CSC 179

Section 01 & Section 02

Team: Red Shift

Team Members: Santiago Bermudez, Sabeeha Baqui, Steven Graham, Hardev Singh, Ivan Gutierrez, Eric Truong, Alex Tran

Project Name: Employee Health Dashboard

Deliverable #2

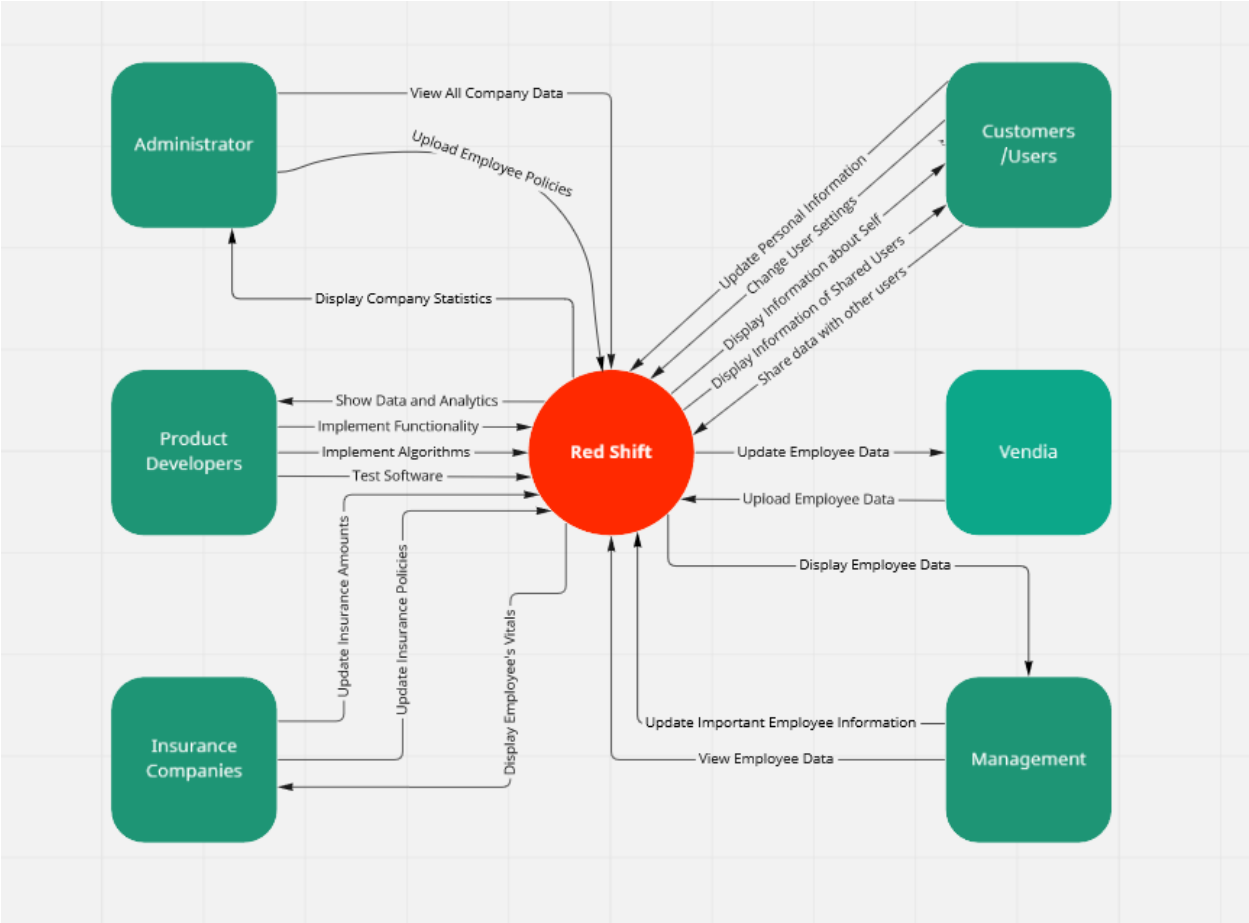
Table of Contents:

1. Software Engineering Team

Team Name: Red Shift

Team Members: Santiago Bermudez, Sabeeha Baqui, Steven Graham, Hardev Singh, Ivan Gutierrez, Eric Truong, and Alex Tran

2. Project Scope



Create a System Context Diagram (SCD) – add a brief description of SCD

Our company should be able to grab employee data from Vendia and allow us to update it if needed. Important vitals by employees should be verified by managers and they will be able to update it for their employees. Users will directly be able to update their personal information and choose who they share that information with. Privacy is of utmost importance for us. Insurance companies will be able to view all this data and upload new policies and plans for employees

based on data. All of this information would be viewed from Vendia and be able to update information in Vendia as well.

3. Requirements Discovery and Analysis

Object Oriented Requirements Analysis (OOA) – Agile User Stories

User Stories

- Develop the necessary User Stories that captures the essence of the project
- Write and **document acceptance criteria for the User Stories**

