

Enthyme

Miles Allain, Cameron Duncan, Lex Graham, Brandt Ross

June 2021

I. Introduction

Enthyme presented our team with an idea for an application that connects reliable employees with quality employers with the goal to reduce turnover rates. Seasonal and temporary employment positions paying near minimum wage are particularly prone to high turnover. This application encourages entry level workers and up to remain with their company longer by rewarding their reliability with direct support toward personal goals they define in the application.

Enthyme was in its infancy and did not have existing software, storyboards, or design plan. Our client proposed many features and designs that combined would not be feasible for our team in the short span of five weeks. Deliberations with the client led to an agreement to focus on a high-level product that can be used as a proof of concept for Enthyme's prospective customers and investors as well as a functioning codebase for future development. Our client wants employees to declare their personal and work-related goals on the platform and see their progress towards completion of each goal. The intent is for employees to be encouraged to stay with their employer for the duration of their contract because they are dedicated toward seeing their goal fulfilled. For the scope of this working period, the application highlights a system for engaging employees through creation, management, and completion of their personal goals.

We delivered our client's vision in the form of a scalable web application. Given the time constraints of field session our feature development was constrained. We focused on designing a skeleton framework that could grow to fulfill all the client's needs. Our web app enables users to log in with a verified email address, manage their profile, and view, add, modify, and complete, their goals. We also have expansive user/usage data collection to provide analytics on how to best improve the site for the continuation of this project.

II. Requirements

Functional Requirements:

1-Design a high-level platform for the Enthyme application.

Since our client came to us with just the idea for the platform, it is our duty to lay out the foundation for the platform which includes deciding what technologies would be used and how users would interact with the web app.

2-Create the infrastructure for a web application using Azure Static Web Apps.

Configuration is needed such that a user can go to <https://www.enthyme.com> to access the website with all database and user authentication running in a manner that is abstracted from the user.

3-Implement a user authentication program.

The web app requires that users can be secure and differentiated. Therefore, it is imperative that user authentication is in place to index user data to keep track of which goals belong to which people and to keep user accounts secure.

4-Create a skeleton for data analytics. Build a data structure for saving user information.

For future development, a system where user interactions can be tracked will be vital. This information allows for navigating which new features should be prioritized. Furthermore, this data can

be used to generate more profit by selling it to other companies (e.g., selling tangible goal data to Amazon).

5-Allow users to create and maintain goals on their dashboard.

What differentiates this platform from other platforms that help people find jobs is that Enthyme allows users to set and manage goals to motivate themselves. Therefore, users should be able to create tangible, educational, and career goals.

Non-Functional Requirements:

1-Keep costs as low as possible due to the newness of the startup and lack of funding.

Because Enthyme is still in its infancy, cost for hosting must be kept to a minimum. This can be accomplished by using services that are free or free to start out.

2-Have high focus on security, especially regarding the financial side of the application.

Enthyme will eventually handle payment for the jobs, so security must be held as a high priority in all aspects of development (though finance-related security is out of scope).

III. System Architecture

Our client presented this project as a product vision without any existing infrastructure. She wanted a web-app hosted through Azure and gave us the liberty to design the interface and infrastructure for the application from the ground-up. We researched well over 100 software components with the goal of maximizing quality, scalability, and licensing/pricing. We settled on the components listed in Figure 1. Every component is affordable, scalable to a larger user base and developer group, and uses modern features of web development.

