

**Enterprise Web** **Software Development**

|  |  |  |
| --- | --- | --- |
| **COLLECTING IDEAS MANAGEMENT SYSTEM** | | |
| **Group members** | **Nguyen Ngoc Chan** | **Scrum Master, Tester** |
| **Hoang Ngan Giang** | **Developer, Designer** |
| **Tran Hoang Long** | **Developer, Database** |
| **Doan Dinh Huy** | **Developer, Designer,** |
| **Trieu Phu Vinh** | **Coder, Tester** |
| **Teacher** | **To Hoai Viet** | |

URL for the Screencast:

<https://drive.google.com/open?id=1hhG8IakT5zDHJVasPwQt3jYs7vycI5ui>

URL for the source code:

<https://github.com/kenguyen01/blogapplication>

**Acknowledgment**

*We want to send many thanks to M.Sc To Viet Hoai who instruction benefits us a*  
*lot in this report. Besides, I'm also thankful to his feedback and answers to general*  
*questions relating to this report.*

Contents

[1. Introduction 5](#_Toc511325639)

[2. Agile Scrum documentation 6](#_Toc511325640)

[2.1. Roles 6](#_Toc511325641)

[2.2. Tools 6](#_Toc511325642)

[2.3. Meetings 7](#_Toc511325643)

[2.3.1. Sprint planning 7](#_Toc511325644)

[2.3.2. Pre-kick off meeting 7](#_Toc511325645)

[2.3.3. Sprint Review 7](#_Toc511325646)

[2.3.4. Sprint Retrospective 7](#_Toc511325647)

[2.3.5. Daily Scrum 7](#_Toc511325648)

[2.4. Product Backlogs 8](#_Toc511325649)

[2.5. Spint Backlogs 8](#_Toc511325650)

[2.6. Project Burn down chart 8](#_Toc511325651)

[3. Design Document 8](#_Toc511325652)

[3.1. Use case diagram 8](#_Toc511325653)

[3.2. Database Design 11](#_Toc511325654)

[3.2.1. Entity Relationship diagram 11](#_Toc511325655)

[3.2.2. Relational Schema 11](#_Toc511325656)

[3.3. Website Design 12](#_Toc511325657)

[3.3.1. Website Wireframes 12](#_Toc511325658)

[3.3.2. Site Map 29](#_Toc511325659)

[3.4. Workflow implementation 29](#_Toc511325660)

[4. Implementation 29](#_Toc511325661)

[4.1. List of files created and Roles 29](#_Toc511325662)

[4.2. Screen shots of system in operation 30](#_Toc511325663)

[5. Testing 39](#_Toc511325664)

[5.1. Test Plan 39](#_Toc511325666)

[5.2. Test case 40](#_Toc511325667)

[5.2.1. Login Test case: 40](#_Toc511325668)

[5.2.2. Admin - Create user account 41](#_Toc511325669)

[5.2.3. Manager – Create new Tag 41](#_Toc511325670)

[5.2.4. Coordinator – Show/Hide Ideas 42](#_Toc511325671)

[5.2.5. Coordinator – Show/Hide Comments 43](#_Toc511325672)

[5.2.6. Student – Add new idea 44](#_Toc511325673)

[5.2.7. Student/Manager/Coordinator – Add comment 45](#_Toc511325674)

[5.2.8. Student/Manager/Coordinator - Give the Thumbs Up or Thumbs Down for any idea 46](#_Toc511325675)

[5.3. Test Log 47](#_Toc511325676)

[6. Evaluation 48](#_Toc511325677)

[6.1. Assumption 48](#_Toc511325679)

[6.2. System Weakness 48](#_Toc511325680)

[6.3. System Strength 48](#_Toc511325681)

[6.4. Learning Outcomes 48](#_Toc511325682)

1. Introduction

It is true that cyber-attacks have become a common phenomenon in every part of our society such as in education .Attacks are targeting different software vulnerabilities on the application layer and bring concerns about software quality and possible measures to react. In any case, it is complicate to enhance and address these vulnerabilities as the complex level of the software and large number of developers participating in applications creation. Although, the security ultimately relate to the general quality of a software product, having only working application is not enough at all ,in addition it must be secure.

We have responsibility to build a strong team to create a secure web-enabled role-based system for collecting ideas for improvement from students in a large University so as to offer a solution that helps solving the fact and need for better understanding of security and better secured software we state the research questions:

The University has a Quality Assurance Manager to oversee the process.

* All Departments have a QA coordinator who is responsible for managing the process
* for their Department, and for encouraging students to contribute.
* All students have the opportunity to submit one or more ideas.
* All students must agree to Terms and Conditions before they can submit.
* All students can optionally upload documents to support their ideas.
* All ideas can be categorised (tagged) from a list of categories at the point when they
* are submitted.
* The QA Manager can add additional categories at any time, and can delete categories,
* but only if they have not been used.
* All staff and students can see all submitted ideas and can comment on any idea. They
* can also give the Thumbs Up or Thumbs Down for any idea, but only once for any
* idea
* Staff comments are only visible to other staff. Student comments are visible to both
* staff and students,
* Ideas and comments can be posted anonymously, although the author’s details will be
* stored in the database so any inappropriate ideas can be investigated.
* All new ideas are disabled after a closure date for new ideas, but comments can
* continue to be done until a final closure date.
* Once an idea is submitted the system emails a notification to the Department’s QA
* Coordinator.
* The author of an idea receives an automatic email notification whenever a comment is
* submitted to any of their ideas by a student (but not by a member of staff).
* Lists of Most Popular Ideas (+1 for Thumbs Up, -1 for Thumbs Down), Most Viewed
* Ideas, Latest Ideas and Latest Comments must be made available to all staff.
* Lists of Ideas need to be paginated (5 per page)
* The University QA Manager needs to be able to download all the selected
* contributions after the final closure date in a ZIP file for transfer out of the system.
* An administrator is needed to maintain any system data, e.g. closure dates for each
* academic year, staff details, student details.
* Statistical analysis (e.g. number of ideas per Department) needs to be available.
* The interface must be suitable for all devices (eg mobile phones, tablets, desktops).

1. Agile Scrum documentation
   1. Roles

The roles needed for the project and circulated between team members are:

* Database designer
* Programmer
* Web designer
* Tester
* Product Owner – Lecturer
  1. Tools

**The technologies develop the system:**

* **The Spring Web MVC Framework[[1]](#footnote-1)**

The Spring Web model-view-controller (MVC) framework is designed around a DispatcherServlet that dispatches requests to handlers, with configurable handler mappings, view resolution, locale and theme resolution as well as support for uploading files.

* **DBMS: Microsoft SQL Server 2012**
* Many security features.
* Multiple Edition.
* **MVC models**

*MVC* provides the ability to separte the *View* (i.e. the UI) from the bussiness logic. That way the bussiness logic can be independently unit tested. The *View* also becomes very thin (i.e. with minimum code) which is good because it is hard to unit test the *View*.

* **Bootstrap**
* Optimal platform.
* Interact well with smartphones.
* Full interface, luxurious.
* Easy customization.
* **Jquery**
* Interface user friendly.
* Save bandwidth.
* Sometimes some coordination function is simple (create, list, edit, delete ...).
  1. Meetings
     1. Sprint planning

Our first meeting was held on 03/03/2018 at 09:00 PM. The agenda was:

* Discussion of personal ability; Strengths, weaknesses that will allow us to divide tasks.
* Review and record technology that we will use.
* Define product backlog:
* The requirement that we say we will have after analyzing the subject.
* Describe the features of the product needed.
* Requires order details by priority, task, person assigned to each hour to spend on duty / s.
* Set a release date
* Determine the content of the finished product
* Discuss how much sprint we will have and the duration of that.
* Define Sprint Backlog (subsets of backlog)
* Requirements for each job as input, then we design, code and test, then with a prototype test as output.
* This will show the deliverables, tasks, hours spent, the status of the tasks; Means start, in the process, complete.
* Create the project burn down char.
  + 1. Pre-kick off meeting

Held on 05/03/2018 04h30’ PM. To ensure that all goals and tasks are understood then start working.

* + 1. Sprint Review

Done after each sprint with temporary deliveries as stated in the courses. *For example:* *Database design, Web site design and test planning. Attendance must also be a trainer in this case as the product owner.*

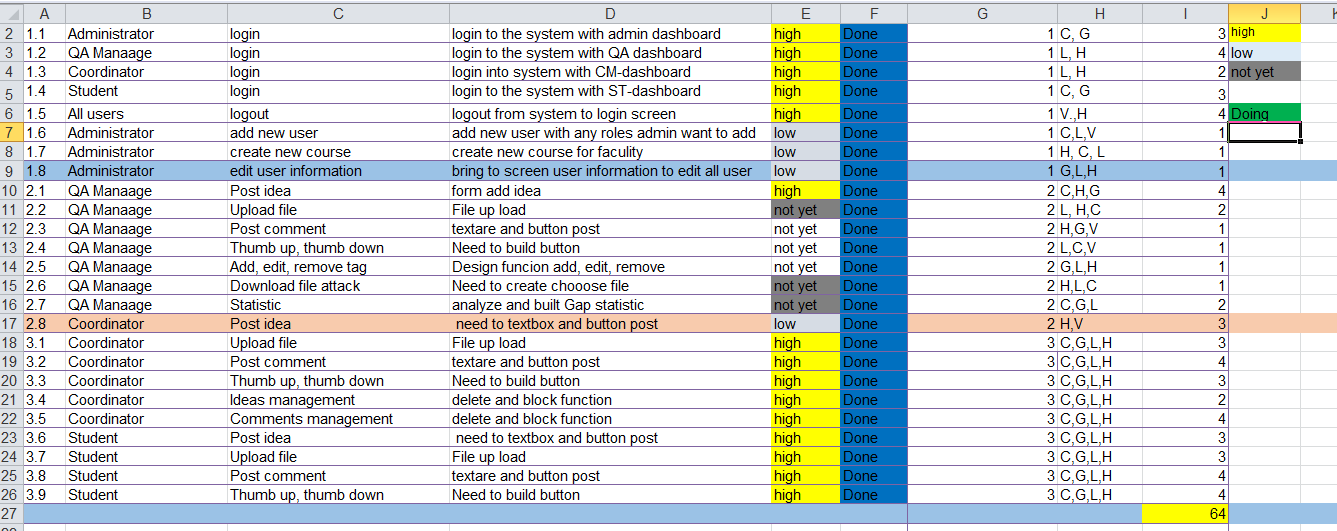
* + 1. Sprint Retrospective

Review work without the presence of product owner (trainer), we have determined whether we should start, stop or continue the project.

* + 1. Daily Scrum

It's been done weekly and sometimes twice a week, not every day.

* 1. Product Backlogs



* 1. Spint Backlogs

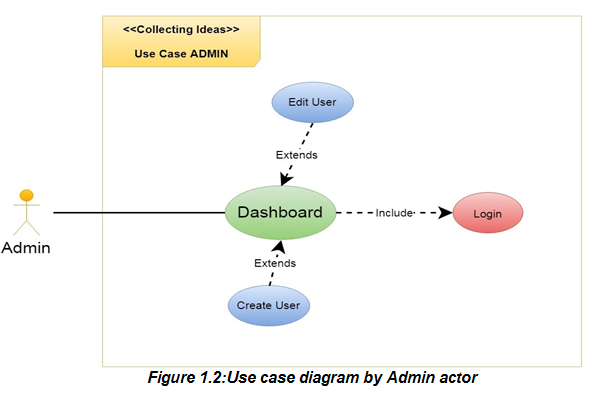
We had a total of four sprints during the project.

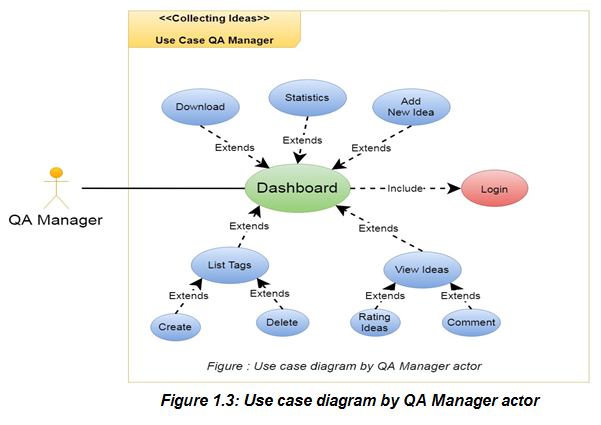
* 1. Project Burn down chart

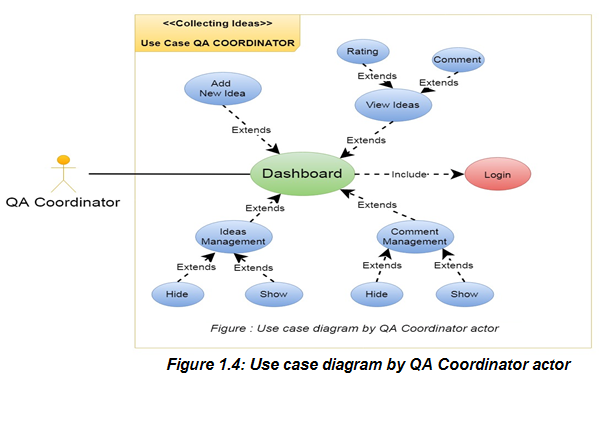
The chart below shows the combustion diagram of the project divided into 220 points representing the whole project task. As we have three sprints; 3 runs for 20 days and last for 84days, totaling 64 days to run the project. The release date is set for Aril 12, 2018.

