



Final Report

Course ID: CPS 5995: CIS Capstone

The Daily Roar Email Broadcasting System

Department of Computer Science and Technology, Kean University

Instructor: Dr. Jing-Chiou Liou

Students: Pankati Patel, Uko Ebreso, Alexander Fisher, Nicholas Moffa, Kevin Parra-Olmedo

Date: May 10, 2023

1. Abstract	3
2. Introduction	3
2.1 Handshake	3
2.2 Googles Gmail	4
3. Goals	4
3.1 Business Goals	4
4. Project Roles & Responsibilities	5
5. Language/Tools Used	6
6. Features	6
7. System Design	6
7.1 System Architecture	6
7.2 Architecture of Daily Roar Application	7
7.3 Technical Details Behind the Layers	7
7.4 Context Diagram	7
7.5 Use-Case Diagram	8
8. Methodology	10
9. Actors	10
11. Testing	11
11.1 Test Cases	11
11.2 Static Code Analysis	11
12. Security Applied	12
13. Current Status	12
14. Future Plans	12
14.1 Future Features	13
15. Duration & Issues	13
16. Insight	13
17. Conclusion	13
18. Acknowledgements	14

1. Abstract

The Kean University Department of Computer Science and Technology works to increase student involvement in events that contribute to career development. Events such as ACM, ACM-W, group advising sessions, visits by industry, other universities, and affinity groups such as AI4All, and WiCYS are announced to students through their university emails. Analysis shows current engagement with these emails could be higher, as only 30% of students view the email, and less than 5% of students interact with the emails by clicking on the embedded hyperlinks. As a result, students are not involved in the opportunities that are offered by the department. This project develops a new system to broadcast emails, it strives to increase the viewing rate of emails by 66% and increase link click-through rate by 100%.

This system seeks to alleviate students of unnecessary information, by allowing the department to broadcast emails based on student classifications. By allowing the department to broadcast to a specific demographic, students will receive opportunities that are personalized, which will raise the involvement rate. The system provides the department with analysis of each email broadcasted. They will be able to view the total number of students that have opened the email, and the total number of students that have clicked on the embedded hyperlinks within the email.

2. Introduction

The department broadcasts to over 700 students, opportunities such as alumni talks, company visits, or research opportunities are regularly announced to students. Job postings are shared with students weekly in the form of an electronic career bulletin. These emails are for students using two tools, the Handshake product and Google's Gmail tool, which manages the university's email system with the Google Workspace tool. The opportunities shared with students are often time-critical and necessary for professional skills development and alignment with internship and graduate opportunities.

2.1 Handshake

Handshake is currently one form of communication being utilized to broadcast emails to students and gather data on their engagement. Handshake is an online recruiting platform widely used in the United States that partners with universities to streamline and simplify the recruiting process. This system allows an academic department at a university to broadcast emails to users who have an active account on Handshake. Emails can be delivered to students within defined categories by choosing filters like major, class standing, or degree program. Employers work with Handshake to distribute internship and job openings nationwide. Statistics on the percentage of students who open an email message, and the percentage of students who click on the hyperlinks embedded within the text of the email can be gathered for every email broadcasted. Analysis gathered from Handshake for this case study showed that student engagement with the broadcast emails was low, as only 30% of targeted students opened the email, and less than 5% of the students clicked on a hyperlink in the email. Broadbase email communication as a method is only successful if all students within the department have an account on Handshake, and if the account is actively updated by the students as they

move up class levels. Handshake does not detect if a student has completed a requirement for credits to categorize them within a class level. Hence, if a student fails to update their account they will be distinguished within a misleading category of students.

2.2 Googles Gmail

Another form of communication being utilized to broadcast information is Gmail. Students' university emails are used to broadcast information to all students enrolled in Computer Science (CS) and Information Technology (IT) programs. Although Gmail does offer a way to apply filters on the recipients, it is not automated. Each filter needs to be manually created and the recipients need to be manually entered. Consequently, students start accumulating many emails that may not pertain to their demographic, as some opportunities are beneficial for a particular category of students. Gmail also does not give statistics on the student engagement with emails, if they viewed the email, or if they clicked on embedded hyperlinks. In order to gain these statistics the sender can request read receipts on Google Workspace accounts. This method comes with limitations, as the recipient has the authority to enable and disable their read receipts, this can skew the statistics.

