

**Building and Evaluating an Application to Analyze Developer Performance
Metrics for Project Managers in Software Engineering**

by
Kavya Jeganathan

Gordon C Anderson, Advisor
Neena Jawaharlal Thota, Second Committee Member

A thesis submitted in partial fulfillment
of the requirements for the
Degree of Bachelor of Science with Honors
in Computer Science

University of Massachusetts - Amherst
Amherst, Massachusetts

May 6th, 2022

Abstract

In this project we examine pre-existing performance evaluation tools and identify their shortcomings for usage in computer science classrooms in a university setting. We then present a newly developed software application called Evalu8, meant to be used as a performance evaluation tool for student managers. We identify three main metrics that are at the heart of performance evaluation in the software engineering industry and convert them to metrics which can be used in a computer science curriculum for university. We adopt these newly established metrics in the performance evaluation tool Evalu8 and configure a method for student managers to evaluate their team members who collaborate on software engineering projects in courses which use GitHub repositories for codebase storage. The main metrics used for team member evaluation provided in Evalu8 are team member engagement, contribution, and progress. These metrics are used to establish clear criteria for student managers in courses such as COMPSCI 529: Software Engineering Project Management at the University of Massachusetts – Amherst, to evaluate and grade their team members without any ambiguity. To establish validity of the application and its performance capabilities, Evalu8 has been usability tested using learnability, memorability, efficiency, and satisfaction scores. Evalu8 opens up new opportunities for performance evaluation tool usage by students in a university setting.

Keywords: performance evaluation, software engineering, computer science curriculum

1. Introduction

In the software engineering industry today, performance reviews must be held periodically in order to increase employee productivity, loyalty, as well as growth. Therefore, a performance evaluation tool which includes features necessary for a quick and easy interaction is much needed. Furthermore, a well-designed performance evaluation tool is essential for certain communities, such as students enrolled at an educational institution who are part of classrooms which involve performance evaluations of their peers.

Performance reviews are critical to the success and productivity of a company. Keeping in mind that classrooms such as COMPSCI 320: Software Engineering at University of Massachusetts, Amherst are meant to simulate an experience similar to working for an actual company in a software engineer role, having an efficient performance evaluation tool in a classroom setting will make the simulated experience all the more realistic and prepare students further for life outside the safety of classroom walls. The current performance evaluation tools available such as Trakstar, BambooHR and Lattice, all provide a way to view and create performance reviews for team members being managed by an individual but fail to meet certain user needs. Moreover, different available performance evaluation tools have drawbacks associated with the features that they offer.

For example, the performance evaluation tool Trakstar has confusing terminology used in its user interface which complicates the user experience and leaves customers puzzled on how to interact correctly with the system. In addition, the current existent performance evaluation software tools have a hefty price associated with using them and offer minimal to no features without money being involved. For college students who may not be able to afford such prices for a software engineering class, using a performance evaluation software which is already available may not be an option due to the fees associated.

For my thesis project, I have designed a solution that is multifaceted and can fit most of my user base's needs when it comes to interaction with a time efficient performance evaluation software tool. After interacting with some of the existing performance evaluation tools such as Lattice and Reviewsnap, I uncovered shortcomings such as ambiguous instructions and poor feedback.

My honors thesis project consisted of creating an evaluation software tool called Evalu8 meant to be used in the scope of the COMPSCI 529: Software Engineering Project Management space. Currently, in COMPSCI 529, managers individually come up with the criteria they wish to use to evaluate their respective software engineering team members. This approach leads to inconsistent evaluation criteria for team members across different teams. The evaluation software tool, Evalu8, provides managers in COMPSCI 529 with clear criteria for evaluating team members which will be standardized across teams. It also would be a good tool for managers to use to collect data on individual team members to provide accurate feedback during team member performance evaluations.

The goal of the software tool, Evalu8, is to be a tool which utilizes the functionality of the pre-existent performance evaluation applications which users like and eliminate the features which users do not like. Evalu8 is a performance evaluation software tool with a user-friendly interface which is highly responsive and intuitive, and provides an efficient method of evaluating team members in the COMPSCI 529: Software Engineering Project Management space.

The metrics used for evaluation for Evalu8 are based on the metrics currently used in the software engineering industry and have been modified as necessary to conform to the requirements and expectation of the COMPSCI 529: Software Engineering Project Management and COMPSCI 320: Software Engineering course expectations. After being a manager for COMPSCI 529, myself, and researching about metrics used in the software engineering industry, I narrowed down the main categories for the evaluation software tool to engagement, contribution, and progress. The specific metrics used for Evalu8's performance evaluation system in order to conform to COMPSCI 529's course requirements are team member attendance, total commits contributed to project codebase, total lines of

code contributed to project codebase, project tasks completed, project tasks assigned, overall team member progress, team member progress satisfaction, team member progress pacing, and team member progress environmental impact.

The application Evalu8 has been evaluated using usability testing guided by Nielsen's definition of usability. This was done by creating a questionnaire to measure qualitative and quantitative feedback on each parameter in Nielsen's definition including learnability, efficiency, memorability, number of errors committed, number of assists and satisfaction.

Evalu8 has been programmed with a ReactJS frontend, a NodeJS backend and a relational PostgreSQL database. It has been deployed on Heroku at the following url: <http://evalu8-app.herokuapp.com/>

2. Background

A software developer's productivity in a corporate environment can be defined as how productive a developer is in a given timeframe based on a specific metric (Kim, 2021). Often, software engineering companies create trackable objectives for their employees, such as a standardized number of commits and code reviews expected to be completed per month, which allows them to set a baseline for what the company deems as acceptable in terms of performance (Kim, 2021).

Career ladders, described as the expected series of evolutionary steps an individual will take at a specific company, are termed as the foundation of an effective performance management system. Career ladders not only indicate the expected progress an individual should make in their job, but also, describes the expected behaviors and responsibilities for a specific role at a particular level of expertise (Larson, 2019). To determine how an individual employee is performing in comparison to the expectations of their career ladder level over a specific period of time, formal feedback in the form of a performance designation must be made (Larson, 2019).

Performance designations are usually represented in software engineering companies as a rating system modelled numbered single scale with the numbers one to five, very similar to a Likert scale. Software engineering companies also may use a rating system for performance designations in a three-by-three grid format in order to measure both performance and trajectory, which are the representations of either axis of the grid (Larson, 2019). The ratings represented in the performance designation are mainly calculated by three main reviews held periodically over the course of employment, namely, self-reviews, peer reviews as well as manager reviews. These three sets of reviews can be used as input to a calibration system which is used to ensure fairness and consistency of ratings across teams, organizations, and the overall company as a whole (Larson, 2019).

Referring to trackable objectives of a company, the most common evaluation metrics used to assess software developers in the software engineering industry, today, mainly consist of quantifiable productivity criteria relative to a project codebase such as lines of code, pull requests, number of commits, code reviews, and number of deployments or releases (Kim, 2021). Unfortunately, only measuring outputs per developer, does not give crucial context in terms of an employee's performance. For example, context such as whether an employee is working more hours than their peers or context relative to quality of work may be ignored by only measuring outputs per developer. Through creating the SPACE framework, Nicole Forsgren and other researchers from Microsoft and GitHub are continuing to tackle this issue of performance evaluation without enough context (Kim, 2021). The SPACE framework signifies the effort to pay attention to satisfaction and well-being, performance, activity, communication, collaboration, and efficiency in order to provide a holistic view of developer productivity at software engineering companies (Kim, 2021).

The developer performance mentioned in the SPACE framework, which denotes the outcome of a software engineer's work at a company, can be difficult to measure while only focusing on outcomes which are developer specific. Since business outcomes cannot be drawn directly from developer output, outcomes of a software engineer's work at a company need to be both developer as well as business specific (Kim, 2021). Developer specific outcomes include quality, reliability, and absence of bugs in the codebase, whereas

business specific outcomes refer to customer satisfaction, customer adoption and customer feature usage (Kim, 2021)

In terms of developer specific outcomes, a common way to measure developer progress and improvement is using a theory called the zone of proximal development. This theory was initially proposed in the 1930s by Lev Vygotsky, a Russian psychologist. The theory was originally used as a way for teachers to understand how students in schools could be best supported in their education (Stanier, 2020). During the time Lev Vygotsky's theory initially came to light, schools used the traditional teaching structure of teacher led instruction along with periodic and frequent assessments. Around the same time, another psychologist Jean Piaget, believed that self-directed curiosity-led development was the way that children learned the best (Stanier, 2020). Vygotsky, on the other hand, believed that Piaget's development theory only applied to certain subjects. He believed that most subject led development progressed quicker with the presence of a more knowledgeable individual who could help advance the learning of a student (Stanier, 2020). The fields that Vygotsky observed this phenomenon of a student progressing quicker with a more knowledgeable individual present were in subjects like mathematics and writing.

For clarity, the zone of proximal development is defined as the area in which a person is not able to grow or progress without the presence of a person with a more advanced skill set who can guide and assist them (Stanier, 2020). In the software engineering industry, an individual usually undergoes task-level proximal development. This means that on a software development team, it is a manager's job to keep their team members constantly in the zone of proximal development to help their team members progress. They do this by assigning tasks which pose as a challenge to a team member and hence, allows them to grow through advancing their ability by solving challenging tasks with the help of individuals on the team with a more advanced skill set. By solving challenging tasks with guidance, individual team members improve upon their own skill set, leading them to become more accomplished software engineers.

A popular way to measure business specific outcomes relative to customer satisfaction as well as feature usage in the software engineering industry is using the

evaluation design of usability testing. Usability testing is commonly computed with Jakob Nielsen's definition of usability heuristics which covers a few of Don Norman's design principles such as learnability, efficiency, memorability, errors and satisfaction (Nielsen, 2012). Don Norman's design principles are used in the software engineering industry in order to ensure that the products made by software engineers are understandable, usable and accomplishes the customer's desired task in a way in which a customer's needs are met and the experience of the customer is positive and enjoyable (Norman, 1990). Usability testing evaluates an interactive system with respect to Nielsen's usability heuristics and always involves real users during testing of the interactive system (Nielsen, 2012).

Performance evaluation applications which exist today assist individuals leading a team in managing their team's performance reviews. Some of the most popular performance evaluation applications are Trakstar, Reviewsnap, BambooHR, and Lattice. However, these applications often fail to perfectly meet most user expectations. A performance evaluation application should be able to predict the needs and preferences of its users, pushing forward dynamic metrics for assessment of employees and automatically providing a list of strong skills for employees in the system based on their assessment statistics. A performance evaluation application should also be easily navigable and have an easy to interact with user interface. The reality is that in their current state, these pre-existent applications have at least one of the following problems: a counter intuitive user interface, a difficult to navigate user interface, a complicated employee performance review system of previous weeks/months, no report which shows employee progress over time, not a lot of training for onboarding in terms of customer adoption, or a lack of upload file options which forces users to spend large amounts of time manually inputting data (Software Testing Help, 2021).

