

EvaluRate

The Hack Jacks

Dylan Grayson

Conner Swann

Brandon Paree

Brian Saganey

Requirements Document

Version 1.1

Table of Contents

<u>Introduction</u>	<u>2</u>
<u>Problem and Solution Statement</u>	<u>3</u>
<u>Functional Requirements</u>	<u>6</u>
<u>Registration System</u>	<u>6</u>
<u>Login System</u>	<u>6</u>
<u>Password Reset System</u>	<u>7</u>
<u>User Profile Admin</u>	<u>7</u>
<u>Unit Management</u>	<u>8</u>
<u>Evaluation Engine</u>	<u>9</u>
<u>Non-Functional Requirements</u>	<u>11</u>
<u>Scalability</u>	<u>11</u>
<u>Look and Feel</u>	<u>11</u>
<u>Real Time Updates</u>	<u>11</u>
<u>Potential Risk</u>	<u>12</u>
<u>Project Plan</u>	<u>13</u>
<u>Glossary</u>	<u>14</u>

Introduction

In today's workplace, it's almost impossible to avoid working with a group of people on a project. Oftentimes, in order to ensure the project proceeds smoothly, a supervisor needs to communicate individually with team members in order to get any level of valuable feedback. If the supervisor isn't contacting them in person, then it's through some sort of online form or via email, methods that require manual tabulation of results and by no means result in immediate feedback.

EvaluRate is a scalable and customizable web application for the purpose of issuing and managing peer evaluations, eliminating the need for combing through and manually computing results from surveys. EvaluRate provides tremendous value to both managers and team members in the form of time savings and reduction of tedious actions. For managers, EvaluRate is adaptable to pre-existing workflows, allowing them to continue doing what they already do but in a manner that is drastically more efficient. For team members, EvaluRate reduces the amount of time spent circling bubbles and filling in blanks and never again will there be a missed survey with email and text reminders.

The resulting data produced by EvaluRate's evaluation process is crucial information that will allow managers to conduct detailed reviews and make informed decisions about the groups they are administering. In addition, with EvaluRate there is no need to bring detailed statistics or mathematics knowledge to the table, as the platform has a host of visualization methods that allows for ease of sharing results with others in an organization.

Problem and Solution Statement

Our sponsor is Dr. Doerry, a professor at the Computer Science Department at Northern Arizona University. Dr. Doerry teaches multiple Computer Sciences classes each semester and prepares future Software Developers for industry work. Many of these classes involve group projects to give students an idea of how working in industry will be, because most companies that employ software developers have them working in teams. In an academic setting, these groups require some internal checking to ensure academic honesty. Dr. Doerry's solution to that problem was peer evaluations. Having each member of the team rate every other member of the team can expose whether or not a member of the team was pulling their weight, or whether a personal problem exists in the team. Knowing this information means Dr. Doerry can feel like he is giving a fair grade to each group member as an individual, rather than letting one slacker ruin the grade for everyone. Through the academic year Dr. Doerry spends valuable time making, maintaining, and grading peer evaluations, either submitted on paper, or in unstructured emails from students. The solution is a web service that will allow Dr. Doerry, and other professors, to administer and collect data from these peer evaluations quickly and easily.

Professors, however, aren't the only ones that can benefit from this service. In addition to teachers and professors all over the world that assign group projects, peer evaluations can be useful for corporations and teams of all sizes as well. While corporations don't have the problem of academic dishonesty or fair grading, the data gathered from peer evaluations can expose team slackers, and personal problems, which could be helpful for the human resources department to deal with early. It can also give an overall health and morale report for the team manager to act on. Professors use peer evaluations to identify problems that would be detrimental to any team, so it only makes sense that the problem exists for all goal oriented and managed teams. It is likely that managers and organizers of goal oriented teams are not performing peer evaluations at all due to the tedious process that is required, and if an easy solution existed to solve that problem, they would use it.

The solution is EvaluRate. An Evaluation Engine to administer custom evaluations to a select group of people, and collect the raw data, as well as display it in an easy-to-digest way. With the Unit Management system, every group can contain

multiple groups, each of which can contain multiple groups, and so on. This means that it can represent any structure of management. A customizable permissions system will ensure that the resulting data is seen only by those intended. A simple and easy user interface will make doing even complex tasks seem simple.