This system seeks to alleviate students of unnecessary information, by allowing the department to broadcast emails based on student classifications. These classification include class level (freshman, sophomore, junior, senior), major (CS,IT), and degree program (undergraduate, graduate). By allowing the department to broadcast to a specific demographic, students will receive opportunities that are personalized. This system also provides the department with analytics containing information regarding the percentage of emails viewed and the percent of interaction with the emails. Multiple drafts of emails will be composed and broadcast to a category of students. Data will be gathered on the emails viewing and interaction rate to analyze which emails gain the most attraction within a given demographic.

3. Goals

3.1 Business Goals

Increase the viewing rate of emails by 66% and increase job posting link click-through rate by 100%.

3.2 Project Goals

4. Project Roles & Responsibilities

Roles and responsibilities for this development were categorized and divided accordingly:

Role(s)	Responsibilities	Participant(s)
Project Manager	<ul style="list-style-type: none">▪ In charge of managing the project itself and the members involved in the team working on the project.▪ In direct communication with Supervisor (Professor) about project progress▪ Skilled in problem solving▪ Handles planning, scheduling, and budgeting▪ Planning out the ‘blueprints’ for the project. Defining the scope, setting deadlines, laying out communication strategies▪ Ensure the project stays on schedule▪ Track milestones, deliverables, and changing requests▪ Communication with the stakeholder	Pankati Patel / Uko Ebreso
Back-End Developer	<ul style="list-style-type: none">▪ Have a good understanding of the user requirements▪ troubleshoot and debug applications▪ collaborate with front end designer to integrate UX for server side logic▪ Software Design Document (SDD)▪ Adaptable to changing requirements	Alex Fisher & Uko Ebreso / Pankati Patel (secondary)
Front-End Developer	<ul style="list-style-type: none">▪ Software Design Document (SDD)▪ Adaptable to changing requirements▪ troubleshoot and debug applications▪ collaborate with back end developer to integrate back end functions with front end interface and functions▪ Create quality prototypes	Alex Fisher / Kevin Parra(secondary)
System Analyst	<ul style="list-style-type: none">▪ Update the documentation of the Project as it moves along through each of its sprints▪ Conduct research on possible solutions for existing issues and make recommendations to the project manager.▪ Suggest designs, tests, and implementations for the system.▪ Train users on how to efficiently use the system by creating a how-to manual	Kevin Parra / Nicholas Moffa (secondary)
Tester	<ul style="list-style-type: none">▪ Create and run test cases for the project to make sure that everything is running smoothly before and after software release▪ Make sure there's aren't any bugs within the system▪ Provide solutions for defects found	Nicholas Moffa / Alex Fisher (secondary)

Figure 1: Roles & Responsibilities Table

5. Language/Tools Used

Numerous coding languages and tools were utilized for the creation of this system, as categorized in the lists below.

Front-end:

HTML, CSS, NES.css, Bootstrap, JavaScript, jQuery, Quill Text Editor, Rich Text Editor

Back-end:

MySQL (MariaDB), PHP

Software tools used:

Github, Visual Studio Code, MySQL Workbench, DBeaver, PuTTY, XAMPP

6. Features

The Daily Roar system has the following features:

- Uploading a CSV File to populate database tables (Available for Faculty and Graduate Assistants)
- Login system with a “Forgot Password” functionality (Available for Faculty and Graduate Assistants)
- Compose an email and send it to a group of students based on a selection of filters (Available for Faculty)
- Manually insert a student into the database (Available for Faculty)
- Update and/or delete students from the database. (Available for Faculty)
- Create a new faculty or graduate assistant account. (Available for Faculty)
- View sent emails and their statistics (total recipients, open rate, link click-through rate) (Available for Faculty)
- Receive and interact with emails. (Available for Student recipients)

7. System Design

7.1 System Architecture

As a web application, this system follows a client-server model with three-tier architecture. The three layers within the architecture are the Presentation layer (Client), the Application layer (Server) and the Data layer (Database).