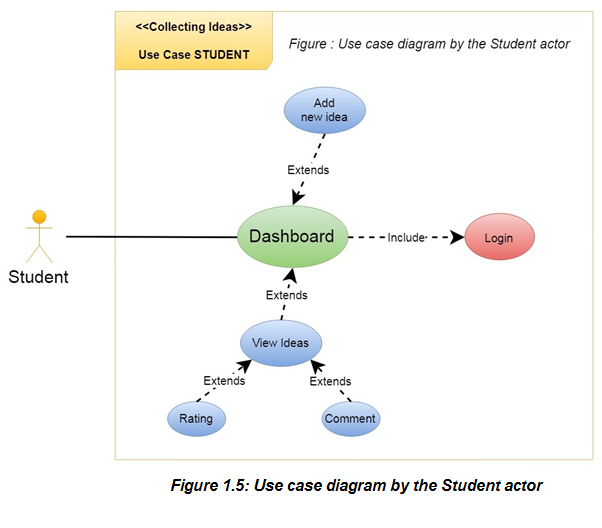
1. Design Document
   1. Use case diagram

For our design, we will use the ‘Use case’ diagram. It should be noted that this design only describes the design of the student claims management system in the university and not anything beyond this scope. The internal and external agents are called actors. Therefore, use case diagram is composed of actors, use cases and their relationship. The diagrams are used to model the systems / subsystems of an application



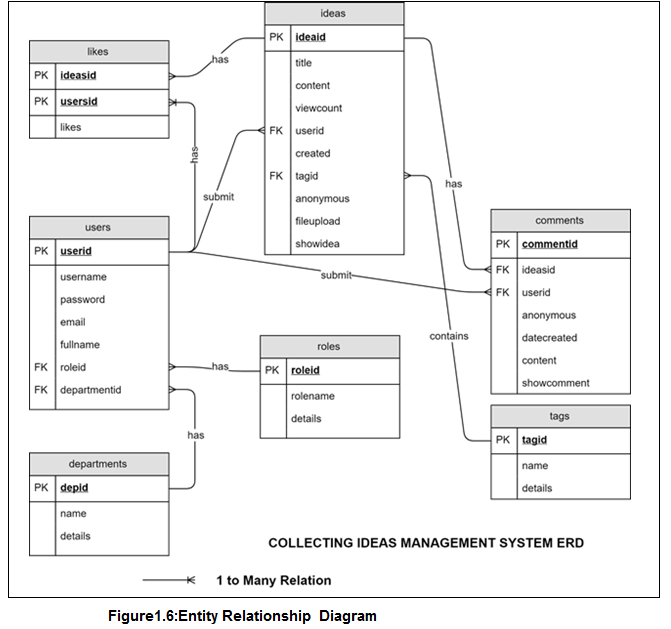






* 1. Database Design
     1. Entity Relationship diagram

ER-model is a data modeling technique used in software engineering to create a conceptual data model of an information system. The diagram generated using the ER technique-this model is called entityrelationship diagrams, ER diagrams or ERDs. So you can say that entity relations diagrams illustrate the logical structure of the database. And so below

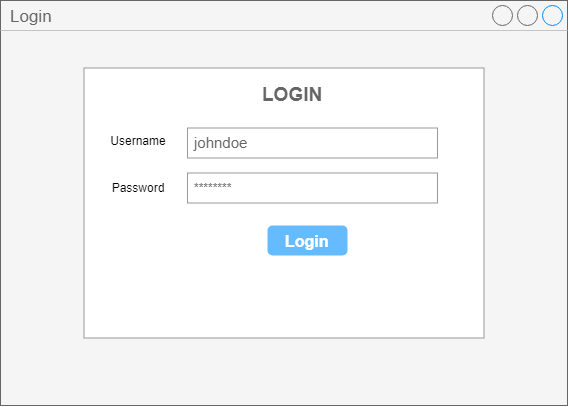


* + 1. Relational Schema

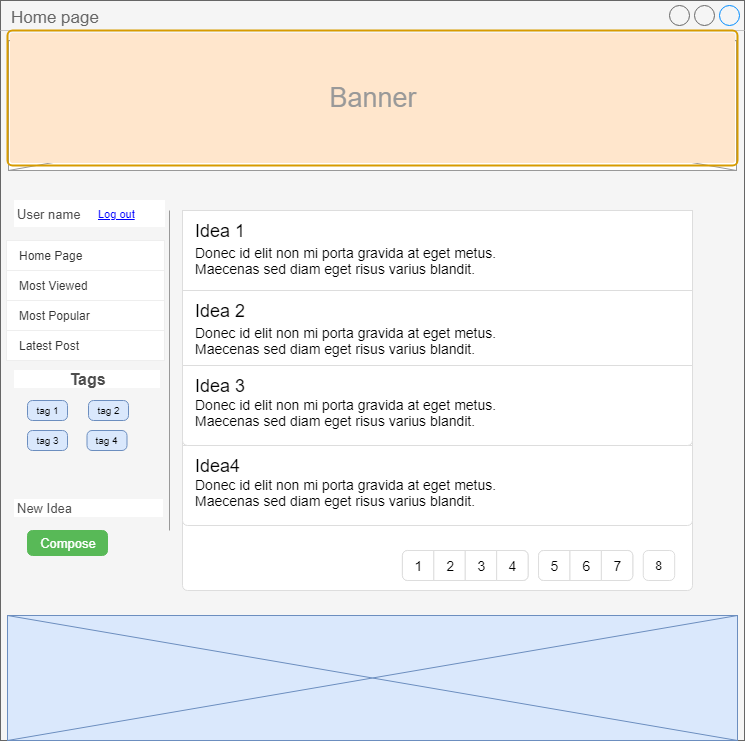
**NOTE:** the Primary Keys are underlined, bold and Foreign Keys are inred

* Table\_Users ( **Users\_userid**, Users\_username, Users\_password, Users\_email, Users\_fullname, Users\_roleid, Users\_departmentid )
* Table\_Ideas ( **Ideas\_ideaid**, Ideas\_title, Ideas\_content, Ideas\_viewcount, Ideas\_userid, Ideas\_created, Ideas\_tagid, Ideas\_anonymous, Ideas\_fileupload, Ideas\_showidea )
* Table\_Departments ( **Departments\_depid**, Departments\_name, Departments\_details )
* Table\_Likes ( **Likes\_ideasid**, **Likes\_userid**, Likes\_likes )
* Table\_Roles ( **Roles\_roleid**, Roles\_rolename, Roles\_details )
* Table\_Comments ( **Comments\_commentid**, Comments\_ideasid, Comments\_userid, Comments\_anonymous, Comments\_datecreated, Comments\_content, Comments\_showcomment)
* Table\_Tags ( **Tags\_tagid**, Tags\_name, Tags\_details )
  1. Website Design
     1. Website Wireframes

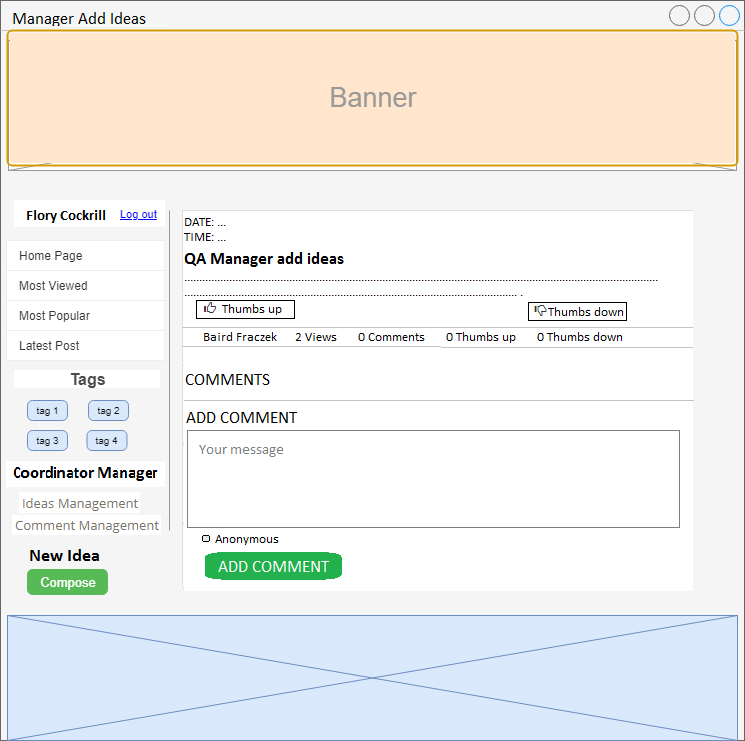
Below is a design of the log in and home pages and other pages (the other pages follow a general design)