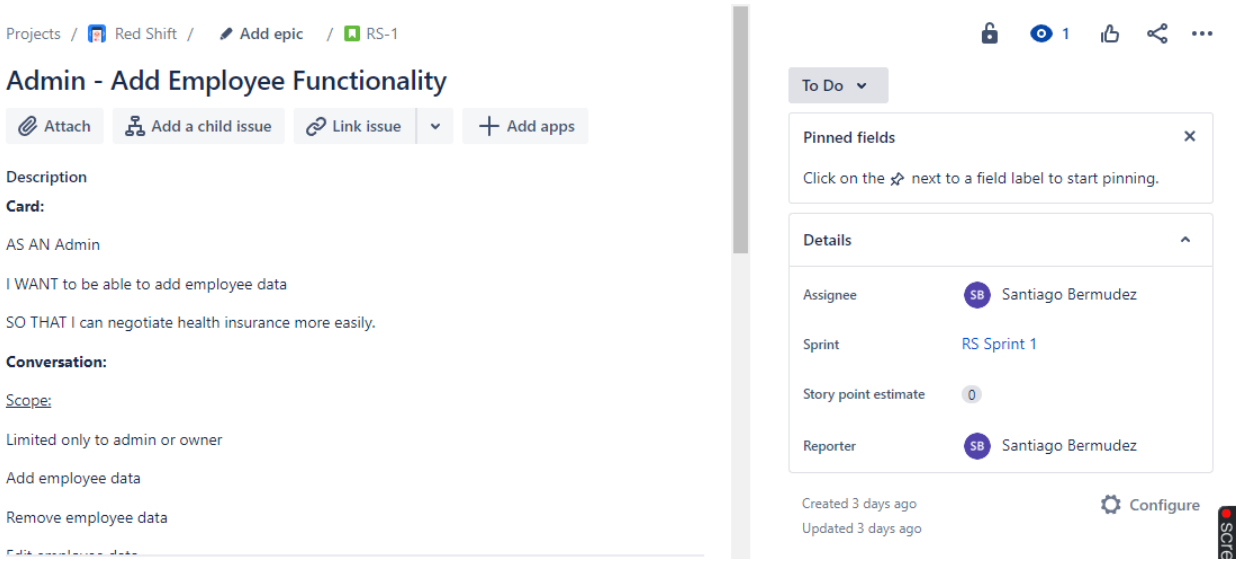
*Please note that we have actually finished a good amount of our user stories, but chose not to mark them as complete so that you can see them! Jira would make the user stories disappear if we mark them as complete!

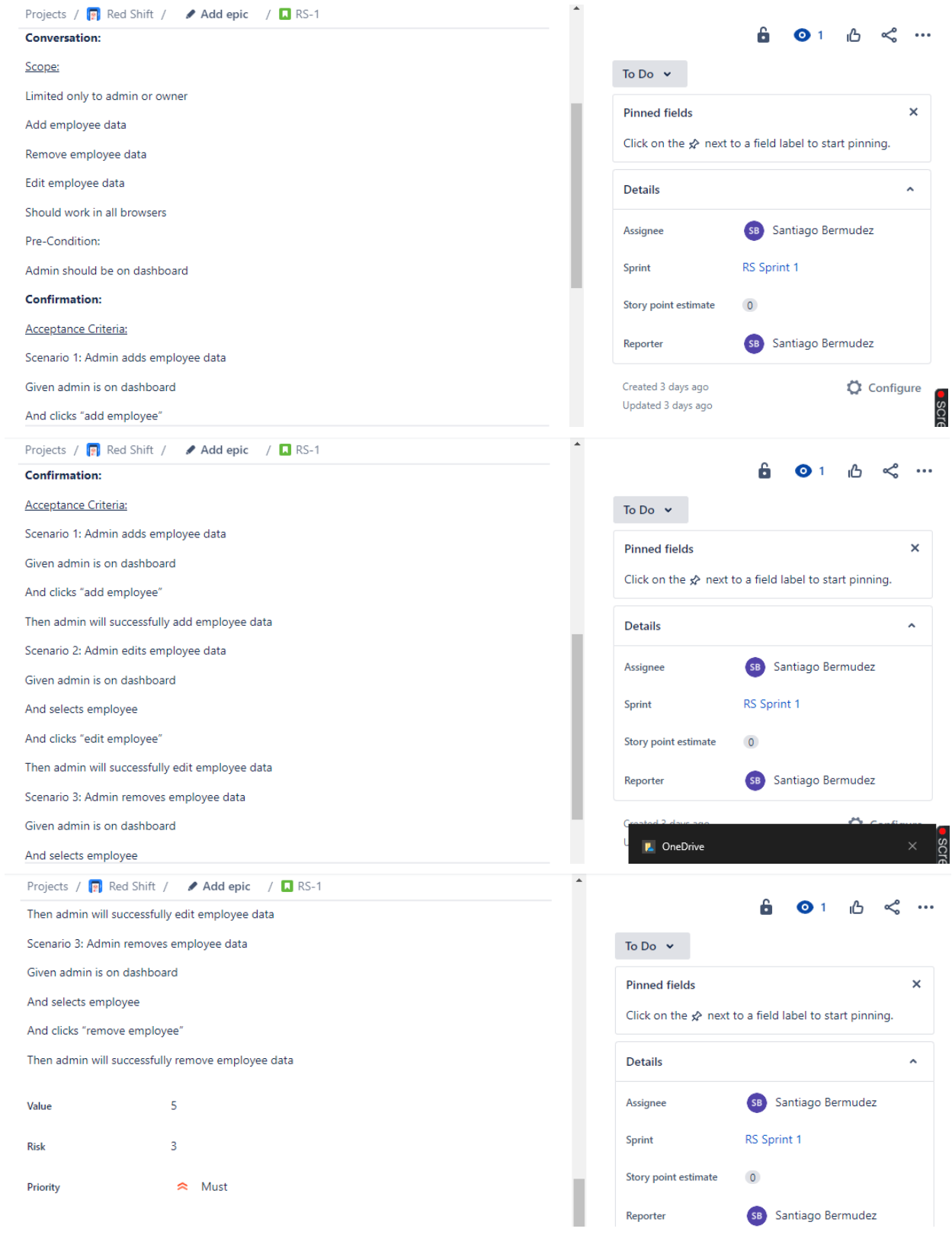
Link to our Jira to view all user stories:
https://santiagobermudez.atlassian.net/jira/software/projects/RS/boards/1/backlog

*You should be able to access the link, but if not, contact:
santiagoabermudez@gmail.com
So, that I can give you access.

