CEN4010 – Principles of Software Engineering.

Fall 2021

Relay

Group 10 – Emanon.

Team members

Andrew Sexton [asexton2020@fau.edu](mailto:asexton2020@fau.edu) (Team Lead)

Maxon Corvil [mcorvil2018@fau.edu](mailto:mcorvil2018@fau.edu) (Front-end Lead)

Ashley Davis [davisashley2020@fau.edu](mailto:davisashley2020@fau.edu) (Back-end Developer)

Patrick Messina [pmessina2017@fau.edu](mailto:pmessina2017@fau.edu) (Front-end Developer)

Michael Niebauer [mniebaue@fau.edu](mailto:mniebaue@fau.edu) (Back-end Lead)

Milestone 3 – Project Proposal.

November 15, 2021

Revision Dates

|  |  |  |
| --- | --- | --- |
| Revision | Date | Changes |
| 1 | 11/10/2021 |  |
| 2 | 11/13/2021 | Added Confirm Input Data Validation section |
| 3 | 11/14/2021 | Added Confirm that Passwords are Encrypted in DB |
| 4 | 11/14/2021 | Added QA test plan. Changed adherence to nonfunctional specs to indicate issues with email validation/2FA |
|  |  |  |

1. **Product Summary**
   1. **Name of Product:** Relay
   2. **All Major Committed Functions**
      1. **Create Accounts –** Users must create an account to use most features of the site. This requires a username, password, email, first name and last name.
      2. **Create Post –** Users with accounts will be able to share text and image content with other users of the site and control who sees their posts.
      3. **Search –** Users will be able to search for posts and other users by typing search terms into a search bar. They will be provided with a list of matching users and posts upon clicking search.
      4. **Add Friend –** Users will be able to send friend requests to other users. Upon approval by the other user, the two users will become friends and unlock certain privileges for the other user’s content.
      5. **Interact with Post –** Users will be able to like posts and reply to them with comments.
      6. **Chatrooms** **–** Users will be able to talk with their friends in real-time using private chatrooms.
      7. **Manage Friends –** Users will be able to set permission levels for each of their friends that determine which posts the friend is able to see on the site.
   3. **Unique Features of the product**

The main unique features of the website are the Chatroom and Friend Management features. Chatrooms are intended to provide a better connection between users than most other social media sites by letting users privately communicate in real time. The Friend Management features let users give users more control over who can see their posts than similar sites. We hope that these features come together to provide better connections between users while maintaining a high level of privacy.

* 1. **URL:** <https://lamp.cse.fau.edu/~cen4010_fa21_g10/relay/>

1. **Usability Test Plan**
   1. **Test Objectives**
   2. **Test Plan**
   3. **Questionnaire Form**
2. **QA Test Plan**
   1. **Test Objectives**

The objective of the QA test is to quantitatively verify that the posts feature meets the non-functional requirements outlined in the specifications. These requirements include a response time of 50ms for uploading a post to the database and privacy requirements where only the type of user specified at post creation may view a post. These requirements must also be met on both Mozilla Firefox and Google Chrome.

* 1. **Hardware and Software Setup**

The website will be hosted on our group’s FAU LAMP server: <https://lamp.cse.fau.edu/~cen4010_fa21_g10/relay/> and will be accessed using both Mozilla Firefox and Google Chrome. Our group’s MySQL database will hold store the posts. Tester will log in to a Relay account and create posts at <https://lamp.cse.fau.edu/~cen4010_fa21_g10/relay/post.php>. Tester will then create a post and use the Network Monitor feature of Firefox or the Network panel in Chrome’s DevTools to observe and record the timing of the post request. Tester will also have three other Relay accounts at each friendship level to verify the privacy restrictions are met.

* 1. **Feature to be Tested**

The feature to be tested is the Create Post feature which allows users to submit text content with an optional image to be displayed on the site. Users may enter up to 200 characters to create a short post, or more than 200 for a long post. Users may also enter a short title for their post and create a teaser of up to 200 characters that will be displayed on the site. Users may also select one of three privacy levels for the post that dictate whether the post may be viewed publicly, may be restricted to their friends, or may be viewed only by their best friends.

