COMP1640 - TCH1602

Coursework Report (Group Report)

Group 7

Team members – ID:

Nguyễn Việt Sơn - GCH15115

Nguyễn Đăng Duy - GCH15343

Nguyễn Mạnh Tuấn - GCH15143

Vũ Trọng Nghĩa - GCH16110

Contents

[Introduction 2](#_Toc7408568)

[Database 2](#_Toc7408569)

[Authentication security 2](#_Toc7408570)

[Referential integrity 3](#_Toc7408571)

[ERD and datatypes 4](#_Toc7408572)

[Validation of input data 6](#_Toc7408573)

[Site design 7](#_Toc7408574)

[Interface design 7](#_Toc7408575)

[Responsive design 8](#_Toc7408576)

[Flat design 13](#_Toc7408577)

[Color choice 13](#_Toc7408578)

[Organization of content 16](#_Toc7408579)

[Navigation within the site 16](#_Toc7408580)

[Usability and Accessibility criteria 17](#_Toc7408581)

[Functionalities 18](#_Toc7408582)

[Role-based menu 18](#_Toc7408583)

[Testing 27](#_Toc7408584)

[Unit testing and Integration testing 27](#_Toc7408585)

[Functional testing 27](#_Toc7408586)

[System testing 30](#_Toc7408587)

[Agile methods followed 30](#_Toc7408588)

[Burn down charts 30](#_Toc7408589)

[Minutes of meetings 30](#_Toc7408590)

[User stories 32](#_Toc7408591)

[Product backlogs 32](#_Toc7408592)

[Sprints 34](#_Toc7408593)

[Screencast and Presentation 42](#_Toc7408594)

# Introduction

We are group AG7 of class TCH1602. This is the report for our final product of our group project for the subject COMP1640 – Enterprise Web Software Development.

Our final product is a website called <http://g7-web.herokuapp.com>

Our website provides basic functionalities for the management of our university magazine. It allows students, faculty coordinator, and magazine administrators to log in, upload articles, manage user accounts and articles, and other essential tasks of managing a magazine.

The source code for our product has been uploaded to this repository: <https://github.com/ngdangduy13/enterprise-web-development>

This report will present the following components of our product and process:

1. Database: the chosen database to store the magazine’s data.
2. Site design: elements of our website’s design.
3. Functionality: the functionalities provided by the website to various user roles.
4. Testing: test plans and results.
5. Agile methods followed: artefacts of our project that demonstrate how we followed Scrum, the Agile methodology chosen for the development of this website.
6. Screencast and presentation: a screen video demonstrating the website’s basic functionalities, and presentation slides for a formal presentation.

Database

The database selected for this project is Firebase, a light-weight, NoSQL, JSON-based service provided by Google.

## Authentication security

The authentication functionality for logging in is already implemented by Google authentication. We chose Email/Password to be the authentication method of this website.

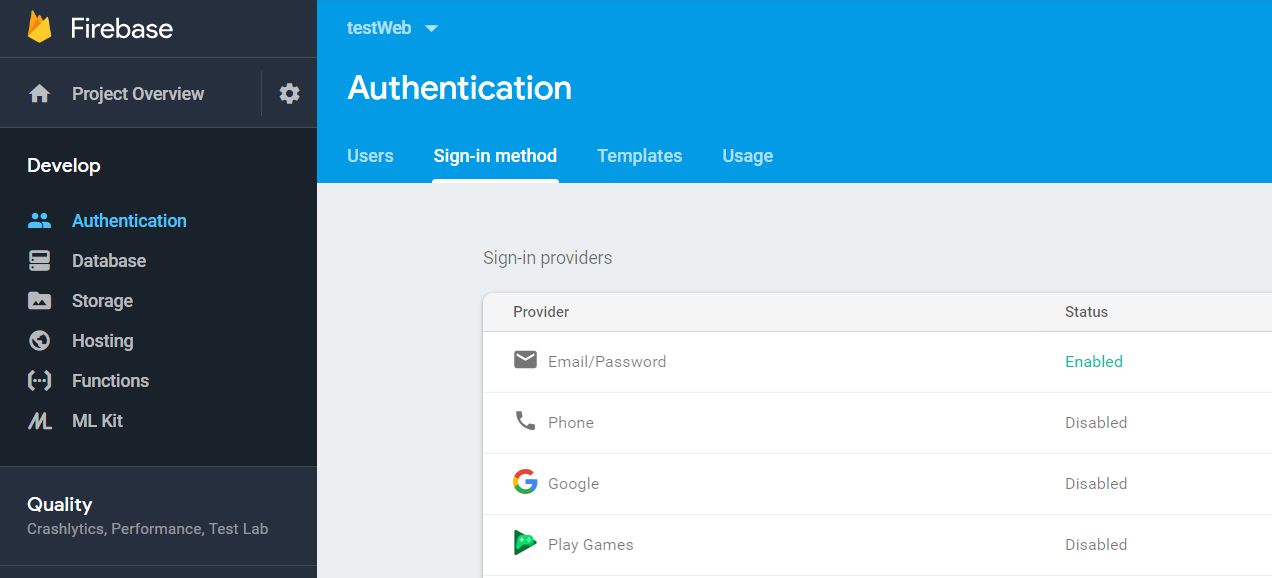


Figure 1 Firebase's authentication by Email/Password

The features for role-based access control is not implemented in the database in this project. It is instead implemented in our main program using a JavaScript framework called Node.js.

User of a role can only access functionalities provided for that role. Trying to overstep another role’s permission will be error and the system will show the message “Unauthorized”.

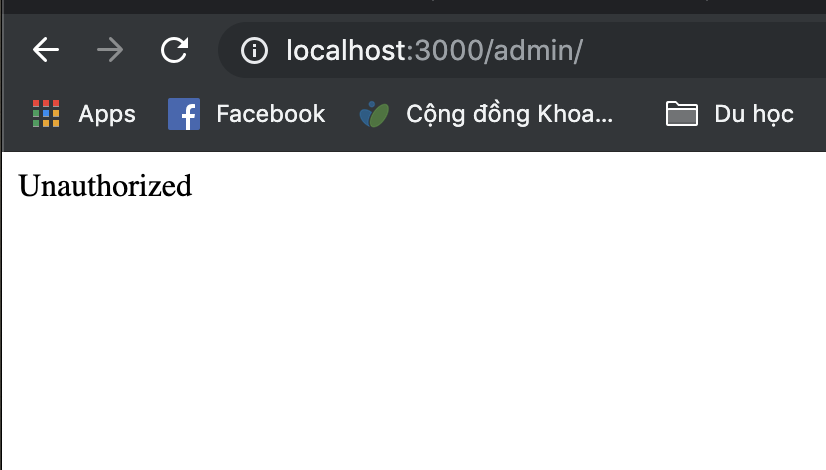


Figure 2 The system show message “Unauthorized” when he tries to access a webpage only authorized for students

## Referential integrity

One disadvantage of Firebase is that it does not have and implemented Referential Integrity protection. For now, our website enforces Referential Integrity also by React through functional requirements in the user interface. Specifically, the site enforces that:

1. Users do not have any access to any interface that may delete or modify the Unique IDs of entities that are referenced by other entities. We simply do not implement those interfaces.
2. When the user attempts to insert a row into the database, and that row has a foreign key, then the user must select a valid value for that foreign key before the request to insert is sent to the server.

For example: The entity User has an optional foreign key *facultyId* pointing to the table Faculties. To ensure referential integrity, when a new user account is created, the choice of faculty can only be selected from a drop-down box that lists out available faculty, as demonstrated in the following picture.

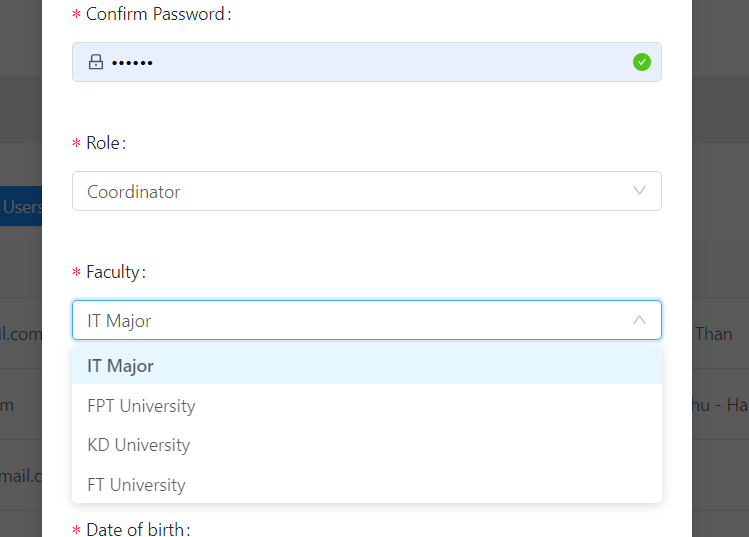


Figure 3 The interface ensures that only valid faculty can be selected for a new user account

## ERD and datatypes

Following is the ERD for the database:

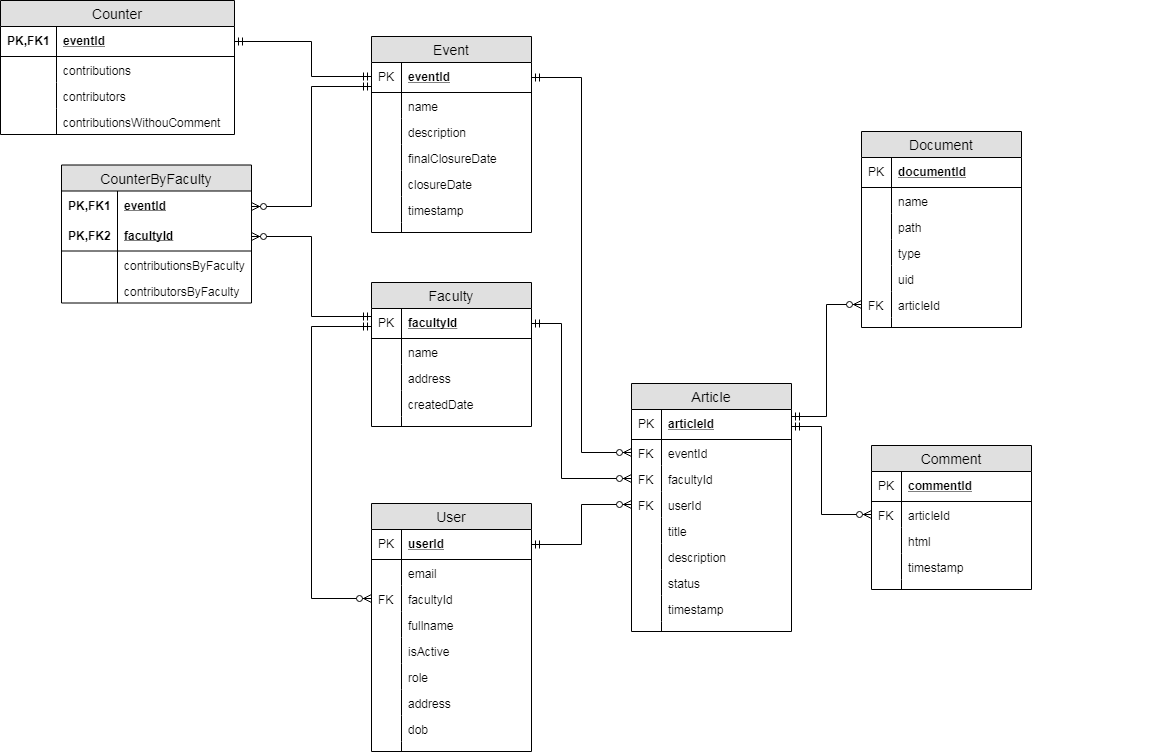


Figure 4 ERD for the database

All tables are normalized to the 3rd Normal Form. The tables **Faculties** and **Users**, and **Articles** are to store data about the faculties, users (including for roles), and submitted articles. The table **Events** are to store is to store the publication numbers of the magazine (e.g.: magazine publication year 2020 is considered an ‘event’).

For statistical purposes, the number of contributions (submissions) and student contributors are stored in the table **Counter**, whose values are adjusted after each submission.

The datatypes in the database are as follows:

* Any field named ‘isActive’ is of type Boolean.
* All fields named ‘timestamp’ are of type Number.
* All non-key fields in the tables Counter and CounterByFaculty are of type Number.
* All other fields are of type String.

Complex data type such as Email or Date are not defined nor validated in the database. Instead, we handle such datatypes from the server-side using Node.js.

## Validation of input data

Validation of input data types is not implemented in the database. Instead, we use React to validate data from the client-side, so that the data must be validated before they can be sent to the server, and the database will only receive validated datatypes.

For example, the following code snippet handles the input item “email” in the input form when adding a new user:

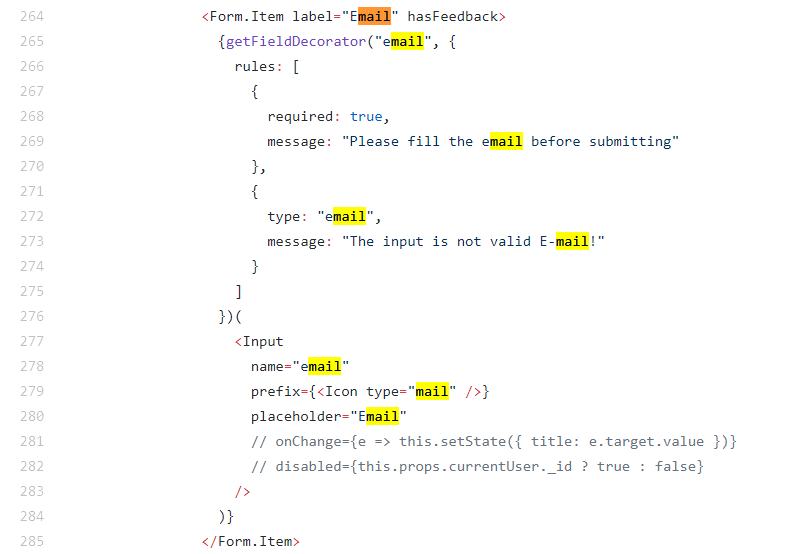


Figure 5 code snippet for inputting a new email

Line 266 to 275 set the validation rules of this input item.

Line 268 **required: true** enforces that the email cannot be empty.

Line 272 **type: “email”** enforces that the input data must be in a valid email format (@ sign followed by domain name)

The messages on lines 269 and 273 inform the users when data are not valid.

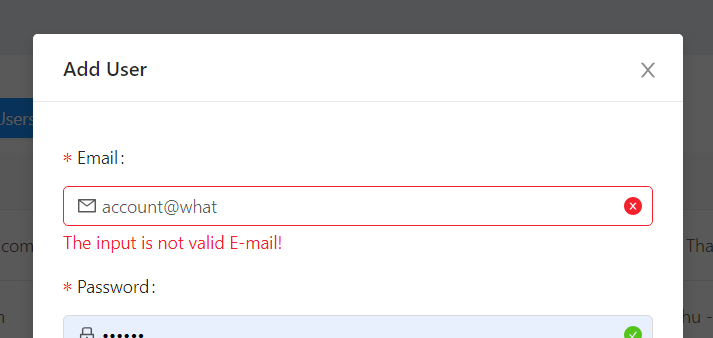


Figure 6 User is informed when the datatype is not valid

Site design

Our website design has clear information architecture, and meets good usability and accessibility criteria. The website is built to be responsive to user’s actions, and to varying screen sizes, including mobile platform sizes.

