

ISTANBUL TECHNICAL UNIVERSITY

İTÜ



Q&A Web Page

Design Specification Report

12/10/2013

Volkan İlbeyli	040100118
Gökhan Çoban	040100057
Faruk Yazıcı	040100112
Emre Gökrem	040100124
Tuğrul Yatağan	040100117

1. Introduction

This design specification report describes all data, architectural, interface and component-level design for the project. In chapter 1, introductory information about the project are provided such as goals and objectives, scope, context and some major constraints. Chapter 2 provides the design of data to be used in the project. Chapter 3 is about the architecture and the component-level design of the project which provides detailed information about the components, their sub-components, interfaces and program structure in general. Chapter 4 goes into detail of user interface design. Chapter 5 provides information about restrictions, limitation and constraints. Chapter 6 provides information about testing issues and finally the last chapter is the appendices.

1.1. Goals and Objectives

Everyone faces various challenges in life including the ones that have a solution that can be obtained easily by asking another individual about the problem rather than digging up for the answer in the details of a resource. The aim of the project is to provide a solution for those challenges by encouraging people to sign up to the web platform and ask or answer questions. Questions may be simple, or severely complicated. However the system encourages its users for all the questions having answers. Therefore, every question, easy or difficult, will have answers right in front of who browses them, thus saving time for searching in a resource.

1.2. Statement of scope

Q&A Website is a web platform where a user can sign up and browse numerous questions about numerous categories. Registered users can create a profile, register to questions that she is interested in by adding the relevant tags to her favorite tag list and browse questions that are interesting to her. She can ask questions and answer questions as well as commenting below them. Questions and answer with higher vote count are displayed first, thus helping good questions and answers to strike first. Among completion of some tasks or achieving production of high quality content, users will be rewarded by the badge system which will increase the participation of the users, help the upkeep of the system.

1.3. Software context

This project will serve as a community service, by helping people exchange information. Any other profit-seeking behaviour will be banned. By encouraging users to embrace the system and encouraging to help the maintenance is the key feature in this project. The platform will be created by its users and maintained by its users for its users once the platform is somewhat established.

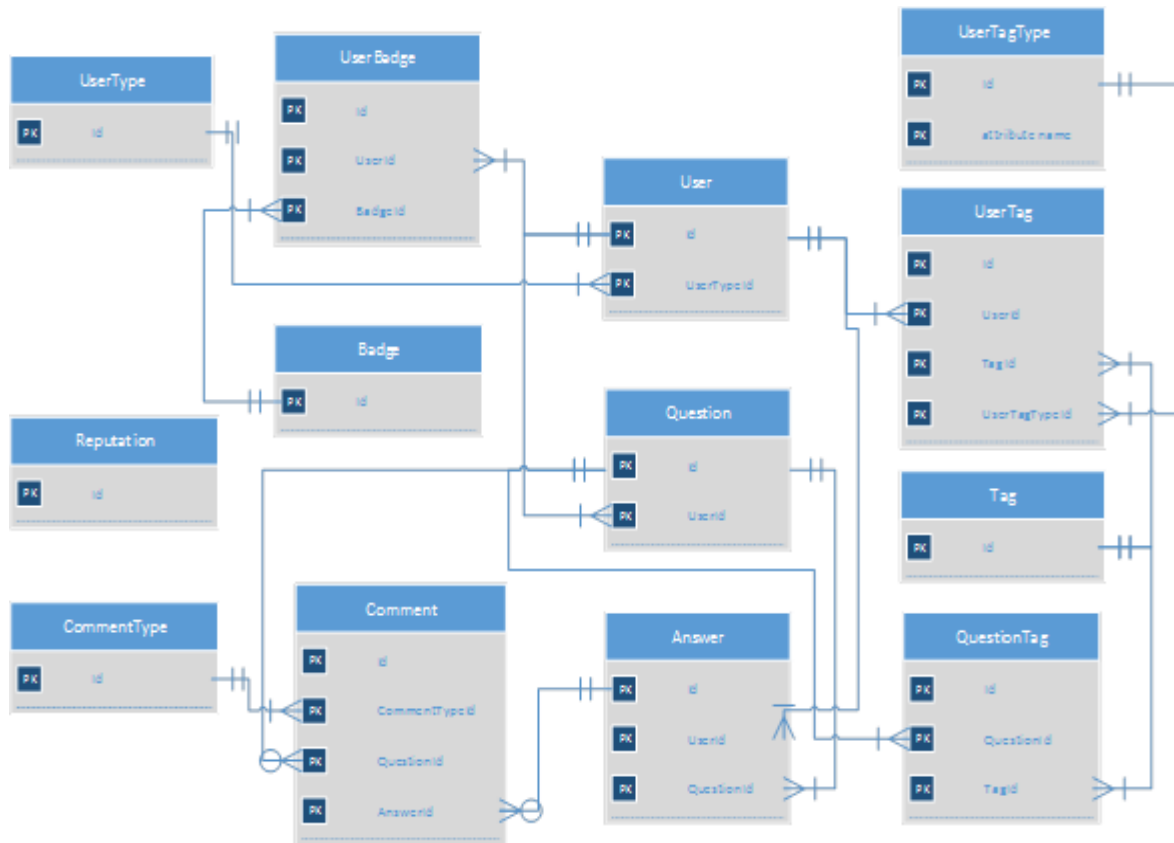
1.4. Major constraints

There are a few constraints for the project:

- Time constraint: project has to be delivered before the deadline.
- Database should be well designed to fulfill the needs of the users of the system in a feasible time interval and efficiently enough.
- Privilege restrictions should be satisfied (admin/user operations).

2. Data design

There are 13 tables designed for the project, including internal and global tables. Some table are designed for n-n relationship between tables such as QuestionTag, UserTag etc. Entity Relationship diagram for the database design is given below.



2.1. Internal software data structure

Some data structures like QuestionTag, UserTag and UserBadge are internal data structures between corresponding tables to provide n-n relationship. These data structures will not be mapped to the application side. For example, there will be no class called QuestionTag, Question class will have a relation to a list of tags.

Some other data structures like UserType, CommentType and UserTagType are internal data structures to satisfy meaningful database implementation. These data structures will be mapped to application side as *enum* variables. These variable should be synchronized to database. These tables will not be queried from application side. They will provide good understanding in further operations on database. They also will provide stability on database with foreign keys pointed to them.

2.2. Global data structure

There are 7 global data structures: User, Badge, Tag, Question, Answer, Comment and Reputation. These are major parts of the database which will offer the behaviour of the project. Almost all sides of the project will use these data structures. Also 95% of data will be in these tables. These tables should be queried carefully from the

application. Several approaches like lazy loading etc. should be used if necessary for better performance of application.

2.3. Temporary data structure

There will no temporary data structure used in the project. There is one issue can be listed in this part. Soft-delete approach will be used in the project instead of full-delete. To achieve this approach, every table will have a bit field called IsDeleted. In add operation this field will be assigned to False as default. When a delete operation called in any tables, the corresponding rows' IsDeleted bit will be assigned to True instead of running SQL DELETE command. All queries in the application side will have a WHERE parameter for IsDeleted = False. By the way, deleted rows will not be processed in the application side. These instances of classes, will be invisible to application. This approach has an issue may lead lower performance. If there is too much deleted items in tables, it could reduce the query time. These rows can be seen as temporary. Hence, one solution to this approach, a background program can run automatically in particular times, say one time in a month and delete unnecessary rows whose IsDeleted field is True.

2.4. Database description

There will be only one database for this project. The Entity Relationship diagram is given above. In the ER diagram the fields which are not primary key or foreign key are not listed for simple look. All of the data structures with all fields are given in the table below with their data type and size.

Table Name	Field	Data Type and Size
UserType	Id	Int
UserType	Name	Varchar(50)
User	Id	Guid
User	UserTypeId	Int
User	Name	Varchar(50)
User	Surname	Varchar(50)
User	Mail	Varchar(50)
User	Password	Varchar(40) => Sha1
User	Username	Varchar(30)
User	Reputation	int
User	RegisterDate	Datetime
Question	Id	Guid
Question	UserId	Guid
Question	Title	Varchar(200)
Question	Text	Varchar(MAX)
Question	CreateDate	Datetime
QuestionTag	Id	Guid
QuestionTag	QuestionId	Guid
QuestionTag	TagId	Int

Tag	Id	Int
Tag	Name	Varchar(50)
UserBadge	Id	Guid
UserBadge	UserId	Guid
UserBadge	BadgeId	Int
Badge	Id	Int
Badge	Name	Varchar(50)
Badge	InformationText	Varchar(MAX)
Badge	ImagePath	Varchar(100)
Answer	Id	Guid
Answer	UserId	Guid
Answer	QuestionId	Guid
Answer	Text	Varchar(MAX)
Answer	CreateDate	Datetime
Comment	Id	Guid
Comment	CommentTypeId	Int
Comment	QuestionId (nullable)	Guid
Comment	AnswerId (nullable)	Guid
Comment	Text	Varchar(MAX)
Comment	CreateDate	Datetime
CommentType	Id	Int
CommentType	Name	Varchar(50)
UserTag	Id	Guid
UserTag	UserTagTypeId	Int
UserTag	TagId	Int
UserTag	UserId	Guid
UserTagType	Id	Int
UserTagType	Name	Varchar(50)
Reputation	Id	Int
Reputation	Name	Varchar(50)
Reputation	InformationText	Varchar(MAX)
Reputation	Value	Int

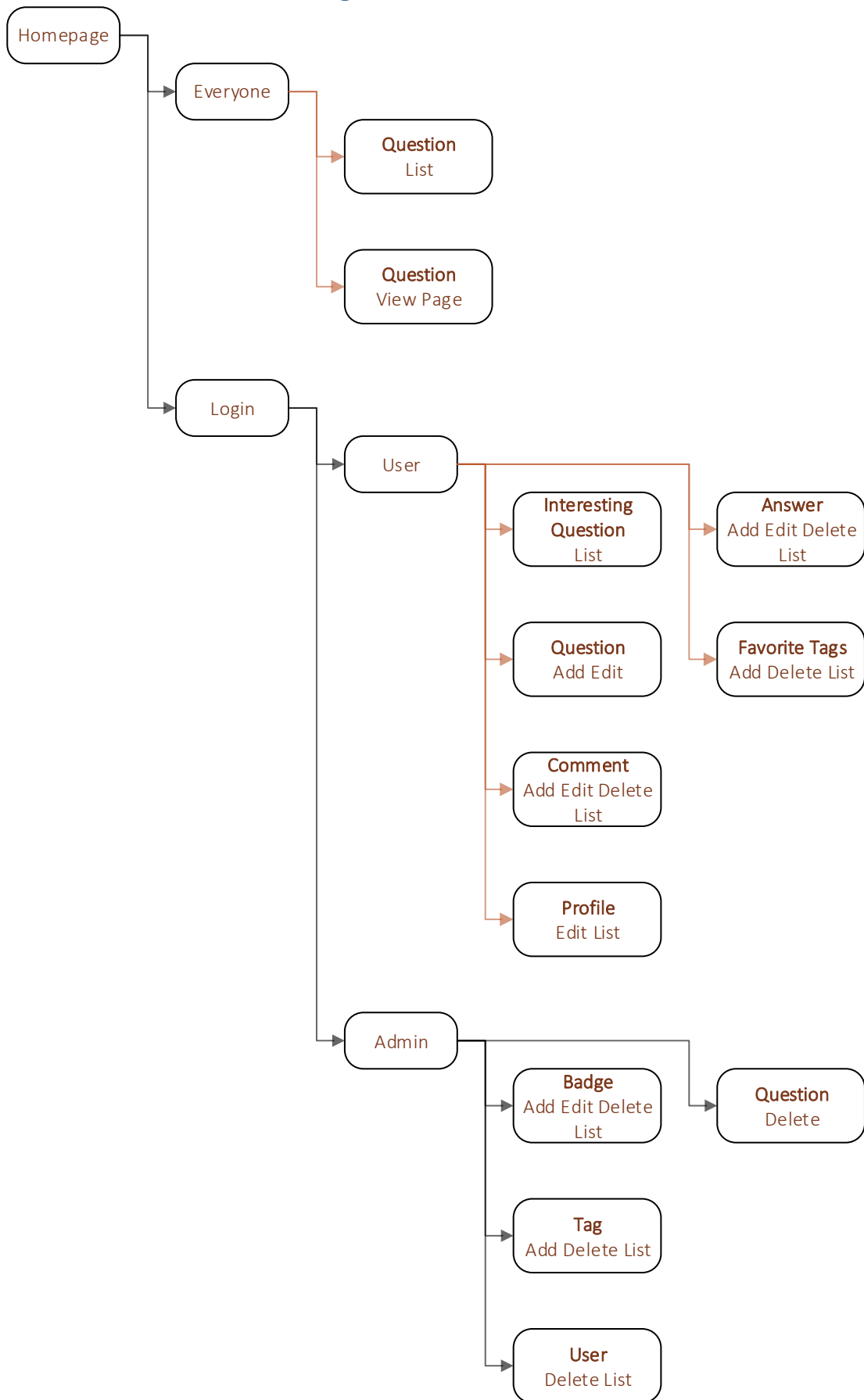
3. Architectural and component-level design

A description of the program architecture is presented in this section.

3.1. Program Structure

When a user enters the website, the first module she interacts with is the question listing module where the hottest questions are listed. She can only see the list of questions and view the question page where its answers are listed. If one is seeking more content from the system, then she must log in by using the login module. Once logged in, a user can view, edit or delete her profile using the profile CRUD module; add, list or delete favorite tags using favorite tag CRUD module; like everyone else list questions and view questions, list questions using the favorite tag filter, ask a question or edit her own questions using question CRUD module; provide answers to questions, edit her own previous answers, delete her own answers or by viewing a question, she can list the answers belonging to a question by using question answer CRUD module, add, edit her own, delete her own comments and may list all comments belonging to an answer or question by using comment CRUD module. If the user is an admin, she can create, edit, list and delete tags and badges using tag CRUD module and badge CRUD module. She can use the modules what every regular user can use. In addition she can delete questions and answers. Finally, an admin can delete users using delete user sub-module of the user CRUD module.

3.1.1. Architecture diagram



3.2. Component Descriptions

Below is the full list of detailed component descriptions.

3.2.1. Description for Badge CRUD component

Using this component an admin may create, list, edit or delete badges which are for awarding users.

3.2.1.1. Processing narrative (PSPEC) for Badge CRUD component

Upon calling this module the list of badges are displayed in the web page, each badge containing edit and delete buttons associated with them. When delete function is called, the user is redirected to delete page, the item is deleted and user is, again, redirected to the listing page. When edit function is called, user is redirected to badge edit page where she can change the values the badge contains. There is also a separate button called add badge, which calls the add badge function when pressed and redirects user to add badge page on which the user can create a new badge. Edit and add badge pages are identical except for the edit page's fields are already filled upon calling.

3.2.1.2. Badge CRUD Component interface description

Like every CRUD page, badge CRUD page will be built on a base page in which there is a table for listing page and each listed item containing two buttons (edit and delete). Add and edit pages will have the same layout, containing textboxes for field values of the item, and a save button. Delete page will have no inputs or outputs.

