



Team Ally
Michigan State University
Ally Employee Recognition Platform
Project Plan
Fall 2022

Ally Contacts

Jared Allmond
Dzmitry Dubarau
Dallas Gale
Catherine Jardines
Harish Naik
Theresa Weaver

Michigan State Capstone Team

Phumapiwat Chanyutthagorn
Harrison McComb
Abby Peterson
Gatesana Vongphachanh
Phillip Yu

Table of Contents

Executive Summary	3
Functional Specifications.....	4
Design Specifications	5
Overview	5
Account Information	5
Home Page Dashboard	5
Sending Recognition.....	6
Admin Page	12
Technical Specifications.....	13
System Components	15
Risks Analysis	17
Schedule	19

Executive Summary

Headquartered in Detroit, Michigan, the leading digital financial service company Ally is one of the top 25 financial holding companies in the United States. Ally offers financial products for consumers, businesses, automotive dealers, and corporate clients. They constantly advocate for a culture of experimentation which enforces a desire for world-class products as they develop platforms and digital experiences for a variety of needs. Ally has committed to researching and evaluating new and emerging technologies. They work to identify opportunities which accelerate new technology adoption.

As a company, Ally takes pride in their organization culture. According to CEO, Jeff Brown, “We’ve strived to be an open, diverse and inclusive culture for years, and I’m proud of how we continue to develop to Be (Even) Better and move us forward.” Ally is committed to learning from one another’s differences to make the company stronger overall. They believe that the ability to influence change comes from the culture, people, and community that is found within this business. Thus, Ally plans to put such company values at high notice so that their employees and business continue to flourish from the positive impact of a healthy culture.

To achieve this goal of fully supporting their organization’s culture, Ally plans to implement a new application that is focused on boosting employee morale in the workplace. Using a web-application, Ally employees will be able to receive positive reinforcement from their peers and managers across the company. Employee recognition has been proven to improve morale which in turn boosts productivity at work. This web-app, Ally Kudos, will offer an online system of recognizing those of the Ally community in a real-time, user-friendly way. By sending and receiving congratulatory “kudos”, people within their organization will feel more seen, appreciated and in turn will be more likely to enjoy their job and do it to the best of their abilities.

Functional Specifications

Workplace recognition allows employees to see that they are appreciated, and they are valued for their hard work, which will in turn boost confidence and performance. For their recognition to be well-received and impactful, it should be given in a timely manner, and it should be built upon the values of the company. This web application allows for an efficient and accessible way to show thanks to one another, bolstering the cycle of positivity and workplace cohesion.

This project has five main focuses: great employee experience, creating lasting memories, driving peer-to-peer recognition, time recognition, and boundaryless recognition.

Each user will be able to send each other personalized messages that are fun, creative, and curated for their needs. In addition, users can attach kudos, which are credits that can be sent alongside these messages. They will be able to view their kudos statistics on their “Recognition Wall”, a personalized and easy-to-navigate dashboard to allow for an exceptional employee/user experience. Another addition to the Recognition Wall is a history of all the memorable rewards and messages that the user has sent and received.

Recognition can be given to anyone, from anyone; this no longer belongs only in the realm of management. Allowing employees to interact with other employees and managers completes the goal of peer-to-peer recognition. Messages and credits can be sent at any time, regardless of position or designation, solving the issue of time and boundaryless recognition.

Design Specifications

Overview

Ally Kudos is a web-application to be used by Ally employees to send and receive recognition from their coworkers. All users have access to the basic features of the application, and the users classified as “manager” and “admin” have access to other features to review the recognition of the employees who report to them. The intent is to boost morale and spread positivity throughout the workplace with simple but effective messages, and the UI is designed with this in mind. This web-application is compatible with computer screens as well as mobile devices.

Account Information

When the user first opens the web-application, they must log in to their account. This application will not be public as it is intended for internal use only. Ally members will have the option to create their account initially and log in with their set password thereon out.