## Interface design

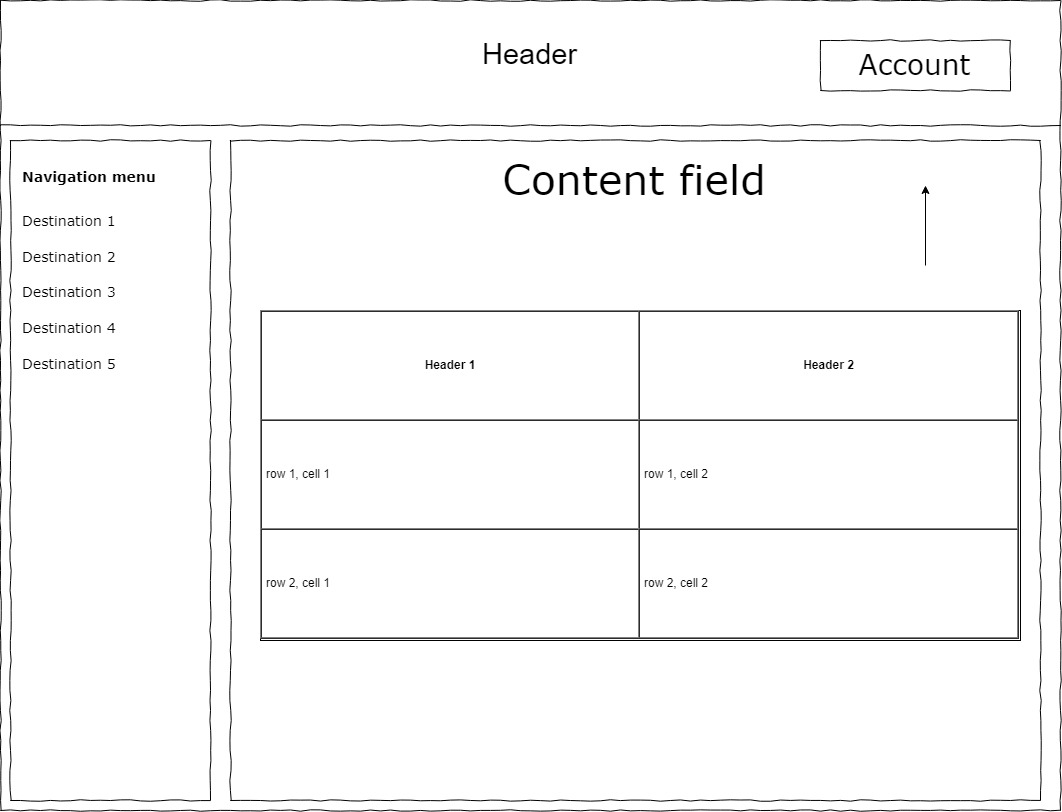


Figure 7 General layout for the website as viewed by Admins, Coordinators and Students

The Header is present on every page of the website.

The Navigation menu is placed inside a collapsible container to the left. The Navigation menu helps users to navigate to various functionalities of the website (e.g.: managing users, submitting articles, …) The destinations on the Navigation menu will vary based on the user’s role.

## Responsive design

The website is designed to respond to user actions, as well as varying screen sizes for multiple platforms.

### Responsiveness to different platforms

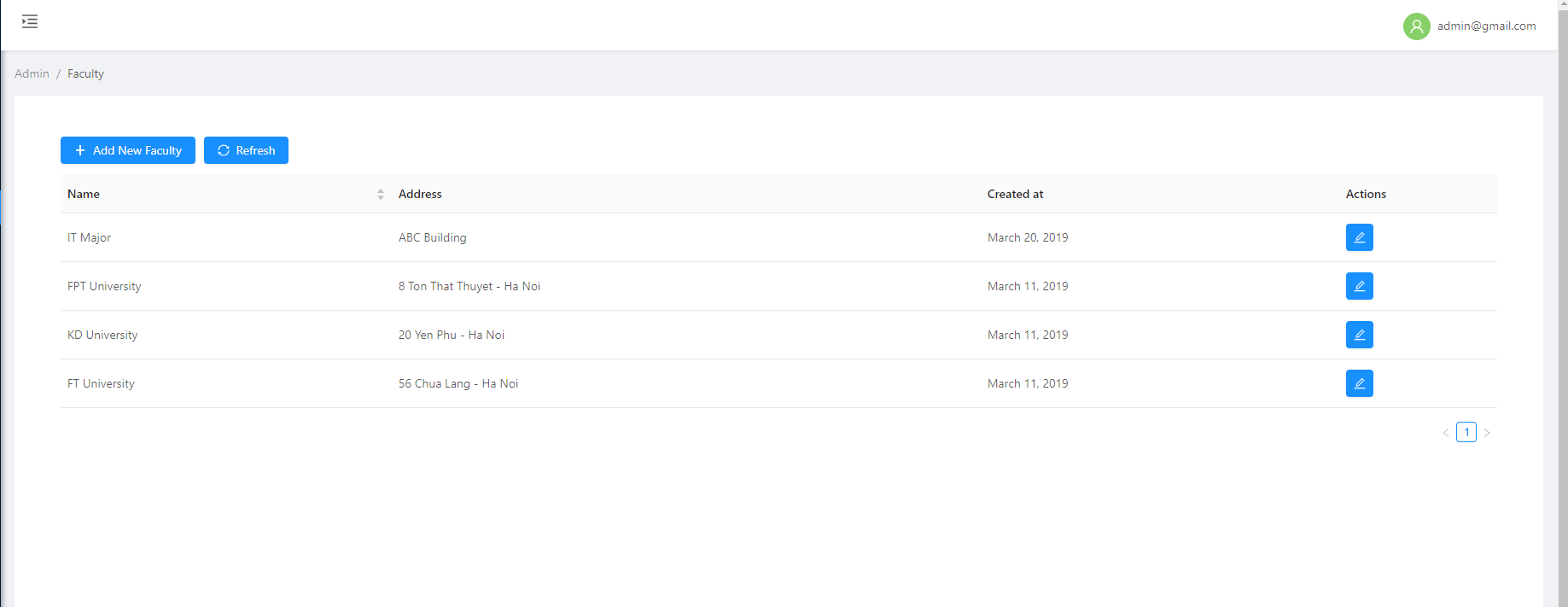


Figure 8 Content is spaced out when the browser is zoomed out

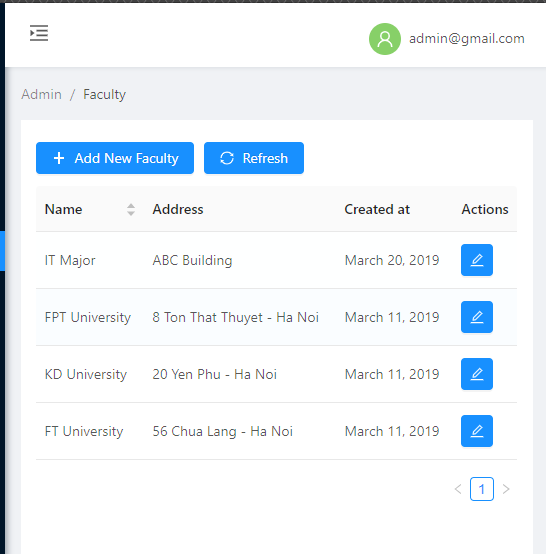


Figure 9 Content is spaced in when zoomed in

The home page for the guest to read published articles can also respond to mobile devices:

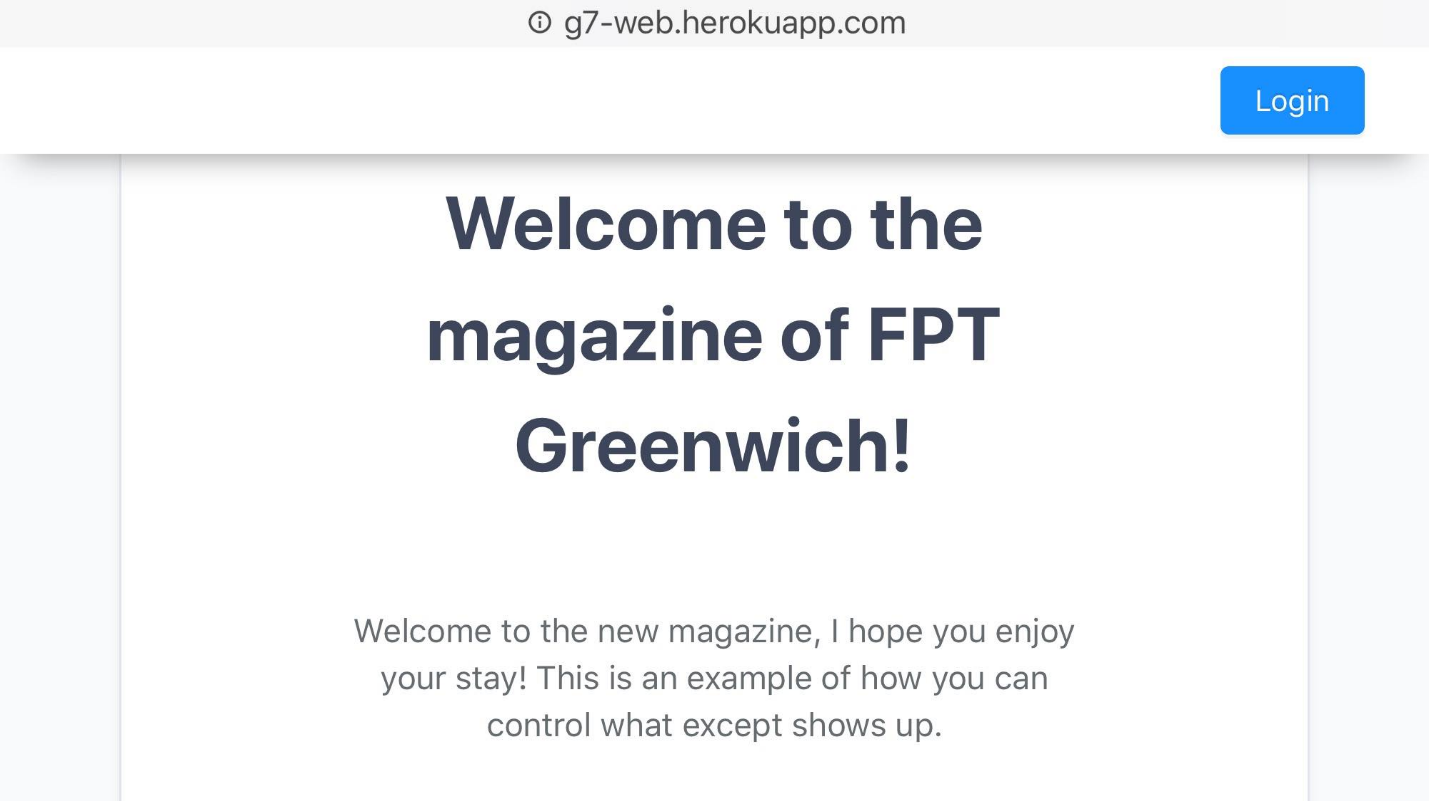


Figure 10 The home page, as viewed on a mobile browser

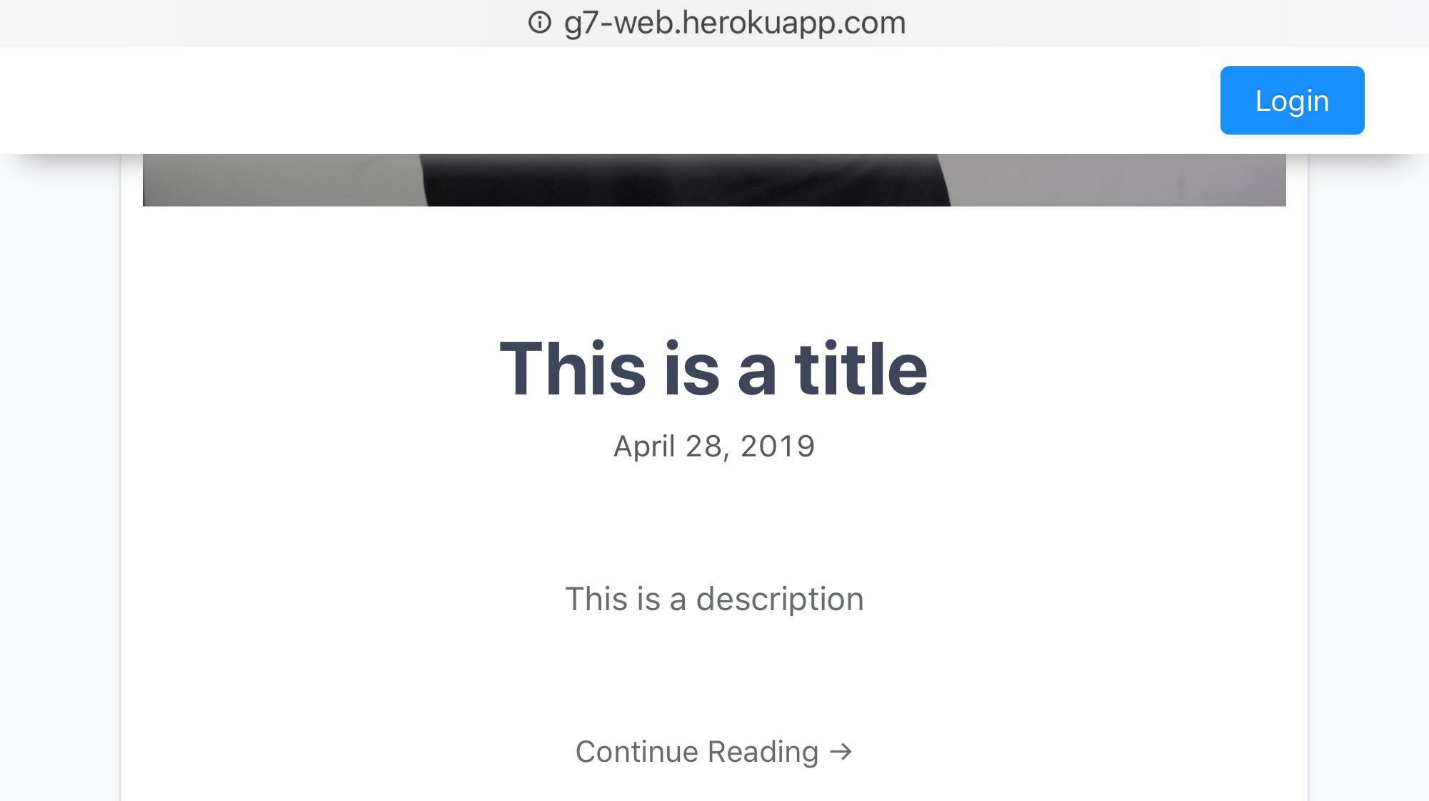


Figure 11 An article on the home page, on a mobile browser

The homepage is also readable on a mobile browser in Portrait mode, but can be improved.