3.2.1.3. Sub-Component Badge List Page of Badge CRUD component processing detail

When badge listing page is called from another page, the page is loaded with the items requested from the database.

3.2.1.3.1. Interface description

Table HTML component will be used to store the results returned from the database. The table will have the operations, name and description fields as headers and will contain the returned items as table data. The operations column will contain the two buttons, which, on click, will redirect the user to relevant pages.

3.2.1.3.2. Restrictions/limitations

There is no privilege-based restrictions or limitations since listing of the badges is a public operation.

3.2.1.3.3. Local data structures

A badge class will be implemented, containing the field id, name, information text and image path along with getters and setters of those fields.

3.2.1.3.4. Performance issues

There are no issues that may affect the run-time performance, security, or computational accuracy of this sub-component unless the database returns the result of a query in a feasible amount of time.

3.2.1.3.5. Design constraints

The HTML/AJAX/JavaScript components should be able to run cross-platform, i.e. on different browsers, correctly.

3.2.1.4. *Sub-Component Badge Add Page of Badge CRUD component processing detail*

When add badge page is called from another page, the page is loaded with the empty textboxes along with its labels representing the fields of the badge item.

3.2.1.4.1. Interface description

Upon the calling of badge creation page, a header will specify the pages purpose, i.e. Add New Badge. Below in the body field of the page will be labels and empty textboxes, each corresponding to a field of the entity. Once the admin fills up the empty boxes, she may click the save button which is located under the form which has the textboxes as the input field.

3.2.1.4.2. Restrictions/limitations

Only admins can create badges. Besides this, since badge name and information text are variables of type string, there will be no restrictions.

3.2.1.4.3. Local data structures

A badge class will be implemented, containing the field id, name and information text along with getters and setters of those fields.

3.2.1.4.4. Performance issues

There are no issues that may affect the run-time performance, security, or computational accuracy of this sub-component unless the database returns the result of a query in a feasible amount of time.

3.2.1.4.5. Design constraints

The HTML/AJAX/JavaScript components should be able to run cross-platform, i.e. on different browsers, correctly.

3.2.1.5. *Sub-Component Badge Edit Page of Badge CRUD component processing detail*

When edit badge page is called from another page, the page is loaded with the textboxes filled with the values of the item being edited along with its labels representing the fields of the tag item.

3.2.1.5.1. Interface description

Upon the calling of badge edit page, a header will specify the pages purpose, i.e. Edit Badge. Below in the body field of the page will be labels and textboxes filled with the values of the item being edited, each corresponding to a field of the entity. Once the admin done editing, she may click the save button which is located under the form which has the textboxes as the input field.

3.2.1.5.2. Restrictions/limitations

Only admins can create badges. Besides this, since badge name and information text are variables of type string, there will be no restrictions.

3.2.1.5.3. Local data structures

A badge class will be implemented, containing the field id, name and description along with getters and setters of those fields.

3.2.1.5.4. Performance issues

There are no issues that may affect the run-time performance, security, or computational accuracy of this sub-component unless the database returns the result of a query in a feasible amount of time.

3.2.1.5.5. Design constraints

The HTML/AJAX/JavaScript components should be able to run cross-platform, i.e. on different browsers, correctly.

3.2.1.6. Sub-Component Badge Delete of Badge CRUD component processing detail

When badge delete page is called, the user is directed to this page, the item which directed user to this page is deleted, and the user is redirected to the listing page.

3.2.1.6.1. Interface description

There is no interface for this page. The page will return to listing page as soon as the query is sent to the database.

3.2.1.6.2. Restrictions/limitations

Only admins are able to display this page since deleting badges is an admin-only operation.

3.2.1.6.3. Local data structures

A badge class will be implemented, containing the field id, name and information text along with getters and setters of those fields.

3.2.1.6.4. Performance issues

There are no issues that may affect the run-time performance, security, or computational accuracy of this sub-component unless the database returns the result of a query in a feasible amount of time.

3.2.1.6.5. Design constraints

There are no design constraints for this page, since nothing is displayed.

3.2.2. Description for Tag CRUD Component

Using this component an admin may create, list, edit or delete tags which are for identifying questions.

3.2.2.1. Processing narrative (PSPEC) for Tag CRUD component

Upon calling this module the list of tags are displayed in the web page, each tag containing edit and delete buttons associated with them. When delete function is called, the user is redirected to delete page, the item is deleted and user is, again, redirected to the listing page. When edit function is called, user is redirected to tag edit page where she can change the values the tag contains. There is also a separate button called add tag button, which calls the add tag function when pressed and redirects user to tag creation page on which the user can create a new tag. Edit and add tag pages are identical except for the edit page's fields are already filled upon calling.

3.2.2.2. Tag CRUD Component interface description

Like every CRUD page, tag CRUD page will be built on a base page in which there is a table for listing page and each listed item containing two buttons (edit and delete). Add and edit pages will have the same layout, containing textboxes for field values of the item, and a save button. Delete page will have no inputs or outputs.

3.2.2.3. Sub-Component Tag Listing Page of Tag CRUD component processing detail

When tag listing page is called from another page, the page is loaded with the items requested from the database.

3.2.2.3.1. Interface description

Table HTML component will be used to store the results returned from the database. The table will have the operations, name and description fields as headers and will contain the returned items as table data. The operations column will contain the two buttons, which, on click, will redirect the user to relevant pages.

3.2.2.3.2. Restrictions/limitations

There is no privilege-based restrictions or limitations since listing of the tags is a public operation.

3.2.2.3.3. Local data structures

A tag class will be implemented, containing the field id, name and description along with getters and setters of those fields.

3.2.2.3.4. Performance issues

There are no issues that may affect the run-time performance, security, or computational accuracy of this sub-component unless the database returns the result of a query in a feasible amount of time.

3.2.2.3.5. Design constraints

The HTML/AJAX/JavaScript components should be able to run cross-platform, i.e. on different browsers, correctly.

3.2.2.4. Sub-Component Tag Creation Page of Tag CRUD component processing detail

When add tag page is called from another page, the page is loaded with the empty textboxes along with its labels representing the fields of the tag item.

3.2.2.4.1. Interface description

Upon the calling of tag creation page, a header will specify the pages purpose, i.e. Add New Tag. Below in the body field of the page will be labels and empty textboxes, each corresponding to a field of the entity. Once the admin fills up the empty boxes, she may click the save button which is located under the form which has the textboxes as the input field.

3.2.2.4.2. Restrictions/limitations

Only admins can create tags. Besides this, since tag name and description are variables of type string, there will be no restrictions.

3.2.2.4.3. Local data structures

A tag class will be implemented, containing the field id, name and description along with getters and setters of those fields.

3.2.2.4.4. Performance issues

There are no issues that may affect the run-time performance, security, or computational accuracy of this sub-component unless the database returns the result of a query in a feasible amount of time.

3.2.2.4.5. Design constraints

The HTML/AJAX/JavaScript components should be able to run cross-platform, i.e. on different browsers, correctly.

3.2.2.5. Sub-Component Tag Update Page of Tag CRUD component processing detail

When edit tag page is called from another page, the page is loaded with the textboxes filled with the values of the item being edited along with its labels representing the fields of the tag item.

3.2.2.5.1. Interface description

Upon the calling of tag edit page, a header will specify the pages purpose, i.e. Edit Tag. Below in the body field of the page will be labels and textboxes filled with the values of the item being edited, each corresponding to a field of the entity. Once the admin done editing, she may click the save button which is located under the form which has the textboxes as the input field.

3.2.2.5.2. Restrictions/limitations

Only admins can create tags. Besides this, since tag name and description are variables of type string, there will be no restrictions.

3.2.2.5.3. Local data structures

A tag class will be implemented, containing the field id, name and description along with getters and setters of those fields.

3.2.2.5.4. Performance issues

There are no issues that may affect the run-time performance, security, or computational accuracy of this sub-component unless the database returns the result of a query in a feasible amount of time.

3.2.2.5.5. Design constraints

The HTML/AJAX/JavaScript components should be able to run cross-platform, i.e. on different browsers, correctly.

3.2.2.6. Sub-Component Tag Delete Page of Tag CRUD component processing detail

When tag delete page is called, the user is directed to this page, the item which directed user to this page is deleted, and the user is redirected to the listing page.

3.2.2.6.1. Interface description

There is no interface for this page. The page will return to listing page as soon as the query is sent to the database.

3.2.2.6.2. Restrictions/limitations

Only admins are able to display this page since deleting tags is an admin-only operation.

3.2.2.6.3. Local data structures

A tag class will be implemented, containing the field id, name and description along with getters and setters of those fields.

3.2.2.6.4. Performance issues

There are no issues that may affect the run-time performance, security, or computational accuracy of this sub-component unless the database returns the result of a query in a feasible amount of time.

3.2.2.6.5. Design constraints

There are no design constraints for this page, since nothing is displayed.

3.2.3. Description for Favorite Tag CRUD Component

3.2.3.1. Processing narrative (PSPEC) for component Favorite Tag CRUD

In this page, users can process edit own tags, add and delete tag processes. Users see own tag list and they add new tag to this list. In addition, if they have a tag, they edit own tag or completely delete it. After users finish process, tag list in this page is again listed. Users don't see others' tag, they only edit own tag list.

3.2.3.2. Component Favorite Tag CRUD interface description.

This page has tag list for user. In this list, there are tags list, add new tag button and delete tag button and main navigation panel. In tag list, users see tag' name. Second interface is adding new tag for user. Users click adding button and go to create new tag area then tag is created and added to list by system. Third interface is deleting own tag for user.

3.2.3.3. Sub-Component Tag List of Favorite Tag CRUD component processing detail

When tag listing page is called from another page, the page is loaded with the items requested from the database. Tags listing shows to users' tags for current user. This list is changeable, because users add new tags or deleting own tags.

3.2.3.3.1. Interface description

Main navigation panel is same for all process. Listing interface has tag names.

3.2.3.3.2. Algorithmic model (e.g., PDL)

For listing, system is takes tags from database for current user and lists in table with tag values. List is changeable depending on other process (adding and deleting tags).

3.2.3.3.3. Restrictions/limitations

Only the user can see her favorite tags from the favorite tags listing page.

3.2.3.3.4. Local data structures

For listing, using user and tag data structure.

3.2.3.3.5. Performance issues

Size of tags list is important arbiter for system running time performance.

3.2.3.3.6. Design constraints

The HTML/AJAX/JavaScript components should be able to run cross-platform, i.e. on different browsers, correctly.

3.2.3.4. Sub-Component Tag Add of Favorite Tag CRUD component processing detail

When tag adding page is called from this page, the page is loaded with the items requested from the database. Adding tag process adds to tags list for current user. Users click add button then open a pop-up, users select their needing tags that is in pop-up page, finally confirm own tags and send. Then tags are added to database and page is redirected to last page.

3.2.3.4.1. Interface description

Main navigation panel is same for all process. But add process has different pop-up panel.

3.2.3.4.2. Algorithmic model (e.g., PDL)

For adding, system takes new tags from user in add pop-up. Then it is added to database fairly by system.

3.2.3.4.3. Restrictions/limitations

User add anyone tag for current user.

3.2.3.4.4. Local data structures

For adding, using tags data structure

3.2.3.4.5. Performance issues

Size of tags list is important arbiter for system running time performance.

3.2.3.4.6. Design constraints

The HTML/AJAX/JavaScript components should be able to run cross-platform, i.e. on different browsers, correctly.

3.2.3.5. Sub-Component Tag Delete of Tag CRUD component processing detail

When tag deleting page is called from this page, the page is loaded with the items requested from the database. Deleting tag process deletes to tags list for current user. Users click delete button. Then tag is deleted to database and page is redirected to last page.

3.2.3.5.1. Interface description

Main navigation panel is same for all process.

3.2.3.5.2. Algorithmic model (e.g., PDL)

For deleting, system takes current tags from user in tags list. Then it is deleted from database fairly by system.

3.2.3.5.3. Restrictions/limitations

User delete anyone tag for current user.

3.2.3.5.4. Local data structures

For deleting, using tags data structure

3.2.3.5.5. Performance issues

Size of tags list is important arbiter for system running time performance.

3.2.3.5.6. Design constraints

The HTML/AJAX/JavaScript components should be able to run cross-platform, i.e. on different browsers, correctly.

3.2.4. Description for Questions List Component

3.2.4.1. Processing narrative (PSPEC) for component Interesting Questions List

This is the part where the questions are listed according to users' ignored tags. Each user has an ignored tag list which he/she creates. Based on this list, the results which exclude the questions with ignored tags are listed on this page. The user is able to access only her/his questions list page. They may access others' ignored tag list, however cannot view the questions list according to that list. The questions list is created and it is viewable only after the user has logged in a session.

3.2.4.2. Component Questions List interface description.

This page is mainly composed of a list that includes the questions which doesn't care about the ignored list of the current user. It's layout is the same as a question list page, such as on the home page or on a result page of a search command. In more detail, there is a list of clickable question headers. Below each question header, the part of question content that fits the field is written. And at the bottom, there is the page buttons for accessing next pages. No extra buttons are required. Other parts on this page are the non-specific parts like navigation bar.

3.2.4.3. Sub-Component Question List of Questions List component processing detail

When question listing page is called from another page, the page is loaded with the items requested from the database. Question listing shows the questions according to favorite tags for current user. This list is changeable, because new questions might be added or current questions might be deleted.

3.2.4.3.1. Interface description

Main navigation panel is same for all process. Listing interface has question header names, and a part of the question content below.

3.2.4.3.2. Algorithmic model (e.g., PDL)

For listing, system is takes questions from database then lists in table with the questions. List is changeable depending on other process (adding and deleting questions).

3.2.4.3.3. Restrictions/limitations

A user can view all the questions on the page.

3.2.4.3.4. Local data structures

User data structure is used overall the process. For listing, question data structure is used.

3.2.4.3.5. Performance issues

Size of questions list is important regarding the performance. Because as the questions list become larger according to a tag, the performance shall be slowed down.

3.2.4.3.6. Design constraints

The HTML/AJAX/JavaScript components should be able to run cross-platform, i.e. on different browsers, correctly.

3.2.4.4. Sub-Component See Answer of Questions List component processing detail

When question title is clicked from questions list page, the page is loaded with the items requested from the database, which are the entire question and the answers below. Question/answer viewing page is only changeable when the question/answer is edited or deleted.

3.2.4.4.1. Interface description

Main navigation panel is same for all process. Question/answer interface has question header, entire of the question content below and the entire of the answers respectively below the question content.

3.2.4.4.2. Algorithmic model (e.g., PDL)

For question/answer viewing, system takes the question and the answers of the question from database. Page is changeable depending on other process (deleting questions/answers).

3.2.4.4.3. Restrictions/limitations

There is no restriction or limitation.

3.2.4.4.4. Local data structures

User data structure is used overall the process. For listing, question and answer data structures are used.

3.2.4.4.5. Performance issues

Size of answers list is the only important part regarding the performance.