Home Page Dashboard

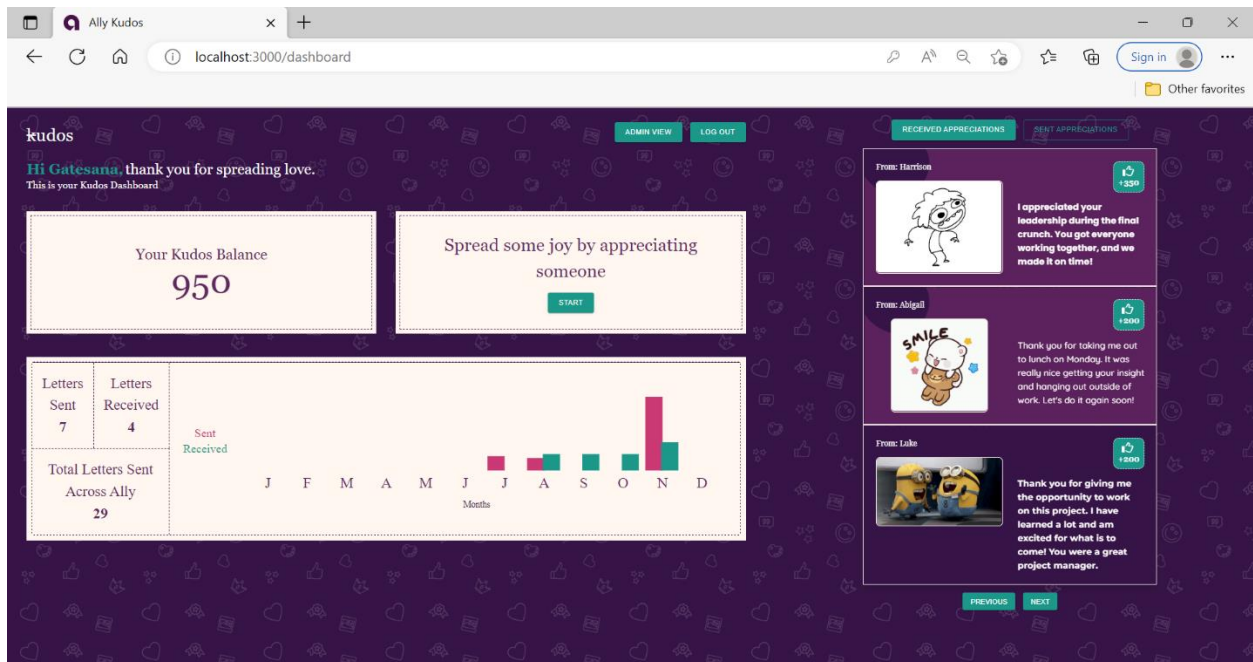


Figure 1: Employee Dashboard

Figure 1 displays the homepage with information about the logged in user. On the top middle of the screen is the log out button, which the user will click when they wish to be redirected to the log-in page, and the admin page button, which only appears if the user is an admin and wants to view other statistics. On the middle of the screen, there is a section which showcases the user's kudos total. The "statistics" section below has a graph that the user can look at to compare how many kudos they have sent to and how many they have received from their coworkers over a monthly timeline. To the left of the graph, the user's totals are shown along with the total amount of letters sent throughout the whole app. The rightmost division of the dashboard has buttons on the top so the user can switch between viewing their most recent kudos received and kudos sent. The user can click the next or back buttons to view their history of kudos. To send kudos, a button is placed near the middle of the screen as the central feature of the application.

Sending Recognition

Pressing the "send kudos" button on the dashboard starts the flow of sending a message of recognition.

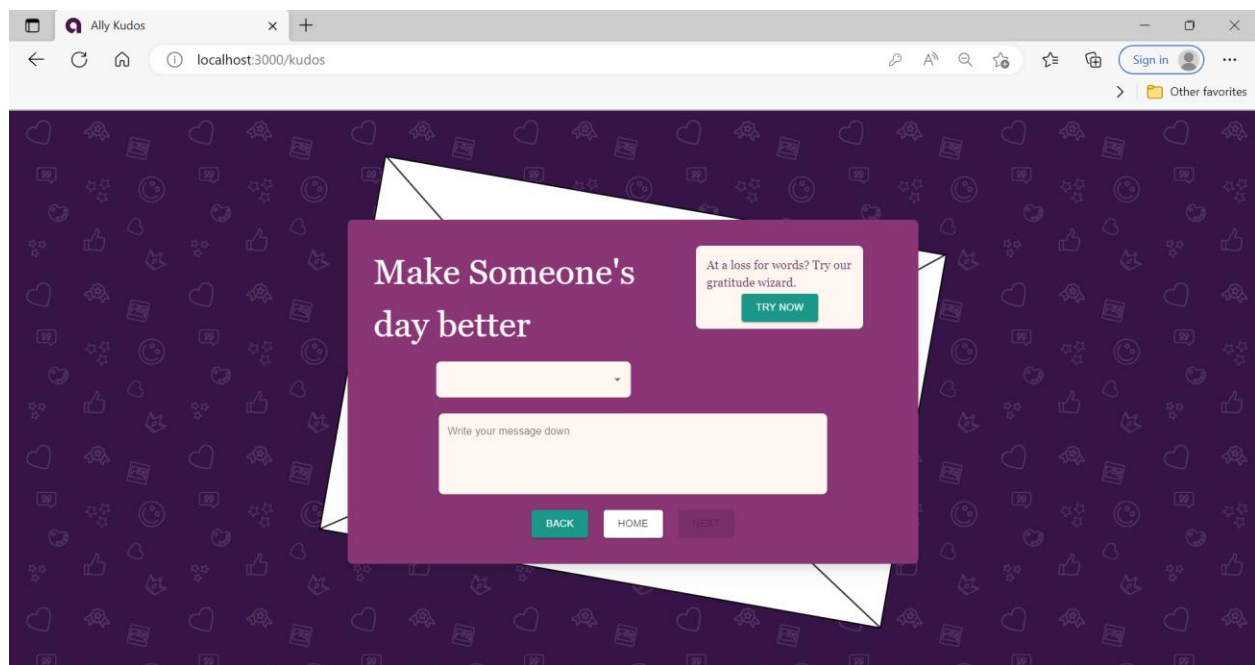


Figure 2: Blank canvas for sending recognition

The UI for this feature portrays a letter, as shown in Figure 2, as a way to lighten up the task of sending messages by adding personality and sentiment of the past form of communication, writing. Now, the user can draft a letter to their coworker in a time-

efficient, fun way. To send kudos, the sender is required to choose the recipient and write a message of appreciation to then hit “next” for further customization such as adding a gif, determining the amount of kudos points to send, or changing the font style.

The “gratitude wizard” option is included to help the user create a message if they do not have one to write manually. When the user presses the “try now” button, they begin a stream of screens of the different steps to constructing the message.