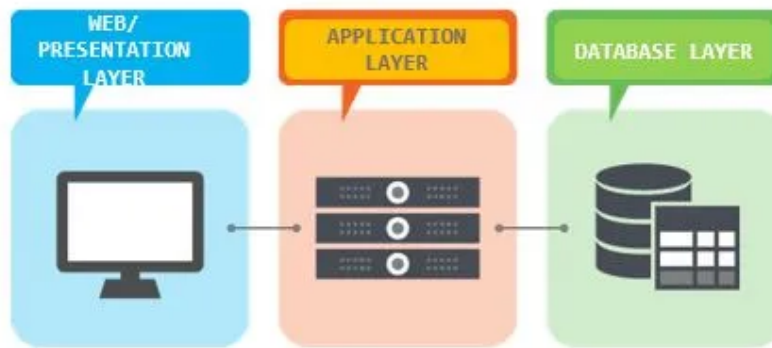


Figure 2: Three Tier Architecture

7.2 Architecture of Daily Roar Application

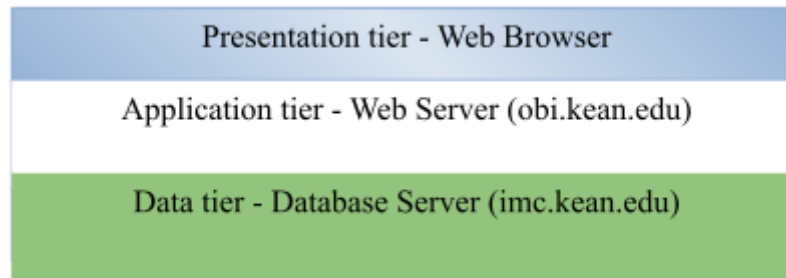


Figure 3: Architecture of Daily Roar Application

7.3 Technical Details Behind the Layers

1. Presentation layer (web pages)
 - a. Built using: HTML, CSS, Bootstrap, JavaScript, jQuery
2. Application layer
 - a. Built using: PHP, Apache Web Server (hosted on obi.kean.edu)
3. Data layer
 - a. Built using: MySQL (hosted on imc.kean.edu)