While it is hard to create a perfect solution to a seemingly complex problem, many of the existing applications fail to meet some of the basic requirements of their user base. Even if some of these applications meet all or most of their user base requirements, existing applications may lock basic features behind a paywall (Software Testing Help, 2021). Most college students cannot afford to pay for these features and in a classroom setting it is not expected for students to pay to use these expensive applications.

Some of these existing applications do have great features. For example, BambooHR sends automatic email reminders to ensure that managers are notified regularly so that they can provide timely employee reviews. The software also provides real-time 360-degree feedback, meaning that it makes it much easier for an employee to receive feedback from every party involved in an organization. However, even though these features are offered by the applications, users have complained that they find that the application has quite a bit of a learning curve and hence, can be frustrating and difficult to use as a lot of their user base may not necessarily be technologically-savvy (Software Testing Help, 2021).

Other applications like Reviewsnap have been deemed by users to be user-friendly for non-technology savvy folks. Unfortunately, the user interface design has been reported to need some improvements and manual input is required in certain areas of the system where a file upload could have been implemented but was not.

Furthermore, there are some well-known performance evaluation applications like TrakStar which seemingly has more pros than cons. Trakstar is fully customizable meaning it allows users to customize their performance evaluation criteria for different teams being evaluated. Trakstar, like BambooHR, also sends out automated reminders to participants, encouraging them to provide feedback for performance evaluation on a regular basis. However, Trakstar locks most of these features behind an expensive paywall which would not be accessible to students who cannot afford the hefty price of this performance evaluation software. Additionally, Trakstar has confusing terminology used on the platform which has received complaints from customers due to making the application more confusing to use than it needs to be.

A new performance evaluation software tool should take all the advantages and small pieces of functionality that users like about each of the pre-existent performance evaluation applications while eliminating all the features that they do not. The goal of Evalu8 was to create a performance evaluation software tool with a user-friendly interface which was highly responsive and intuitive, and provided an efficient method of evaluating team members in the COMPSCI 529: Software Engineering Project Management space. Given a team, each manager of the team is able to upload their team members to the

software tool through the GitHub API which automatically loads in collaborators for any project repository. The GitHub API connection with Evalu8 also allows managers to view their team members contributions to the semester long team project for COMPSCI 320: Software Engineering. Another upload of a CSV file containing the outcome of goal-setting surveys distributed three times throughout the semester in COMPSCI 320: Software Engineering, allows for managers to keep track of their team members goal-setting progress throughout the semester without having to manually enter any data into Evalu8. College students will hopefully now be able to quickly and efficiently evaluate their peers with the help of the application Evalu8, especially in a classroom setting which requires performance evaluations such as COMPSCI 320 and COMPSCI 529 at University of Massachusetts, Amherst.

3. Methodology

For the performance evaluation tool, Evalu8, the languages used for the system are mainly JavaScript, CSS and HTML. The frontend web framework for the tool is ReactJS and the backend framework is NodeJS. The database which has been used is a relational PostgreSQL database. Deployment is done on Heroku.

Types of Users

There are two types of users for the tool Evalu8:

1. Admins or Professors who have the ability to:
 - a. Add attendance dates into the system
 - b. Add the number of manager, peer, and goal-setting surveys for the semester into the system
2. Student managers who have the ability to:
 - a. View their team's contribution to a repository codebase through statistical evidence provided via the GitHub API

- b. Record attendance as well as record completion of evaluations for their team members
- c. Download attendance of their team members in CSV form in order to upload attendance to Moodle
- d. View progress of team members in accomplishing goals set for the semester
- e. View engagement of team members with the course

User Interface

See Appendix A for visuals of the user interface pages of Evalu8. The user interface pages present in Evalu8 are as follows:

For Admins or Professors

| UI View | Purpose |
|-------------------------|--|
| Admin Create an Account | This page enables an admin to create an account. |
| Admin Login | This page enables an admin to login to the software tool web platform. |
| Admin Landing Page | This page provides a manager with the ability to select dates from a calendar which they would like students to log attendance for. It also allows admins to choose the number of manager, peer and goal-setting evaluations they expect students to complete throughout the semester. |

For Students or Managers

| UI View | Purpose |
|---------------------------|---|
| Manager Create an Account | This page enables a manager to create an account. |

| | |
|----------------------|---|
| Manager Login | This page enables a manager to login to the software tool web platform. |
| Team Members Display | This page provides a manager with an overview of the members of their teams in the form of team member cards with a team member's name, email, and GitHub username. |
| Engagement Profile | This page provides a manager with an overview of their team member's engagement in terms of attendance, as well as completing of peer, goal setting and manager evaluations. |
| Progress Profile | This page provides a manager with an overview of their team member's progress in terms of pacing, satisfaction, and environmental impact toward achieving goals set in COMPSCI 320. |
| Contribution Profile | This page provides a manager with an overview of their team member's contribution to a project with a codebase stored in a GitHub repository, through providing information on total lines of code contributed and total commits contributed to the project repository. It also provides details on tasks assigned and completed by team members. |

Modals

See Appendix B for visuals of the modals in Evalu8. The modals present in Evalu8 are as follows:

| | |
|------------------|---|
| Attendance Modal | This modal provides managers with the ability to log attendance for their team members. It also provides managers with the status of whether or not they have completed logging attendance for a particular date. It also provides the ability to download a CSV file of logged attendance which can be directly uploaded to Moodle for recording class attendance on Moodle. |
| Engagement Modal | This modal provides managers with the ability to log how many manager, peer and goal-setting surveys their team members have completed over the semester. |

Database Tables

The tables present in the evaluation tool's PostgreSQL database are as follows:

Admin Table

| Column | Data Type | Description |
|-------------------|-----------|---|
| name | text | the full name of the admin or professor |
| course | text | the course taught by the professor |
| attendanceondates | text[] | an array of dates chosen by the professor |

| | | |
|-------------|---------|---|
| | | for their students to log attendance for |
| evalmetrics | text[] | an array of the number of manager, peer and goal-setting surveys assigned by a professor for the semester |
| password | text | the password for a professor's Evalu8 account |

Managers Table

| Column | Data Type | Description |
|-----------------|-----------|---|
| name | text | the full name of the manager |
| classroom | text | the course of the manager |
| admin | text | the professor of the course |
| team members | text[] | the team members of the manager's team |
| github reponame | text | the repository name for a semester long project |
| github username | text | the repository owner name for a semester long project |
| github token | text | the GitHub personal access token generated for a manager's GitHub account |
| password | text | the password for a manager's Evalu8 account |

Team Member Table

| Column | Data Type | Description |
|-----------------|-----------|--|
| name | text | the full name of the team member |
| course | text | the course of the team member |
| email | text | the email of the team member |
| manager name | text | the GitHub username of the team member's manager |
| GitHub username | text | the GitHub username of the team member |

Attendance Table

| Column | Data Type | Description |
|------------------|-----------|--|
| full name | text | the full name of the team member |
| course | text | the course of the team member |
| admin | text | the professor of the course |
| email | text | the email of the team member |
| manager | text | the GitHub username of the team member's manager |
| team member info | text | the GitHub username of the team member |
| date | text | the date which attendance is being taken for |
| status | text | the present, absent or excused status of a team member for a particular date |

Evaluations Table

| Column | Data Type | Description |
|------------------|-----------|--|
| course | text | the course of the team member |
| admin | text | the professor of the course |
| manager | text | the name of the team member's manager |
| team member info | text | the GitHub username of the team member |
| eval type | text | the type of the evaluation out of manager, peer or goal-setting types |
| eval number | int4 | the evaluation number of the manager, peer or goal-setting surveys |
| is completed | bool | a boolean which is true if an evaluation has been completed by a team member and false when the evaluation is not completed by a team member |

Progress Table

| Column | Data Type | Description |
|--------------|-----------|--|
| email | text | the email of the team member |
| course | text | the course of the team member |
| progress | int4 | the progress made by a team member in achieving their set goals on a scale from 1 to 5 |
| satisfaction | int4 | the satisfaction of a team member in achieving their set goals on a scale from 1 to 5 |
| pacing | int4 | the pacing of a team member in achieving their set goals on a scale from 1 to 5 |
| environment | int4 | the amount environment impacts team member goal achievement on a scale from 1 to 5 |

Backend Restful APIs

A list of the RESTful APIs which have been established in the system are as follows:

| Route | Description |
|------------------------------------|--|
| <i>Admin Endpoints</i> | |
| /api/createAdmin | adds a new admin or professor to admin table in the database |
| /api/adminAttendance | adds the attendance dates chosen by the professor for the semester's attendance to the attendance table |
| /api /adminEvalMetrics | adds the number of manager, peer and goal-setting evaluation surveys chosen by the professor for students to complete during the semester to the evaluations table |
| /api/getAllAdmins | gets all the admins from the admin table |
| <i>Manager Endpoints</i> | |
| /api/createAccount | adds a new manager to manager table in the database |
| /api/getRepoNameByManagerAndCourse | gets the repository name from the manager table by a manager's GitHub username and course |
| /api/getTokenAndAdminByManager | gets the GitHub token and admin associated with a manager from the manager table by a manager's GitHub username and course |
| <i>Team Member Endpoints</i> | |
| /api/addToTeamMemberTable | adds a new team member to team member table in the database |

| | |
|--------------------------------------|--|
| /api/getAllMembersByManagerAndCourse | gets all team members who have a particular manager for a certain course |
| /api/getEmailByGitUsername | gets the email of a team member from the team member table by the team member's GitHub username |
| <i>Attendance Endpoints</i> | |
| /api/getAllAttendanceByDate | gets attendance from the attendance table for a team member based on date, course, and the manager of the team member |
| /api/addAttendanceByDate | adds the attendance of a team member to the attendance table |
| /api/updateAttendanceByDate | updates the attendance status of a team member already in the attendance table |
| /api/viewAttendanceByDate | gets attendance from the attendance table for a team member based on date, course and the GitHub username of a team member |
| /api/statusAttendanceByDate | gets the attendance status (present, absent or excused) for a specific team member for a particular course based on the date. |
| <i>Evaluation Endpoints</i> | |
| /api/addEvaluation | adds evaluation information of team members to the evaluation table in the database |
| /api/getAllEvaluations | gets evaluation information from the evaluations table for a team member based on evaluation number, evaluation type, course, and the manager of a team member |
| /api/viewEvaluation | gets evaluation information from the evaluations table for a team member based on evaluation number, evaluation type, |

| | |
|---|---|
| | course and the GitHub username of a team member |
| /api/getCompleted | gets evaluation completion status from the evaluations table for a team member |
| /api/getDistinctEvalType | gets the distinctive evaluation types from the evaluations table |
| /api/updateEvaluation | updates evaluation information of team members and adds updated information to the evaluation table in the database |
| /api/getEvalMetricsByAdminAndCourse | gets the number of manager, peer, and goal-setting surveys of team members from the evaluations table based on a team member's admin or professor for a particular course |
| <i>Progress Endpoints</i> | |
| /api/addProgress | adds progress information of team members to the evaluation table in the database |
| /api/getProgressByEmailAndCourse | gets progress information of a team member based on the email and course of the team member |
| /api/updateProgress | updates progress information of team members to the evaluation table in the database |
| <i>GitHub REST API Endpoints</i> | |
| /gitapi/github/userInfo/:user/:token | gets user details from the GitHub API present on a user's GitHub public profile |
| /gitapi/github/teamInfo/:owner/:reponame/:token | gets user details of the collaborators on a GitHub repository |
| /gitapi/commitInfo/:owner/:reponame/:token | gets the number of additions, number of deletions, and commit information of the |

| | |
|---|--|
| | collaborators on a GitHub repository |
| /gitapi/github/issueInfo/:owner/:reponame /:state/:token | gets the number of tasks assigned and completed by collaborators on a GitHub repository using the GitHub Issues board for project planning |