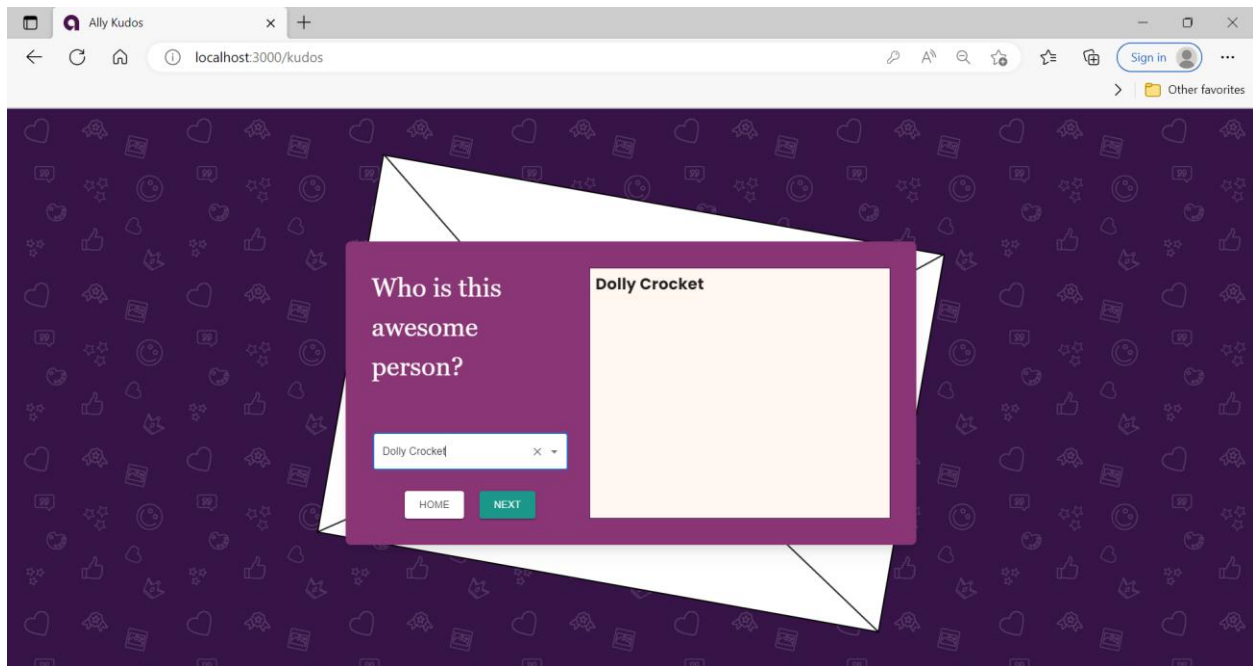


Figure 3: Writing the name of the receiver

The first step, as shown in Figure 3, is to write the name of the employee receiving the kudos. If the user is recognized to be in the database of employees, the message on the left will automatically fill in the employee’s first name. Clicking the “next” button continues the process.

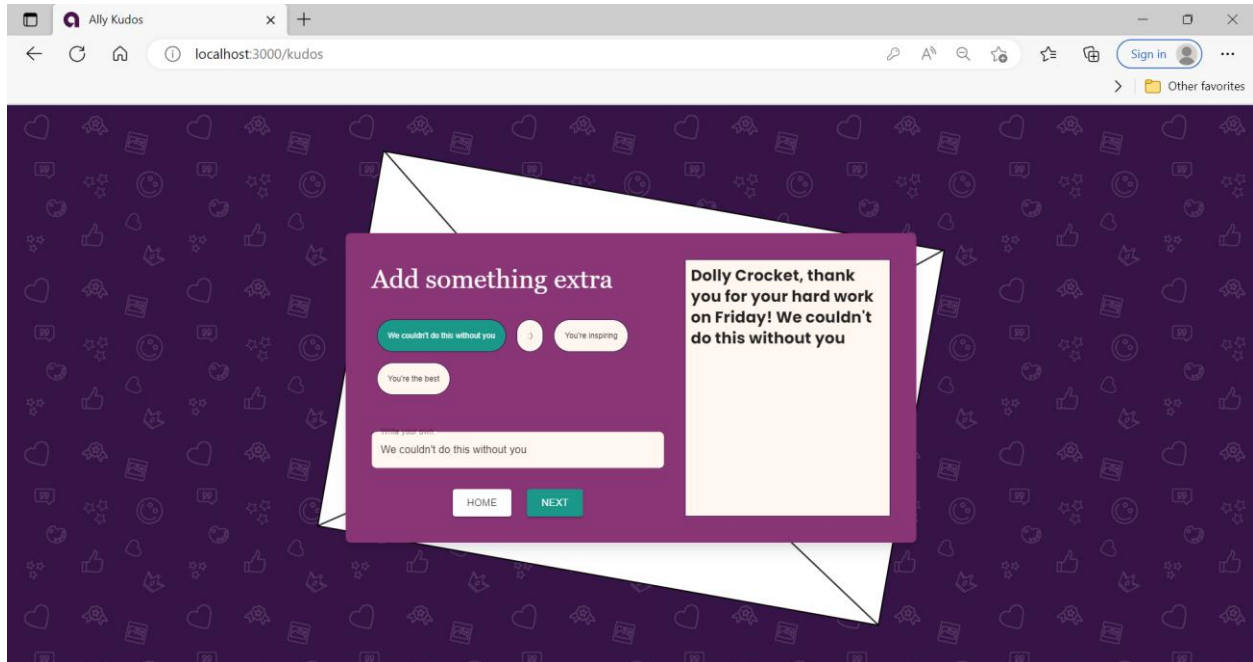


Figure 4: Words are generated for the user to pick to add to their message

Seen in Figure 4, the second step is composing the message in a “Mad Libs” style. Words will be generated based on the occasion and topic which the user can click on to fill in the message, or the user can write their own topic if they require something specific. The “add more” button will generate more options to lengthen the message, otherwise the user can click next to move on to the next step.

