

Project Requirements Document - Version 1

Authors:

- Cambria Tolsma (cambriatolsma@ucsb.edu) (Team Lead)
- Rucha Kolhatkar (r_s_kolhatkar@ucsb.edu) (Team Scribe)
- Rain Wang (runyu@ucsb.edu)
- Kyle Gusdorf (kgusdorf@ucsb.edu)
- Hill Xiong (junfengxiong@ucsb.edu)

Team Name: O(MG)

Project Title: Synergi

Introduction:

In most companies, managers have a difficult time quantifying the non-numerical progress of their teams: how individuals are feeling, how they are working with one another, and other similar social aspects. With many teams now being semi- or mostly remote, and with individual employees working across more than one team, the challenge to gauge how one's team is doing outside of completed tasks or projects is even greater than ever before. Not only this, but without consistent face-to-face interaction, employees have fewer opportunities to privately voice concerns to their managers or demonstrate a team's lack of cohesion. If employees cannot express their feelings properly, companies may lose employees due to burn out from mismatched teams or even "toxic working environments". Both managers and employees need a tool to allow for open communication about the non-quantitative aspects of their teams.

One solution, and seemingly the most popular, is having monthly, weekly, or daily meetings. However, these do little to bridge the gap, as people are hesitant to mention issues they have with the project or team in front of all their other team members. When issues are brought up in a team setting, it often causes unnecessary drama or fractures in the team. Another solution is sending out surveys periodically. However most of the time these surveys are just one-time things and are discarded afterwards. They may be more revealing than meetings, but fail to track overall team progress. However, with both meetings and surveys, employees may not always be honest in their responses. Without guarantee of anonymity, it can be hard for their manager to get the full picture of how they're doing and how well they're working with their team.

We are creating a web application that will allow managers to have simple and meaningful insight into their teams' synergy and performance. Through our platform, managers will be able to send out pre-populated surveys to their employees at regular intervals. Employees will then be able to provide both identified and anonymized feedback about previously non-quantifiable data, such as how they are feeling and how well they're working with their fellow team members. Because they will be receiving surveys at regular intervals with standardized questions, managers can monitor trends in their teams and employees over time. Managers can observe trends from the data and use it to make decisions regarding the division of labor or redistribute members if needed.

Implementation Choices:

For the web application, we are using Ruby on Rails as the framework, because our mentors have the most expertise with this framework, and Rails emphasizes "Convention over Configuration" which makes the set up of our web app easy. To make our website more visually appealing, we are using ReactJS for front end design, because it provides more freedom to customize the UI. We are choosing PostgreSQL as our database because it's an easy relational database that works well with Rails and can be deployed on Heroku. The database will be used to store login credentials for managers, as well as survey data and employee responses. From this data, we can extrapolate trends and graph them in ways that would be helpful to a manager so that they are better informed on their team's chemistry and overall mood.