* 1. **Actual Test Cases**

**Test Case 1:** When logged into a Relay account, the tester will create a post of exactly 200 characters and include an image. The tester should set the privacy setting to Community. Upon clicking submit, the tester should record in the results table how long the post request took using Firefox’s Network Monitor or Chrome’s DevTools depending on the browser used. The tester should also record whether post creation met the requirement of 50ms or less, or if it took longer. The tester should then log in to three different Relay accounts at each of the three friendship levels with the account used to create the post and verify that they are all able to see the post.

**Test Case 2:** When logged into a Relay account, the tester will create a post of exactly 500 characters and include an image. The tester should set the privacy setting to Friends Only. Upon clicking submit, the tester should record in the results table how long the post request took using Firefox’s Network Monitor or Chrome’s DevTools depending on the browser used. The tester should also record whether post creation met the requirement of 50ms or less, or if it took longer. The tester should then log in to three different Relay accounts at each of the three friendship levels with the account used to create the post and verify that the only account unable to see the post is the account that is not a friend or best friend of the account used to create the post.

**Test Case 3:** When logged into a Relay account, the tester will create a post of exactly 500 characters with an image that includes a 200-character teaser that is different from the post’s main content. The tester should set the privacy setting to Best Friends Only. Upon clicking submit, the tester should record in the results table how long the post request took using Firefox’s Network Monitor or Chrome’s DevTools depending on the browser used. The tester should also record whether post creation met the requirement of 50ms or less, or if it took longer. The tester should then log in to three different Relay accounts at each of the three friendship levels with the account used to create the post and verify that the only account able to view the post is the one that is a best friend of the account used to create the post.

* 1. **Test Plan Results Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test # | Test Title | Test Description | Test Input | Expected Correct Output | Test Result |
| 1a | Test 1 (Firefox) | Tester measures how long it takes a Community post to be added to the database and verifies its privacy. | 200-character long Community post with an image. | Post is visible by anyone using Relay. Post request takes 50ms or less. |  |
| 2a | Test 2 (Firefox) | Tester measures how long it takes Friends Only post to be added to the database and verifies its privacy. | 500-character long Community post with an image. | Post is visible by author account’s Friends using Relay. Post request takes 50ms or less. |  |
| 3a | Test 3 (Firefox) | Tester measures how long it takes Best Friends Only post to be added to the database and verifies its privacy. | 500-character long Community post with an image and 200-character long teaser. | Post is visible by author account’s Best Friends using Relay. Post request takes 50ms or less. |  |
| 1b | Test 1 (Chrome) | Tester measures how long it takes a Community post to be added to the database and verifies its privacy. | 200-character long Community post with an image. | Post is visible by anyone using Relay. Post request takes 50ms or less. |  |
| 2b | Test 2 (Chrome) | Tester measures how long it takes Friends Only post to be added to the database and verifies its privacy. | 500-character long Community post with an image. | Post is visible by author account’s Friends using Relay. Post request takes 50ms or less. |  |
| 3b | Test 3 (Chrome) | Tester measures how long it takes Best Friends Only post to be added to the database and verifies its privacy. | 500-character long Community post with an image and 200-character long teaser. | Post is visible by author account’s Best Friends using Relay. Post request takes 50ms or less. |  |

1. **Code Review**
   1. **State the coding style we chose**

The coding style chosen is functional. The code has been separated between php files that incorporate html/ css/ js for the user interface and php files that house the functionally of a specific page. For example, the login.php file contains the elements that the user will see and interact with. The include\_login.php file houses the logic to check if the user has entered valid credentials. Having the code split between the user interface and functionally allows for easier error handling. It also allows the team to work more efficiently between the back end and front-end coding.

* 1. **Document Peer Review process for feature from Usability and QA test plans**

Code review was performed by the team leader and back-end lead. The specific code reviewed was regarding creating posts. PHP, CSS, JS files were presented in a zipped archive along with jpgs of the code. Having both the code and images allowed the back-end lead to look over and run the code. The code was tested and found to have no errors in logic or functionality. The code was then integrated to the latest release of the Relay application.

Graphical user interface, text, application, email

Description automatically generated

1. **Self-Check on Best Practices for Security**
   1. **Major Assets we are Protecting**

Account information, including passwords, names, and emails. Posts, by restricting access based on the poster’s preferences. Chatroom messages, by deleting them after a short timeframe to allow for any moderation.

* 1. **Confirm That Passwords are Encrypted in DB**

Passwords are encrypted and decrypted using PHP native password hashing API. After the password is validated and goes threw the sanitize function it is then processed using password\_hash($passd, PASSWORD\_DEFAULT). The PASSWORD\_DEFAULT algorithm is updated and changed automatically over time. This ensures that the latest technology is being used to secure passwords. The encrypted password is saved in the database upon successful new registration. When the user logins to the Relay application password\_verify is used to validated if the password is correct or not.