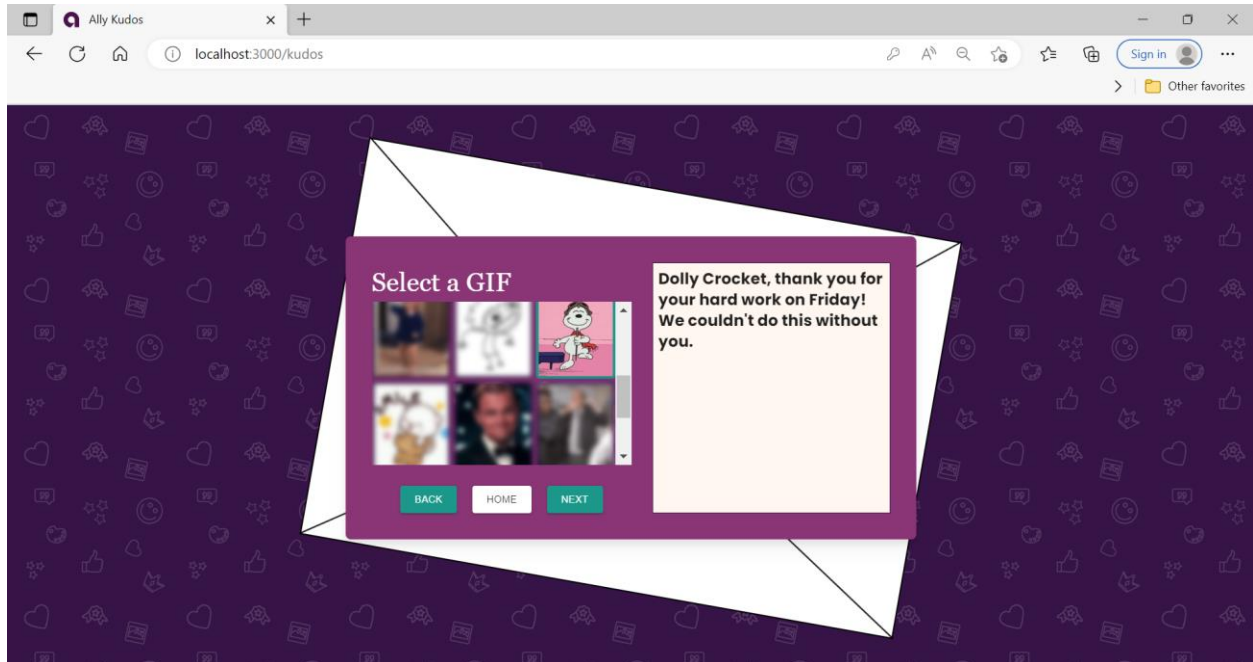


Figure 5: Gif selector

For the third step, the user can select a gif as shown in Figure 5. Upon choosing a gif, the user can easily remove it by click on that same gif or can click on a different one if the user changes their mind.

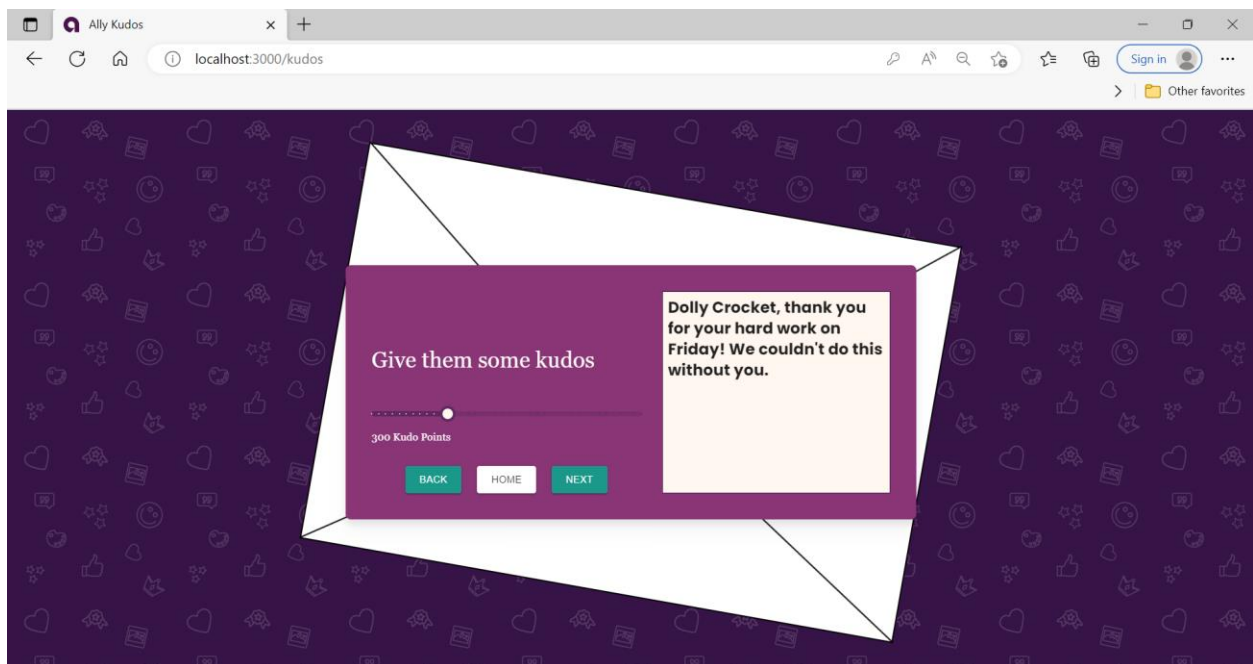


Figure 6: Kudos points selector

Going to the fourth step in Figure 6, the user is given a slider to select an amount of kudos points within the range of 25-1000.