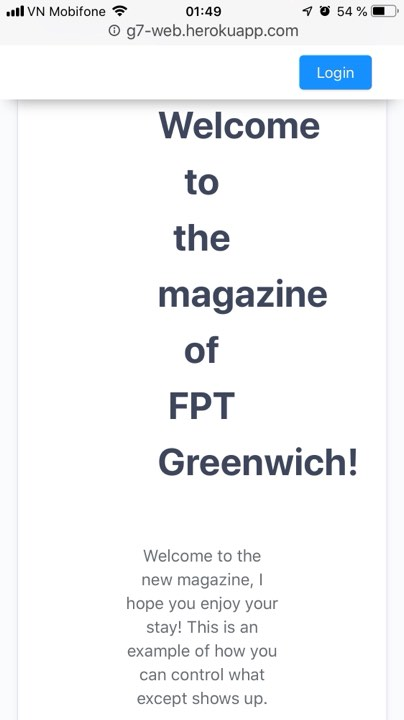


Figure 12 The home page for portrait mode can still be improved

Responsiveness to mobile screen sizes has only been implemented on the homepage. We have not implemented it for other functionalities (managing user, managing articles, …)

### Responsiveness to user’s actions

A thin light-blue progress bar appears at the top of the website to respond to user’s action (loading another page):

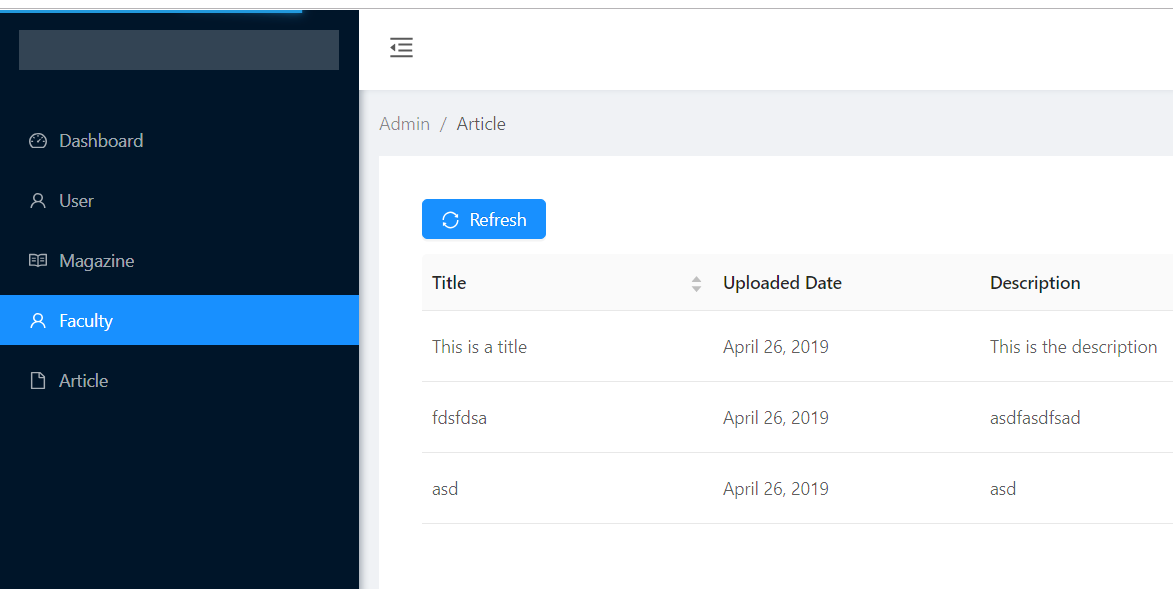


Figure 13 The thin progress bar appears at the top as a page is loading

When user hovers a button, a tooltip appears to clarify the button’s function

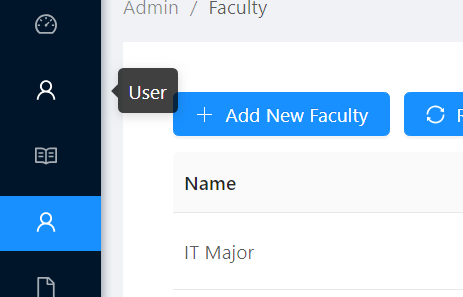


Figure 14 Black tooltip appears when hovering over a button

## Flat design



Figure 15 Our choice of background picture for the Home page

## Color choice

The default color scheme is black text on white background. Buttons are blue with white text.

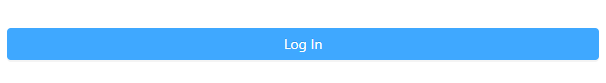


Figure 16 A sample button

Input boxes may be highlighted with a red border:

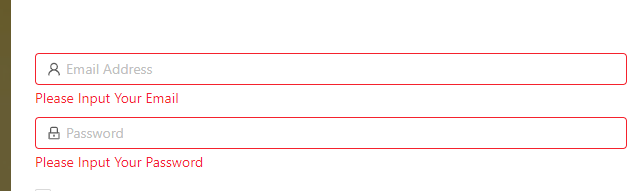


Figure 17 Highlighted textbox for validation purpose

Statuses of entities (users and articles) are suitably color-coded:

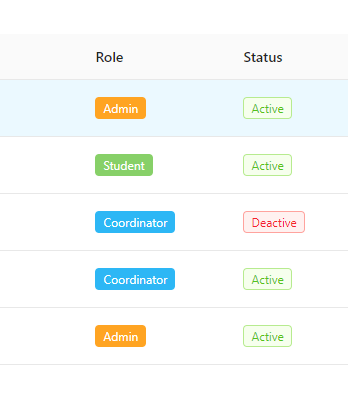


Figure 18 Statuses are color-coded

The menu’s background is colored in contrast with the overall background for an aesthetically pleasing structure.

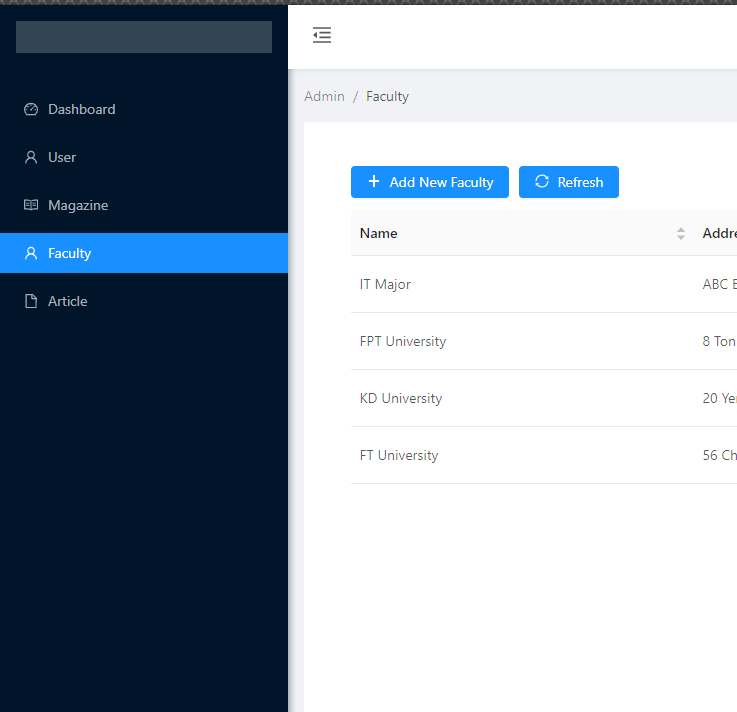


Figure 19 Dark menu in contrast with the white webpage

## Organization of content

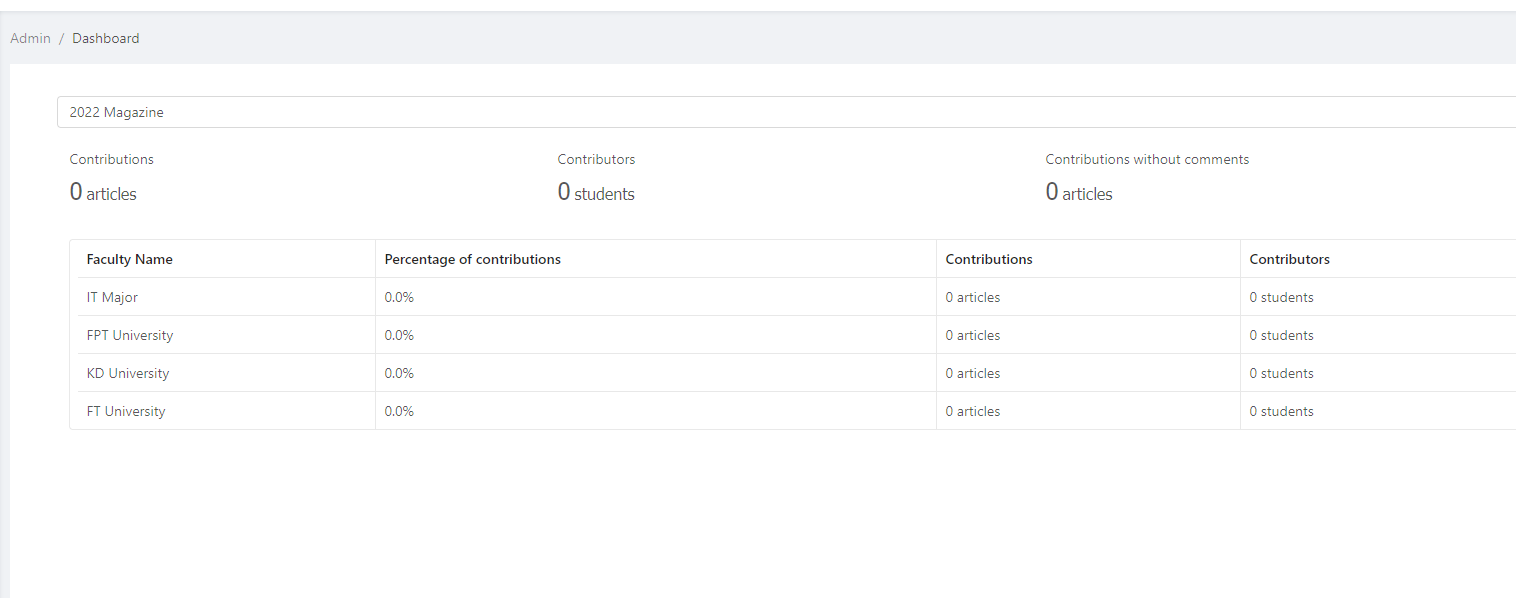


Figure 20 Content is displayed in table

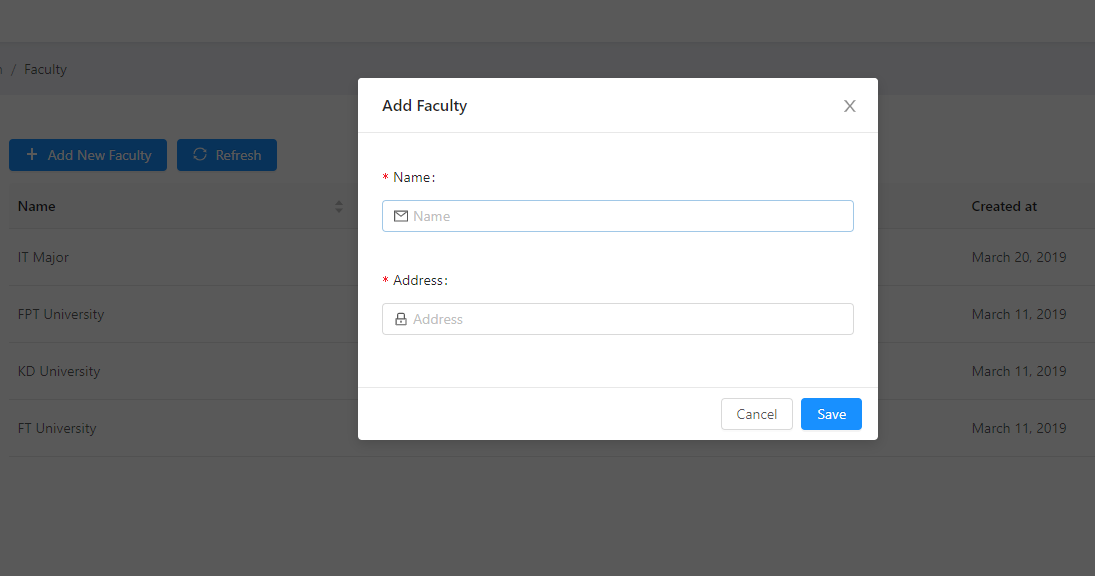


Figure 21 An example of a Form to input data

## Navigation within the site

On the left side of the screen is a collapsible Navigation menu, with buttons that redirect users to various functionalities provided for them.

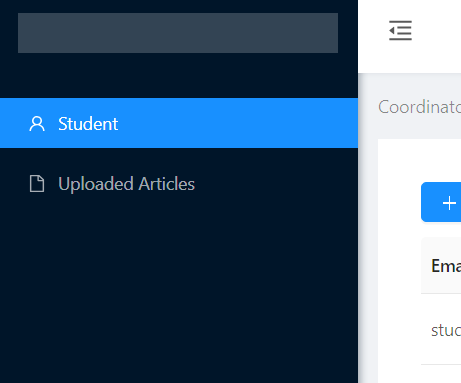


Figure 22 Navigation buttons on a collapsible menu on the left side of the layout

## Usability and Accessibility criteria

The usability and accessibility criteria that out website strives to meet are:

* Clear, consistent language
* Simple, attractive design
* Intuitive signs and symbols
* Title of each page for identification and navigation
* All information available with color and without color specially for context and mark-up language
* All pages are usable when script languages are turned off
* All pages can work well with CSS and without CSS

Besides having intuitive signs and symbols, textless buttons also display a tooltip when mouse-hovered to help user understand them.

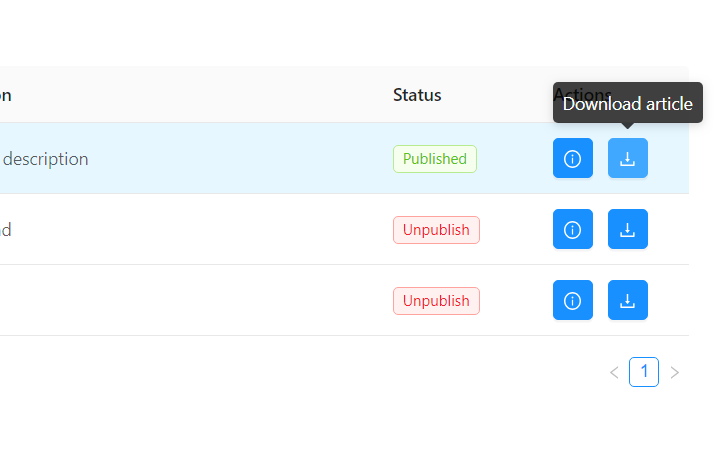


Figure 23 Black tooltip over the 'Download article' button

Functionalities

## Role-based menu

The options the menu on the left responds to user’s role. Below is an example navigation menu for an Administrator:

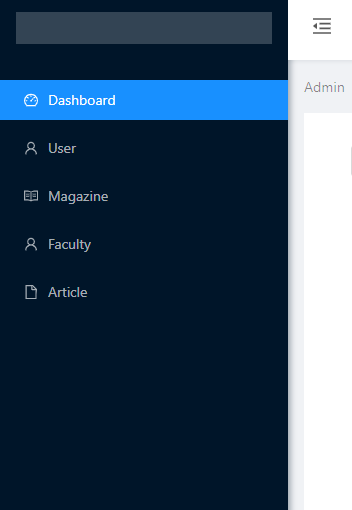


Figure 24 Admin's menu

### Administrator

Administrators are able to manage Users, Magazines, Faculties, and Articles.