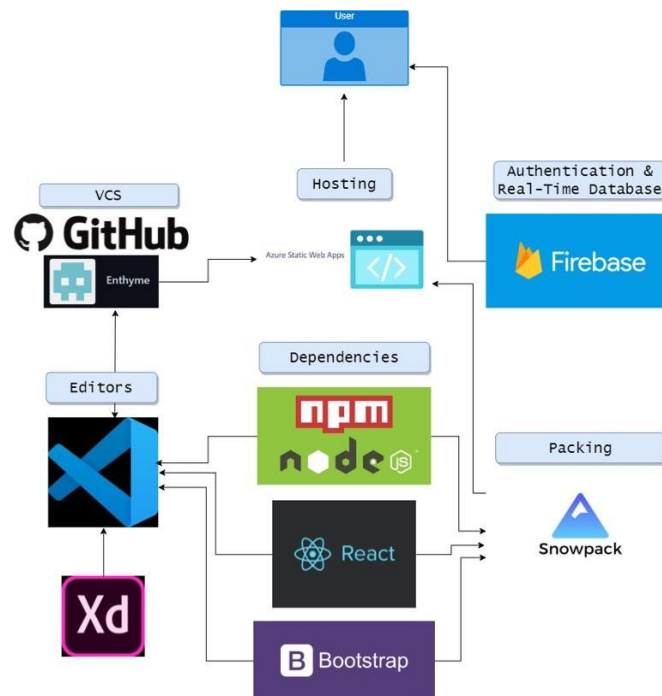


Figure 1: Tools and Components

GitHub

We use GitHub as our main version control system. By creating an organization for Enthyme, we were able to use private repositories free of charge. Pricing only scales by making use of build-automation (but 2500 minutes of automation per month are free).

Azure Static Web Apps

The site is hosted through Azure Static Web Apps and is accessible at the domain www.enthyme.com from anywhere in the world. Figures 2 and 3 are from the Enthyme admin account on the Microsoft Azure portal and Figure 4 shows that the site is live from the correct link. This service also integrates well with GitHub. Pushing new code onto the master branch will kick off a build script that will rebuild the site and post it to Azure, allowing for continuous integration.