****

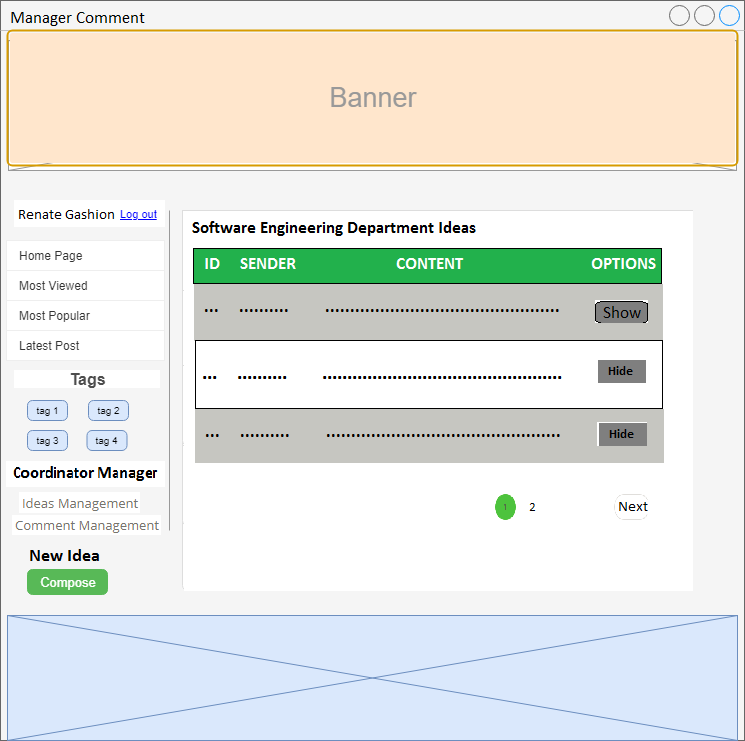
**Login**

****

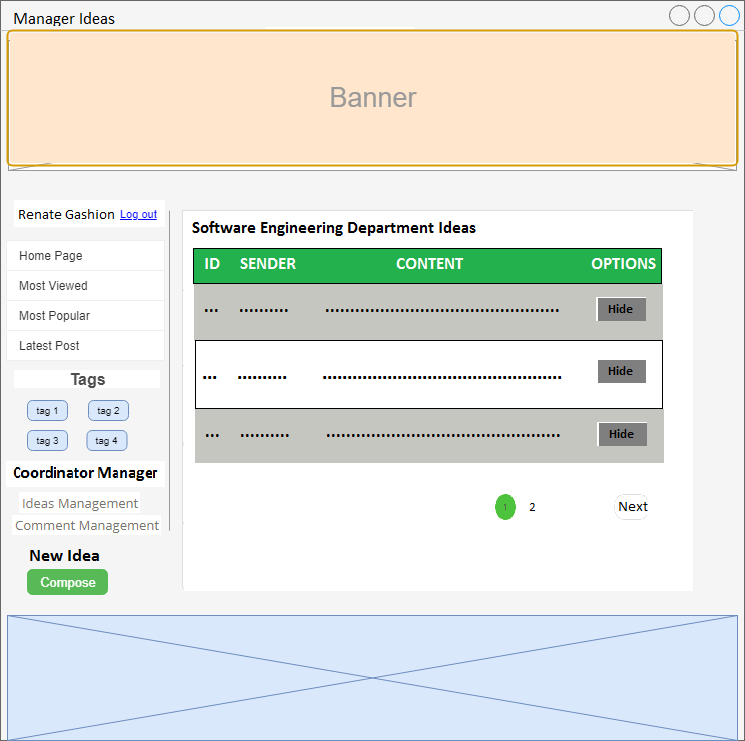
**Home Page**

****

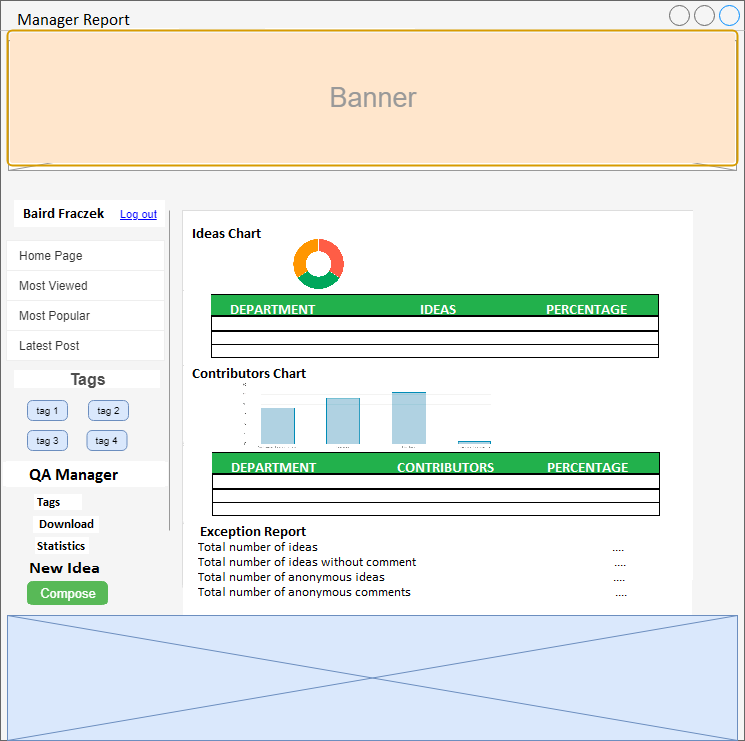
**Manager Add Ideas**

****

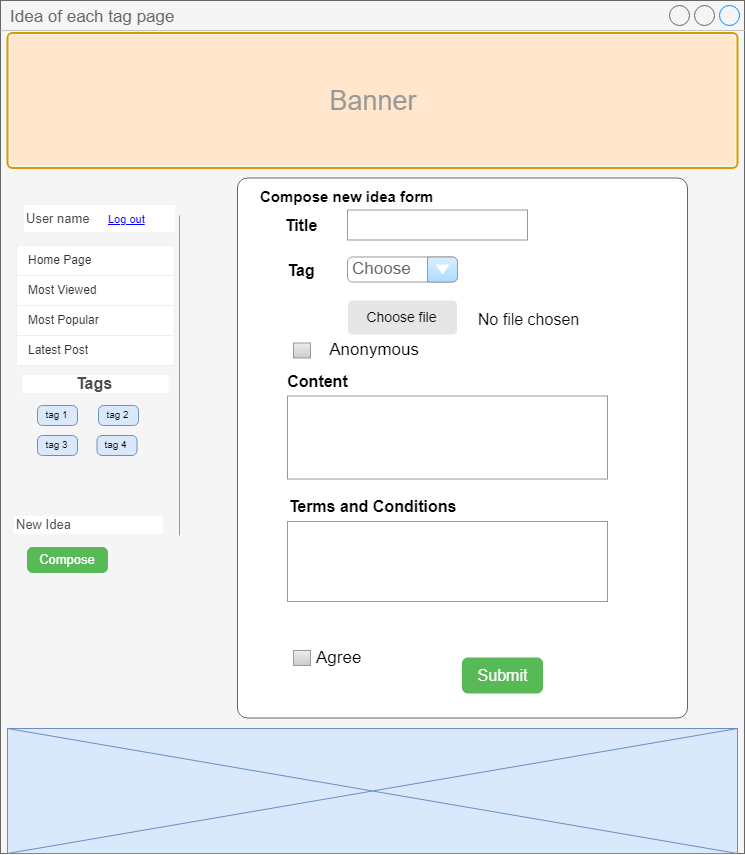
**Manager comment**

****

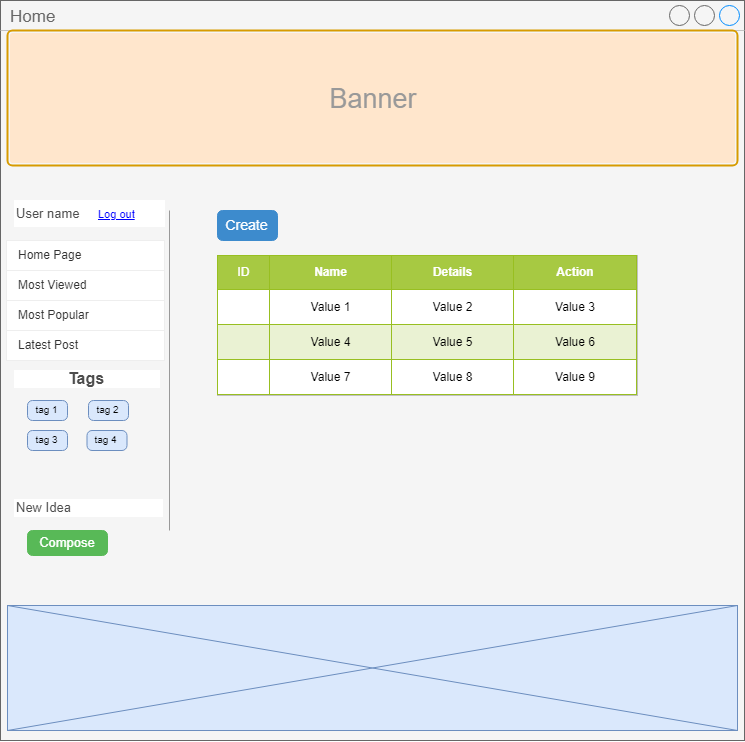
**Manager ideas**

****

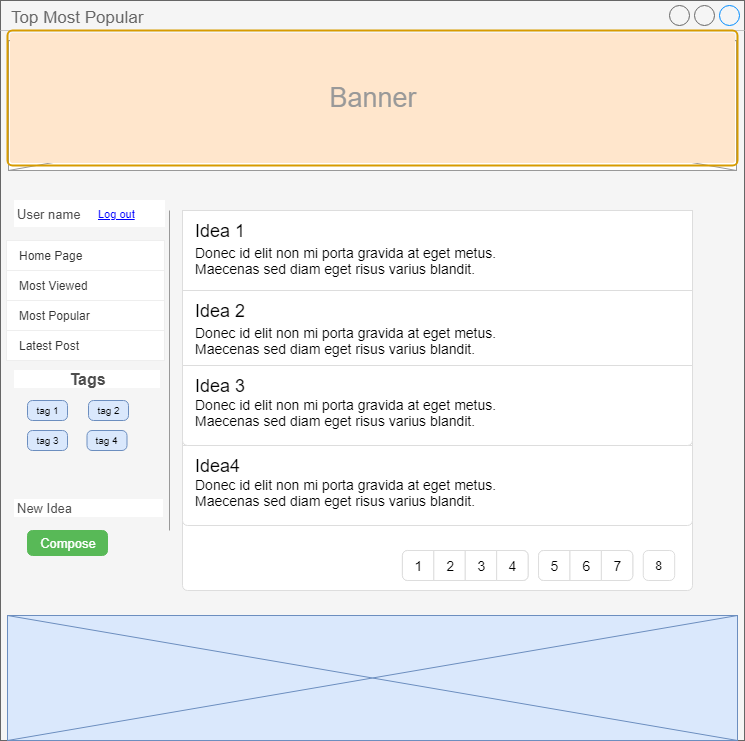
**Manager Report**

****

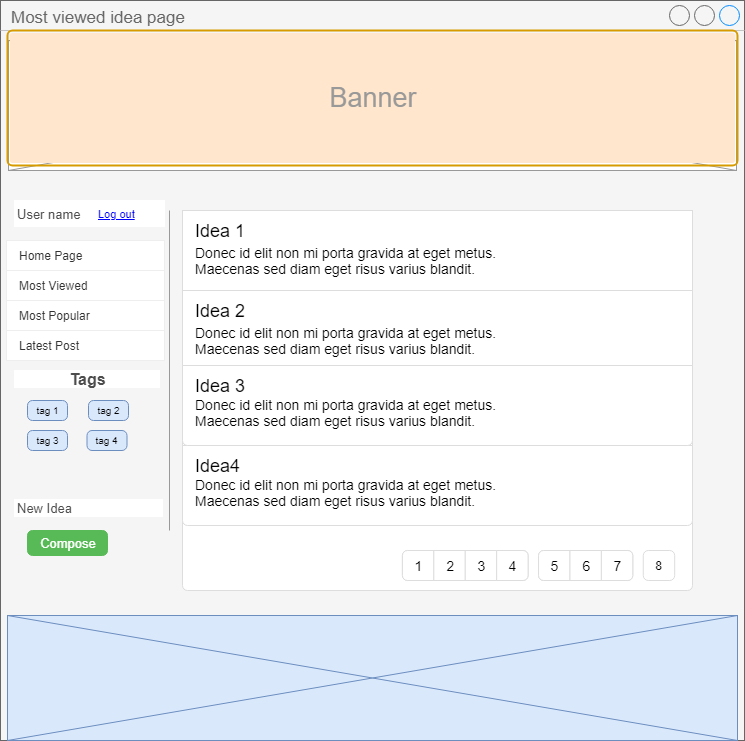
**Tag idea page**

****

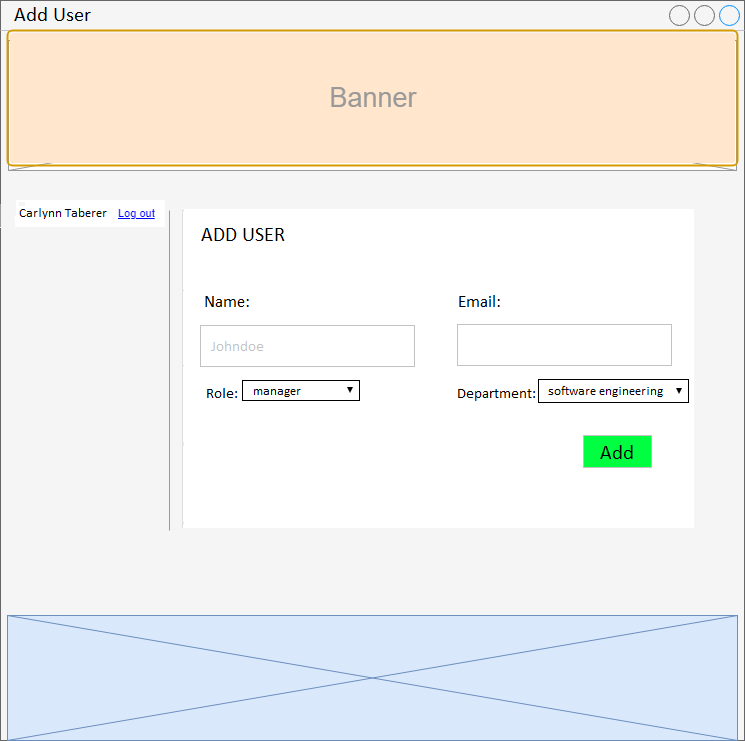
**Tag manager**

****

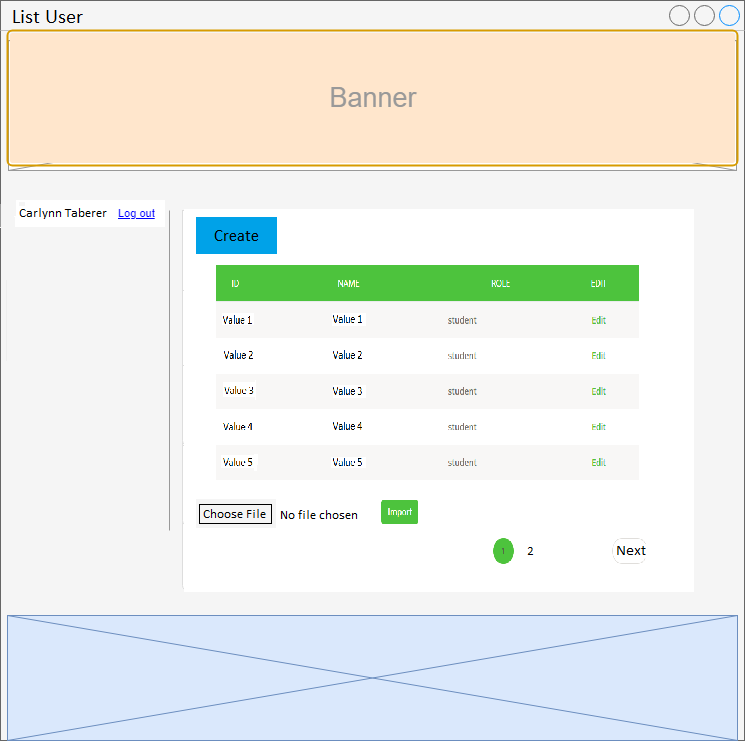
**Most popular**

****

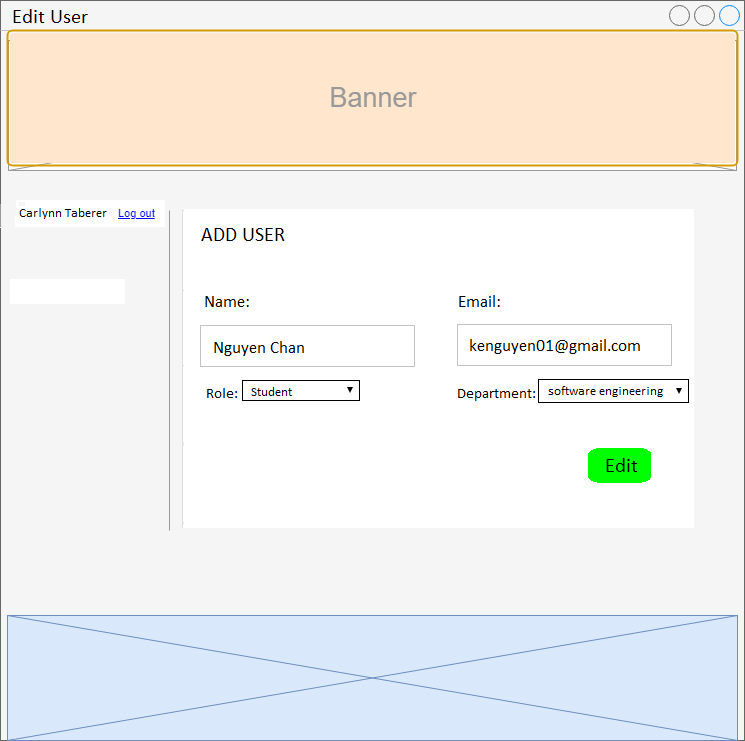
**Most viewed**

****

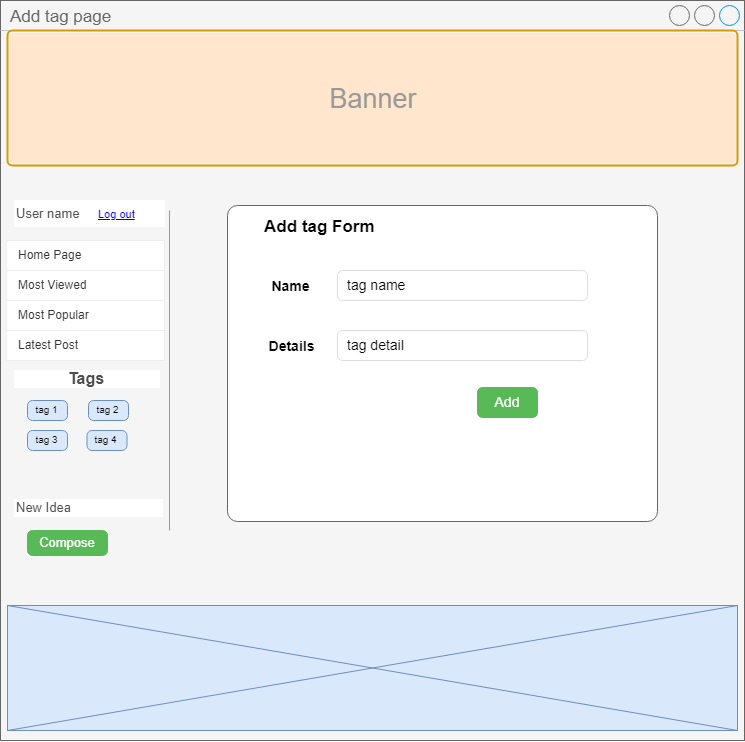
**Add user**

****

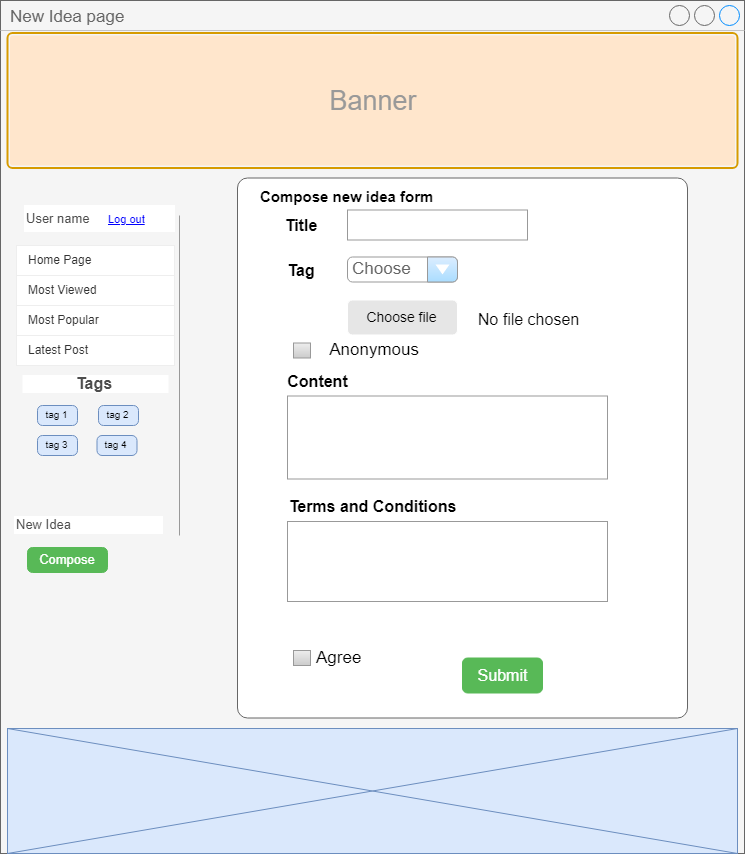
**List user**

****

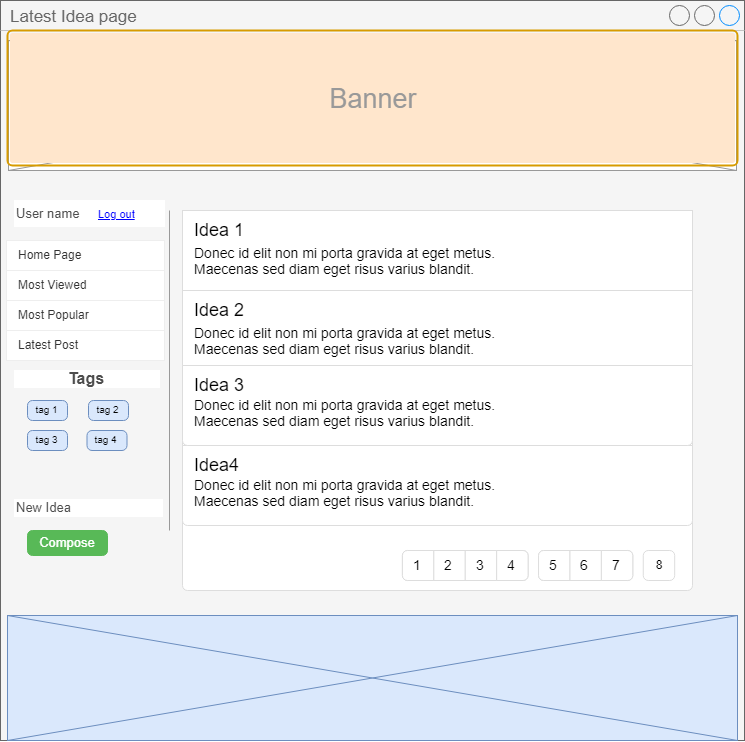
**Edit user**

****

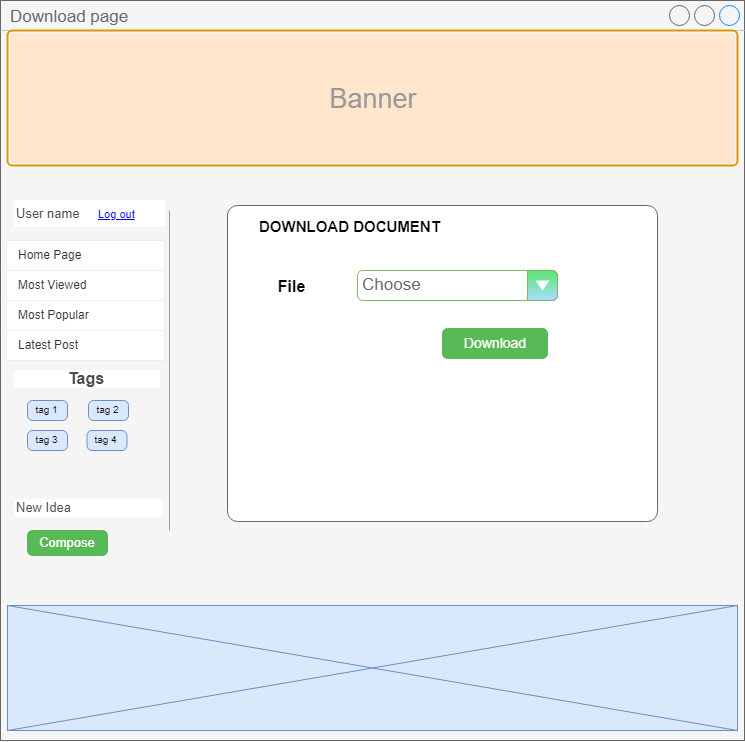
**Add tag**

****

**New idea**

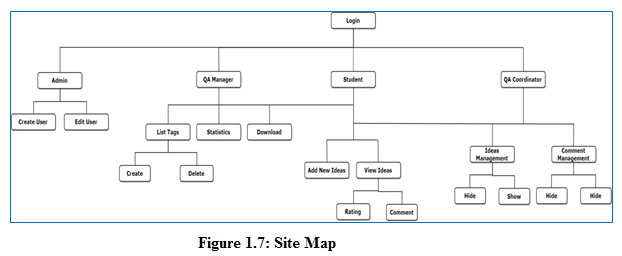
****

**Latest idea**

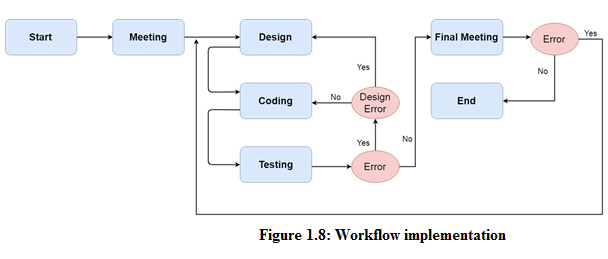
****

**Download page**

* + 1. Site Map



* 1. Workflow implementation

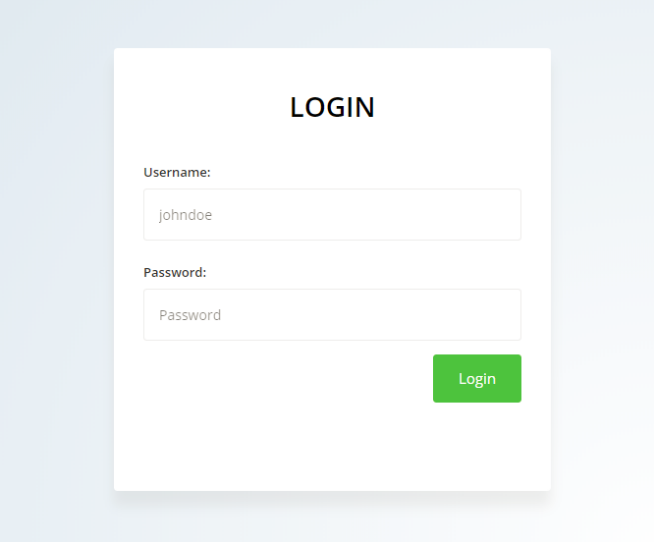


1. Implementation
   1. List of files created and Roles

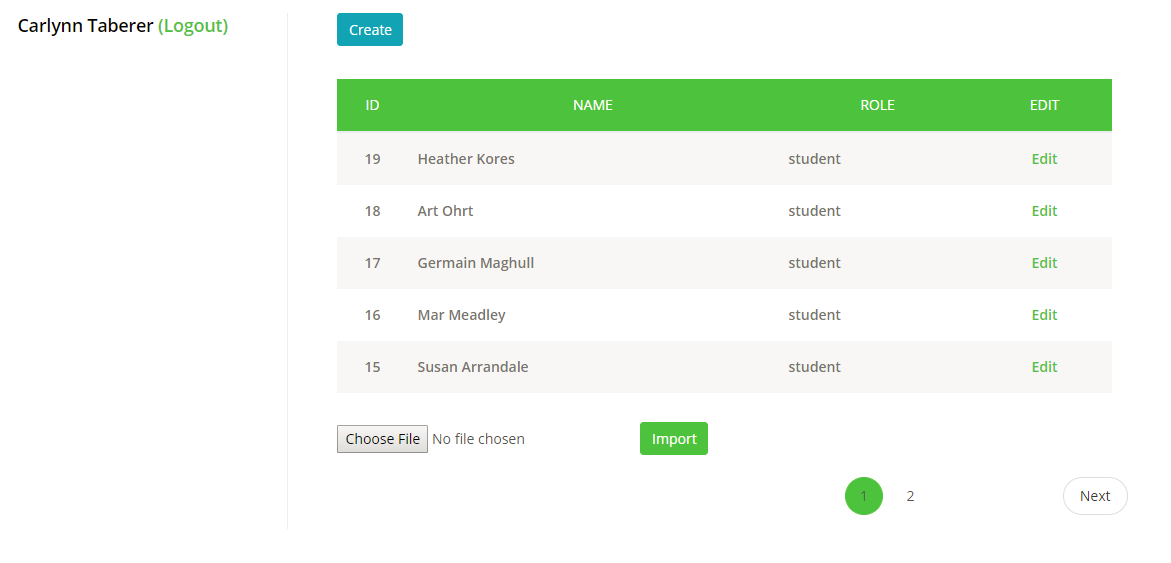
|  |  |  |
| --- | --- | --- |
| No | Page Name | Role |
| 1 | Comments.java | Model class |
| 2 | Departments.java | Model class |
| 3 | Ideas.java | Model class |
| 4 | Likes.java | Model class |
| 5 | Roles.java | Model class |