Below are the interfaces of each navigation destination on the Admin’s menu:

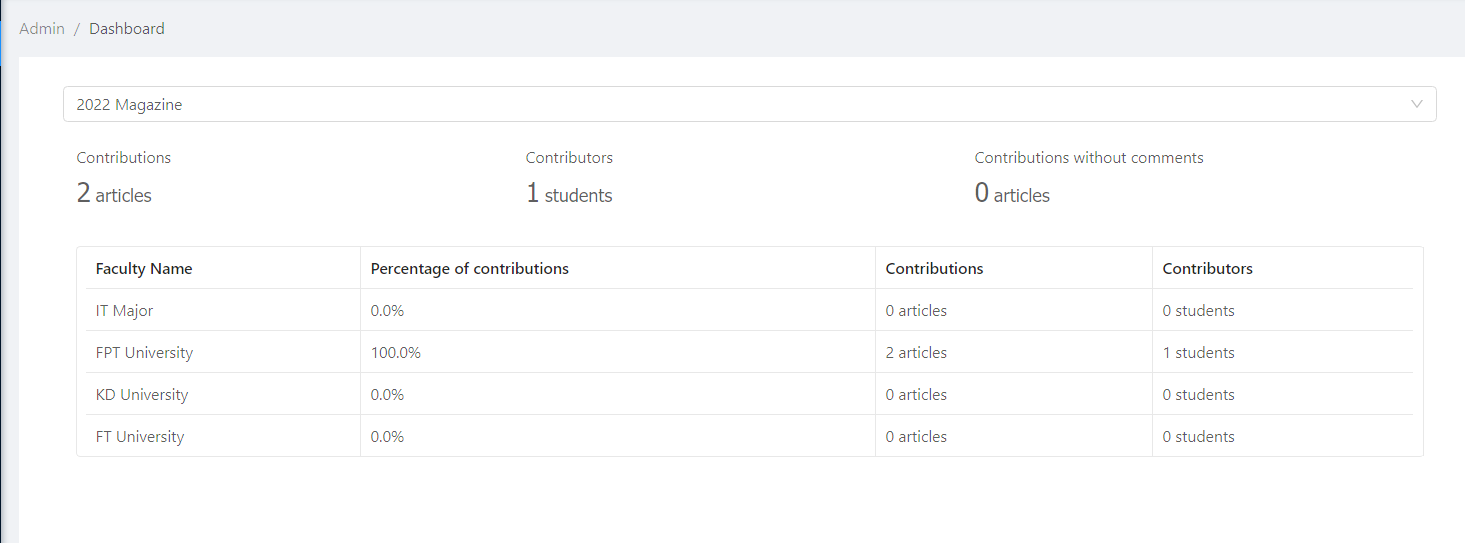


Figure 25 Statistical analysis of contributions

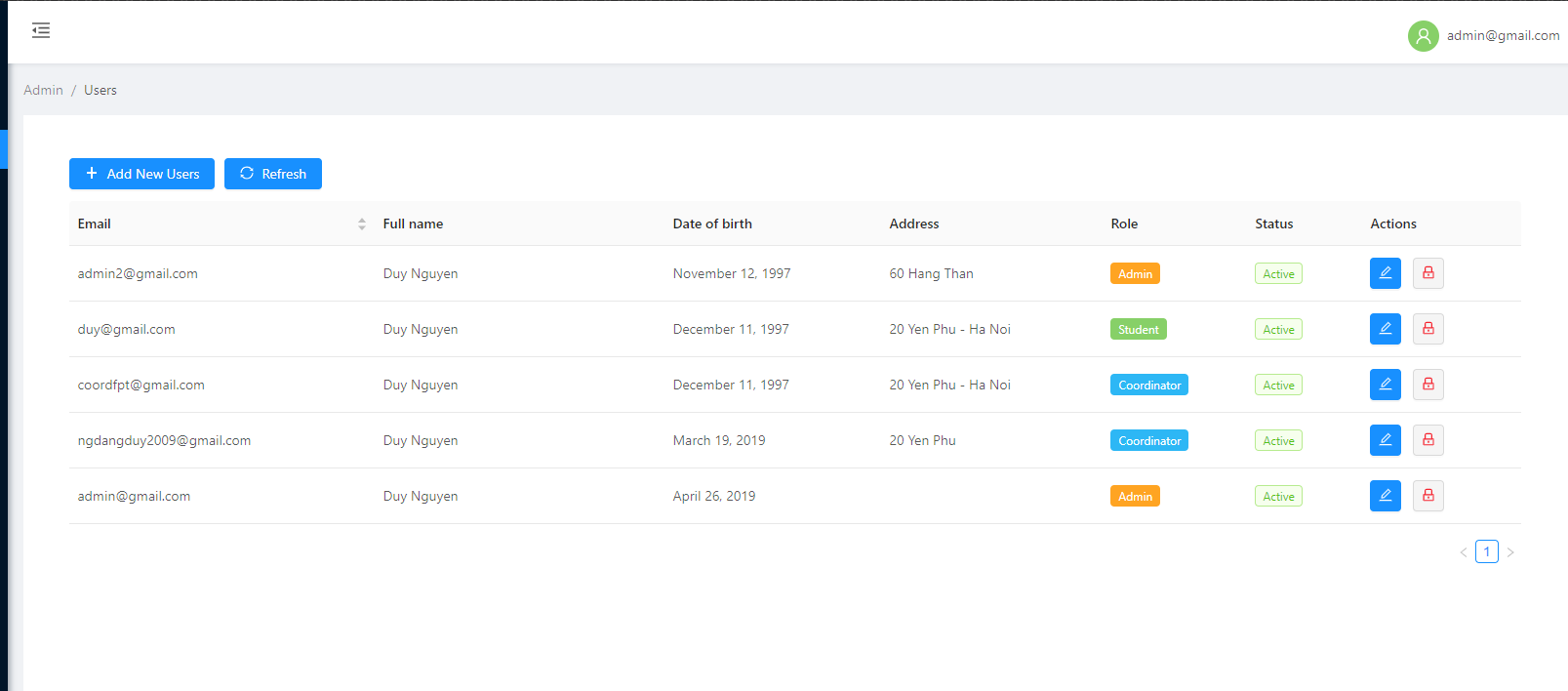


Figure 26 Managing users

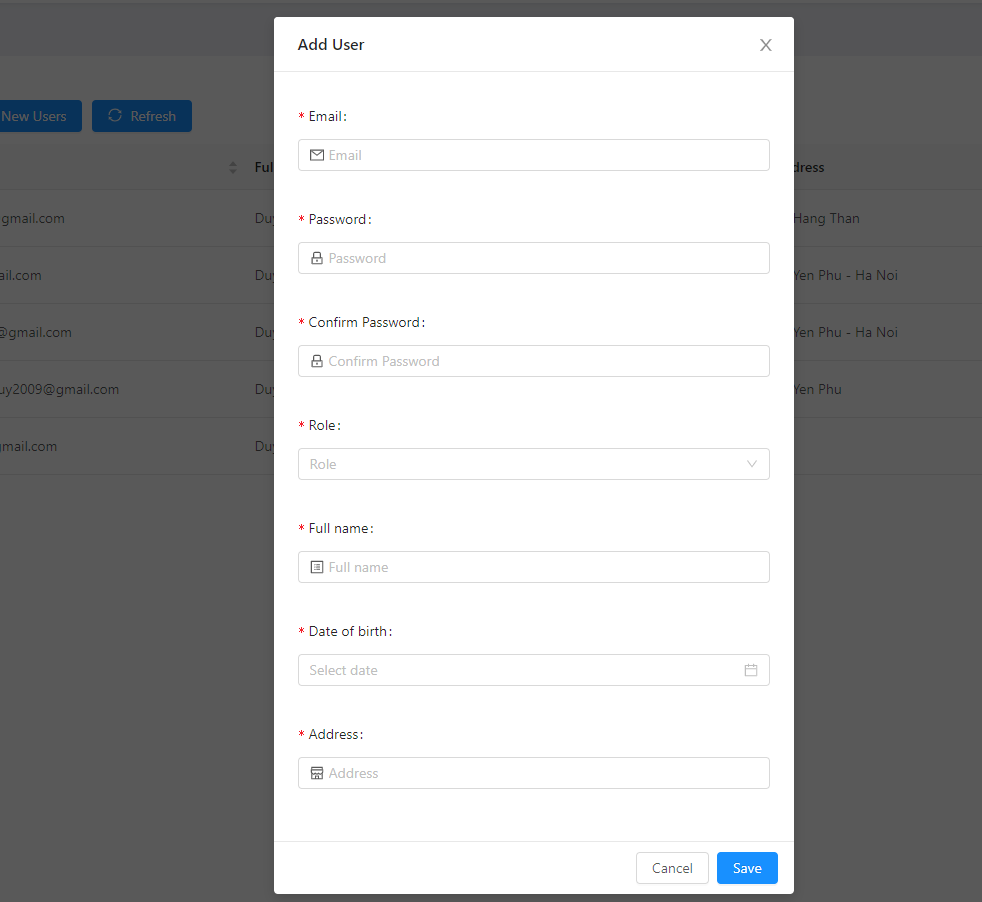


Figure 27 Adding a new user

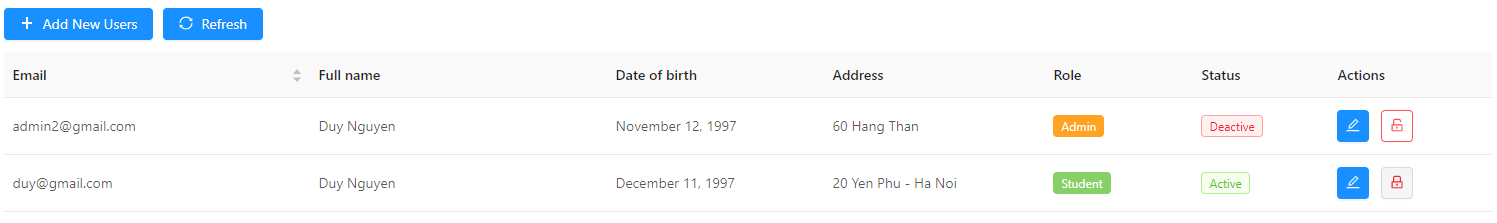


Figure 28 User's status can be changed

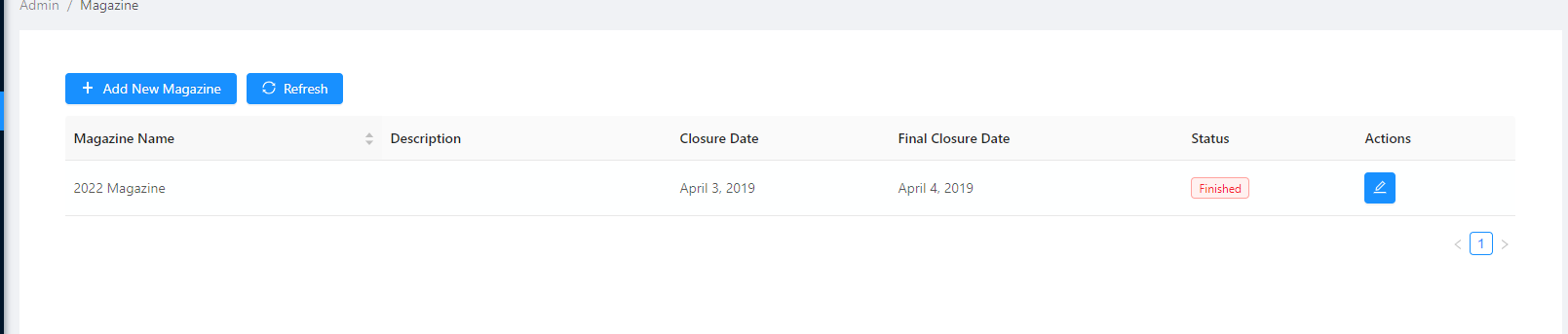


Figure 29 Managing magazine's information and deadlines

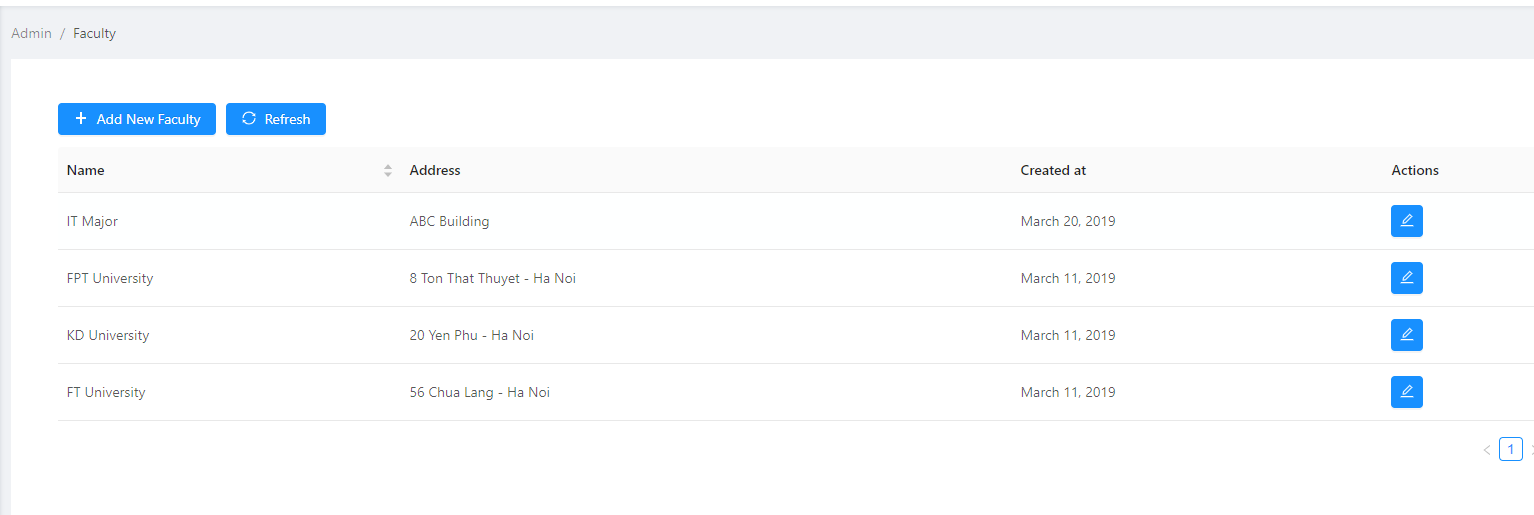


Figure 30 Managing faculties

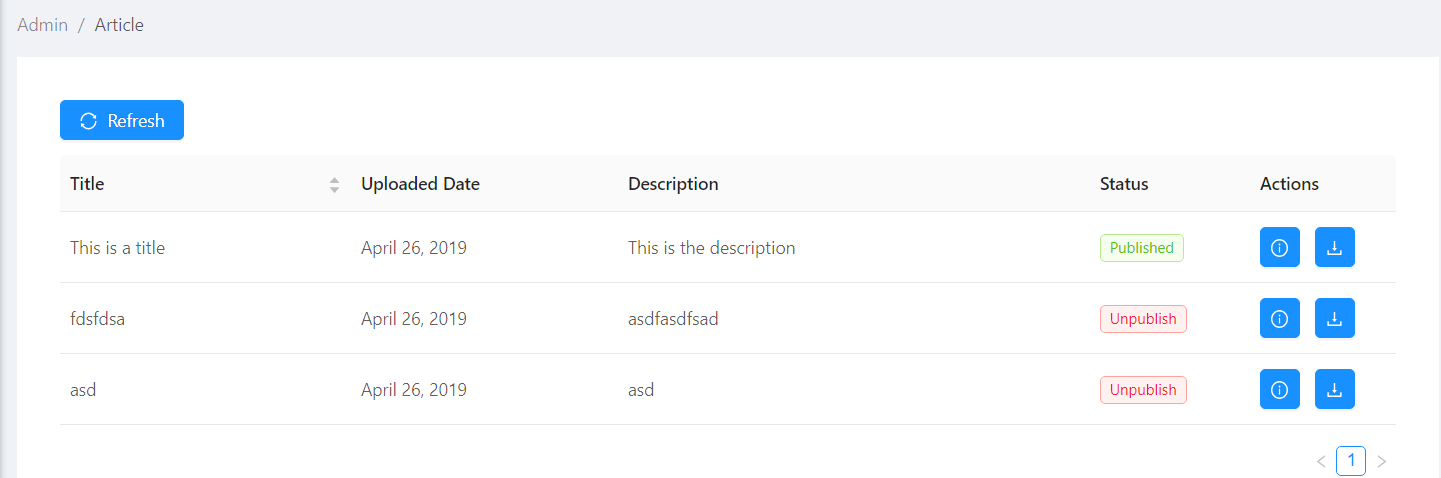


Figure 31 Managing articles

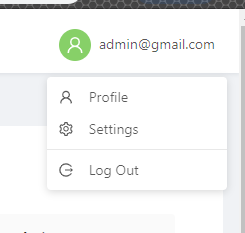


Figure 32 Account menu

The ‘Profile’ and ‘Settings’ page on the Account menu are not yet implemented.

### Students

Students are able to manage their own articles.

Below are the interfaces accessible to students:

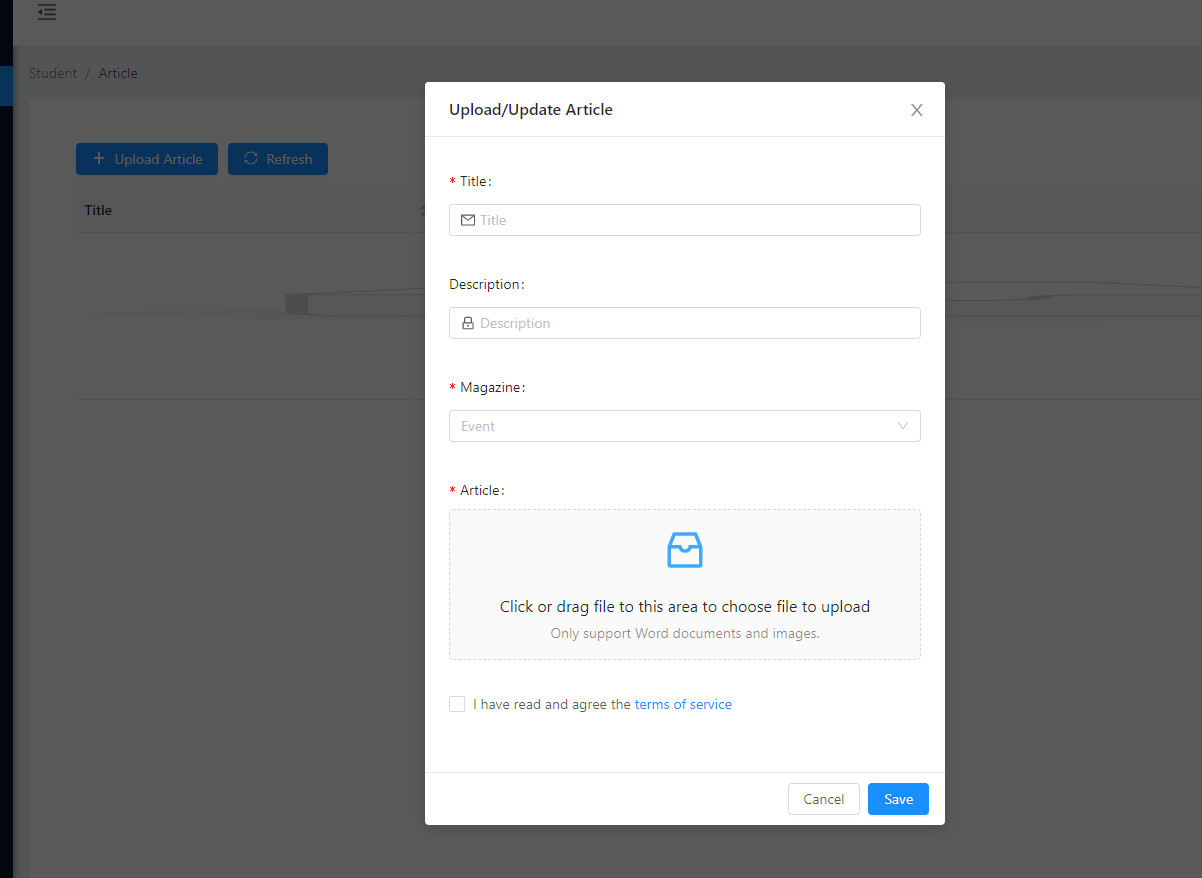


Figure 33 Submitting an article

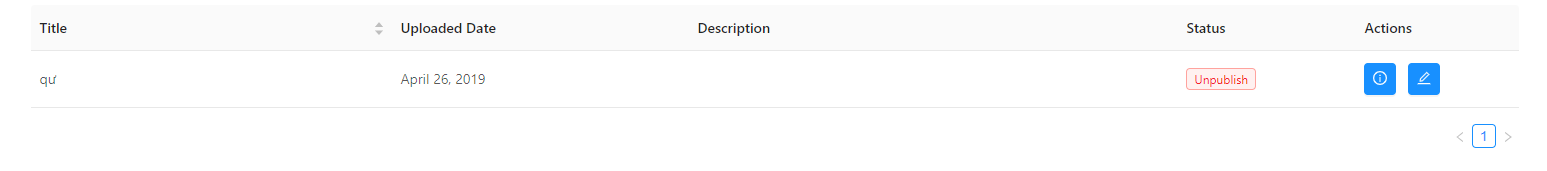


Figure 34 Viewing and editing their own submitted articles