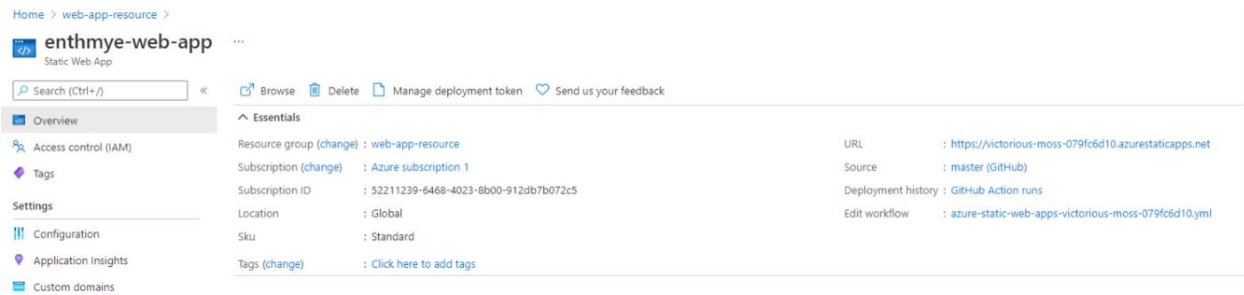


Figure 2: Azure Resources

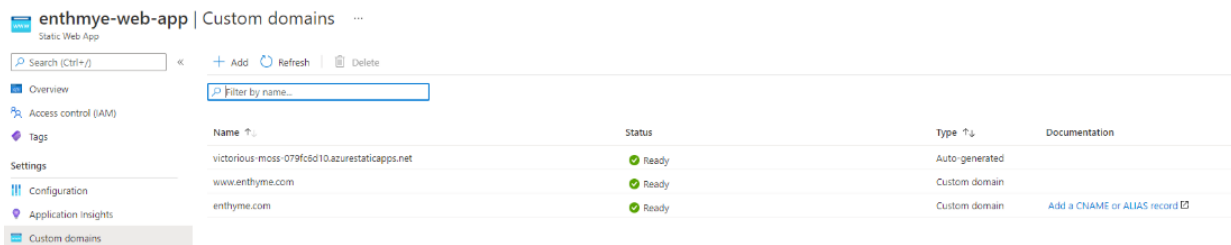


Figure 3: Azure Domains

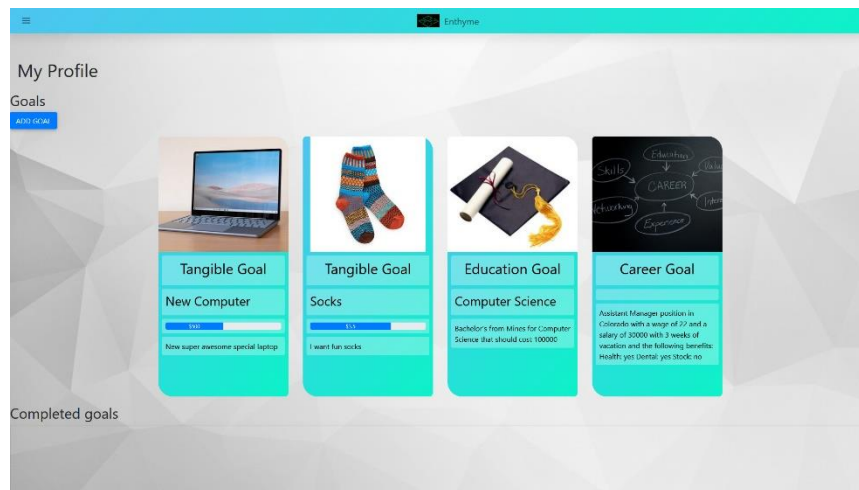


Figure 4: User Dashboard

Firestore

Firestore allows for easy and secure user credential and information storage and is free for up to 10,000 users and allows for several thousand database accesses per day. We use this service to authenticate our users, store information regarding each user and their goals, and run analytics on how users interact with the site.

Visual Studio Code

Visual Studio Code is an editor that is free to use for commercial purposes. It has a wide range of open-source extensions that makes web-development particularly easy.

Adobe XD

Adobe XD is a wire-framing and animation tool that we used to develop an initial idea of how the site should work. It requires a license that is roughly \$200, but we already had access to the service, so it was of no additional cost to our client.

Software Dependencies

The three most prominent components of our site were NPM/Node.js, React, and Bootstrap, all of which were used for full-stack web design. Node acts as the package manager, React is a build tool that enables dynamic site building, and Bootstrap is a large set of pre-built UI components. Other less-prominent tools include Material UI, MDB, Jest, and the React-Testing library. All these tools are free to use for commercial purposes.

Snowpack

In order to use React, which is written in JSX, a special compiling tool must be used. We decided to use Snowpack because it does not require additional configuration files (unlike its alternative, webpack).

IV. Technical Design

User Authentication

User authentication was needed in order to provide users with the ability to create and edit individual goals. Due to the private nature surrounding OAuth, it is important to consider security when implementing authorization. This led our team to look for a vetted system that would be able to handle OAuth. For this, we chose Firestore, which gives our application added security and allowed us to work on other parts of the application, trusting security to a more developed third party.

Firestore is a robust system for handling user authentication. It provides functionality to handle user creation and logging in. Additionally, it offers email verification and password resets. This allowed us to implement an authentication system fairly easily. By using forms in React, we were able to transfer a user's information and send it to the firestore to set up and manage user profiles.

We connected our authentication to our database by using the authentication ids to index the database, which gave us the ability to keep user information private from other users. By using React contexts, the app can check whether a user is signed in, even when moving across different pages of the website. This allows for consistent and fluid usage, without interruption. Another Benefit to this is the

ability to redirect users if there is no one currently signed in. This prevents unauthorized users (such as those who do not have an account) or unverified users from accessing parts of the website that require a user to be logged in.

Database

Part of the requirements for this project necessitates the creation and use of a database. The scope of this project involved the creation of an employee and goals database that can easily be expanded in the future. This resulted in the preliminary entity relationship diagram shown in Figure 5. Each employee has identifiable information with a primary key that is based on their ID from authentication which can then be used to index through goals. The goals are a variant data type that consists of either an educational, career, or tangible goal.