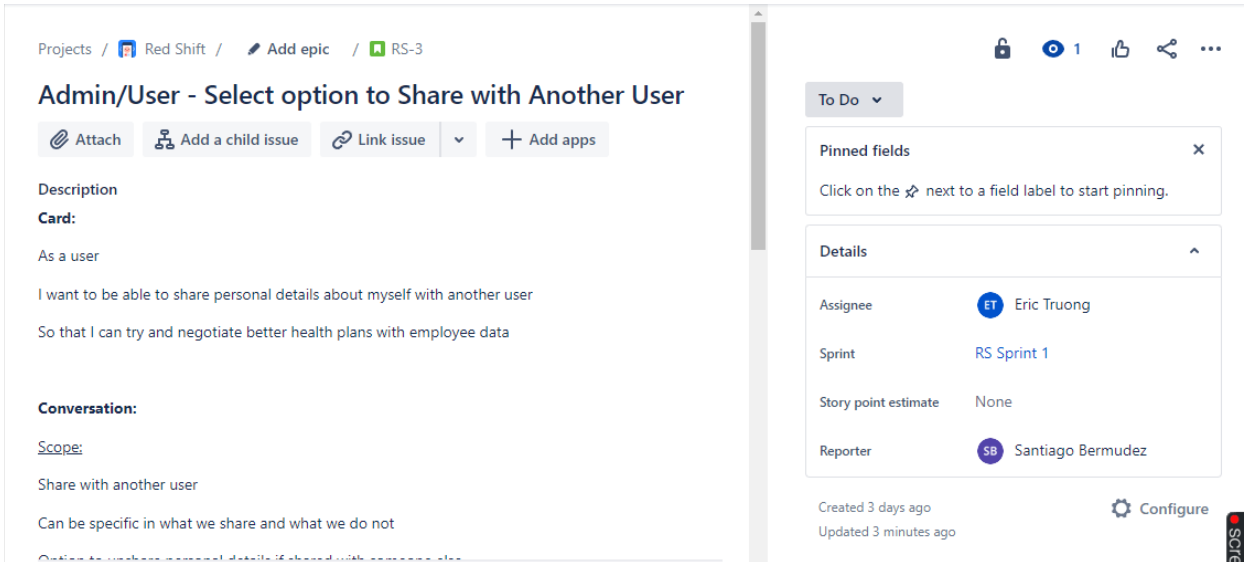
Below are some examples of our user stories, you can see more in the website. We made sure to include prioritization, task assignment, and risk and value assessment within Jira. You can also see our acceptance criteria as well as use of CCC (Card, Conversation and Confirmation) in the descriptions.

Example 1:




Projects / Red Shift / Add epic / RS-1
Conversation:
Scope:
Limited only to admin or owner
Add employee data
Remove employee data
Edit employee data
Should work in all browsers
Pre-Condition:
Admin should be on dashboard
Confirmation:
Acceptance Criteria:
Scenario 1: Admin adds employee data
Given admin is on dashboard
And clicks "add employee"
To Do
Pinned fields
Click on the next to a field label to start pinning.
Details
Assignee
Santiago Bermudez
Sprint
RS Sprint 1
Story point estimate
0
Reporter
Santiago Bermudez
Created 3 days ago
Updated 3 days ago
Configure
Projects / Red Shift / Add epic / RS-1
Confirmation:
Acceptance Criteria:
Scenario 1: Admin adds employee data
Given admin is on dashboard
And clicks "add employee"
Then admin will successfully add employee data
Scenario 2: Admin edits employee data
Given admin is on dashboard
And selects employee
And clicks "edit employee"
Then admin will successfully edit employee data
Scenario 3: Admin removes employee data
Given admin is on dashboard
And selects employee
To Do
Pinned fields
Click on the next to a field label to start pinning.
Details
Assignee
Santiago Bermudez
Sprint
RS Sprint 1
Story point estimate
0
Reporter
Santiago Bermudez
OneDrive
Projects / Red Shift / Add epic / RS-1
Then admin will successfully edit employee data
Scenario 3: Admin removes employee data
Given admin is on dashboard
And selects employee
And clicks "remove employee"
Then admin will successfully remove employee data
Value 5
Risk 3
Priority Must
To Do
Pinned fields
Click on the next to a field label to start pinning.
Details
Assignee
Santiago Bermudez
Sprint
RS Sprint 1
Story point estimate
0
Reporter
Santiago Bermudez

Example 2:



Projects / Red Shift / Add epic / RS-3

Admin/User - Select option to Share with Another User

Attach | Add a child issue | Link issue | Add apps

Description

Card:

As a user

I want to be able to share personal details about myself with another user

So that I can try and negotiate better health plans with employee data

Conversation:

Scope:

Share with another user

Can be specific in what we share and what we do not

1 | To Do

Pinned fields

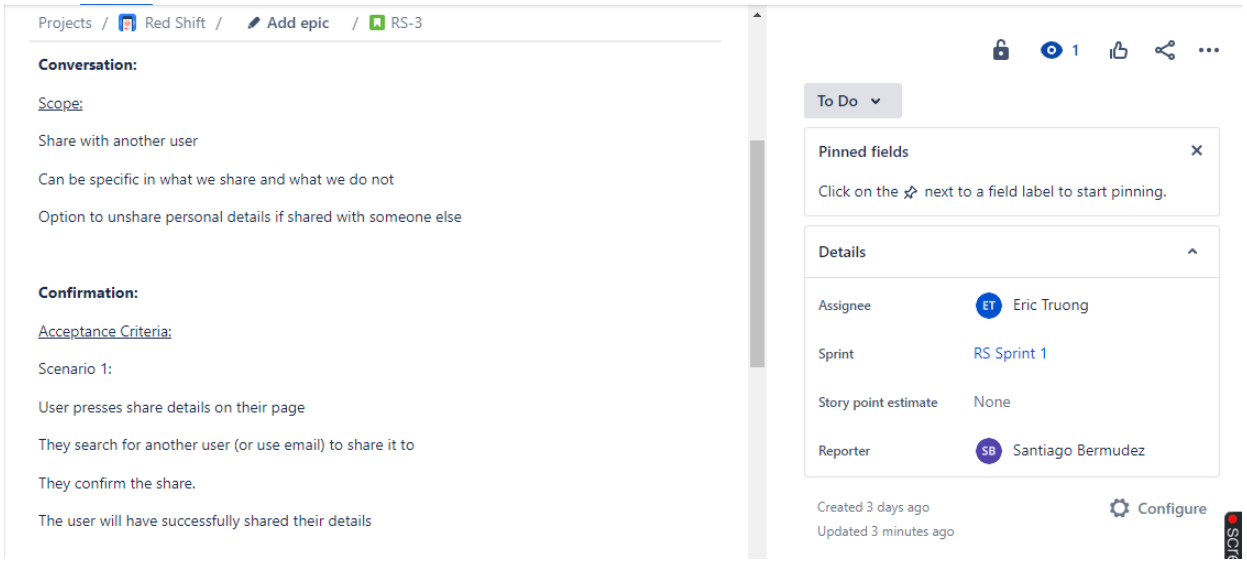
Click on the next to a field label to start pinning.