| 6 | Tags.java | Model class |
| 7 | Users.java | Model class |
| 8 | HomeCtrl.java | Controller class, it controls all main functions of the application |
| 9 | AdminCtrl.java | Controller class, it contains functions of admin |
| 10 | IndexCtrl.java | Controller class, it contain login and logout function. |
| 11 | Addidea.html | HTML page, it is used to add new idea. |
| 12 | Addtags.html | HTML page, it is used to add new tag |
| 13 | Adduser.html | HTML page, it is used to add new user. |
| 14 | Admin.html | HTML page, it is used to display main page of admin |
| 15 | Depscoms.html | HTML page, it is used to hide or show comment of one department. |
| 16 | Depsideas.html | HTML page, it is used to hide or show ideas of one department |
| 17 | Download.html | HTML page, it is used to download file attack for manager. |
| 18 | Edituser.html | HTML page, it is used to edit user information. |
| 19 | Homepage.html | HTML page, it is used to show ideas. |
| 20 | Index.html | HTML page, it is used to login to the system. |
| 21 | Singlepage.html | HTML page, it is used to display one single idea. |
| 22 | Statistics.html | HTML page, it is used to view report for manager |
| 23 | Tags.html | HTML page, it is used to view all tags for manager |
| 24 | Tagsidea.html | HTML page, it is used to display all ideas of one tag. |
| 25 | Terms.html | HTML page, it is used to view terms and conditions for adding new idea. |