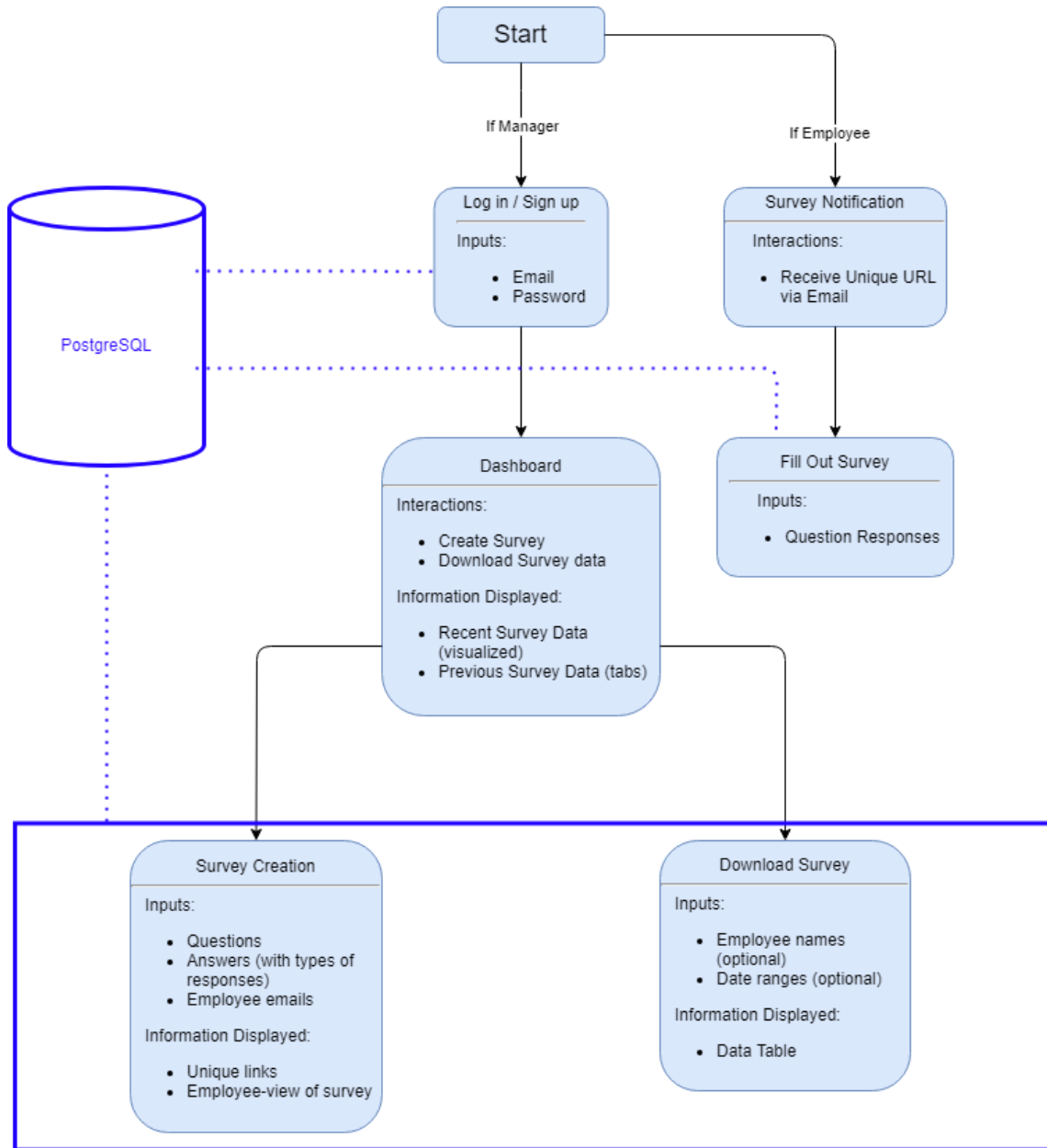
Stretch Goals:

We may include additional features that allow for more customization and flexibility when it comes to tracking data. We may provide a completely customizable survey, where the manager can create and store specific question-answer combinations to be used in future surveys at their discretion - as opposed to choosing from pre-populated surveys with specific question-answer combinations. We may also include an "upload" feature, where managers can upload .csv files and store the data as they would with a survey from our application. This would allow them to host surveys on other sites, such as Google Forms, and be able to compare it to surveys they have already issued through our platform.

We may also include kanban board integration, through platforms like Trello and Github. This would allow managers to compare survey data with the actual task data from their teams: tasks completed, length of time taken on tasks, which team members are working together on a task, and so on.

I. System Architecture Overview

A. High Level Diagram



B. User Interaction and Design: Any user going to our web application will first see the homepage, which will contain information about the project and a button redirect to log in or sign up.

1. If the user is a manager, they can either log in or sign up with an email and password. Once they are logged in they will be redirected to their “dashboard”, or account page. Here they can:
 - Create a new survey using the built in form
 - Download survey data (in the form of a CSV file)
 - See recent survey data visualized in graphs
 - See previous survey data
 - Compare survey data from past and recent surveys
 - Enter in their team member’s names and email addresses into a table (for quick reference when sending out surveys)
2. If the user is an employee, they may skip viewing the home page, as each of them will be given a unique survey link from their manager (in an email).