### Coordinator

Coordinators are able to manage students and submitted articles within their own faculties.

Below are the interfaces accessible to a Coordinator:

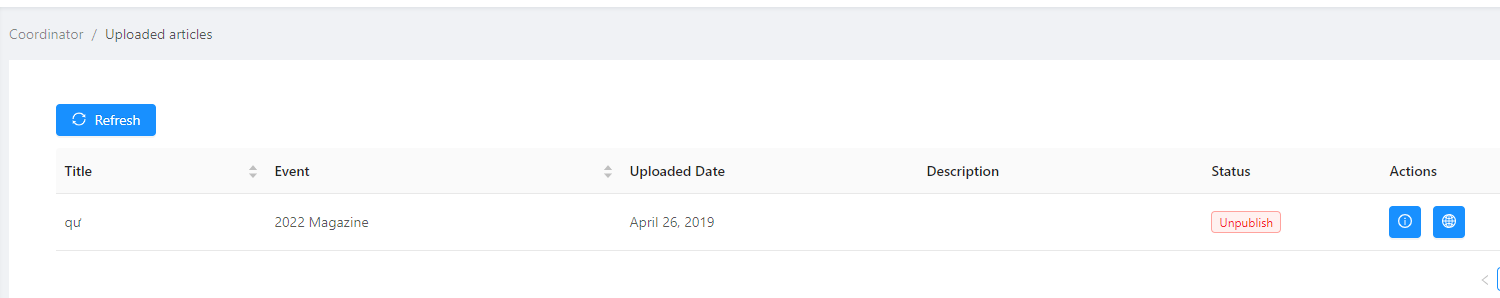


Figure 35 Managing submitted articles

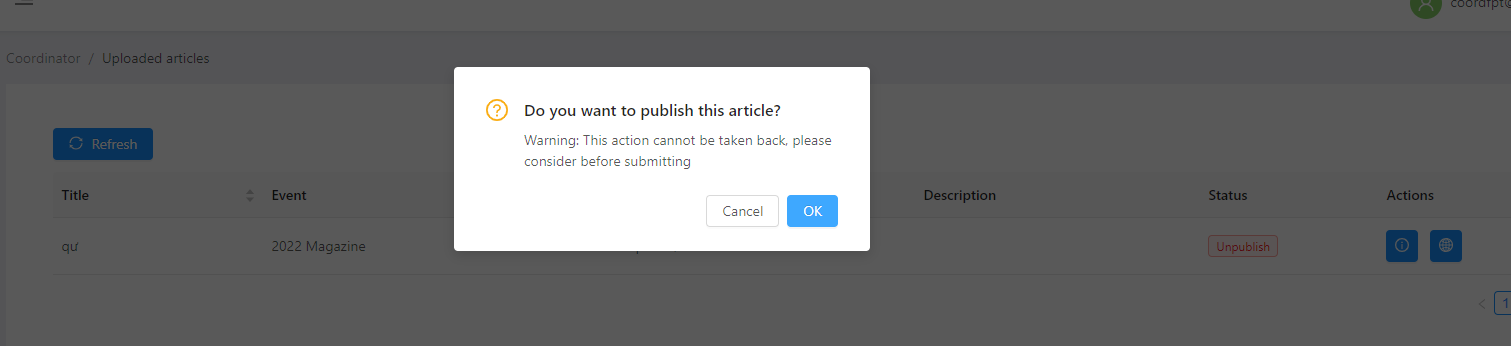


Figure 36 Selecting an article to publish

When a new article is uploaded to a faculty’s database, the website automatically sends an email to all coordinators in that faculty to ask them to comment of the article within 14 days. The following code snippet handles that functionality:



Figure 37 Emails are automatically sent to all coordinators in the faculty when a new article is uploaded

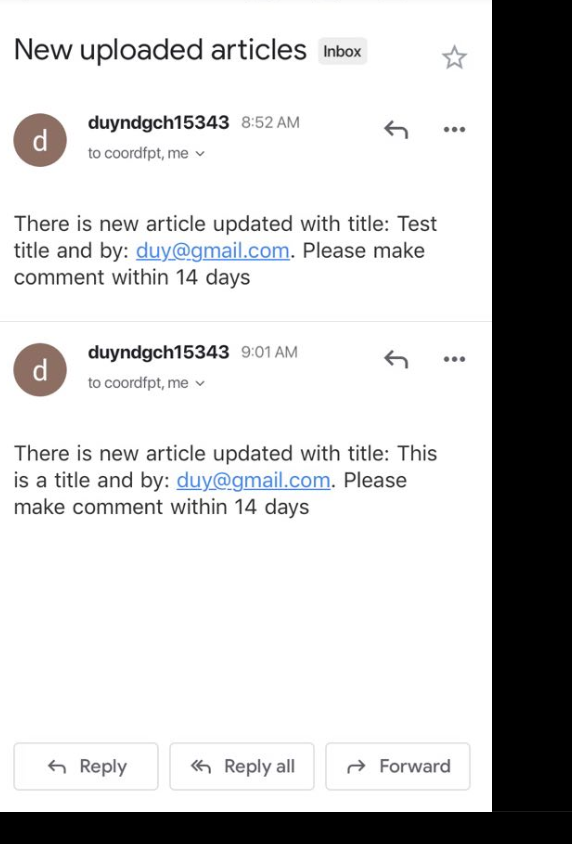


Figure 38 Email notifications sent to coordinator when new articles are uploaded

### Guest

Anybody who lands on the website can view published articles

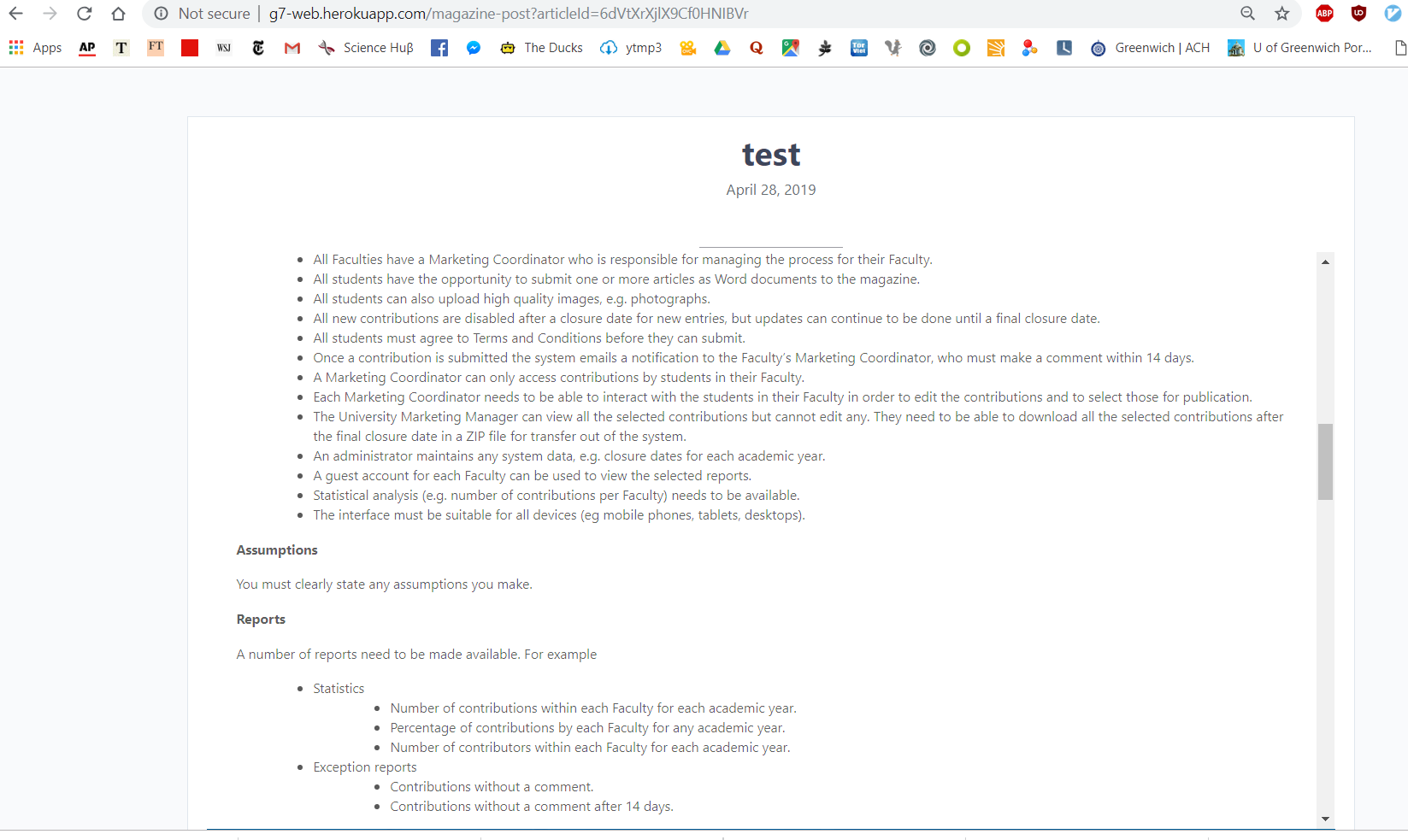


Figure 39 A test article

# Testing

The following are the various stages of testing that were carried out in our project.

