**Requirements**

Application should have several categories for the top -level topics (categories) like “sports”, “hr”. Each category should have several subcategories like “football” in “sports”. And the final threads should be places inside the subcategories.

Generally speaking category should have one-to-many relation to the subcategories, subcategory should have one-to-many relation to thread.

Some categories should have authenticated login access only. On the screen below those are the “hr”, “big boss” categories. Other categories should be opened for both anonymous and logged in users.

Application should have the following pages (discussed in more details further):

• Dashboard (home page)

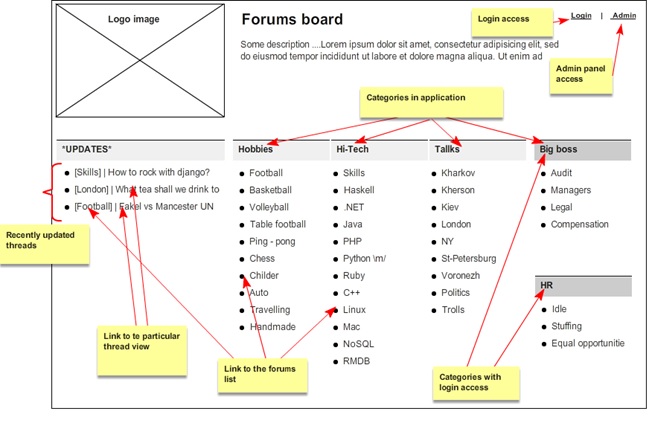
• Forums list

• Thread view

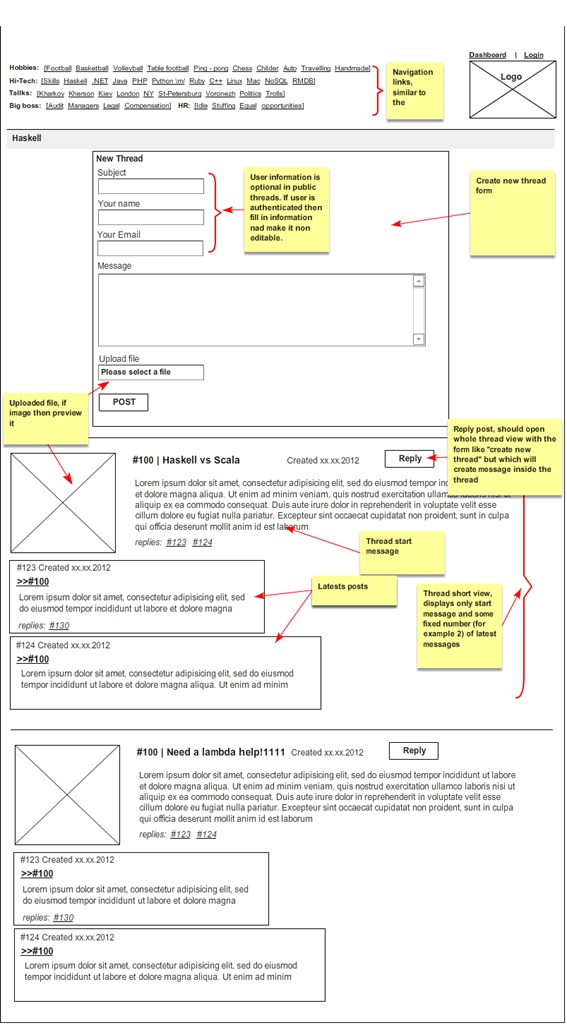
• Admin panel

There is no need to implement sophisticated HTML layout. Keep all in standard django forms. Authentication should be a standard django authentication but extended to allow both email and username as a login.   
No need to use a roles system.   
Forms should contain cross-site scripting protection.   
Database layer should be handled with django ORM.   
Layouts for the pages should use extendable templates to have one main page and specific blogs. Admin access should be some predefined login/password pair.

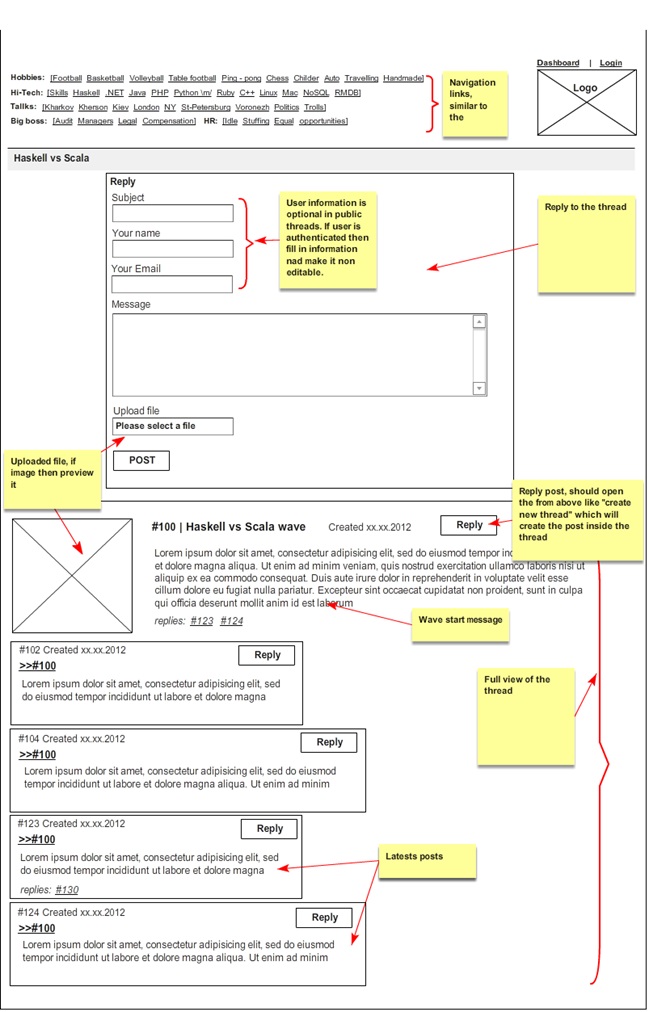
# Dashboard (Dashboard screen)



# Forums list   (Forums list screen (subcategory view))



# Thread view (Thread view screen)



# Admin panel

Admin panel should allow editing the categories and subcategories with default django approach. No additional customizations.

# Optional tasks

• Implement a thread update check and show “update available” message if there are new responses in the thread. Use AJAX calls to determine this.

• Good design is also a good feature; create couple of themes for the site. Use CSS3 styles to implement good theming; use transitions, transforms, pseudo elements, etc.

• Implement ability for Admin to alter messages on the fly, without using django admin. That task doesn’t mean that you shouldn’t use django admin.

• Create a script file to start development environment. The script should start a development server, initialize sqlite database, load database sample entries, etc.