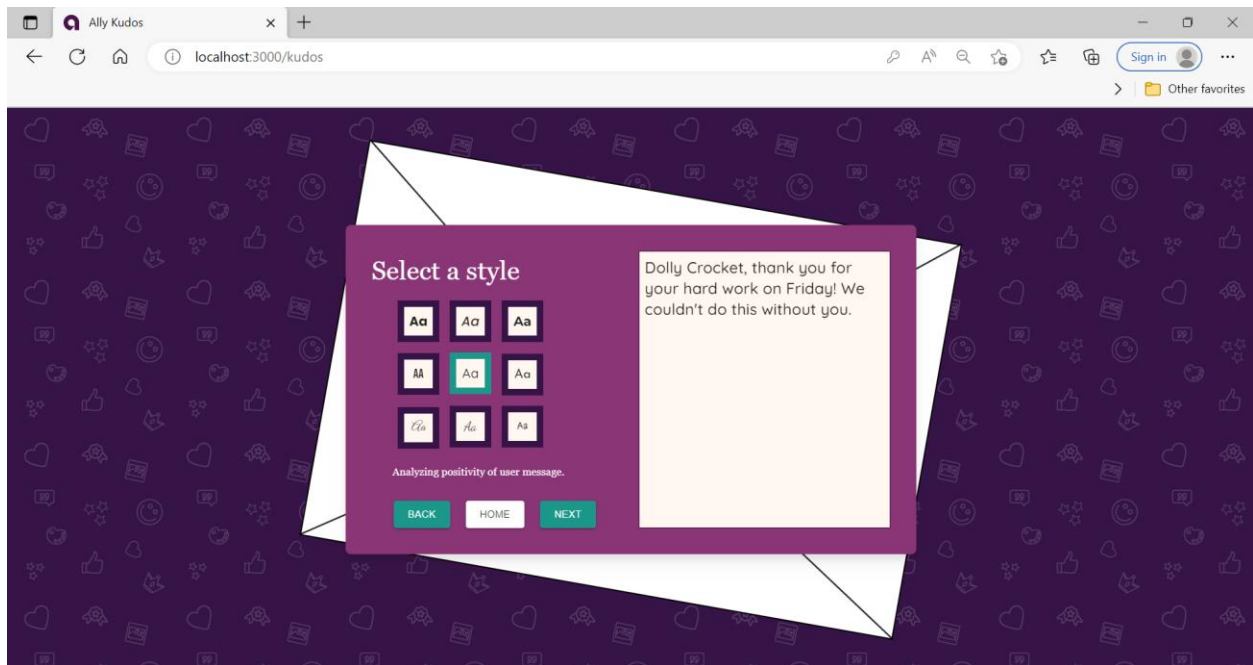


Figure 7: Next the user can pick the font style

Light customization is shown in Figure 7 in the form of changing the font style. After pressing the next button here, the message is analyzed for its positivity. If the message is determined as negative, the next button will be disabled, and the user will not be able to send the message. Instead, if determined as negative, the user will be notified that changes to the message are needed, If it is determined as positive, the user can move on to review the message.

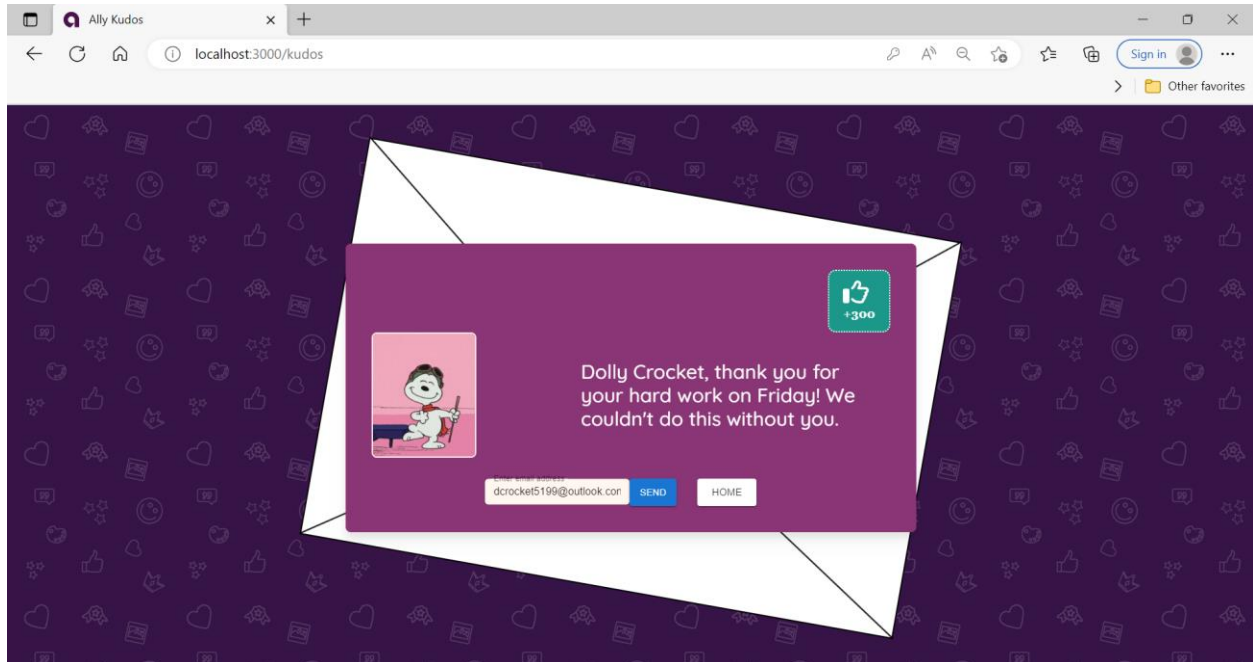


Figure 8: Final step to send the message

Once the kudos is finalized, the user can send the message through the receiver's email which is auto populated in the send text bar. The message is also automatically sent through the web application which is then seen in the sender and receiver's history side bar.