Graphical user interface

Description automatically generated

* 1. **Confirm Input Data Validation (list of what is being validated and what code we use to validate)**

Input data that is validated and sanitized includes usernames, passwords, firstnames, lastnames, emails, bios, chatroom messages, post titles, post teasers, post content, and search terms.

Validation code is given below, double click on the sample to see all the code.



User input is sanitized with the function below.

Text

Description automatically generated

This uses php’s [filter\_var](https://www.php.net/manual/en/function.filter-var) function with [FILTER\_SANITIZE\_STRING](https://www.php.net/manual/en/filter.filters.sanitize.php) as the filter and [flags](https://www.php.net/manual/en/filter.filters.flags.php) of FILTER\_FLAG\_STRIP\_LOW, FILTER\_FLAG\_STRIP\_BACKTICK, and FILTER\_FLAG\_ENCODE\_AMP. This removes html and php tags, Unicode characters with value < 32, backticks, and encodes ampersands, single quotes, and double quotes.

1. **Self-Check: Adherence to Original Non-Functional Specs**

|  |  |  |
| --- | --- | --- |
| Spec | Status | Notes |
| **Expected Load:** The system will be designed to support up to 30 concurrent users. | DONE | Nothing in our site limits the number of users. We simply make no guarantees about site performance after 30 users. |
| **Response Time:** The maximum response time allowed to the system for transactions will depend on both the number of concurrent users, and the type of transaction. At expected load of 5-10 users, all transactions should take no more than 50 milliseconds. At maximum capacity of 30 concurrent users, creating a new account, logging in, and creating a post should take no more than 500 milliseconds, searching for posts should take no more than 250 milliseconds, both sending and receiving chatroom messages should take 125 milliseconds for a total delay of 250 milliseconds between when a user sends a message, and the other participants receive it. Beyond 30 users, response times are not guaranteed. | ISSUE | Most response times are ON TRACK, the exception is with chatroom messages. We believe that the original requirement was far too ambitious due to a lack of knowledge at the time of the original proposal. We believe an achievable total delay for this feature to be 2500ms instead of 250ms. |
| **Reliability:** The final system will allow no more than 30 minutes mean downtime per month. | DONE | Currently we plan to update the site without taking it down. |
| **Ease of Use:** The system should be designed to be intuitive and require a few minutes at most for users to become familiar with its features. | ON TRACK | Currently implemented features have descriptive names and the site is easily navigable. |
| **Browser Compatibility:** The system will function on a minimum of Google Chrome and Mozilla Firefox. | DONE |  |
| **Privacy:** The system will comply with all privacy regulations in the locations it operates in | ON TRACK | A Privacy Policy and Terms of Service are being drafted using [privacypolicies.com](https://www.privacypolicies.com/). These documents will be available on the site and users will be required to confirm that they agree to them before creating an account. |
| **Login System:** Accounts will require an email and password to create and log in to. On account creation, users will be required to re-enter their password for confirmation. Accounts may also be secured with two-factor authentication that links to the user’s email or phone number | ISSUE | Password confirmation is in place. We do not think we will be able to do two-factor authentication at this time due to FAU’s LAMP server not allowing us to send emails. |
| **System Access:** All members of the development team will be able to make changes to the database, front-end code, and back-end code | DONE |  |
| **Site Host:** The system will be stored on our LAMP server on lamp.cse.fau.edu. Both the database and site files will be stored here | DONE |  |
| **Backups:** Backups will be stored on our project’s Github repository to avoid data loss in case of an issue with the LAMP server | DONE |  |
| **Space:** The system should not use more than 85% of the available space to leave room for bug fixes | ON TRACK |  |
| **Availability:** The system should be available whenever the LAMP server is available, minus any additional downtime. Downtime will be limited to an average of 30 minutes per month | DONE |  |
| **Exception Handling:** The system will have exception handling that will display explanations of the exception to the user. Users will be able to either try another input or redirected away from the page that had the exception | ON TRACK | Currently implemented features have descriptive exception handling and either re-prompt for input or redirect the user. |
| **Functional Requirements:** Tests will be devised and conducted for all functional and non-functional requirements | ON TRACK |  |