3.2.4.4.6. Design constraints

The HTML/AJAX/JavaScript components should be able to run cross-platform, i.e. on different browsers, correctly.

3.2.4.5. Sub-Component See Comment of Interesting Questions List component processing detail

When question title is clicked from questions list page, the page is loaded with the items requested from the database, which are the entire question and the answers below. Then behind the question, there is an option to view the comments. Comments part is only changeable when the comment is added, or deleted.

3.2.4.5.1. Interface description

Main navigation panel and question/answer part are same for all process. Comments interface has the content of the comments respectively below the question.

3.2.4.5.2. Algorithmic model (e.g., PDL)

For comment viewing, system takes the comments of the question from database. Page is changeable depending on other process (adding/deleting comments).

3.2.4.5.3. Restrictions/limitations

There is no restriction or limitation.

3.2.4.5.4. Local data structures

User data structure is used overall the process. For listing comments, comment data structure is used.

3.2.4.5.5. Performance issues

Size of comments list is the only important part regarding the performance.

3.2.4.5.6. Design constraints

The HTML/AJAX/JavaScript components should be able to run cross-platform, i.e. on different browsers, correctly.

3.2.5. Description for Interesting Questions List Component

3.2.5.1. Processing narrative (PSPEC) for component Interesting Questions List Component

This is the part where the questions are listed according to users' favorite tags. Each user has a favorite tag list which he/she creates. Based on this list, most appropriate results which are related to the tags are listed on this page. The user is able to access only her/his interesting questions list page. They

may access others' favorite tag list, however cannot view the questions list according to that list. The interesting questions list is created and it is viewable only after the user has logged in a session.

3.2.5.2. Component Interesting Questions List interface description.

This page is mainly composed of a list that includes the questions according to the favorite tags of the current user. It's layout is the same as a question list page, such as on the home page or on a result page of a search command. In more detail, there is a list of clickable question headers. Below each question header, the part of question content that fits the field is written. And at the bottom, there is the page buttons for accessing next pages. No extra buttons are required. Other parts on this page are the non-specific parts like navigation bar.

3.2.5.3. Sub-Component Question List of Interesting Questions List component processing detail

When question listing page is called from another page, the page is loaded with the items requested from the database. Question listing shows the questions according to favorite tags for current user. This list is changeable, because users might add new tags or delete own tags, or new questions might be added/current questions might be deleted which are suitable with the tag.

3.2.5.3.1. Interface description

Main navigation panel is same for all process. Listing interface has question header names, and a part of the question content below.

3.2.5.3.2. Algorithmic model (e.g., PDL)

For listing, system is takes tags from database for current user, filters the questions database according to these tags, then lists in table with the questions. List is changeable depending on other process (adding and deleting tags, adding and deleting questions).

3.2.5.3.3. Restrictions/limitations

A user can view all the questions on the page according to the tags. He/she may not view other user's list of interesting questions.

3.2.5.3.4. Local data structures

User data structure is used overall the process. For listing, question data structure is used, for filtering with the tag, tag data structure is used.

3.2.5.3.5. Performance issues

Size of questions list is important regarding the performance. Because as the questions list become larger according to a tag, the performance shall be slowed down.

3.2.5.3.6. Design constraints

The HTML/AJAX/JavaScript components should be able to run cross-platform, i.e. on different browsers, correctly.

3.2.5.4. Sub-Component See Answer of Interesting Questions List component processing detail

When question title is clicked from questions list page, the page is loaded with the items requested from the database, which are the entire question and the answers below. Question/answer viewing page is only changeable when the question/answer is edited or deleted.

3.2.5.4.1. Interface description

Main navigation panel is same for all process. Question/answer interface has question header, entire of the question content below and the entire of the answers respectively below the question content.

3.2.5.4.2. Algorithmic model (e.g., PDL)

For question/answer viewing, system takes the question and the answers of the question from database. Page is changeable depending on other process (deleting questions/answers).

3.2.5.4.3. Restrictions/limitations

There is no restriction or limitation since this page is available within the user's session, which is the restriction of the previous level. However, even though the question might not be accessed through the favorite tags list, it is still accessible from the ordinary questions page.

3.2.5.4.4. Local data structures

User data structure is used overall the process. For listing, question and answer data structures are used.

3.2.5.4.5. Performance issues

Size of answers list is the only important part regarding the performance.

3.2.5.4.6. Design constraints

The HTML/AJAX/JavaScript components should be able to run cross-platform, i.e. on different browsers, correctly.

3.2.5.5. Sub-Component See Comment of Interesting Questions List component processing detail

When question title is clicked from questions list page, the page is loaded with the items requested from the database, which are the entire question and the answers below. Then behind the question, there is an option to view the comments. Comments part is only changeable when the comment is added, or deleted.

3.2.5.5.1. Interface description

Main navigation panel and question/answer part are same for all process. Comments interface has the content of the comments respectively below the question.

3.2.5.5.2. Algorithmic model (e.g., PDL)

For comment viewing, system takes the comments of the question from database. Page is changeable depending on other process (adding/deleting comments).

3.2.5.5.3. Restrictions/limitations

There is no restriction or limitation since this page is available within the user's session, which is the restriction of the previous level. However, even though the question might not be accessed through the favorite tags list, it is still accessible from the ordinary questions page.

3.2.5.5.4. Local data structures

User data structure is used overall the process. For listing comments, comment data structure is used.

3.2.5.5.5. Performance issues

Size of comments list is the only important part regarding the performance.

3.2.5.5.6. Design constraints

The HTML/AJAX/JavaScript components should be able to run cross-platform, i.e. on different browsers, correctly.

3.2.6. Description for Question Answer CRUD Component

3.2.6.1. Processing narrative (PSPEC) for Question Answer CRUD component

In this page, users can process edit, add and delete answer processes. Users see answer list for a question and they add new answer to list for this question. In addition, if they have an answer for current question, they edit own answer or completely delete it. After users finish process, answer list in this page is again listed. Users don't edit others' answer, they only edit own answer, others' answer just are seen in answer list. But users can vote answers negatively or positively. Answer list is determined according to height of answer votes

3.2.6.2. Question Answer CRUD component interface description.

This page has answer list for question. In this list, there are answers list, add new answer button, edit answer button, and delete answer button and main navigation panel. In answer list, users see answers text, user name of adding answers (responders) and votes of answers. Second interface is adding new answer for question. Users click adding button and go to create new answer area then answer is created and added to list by system. Third interface is

deleting own answer for question. Users click deleting button and answer is deleted by system then answer is again show. Fourth interface is editing own answer for question. Users click editing button and answer is edited by system then answer list is again showed.

3.2.6.3. Sub-Component List Answer of Question Answer CRUD component processing detail

When answer listing page is called from another page, the page is loaded with the items requested from the database. Answers listing shows to users' answers for questions. This list is changeable, because users add new answers or deleting own answers or editing own answers or voting other answers. . Another process is voting for this component. Users is vote answers according to quality of answer.

3.2.6.3.1. Interface description

Main navigation panel is same for all process. Listing interface has answer text, user name and vote.

3.2.6.3.2. Algorithmic model (e.g., PDL)

For listing, system is takes answers from database for current question and lists in table with answer text, user name and vote. List is changeable depending on other process (adding, editing and deleting answer). For voting, system takes voting command from user for anyone answer. Then answer vote is changed in database by system.

3.2.6.3.3. Restrictions/limitations

A user can see all answer from answer list. Therefore there are no restrictions or limitation on this component.

3.2.6.3.4. Local data structures

An answer class will be implemented, containing the field id, name and description along with getters and setters of those fields.

3.2.6.3.5. Performance issues

Size of answers list is important arbiter for system running time performance.

3.2.6.3.6. Design constraints

The HTML/AJAX/JavaScript components should be able to run cross-platform, i.e. on different browsers, correctly.

3.2.6.4. Sub-Component Add Answer of Question Answer CRUD component processing detail

When tag adding page is called from this page, the page is loaded with the items requested from the database. Adding answer process adds to answers list for current question. Users click add button then open a pop-up, users enter their answer label that is in pop-up page, finally confirm own answer and send. Then answer is adding to database and page is redirected to last page.

3.2.6.4.1. Interface description

Main navigation panel is same for all process. But add process has different pop-up panel.

3.2.6.4.2. Algorithmic model (e.g., PDL)

For adding, system takes new answer from user in add pop-up. Then it is added to database fairly by system.

3.2.6.4.3. Restrictions/limitations

A user can add answer to any question. Therefore there are no restrictions or limitation on this component.

3.2.6.4.4. Local data structures

For adding, using answer data structure

3.2.6.4.5. Performance issues

Size of answers list is important arbiter for system running time performance.

3.2.6.4.6. Design constraints

The HTML/AJAX/JavaScript components should be able to run cross-platform, i.e. on different browsers, correctly.

3.2.6.5. Sub-Component Edit Answer of Question Answer CRUD component processing detail

When tag adding page is called from this page, the page is loaded with the items requested from the database. Editing answer process provide editing to users for own answer in answer list. Users click edit button then open a pop-up, users edit their answer that is in label of pop-up page, finally confirm own new answer and send.

3.2.6.5.1. Interface description

Main navigation panel is same for all process. But edit process has different pop-up panel.

3.2.6.5.2. Algorithmic model (e.g., PDL)

For editing, system takes editing command from user for user' anyone answer. Then this answer is redirected to editing pop-up. System takes editing answer from user and this change is implemented in database.

3.2.6.5.3. Restrictions/limitations

A user can only edit her own answers. Therefore there are no restrictions or limitation on this component.

3.2.6.5.4. Local data structures

For editing, using answer data structure.

3.2.6.5.5. Performance issues

Size of answers list is important arbiter for system running time performance.

3.2.6.5.6. Design constraints

The HTML/AJAX/JavaScript components should be able to run cross-platform, i.e. on different browsers, correctly.

3.2.6.6. *Sub-Component Answer Delete of Question Answer CRUD component processing detail*

When tag adding page is called from this page, the page is loaded with the items requested from the database. Deleting answer process provide deleting to users for own answer in answer list. Users click delete button and current answer is deleted by system.

3.2.6.6.1. Interface description

Main navigation panel is same for all process.

3.2.6.6.2. Algorithmic model (e.g., PDL)

For deleting, system takes deleting command from user for user' anyone answer. Then this answer is deleting database.

3.2.6.6.3. Restrictions/limitations

A user can only delete her own answers. Therefore there are no restrictions or limitation on this component.

3.2.6.6.4. Local data structures

For deleting, using answer data structure.

3.2.6.6.5. Performance issues

Size of answers list is important arbiter for system running time performance.

3.2.6.6.6. Design constraints

The HTML/AJAX/JavaScript components should be able to run cross-platform, i.e. on different browsers, correctly.

3.2.7. Description for Question Comment CRUD

3.2.7.1. *Processing narrative (PSPEC) for Question Comment CRUD component*

In this page, users can process edit, add and delete comment processes. Users see comment list for a question and they add new comment to list for

this question. In addition, if they have a comment for current question, they edit own comment or completely delete it. After users finish process, comment list in this page is again listed. Users don't edit others' comment, they only edit own comment, others' comment just are seen in comment list.

3.2.7.2. Question Comment CRUD component interface description.

This page has comment list for question. In this list, there are comment list, add new comment button, edit comment button, and delete comment button and main navigation panel. In comment list, users see comments text and user name of adding comment (responders). Second interface is adding new comment for question. Users click adding button and go to create new comment area then comment is created and added to list by system. Third interface is deleting own comment for question. Users click deleting button and comment is deleted by system then comment list is again show. Fourth interface is editing own comment for question. Users click editing button and comment is edited by system then comment list is again show.

3.2.7.3. Sub-Component Comment List of Question Comment CRUD processing detail

When comment listing page is called from another page, the page is loaded with the items requested from the database. Comments listing shows to users' comments for questions. This list is changeable, because users add new comments or deleting own comments or editing own comments or voting other comments.

3.2.7.3.1. Interface description

Main navigation panel is same for all process. Listing interface has comment text, user name and vote.

3.2.7.3.2. Algorithmic model (e.g., PDL)

For listing, system is takes comments from database for current question and lists in table with comment text, user name and vote. List is changeable depending on other process (adding, editing and deleting comments).

3.2.7.3.3. Restrictions/limitations

A user can see all comment from comments list. Therefore there are no restrictions or limitation on this component.

3.2.7.3.4. Local data structures

For listing, using user and comment data structure.

3.2.7.3.5. Performance issues

Size of comments list is important arbiter for system running time performance.

3.2.7.3.6. Design constraints

The HTML/AJAX/JavaScript components should be able to run cross-platform, i.e. on different browsers, correctly.

3.2.7.4. Sub-Component Comment Add of Question Comment CRUD processing detail

When tag adding page is called from this page, the page is loaded with the items requested from the database. Adding comment process adds to comments list for current question. Users click add button then open a pop-up, users enter their comment label that is in pop-up page, finally confirm own comment and send. Then comment is added to database and page is redirected to last page.

3.2.7.4.1. Interface description

Main navigation panel is same for all process. But add process has different pop-up panel.

3.2.7.4.2. Algorithmic model (e.g., PDL)

For adding, system takes new comment from user in add pop-up. Then it is added to database fairly by system.

3.2.7.4.3. Restrictions/limitations

Any user can add comments to any question. Therefore there are no restrictions or limitation on this component.

3.2.7.4.4. Local data structures

For adding, using comment data structure

3.2.7.4.5. Performance issues

Size of comments list is important arbiter for system running time performance.

3.2.7.4.6. Design constraints

The HTML/AJAX/JavaScript components should be able to run cross-platform, i.e. on different browsers, correctly.

3.2.7.5. Sub-Component Comment Edit of Question Comment CRUD processing detail

When tag adding page is called from this page, the page is loaded with the items requested from the database. Editing comment process provide editing to users for own comment in comment list. Users click edit button then open a pop-up, users edit their comment that is in label of pop-up page, finally confirm own new comment and send.

3.2.7.5.1. Interface description

Main navigation panel is same for all process. But edit process has different pop-up panel.

3.2.7.5.2. Algorithmic model (e.g., PDL)

For editing, system takes editing command from user for user' anyone comment. Then this comment is redirected to editing pop-up. System takes editing comment from user and this change is implemented in database.

3.2.7.5.3. Restrictions/limitations

A user only edits own comments.

3.2.7.5.4. Local data structures

For editing, using comment data structure.

3.2.7.5.5. Performance issues

Size of comments list is important arbiter for system running time performance.

3.2.7.5.6. Design constraints

The HTML/AJAX/JavaScript components should be able to run cross-platform, i.e. on different browsers, correctly.

3.2.7.6. *Sub-Component Comment Delete of Question Comment CRUD processing detail*

When tag adding page is called from this page, the page is loaded with the items requested from the database. Deleting comment process provide deleting to users for own comment in answer list. Users click delete button and current comment is deleted by system.

3.2.7.6.1. Interface description

Main navigation panel is same for all process.

3.2.7.6.2. Algorithmic model (e.g., PDL)

For deleting, system takes deleting command from user for user' anyone comment. Then this comment is deleting database.