Procedures and Tools

The procedures, tools, and techniques I had used to gather data for my thesis include:

1. Secondary data collection through retrieving data about metrics used by managers for evaluation in COMPSCI 529: Software Engineering Project Management in past semesters from faculty who have taught and collected data on the same in COMPSCI 529 in the past.
2. Secondary data collection through analyzing the key readings mentioned in the literature review for the metrics currently used in the software engineering industry. This has helped to finalize a list of criteria which portrays the traits of a software engineer which are evaluated in the actual workplace so that students get an understanding of how they will be evaluated in software engineer roles outside the classroom.
3. Secondary data collection of documented manager reviews of their team members from previous semesters with personal data anonymized as well as documented peer reviews amongst team members.
4. Primary data collection through close ended question surveys for goal setting, mainly with Likert scale questions in order to determine more about goal-setting progress for students in COMPSCI 320: Software Engineering. Personal information of participants is anonymized.

5. Primary data collection through questionnaires for usability testing which will measure qualitative and quantitative feedback on each parameter in Nielsen's definition including learnability, efficiency, memorability, errors, and satisfaction in terms of user experience while working with the software tool Evalu8. Personal information of participants is anonymized.
6. Retrieving team member contribution and commit data from GitHub repositories using the GitHub REST API in order to retrieve data about student participation in terms of tasks assigned as well as commits and lines of code contributed to the project codebase for COMPSCI 320: Software Engineering.

4. Results

The application Evalu8 was tested using usability testing guided by Nielsen's definition of usability. This was done by creating a usability questionnaire or survey to measure qualitative and quantitative feedback on each parameter in Nielsen's definition including learnability, efficiency, memorability, errors, assists and satisfaction. The questionnaire was available through a Google Form which consisted of questions measuring the user's experience through a Likert scale and open-ended responses. 5 participants were gathered to interact with and evaluate the application Evalu8. Their responses were analyzed using the analytics provided by Google Forms as well as documented using a tableau dashboard to view data trends and visualize findings. Furthermore, an affinity diagram was created to learn more about open-ended, qualitative responses.

Participants

Four out of the five participants in the study were undergraduate students. One of the participants was a master's student, with a concentration in Human Computer Interaction. In terms of majors for the undergraduate students, two of the participants were computer science majors, one of the participants was an informatics major and the last was a public health major. The participants who had pursued a computer science or informatics major

had worked on full stack projects previously which had been stored in a GitHub repository. The participant pursuing a public health major had also worked on a statistical project stored in a GitHub repository using the R coding language.

Simple Task List

Before participants were asked to complete the usability survey, they were asked to complete a list of simple tasks while interacting with Evalu8 for the first time. The list of simple tasks are as follows:

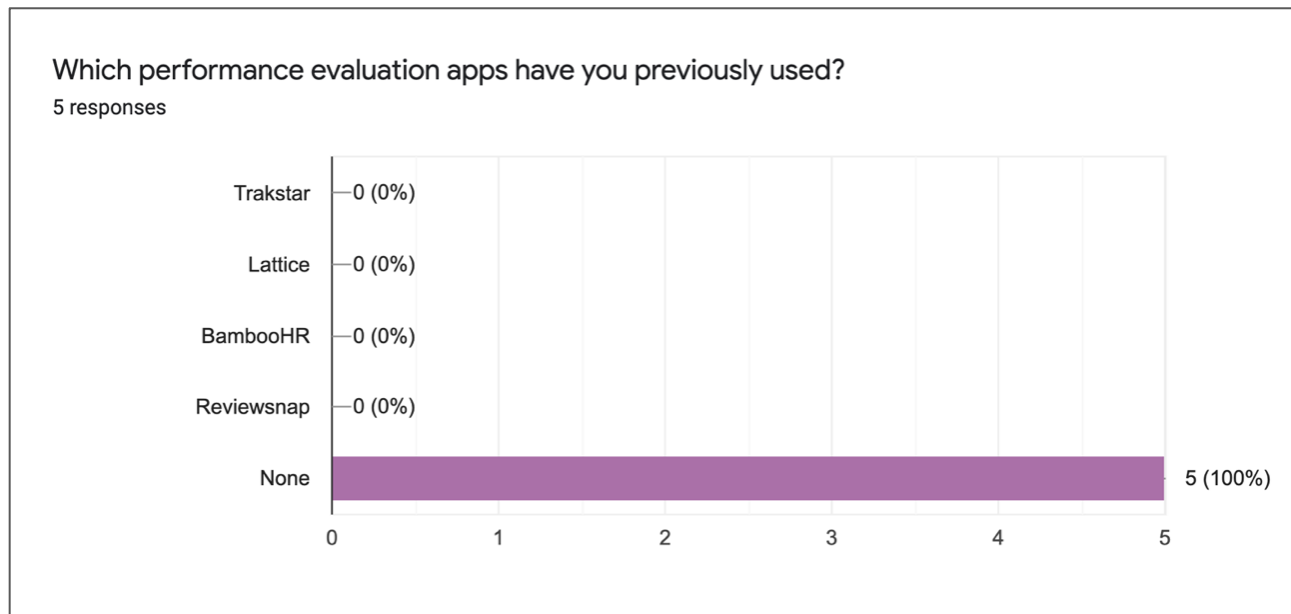
1. Create a new account with a repository of your choice from your GitHub account.
2. Login to your account.
3. Look into the contribution details of one of your team members. Note down the number of lines of code added to the codebase by the team member.
4. For the same team member, access the sidebar and navigate to the attendance modal. Then, add the attendance of the team member for any two dates, marking them absent on one day and present on another.
5. For the same team member, access the sidebar and navigate to the evaluation modal. Then, mark a manager evaluation as 'complete' for the team member.
6. Look into the engagement details of the same team member. Note down the number of days absent for the team member as well as the percentage of manager evaluations complete.
7. Look into the progress details of the same team member. Then, upload the team member goal-setting survey provided and note down the percentage of progress satisfaction for the team member.

Analyzing Quantitative Results

Using Tableau Dashboard and Google Form Responses

1. Previously used performance evaluation tools

Through evidence provided by the responses of the participants to the user survey, it can be noted that pre-existing performance evaluation tools such as Trakstar, Lattice, BambooHR as well as Reviewsnap are not used commonly by university students. All the participants had previously worked on semester long projects using GitHub but were weary of paying large fees to use pre-existing performance evaluation tools to track the engagement and contribution of other team members.

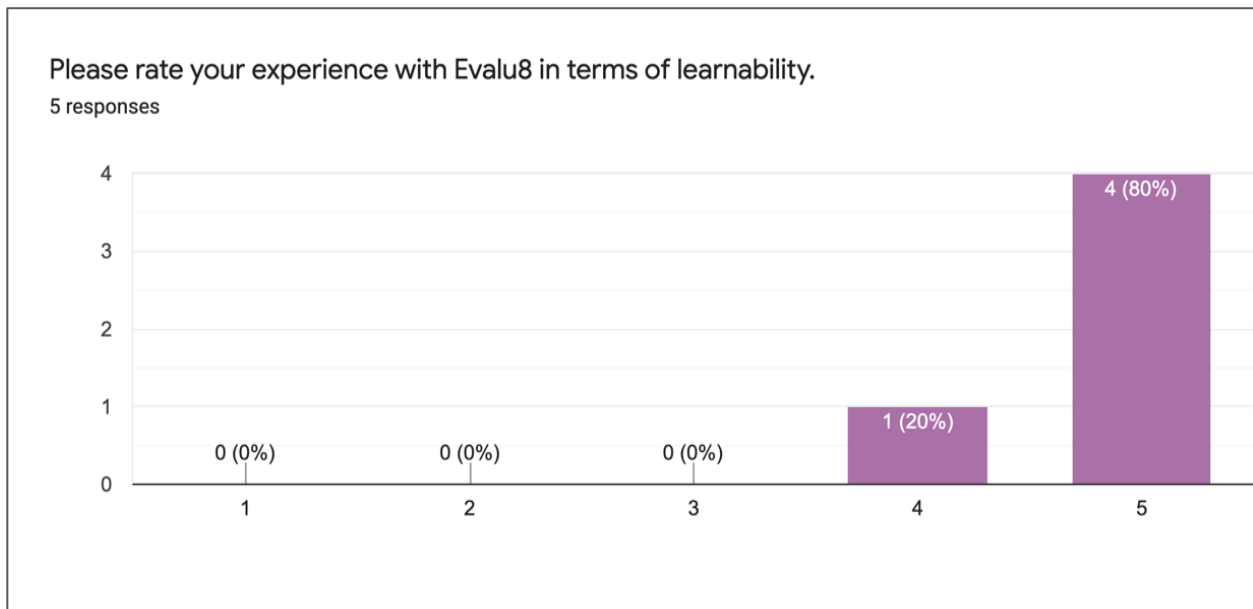


2. Nielson's definition of usability

Looking at the survey responses to the parameters involved in Nielson's definition of usability, the general response to Evalu8 is a positive one.