7.4 Context Diagram

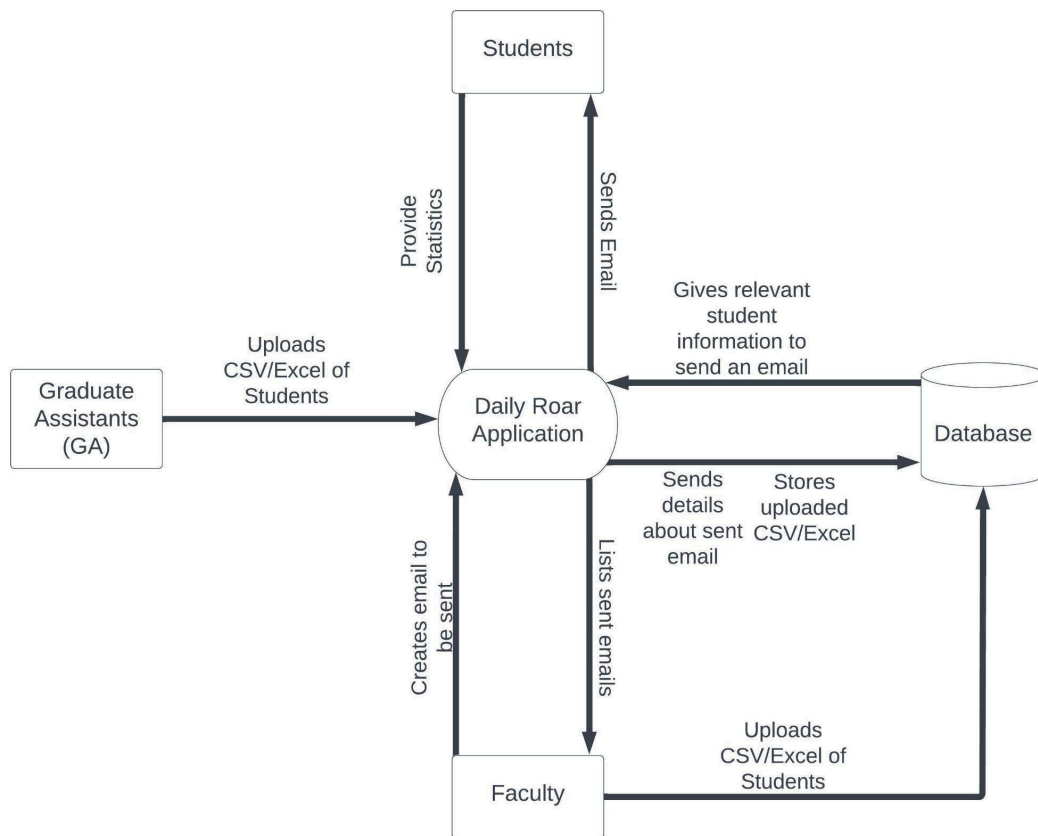


Figure 4: Context Diagram

7.5 Use-Case Diagram

To give the system a visual representation, use case diagrams are made. They are also used to depict how the system is used by its users, who in this case are the staff, graduate assistants, and students from the department of computer science and technology at Kean University. Each type of user and their interactions with the system were represented by a different diagram.

Users who will interact with the system are portrayed by actors. Arrows pointing to the ovals that stand in for the specific tasks that could be performed by each user show the tasks that they could perform. The tasks are contained within the sizable rectangle that is the system itself.