3.2.7.6.3. Restrictions/limitations

A user only deletes own comments.

3.2.7.6.4. Local data structures

For deleting, using comment data structure.

3.2.7.6.5. Performance issues

Size of comments list is important arbiter for system running time performance.

3.2.7.6.6. Design constraints

The HTML/AJAX/JavaScript components should be able to run cross-platform, i.e. on different browsers, correctly.

3.2.8. Description for Question Delete component

Using this component an admin may delete questions which are unnecessary or violate web page policies.

3.2.8.1. Processing narrative (PSPEC) for Question Delete component

Delete page is an empty page and there is no direct access to it. Only admin can see a button on Question View page to delete questions. The questions which violate the rules should be deleted to offer better experience for users. When delete function is called, admin is redirected to delete page, the question is deleted and admin is, again, redirected to the Question View page. Only admins can see the deleted questions. The operation will be soft-delete and the questions will not really be deleted from database for further control mechanisms.

3.2.8.2. Question Delete Component interface description

Delete page will have no inputs or outputs. It will call related deleted operation on the server side and after a successful deletion, admins will be redirected to the page where they came from.

3.2.8.3. Sub-Component of Question Delete component processing detail

Since delete pages have no inputs or outputs, there is no sub-component present.

3.2.9. Description for User Delete Component

Using this component an admin may delete users which violate web page policies. Users can not delete themselves.

3.2.9.1. Processing narrative (PSPEC) for User Delete component

Delete page is an empty page and there is no direct access to it. Only admin can see a button on User Profile page to delete users. The users which violate the rules should be banned to offer better experience for other users. When delete function is called, admin is redirected to delete page, the user is deleted and admin is, again, redirected to the listing page. Only admins can see the deleted users. The operation will be soft-delete and the users will not really be deleted from database for further control mechanisms.

3.2.9.2. User Delete Component interface description

Delete page will have no inputs or outputs. It will call related deleted operation on the server side and after a successful deletion, admins will be redirected to the page where they came from.

3.2.9.3. Sub-Component of User Delete component processing detail

Since delete pages have no inputs or outputs, there is no sub-component present.

3.2.10. Description for User Login Component

User login component is responsible for user login operations. One can access the login module by clicking the log in button located in the header of the web page. Upon click, user will be prompted for e-mail address and password. If the user has entered correct information, she will be redirected to question listing page. The header component will change to reflect the user details as well.

3.2.10.1. Processing narrative (PSPEC) for User Login component

User login component will take 2 inputs from the user: username (e-mail address) and the password. When the user presses login button, the inputs will be sent to database to check if they are correct inputs. If not, a login error screen will be displayed and user will be prompted for username and password again. If the input user entered is correct, a successful login will be initiated. Since this is a simple module, there are no sub components.

3.2.10.2. User Login component interface description

User login component is a simple HTML form which has two labels, two input fields and a submit button. Upon inconsistent input submission, the page displaying the error message will contain an HTML div element below the form for displaying the error message.

3.2.11. Description for User Profile CRUD Component

This module is the part of web application in which the users can see and edit their profile information such as their contact, biography, location and website. Users also can see their site status information such as previous answers, badges, tags and reputations.

3.2.11.1. Processing narrative (PSPEC) for user profile CRUD component

In this component users can see and edit their profile information such as their contact, biography, location and website can be edited and deleted with buttons associated with them. When edit function is called there will be appear an edit box for every item. Users also can see their site status information such as previous answers, badges, tags and reputations. In user profile page users can list questions, answers, tags and they can reach these items via their links.

3.2.11.2. User profile CRUD component interface description.

User profile CRUD page will be built made on a page which users can see and edit their profile informations like nickname, bio and location. When the edit button is pressed user can edit these informations and the delete button is pressed user can delete this informations.

3.2.11.3. *Sub-Component user profile view module of user profile detail*

When user profile page is called from another page, the page is loaded with the items requested from database.

3.2.11.3.1. *Interface description*

Profile page will be used to show the users informations returned from the database. The page will have username, user status, and user bio and user location at the beginning of the page. User's answers and users questions are shown in the middle of the page. Users every questions and answer has a vote counter on next to them. At the end of the page users tags are will be represented.

3.2.11.3.2. *Restrictions/limitations*

All users and admins can see other users profile pages but only the users itself can have the right of edit or delete their profile informations on profile page.

3.2.11.3.3. *Local data structures*

General user class, general question class and general answer class will be used in this page. But not the all methods of these classes are required. User class's setter methods can be used in only in this page. Question and answer class's only getter methods are used in user profile page.

3.2.11.3.4. *Performance issues*

This page is not a critical component of the web site so there are no issues that may affect the run-time performance, security, or computational accuracy of this sub-component unless the database returns the result of a query in a feasible amount of time.

3.2.11.3.5. *Design constraints*

The HTML/AJAX/JavaScript components should be able to run cross-platform and displayed same on different operating systems on different browsers, correctly.

3.2.11.4. *Sub-Component user profile create module of user profile detail*

When user profile create component is called from page, the page is loaded with the empty items for creating new page from database.

3.2.11.4.1. *Interface description*

Profile page will be used to show the users informations returned from the database. The page will have username, user status, and user bio and user location at the beginning of the page. User will be able to add profile informations like website, bio, and locations. User also can add tags to their profile page. Tags will be prepared by admins. Users can not create tags, but they can add tags to their profile page

3.2.11.4.2. Restrictions/limitations

All users and admins can see other users profile pages but only the users itself can have the right of add item to their profile informations on profile page.

3.2.11.4.3. Local data structures

General user class, general question class and general answer class will be used in this page. But not all methods of these classes are required. User class's setter methods can be used in only in this page to add item on user profile. Other data structures are only use for taking informations

3.2.11.4.4. Performance issues

This module is not a critical component of the web site nonetheless this module should be use very rarely so there are no issues that may affect the run-time performance, security, or computational accuracy of this sub-component unless the database returns the result of a query in a feasible amount of time.

3.2.11.4.5. Design constraints

There are no design constraints for this module since these operations are mostly database operations which are simple to implement.

3.2.11.5. *Sub-Component user profile update module of user profile detail*

When user profile update component is called from page, the page is loaded with the contents filled with the values of the item being edited along with its labels by requesting from database.

3.2.11.5.1. Interface description

Profile page will be used to show the users informations returned from the database. The page will have username, user status, and user bio and user location at the beginning of the page. User will be able to update or edit profile informations of themselves like website, bio, and locations. User also can edit tags to their profile page. Tags will be prepared by admins. Users can not create new tags, but they can add tags to their profile page

3.2.11.5.2. Restrictions/limitations

All users and admins can see other users profile pages but only the users itself can have the right of edit items of their profile informations on their profile page.

3.2.11.5.3. Local data structures

General user class, general question class and general answer class will be used in this page. But not all methods of these classes are required. User

class's setter methods can be used in only in this page to add item on user profile. Other data structures are only use for taking informations

3.2.11.5.4. Performance issues

This module is not a critical component of the web site nonetheless this module probably will be use very rarely so there are no issues that may affect the run-time performance, security, or computational accuracy of this sub-component unless the database returns the result of a query in a feasible amount of time.

3.2.11.5.5. Design constraints

The page has HTML and JavaScript components in it so it should run cross-platform on different browsers, correctly.

3.2.11.6. *Sub-Component user profile delete module of user profile detail*

When user profile delete component is called, the user is directed their profile page and the item which user selects to delete will be deleted.

3.2.11.6.1. Interface description

There is no interface for this page. The page will return to user profile page as soon as the delete query is sent to the database.

3.2.11.6.2. Restrictions/limitations

All users and admins can see other users profile pages but only the users itself can have the right of delete items of their profile informations on their profile page.

3.2.11.6.3. Local data structures

General user class, general question class and general answer class will be used in this page. But not all methods of these classes are required. User class's setter methods can be used in only in this page to add item on user profile.

3.2.11.6.4. Performance issues

This module is not a critical component of the web site nonetheless this module probably will be use very rarely so there are no issues that may affect the run-time performance, security, or computational accuracy of this sub-component unless the database returns the result of a query in a feasible amount of time.

3.2.11.6.5. Design constraints

There are no design constraints for this module since these operations are mostly database operations which are simple to implement.

3.2.12. Description for Questions Page

This module is the part of web application in which users can list question's titles, tags and question's text itself but not the answers of questions. Also every question has vote number, answers number, view number and date on the near of question title. Users can access to specific question page to click on "List answers button".

3.2.12.1. *Processing narrative (PSPEC) questions CRUD component*

In this module, users can list question's titles, tags and question's text itself but not the answers of questions. Also every question has vote number, answers number, view number and date on the near of question title. Users can access to specific question page to click on "List answers button".

3.2.12.2. *Questions CRUD component interface description*

Questions page will be built on base page in which there is a list for questions and each question contains title, tag and link to the question's page itself. Users will be able to briefly view common questions and they can go the questions itself then they can answer or only view the questions itself.

3.2.12.3. *Sub-Component questions view module of questions detail*

When questions list page is called from another page, the page is loaded with the items requested from database.

3.2.12.3.1. *Interface description*

Questions page will be used to show mainly questions and their votes, answer and view numbers next to the question titles. Under the each questions title there will be a tag list which are related tags about questions.

3.2.12.3.2. *Restrictions/limitations*

There are no privilege-based restrictions or limitations since listing of the tags is a public operation.

3.2.12.3.3. *Local data structures*

General question class will be used in this page. But not the all methods of these classes are required. Question class's only getter methods are used in this page.

3.2.12.3.4. *Performance issues*

This page is one of the critical components of the web site so these pages must be run a feasible amount of time and accurate. This page is viewed public and this component has not edit or update operation so there is no security concern.

3.2.12.3.5. *Design constraints*

The HTML/AJAX/JavaScript components should be able to run cross-platform operation systems on different browsers, correctly.

3.2.12.4. Sub-Component questions create module of questions detail

When question create component is called from another page, the page is loaded with the items requested from database.

3.2.12.4.1. Interface description

Questions page will be used to show mainly questions and their votes, answer and view numbers next to the question titles. User can ask questions by the button on right upper corner of the page.

3.2.12.4.2. Restrictions/limitations

To add a question user must be register to site so public user cannot ask question. Only users and admins can ask questions.

3.2.12.4.3. Local data structures

General question class will be used in this page. But not the all methods of these classes are required. Question class's only setter methods are used in this page.

3.2.12.4.4. Performance issues

This page is one of the critical components of the web site so these pages must be run a feasible amount of time and accurate. This page is viewed public and this component has not edit or update operation so there is no security concern.

3.2.12.4.5. Design constraints

There are no design constraints for this module since these operations are mostly database operations which are simple to implement.

3.2.13. Description for Question CRUD Module

This module is the part of web application in which users can see mainly question's titles, tags and question's text itself, answers and comments. Also question and answers has vote number and date on the near of their title. Users can answer the question on this page by clicking on "Answer This Question Button".

3.2.13.1. Processing narrative (PSPEC) question view CRUD component

In this page, users can see question's titles, tags and question's text itself. Also every question has vote number, answers number, view number and date on the near of question title.

3.2.13.2. Question view component interface description

Question view page will be built on base page in which there is question and a list of answer of questions. Users will be able to briefly view question and answers.

3.2.13.3. *Sub-Component view module of question view detail*

When question view page is called from another page, the page is loaded with the items requested from database.

3.2.13.3.1. *Interface description*

Question view page will be used to show one question and its answers, answers are listed under the question. Every answer has a vote field. Under the each questions title there will be a tag list which are related tags about questions.

3.2.13.3.2. *Restrictions/limitations*

Everyone can see question and list of the answers so this is a public operation. There are no view restrictions on this module

3.2.13.3.3. *Local data structures*

General question and answer classes will be used in this page. But not the all methods of these classes are required only getter methods are used in this page.

3.2.13.3.4. *Performance issues*

This page is one of the critical components of the web site so these pages must be run a feasible amount of time and accurate. This page is viewed public and this component has not edit or update operation so there is no security concern.

3.2.13.3.5. *Design constraints*

The HTML components should be able to run cross-platform operation systems on different browsers, correctly.

3.2.13.4. *Sub-Component question create module of question view detail*

When user profile page is called from another page, the page is loaded with the items requested from database.

3.2.13.4.1. *Interface description*

Question view page will be used to show question and their answer and vote numbers next to them. User can answer questions by the help of “Answer Question” button on right lower corner of the question.

3.2.13.4.2. *Restrictions/limitations*

To answer a question user must be register to site so public user cannot answer question. Only users and admins can answer questions.

3.2.13.4.3. *Local data structures*

General question class will be used in this page. But not the all methods of these classes are required. Question class's getter methods are used in this page. Also Objects of answer classes will be generated on this page by adding answers.

3.2.13.4.4. Performance issues

This page is one of the critical components of the web site so these pages must be run a feasible amount of time and accurate.

3.2.13.4.5. Design constraints

The HTML/Javascript components should be able to run cross-platform operation systems on different browsers, correctly.

3.2.13.5. *Sub-Component update module of question view detail*

When question view update page is called from another page, , the page is loaded with the contents filled with the values of the item being edited along with its labels by requesting from database.

3.2.13.5.1. Interface description

Question view page will be used to update question by user who asked the question and update answers by user who answered the answers. Other user cannot edit any item on this page.

3.2.13.5.2. Restrictions/limitations

To update a question or answer user must be register to site and content must be owned by these users so public user cannot update anything on thing site.

3.2.13.5.3. Local data structures

General question class will be used in this page. But not the all methods of these classes are required. Question class's getter methods are used in this page.

3.2.13.5.4. Performance issues

This module is one average important components of the web site so these pages must be work accurate.

3.2.13.5.5. Design constraints

Question update module must preserve previous changes in the update question view page. There are no other design constraints for this module since these operations are mostly database operations which are simple to implement.

3.2.13.6. *Sub-Component delete module of question view detail*

When question delete module is called from question view page, the question and all answers of this question is deleted permanently. And then the user is directed to main page.

3.2.13.6.1. Interface description

There is no interface for this page. The page will return to main page as soon as the delete query is sent to the database.

3.2.13.6.2. Restrictions/limitations

Only the owner of the answer or owner of the question can delete their contents.

3.2.13.6.3. Local data structures

General question or answer class will be used in this page with setters of these classes' methods.

3.2.13.6.4. Performance issues

This module is one average important components of the web site so these pages must be work accurate.

3.2.13.6.5. Design constraints

There are no design constraints for this module since these operations are mostly database operations which are simple to implement.

3.3. Software Interface Description

Since this is a webpage application, the whole system is human-interaction-based. The visuals for all of the interfaces (pages) can be found on section 4. Other machines trying to browse the contents of the page may use any XML parser to extract data.

4. User interface design

A description of the user interface design of the software is presented in this section.

4.1. Description of the user interface

User interface component descriptions are listed below.