Details

Assignee	ET Eric Truong
Sprint	RS Sprint 1
Story point estimate	None
Reporter	SB Santiago Bermudez

Created 3 days ago
Updated 3 minutes ago

Configure



Projects / Red Shift / Add epic / RS-3

Conversation:

Scope:

Share with another user

Can be specific in what we share and what we do not

Option to unshare personal details if shared with someone else

Confirmation:

Acceptance Criteria:

Scenario 1:

User presses share details on their page

They search for another user (or use email) to share it to

They confirm the share.

The user will have successfully shared their details

1 | To Do

Pinned fields

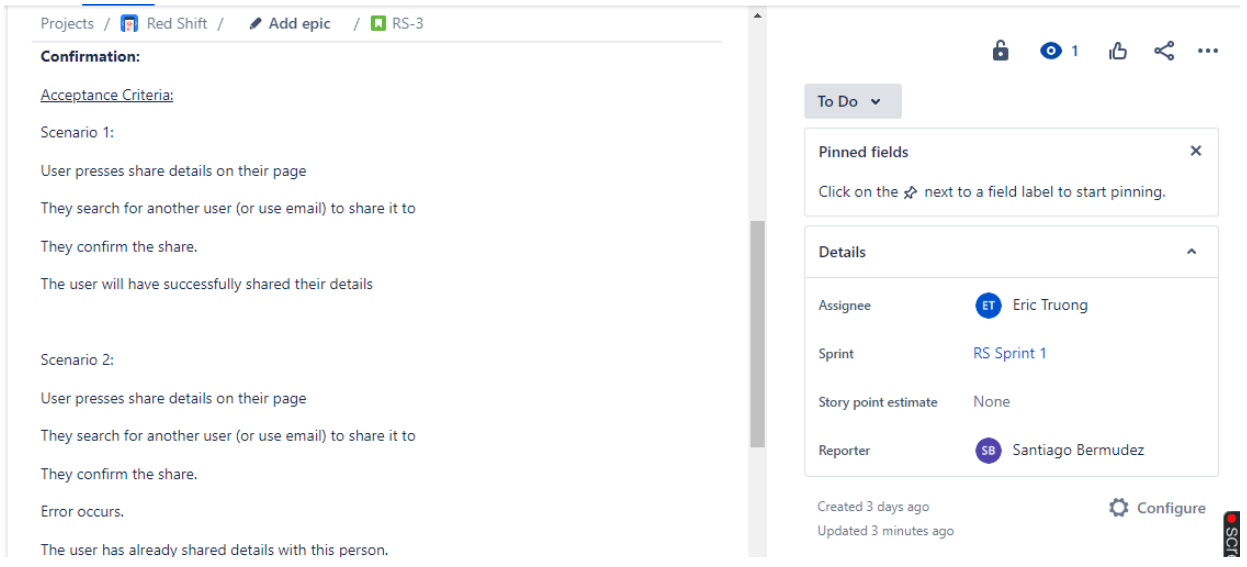
Click on the next to a field label to start pinning.

Details

Assignee	ET Eric Truong
Sprint	RS Sprint 1
Story point estimate	None
Reporter	SB Santiago Bermudez

Created 3 days ago
Updated 3 minutes ago

Configure



Projects / Red Shift / Add epic / RS-3

Confirmation:

Acceptance Criteria:

Scenario 1:

User presses share details on their page

They search for another user (or use email) to share it to

They confirm the share.

The user will have successfully shared their details

Scenario 2:

User presses share details on their page

They search for another user (or use email) to share it to

They confirm the share.

Error occurs.

The user has already shared details with this person.

1 | To Do

Pinned fields

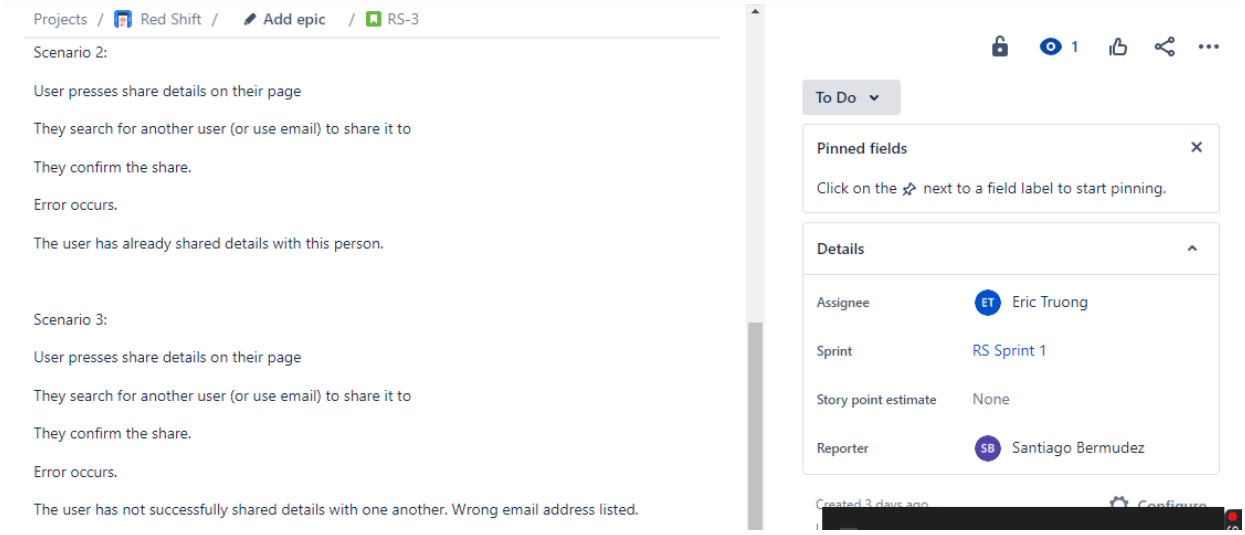
Click on the next to a field label to start pinning.