Each step of the kudos will present two buttons, one to quit the process and go back to the dashboard as well as one to take the user back to the previous step. These buttons will ensure that the user can be guided through all possible transitions while using the web-app and sending kudos.

Admin Page

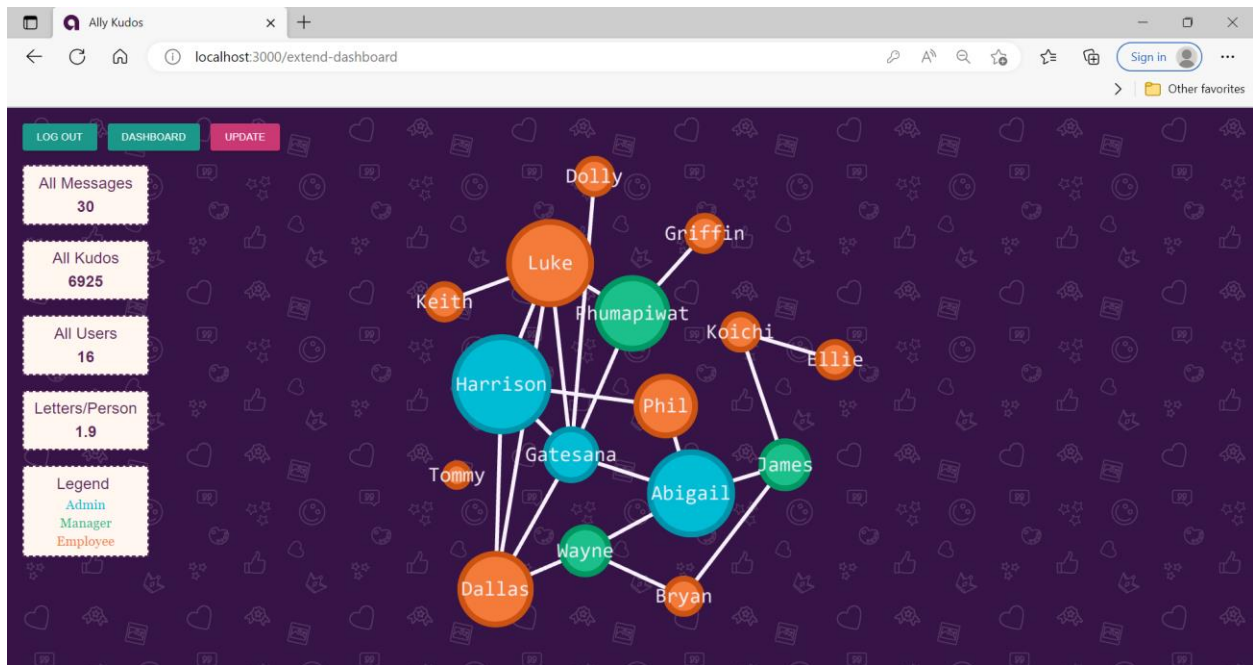


Figure 9: Graph of users and who they've interacted with

Users with the manager or admin status have access to the admin page which gives the user access to more company statistics as shown in Figure 9. On the side, the user can see the number of messages sent through the entire app, the number of kudos points, the number of users, and a ratio of letters per person. Below the stats is a legend explaining the colors for the graph in the middle. The graph displays every user represented as a node with the name of the user. The nodes grow and shrink depending on the number of letters the user has sent and received. The lines represent who has interacted with whom. The graph can be zoomed in and out and dragged around to analyze it.