4.1.1. Tag CRUD Component

As described in section 3, there are four discrete pages for user to interact with the CRUD component. The Tag List page will list all of the tags. The table in the page will contain operations, id, name and information headers. In operations column, each item will have two possible operation buttons, edit and delete. Upon clicking these buttons, user will be redirected to the relevant pages. The listing page will have a separate button in the left column of the page for adding new tags, which upon clicking will redirect user to the Add Tag Page. Delete Tag page is an empty page, containing no HTML elements, which redirects user to listing page when deletion operation is requested from the database. Add and edit pages have exactly the same components. Labels for identifying the field names and textboxes for user data, and a save button.


4.1.1.1. Screen images

Q&A Web Page

← → × 🏠 🔍

Q&A Web Page
You have Questions, we have Answers

[Admin](#) > Add Tag



Add Tag

* required

Name: *

Information Text: *

Q&A Web Page

← → × 🏠

http://www.questionsandanswers.com

🔍

Q&A Web Page

You have Questions, we have Answers

Admin > Tag List

QUESTIONS QUESTIONS

& &

ANSWERS ANSWERS

İşlemler

⊕ Add Tag

Tag List

Action	ID	Name	Information Text
<div>☒</div>	1	Machine Learning	Machine learning provides computer..
<div>☒</div>	2	Artificial Intelligence	Artificial intelligence (AI) is the branch..
<div>☒</div>	3	C#	C# is a multi-paradigm programming..
<div>☒</div>	4	Asp.Net	ASP.NET is a web application framework..
<div>☒</div>	5	Twitter4j	Twitter4J is a library enabling calls..

Q&A Web Page

← → × 🏠

http://www.questionsandanswers.com

🔍

Q&A Web Page

You have Questions, we have Answers

Admin > Edit Tag: Machine Learning

QUESTIONS QUESTIONS

& &

ANSWERS ANSWERS

Edit Tag: Machine Learning

* required

Name: *

Machine Learning

Information Text: *

Machine learning provides computer algorithms that automatically discover patterns in data and make intelligent decisions from them.

Save

Cancel

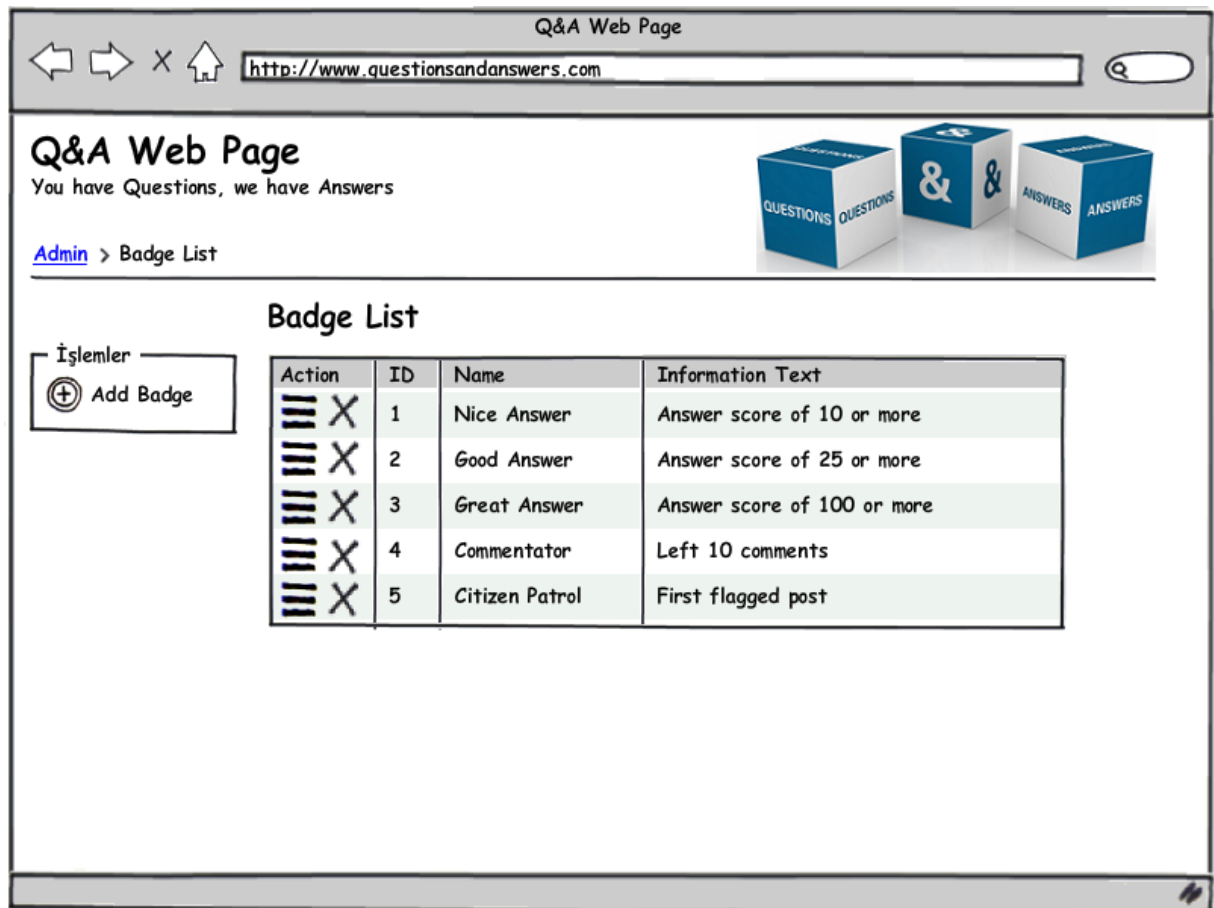
4.1.1.2. Objects and actions

Tag listing page will contain a table element for listing the items. Each table item will have two buttons representing possible operations which redirects user to the relevant pages when clicked. There is also a separate button for adding new tags to the database which takes user to add tag page. Delete tag is an empty page. Add and edit pages contain same elements. A form containing labels and textboxes and a button resides in those pages. Save button creates the new tag or completes the edit on a chosen tag and then redirects user to tag listing page.

4.1.2. Badge CRUD Component

As described in section 3, there are four discrete pages for user to interact with the CRUD component.

4.1.2.1. Screen images



The Badge List page will list all of the badges. The table in the page will contain operations, id, name and information text. In actions column, each item will have two possible action buttons, edit and delete. Upon clicking these buttons, user will be redirected to the relevant pages. The listing page will have a separate button in the left column of the page for adding new badges, which upon clicking will redirect user to the Add Badge Page. Delete Badge page is an empty page, containing no HTML elements, which redirects user to listing page when deletion operation is requested from the database.

Q&A Web Page

X

http://www.questionsandanswers.com

Q&A Web Page

You have Questions, we have Answers

QUESTIONS

QUESTIONS

&

&

ANSWERS

ANSWERS

[Admin](#) > Add Badge

Add Badge

* required

Name: *

Information Text: *

Image Path: *

Save

Cancel

Add and edit pages have exactly the same components. Labels for identifying the field names and textboxes for user data, a save and a cancel button.

Q&A Web Page

You have Questions, we have Answers

[Admin](#) > Edit Badge: Nice Answer

Edit Badge: Nice Answer

* required

Name: *

Information Text: *

Image Path: *

4.1.2.2. Objects and actions

Badge listing page will contain a table element for listing the items. Each table item will have two buttons representing possible actions which redirects user to the relevant pages when clicked. There is also a separate button for adding new badges to the database which takes user to add badge page. Delete badge is an empty page. Add and edit pages contain same elements. A form containing labels and textboxes and a button resides in those pages. Save button creates the new badge or completes the edit on a chosen badge and then redirects user to badge listing page.

4.1.3. Question Delete Component

Since delete pages have no inputs or outputs, there is no user interfaces specific for them. Question Delete component will be triggered by a button which is visible to admins on the Question View page. After the delete operation is completed, admin will be redirected to Question View page again.

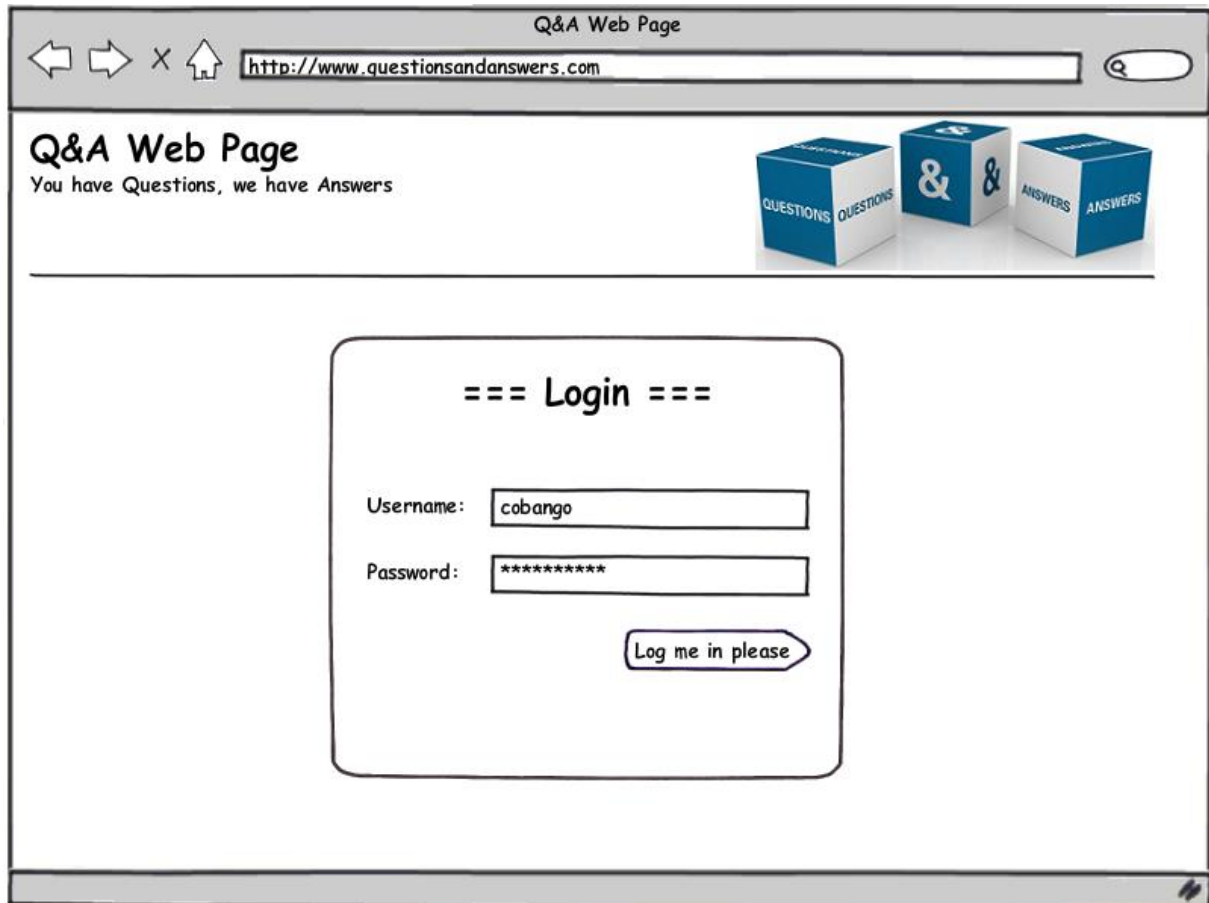
4.1.4. User Delete Component

Since delete pages have no inputs or outputs, there is no user interfaces specific for them. User Delete component will be triggered by a button which is visible to admins on the User Profile page. After the delete operation is completed, admin will be redirected to User Profile page again.

4.1.5. Login Component

Login page is a simple component which contains only a form with 2 labels, 2 textboxes and one button.

4.1.5.1. Screen images



The screenshot shows a web browser window titled "Q&A Web Page". The address bar displays "http://www.questionsandanswers.com". The page header includes the text "Q&A Web Page" and "You have Questions, we have Answers", along with a graphic of three blue cubes labeled "QUESTIONS", "&", and "ANSWERS". The main content area features a login form titled "=== Login ===". The form contains two labels: "Username:" and "Password:". The "Username:" label is followed by a text input field containing the text "cobango". The "Password:" label is followed by a password input field containing ten asterisks "*****". Below the password field is a button labeled "Log me in please".

4.1.5.2. Objects and actions

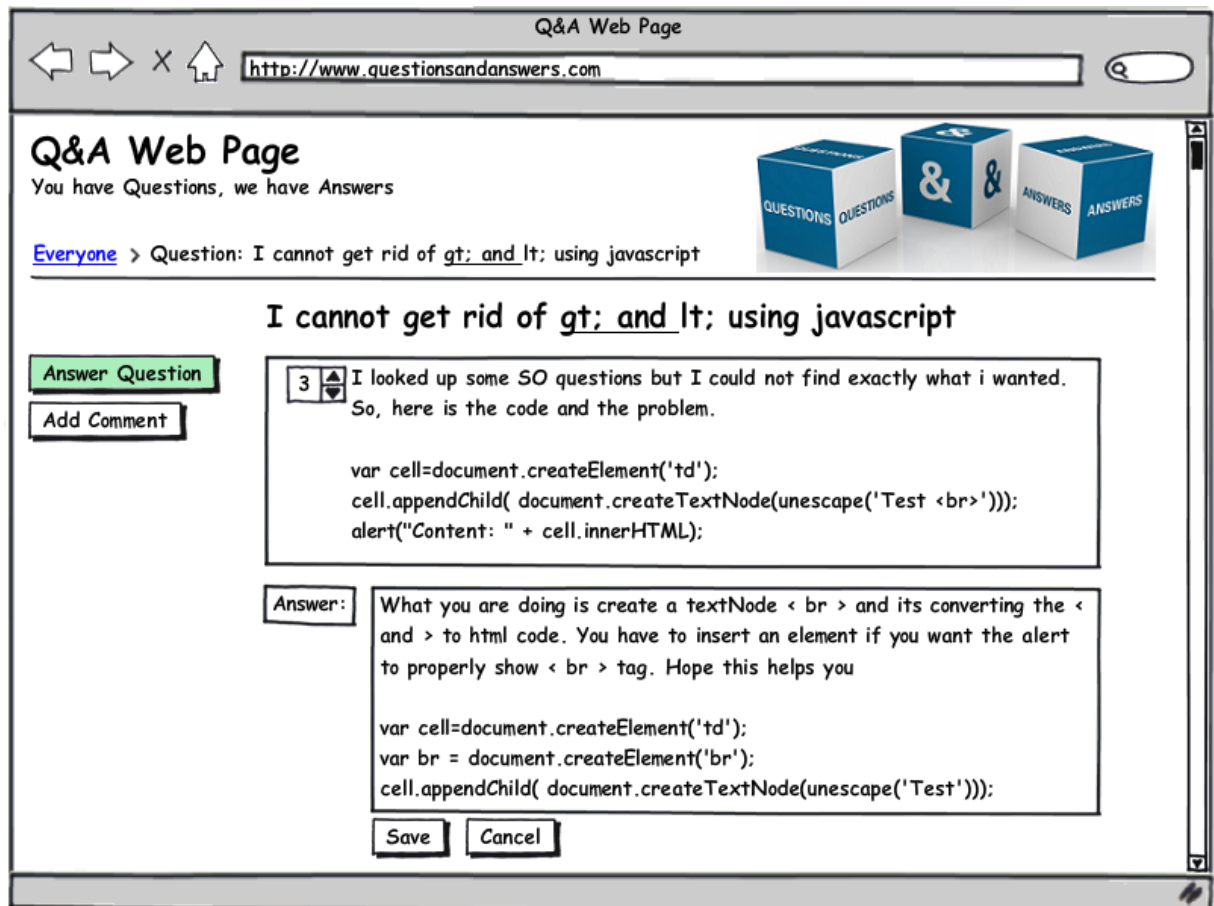
Username textbox is a plain-text textbox while password textbox hides what is typed in the textbox. Login button initiates the login operation.

