model	searchMessages(\$string)	Returns all messages containing the specified search string, most recent first

#### Controllers

Controller Name	Action Name	Description
User	view(\$name)	Loads Messages_model, runs getMessagesByPoster() passing the specified name, and displays output in the ViewMessages view
Search	index()	Displays the Search view
Search	dosearch()	Loads Messages_model, retrieves search string from GET parameters, runs searchMessages() and displays the output in the ViewMessages view

#### Views

View Name	Description
ViewMessages	Displays a list of messages, with details of poster, content and time of each message. Poster's name should be linked to the user/view/{poster} for that user
Search	Displays a form with a search box, in which the user can enter search terms. The form should submit to /search/dosearch via GET

## Part II - Logging in & Posting

---

### Design

#### Models

Model Name	Function name	Description
Users_model	checkLogin(\$username,\$pass)	Returns TRUE if a user exists in the database with the specified username and password, and FALSE if not. Note: passwords in the database are SHA1 hashed, so hash supplied passwords before searching the database
Messages_model	insertMessage(\$poster, \$string)	Adds the supplied message to the messages table in the database

#### Controllers

Controller Name	Action Name	Description
User	login()	Displays the Login view
User	doLogin()	Loads the Users_model, calls checkLogin() passing POSTed user/pass & either re-displays Login view with error message, or redirects to the user/view/{username} controller to view their messages. If login is successful, a session variable containing the username is set.
User	logout()	Logs the user out, clearing their session variable, and redirects to /user/login
Message	index()	Redirects to /user/login if not logged in. Displays the Post view
Message	doPost()	Redirects to /user/login if not logged in. Loads the Messages_model, runs the insertMessage function passing the currently logged in user from your session variable, along with the posted message. Redirects to /user/view/{username} when done, which should show the user's new post

#### Views

View Name	Description
Login	Displays a login form for a user to supply their username and password. This form should submit via POST to /user/dologin
Post	Displays a form in which the user can write a new post. This form should submit via POST to the /message/doPost action

### Design

#### Models

Model Name	Function name	Description
Messages_model	getFollowedMessages(\$name)	Returns all of the messages from all of those followed by the specified user, most recent first
Users_model	isFollowing(\$follower,\$followed)	Returns TRUE if \$follower is following \$followed, FALSE otherwise
Users_model	follow(\$followed)	Inserts a row into the Following table indicating that the logged-in user follows \$followed

#### Controllers

Controller Name	Action Name	Description
User	view(\$name)	This action will need modifying to add a “follow” button or link if, after loading Users_model, the isFollowing() function indicates the currently logged in user (from your login session variable) isn’t following the viewed user. This link should point to user/follow/
User	follow(\$followed)	Loads the Users_model, and invokes the follow() function to add the entry into the database table. Should redirect back to the /user/view/{followed} page when done
User	feed(\$name)	Loads the Messages_model, invokes the getFollowedMessages() function, and puts the output into the ViewMessages view.