By implementing a fully customizable Evaluation Engine, group admins will be able to let their subordinates evaluate each other on any number of custom criteria. This will allow almost any team or group oriented organization to be able to use our application to get the most relevant data for their needs. The data from the evaluations will be collected and compiled into multiple different raw data formats. This data will also be available for group admins in easy to read charts and graphs.

The Unit Management system is what gives EvaluRate flexibility when handling different structures of management or group division. It allows a group of managers to individually manage a team of employees, while their manager can administer evaluations to all the managers beneath them. All groups behave the same in that the admin can issue evaluations as well as create new child groups, that may behave the same as the parent, depending on which settings the administrator has chosen for the new groups. There's a good chance this flexibility will lead to use cases that we haven't even thought of yet.

The customizable permission system will have three different types of users being basic users, admins, and parent admins. Basic users are any users that are not admins or parent admins, and parent admins are the admins of parent groups. Each group will indicate which permissions each group has as set by the group's creator. This lets all users of the software feel confident that only those intended are seeing certain information. The anonymity of peer evaluations and survey results is important, because guaranteeing anonymity is how you ensure honest responses, we owe it to our users to have flexible permissions and a secure system.

Of course none of this flexibility will be worth anything if the user interface is too complicated. The average user will not be willing to deal with too steep of a learning curve, so the user interface for EvaluRate will be extremely intuitive. Sliders and drag-and-drop functionality with immediate feedback will model the task as it exists in the world. On top of that, many of the settings will be determined computationally where appropriate.

Our software will change the way people manage teams or even teams of teams, by giving them real time access to evaluation data. The data collected from regular evaluation can indicate that an HR problem may be occurring before it gets too severe, or identify team slackers quickly. What used to take time, energy, and sometimes paper will instead be an easy to use web application with enough flexibility to fit the needs of anybody. Because this application could potentially save a corporation money, it can and will be monetized.

Functional Requirements

1. Registration System

- 1.1. Can create account with NAU Central Authentication System
- 1.2. Can create account with Google Account Credentials
- 1.3. Can create account with basic Email & Password
- 1.4. All users will be able to add the following information to their account.
Some of it will be mandatory, and others, optional:
 - 1.4.1. First and last name
 - 1.4.1.1. Mandatory
 - 1.4.2. Email Address
 - 1.4.2.1. Mandatory
 - 1.4.3. Profile picture
 - 1.4.3.1. Optional
 - 1.4.4. Mini biography
 - 1.4.4.1. Optional
 - 1.4.5. Job title
 - 1.4.5.1. Optional
 - 1.4.6. University attended
 - 1.4.6.1. Optional
 - 1.4.7. Skills
 - 1.4.7.1. Optional
 - 1.4.8. Mobile phone number
 - 1.4.8.1. Required for notifications, otherwise optional.
 - 1.4.8.2. Text message will be sent to the user phone with a pin code
 - 1.4.8.2.1. User must enter the pin code sent to them into the website before using text message notification
- 1.5. Users have to create security questions
 - 1.5.1. Mandatory
 - 1.5.2. Users will pick three security questions from questions that we provided to the users to use

2. Login System

- 2.1. Users can login with password or external services (see 1)

- 2.2. Users have 5 attempts before account is disabled
 - 2.2.1. The disable period will last for 10 minutes
 - 2.2.1.1. An Email will be sent to the user notifying them that their account has been disabled
 - 2.2.1.2. If the user has a mobile number, a text message will be sent to the user notifying them that their account has been disabled

3. Password Reset System
 - 3.1. Users can request a password reset via email
 - 3.1.1. Email is sent to address on record with temporary password
 - 3.1.1.1. Email will be sent to the user notifying them that their password has been changed
 - 3.1.1.2. If the user has a mobile number, a text message will be sent to the user notifying them that their password has been changed
 - 3.2. Users can reset password by answering security questions.
 - 3.2.1. Users will have to answer all three security questions before password reset
 - 3.2.1.1. Email will be sent to the user notifying them that their password has been changed
 - 3.2.1.2. If the user has a mobile number, a text message will be sent to the user notifying them that their password has been changed