4.1.6. Question Answer CRUD

This model has five main process. Firstly, answers listing shows to users' answers for questions. This list is changeable, because users add new answers or deleting own answers or editing own answers or voting other answers. Second process is adding answer to answers list for question. Users click add button then open a pop-up, users enter their answer label that is in pop-up page, finally confirm own answer and send. Then answer is adding to database and page is redirected to last page. Answers list is refreshed and new answer added by system. Third process is editing answer for own answer in answer list. Users click edit button then open a pop-up, users edit their answer that is

in label of pop-up page, finally confirm own new answer and send. Then answer is editing in database and page is redirected to last page. Answers list is refreshed and editing answer is showed by system. Forth process that is listing to answers, is come round for each other process by system. Fifth process is voting. Users is vote answers according to quality of answer.

4.1.6.1. Screen images



4.1.6.2. Objects and actions

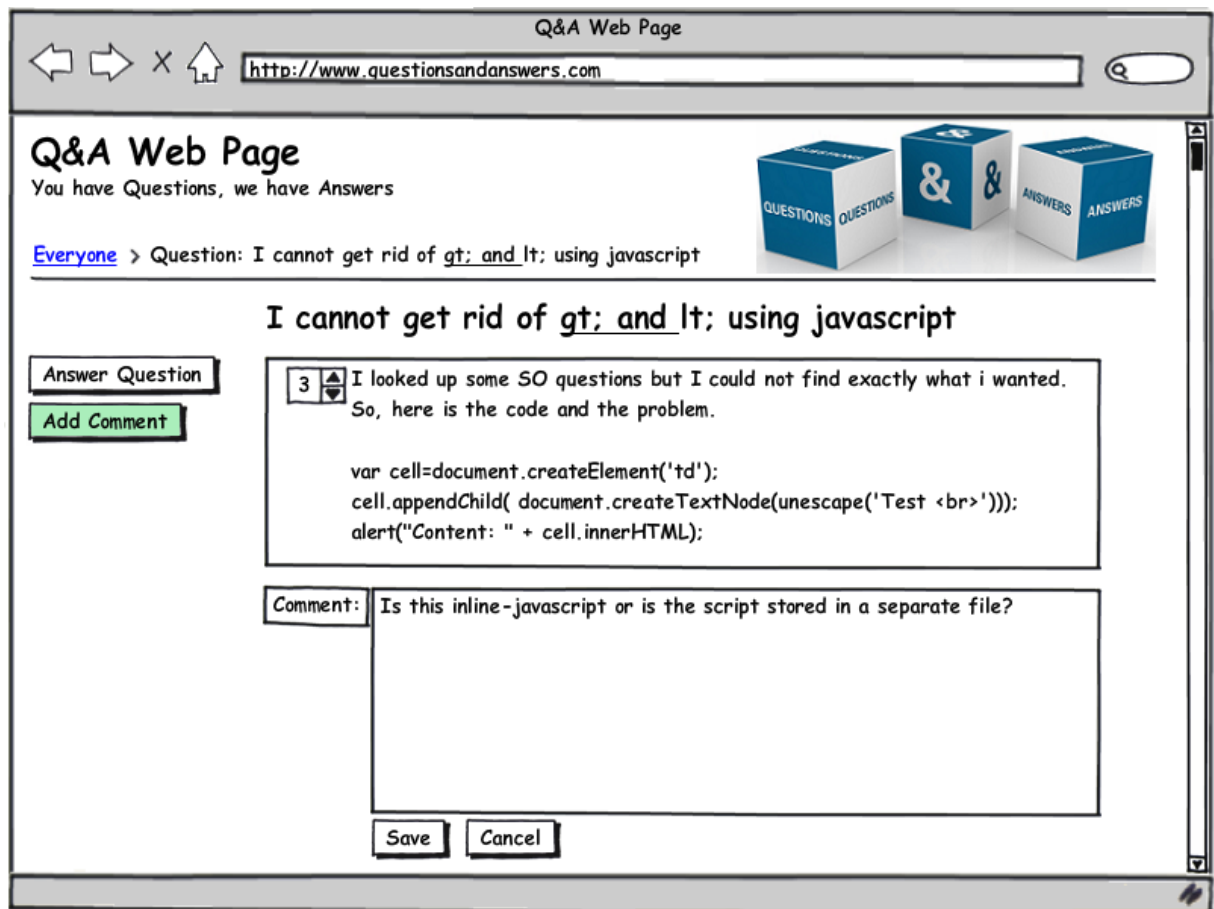
For listing, system is takes answers from database for current question and lists in table with answer text, user name and vote. For adding, system takes new answer from user in add pop-up. Then it is added to database fairly by system. For deleting, system takes deleting command from user for user' anyone answer. Then this answer is deleting database. For editing, system takes editing command from user for user' anyone answer. Then this answer is redirected to editing pop-up. System takes editing answer from user and this change is implemented in database. For voting, system takes voting command from user for anyone answer. Then answer vote is changed in database by system.

4.1.7. Question Comment CRUD

This model has four main process. Firstly, comment listing shows to users' comment for questions. This list is changeable, because users add new comment or deleting own comment or editing own comment. Second process is adding comment to comment list for question. Users click add button then

open a pop-up, users enter their comment label that is in pop-up page, finally confirm own comment and send. Then comment is adding to database and page is redirected to last page. Comment list is refreshed and new comment added by system. Third process is editing comment for own comment in comment list. Users click edit button then open a pop-up, users edit their comment that is in label of pop-up page, finally confirm own new comment and send. Then comment is editing in database and page is redirected to last page. Comment list is refreshed and editing comment is showed by system. Forth process that is listing to comment, is come round for each other process by system.

4.1.7.1. Screen images



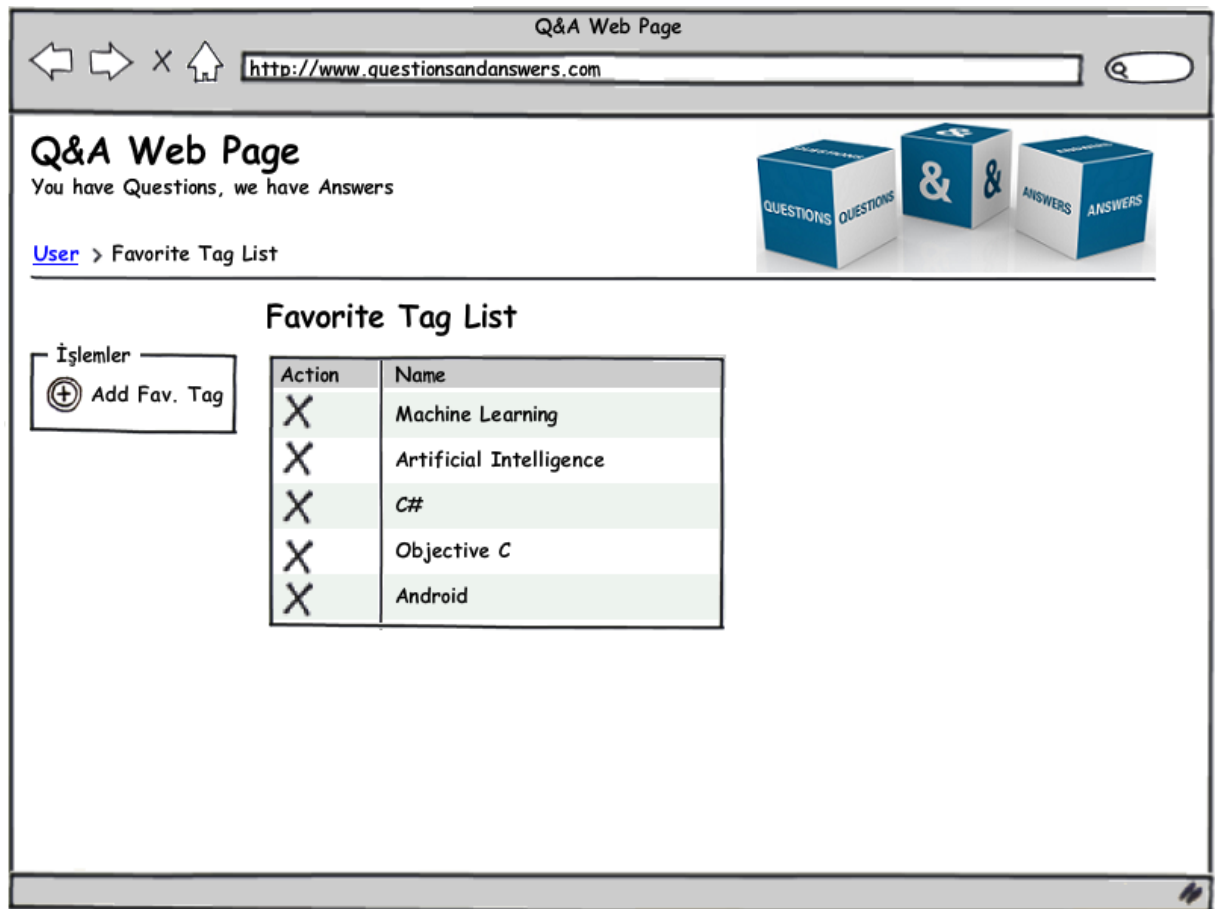
4.1.7.2. Objects and actions

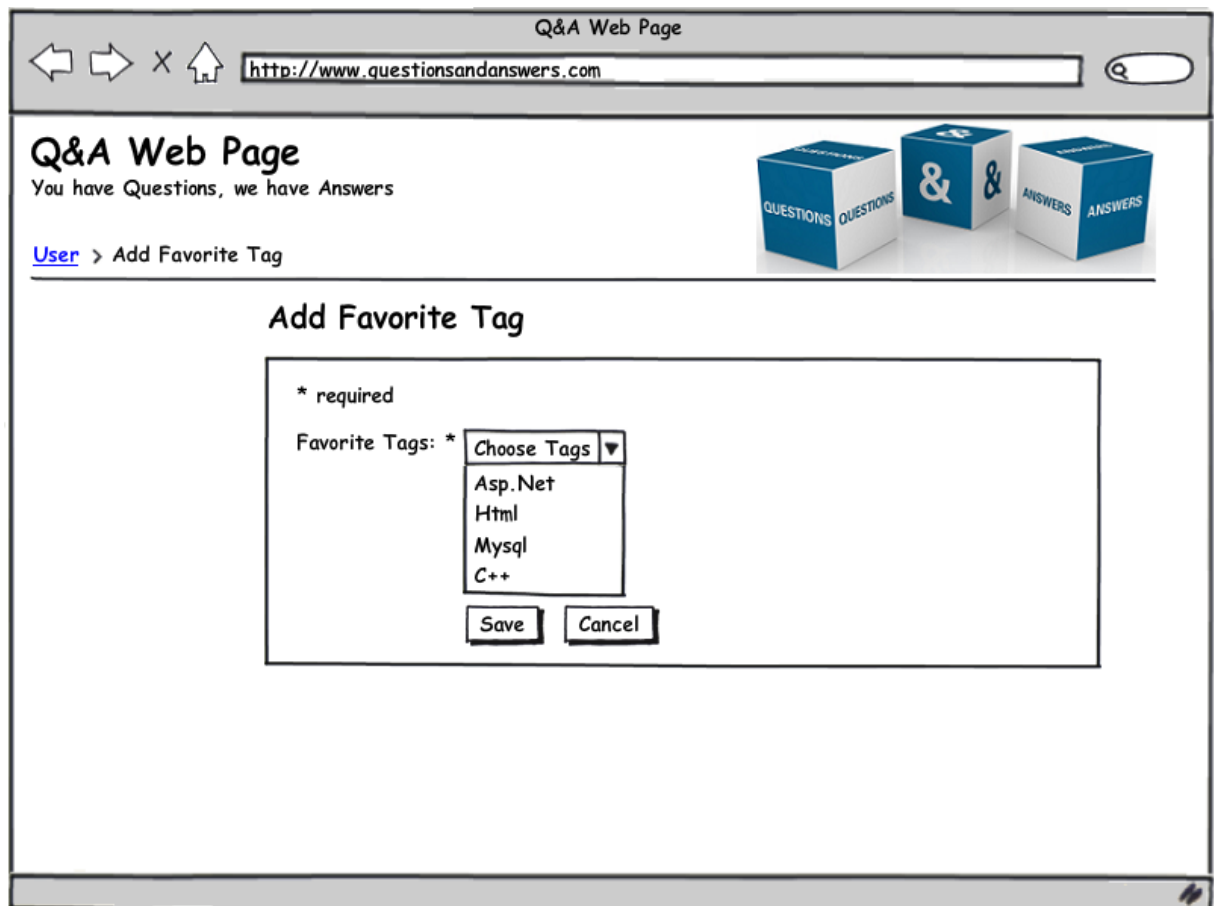
For listing, system is takes comments from database for current question and lists in table with comment text, user name and vote. For adding, system takes new comment from user in add pop-up. Then it is added to database fairly by system. For deleting, system takes deleting command from user for user' anyone comment. Then this comment is deleting database. For editing, system takes editing command from user for user' anyone comment. Then this comment is redirected to editing pop-up. System takes editing comment from user and this change is implemented in database.

4.1.8. Favorite Tag CRUD

This model has four main process. Firstly, tag listing shows to users' tags for current user. This list is changeable, because users add new tags or deleting own tags. Second process is adding tag to tag list for current user. Users click add button then open a pop-up, users choose desiring tags that is in pop-up page, finally confirm tags and send. Then comment is adding to database and page is redirected to last page. Tag list is refreshed and new tag added to profile tag information by system. Third process is deleting tag in user tag list. Users click delete button then tag is removed from profile tag information.