II. Functional & Non-Functional Requirements

A. User Stories and Use Cases:

1. As a new user, I want to see a clean looking landing page, so I have an easier time navigating through the site.
 - a) Acceptance Criteria: The website has clearly labeled tabs to help the user find more information and buttons to create or sign in to an account.
 - b) <https://github.com/CMT-UCSB/Capstone-AppFolio/commit/cec8dfd85110738786ef9fa4a2110b10a17e868a>
2. As a manager, I want to be able to set up an account if I do not have one already, so that my survey data can be stored.
 - a) Acceptance Criteria: The user has access to registration form. Once the user fills out the form, their login information will be stored so that they can login.
 - b) <https://github.com/CMT-UCSB/Capstone-AppFolio/commit/c274389c7776fa778878dfbe51e4a68600aebae3>
3. As a manager, I want to sign into my account, so I can retrieve saved data.
 - a) Acceptance Criteria: The user has access to the login form and can enter their email and password to login. After logging in, they

should be redirected to the account page where they can see their data or sign out.

- b) <https://github.com/CMT-UCSB/Capstone-AppFolio/commit/c274389c7776fa778878dfbe51e4a68600aebae3>
4. As a manager, if I forget my account password, I want to be able to reset my password with my email.
- a) Acceptance Criteria: The user has access to a link which they can submit their email address. If their email is registered, a password reset link will be sent to their email.
 - b) <https://github.com/CMT-UCSB/Capstone-AppFolio/commit/943fa5cf9ffdc329c7b178c05e7031a7cd84f45f>
5. As a manager, I want to save my team's names and emails on a list, so that I can quickly reference it for sending out surveys
- a) Acceptance Criteria: The user will be able to click on a "employees" section from their dashboard. Here they can enter the names and email addresses of their team members into a table. The table can be recalled from the same section on their dashboard.
 - b) <https://github.com/CMT-UCSB/Capstone-AppFolio/commit/ce92cf33edd3ce820eb41c5df0f0e3782232a843>
6. As a manager, I want to see my survey data visually represented, so I quickly understand the results of my surveys.
- a) Acceptance Criteria: The user will see the survey data depicted as a pie chart and/or bar graph, depending on the question type. If there is more than one question in the survey data, then the user will see one pie chart or bar graph per question.
7. As a manager, I want to be able to download my survey data, so I can share it and import it to other programs as needed.
- a) Acceptance Criteria: The user can select the data of a single survey from their dashboard and download it as a .csv file to their computer.
8. As a manager, I should be able to send a pre-made survey to the employees, so that I don't have to make the survey myself.

- a) Acceptance Criteria: One question survey template that's available for the managers to send out.
9. As a manager, my employees should be able to take the survey using a unique link, so that each employee can only take the survey once and so the data is tied to my account.
- a) Acceptance Criteria: User is sent an email containing hyperlinks to specific survey responses. When the user selects a response, the data is recorded into the database. If the user visits the link again, they are shown a message saying that they have already responded.
10. As a manager, I should be able to compare the results of two or more surveys, so that I can see the differences between them and trends over time.
- a) Acceptance Criteria: When looking at one survey's data, there is an option to bring up one or more additional surveys and visualize the differences side-by-side.
11. As a manager, I want to visualize the change in answers of a specific employee, so that I can gauge how individual employees are doing.
- a) Acceptance Criteria: A specific employee can be selected through the Dashboard. When this is done, their previous responses are displayed together and visualized in the form of a graph.
12. As a manager, I should be able to add notes to the survey data, so that I can track confounding variables in the survey data (such as how my changes are affecting employees).
- a) Acceptance Criteria: The user will be able to select a single survey and click the option labeled "notes". This will open a text box for the user to type in a note or notes related to the survey. The note(s) will be saved along with the survey data.
13. As a user, I should know whether or not my answers will be identifiable or anonymous, so that I can make informed responses to the surveys.
- a) Acceptance Criteria: The user will see an identifiable or anonymous label on the top of the survey when they take one.

A. Technologies Employed

1. Ruby on Rails
2. PostgreSQL
3. ReactJS
4. Heroku
5. GitHub