* 1. Screen shots of system in operation

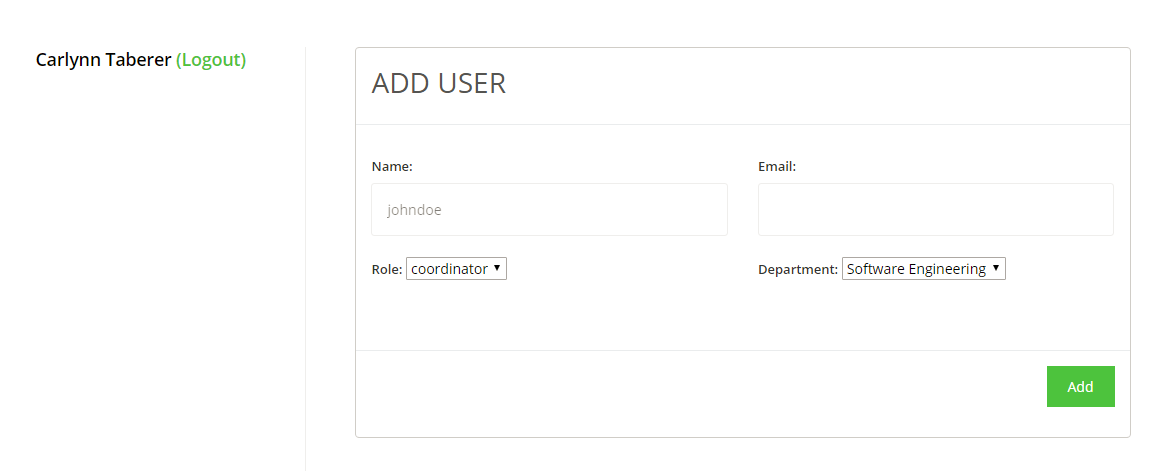
**Login Page**

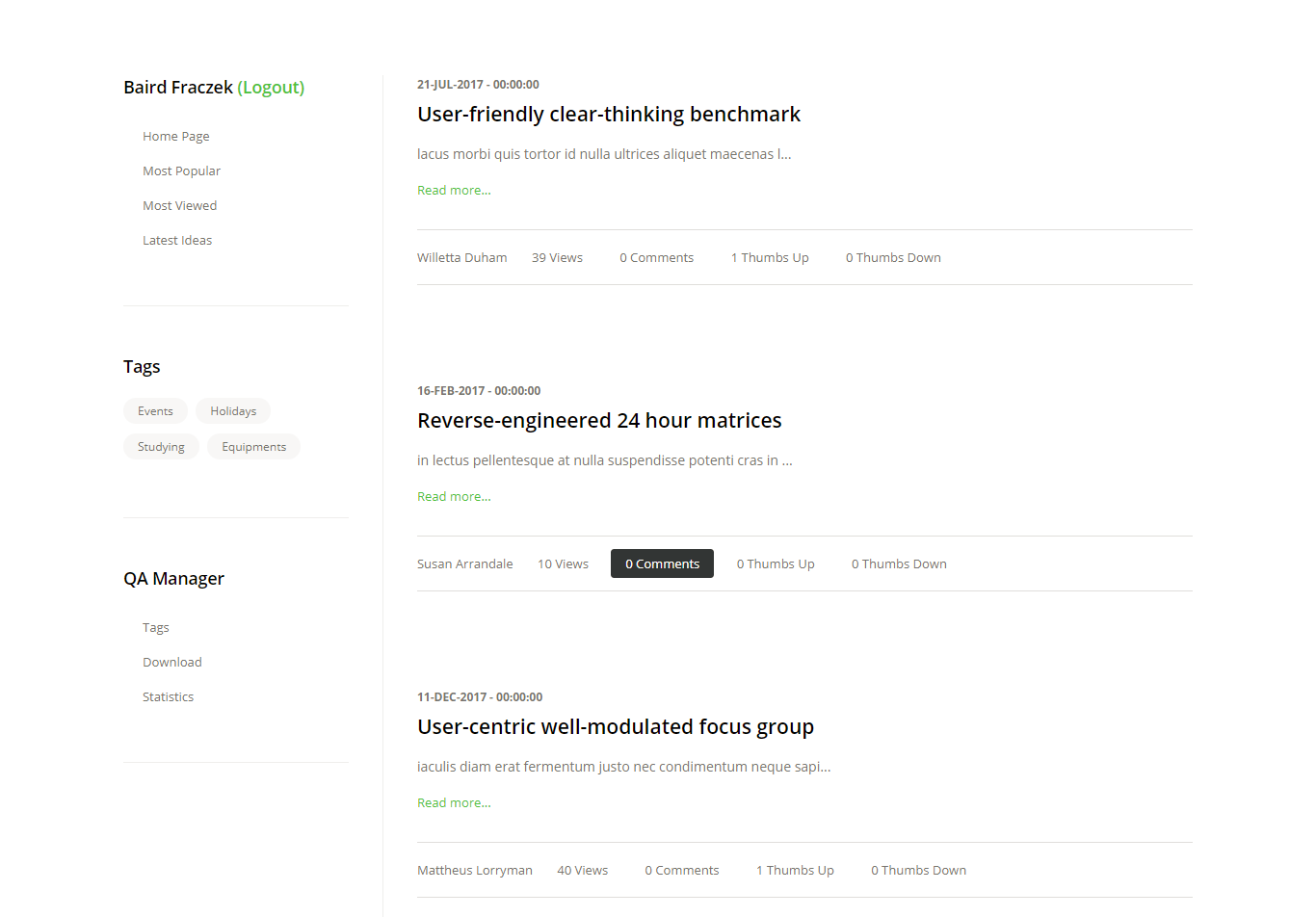


**Admin Homepage**

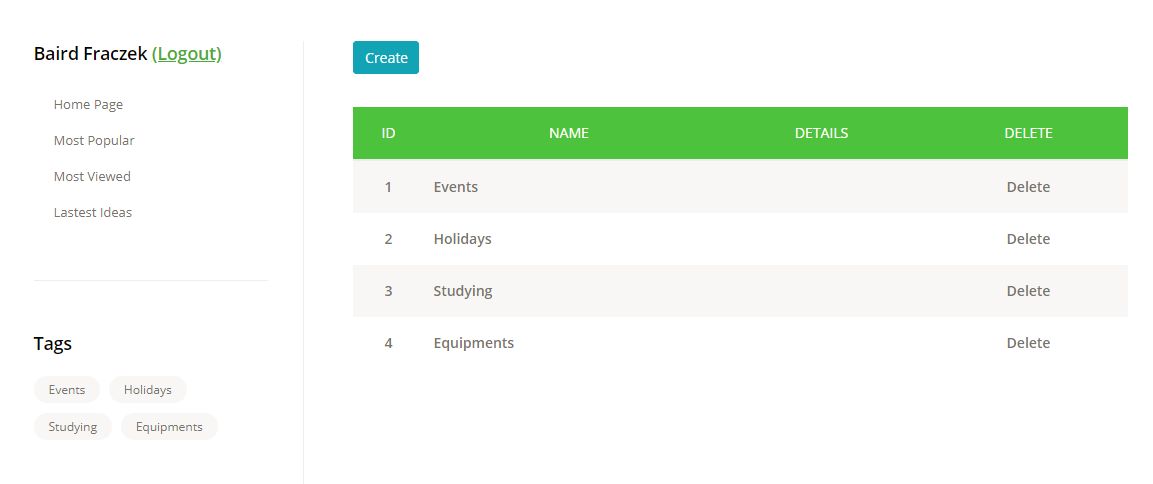


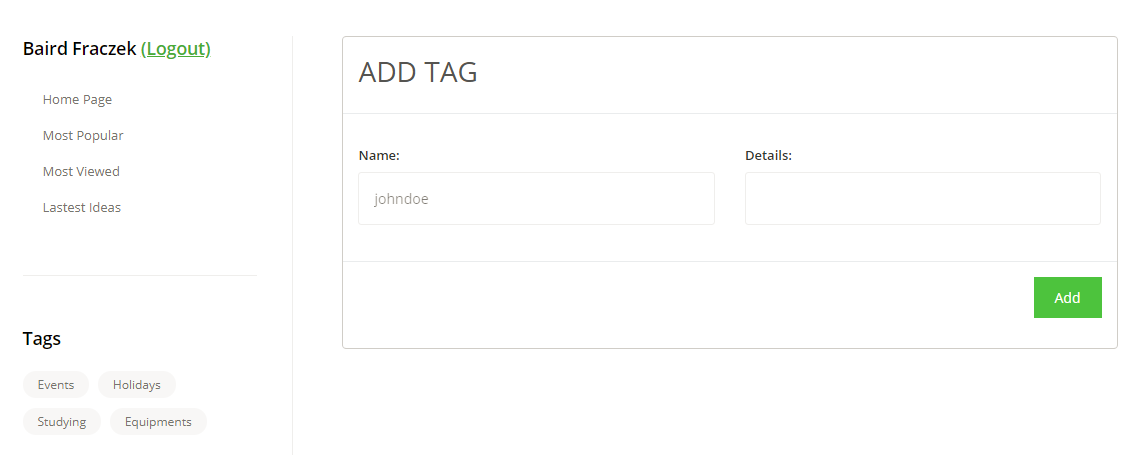
**Add New User Page**



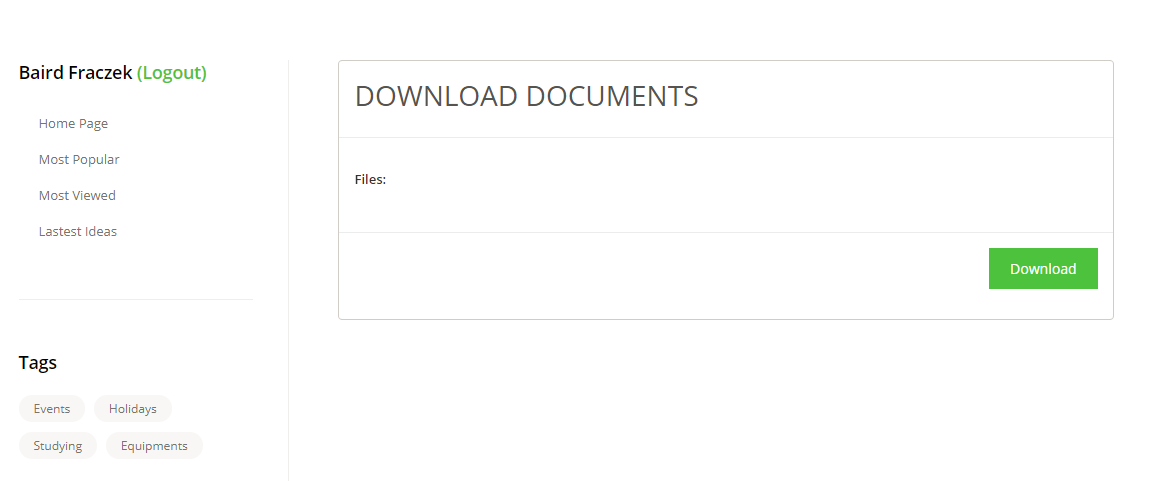
**Home Page**

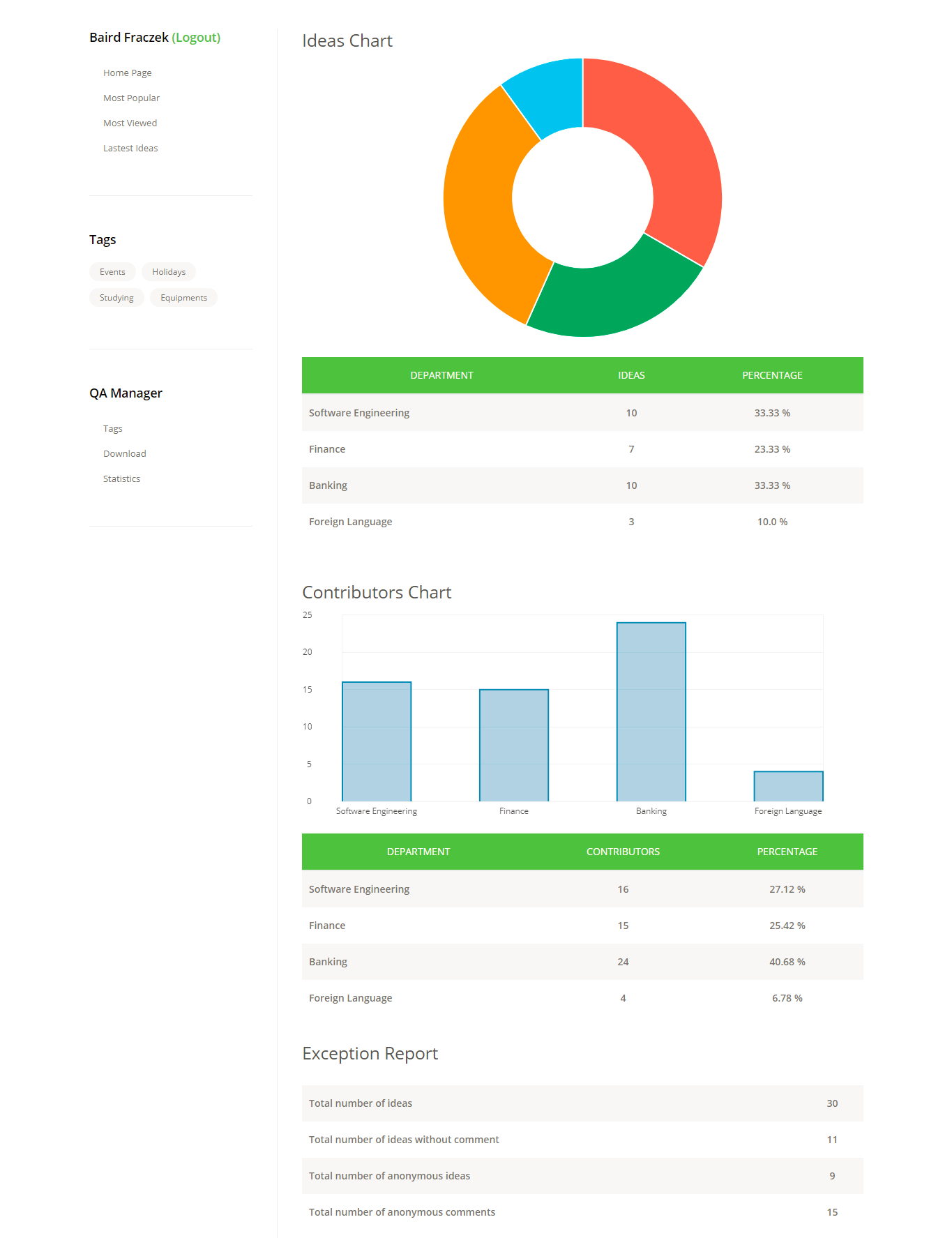
**Tags Homepage**



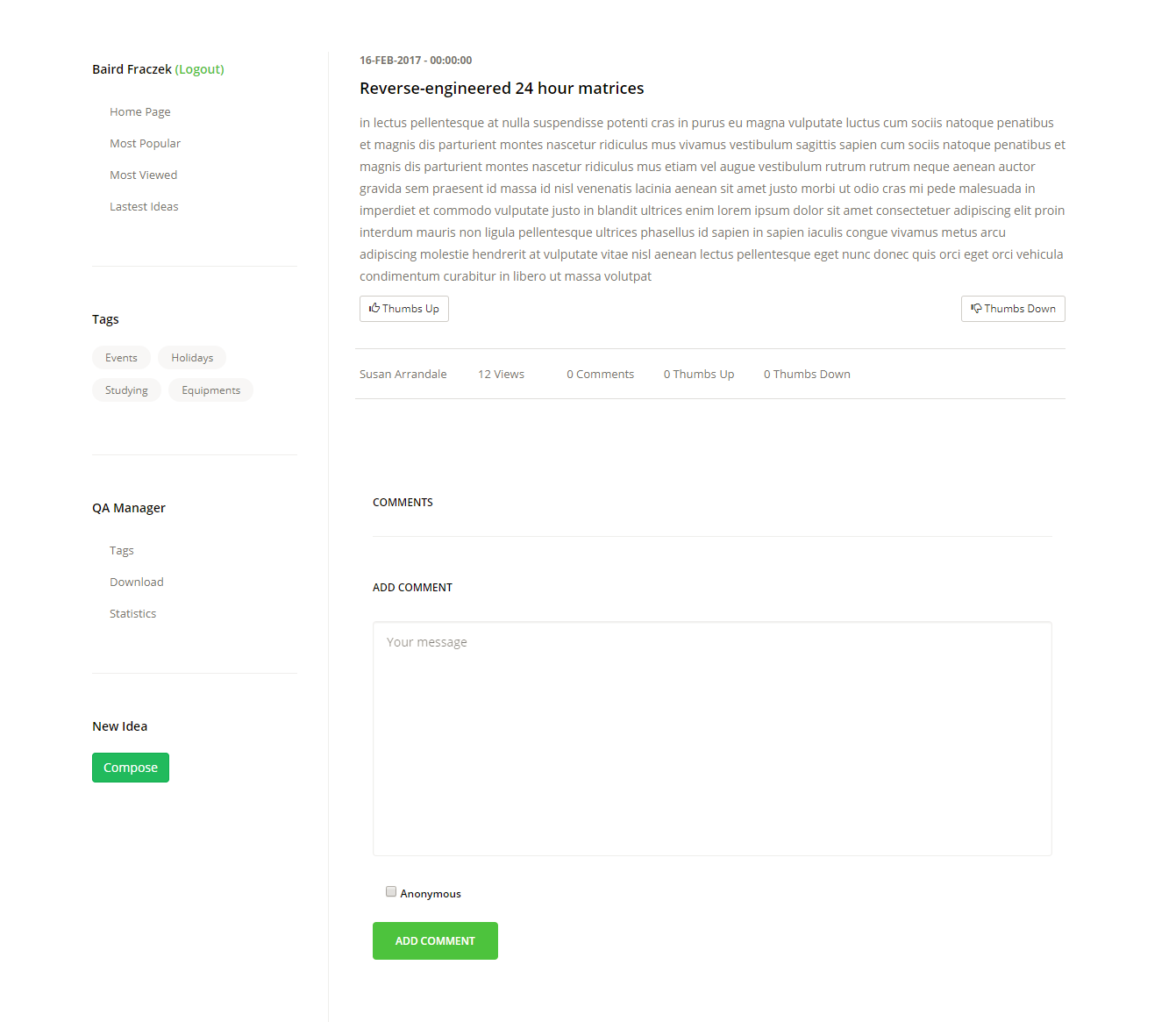
**Add New Tag Page**

**Download File Page**

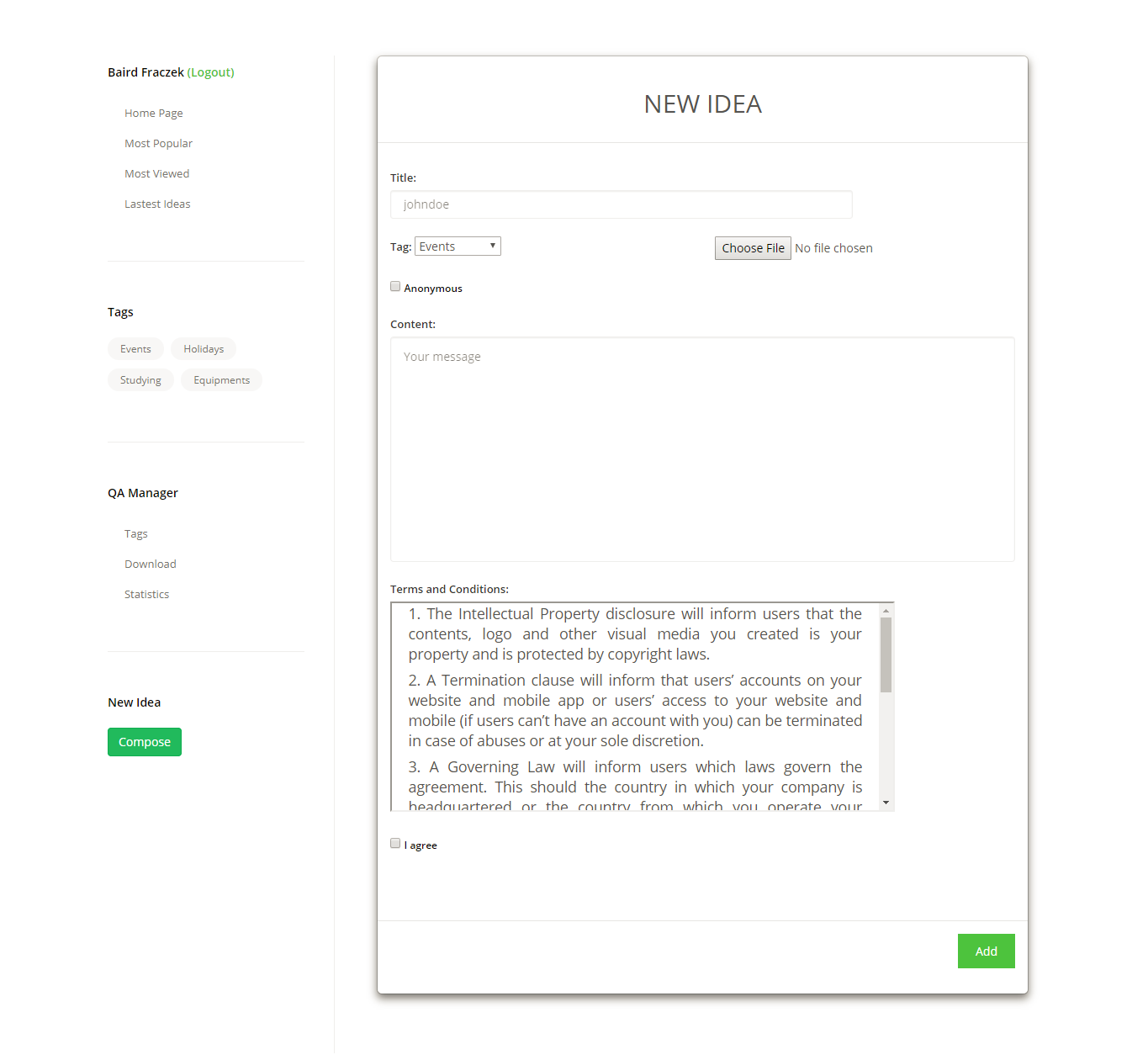


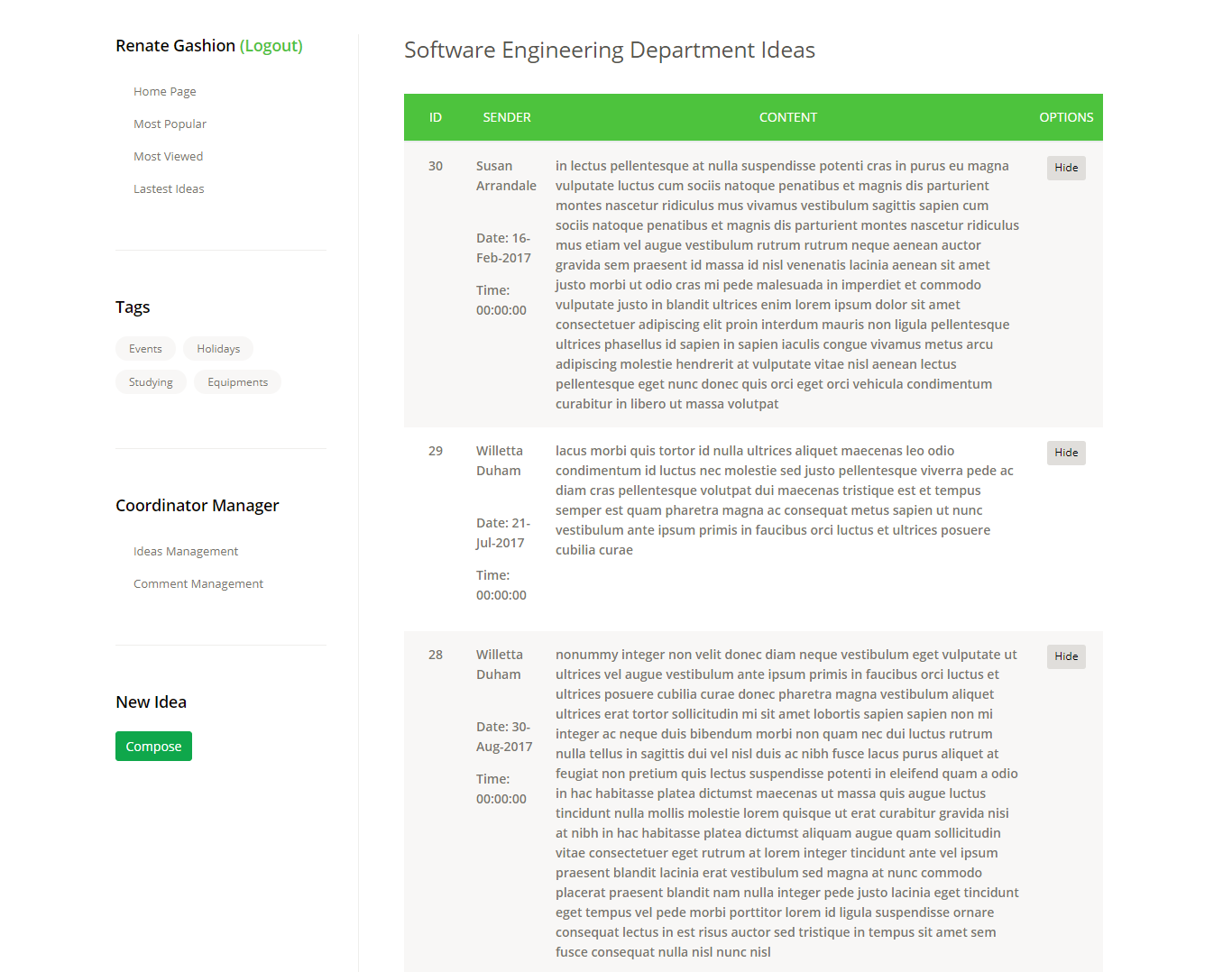
**Statistics Page**

**View One Idea Page**

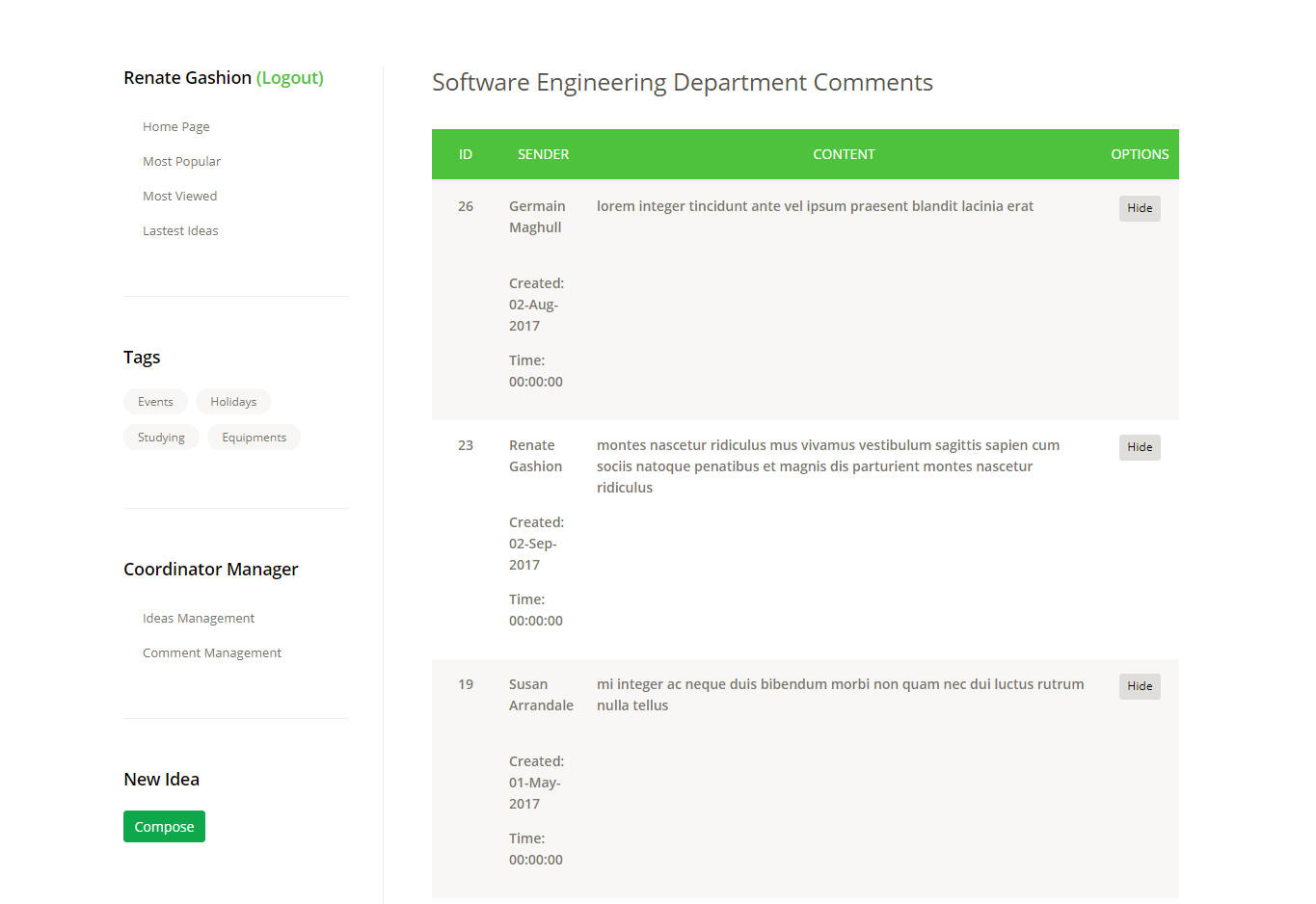


**Add New Idea Page**



**Show Hide Idea Page**

**Show Hide Comment Page**

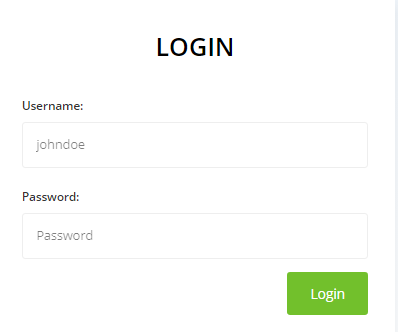


1. Testing
2. 1. Test Plan

|  |
| --- |
| **1. Scope of The Tests**  1.1 The Blog Application version 1.0  1.2 The documents: Requirement Document, Analysis Document, Coding Document, Database Document, Development Document  1.3 Testing time: from March 23rd 2018 to April 8th 2017 |