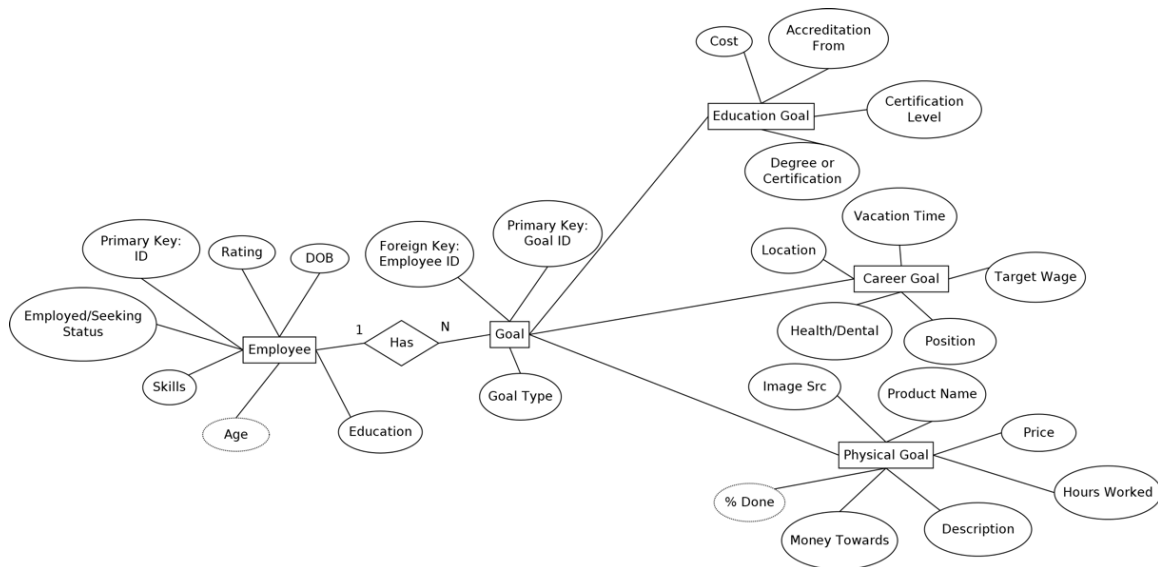


Figure 5: Relationship Diagram

This database was implemented using the Realtime Database feature in Firebase. This feature allows for storing JSON objects in folders. Employee objects are stored with their primary key as the title of the object with all relevant information inside of that object as shown in Figure 6. Furthermore, goals are stored in a different directory of the database with subdirectories that are titled with the primary keys of employees with subdirectories within that with arbitrary primary id keys as the title to goal objects as shown in Figure 7. These goals are kept as a variant by storing what type of goal they are within the JSON file and then only accessing the information that can be associated with that goal (i.e., if goalType is "Career," the attribute for price will never be called). A second goals database is kept for all completed goals that has the same structure of the uncompleted goals database. Finally, each goal has an attribute called "priv" which will hide the goal from the employer if set to true. This can be helpful to hide a career goal at Target if someone is currently working at McDonalds.

```
employee + x
├── RdsI5Q7WrLYEHERdbBMQqf0JdiT2
├── rVAUVdcw33hNbBS6HJJVQZK3Q302
└── wDarKo08fSdSyuWIXSkalAnQZpi2
    ├── dob: "01/01/2011"
    ├── edu: "High School"
    ├── eduDesc: ""
    ├── name: "Nisa"
    ├── profPic: "200px-Tentomon_yukata.png"
    └── seeking: false
```

Figure 6: Employee Database Object

```
rVAUVdcw33hNbBS6HJJVQZK3Q302
├── -MbmNLIJ8uuPoVHlmeMi
│   ├── desc: "2020 Ducati Hypermotard 950"
│   ├── goalType: "Tangible"
│   ├── image: "https://external-content.duckduckgo.com/iu/?u=h..."
│   ├── moneyTowards: 5300
│   ├── name: "Ducati"
│   ├── price: "17000"
│   └── priv: ""
```

Figure 7: User Goal Database Object

When the user accesses their dashboard, the database is combed through to find all goals that are associated with the currently logged in user's id. These goals are then displayed as a section of cards that can be edited, created, or deleted. This display is shown in Figure 8. Additional UI elements can be viewed in the Appendix.