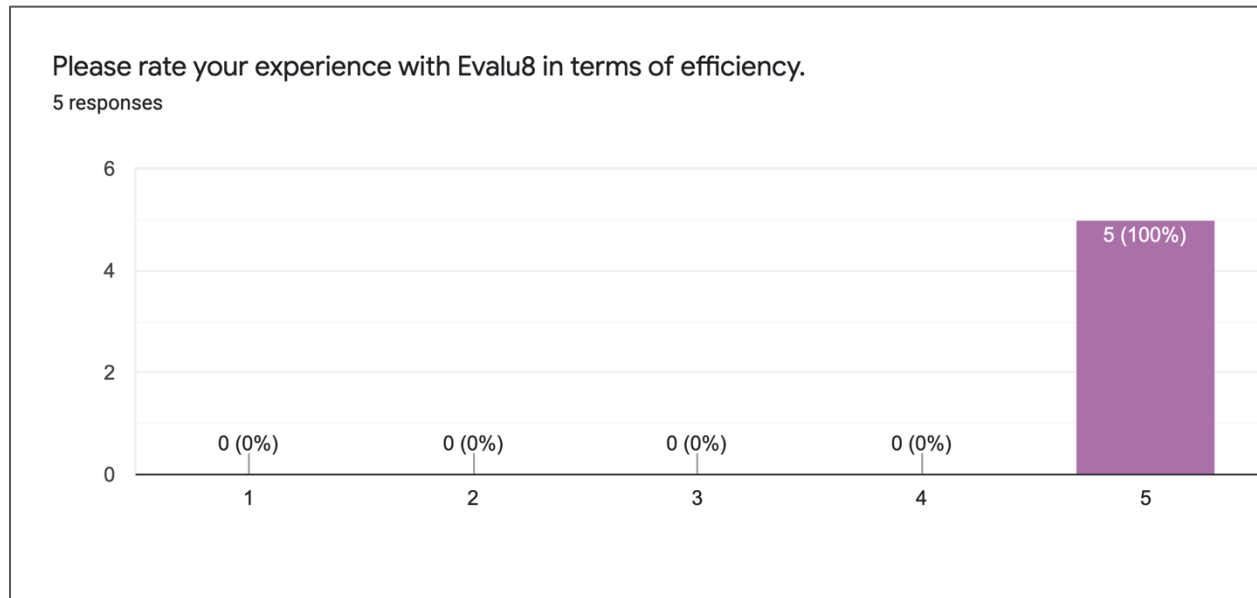
2.1 Learnability

Learnability was operationalized as how easy it was to learn and immediately begin using the application Evalu8. The Likert scale was defined with the number 1 representing “not easy to learn at all” and the number 5 representing “very easy to learn.” The majority of responses by participants leaned toward Evalu8 being very easy to learn and hence, having a high learnability.



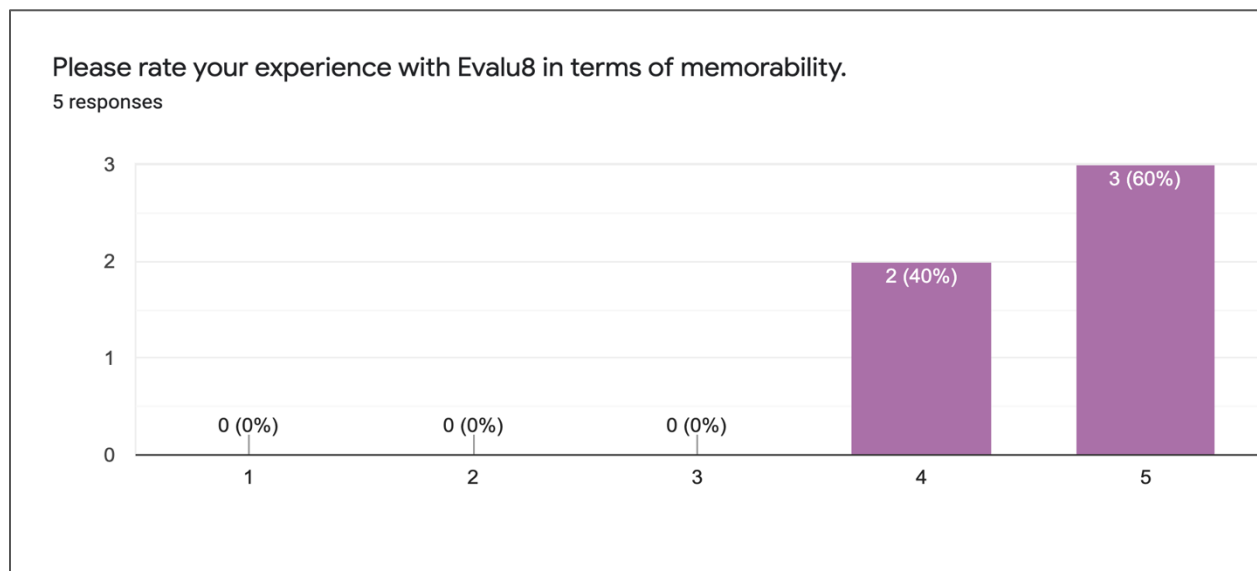
2.2 Efficiency

Efficiency was operationalized as how time efficient it was to learn the system and perform tasks. The Likert scale was defined with the number 1 representing “extremely time inefficient” and the number 5 representing “very time efficient.” All responses by participants indicated that Evalu8 was very time efficient when it came to performing tasks after learning the system, suggesting that Evalu8 has a really high efficiency.



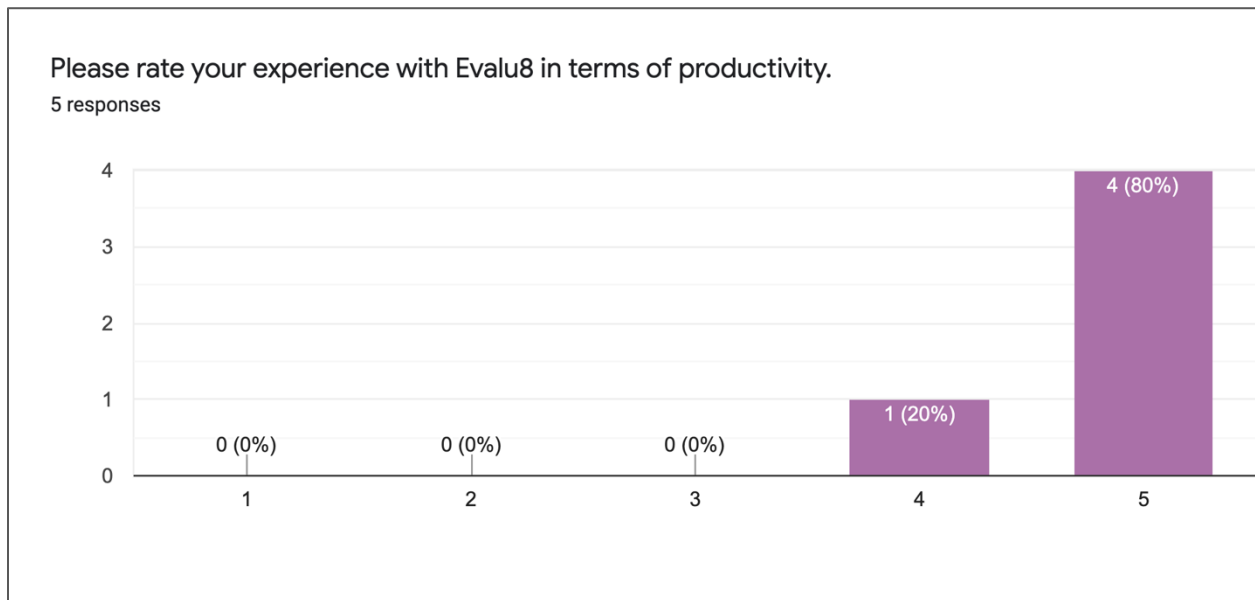
2.3 Memorability

Memorability was operationalized as how easy it would be to use the system if a person was to use it again in the near future. The Likert scale was defined with the number 1 representing “very difficult” and the number 5 representing “very easy.” Though, not every participant thought that it was very easy, they still thought that the system would be quite easy to use if they were to use it again in the near future. This indicates that Evalu8 has good memorability.



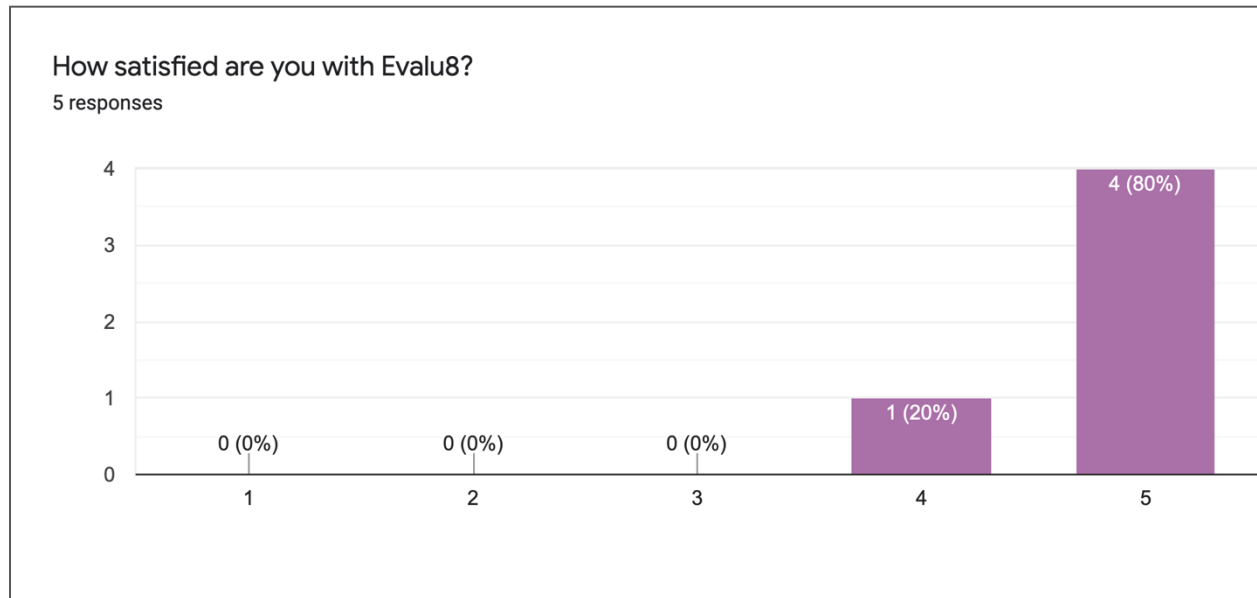
2.4 Productivity

Productivity was operationalized as whether a participant was able to achieve their desired productivity by completing the list of simple tasks. The Likert scale was defined with the number 1 representing “low productivity” and the number 5 representing “high productivity.” The majority of responses by participants leaned toward Evalu8 providing high productivity.