4.1.8.1. Screen images





4.1.8.2. Objects and actions

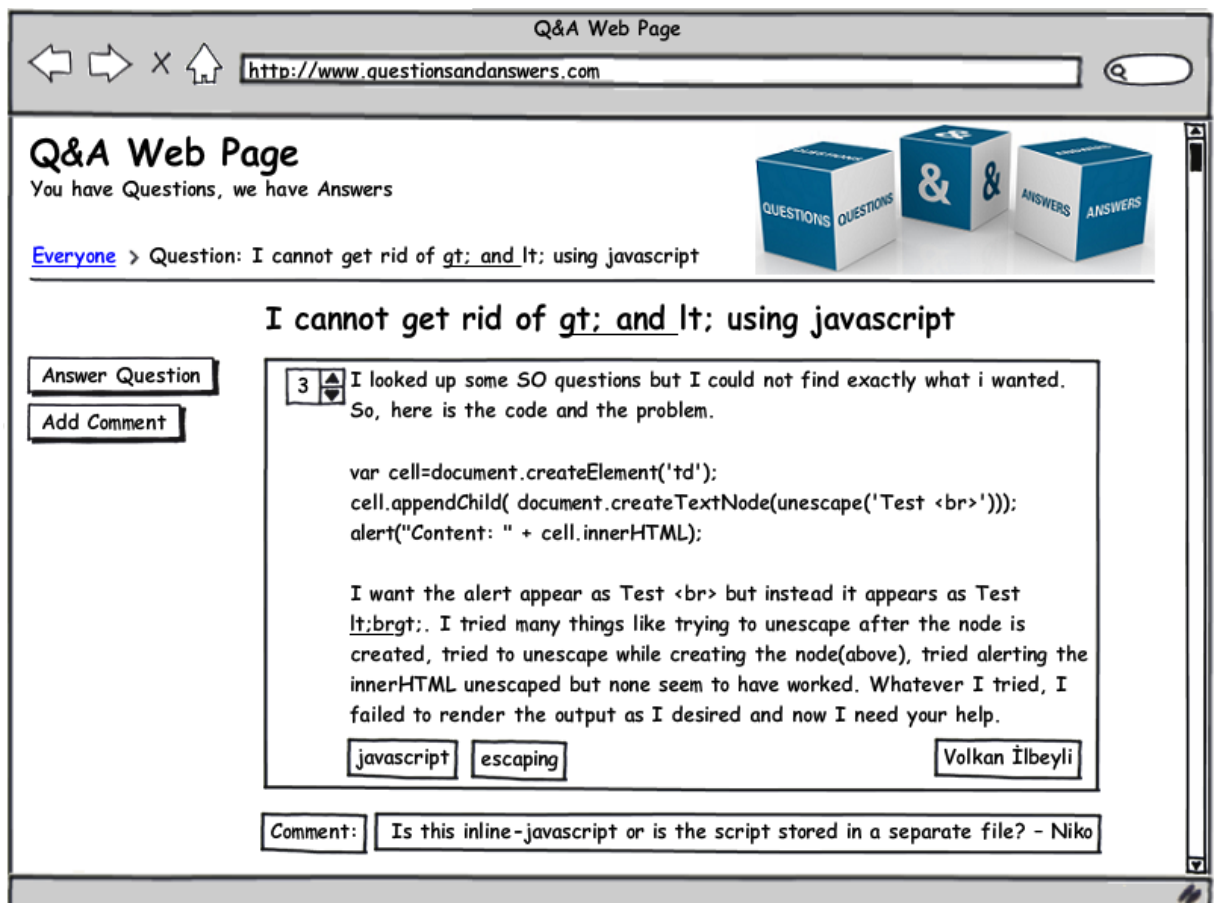
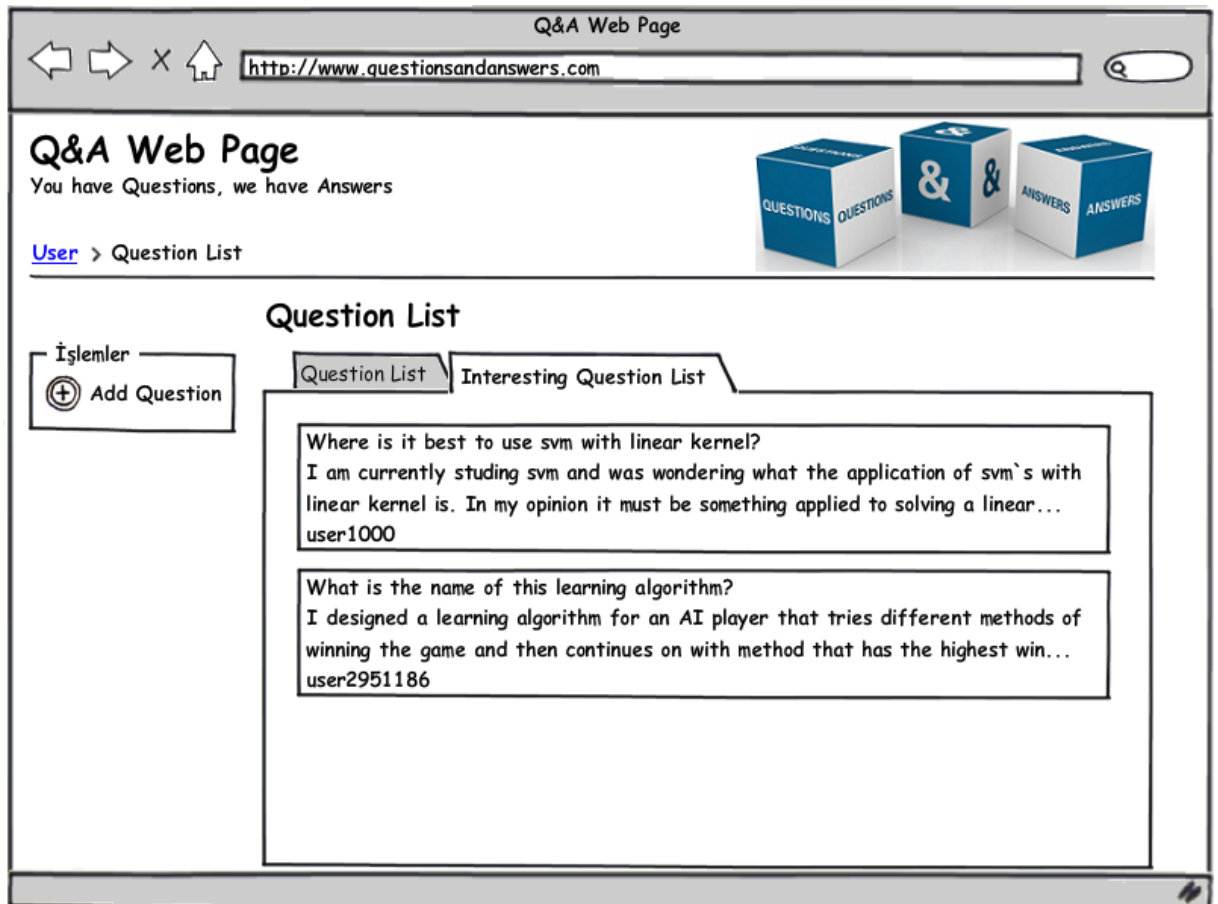
For listing, system is takes tags from database for current user and lists in table with tags values. For adding, system takes adding new tags command from user and user choose tags for adding in pop-up screen. Then it is added to database fairly by system. For deleting, system takes deleting tag command from user for user' anyone tag. Then this tag is deleting from database.

4.1.9. Interesting Questions List

As described in the section 3.2.1.2, Interesting Questions Page is composed of a main division for list of interesting questions besides the navigation parts. Questions are listed below each other. Each division of a question includes the question title, content of the question (only two lines of the whole question, whose rest is available after opening the question), and the name of the user at the bottom.

4.1.9.1. Screen images

Interesting questions list, question view, and answers/comments of a question are as follows.



Q&A Web Page

http://www.questionsandanswers.com

Q&A Web Page

You have Questions, we have Answers

QUESTIONS

QUESTIONS

&

&

ANSWERS

ANSWERS

Everyone > Question: I cannot get rid of `gt;` and `lt;` using javascript

I cannot get rid of `gt;` and `lt;` using javascript

Answer Question

Add Comment

3

I looked up some SO questions but I could not find exactly what i wanted. So, here is the code and the problem.

```
var cell=document.createElement('td');
cell.appendChild( document.createTextNode(unescape('Test <br>')));
alert("Content: " + cell.innerHTML);
```

Answer:

What you are doing is create a textNode `
` and its converting the `<` and `>` to html code. You have to insert an element if you want the alert to properly show `
` tag. Hope this helps you

```
var cell=document.createElement('td');
var br = document.createElement('br');
cell.appendChild( document.createTextNode(unescape('Test')));
```

Save

Cancel

Q&A Web Page

http://www.questionsandanswers.com

Q&A Web Page

You have Questions, we have Answers

QUESTIONS

QUESTIONS

&

&

ANSWERS

ANSWERS

Everyone > Question: I cannot get rid of `gt;` and `lt;` using javascript

I cannot get rid of `gt;` and `lt;` using javascript

Answer Question

Add Comment

3

I looked up some SO questions but I could not find exactly what i wanted. So, here is the code and the problem.

```
var cell=document.createElement('td');
cell.appendChild( document.createTextNode(unescape('Test <br>')));
alert("Content: " + cell.innerHTML);
```

Comment:

Is this inline-javascript or is the script stored in a separate file?

Save

Cancel

4.1.9.2. Objects and actions

Interesting questions page consists of a table element which lists the questions. This list is opened through the tab above the page. A question element simply includes three text fields where the question title, question content and user name are written respectively below each other. On the left, there is a button for adding a new question.

Question view page consists of a large text field where the whole question is written, and two buttons for adding either an answer or a comment at the left top . At the bottom left, tags are written, at the bottom right, username of the user who asked the question is written.

Answer question page has a text field opening below the question. Since new text field is required, the question is shortened. Below the answer text field, there are two options which are save and cancel. Save button adds the answer to the list and database. Cancel button cancels the answer.

Comment on question page has a text field opening below the question. Since new text field is required, the question is shortened. Below the comment text field, there are two options which are save and cancel. Save button adds the comment to the list and database. Cancel button cancels the comment.

4.1.10. Questions List

As described in the section 3.2.2.2, Questions Page is composed of a main division for list of interesting questions besides the navigation parts. Questions are listed below each other. Each division of a question includes the question title, content of the question (only two lines of the whole question, whose rest is available after opening the question), and the name of the user at the bottom.

4.1.10.1. Screen images

Questions list, question view, and answers/comments of a question are as follows.


Q&A Web Page

http://www.questionsandanswers.com

Q&A Web Page

You have Questions, we have Answers

[User](#) > Question List



Question List

İşlemler

[Add Question](#)

Question List

Interesting Question List

Phpstorm Intellisense / Code completion for inherited properties

I have a problem with Phpstorm's code completion for inherited properties. There is an example from my code below...

Gökhan Çoban

I cannot get rid of `gt;` and `lt;` using javascript

A second line of text

I looked up some SO questions but I could not find exactly what i wanted. So, here is the code and the problem...

Volkan İlbeyli

Unseen nominal values in weka

I have a dataset with some nominal values as features. The training set I have has a set of values for the nominal features which are absent in my test set...

DaTaBomB


Q&A Web Page

http://www.questionsandanswers.com

Q&A Web Page

You have Questions, we have Answers

[Everyone](#) > Question: I cannot get rid of `gt;` and `lt;` using javascript



I cannot get rid of `gt;` and `lt;` using javascript

Answer Question

Add Comment

3

I looked up some SO questions but I could not find exactly what i wanted. So, here is the code and the problem.

```
var cell=document.createElement('td');
cell.appendChild( document.createTextNode(unescape('Test <br>')));
alert("Content: " + cell.innerHTML);
```

I want the alert appear as Test
 but instead it appears as Test `lt;brgt;`. I tried many things like trying to unescape after the node is created, tried to unescape while creating the node(above), tried alerting the innerHTML unescaped but none seem to have worked. Whatever I tried, I failed to render the output as I desired and now I need your help.

javascript escaping

Volkan İlbeyli

Comment:

Is this inline-javascript or is the script stored in a separate file? - Niko

Q&A Web Page

← → × ↗

http://www.questionsandanswers.com

Q

Q&A Web Page

You have Questions, we have Answers

QUESTIONS QUESTIONS

& &

ANSWERS ANSWERS

Everyone > Question: I cannot get rid of `gt;` and `lt;` using javascript

I cannot get rid of `gt;` and `lt;` using javascript

Answer Question

Add Comment

3 ▲ ▼

I looked up some SO questions but I could not find exactly what i wanted. So, here is the code and the problem.

```
var cell=document.createElement('td');
cell.appendChild( document.createTextNode(unescape('Test <br>')));
alert("Content: " + cell.innerHTML);
```

Answer:

What you are doing is create a textNode `
` and its converting the `<` and `>` to html code. You have to insert an element if you want the alert to properly show `
` tag. Hope this helps you

```
var cell=document.createElement('td');
var br = document.createElement('br');
cell.appendChild( document.createTextNode(unescape('Test')));
```

Save

Cancel

Q&A Web Page

← → × ↗

http://www.questionsandanswers.com

Q

Q&A Web Page

You have Questions, we have Answers

QUESTIONS QUESTIONS

& &

ANSWERS ANSWERS

Everyone > Question: I cannot get rid of `gt;` and `lt;` using javascript

I cannot get rid of `gt;` and `lt;` using javascript

Answer Question

Add Comment

3 ▲ ▼

I looked up some SO questions but I could not find exactly what i wanted. So, here is the code and the problem.

```
var cell=document.createElement('td');
cell.appendChild( document.createTextNode(unescape('Test <br>')));
alert("Content: " + cell.innerHTML);
```

Comment:

Is this inline-javascript or is the script stored in a separate file?

Save

Cancel

4.1.10.2. *Objects and actions*

Questions page consists of a table element which lists the questions. This list is opened through the tab above the page. A question element simply includes three text fields where the question title, question content and user name are written respectively below each other. On the left, there is a button for adding a new question.

Question view page consists of a large text field where the whole question is written, and two buttons for adding either an answer or a comment at the left top . At the bottom left, tags are written, at the bottom right, username of the user who asked the question is written.