4. User Profile Management
 - 4.1. User can modify full name
 - 4.2. User can modify password (see 3)
 - 4.3. User can modify profile picture
 - 4.4. User can modify mini biography
 - 4.5. User can modify skills
 - 4.6. User can modify job title
 - 4.7. User can modify university attended
 - 4.8. User can modify mobile phone number

- 4.9. User can modify security questions
- 4.10. User can delete their EvaluRate account
 - 4.10.1. An Email will be sent to the user notifying them that their account has been deleted
 - 4.10.2. A text message will be sent to the user notifying them that their account has been deleted

5. Unit Management

- 5.1. User can explicitly administer units they are *admin* of
 - 5.1.1. Unit creator is admin by default
 - 5.1.2. Admin can add users to unit
 - 5.1.2.1. Admin can generate a unique code and disseminate code to unit members
 - 5.1.2.1.1. Users create an account and enter code during account creating. These users are automatically added to the unit.
 - 5.1.2.2. Users can be sent an invitation via email
 - 5.1.2.3. Users that currently have accounts can be invited by the admin.
 - 5.1.2.3.1. Invitations will notify the invited user and give the option to accept or decline.
 - 5.1.3. Admin can attach new tasks to a unit
 - 5.1.3.1. Admin can create new peer evaluations
 - 5.1.3.2. Admin can create new survey
 - 5.1.3.3. Every member of a unit will receive a notification for every new task
 - 5.1.4. Admin can modify existing tasks on a unit
 - 5.1.4.1. Admin can modify existing peer evaluations (see 6.2.2)
 - 5.1.4.2. Admin can modify existing surveys (see 6.2.3)
 - 5.1.5. Admin can delete unit and its children
 - 5.1.6. Admin can delete unit and leave children.
 - 5.1.6.1. These new children become root nodes.
- 5.2. User can implicitly administer units that are direct descendants of units they are *admin* of

- 5.2.1. This implicit access can be modified by the creator of descendant units by altering permissions for *parent admins*
 - 5.3. Unit Division System
 - 5.3.1. User with proper permissions can create new subunits
 - 5.3.1.1. Can specify number of subunits based on
 - 5.3.1.1.1. Number of members per subunit
 - 5.3.1.1.2. Number of subunits within unit
 - 5.3.1.2. Can divide users into new units based on the following criteria
 - 5.3.1.2.1. Manual (drag and drop)
 - 5.3.1.2.2. Randomly distribute users into subunits
 - 5.3.1.3. Can add users to subunits after initial grouping
 - 5.3.1.3.1. Manual (drag and drop)
 - 5.3.1.4. Can manage permissions for users in new subunits
 - 5.3.1.4.1. Can create admins of subunits by
 - 5.3.1.4.1.1. Manually assigning admin
 - 5.3.1.4.1.2. Letting a user in the subunit claim admin
 - 5.3.1.4.1.3. Letting all users of subunit vote for an admin
 - 5.3.1.4.2. Can set permissions for *admins*, *parent admins*, and basic users for every new subunit
6. Evaluation Engine
 - 6.1. Create new Evaluation
 - 6.1.1. User can add survey question
 - 6.1.1.1. Qualitative
 - 6.1.1.1.1. "Leave a comment"
 - 6.1.1.1.2. General open ended questions.
 - 6.1.1.1.2.1. The evaluation creator can ask any question that they would like to.
 - 6.1.2. User can add evaluation criteria
 - 6.1.2.1. Quantitative
 - 6.1.2.1.1. Integer response
 - 6.1.2.1.2. Boolean response
 - 6.1.2.1.3. Scale between X and Y
 - 6.1.2.1.4. Zero Sum Slider