| **2. Testing Environment:**  2.1 Testing app: All the functions developed  2.2 Software   |  |  | | --- | --- | | Documentation tool | Microsoft word 2013 | | Scheduling tool | Microsoft project 2013 | | DBMS | Microsoft SQL Server 2012 | | Operating System | Windows 8.1, 10, Linux | | IDE | Ecllipse IDE |   2.3 Resources: Colleagues in group |
| **3. Test Detail (For Each Test):**  3.1 Test identification: Test features  3.2 Test objective: All of the requirement function  3.3 Cross- reference to relevant design document and the requirement document: Requirement Document, Analysis Document, Database Document, Development Document  3.4 Test requirements:   * Login. * Compose idea. * Payment * Report. * Checking all kinks and buttons. * Animations and modals work, correct spelling and grammar. * Correct responsive appearance on mobile, tablet and desktop. * Correct content.   3.5 Test level (unit, integration or system tests)   1. Unit: States, transaction, functionally 2. Integration: Error handling, functionally, performance 3. System: States, Transaction, functionally   3.6 Special requirements: Measurements of response times, Security requirements: Fast response time and good performance, all the requirement met the standard of security  3.7 Data to be recorded: Yes |
| **4. Test Schedule (For Each Test or Test Group) including time estimates for the following:**  4.1 Preparation: 5 days  4.2 Testing: 8 days  4.3 Error correction: 3 days  4.4 Regression test: 4 days |

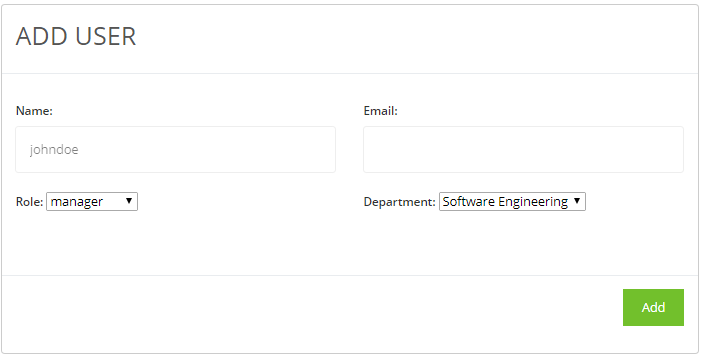
* 1. Test case
     1. Login Test case:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Scenario** | **Test Case** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| Check Login Functionality | Check response on entering valid username and password | 1. Launch the website 2. Enter username 3. Enter password 4. Click “Login” button | Username: admin  Password: admin | The website will be redirect by the role of user. | Redirect to Admin page. | Pass |