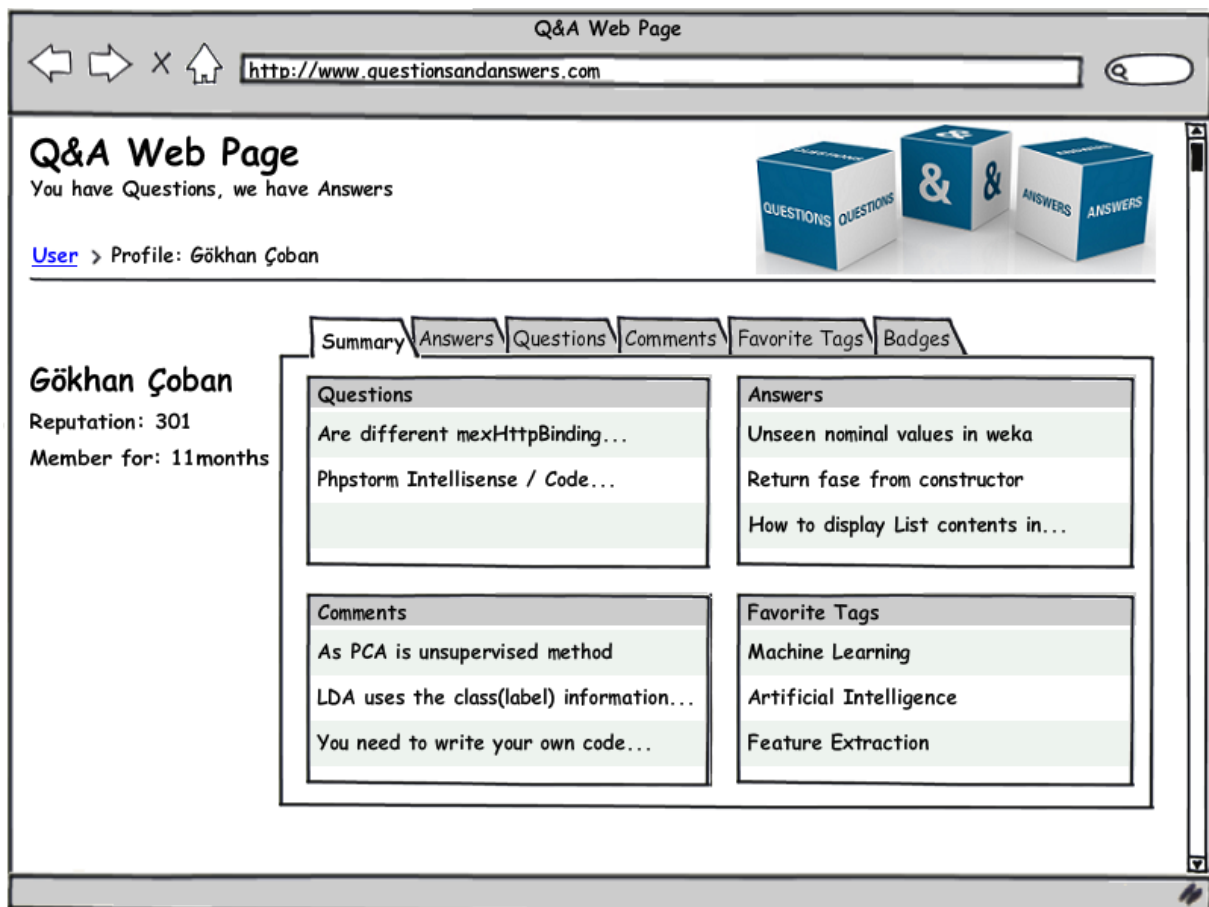
Answer question page has a text field opening below the question. Since new text field is required, the question is shortened. Below the answer text field, there are two options which are save and cancel. Save button adds the answer to the list and database. Cancel button cancels the answer.

Comment on question page has a text field opening below the question. Since new text field is required, the question is shortened. Below the comment text field, there are two options which are save and cancel. Save button adds the comment to the list and database. Cancel button cancels the comment.

4.1.11. User Profile Page

User profile page will be built made on a page which users can see and edit their profile informations like nickname, bio and location. When the edit button is pressed user can edit these informations and the delete button is pressed user can delete this informations. Users also can see their site status information such as previous answers, badges, tags and reputations. Obviously this page can be displayed to only registered users.

4.1.11.1. Screen images



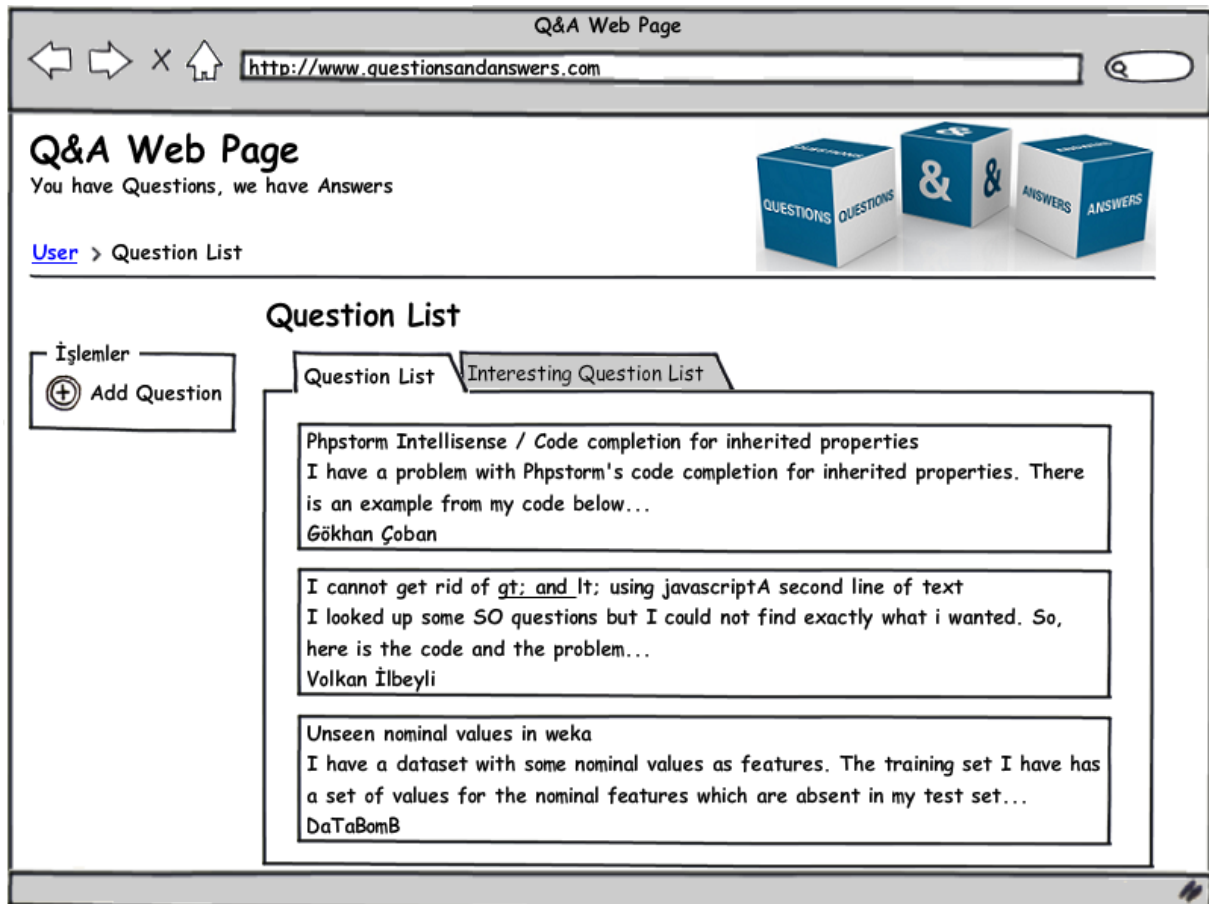
4.1.11.2. Objects and actions

Summary tab is the main tab in user profile page. In this tab brief information about user is listed about questions, answers, comments, favorite tags. Besides, these items have their own tabs upper side of the main frame. When clicking tabs changes the main frame to their content in the same page. Left side of the main frame users personal informations are listed.

4.1.12. Questions Page

Question list and question pages are one of the primarily pages of Q&A site. These pages construct one of the main objectives of the web site. In these pages users view questions and then ask questions via their links on listing questions.

4.1.12.1. Screen images



Q&A Web Page

You have Questions, we have Answers

[User](#) > Add Question

Add Question

Please follow the rules in your questions.
Homework questions are not allowed.
=>Detailed Rules<=

* required

Title: *

Text: *

Question Tags:

Choose Tags ▼

- Asp.Net
- Html
- Mysql
- C++

Save Cancel

4.1.12.2. Objects and actions

“Add question” button on the question list page directs to add question page which contains text input areas for question. These areas are title and question text itself also a question tags dropdown menu is displayed for choosing question tags. Question title and question texts are required areas. “Save” button on the add question page sends question information to database and question begins to show on questions list page. “Cancel” button is cancels the adding question process and returns the man page.

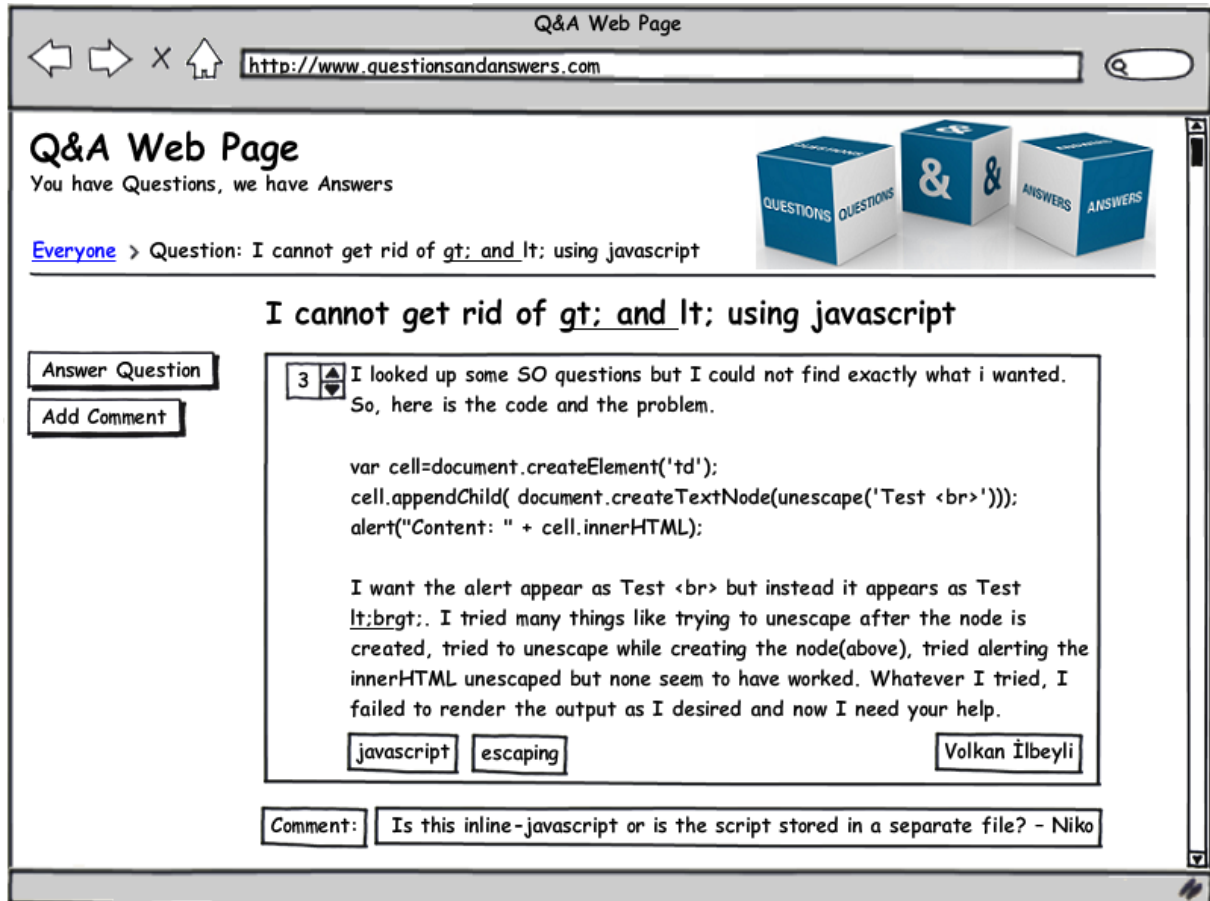
Question list page has a table which contains questions titles, brief of question text and the user who asked the question. If user clicks the one of the questions in the question list page, user directs to question’s page.

4.1.13. Questions View Page

Question page is contains question’s itself. This page is the only page in the site that you can display the question texts and its comments. Everyone can display question page but only the

registered user can answer or comment the question. And only the user who owns of the content can change the content of these page after the adding them.

4.1.13.1. Screen images



4.1.13.2. Objects and actions

"Answer Question" button on the question page directs to answer page of this question. "Add Comment" button on the question page also directs to comment question page of this question.

Comments are listed under the question text by next to its owner. Question frame has question title on the top, vote number and vote up / vote down buttons on the left upper corner, question tags under the question text frame and name of the user who asked the question on the right lower corner of the question text.

Up vote button increments the vote counter and down vote button decrements vote counter.

4.2. Interface design rules

For the database related operations (CRUD), a convention will be used. There will be a page for each operation (add, edit, delete, list). Add and edit lists will be almost identical, only differing in the input contents, i.e., add item page will have blank input fields while edit item page will have input fields filled with the information of the edited item. Other than those, the two pages will be identical. Delete operation doesn't really require a distinct page, however for modularity a delete page is created for every item. When it is called, the blank page will appear, item will be deleted from database, and user is redirected to listing page. List item page will have a separate add button and edit/delete buttons near each listed item. This convention will be used for all database-related items of the webpage.

4.3. Components available

Twitter Bootstrap CSS framework will be used for the whole website for its simplicity and elegance. It is readily available at Twitter's page for Bootstrap. A base page will be written for the CRUD operations that each developer will use when writing their pages.

4.4. UIDS description

Balsamiq Mockups software will be used to visualize the layout of the pages before the implementation, thus helping developers to develop pages without the concerns of layout problems.

5. Restrictions, limitations, and constraints

- Tag CRUD Pages are only seen by admin. Admin adds new tags or deletes current tags or edits current tags value. Admin uses this page for tag process.
- Badge CRUD Pages are only seen by admin. Admin adds new badges or deletes current badges or edits current badges value. Admin uses this page for tag process.
- Question Delete Page is only seen by admin. Admin deletes some unnecessary and wrong format questions from database.
- User Delete Page is only seen by admin. Admin deletes some users who use application maliciously or don't use application for a long while.
- Question Answer CRUD Pages are used by users but also admin can see this page. Users add, delete or edit answers for questions in this page.
- Question Comment CRUD Pages are used by users but also admin can see this page. Users add, delete or edit comments for questions in this page.
- Interesting Question List page is only seen by current user. Question list in this page is determined according to current user favorite tags.

- User Profile Edit Page is only seen by current user. User change profile information thank to this page.
- Question CRUD Pages are used by current user but admin can see this page. In this page, current user adds to database new question or edits existing question or deletes existing question.
- Favorite Tag CRUD Pages are only used by current user. User adds new tags to own profile or delete existing tags from profile tag list.
- Question List Page is seen from by users and admins. In this page, last adding questions are listed by system.
- User Login is seen by admins, users and guests. This page is creating for login process of existing accounts.
- Question View page is seen by admins and users. In this page, information (question content, owner name, adding time) is showed by system.

6. Testing Issues

Test strategy and preliminary test case specification are presented in this section.

6.1. Classes of tests

Interfaces will be tested (black-box testing) for if they have a proper layout or not and for error displaying. On the white-box testing case, the components will be tested for whether they can perform the CRUD operations correctly and users will be tested for privilege cases.

6.2. Expected software response

The CRUD pages should be able to perform their operations (adding, deleting, editing and listing items) correctly. Login page should be safe and display errors when wrong input is given. A user should not be able to perform admin operations.

6.3. Performance bounds

The website should be able to display results of the queries from database in a feasible amount of time.

6.4. Identification of critical components

Since question and answer listing will be most used components they are the most critical components.