2.5 Satisfaction

Satisfaction was operationalized as whether Evalu8 supported user performance. The Likert scale was defined with the number 1 representing “not satisfied” and the number 5 representing “very satisfied.” The majority of responses by participants leaned toward Evalu8 providing high user satisfaction.

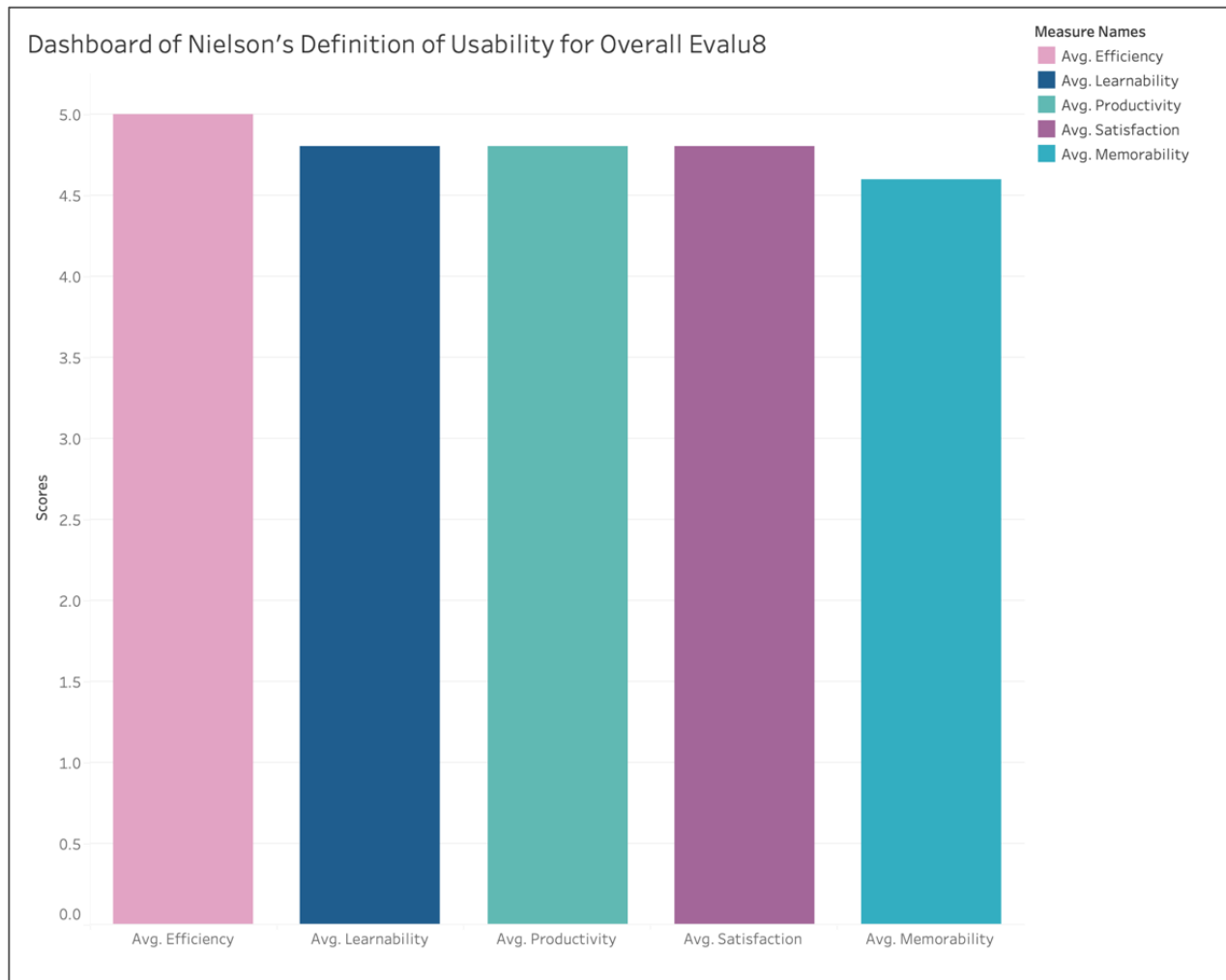


Overall Outlook on Nielson's Definition of Usability Parameters

In terms of all the parameters associated with Nielson's definition of usability, efficiency seems to be the parameter with the highest rating for the application Evalu8. In other words, the participants found that learning the system and performing tasks was extremely time efficient using Evalu8.

The average learnability, productivity and satisfaction are also quite high, though not as high as the average efficiency of the application. This indicates that Evalu8 is easy for first time users to learn. It also suggests that the application supports user performance and allows users to complete their desired tasks in a timely manner.

Average memorability of the application seems to be the lowest scored parameter when compared to learnability, productivity, satisfaction and efficiency, though, it still has quite a high score. This suggests that though the application Evalu8 is easy to learn for first time users, when a user tries to use the application in the future, they may have to relearn how to use some features of the application since memorability is a little lower. This could be solved through building familiarity of the application by periodic usage.

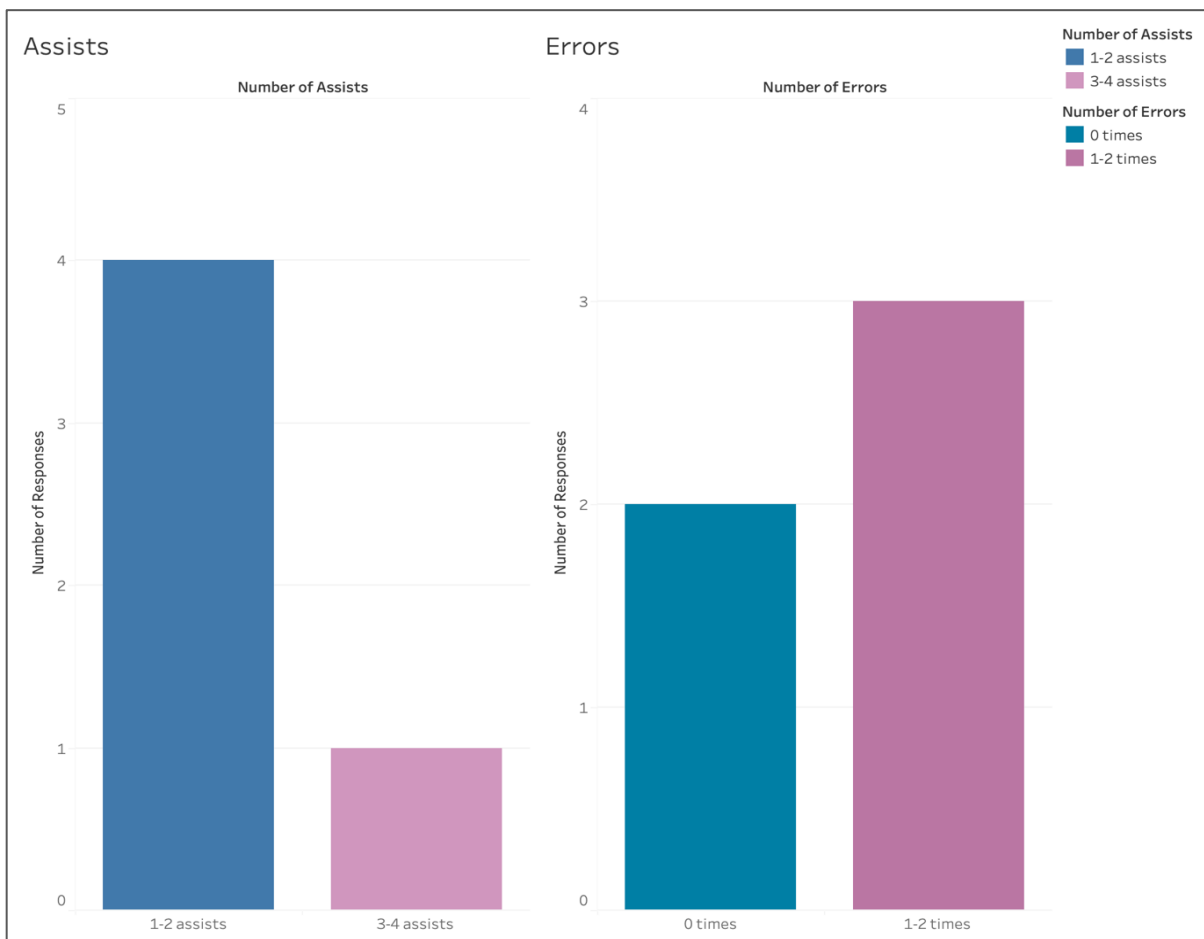


3. Number of Assists

The total number of assists needed for first time users of the application Evalu8 was predominantly 1 to 2 assists while completing the list of simple tasks. There was one participant who needed 3-4 assists. This participant was the participant who was pursuing a public health major and had the experience of only working on one coding project in the past which had been stored on GitHub. All the other participants had worked on multiple projects stored on a GitHub repository. This seems to indicate that users who have had more interaction with GitHub in the past may find using Evalu8 easier than those who have not had as much interaction.