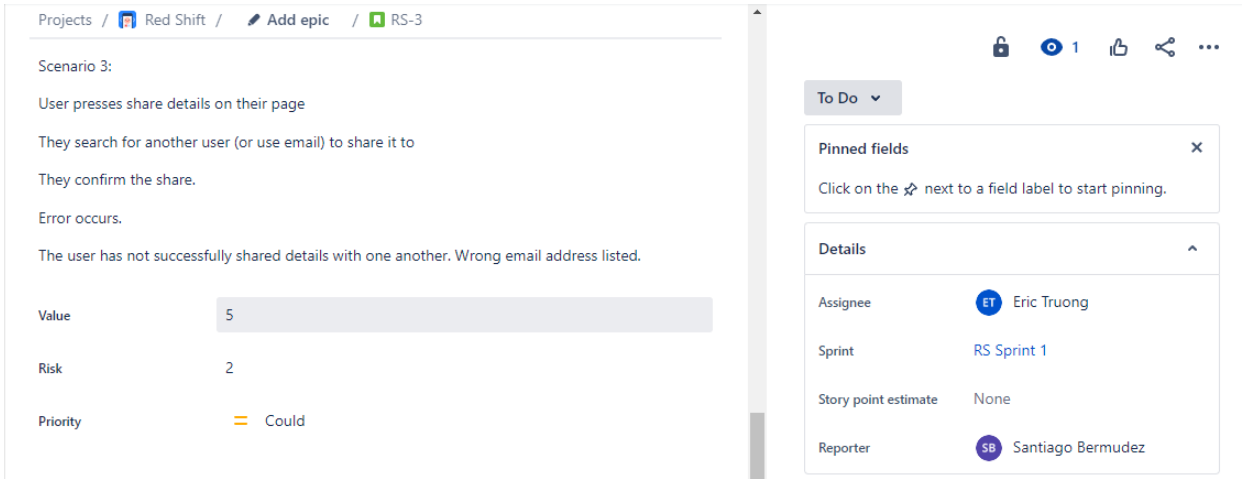
Details

Assignee	ET Eric Truong
Sprint	RS Sprint 1
Story point estimate	None
Reporter	SB Santiago Bermudez

Created 3 days ago
Updated 3 minutes ago

Configure

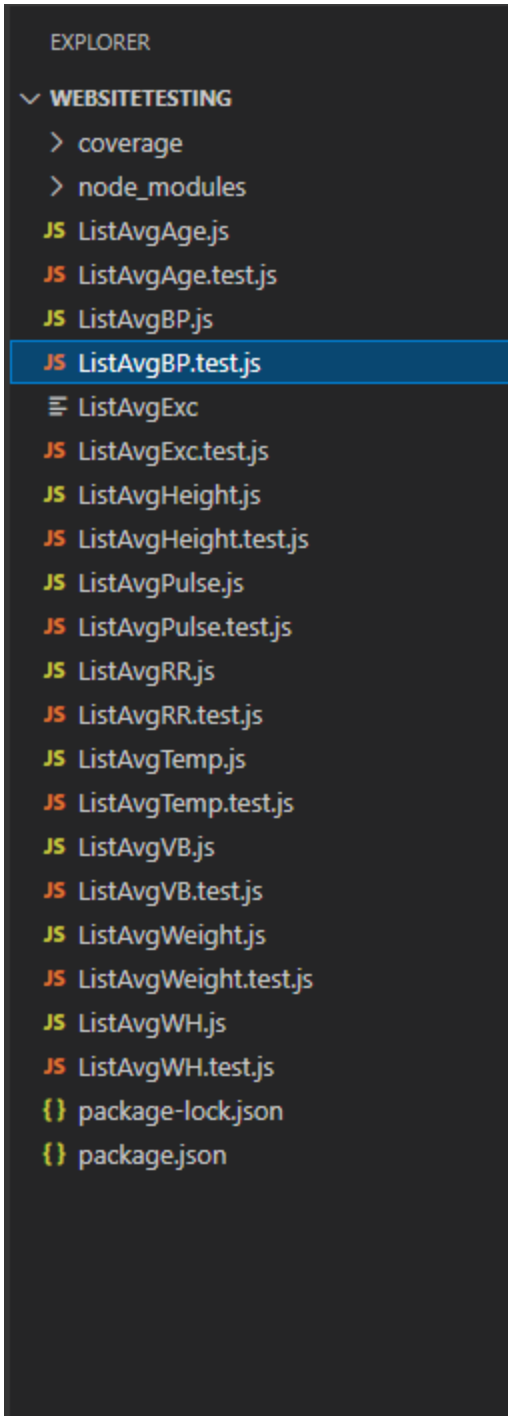


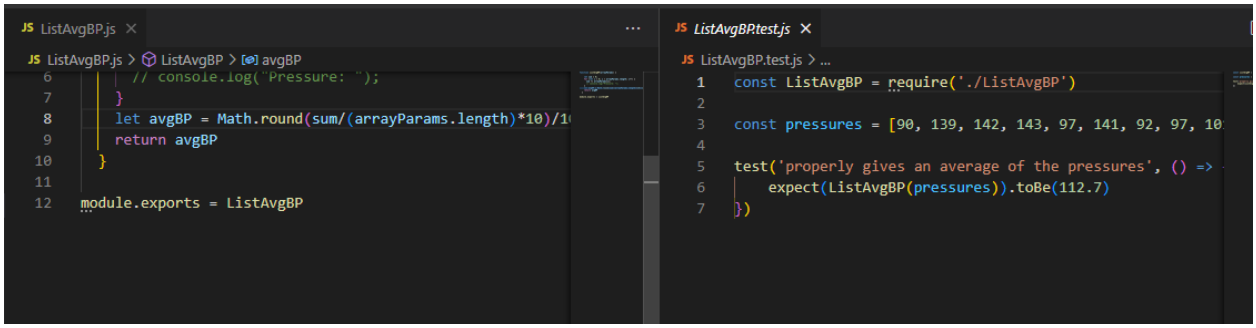


Testing:

- Describe the teams test strategy - Create test plan (see test plan doc and sample test plan)
- Design the test cases based on the User Stories and the **acceptance crater for the User Stories**
- Document all test cases and relevant test results –* see the test report

The test plan involved using an automated tester and this is Integration testing which was our main strategy. Below are some examples of our testing. We used Jest which is a testing framework in JavaScript. These are snippets of the testing process. You can see one of our tests failing below and what that result looks like later when all the tests work properly.


EXPLORER
WEBSITETESTING
coverage
node_modules
ListAvgAge.js
ListAvgAge.test.js
ListAvgBP.js
ListAvgBP.test.js
ListAvgExc
ListAvgExc.test.js
ListAvgHeight.js
ListAvgHeight.test.js
ListAvgPulse.js
ListAvgPulse.test.js
ListAvgRR.js
ListAvgRR.test.js
ListAvgTemp.js
ListAvgTemp.test.js
ListAvgVB.js
ListAvgVB.test.js
ListAvgWeight.js
ListAvgWeight.test.js
ListAvgWH.js
ListAvgWH.test.js
package-lock.json
package.json



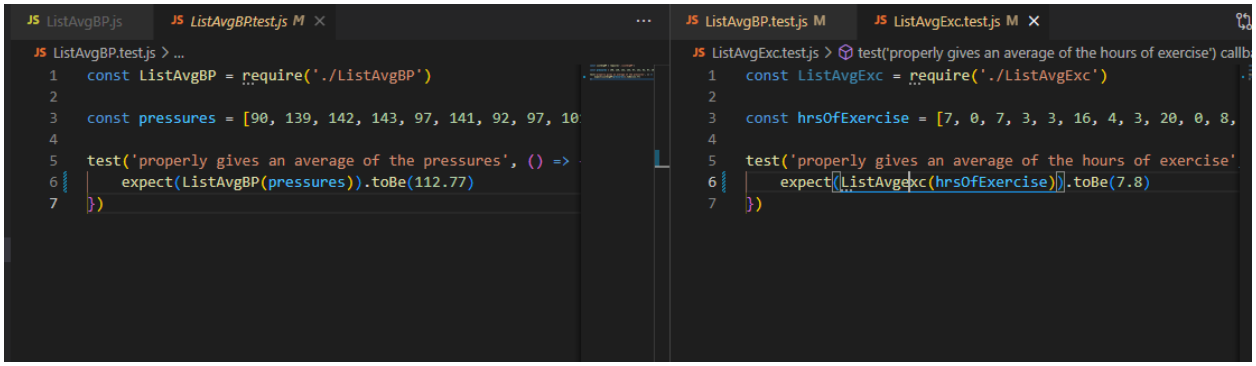

```
PS C:\Users\steve\OneDrive\Documents\GitHub\websiteTesting> npm test
npm WARN config global `--global`, `--local` are deprecated. Use `--location=global` instead.

> websitetesting@1.0.0 test
> jest --coverage

FAIL ./ListAvgBP.test.js
  ● properly gives an average of the pressures

    expect(received).toBe(expected) // Object.is equality

    Expected: 112.77
    Received: 112.7

      4 |
      5 | test('properly gives an average of the pressures', () => {
    > 6 |     expect(ListAvgBP(pressures)).toBe(112.77)
        |                                  ^
      7 | })

      at Object.toBe (ListAvgBP.test.js:6:34)

PASS ./ListAvgWeight.test.js
PASS ./ListAvgVB.test.js
PASS ./ListAvgWH.test.js
FAIL ./ListAvgExc.test.js
```

Here, we can see that average blood pressure was given incorrectly thus failing ./ListAveExc.test.js. Accuracy matters in this case even if the expected result was off by a miniscule amount.