- 6.1.2.1.4.1. There are $N * X$ points available, N being the number of group members, X being a value that represents the ideal average score for each member
 - 6.1.2.1.4.1.1. All points must be used
- 6.1.3. User can add evaluation reminders
 - 6.1.3.1. Email reminders
 - 6.1.3.2. Text message reminders
- 6.1.4. User can set whether or not the data collected from the evaluation will be anonymous.
- 6.2. Modify Evaluation
 - 6.2.1. Modifying an evaluation can only occur before any users have taken it
 - 6.2.2. Modifying an evaluation after users have taken it will instead create a copy that can be modified
 - 6.2.3. User can add survey question (6.1.1)
 - 6.2.4. User can add evaluation criteria (6.1.2)
- 6.3. Submit Evaluation
 - 6.3.1. User can fill out peer evaluation
 - 6.3.2. User can submit peer evaluation
- 6.4. Viewing Completed Evaluations
 - 6.4.1. Evaluations
 - 6.4.1.1. The administrator of a specific evaluation should be able to see an overview of the data that is submitted with an evaluation.
 - 6.4.1.1.1. Charts and graphs of quantitative data will be shown, as well as a compiled list of qualitative survey answers.
 - 6.4.1.2. The administrator should be able to view the raw data
 - 6.4.1.3. The administrator should be able to download the raw data of an evaluation in multiple formats.
 - 6.4.1.3.1. Downloaded data will be structured based on the schema associated with an evaluation.

Non-Functional Requirements

1. Scalability.
 - 1.1. The user experience for users in a small classroom should be similar to users at a large company.
2. Look and Feel.
 - 2.1. The UI should be pleasant to look at for long period of time
 - 2.2. The UI should be easily navigated by any users
 - 2.3. The UI should be able to be understood without any tutorials or walkthroughs
 - 2.4. Drag and drop functionality
 - 2.4.1. Drag and drop will be used by unit administrators to place users into units
3. Real Time Updates
 - 3.1. Any updates that happen on the server are given to all connected users without a page reload.
 - 3.1.1. If a user is on a page listing all groups they are in, and they are added to another group, it will display immediately.
 - 3.2. When a user is connected to the web application and any relevant action occurs to their account, the notification system will alert them that had happened.
4. Organization Structure
 - 4.1. Should support school>college>department>class>group
 - 4.2. Should support company>department>team
 - 4.3. Should support one-off groups with no other structure

Potential Risk

There are a few risks associated with building this application. The first one being the existing competition. Right now CATME (Comprehensive Assessment of Team Member Effectiveness), iPeer, and TEAMMATES are the only real players, however EvaluRate still has at least twice as many features as each one. The risk is involved if any of them decide to implement more features to closer resemble EvaluRate. This, however, is unlikely because all of them specifically target academic settings and offer no support for businesses.

Another risk is making the user interface too complex for the average user to use. Since the user scope for this application is very broad, the system itself will require some fine tuning of the settings in order to fit the specific needs of a given user. The way to combat this is to use intuitive commands on the user interface, like drag-and-drop and sliders, which will give immediate feedback as to the result of changing those settings. With enough time dedicated to the user interface, this application will be both powerful and user friendly.

Project Plan

The plan starts with database design and building a client wireframe. We will be doing the database design early to make sure the objects in our schema do not need to be changed later. The client wireframe is so we have a simple client that we can use to test the features that will be built in the following steps. The Evaluation Engine is the biggest task for the backend, and will require a lot of time. The second biggest task for the backend is the unit management system. We will create a rudimentary version of the unit management system in tandem with the evaluation engine so that evaluations can be created for specific units. After some basic functionality is implemented the user interface will be started. The user interface, aside from usability testing, is the largest task for the project, and is therefore allotted the most time. This plan is still a very broad overview of what we intend to accomplish and is subject to change.

week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Database	Backend	Backend	Backend											
Wireframe	Frontend	Frontend	Frontend	Frontend										
Evaluation Eng			Backend	Backend	Backend	Backend	Backend	Backend	Backend	Backend	Backend			
Unit Manage			Backend	Backend					Backend	Backend	Backend	Backend	Backend	Backend
User Interface					Frontend	Frontend	Frontend	Frontend	Frontend	Frontend	Frontend	Frontend	Frontend	Frontend
Usability Test				Testing	Testing	Testing	Testing	Testing	Testing	Testing	Testing	Testing	Testing	Testing

Backend	Frontend	Testing
---------	----------	---------

Glossary

Unit - How groups/teams are represented in EvaluRate.

Evaluation Engine - The software system that encompasses the creation, management, and data analysis of evaluations in EvaluRate.

Admin - Any user that exists in the admins list corresponding to a particular unit.

Parent Admin - The admin of the parent unit corresponding to a particular unit.

Zero Sum - There are $N * X$ points available, N being the number of group members. X being any value. The $N * X$ points are distributed among each member. All points must be used.