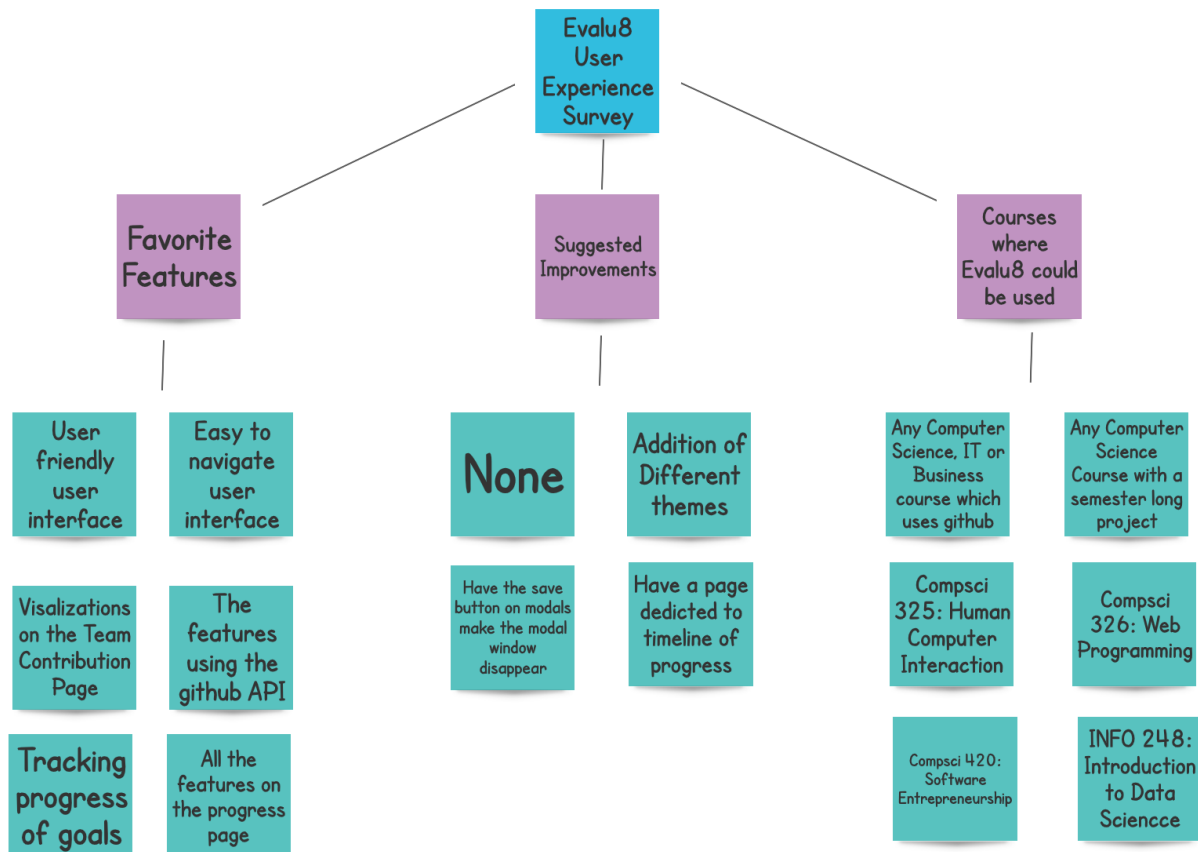
4. Number of Errors

The total number of errors made by first time users of the application Evalu8 was predominantly 1 to 2 errors while completing the list of simple tasks. There were some participants who made 0 errors. These participants were the participants who had spent the maximum amount of time working on projects stored on GitHub during their time at university for both course work as well as personal projects.



Analyzing Qualitative Results

Using an Affinity Diagram



1. Favorite Features

Many of the participants enjoyed working with the contribution page of the application. They seemed really fascinated by the usage of the GitHub REST API to pull in data in real time, allowing them to view the contributions of their team members for different projects which they had stored in GitHub. Another feature which the participants enjoyed was the progress page. The participants were enthusiastic about goal-setting surveys and their ability to help measure team member growth in a course setting. Furthermore, participants liked the user friendly and easy to navigate user interface along with the default theme of the application.

2. Suggested Improvements

Though many participants enjoyed the default color scheme for Evalu8, one participant suggested providing a set of three different themes for the application so that users could choose what color scheme they could use while interacting with the application. Another great suggestion was having a line graph on the progress page which would show a person's progress throughout the semester. This graph would allow for an easy visualization of a person's progress in terms of attaining their personal set goals throughout the semester.

3. Courses

Besides using Evalu8 in courses such as COMPSCI 320: Software Engineering and COMPSCI 529: Management in Software Engineering, the participants suggested using Evalu8 in other computer science classrooms such as COMPSCI 325: Human Computer Interaction and COMPSCI 326: Web Programming. Outside of the computer science major, it was suggested that an application like Evalu8 could be used in classrooms such as INFO 248: Introduction to Data Science.

5. Conclusion

Through the usability test which was run on Evalu8, it was discovered that students at University of Massachusetts – Amherst are not only interested in a well-functioning performance evaluation application, but are also interested in the metrics which the performance evaluation supports. Since Evalu8 was developed using metrics which were converted from those used for performance evaluation in the actual software engineering industry to adhere to a classroom setting, many students find Evalu8 much more appealing to use rather than the pre-existing performance evaluation software's which were not created with university students in mind.

By developing Evalu8, I discovered how to effectively design the architecture of an evaluation software tool through successfully establishing the API endpoints, user interface

pages, and database tables for the tool. I also learned how to carry out an effective usability test on a software application to support the validity of the performance evaluation application, Evalu8.

One challenge which I faced while working on Evalu8, was struggling to establish the connection between Evalu8 and the GitHub RESTful API which would provide insight into team member contribution to a project repository. This was primarily due to a lack of experience working with the GitHub Restful API, but, was resolved after doing extra research on the API and extensively testing the endpoints offered by it via Postman, an API platform for building and testing APIs.

If I were to start my research over from scratch, I would focus more on ensuring the functionality of the application worked first, before working on designing the user interface. During the development of Evalu8, I spent a long time at the start of my development process working on figuring out a color palette and design template which I thought suited the application. As important as it was to establish the design for the user interface, I wish that I had started out the application working more on functionality rather than design, since then, I would have been able to look at designing the application in more detail after establishing most of the functionality of the application. That way, I may have been able to think about adding in a dark or light mode as well as different themes for the application after all the functionality was already in place.

In terms of future scope of the evaluation software tool, the tool could move on to being used in classrooms other than COMPSCI 529, especially in the College of Information and Computer Sciences, in order to help with performance evaluation tasks in the Computer Science space. Furthermore, modifiable and configurable metrics could be introduced to the application in case student managers would like to add new metrics to the default list of metrics currently offered in Evalu8.

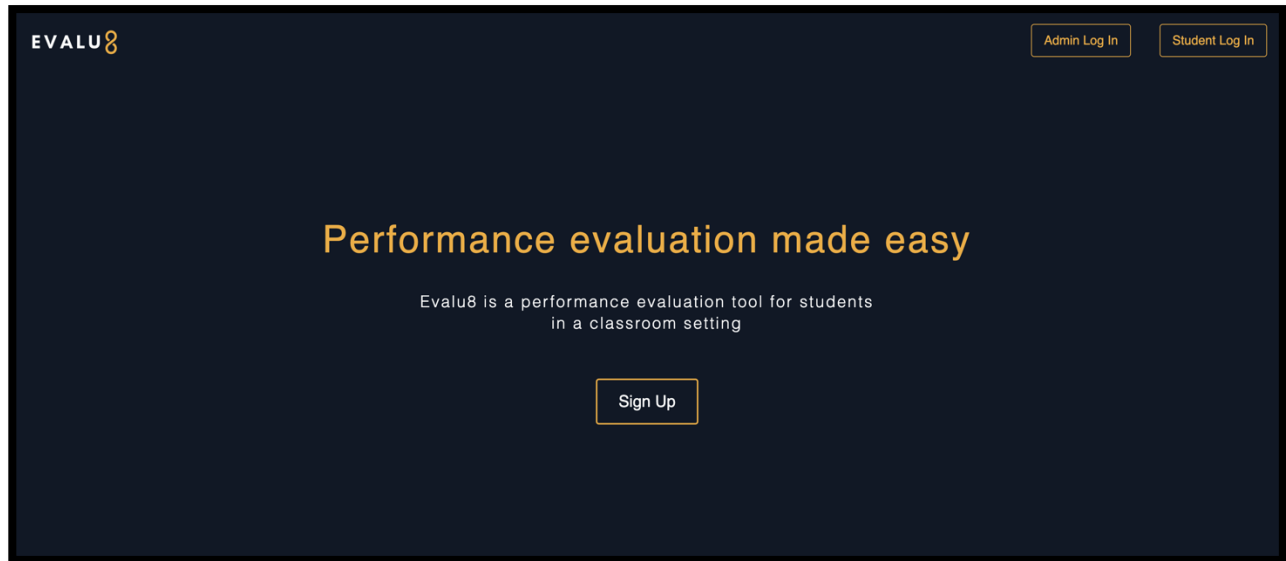
6. References

- Top 10 best performance review software of 2021*. Software Testing Help. (2021, November 29). Retrieved December 14, 2021, from <https://www.softwaretestinghelp.com/performance-review-software/>
- Nielsen, J. (2012, January). *Usability 101: Introduction to usability*. Nielsen Norman Group. Retrieved December 14, 2021, from <https://www.nngroup.com/articles/usability-101-introduction-to-usability/>
- Norman, D. A. (1990). *The design of everyday things*. New York: Doubleday.
- Larson, W. (2019). *An Elegant Puzzle: Systems of Engineering Management*. New York Kim,
- A. (2021, July). *Measuring & Maximizing Developer Productivity*. Harness. Retrieved December 14, 2021, from <https://harness.io/blog/developer-productivity/>
- Stanier, J. (2020). *Become an effective software engineering manager: How to be the leader your development team needs*. The Pragmatic Bookshelf.