Expected: 112.77
Received: 112.7

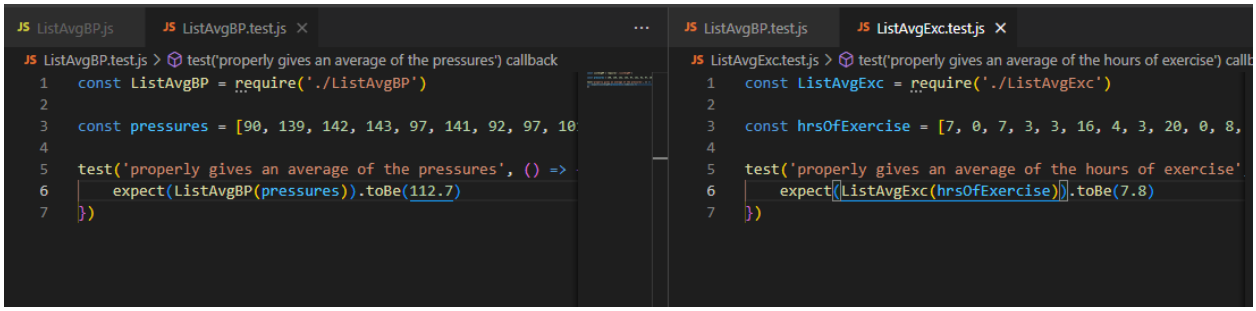
***Test has failure**

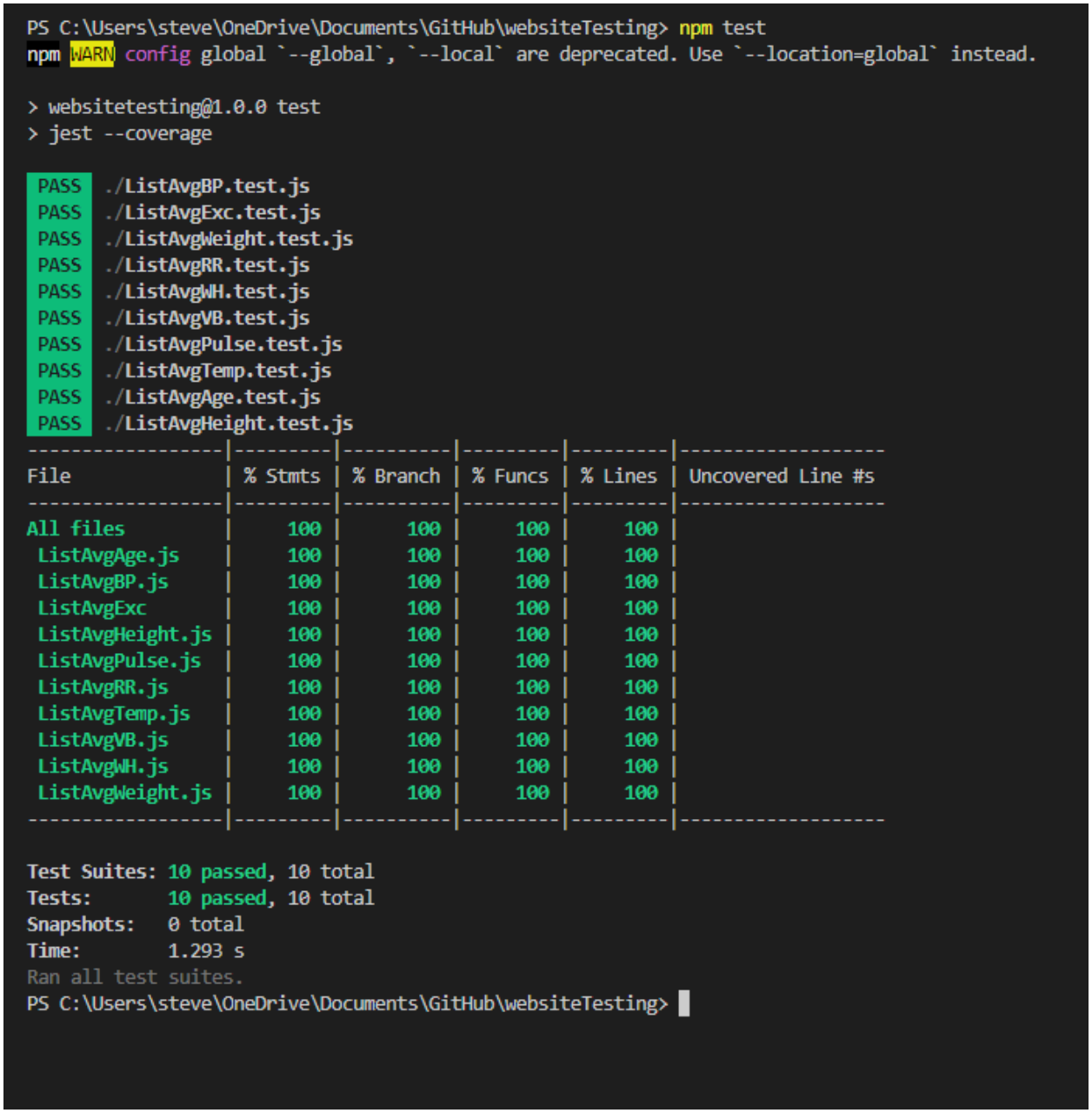
Here, we can see that the test has a failure. 2 of the tests have failed, 8 of the tests have passed, and there are 10 tests in total. The error resided in the ListAveExc file.



Error on BP is too many decimals

Error on Exercise is ListAvgexec instead of ListAvgExec





*** All tests passed**

Here, we can see that all the tests have successfully passed after all the corrections were made.

10 passed out of 10 in total.

4. System Requirements

4.1 FRs- Based on the project description and the Use Case model, list all system functional requirements.
Number the Functional Requirements (FR1, FR2, FR3, etc.) in a systematic manner.

FR1. All data must be stored through Vendia.

FR2. Employee data must be formatted in the given format and must include all given parameters such as medical info, demographic info, and physical health info.

FR3. The project must be tested with automated testing using Selenium or equivalent.

FR4. View aggregate data for all employees without seeing specific names. Be able to select gender, see averages for 4-12 of employee data without gender selected, and see total averages for 4-12 of employee data.

FR5. Select option to share with another user using a second Vendia account from your team, invite someone to participate and create a partner node.
a. Share with the second account the data without personal identification information - in this case we only have the name that needs to be held back.
b. Revoke access to the second account.
c. Note that demonstration can be done from Vendia itself. To keep this project sized for summer there doesn't need to be a second Web App to display what the second account can see.

FR6.

1. Do more than Average...
a. Standard Deviation
b. Mean
2. Find different cross-sections for the data. Some ideas (you may have more)
a. % of gender working more than 40 hours a week
b. Do people that workout more than x hours per week have
i. More/less time off available
ii. Lower/higher blood pressure
iii. Lower/higher respiration rate
3. Come up with other vitals for the employee
4. Create a mini page for the second account that displays info nicely and uses some key
metric to determine if they will provide cheap or expensive health coverage.
+