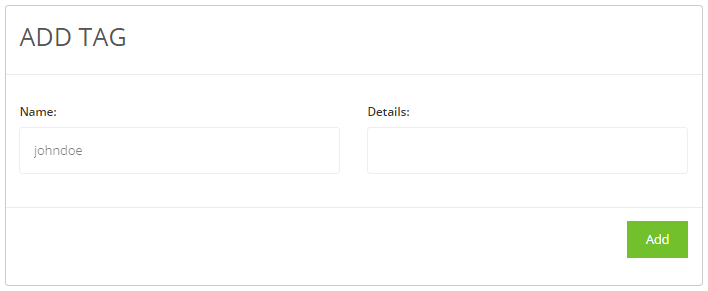
* + 1. Admin - Create user account

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Scenario** | **Test Case** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| Check Create new account functionality | Check create user function with valid information | 1. Login with admin account 2. Select button Create 3. Enter name / email 4. Select role / department 5. Click Add button | (‘Doan Dinh Huy’, ‘huydoan@gmail.com’, ‘manager’, ‘software engineering’) | A new user account is created. | A new user account is created. | Pass |



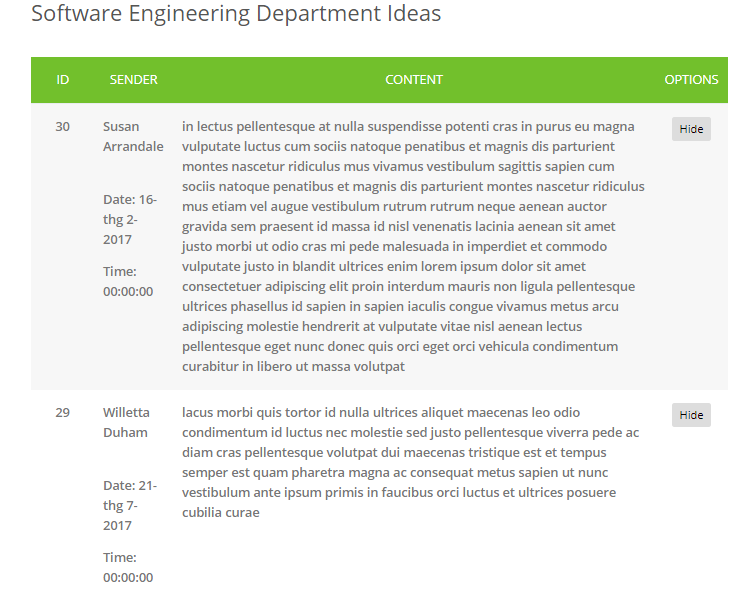
* + 1. Manager – Create new Tag

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Scenario** | **Test Case** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| Check Create new tag functionality | Check create tag function with valid information | 1. Login with manager account 2. Click button Create 3. Enter name of tag 4. Enter detail of tag 5. Click Add button | (‘Science’, ‘ ’) | A new tag is created. | A new tag is created. | Pass |



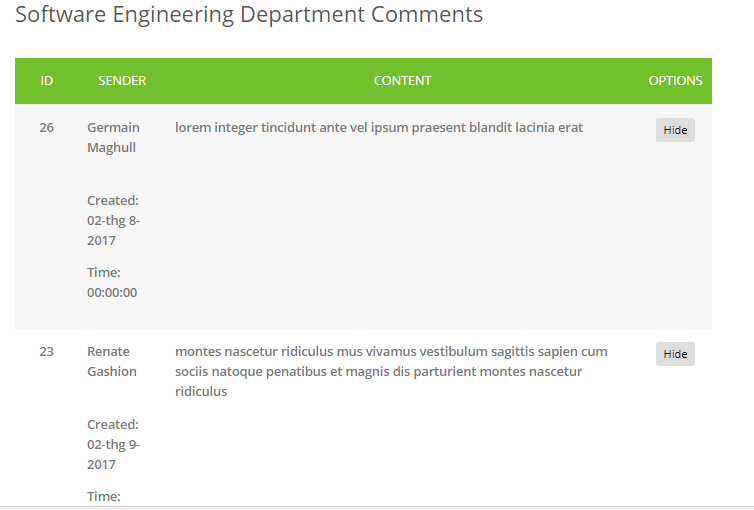
* + 1. Coordinator – Show/Hide Ideas

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Scenario** | **Test Case** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| Check Show/Hide Ideas functionality | Show/Hide ideas existed in database. | 1. Login with coordinator account 2. Click Ideas Management link 3. Click button Hide to hide the idea. 4. Click button Show to show the idea. |  | If the idea is hidden, it will be shown. | If the idea is hidden, it will be shown. | Pass |



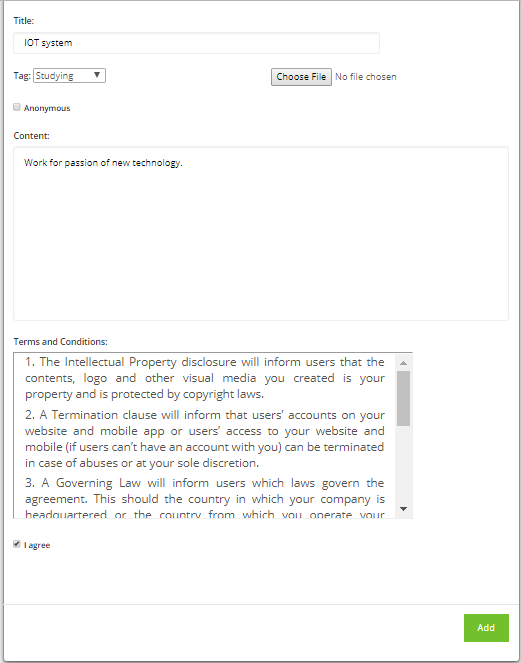
* + 1. Coordinator – Show/Hide Comments

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Scenario** | **Test Case** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| Check Show/Hide comments functionality | Show/Hide comments existed in database. | 1. Login with coordinator account 2. Click Comments Management link 3. Click button Hide to hide the comment. 4. Click button Show to show the comment. |  | If the comment is hidden, it will be shown. | If the comment is hidden, it will be shown. | Pass |



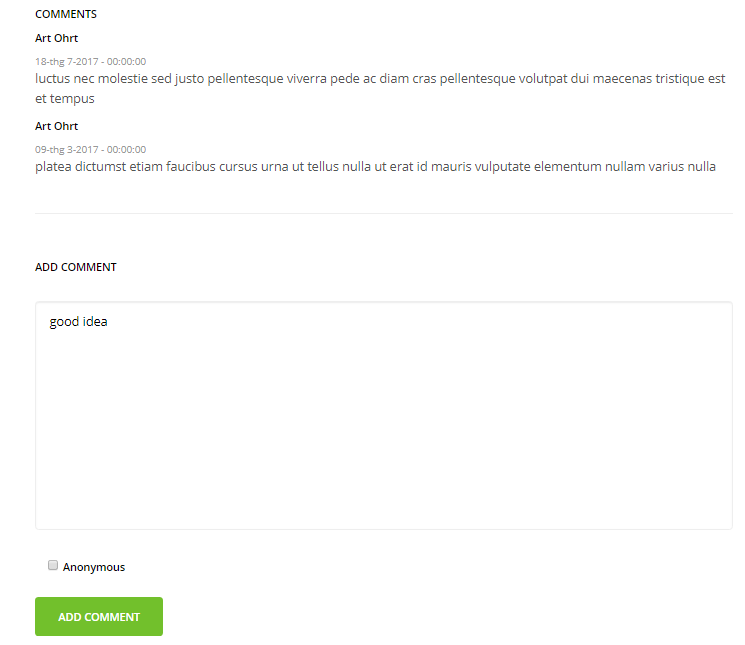
* + 1. Student – Add new idea

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Scenario** | **Test Case** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| Check Add new idea functionality | Add new idea with valid information. | 1. Login with student account 2. Click Compose button 3. Fill all fields with valid inputs. 4. Check the Terms. 5. Click Add button | (‘IOT system, ‘Studying, ‘iot.pdf, ‘Work for passion of new technology’) | A new idea is stored in database. | A new idea is stored in database. | Pass |



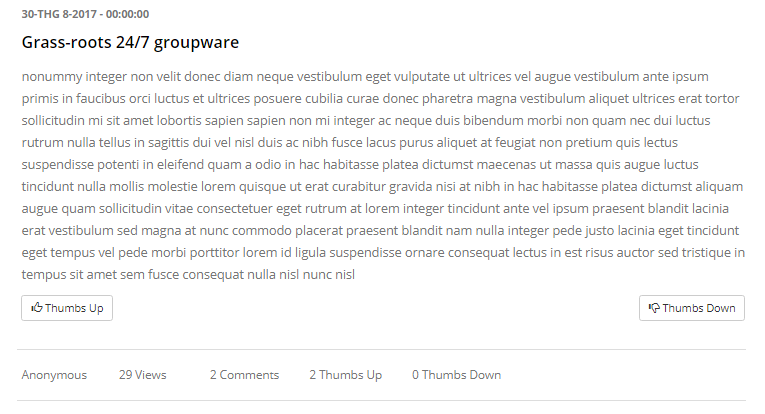
* + 1. Student/Manager/Coordinator – Add comment

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Scenario** | **Test Case** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| Check Add new comment functionality | Add new comment for an idea. | 1. Login with student/manager/coordinator account 2. Click ‘Read more’ on an idea. 3. Input comment on Comment field. 4. Click ‘Add Comment’ button. | (‘That is a good idea.’) | A new comment of idea is stored in database. | A new comment of idea is stored in database. | Pass |



* + 1. Student/Manager/Coordinator - Give the Thumbs Up or Thumbs Down for any idea

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Scenario** | **Test Case** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| Check Add new comment functionality | Add new comment for an idea. | 1. Login with student/manager/coordinator account 2. Click ‘Read more’ on an idea. 3. Click ‘Thumbs Up’ or ‘Thumbs Down’ button. |  | The count of Thumbs Up or Thumbs Down will increase if user clicked. | The count of Thumbs Up or Thumbs Down increases when users click button. | Pass |



* 1. Test Log

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TEST CASE | PASS | PARTLY PASS | FAIL | NOT TESTED |
| 5.2.1 | 5 | 3 | 0 | 2 |
| 5.2.2 | 6 | 3 | 1 | 0 |
| 5.2.3 | 6 | 2 | 1 | 1 |
| 5.2.4 | 8 | 2 | 0 | 0 |
| 5.2.5 | 5 | 2 | 1 | 2 |
| 5.2.6 | 6 | 4 | 0 | 0 |
| 5.2.7 | 9 | 1 | 0 | 0 |
| 5.2.8 | 5 | 3 | 1 | 1 |
| TOTAL | 50 | 20 | 4 | 6 |

1. Evaluation
2. 1. Assumption

* Students can login and post one or more ideas.
* Students can upload documents to support their ideas.
* Ideas and comments can be posted anonymously if the authors want.
* Student and other users can comment on submitted ideas and give the Thumbs up/Thumps down for any ideas.
* The QA Managers can add new categories of idea.
* The Coordinator can manage the status of Ideas and Comments.
* When an idea is submitted, the system emails a notification to the Coordinator.
* Users can view ideas grouped by Most popular, Most Viewed, Lasted, Lasted Comments.
* The Manager can downloads all zip files posted by Students after the final closure date.
* The system has an available Statistical analysis.
* The author of each idea will receive an automatic email whenever a comment is submitted to any of their ideas by a student.
  1. System Weakness
* The form validation is so simple.
* The user interface is not so attractive because the design is simple.
* The system is not built to work with big data.
  1. System Strength
* The system meets all requirements of the coursework.
* The design is simple so the user will feel easy to use the system.
* The system has a role-based login for each functionality.
  1. Learning Outcomes

By applying Agile methodology, we have learned each other the way how to manage time and resources in system development. Sprint meetings weekly can help us to increase project control and know clearly about the errors when it occurs.

Agile is a powerful tool for software development, not only providing benefits to the development team, but also providing a number of important business benefits to the client. Agile helps project teams deal with many of the most common project pitfalls (such as cost, schedule predictability and scope creep) in a more controlled manner. By reorganizing and re-envisioning the activities involved in custom software development, Agile achieves those same objectives in a leaner and more business-focused way.

1. *The Spring Framework is an*[*application framework*](https://en.wikipedia.org/wiki/Application_framework)*and*[*inversion of control*](https://en.wikipedia.org/wiki/Inversion_of_control)[*container*](https://en.wikipedia.org/wiki/Servlet_container)*for the*[*Java platform*](https://en.wikipedia.org/wiki/Java_platform)*.* [↑](#footnote-ref-1)