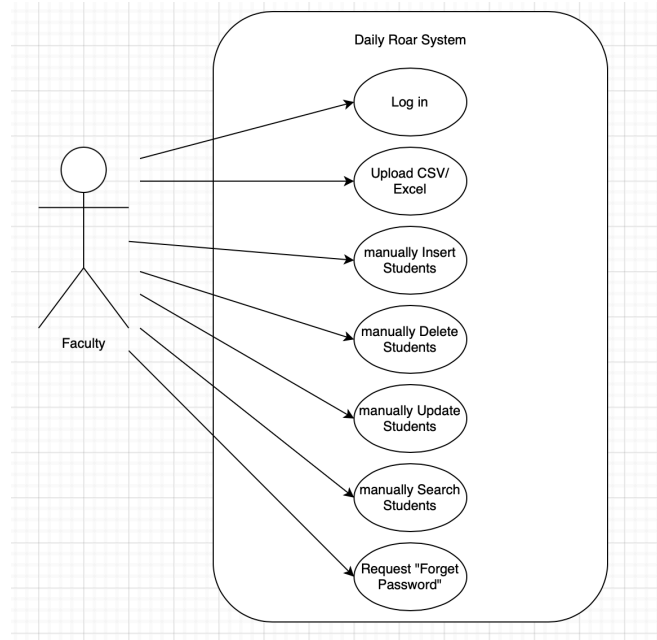


Figure 5: Faculty Use Case

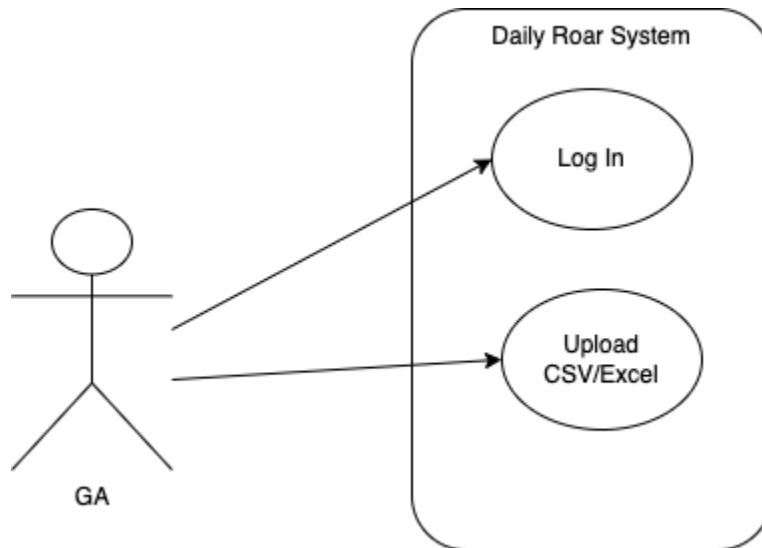


Figure 6: Graduate Assistant (GA) Use Case Diagram

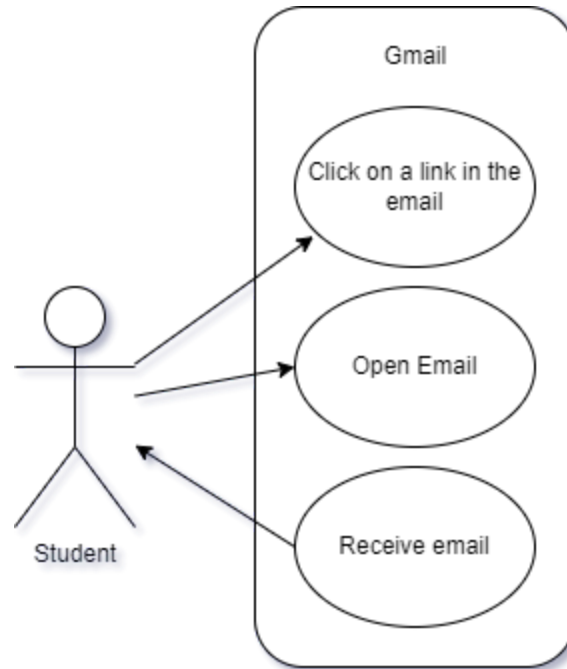


Figure 7: Student Use Diagram

8. Methodology



Figure 8: Agile Methodology

For this software project, as a team we decided to adopt the Agile Method to go through the multiple phases of the development life cycle. The agile methodology can be defined as a way to manage a project by breaking it up into several phases. It involves constant collaboration with stakeholders and continuous improvement at every stage. With this being said, after our adoption we organized the project into Two large sprints where we planned to go through the Software Development Life Cycle (SDLC) within the Two large sprints. We set due dates for each sprint and presented the progress to our instructor for feedback on said progress. The Agile method involved continuous integration, testing, and deployment.

The project benefits from the agile development method because from the very beginning the requirements for the project were made clear to us based on what the user wanted. Also providing flexibility for the team to make quick changes in case of anything. Following through with the phases helped the team keep everything organized and understandable for when it came time to meet the goals and present them by the time the phase came to a close.

9. Actors

Within this project, there are 3 actors involved:

- **Kean University Staff:** Primary projected users of the Daily Roar systems. Will have the ability to upload CSV files, send mass email broadcasts, create new users, view students within the database and monitor email data.
- **Graduate Assistants:** Secondary projected users of the Daily Roar System. Will have the ability to upload CSV files.
- **Student Recipients:** Primary target demographic of the Daily Roar System. By opening sent emails, students will be able to read emails and their interactions will compose the data used for email analysis. Students from the Department of Computer Science and Technology at Kean University from the class level ranges of Freshman to Graduate were used.

11. Testing

11.1 Test Cases

This is an example of one of the test cases for the Daily Roar System. This test case encompasses the login functionality.

Login Test Case:

Pre-condition

1. Have access to the internet
2. Login page should be displayed when going to the website

Test Case #	Test Summary	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Going to the login screen website	Access the login screen	URL should work	When the url is inputed it should navigate to the login screen	The URL goes to the login page	Pass
2	Enter a username and password	Should display username and password input	Username and password should be displayed	Once the username and password is inputed it should display what exactly the user typed out	The inbox worked for username and password and displayed what was typed	Pass
3	Try to login with wrong username or password	Enter wrong username and password	The wrong username and password should be set	If the username and password is wrong it shouldnt let you log in	Not able to login with the wrong credtials	Pass
4	Display error message	When a username or password is incorrect an error message should be displayed	Input wrong username or password	An error message should be displayed if password or username is incorrect	An error message appeared when the password or username was incorrect	Pass
5	Try to login with the username or password blank	With one of the inputs blank it should display a "fill out this field"	One of the inputs should be left blank	With one of the inputs blank it should display a "fill out this field" and not be able to login	A "fill out this field" showed up when tried to login with one of the fields blank	Pass
6	Try to login with correct username and password	Enter the correct username and password	The correct username and password should be in the input before the "Login" function is clicked	Once the correct username and password is inputed click the "login" function and should take you to another page logged in as the user	The username and password worked and was able to login	Pass

Figure 9 & 10: Test Case Precondition and Summary

- Post-conditions - Pass
 - User shouldn't be able to login if any of the credentials are empty or incorrect
 - Once the user clicks login it should check with the database to see if it matches.
 - If it does then the login should take user to the respective homepage
 - If it doesn't then an error message should be displayed

11.2 Static Code Analysis

We used the tools Fortify and Snyk to examine our code and see if there were any debugging issues without executing the program. When we did this we found a few problems with our code which were cross scripting and SQL injection. With this we made the fixes that we needed so there weren't any issues with our code.