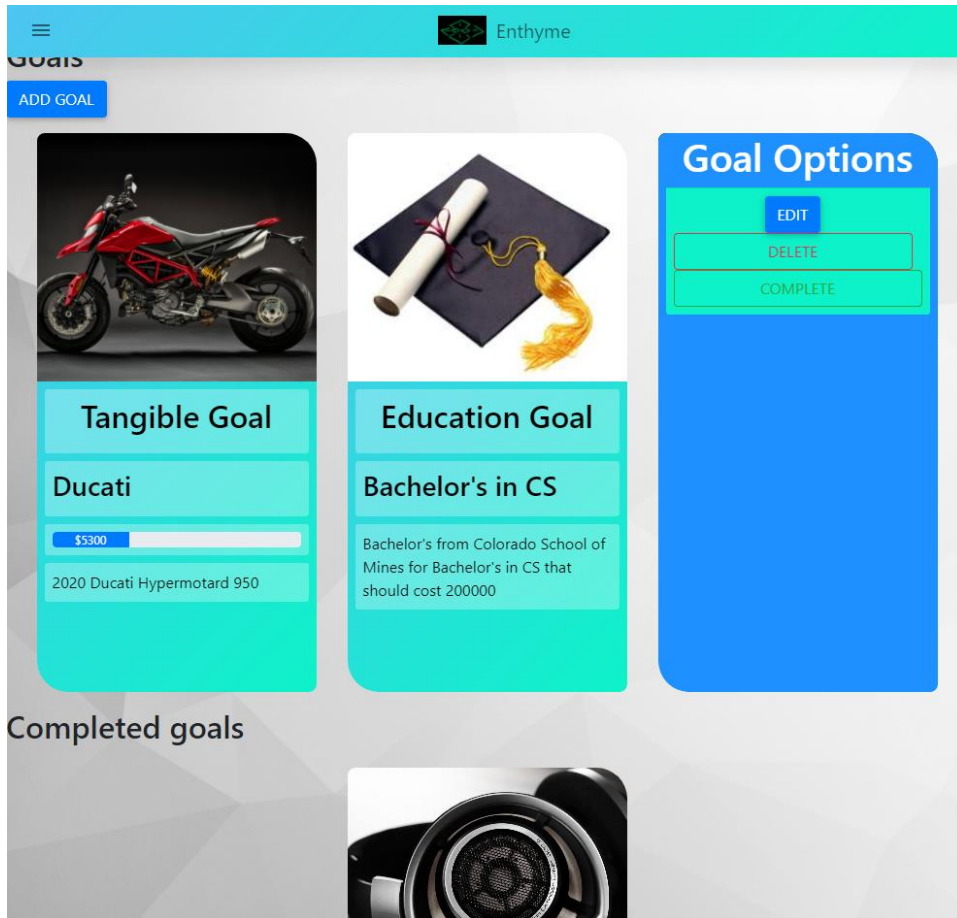


Figure 8: User Dashboard

V. Quality Assurance

React Testing

We added various tests to the web application to ensure code stability and quality as well as support future development. First, we made sure each part of the website was rendered using Jest and the React testing library. We also implemented several tests to simulate the actual use of the website. The React testing library allows us to click buttons and input into text fields like a user would. This allowed us to test the form handling of our site by testing how a user may input things into a form. This also provides us with the opportunity to check edge case scenarios with ease. We even faked a log in to ensure the login function works as well as make sure the firebase side of things is still functioning properly.


```
//screen.debug() //this prints the DOM to the console
fireEvent.click(screen.getByText("Add Goal"));

//inputing information into the form
screen.getByText("What type of goal is this?");
fireEvent.click(screen.getByText("Let's Go!"));
const degreeInput = screen.getByLabelText("Degree/Certification");
fireEvent.change(degreeInput, { target: { value: "High School diploma" } })
const accredInput = screen.getByLabelText("Accredidation From:");
fireEvent.change(accredInput, { target: { value: "DPS" } })
fireEvent.click(screen.getByTestId("education-submit"))

//checking the confirmation text
screen.getByText("Career goal successfully added")
```

Figure 9: Portion of Software Tests

```
Test Suites: 3 passed, 3 total
Tests:       19 passed, 19 total
Snapshots:  0 total
Time:        55.909 s
Ran all test suites.
```

Figure 10: Example Testing Output

This preliminary testing offers us the ability to test the code before sending it into production. This allows the quality of the code to benefit as well as ensures that the product is behaving as expected and is functional for when our client starts to present to investors and potential clients.

Analytics

Our analytics are collected entirely through Firebase. Firebase Analytics gives us information on how users interact our site over different time frames, how long those users are active, where those users log in from, what actions those users take (Figure 11), and what type of device those users log in on (Figure 12). This information can be used to guide future development of the site. For example, if there is a significant drop in users or how long users interact with the site, there may be a UI change that was not accepted or a bug that users couldn't work with. Or, if most users are signing in from their phones, Enthyme may want to focus on more mobile development.

Creating education goal	1
Creating tangible goal	6
Deleting tangible goal	5
first_visit	16
page_view	352
session_start	31
Submitting new profile info	1
Updating career goal	1
Updating education goal	1
Updating tangible goal	4

Figure 11: Site Usage Data



Figure 12: System Access Analytics

If more data analysis is needed, there is also the user data that is in the real-time database on Firebase. This data includes specific information about the user and the goals that the user has.