Technical Specifications

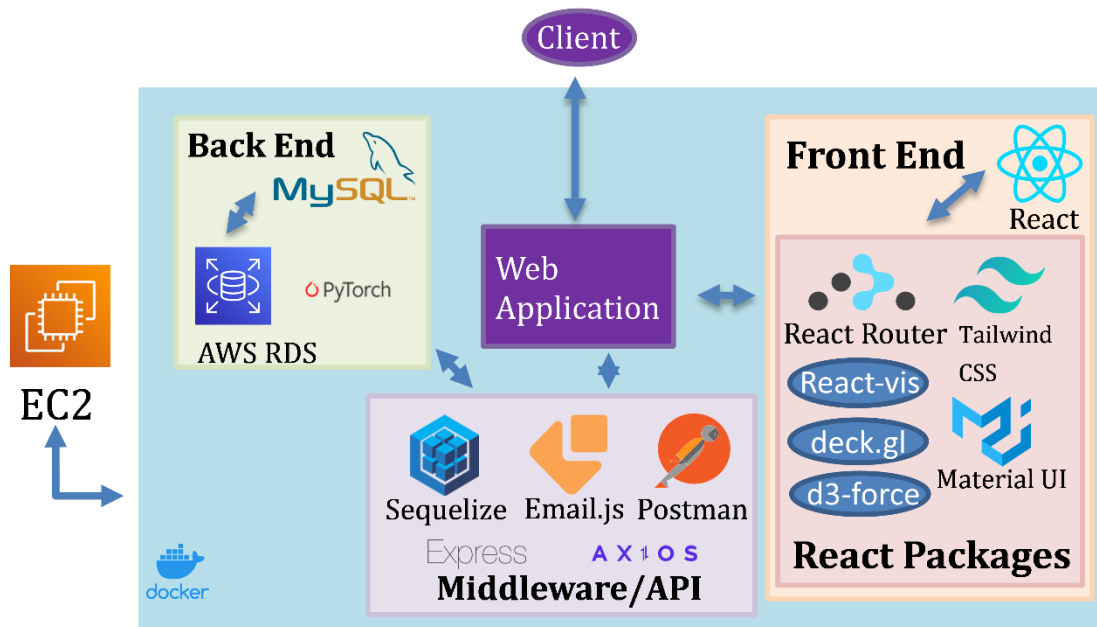


Figure 10: System Architecture

“Ally Kudos” is a web application built on React, JavaScript, HTML, CSS, and Node.js. Front end development is assisted by multiple libraries and frameworks installed as packages. Material UI provides predefined components and expedites the process of creating new ones, Tailwind CSS aids in efficient and clear styling, and React Router enables effortless page navigation and routing with components that inherit the behavior of their parents.

The middleware, which allows for communication between the application and the data in the Back End, will include Axios, Express, Flask, and Postman. Axios will be implemented to allow the creation of routes or APIs. Express acts as a router to connect our front-end calls to our back end. Flask helps to connect the machine learning python functions to our front end. Postman is an API tool and platform that allows developers to model, create, build, and test endpoints and custom-built APIs with various predefined request types.

The database and machine learning resides in the Back End, which will be built on AWS RDS and PyTorch. RDS is an AWS-managed version of a relational database service that supports MySQL, which is used for structured data. This is useful for storing user information such as usernames, passwords, etc. The goal for machine learning is

sentiment analysis: the act of identifying and categorizing the tones behind words.

PyTorch will be used to implement a message builder, which will analyze a message and deem it positive or negative, suggest words and prompts that match the sentiment to complete the user's thoughts, as well as suggesting the number of points or kudos to be attached based on the message generated.

System Components

Web Application

- React
 - Is an open-source front-end JavaScript library developed and maintained by Meta, which facilitates the creation of user interfaces based on custom UI components and handles state management.
 - Material UI
 - An open source React component library. Material UI provides a collection of predefined components that can be used out of the box or serve as templates for custom components.
 - React Router
 - A popular routing framework for React. It enables page navigation, multiple route definition, and route recognition for redirection. When a route component is implemented as a child of a UI component, the UI component's functionality is extended.
 - React-vis
 - An open source React component library and data visualization tool that generates and animates a wide variety of predefined graphs based on user data.
 - D3-Force and Deck.gl
 - Visualization libraries used for the administration page's graph. D3-Force calculates node locations based on established forces. Deck.gl offers layering functionality so that the nodes, edges, and text layer can all render as the graph.
- HTML
 - The default markup language for representing documents in web applications. They are often supported by style sheets and scripting languages.
- CSS
 - A style sheet language which complements HTML.
 - Tailwind CSS
 - An open-source "utility-first" CSS framework that enables efficient and effective custom styling by mixing and matching a list of CSS

classes. Custom components built with Material UI are stylized with the predefined classes Tailwind provides.

- MySQL
 - Server-side software that manages relational databases.
- Amazon Web Services
 - A widely used cloud platform, providing useful software tools.
 - RDS
 - Amazon's relational database service that supports MySQL and other querying languages.
 - EC2 (Elastic Compute Cloud)
 - A service that creates virtual machines; the web app will be deployed and run off a server on an instance.
- Axios
 - A JavaScript library for React that is used to make HTTP requests.
- Express
 - An API that acts as a router so that both the React application and the back end can be connected.
- Sequelize
 - A promise-based object-relational mapping that allows for easier querying into the database.
- Postman
 - An API tool and platform that allows developers to model, create, build, and test endpoints and custom-built APIs with various predefined request types.
- PyTorch
 - An open-source machine learning framework based on the Torch library, used for applications such as computer vision and natural language processing. It will be used to implement the Machine Learning aspect of the project such as the message wizard and points or kudos prediction based on the message content.

Risks Analysis

Implementing machine learning to help the user create a message as well as recommending the number of kudos for the generated message

Difficulty: Hard

Description: There is a section on the website when a user wants to send out any points/kudos to their peers, they will be asked to write a message alongside it. Machine learning will help the user formulate a message in a simple and enjoyable way to reduce the friction of effort. Another aspect of machine learning will be used to make predictions of kudos based on the message's appreciation through sentiment analysis. The team is unfamiliar with machine learning and are unsure of where to start for this.

Mitigation: Team members have been delegated to start research and create a prototype/test of the desired functionality. We will ask the client for their guidance in this area and other specifics for what they desire to have in the web-app. By developing prototypes and talking through use cases with our sponsor, we will be able to decipher what aspects of machine learning to further develop and how.

Making real-time interaction of notifications and updates between users

Difficulty: Medium

Description: This project offers the capability of real-time updates whenever kudos are sent and received alongside messages. We are unsure of the process, efficiency, or timing of when these notifications show up or how messages should be best received on the web-app.

Mitigation: Two of the team members have taken a class in web application and have some experience in making live messaging and notifications. The plan we have is to review that old class material, research other practices from outside and receive guidance from our client. Then we will develop a prototype and identify problems as well as any extra needs for this component's functionality within the next two weeks.