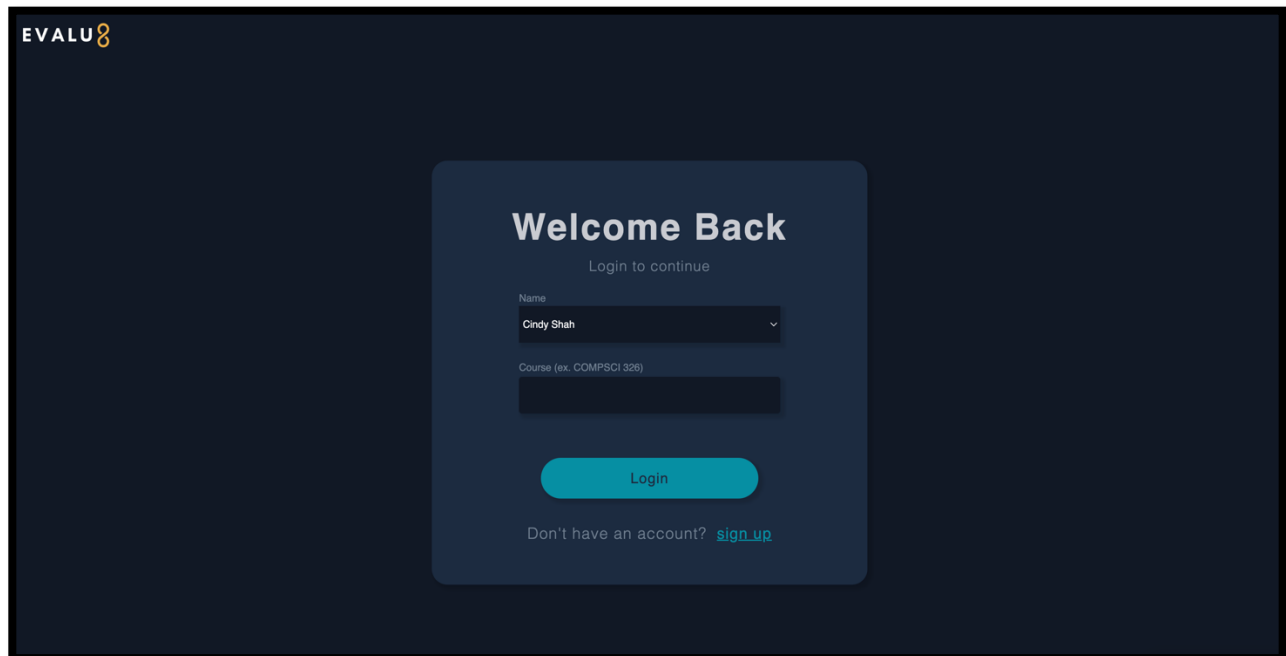
Appendix

Appendix A

Visual of the homepage of Evalu8



Visual of the Admin Login page of Evalu8



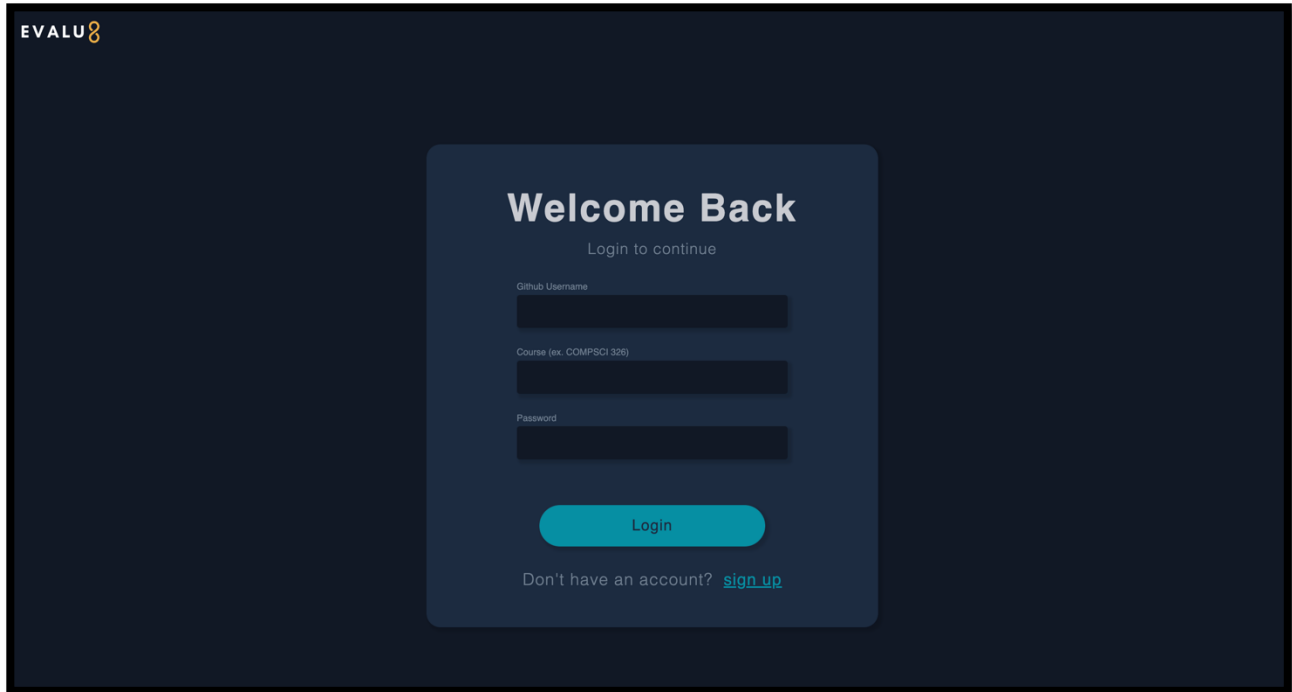
Visual of the Admin Create an Account page of Evalu8

The screenshot shows the 'Create Your Account' page of the Evalu8 application. The page has a dark blue background. In the top left corner, the 'EVALU8' logo is displayed. The main content is centered in a lighter blue rounded rectangle. At the top of this rectangle is the heading 'Create Your Account' in white, followed by the subtext 'Create an account to use Evalu8'. Below this are two input fields: 'Name' and 'Course (ex. COMPSCI 326)'. A red 'Create Account' button is positioned below the input fields. At the bottom of the rectangle, there is a link that says 'Already have an account? [log in](#)'.

Visual of the Admin Landing page of Evalu8

The screenshot shows the Admin Landing page of the Evalu8 application. The page has a dark blue background. In the top left corner, the 'EVALU8' logo is displayed. The page is divided into two main sections by a vertical white line. The left section is titled 'Attendance' in orange, with the subtext 'Choose Attendance Dates' and a white input field below it. The right section is titled 'Evaluation' in orange, with the subtext 'Choose Number of Evaluations'. Below this, there are three rows of input fields: 'Manager Evaluations (0-15):', 'Peer Evaluations (0-15):', and 'Goal-Setting Evaluations (0-15):'. A 'Submit' button is located at the bottom of the right section. In the top right corner, there are three small icons: a back arrow, a trash can, and a document icon.

Visual of the Manager Login Page of Evalu8



The screenshot shows the Evalu8 Manager Login Page. The page has a dark blue background. In the top left corner, the Evalu8 logo is displayed. The main content is a light blue rounded rectangle centered on the page. Inside this rectangle, the text "Welcome Back" is prominently displayed in a large, bold, white font. Below it, the text "Login to continue" is shown in a smaller, lighter blue font. There are three input fields: "Github Username", "Course (ex. COMPSCI 326)", and "Password". Each field is a dark blue rectangle with a light blue border. Below the input fields is a teal rounded button with the text "Login" in white. At the bottom of the rectangle, the text "Don't have an account? [sign up](#)" is displayed, with "sign up" being a teal link.

EVALU8

Welcome Back

Login to continue

Github Username

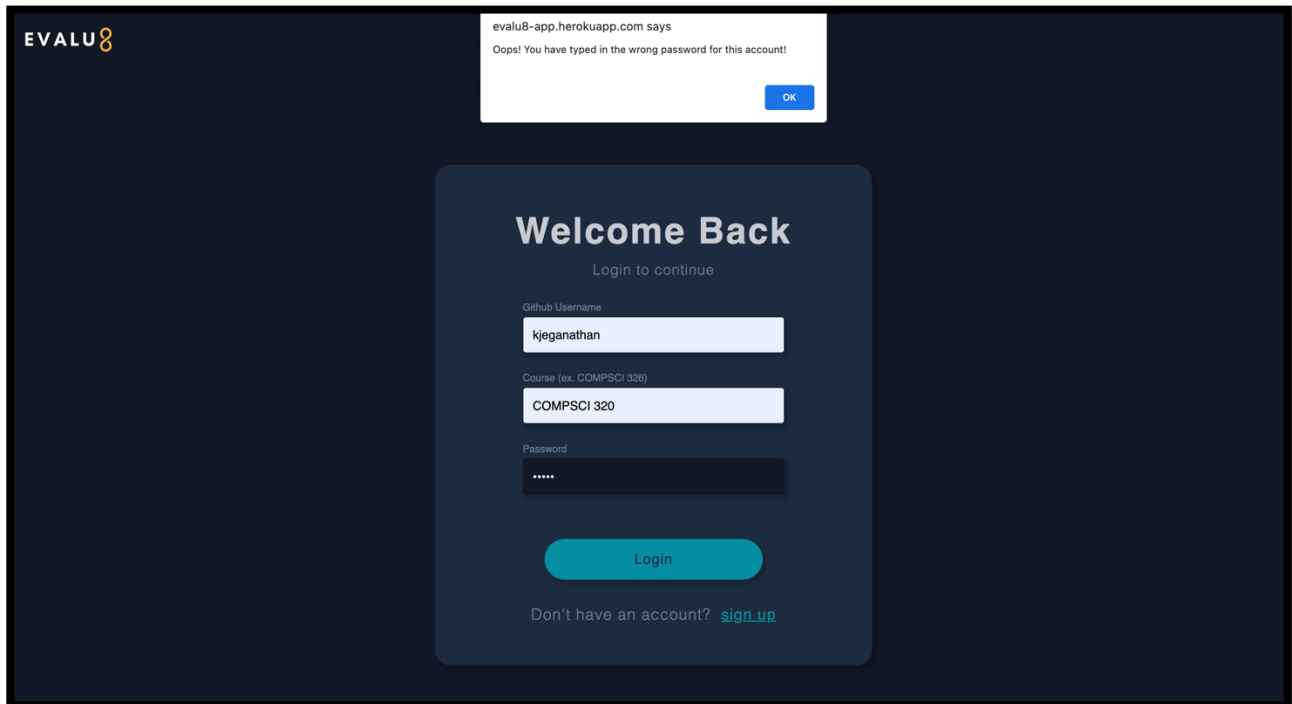
Course (ex. COMPSCI 326)

Password

Login

Don't have an account? [sign up](#)

Visual of the Manager Login Page of Evalu8 with Wrong Password Provided



This screenshot shows the same Evalu8 Manager Login Page as the previous one, but with a password error. A white error message box is positioned at the top center of the page. The message reads: "evalu8-app.herokuapp.com says" followed by "Oops! You have typed in the wrong password for this account!". There is a blue "OK" button to the right of the message. The login form below remains the same, but the "Github Username" field now contains the text "kjeaganathan" and the "Course" field contains "COMPSCI 320". The "Password" field is filled with six dots. The "Login" button and the "sign up" link are still present at the bottom of the form.

EVALU8

evalu8-app.herokuapp.com says
Oops! You have typed in the wrong password for this account!

OK

Welcome Back

Login to continue

Github Username

kjeaganathan

Course (ex. COMPSCI 326)

COMPSCI 320

Password

.....