VI. Results

Functional Requirements:

1 - Design a high-level platform for the Enthyme application.

Platform design required a considerable amount of time for storyboarding with our client. We developed the framework for the user interface and a backend that could support our desired features and functionality. Our initial design for the site was simple and a basis that translated well into the final product.

Users will login to the site using a personal account and are presented with a dashboard containing in progress and completed goals. A message from their employer and contract fulfillment status (stubbed) are also displayed on the dashboard. Users can add goals edit their profile information all through links available on the main page.

2 - Create the infrastructure for a web application using Azure Static Web Apps.

As shown in the System Architecture section earlier, the infrastructure of the web app was created through services like Github, Azure Static Web Apps, and Firebase with the software dependencies of Npm/Node.js, React, Bootstrap, and Snowpack. Using these tools and associating the domain name correctly gives a working website that can be accessed through www.enthyme.com.

3-Implement a user authentication program.

Firebase Authentication was used to implement user logins on the application. Employees can log in and create private goals for themselves and keep track of personal information. Firebase does the heavy lifting by creating functions for users to reset their passwords and verify email, and it also does the background handling of the users, allowing for more ease when developing as well as offering reliable security.

4-Create a skeleton for data analytics. Build a data structure for saving user information.

Firebase Analytics was used to log user information with more data being available through the real-time database. The extent of data that is being collected is shown in Database section of the Technical Design section and the Analytics section of the Quality Assurance section.

5-Allow users to create and maintain goals on their dashboard.

The Database section of the Technical Design section shows the goals section being rendered with the ability to edit, create, or delete goals.

Non-Functional Requirements:

1-Keep costs as low as possible due to the newness of the startup and lack of funding.

This goal was achieved by using as many free tools as possible. There are two dependencies of this project that are free to use to start that then require payment. These tools are Azure, which is used for hosting the website, and Firebase, which is used for user authentication and storage. All other dependencies have free licenses to use in a commercial setting. It should be noted, however, that if GitHub exceeds 2500 minutes of automation in a month, it will also require payment. Luckily, this is avoidable.

2-Have high focus on security, especially regarding the financial side of the application.

We ensure security in this application by using the trusted third-party application, Firebase, to manage user authentication, password hashes, and storage. This is better than creating our own implementation because Firebase has undergone more development time by more qualified people than those working on this project. In other words, the best answer is to not reinvent the wheel.

VII. Future Work

Only a fraction of what is needed for this project to be completed could be achieved with the time constraints of field session. There are two key pieces of additional development we recommend. The entity relationship diagram for the database shows where this project still needs (Figure 13). The focus of this section of development was on making the employee profiles and the goals. This leaves employers and jobs to be developed. The employer side should be the next focus followed by the jobs data that bridges the gap between employers and employees. Implementing the employer data should be like the work done here with implementing the employee data. The job section should involve creating a job database inside of Firebase that is based off the data from employees and employers. This step of creating jobs should also focus on making or finding an algorithm that could match employees with jobs.