Finding the balance between privacy and public recognition

Difficulty: Low

Description: The goal of Ally Kudos is to make the user feel recognized through other users' appreciation of their help through kudos messages; however, there is a fine line we need to draw between being able to share the recognition given while not thoroughly exposing themselves to be judged or compared to someone else.

Mitigation: We plan to add options to share the message through various means such as email, Slack, or as a downloadable image file, but it will not be a requirement to share any kudos a user receives. We also plan to work with the client and develop other solutions to help with this problem as we make new discoveries throughout our project development.

Mitigation of web-app misuse

Difficulty: Hard

Description: There is the potential for misuse of the web-app. For example, inappropriate language, complaints, and threats can all be sent via kudos messages. While we view this scenario to be unlikely, we need to figure out a countermeasure to prevent this.

Mitigation: Currently, we are discussing ways to mitigate the improper use of messages that would lower company morale and harm the goal of the project. While it cannot be completely erased, we hope to at least persuade or add friction to make misuse harder. We will be running scenarios by both our team manager and sponsor to develop the cases of misuse that are most important to mitigate. We will then produce some sort of censoring/flagging API to counteract these messages.

Schedule

Week 1: Start 9/1 - 9/4

- First team meeting, establish weekly meeting time
- Introductions and going over proposal and adjacent emails
- Assign client contact (Phil Yu)

Week 2: 9/5 - 9/11

- Share rough draft of Project Plan document with team members
- Analyze Project Proposal as a team and draft UI/UX ideas
- Set weekly Triage meeting with Team Manager (TM) Luke
- Recognize general risks and delegate roles within team
- Meet with Sponsor to develop UI design and answer initial questions

Week 3: 9/12 - 9/18

- Triage meeting with TM Luke (Weekly on Mondays through 12/6)
- Share and develop Status Report presentation with team members
- Create GitHub repository and get a barebones website up and running
- Run Status Report presentation on 9/12 to present on 9/13
- Standup meeting with Sponsor on Friday (Weekly through 12/6)
- Get success “Hello World” for React based web-app
- Develop first draft of Project Plan document to run by Sponsor
- Finalize database attributes to be used in web-app
- Implement UI/UX of web-app with functioning links to show Sponsor

Week 4: 9/19 - 9/25

- Complete first draft of Project Plan document to submit 9/21
- Develop elevator pitch for our project
- Get desired UI of web-app dashboard and message working for website

- Web-app local server up and running for prototype of database interaction
- Create functional employee log-in page
- Run presentation with team members multiple times for presentation
- Developing MadLib-like machine learning prototype

Week 5: 9/26 - 10/2

- Create prototype of real-time messaging, successfully running
- Implement the option for admin to see employee data on website
- Develop prototype of interactive data graphics
- Alpha Presentation slide deck downloaded

Week 6: 10/3 - 10/9

- Implement login/logout functionality with user status noted
- Successfully have real-time notification send/receive between users
- Successful machine-learning based message drafting prototype
- Alpha Presentation slide deck filled-out to submit 10/10
- Have complete Alpha system developed without bugs or major risks

Week 7: 10/10 - 10/16

- Create restriction to make web-app mobile device friendly
- Develop the transaction of kudos between users
- Prepare and run Alpha Presentation at Team Meeting on 10/10
- Submit Alpha Presentation slides and Project Plan document to Sponsor

Week 8: 10/17 - 10/23

- Continue to develop the transaction of kudos between users
- Implement real-time notifications in web-app, update UI interaction
- Create visual graphics of employee data within web-app
- Establish status and goals for coming Beta Presentation

Week 9: 10/24 - 10/30

- Continue developing real-time notifications in web-app
- Research ML library and optimize sentiment analysis function
- Draft plan for video vision and set dates for recording
- Confirm that each team member can access web-app on mobile device

Week 10: 10/31 - 11/6

- Have fully functioning kudos transaction implemented in web-app
- Update the UI by fixing the CSS to better fit the company vision/colors
- Have fully functional graphs of employee data for managers/admin view
- Work on integrating ML prototype into web-application

Week 11: 11/7 - 11/13

- Have fully functioning real-time notifications across users implemented
- Complete ML integration of message validation into the web-application
- Beta Presentation slide deck downloaded and filled-out to submit 11/14
- Set goals and check status on achieving quality deliverables

Week 12: 11/14 - 11/20

- Record Project Video content and audio throughout the week
- Fix any UI issues or bugs of web-app when on mobile device
- Prepare and run Beta Presentation at Team Meeting on 11/14
- Submit updated slide deck and Project Plan to TM and Sponsor

Week 13: 11/21 - 11/27

- Start editing Project Video recordings and audio together
- Update employee data accessibility and visualization as Sponsor sees fit
- Delegate what needs to be prepared for Design Day
- Finish editing video clips with audio to create Project Video

- Work towards stretch goals of ML kudos points and dynamic graph
- Get feedback from Sponsors on our deliverables and video

Week 14: 11/28 - 12/4

- Make final adjustments to web-app design based on Sponsor's feedback
- Perfect product elevator pitch and Design Day presentation information
- Complete the video that demonstrates our Ally Kudos web-app
- Finish refactoring and commenting code and write installations list
- Make final updates to Project Plan document as well as project

Week 15: 12/5 - 12/11

- Submit Project Video by 11:59pm on 12/5
- Submit source code, project plan document, team presentation slide decks, project video, and any added information by 11:59pm on 12/7
- Setup for Design Day at the Engineering Building at 1:00pm on 12/8
- Participate in Design Day starting at 6:00am on Friday 12/9

Week 16: 12/12 - 12/14 End

- Attend the final course presentation on 12/14 to wrap-up the semester