* Unit testing: testing each unit of code at the smallest level.
* Integration testing: the program is tested to see how well each component work with each other (e.g.: information from the *database* is retrieved by the *server* then displayed onto the *user interface*)
* Functional testing: using black box testing, each implemented functionality is tested and the output recorded. Actual output is compared against expected output to determine the functionality Pass/Fail status.
* System testing: the whole website system is tested against the client’s requirements. While functional testing checks how well the functionalities meet the developers’ expectations, system testing checks how well they meet the client’s expectations.

Unit testing, Integration testing and Functional testing are carried out within each Sprint.

Acceptance testing are carried out in the last Sprint during the finalization of the product for delivery.

## Unit testing and Integration testing

Adhering to the Agile principle of “Working software over comprehensive documentation”, unit testing and integration testing are carried out by developers during the implementation process in a Test-Driven-Development-style: each developer tests his own code.

Tests at this stage are not documented. Instead, found errors are quickly and verbally communicated among developers to be fixed right away for optimal productivity.

## Functional testing

### Log in

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Functionality** | **Input** | **Excepted result** | **Actual result** | **Evaluate** |
| Authentication security | Username and password match a record in DB:  Username: test1- pw: test1 – role: seller | Login success, navigate to appropriate site | Login success, navigate to appropriate store home page, | pass |
| Input a wrong username / password | Notice: invalid username/pw. Login deny | Notice: invalid username/pw. Login deny | pass |
| Let username/Pw empty | Notice: username/ pw required | Notice: username/ pw required | Pass |
| Try using SQL injection | Notice: invalid username/pw. Login deny | Notice: invalid username/pw. Login deny | pass |

### Manage users

|  |  |  |  |
| --- | --- | --- | --- |
| **Functionality** | **Expected result** | **Actual Result** | **Evaluate** |
| View all users | List all users | List all users | Meet the expectation |
| Add new user | Add new user to existed list, required information must be validated. | Add new user to existed list, required information must be validated. | Meet expectation. But the validation can be improved. |
| Update user | Select user on list view, show current information and edit current information. | Select user on list view, show current information and edit current information. | Meet expectation.  But the validation must more carefully. |
| Deactivate user | User is deactivated. | User is deactivated. | Meet expectation |
|  |  |  |  |

### Manage articles

|  |  |  |  |
| --- | --- | --- | --- |
| **Functionality** | **Expected result** | **Actual Result** | **Evaluate** |
| View all articles | List all articles of product | List all articles of product | Meet the expectation |
| Add 1 article | New article is added to db | New article is added to db | Meet the expectation |
| Update selected article | article selected will display on a form let user edit. Form still need validated like add function | article selected will display on a form let user edit.  Form validated to avoid null value for articleName | Meet the expectation |
| Change status of selected article | Selected article will be Change status | Selected article will be Change status | Meet expectation |
| Download article | Article downloaded completely | Article downloaded completely | Meet expectation |
| View detail of article: | Article detail shown | Article detail shown | Meet expectation |

### Manage faculty

|  |  |  |  |
| --- | --- | --- | --- |
| **Functionality** | **Expected result** | **Actual Result** | **Evaluate** |
| View all faculties | List all faculties | List all faculties | Meet the expectation |
| Add 1 faculty | faculty name is validated to avoid null values | faculty name is validated to avoid null values | Meet the expectation |
| Update selected faculty | faculty selected will display on a form let user edit. Form still need validated like add function | faculty selected will display on a form let user edit.  Form validated to avoid null value for facultyName | Meet the expectation |

Manage magazine:

|  |  |  |  |
| --- | --- | --- | --- |
| **Functionality** | **Expected result** | **Actual Result** | **Evaluate** |
| View all articles | List all articles of product | List all articles of product | Meet the expectation |
| Add 1 article | article name is validated to avoid null values | article name is validated to avoid null values | Meet the expectation |
| Update selected articles | Chosen article is displayed on a form for the user edit.  Form to edit article still requires validation like the form to add new articles. | Chosen article is displayed on a form for the user edit. Form validated to avoid null value for articleName | Meet the expectation |
| Change status of selected article | Selected article will be Change status | Selected article will be Change status | Meet expectation |

### Interactions

|  |  |  |  |
| --- | --- | --- | --- |
| **Test** | **Expected Result** | **Actual result** | **Status** |
| Using button on the main menu to navigate between product site, home site and order site | Navigate successfully on all functions, no runtime error. | Navigate successfully on all functions, no runtime error. | Meet the expectation |
| Using Button on each page (add, update, delete) to navigate to pages | Navigate successfully on all functions, no runtime error. | Navigate successfully on all functions, no runtime error. | Meet the expectation |
| Login | Navigate to site belong the user (e.g.: store 1 user will be navigated to store1 store) | Navigate to site belong the user (e.g.: store 1 user will be navigated to store1 store) | Meet the expectation |
| Log out | Navigate to login site, all session data deleted. | Navigate to login site, all session data deleted. | Meet the expectation |
| Coordinator comment on a submitted article | Comment added | Comment added | Meet the expectation |
| Coordinator published a submitted article | Article published | Article published | Meet the expectation |
| Attempt to log in with an account deactivated by admin | Can’t login | Can’t login | Meet the expectation |

### Email notification

|  |  |  |  |
| --- | --- | --- | --- |
| **Test** | **Expected result** | **Actual Result** | **Evaluate** |
| Upload 1 article | Email sent to coordinator | Email sent to coordinator | Meet the expectation |
| Publish an article | Email sent to uploader | No email sent | Not Meet the expectation |
| Comment | Email sent to uploader | No email sent | Not Meet the expectation |

## System testing

This tests the whole system to see how well it suits the client’s requirements. Each item is rated from 0 to 5; 5 means the system meets the requirement perfectly.

### Compliance testing

|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | **User’s specification** | **Rating** | **Note** |
| 1 | The University has a Marketing Manager to oversee the process. | 5 | Meets the expectation. |
| 2 | All Faculties have a Marketing Coordinator who is responsible for managing the process for their Faculty. | 4 | Each Coordinator can only manage 1 Faculty.  Each Faculty may have multiple Coordinators.  Once an article is published, there is no interface yet to un-publish it. |
| 3 | All students have the opportunity to submit one or more articles as Word documents to the magazine. | 4 | Data validation fails to prevent students from uploading non-Word documents |
| 4 | All students can also upload high quality images, e.g. photographs. | 3 | Students are expected to put their image in the Word document and upload the document as a whole |
| 5 | All new contributions are disabled after a closure date for new entries, but updates can continue to be done until a final closure date. | 0 | Students can still upload documents after the closure date.  Managers are allowed to set the final closure date to precede the submission closure date, which is illogical. |
| 6 | All students must agree to Terms and Conditions before they can submit. | 5 | Meets the expectation |
| 7 | Once a contribution is submitted the system emails a notification to the Faculty’s Marketing Coordinator, who must make a comment within 14 days. | 4 | The coordinators receive email for every contribution, but there is no other measure to enforce that the contribution is commented within 14 days.  Note: it would help if the client specifies on how to deal with late comments. |
| 8 | A Marketing Coordinator can only access contributions by students in their Faculty. | 5 | Meets the expectation. |
| 9 | Each Marketing Coordinator needs to be able to interact with the students in their Faculty in order to edit the contributions and to select those for publication. | 5 | Meets the expectation.  Coordinators can comment on contributions, or send emails to students. |
| 10 | The University Marketing Manager can view all the selected contributions but cannot edit any. They need to be able to download all the selected contributions after the final closure date in a ZIP file for transfer out of the system. | 4 | Meets the expectation.  The list view of all contributions can also be improved with a search function. |
| 11 | An administrator maintains any system data, e.g. closure dates for each academic year. | 5 | Meets the expectation. |
| 12 | A guest account for each Faculty can be used to view the selected reports. | 4 | Any guest can view published article without an account. |
| 13 | Statistical analysis (e.g. number of contributions per Faculty) needs to be available. | 2 | The website provides basic statistics in table format. But the numbers seem to be incorrect.  Suggested improvement:   * Correct statistics * Graphical presentation (bar chart, pie chart, …) |
| 14 | The interface must be suitable for all devices (e.g., mobile phones, tablets, desktops). | 2 | Mobile interface is only partially implemented. |

### Usability and Accessibility testing

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Criteria** | **Rating** | **Note** |
| 1 | Site’s language must be clear and consistent | 4 | Mostly clear language. Only some minor spelling errors and inconsistencies. |
| 2 | Title of each site for identification and navigation | 5 | Meets the expectation |
| 3 | Tables have headers for rows and columns | 5 | Meets the expectation |
| 4 | Information display is visible, simple and un-distractive | 5 | Meets the expectation |
| 5 | All information available with and without color, especially for context and markup language | 5 | Highly contrasted color settings are perfectly visible in black and white |
| 6 | All pages are usable when script languages are turned off | 0 | The log-in function requires google scripts to work. Without script languages, users cannot log in. |
| 7 | All pages can work well with CSS and without CSS | 4 | Pages are hard to read without CSS, but workable |
| 8 | Symbols are intuitively easy to understand | 5 | Symbols are intuitive. Buttons have tooltips to clarify their functionalities |
| 9 | Font families and font sizes are appropriate | 5 | Meets the expectation |
| 10 | Sizes of elements are well proportioned | 3 | Elements on the home page are not very well proportioned |
| 11 | Images are attractive | 4 | Meets the expectation |
| 12 | Background and color themes are pleasant | 5 | Meets the expectation |

# Agile methods followed

In this section are the evidences of how our development process adheres to Scrum, the chosen Agile method.

## Minutes of meetings

Below are the minutes of our meetings. Meeting 0 was about the conception and initiation of the project. Meeting 1 to 5 were held at the start of each Sprint (from Sprint 1 to Sprint 5) to discuss the tasks to be carried out in each Sprint. In the Final meeting, we discussed the finalization, submission and demonstration of the project’s product.

### Meeting 0

Initiation of project:

* Problem: We were not sure which development platform should be chosen.
* Solution: We chose Firebase as the database, Node.js as the server development technique, and React as the front-end development technique based on over team members’ familiarity to them.
* Problem: A project schedule was needed.
* Solution: We chose a Scrum-style development, setting each Sprint to be 1 week (5 working days).
* Problem: Roles had to be assigned.
* Solution: Each member volunteered for a number of roles based on own’s level of expertise in each area:

|  |  |  |
| --- | --- | --- |
| Student ID | Name | Role |
| GCH15343 | Nguyễn Đăng Duy | Programmer, DB Designner, Product Owner |
| GCH15115 | Nguyễn Việt Sơn | Interface Design, SCRUM Master, Programmer |
| GCH15143 | Nguyễn Mạnh Tuấn | Tester, Programmer, DB Designner |
| GCH16110 | Vũ Trọng Nghĩa | Scrum master, Tester, DB Designner |

### Meeting 1

* Problem: Client’s specifications were non-technical and quite general.
* Solution: Client’s specifications were analysed to produce a list of user stories.
* Problem: Client’s specifications were quite general, thus open to interpretation.
* Solution: A number of assumptions were made and recorded.
* Problem: Tasks needed to be put into a schedule.
* Solution: User stories were categorized and prioritized. A number of stories were set into the first Sprint 1.

### Meeting 2

* Problem: 2 members are absent in the meeting (Tuan and Nghia)
* Solution: Contacted each absent member individually within next week to discuss their tasks for the Sprint 2.
* Problem: Both database designers were absent; there was nobody to discuss designing the database.
* Solution: Luckily, the database was largely finished by Sprint 1. Development for Sprint 2 could be continued without database designer.

### Meeting 3

* Problem: A functionality from Sprint 2 (editing a submitted article, while archiving the editing history for reversion purpose) was too complex and could not be completed.
* Solution: The function was simplified and scheduled to be completed in Sprint 3.
* Problem: 1 member is still absent (Nghia) and did not contribute to the project, even though online means.
* Solution: The team considered expelling Nghia from the group project.
* Problem: There was no Scrum master, as he was absent.
* Solution: Son volunteered to fill in for the role.

### Meeting 4

* Problem: Nghia returned to the project. He reported an eye infection made him incapable of leaving his room or use the computer.
* Solution: The team agreed to let him resume his role as Scrum master and Tester.
* Problem: The returning member needed to catch up with the project’s progress.
* Solution: Server-side developer had a meeting with Nghia to help him go through the components of the server.

### Meeting 5

* Problem: As this is the meeting for the final Sprint before delivery, the team needs to decide which additional functionalities could be completed and which could be omitted.
* Solution: The team discussed and agreed on a number of tasks to be added to the final Sprint. Other features were omitted from the final product.

### Final meeting

* Problem: The team needed to finalize the work to be ready for submission.
* Solution: The team discussed and finalized the project. The team agreed on a number of artefacts (documents, diagrams, …) to be included. The final product was ready for demonstration and submission.

## User stories

Below are the user stories that we have implemented in our website:

* As a user I want to be able to log in to website and identify my role.
* As a student I want to be able to Upload Articles to be selected for publication.
* As a student I want to be able to view Articles I have uploaded.
* As a student I want to edit my article that I have submitted.
* As a student I want to see comments from coordinator on my submissions.
* As a coordinator I want to comment in student's uploaded article.
* As a coordinator I want to activate and deactivate student's account.
* As a coordinator I want to see submitted articles in my faculty.
* As a coordinator I want to select article for publication.
* As an admin I want to be able to view all submitted and selected articles.
* As an admin I want to be able to download all selected articles.
* As an admin I want to be able to set submission closure date and final closure date.
* As an admin I want to create new accounts for student in my faculty.
* As an admin I want to be able to view statistical analysis of article contributions.
* As an admin I want to create new events.
* As a guest I want to be able to view all published articles.