Login

Don't have an account? [sign up](#)

Visual of the Manager Create an Account page of Evalu8

log_in'."/>

EVALU8

Create Your Account

Create an account to use Evalu8

Name

Course (ex. COMPSCI 320)

Course Professor

GitHub Username

GitHub Repo Name

GitHub Token

Password

Create Account

Already have an account? [log_in](#)

Visual of the Team Member Display page of Evalu8

EVALU8

←

Team Member Display

| Name | GitHub Username | Course | Details |
|-------------------|-----------------|-------------|---------|
| K.J.Justice | kjjustice0818 | COMPSCI 320 | Details |
| Rinija Raja | rinraja | COMPSCI 320 | Details |
| Nisarga Patil | nisargapatil | COMPSCI 320 | Details |
| Disha Srivastava | dishsrivastava | COMPSCI 320 | Details |
| DanGithub200 | | COMPSCI 320 | Details |
| meghana-jaladanki | | COMPSCI 320 | Details |
| Janvi Tanniru | jtanniru | COMPSCI 320 | Details |
| Abby Watterson | awatterson22 | COMPSCI 320 | Details |

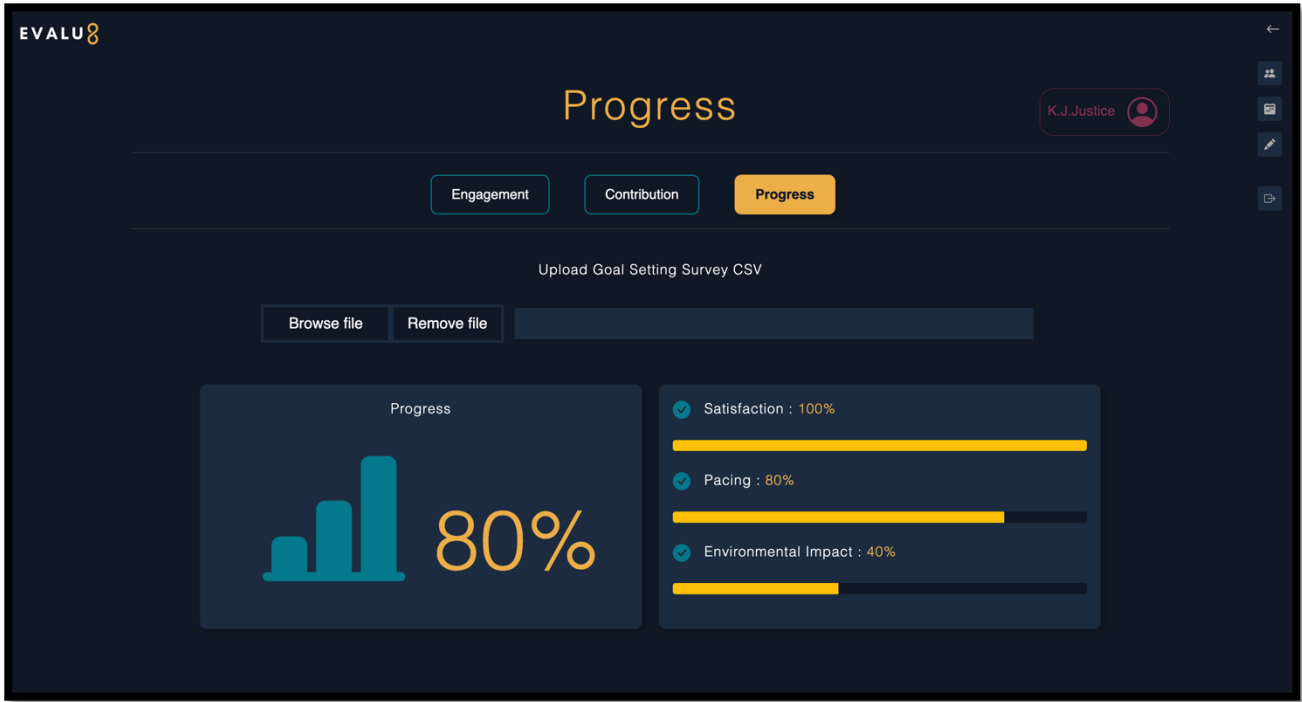
Visual of the Contribution Profile page of Evalu8



Visual of the Engagement Profile page of Evalu8

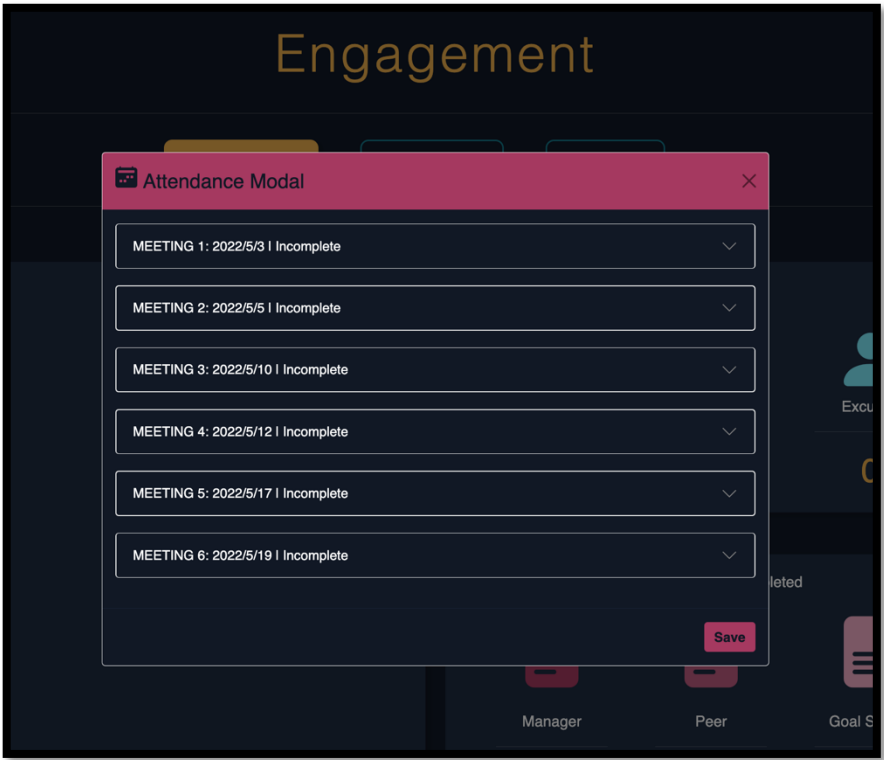


Visual of the Progress Profile page of Evalu8

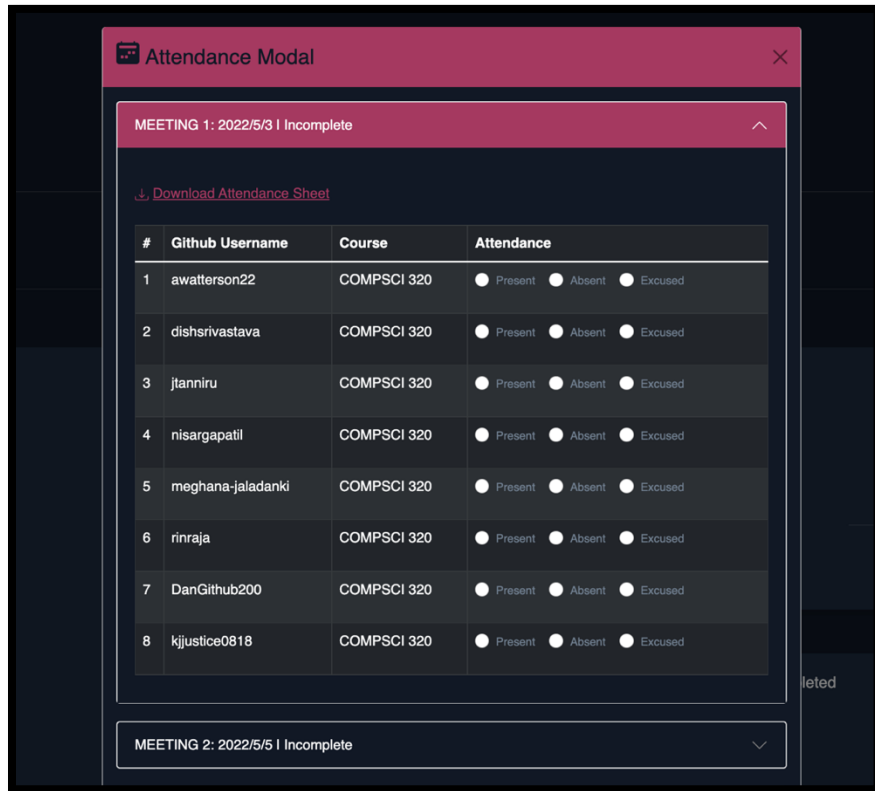


Appendix B

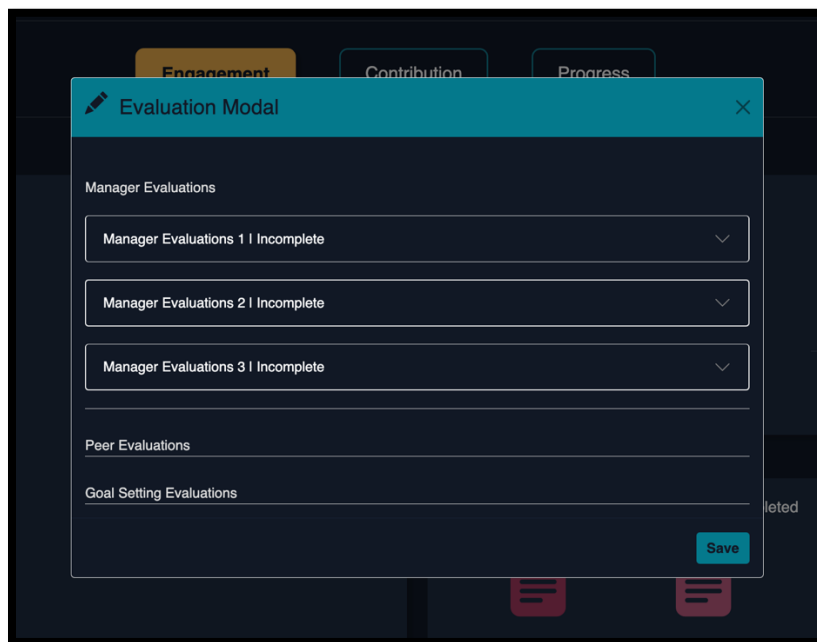
Visual of the Attendance Modal of Evalu8



Visual of an Item Expanded in the Attendance Modal of Evalu8



Visual of the Evaluation Modal of Evalu8



Visual of an Item Expanded in the Evaluation Modal of Evalu8

Evaluation Modal

Manager Evaluations

Manager Evaluations 1 | Incomplete

| # | Github Username | Name | Evaluation Completion | |
|---|-------------------|------------------|---------------------------------|-------------------------------------|
| 1 | awatterson22 | Abby Watterson | <div><div></div>Completed</div> | <div><div></div>Not Completed</div> |
| 2 | dishsrivastava | Disha Srivastava | <div><div></div>Completed</div> | <div><div></div>Not Completed</div> |
| 3 | jtanniru | Janvi Tanniru | <div><div></div>Completed</div> | <div><div></div>Not Completed</div> |
| 4 | nisargapatil | Nisarga Patil | <div><div></div>Completed</div> | <div><div></div>Not Completed</div> |
| 5 | meghana-jaladanki | | <div><div></div>Completed</div> | <div><div></div>Not Completed</div> |
| 6 | rinraja | Rinija Raja | <div><div></div>Completed</div> | <div><div></div>Not Completed</div> |
| 7 | DanGithub200 | | <div><div></div>Completed</div> | <div><div></div>Not Completed</div> |
| 8 | kjustice0818 | K.J.Justice | <div><div></div>Completed</div> | <div><div></div>Not Completed</div> |

Manager Evaluations 2 | Incomplete