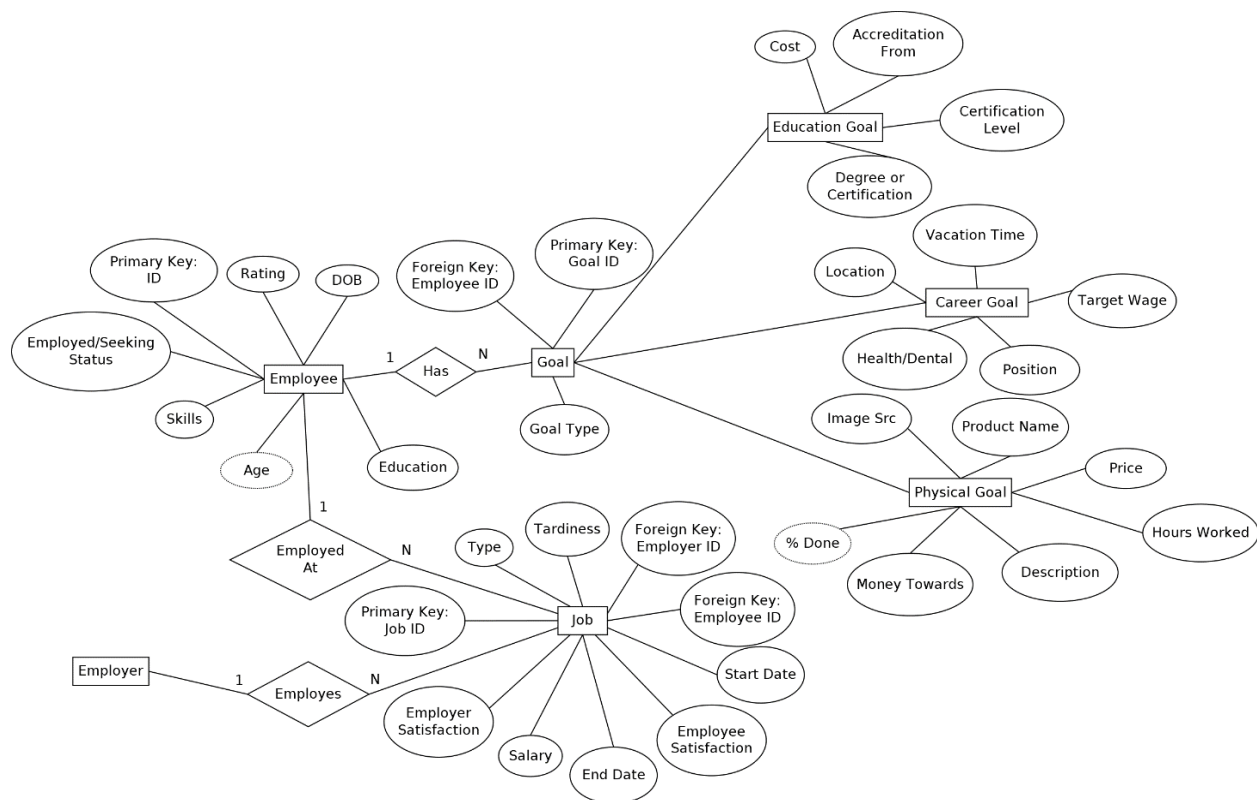


Figure 13: Relationship Diagram with Future Work

Furthermore, in terms of growing the company, much of the information we use for analyzing how the site can grow can also be sold to third party companies. For example, if we had 1000 users who all have a tangible goal of buying a laptop, we could sell the emails of this collection of users to a company like Asus which could then run targeted advertisements targeting these individuals.

Alternatively, a machine learning model could be developed to interpret more information regarding users on this site (namely rating how good of an employee / employer they are). The market for this information from other companies, namely social media sites, has proven to be substantial and could be a good means of funding Enthyme's future endeavors.

Other goals, which have been listed as stretches for the scope of this project, include the following:

- Add a dark mode (and other theme-customization options)
- Real-time message board for goals
- User decisions understanding system (e.g. surveys on why they left a job)
- Creating an application system to link employees with employers (and show applicants based on our rating system)
- Finance system for withholding 5% of an employee's paycheck)
- Reward system for contract fulfillment, goal completion, etc.
- Present to Investors
- Demo with intended user audience

Finally, it should be noted that there is much to be done regarding legal requirements for the site. First, we make use of many different software tools that are currently open source / free to use, but they may require disclosure that the service is being used. Second, our users need to understand their rights and responsibilities regarding the Enthyme platform, which will require an end-user license agreement. Lastly, Enthyme will also need to make users aware of their data collection methods and of any intentions to sell user information (which should constitute a privacy agreement).

VIII. Appendix

Figure 14 shows the form for creation of a goal, and Figure 15 shows the sign-up form.

What type of goal is this?
Tangible
LET'S GO!

What is being saved for?
Enter product name ⓘ
How much does this item cost?
Enter product price ⓘ
Please input a link to an image of the product
Enter product image source
Please enter a brief description of this product
Enter description here
 Is this goal private
SUBMIT GOAL
GO BACK

Figure 14: Goal Creation Form

Sign Up

Email

Password

Password Confirmation

Please input your name

Please input your date of birth

Please input your highest education level

Please add any extra information about your education here

Are you seeking employment now?

Already have an account?
[Log In](#)

Figure 15: Sign Up Form