## Assumptions

Below are the assumptions made to areas not specified by the clients:

Each student has to accept T&C before submitting each article

Each student account or coordinator account belongs to a specific faculty; the student or coordinator can only manage articles in that faculty.

Decisions to update articles are final; articles cannot be retracted. They can, however, be edited.

All uploaded and updated versions of each article is stored in the DB

Student account is provided in advance by admin (students cannot create their own accounts).

## Product backlogs

Below is the list of all product backlog items:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **As/An** | **I want to** | **So that** | **Acceptance Criteria** | **Added in sprint** | **Story points** | **Priority** |
| 1 | All | Contact admin | For activities that are not within my permission | - Provide information |  | 10 | 3 |
| 2 | All | Change my password | I can manage my account's information | - Must provide old password - Must be different with the old password |  | 10 | 3 |
| 3 | All | Login/Logout | I have access to the website functionalities according to my role | - Correct username and password | 1 | 20 | 1 |
| 5 | Student | Upload article | Coordinator can view and give feedback | - File .docx .png - Before closure date - Has to accept Terms and Conditions | 1 | 30 | 1 |
| 6 | Student | View my uploaded articles | I can see what I have uploaded | I can see what I have uploaded | 2 | 40 | 1 |
| 7 | Student | Update my articles | I can edit my article when I want to change the content | - Old versions are saved in a history - View article's history | 3 | 20 | 2 |
| 8 | Student | View updated history | Can revert the update if necessary, or compare it with the new version | - Compare different versions |  | 30 | 3 |
| 9 | Coord | View articles submitted by students | I can see the article of student and select it for publication. | - Can only view articles in my faculty - View student's personal info | 3 | 20 | 2 |
| 10 | Coord | Comment on articles | I can send the students my feedback about their article | - Cannot modify the articles - Must comments within 14days | 2 | 15 | 2 |
| 11 | Coord | Receive an email when an article is submitted | I can know when the article has been submitted and commented on | - Coord receive the email immediately after submission | 3 | 15 | 2 |
| 12 | Coord | Select articles for publication | I can choose which article I want to publish | - View article (modified history) - Updating an article also changes its 'Selected' status | 3 | 15 | 1 |
| 13 | Manager | View all submitted and selected articles | I can download and transfer it | - Cannot edit articles - View all info | 4 | 20 | 1 |
| 14 | Manager | Download the selected articles | I can transfer it to another place for storage | - Notified when a selection is updated - File .ZIP | 4 | 30 | 2 |
| 15 | Manager | Set submission closure date and final closure date | I can set the deadline to student and I will know when they are late | - Can only choose dates in the future - Final closure date must NOT be before submission closure date | 4 | 30 | 1 |
| 16 | Manager | View statistics | I can see overview about students and their work | - No. of contributions per faculty | 4 | 20 | 1 |
| 17 | Manager | Create account for student and coord | Student and Coord can be able to log in the system |  | 4 | 20 | 1 |
| 18 | Guest | View selected article | I can read the article which I selected | - Have only View permission and only to Selected articles | 4 | 40 | 1 |

## Sprints

Following are the records of 5 sprints of our project:

### Sprint 1

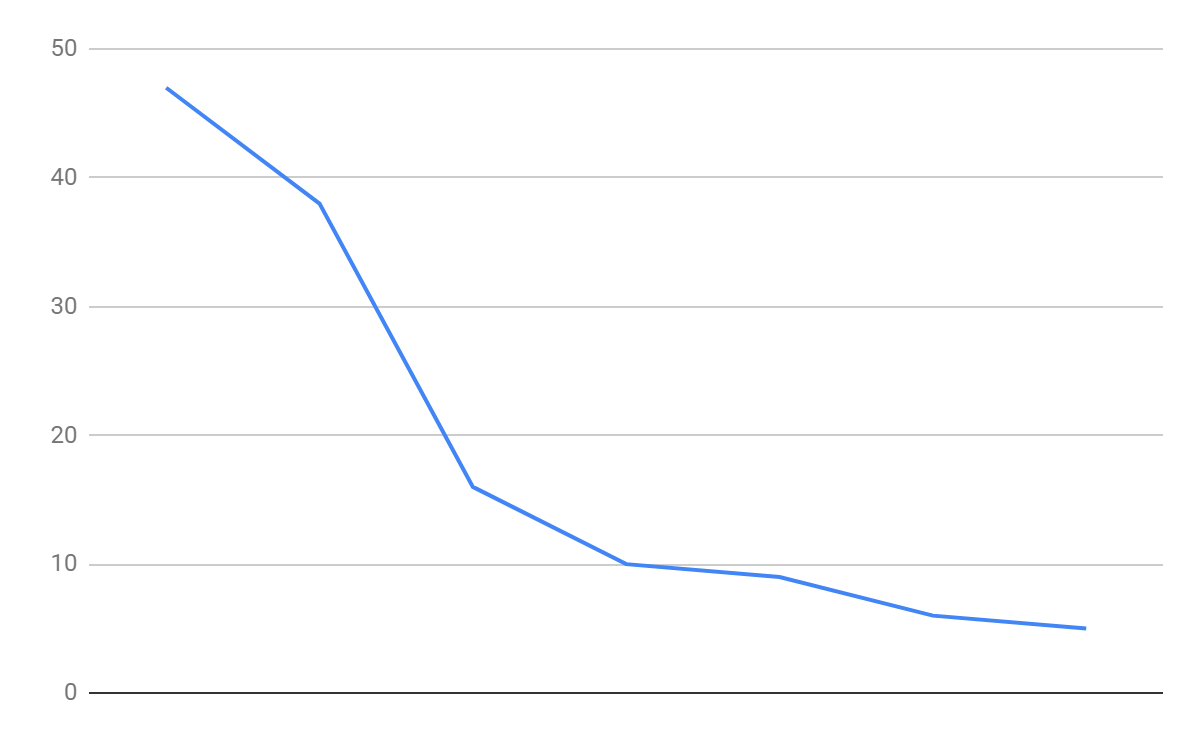


Figure 40 Sprint 1's burn down chart

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Backlog Item** | **Task** | **Volunteer** | **Status** | **Original Estimate** | **Day 1** | **Day 2** | **Day 3** | **Day 4** | **Day 5** | **Sprint Review** |
| As a user I want to be able to log in to website and identify my role | Interface to login | Son | Done | 5 | 3 | 2 | 0 | 0 | 0 | 0 |
| Interface to log out | Son | Done | 5 | 3 | 2 | 0 | 0 | 0 | 0 |
| Set up login validation in Database | Duy | Done | 5 | 3 | 2 | 0 | 0 | 0 | 0 |
| Functional testing | Tuan | Done | 3 | 3 | 0 | 0 | 0 | 0 | 0 |
| As a Student I want to be able to Upload Articles to be selected for publication | Set up database for Students, Articles and Department | Son | Done | 5 | 3 | 1 | 1 | 0 | 0 | 0 |
| Interface to upload articles | Duy | Done | 5 | 5 | 0 | 0 | 0 | 0 | 0 |
| Connection from DB to UI | Duy | Done | 2 | 3 | 0 | 0 | 0 | 0 | 0 |
| Integration testing | Nghia | Done | 3 | 0 | 0 | 0 | 3 | 0 | 0 |
| As a Student I want to be able to view Articles I have uploaded | Interface to view articles in database | Tuan | Not done | 5 | 3 | 3 | 3 | 3 | 3 | 2 |
| Connection from DB to UI | Duy | Done | 3 | 3 | 0 | 0 | 0 | 0 | 0 |
| Functional testing | Nghia | Not done | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
|  | Total |  |  | 47 | 38 | 16 | 10 | 9 | 6 | 5 |

### Sprint 2

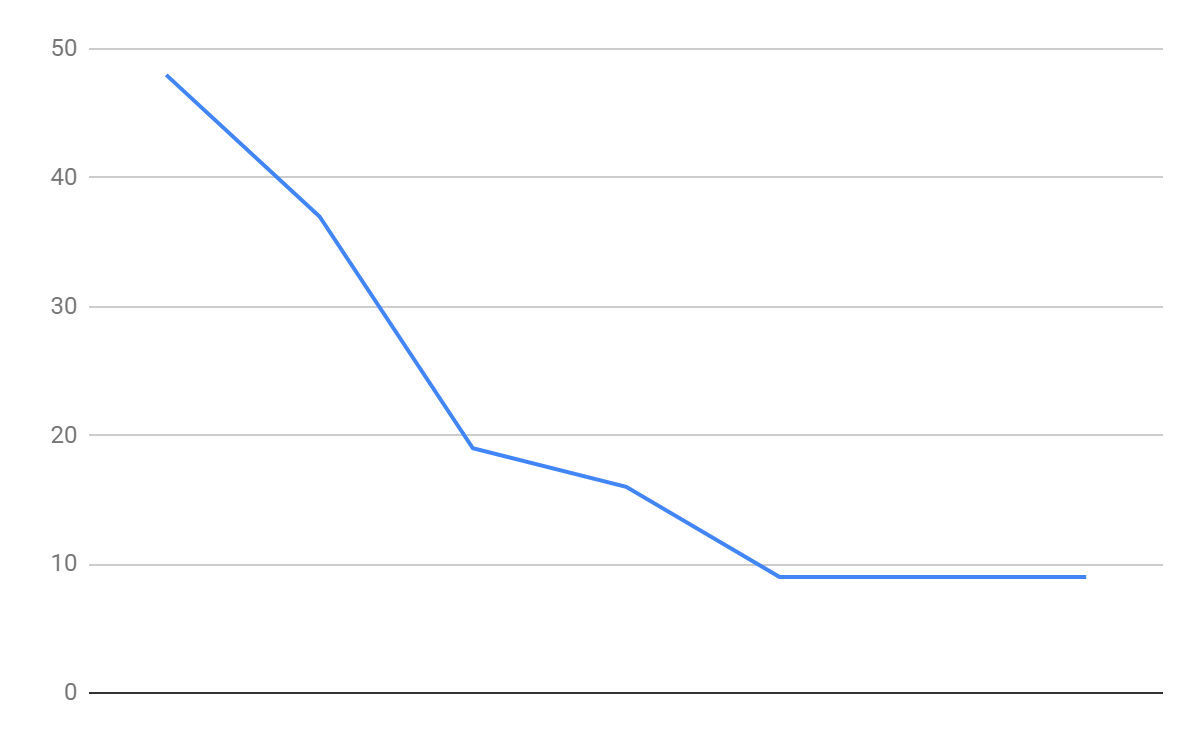


Figure 41 Sprint 2's burn down chart

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Backlog Item** | **Task** | **Volunteer** | **Status** | **Original Estimate** | **Day 1** | **Day 2** | **Day 3** | **Day 4** | **Day 5** | **Sprint Review** |
| As a student, I want to edit my article that I have submitted | Interface to edit | Son | In process | 5 | 3 | 3 | 3 | 3 | 3 | 3 |
| Set up login validation in Database | Duy | In process | 5 | 3 | 3 | 3 | 3 | 3 | 3 |
| Functional testing | Tuan | Not done | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| As a student, I want to see comment from coordinator on my submissions | Set up database for comment | Duy | Done | 5 | 3 | 1 | 1 | 0 | 0 | 0 |
| Interface to view comment | Son | Done | 5 | 5 | 0 | 0 | 0 | 0 | 0 |
| Connection from DB to UI | Duy | Done | 2 | 3 | 0 | 0 | 0 | 0 | 0 |
| Functional testing | Tuan | Done | 3 | 3 | 3 | 3 | 0 | 0 | 0 |
| As a coordinator I want to comment in student's uploaded article | Interface to view articles and comment's placeholder | Son | Done | 5 | 3 | 2 | 0 | 0 | 0 | 0 |
| Connection from DB to UI | Duy | Done | 3 | 3 | 0 | 0 | 0 | 0 | 0 |
| Functional testing | Tuan | Done | 3 | 2 | 1 | 0 | 0 | 0 | 0 |
| As a student I want to be able to view Articles I have uploaded | Interface to view articles in database | Son | Done | 2 | 1 | 1 | 1 | 0 | 0 | 0 |
| Connection from DB to UI | Duy | Done | 3 | 3 | 0 | 0 | 0 | 0 | 0 |
| Functional testing | Tuan | Done | 4 | 2 | 2 | 2 | 0 | 0 | 0 |
|  | Total |  |  | 48 | 37 | 19 | 16 | 9 | 9 | 9 |