12. Security Applied

Aspects related to security were implemented into different components of the project, primarily in the password portion. When creating a new user, after the user enters the new data (username and password) for the account, the information is salted and hashed to increase security. If a new password is requested, the user is then prompted to enter an email address in the system and if applicable, a temporary code (lasts 10 minutes) is sent to the corresponding

email address with a link to the password reset page. After entering the code within the allotted time, the user is allowed to set their new password and confirm it by entering it twice. Once submitted, the new password is salted and hashed before being entered into the database.

13. Current Status

The current status of the system is that the system is fully functional, The requirements handed to us at the beginning of this project have since been reviewed and implemented onto the current system. This includes the ability to send emails as well as check the statistics of them, the system also offers a login system for users to log in and out of accounts to access the system. All of the other requirements such as manually inserting students, inserting new users, modifying said students, and uploading the student CSV file have all been implemented and are now ready to use.

As of now the project is complete when speaking in terms of what was required from us since the project began. Anything else from that will be placed in the future plans section of this paper. Aside from the main requirements The retro themed style of the site was also implemented and ready for use. This includes the animation of the site as well as their forms and buttons are fully functional.

14. Future Plans

In the future we are going to be implementing a draft function for later editing. We realized that if someone was creating an email and they needed to exit out of the program for some reason there was no way to save what has already been written in the email. Another thing we are going to implement is a progress bar in upload CSV and send email. We would like to do this to show the progress of the CVS uploading and the sending emails. One last thing we would like to implement is a link tracking for individual link opportunities on each individual email.

14.1 Future Features

There are still plans for the future as stated previously, There are many things that the team plans to add to the system. These include:

- Progress bars to give visual representation of progression of the system.
- Draft system for emails that can be started saved for later as well as sent later at another time.
- Tracking system for individual links on individual emails.
- Integrating a more efficient rich text editor when composing emails.

15. Duration & Issues

The duration of this project was three months and two weeks. We started from the week of January 25th, 2023 to the week of May 10th, 2023. Throughout this time was the time used for the construction of this project.

Some issues that the team encountered while developing this project was the implementation of the schema for the database. As we came to find out, the database for this project was much more complex than we expected so designing the schema for it took a little longer than expected. Another issue was creating the email system that would be able to send mass emails to all of the students in the department. However, although we did have some issues they were very manageable and not difficult to resolve, more so time consuming but not majorly difficult.

16. Insight

There was much learned by the team while developing this project some of those include:

- How to apply the Agile Methodology to our project
- Organizing and documenting everything that was intended and planned for the project.
- We learned how to work as a team and stay in constant communication whether virtual or in person to make sure we were all on the same page.
- We learnt various methods of testing our software as well as creating and using test cases to make sure everything was performing as it should.
- We learned how to think outside of the box and foresee how the system needed to behave in the future so that we can make sure that it will hold once passed over to the university.
- We learned how to prepare everything at an organized rate of time making sure goals were met at the correct times.

17. Conclusion

In conclusion, faculty accounts are able to send mass emails to specific students to show job opportunities and important information throughout the computer science and information technology department. We are also able to see how many students opened the email that was sent and how many students interacted with the email and clicked the links that were sent. The faculty user is also able to upload students CSV files, update student information, manually insert students and create new users. Then the other user, GA only has access to upload CSV files to the database.

With all this, the system was mainly created to be able to track the interactions of students and their emails. The reason for this is so the staff can observe patterns of the emails to

see which gets more attention and which does not. With this information staff can set up emails that get the most attention from students. The objective of the system is to pass along opportunities to the students of Kean University and with the help of the Daily Roar this will now be possible, giving said students all these opportunities for their benefit.

18. Acknowledgements

We would like to thank Professor Jing-Chiou Liou for his guidance and motivation throughout this semester. We would also like to thank Nohelia Diplan for her feedback as an end-user and as a stakeholder for the Daily Roar.