### Sprint 3

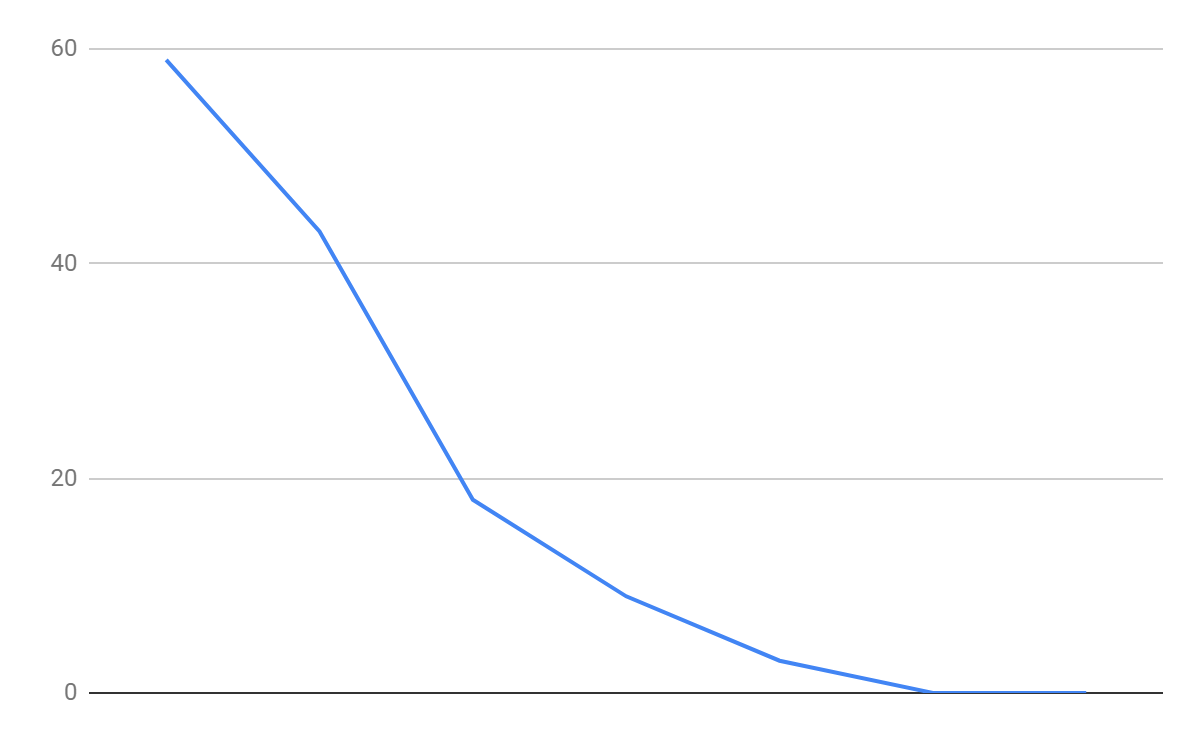


Figure 43 Sprint 3's burn down chart

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Backlog Item** | **Task** | **Volunteer** | **Status** | **Original Estimate** | **Day 1** | **Day 2** | **Day 3** | **Day 4** | **Day 5** | **Sprint Review** |
| As a coordinator I want to activate and deactivate student's account | Set up database for Students and Articles | Duy | Done | 5 | 3 | 1 | 0 | 0 | 0 | 0 |
| Interface to active and deactivate | Son | Done | 5 | 5 | 0 | 0 | 0 | 0 | 0 |
| Connection from DB to UI | Duy | Done | 3 | 3 | 0 | 0 | 0 | 0 | 0 |
| Functional testing | Tuan | Done | 3 | 2 | 1 | 0 | 0 | 0 | 0 |
| As a student, I want to edit my article that I have submitted | Interface to edit | Son | Done | 3 | 3 | 3 | 3 | 0 | 0 | 0 |
| Set up login validation in Database | Duy | Done | 3 | 3 | 3 | 3 | 0 | 0 | 0 |
| Functional testing | Tuan | Done | 3 | 3 | 3 | 3 | 3 | 0 | 0 |
| As a coordinator I want to see submitted articles in my faculty | Interface to view articles in database | Son | Done | 5 | 3 | 2 | 0 | 0 | 0 | 0 |
| Connection from DB to UI | Duy | Done | 3 | 3 | 0 | 0 | 0 | 0 | 0 |
| Functional testing | Tuan | Done | 3 | 0 | 0 | 0 | 0 | 0 | 0 |
| As a coordinator I want to select article for publication | Interface for public | Son | Done | 5 | 3 | 2 | 0 | 0 | 0 | 0 |
| Connection from DB to UI | Duy | Done | 3 | 3 | 0 | 0 | 0 | 0 | 0 |
| Functional testing | Tuan | Done | 3 | 3 | 0 | 0 | 0 | 0 | 0 |
| As a coordinator I want to receive email right after students summited their article | Function to send email | Duy | Done | 6 | 3 | 3 | 0 | 0 | 0 | 0 |
| Connection from function to UI | Duy | Done | 3 | 3 | 0 | 0 | 0 | 0 | 0 |
| Functional testing | Tuan | Done | 3 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | Total |  |  | 59 | 43 | 18 | 9 | 3 | 0 | 0 |

### Sprint 4

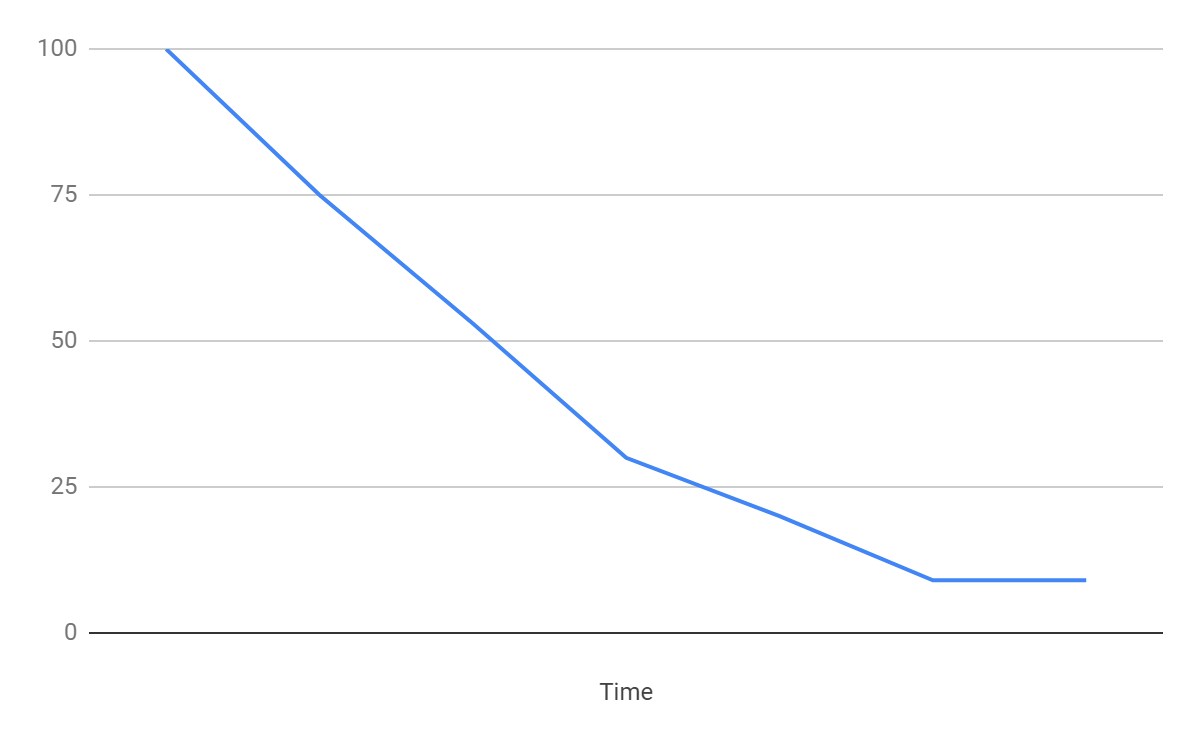


Figure 44 Sprint 4's burn down chart

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Backlog Item** | **Task** | **Volunteer** | **Status** | **Original Estimate** | **Day 1** | **Day 2** | **Day 3** | **Day 4** | **Day 5** | **Sprint Review** |
| As a manager I want to be able to view all submitted and selected articles | Interface to view | Son | Done | 10 | 7 | 3 | 0 | 0 | 0 | 0 |
| Get article in DB | Duy | Done | 5 | 3 | 2 | 0 | 0 | 0 | 0 |
| Functional testing | Tuan | Done | 3 | 3 | 3 | 0 | 0 | 0 | 0 |
| As a manager I want to be able to download all selected articles | Set up database for download | Son | Done | 5 | 4 | 3 | 3 | 3 | 3 | 3 |
| Interface to download articles | Duy | Done | 5 | 3 | 3 | 3 | 3 | 3 | 3 |
| Connection from DB to UI | Duy | Done | 3 | 3 | 1 | 1 | 1 | 1 | 1 |
| Functional testing | Tuan | Done | 3 | 3 | 3 | 2 | 2 | 2 | 2 |
| As a manager I want to be able to set submission closure date and final closure date | Interface | Duy | Done | 5 | 3 | 2 | 0 | 0 | 0 | 0 |
| Connection from DB to UI | Tuan | Done | 3 | 3 | 3 | 0 | 0 | 0 | 0 |
| Functional testing | Tuan | Done | 3 | 3 | 3 | 3 | 0 | 0 | 0 |
| As a manager I want to create new accounts for student in my faculty | Interface to create account | Son | Done | 5 | 3 | 2 | 0 | 0 | 0 | 0 |
| Connection from DB to UI | Duy | Done | 5 | 3 | 2 | 0 | 0 | 0 | 0 |
| Functional testing | Tuan | Done | 3 | 3 | 3 | 3 | 0 | 0 | 0 |
| As a manager I want to be able to view statistical analysis of article contributions | Interface to view statistics | Duy | Done | 5 | 3 | 2 | 0 | 0 | 0 | 0 |
| Connection from DB to UI | Tuan | Done | 3 | 3 | 3 | 3 | 3 | 0 | 0 |
| Functional testing | Tuan | Done | 3 | 3 | 3 | 3 | 3 | 0 | 0 |
| As a manager I want to create new events | Interface to create event | Son | Done | 5 | 3 | 2 | 0 | 0 | 0 | 0 |
| Connection from DB to UI | Duy | Done | 3 | 3 | 0 | 0 | 0 | 0 | 0 |
| Functional testing | Tuan | Done | 3 | 3 | 0 | 0 | 0 | 0 | 0 |
| As a guest I want to be able to view all published articles | Interface to view all article | Son | Done | 7 | 4 | 3 | 2 | 0 | 0 | 0 |
| Interface to view a specific article | Son | Done | 5 | 3 | 2 | 2 | 1 | 0 | 0 |
| Connection from DB to UI | Duy | Done | 5 | 3 | 2 | 2 | 1 | 0 | 0 |
| Functional testing | Tuan | Done | 3 | 3 | 3 | 3 | 3 | 0 | 0 |
|  | **Total** |  |  | **88** | **67** | **44** | **24** | **10** | **3** | **0** |

### Sprint 5

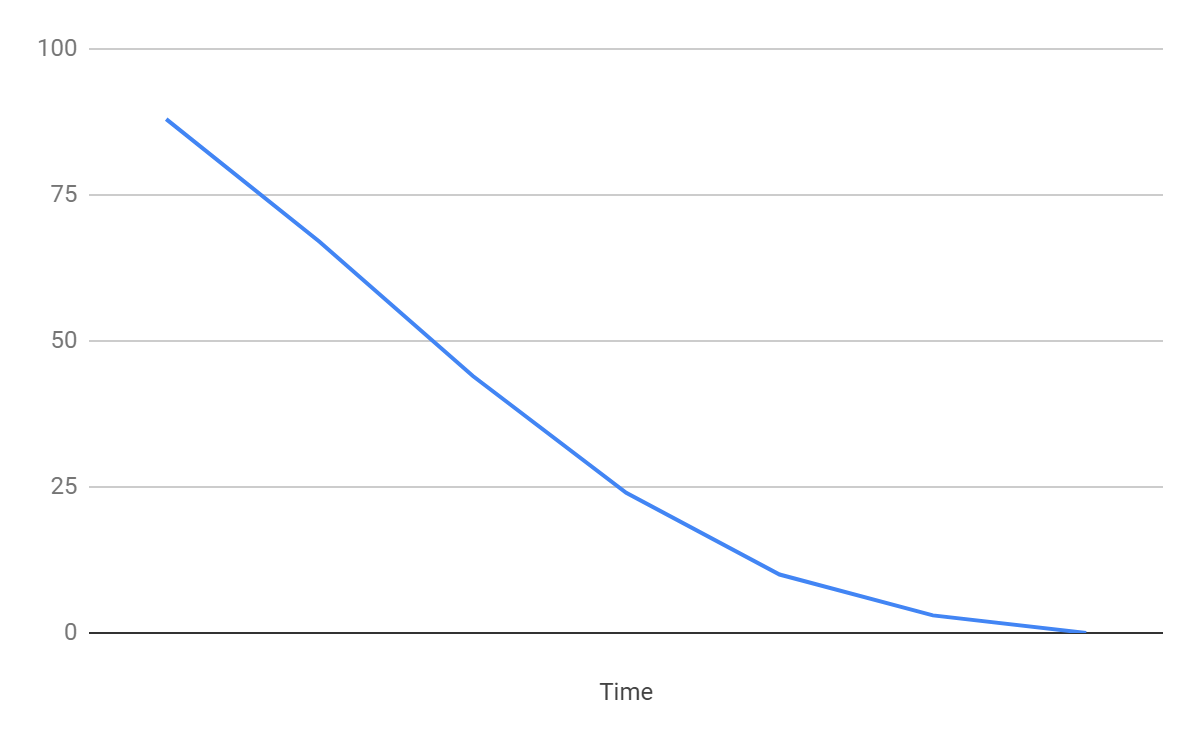


Figure 45 Sprint 5's burn down chart

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Backlog Item** | **Task** | **Volunteer** | **Status** | **Original Estimate** | **Day 1** | **Day 2** | **Day 3** | **Day 4** | **Day 5** | **Sprint Review** |
| As a student I want to have a notification by email if I late the deadline 14 days | Interface to view | Son | Done | 7 | 7 | 3 | 0 | 0 | 0 | 0 |
| Get article in DB | Duy | Done | 5 | 3 | 2 | 0 | 0 | 0 | 0 |
| Test | Nghia | Done | 3 | 3 | 3 | 0 | 0 | 0 | 0 |
| As a manager I want to be able to download all selected articles | Set up database for download | Son | Done | 3 | 2 | 2 | 0 | 0 | 0 | 0 |
| Interface to download articles | Duy | Done | 3 | 3 | 2 | 0 | 0 | 0 | 0 |
| Connection from DB to UI | Duy | Done | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| Test | Nghia | Done | 2 | 2 | 2 | 2 | 0 | 0 | 0 |
| As a coordinator I want publish an | Interface | Duy | Done | 5 | 3 | 2 | 0 | 0 | 0 | 0 |
| Connection from DB to UI | Tuan | Done | 3 | 3 | 3 | 0 | 0 | 0 | 0 |
| Test | Nghia | Done | 3 | 3 | 3 | 3 | 0 | 0 | 0 |
| As a manager I want to view each user's information | Interface to view | Son | Done | 3 | 3 | 2 | 2 | 1 | 0 | 0 |
| Connection from DB to UI | Duy | Done | 3 | 3 | 3 | 2 | 1 | 0 | 0 |
| Test | Nghia | Done | 3 | 3 | 3 | 3 | 3 | 0 | 0 |
| As a manager I want to edit each user's information | Interface to edit | Son | Done | 3 | 3 | 0 | 0 | 0 | 0 | 0 |
| Connection from DB to UI | Duy | Done | 3 | 2 | 0 | 0 | 0 | 0 | 0 |
| Test | Nghia | Done | 3 | 1 | 0 | 0 | 0 | 0 | 0 |
| As a manager I want to view each faculty's information | Interface to view | Son | Done | 3 | 0 | 0 | 0 | 0 | 0 | 0 |
| Connection from DB to UI | Duy | Done | 3 | 0 | 0 | 0 | 0 | 0 | 0 |
| Test | Nghia | Done | 3 | 0 | 0 | 0 | 0 | 0 | 0 |
| As a manager I want to edit each faculty's information | Interface to edit | Son | Done | 3 | 2 | 0 | 0 | 0 | 0 | 0 |
| Connection from DB to UI | Duy | Done | 3 | 2 | 0 | 0 | 0 | 0 | 0 |
| Test | Nghia | Done | 3 | 3 | 0 | 0 | 0 | 0 | 0 |
| As the product owner I want to ensure that the final product complies with users' specifications | System compliance testing | Nghia | Done | 12 | 10 | 8 | 7 | 3 | 1 | 0 |
| System accessibility testing | Nghia | Done | 5 | 5 | 5 | 5 | 2 | 2 | 0 |
|  | Total |  |  | 88 | 67 | 44 | 24 | 10 | 3 | 0 |

# Screencast and Presentation

Here is the follow links to the screencast and presentation slides to demonstrate the functionalities of our product:

* Screencast:

<https://drive.google.com/file/d/1ewbILilpMe2ONgrzDCoCrOraL23iMDCp/>

* Presentation slides:

<https://drive.google.com/open?id=1dGTu0sQxhFV98a9G4-WCqjtza-Tj6YX_>