4.2 NFRs - system attributes such as usability, reliability, and performance, etc.

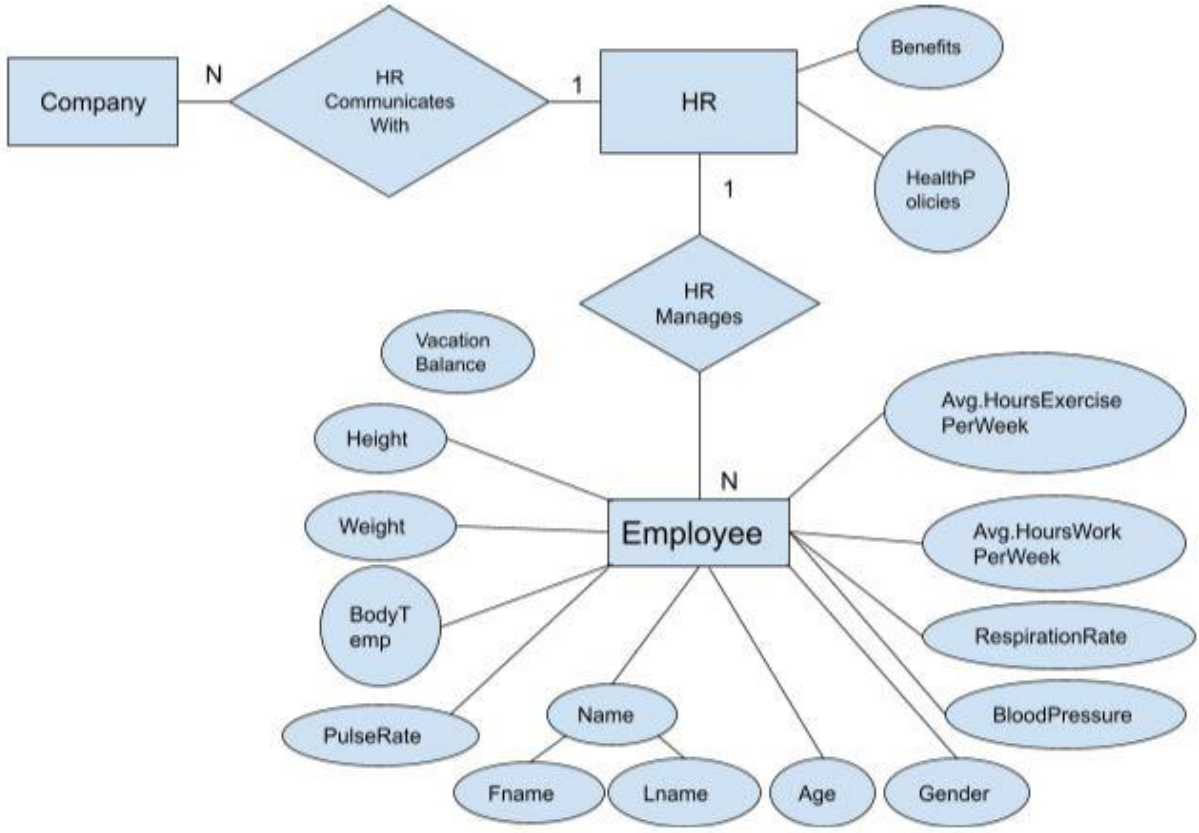
NFR 1. Implement minimum features and create a view for each patient's details.

NFR 2. It should be easily maintainable and serviceable throughout its lifespan and development.

NFR 3. It should be able to handle usage by a few users, each varying in status and with differing uses of the system.

5. Data Design

Develop an ERD diagram - Briefly describe the ERD

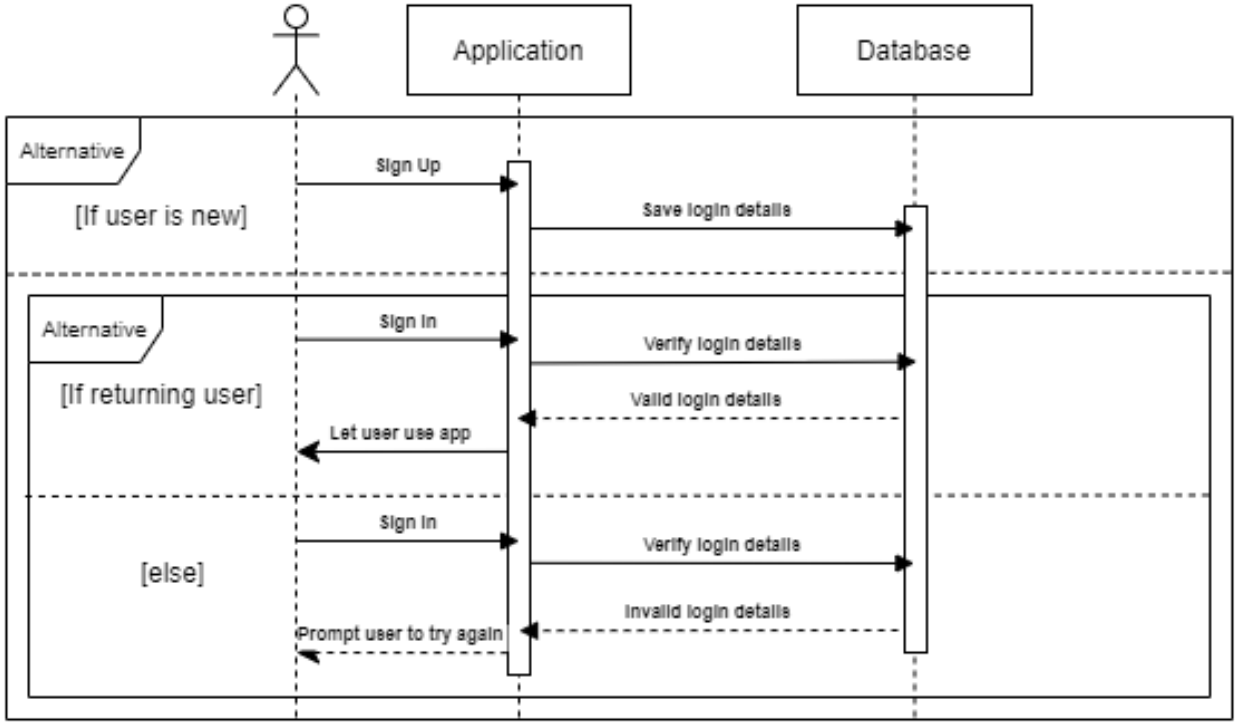


The ERD you see above illustrates how “entities” like employees, HR, and companies relate to each other within a system. Here, you can see that the database will store employee info, like their names, ages, weights, and so on. Each employee is managed by HR and HR could be communicating with multiple companies.

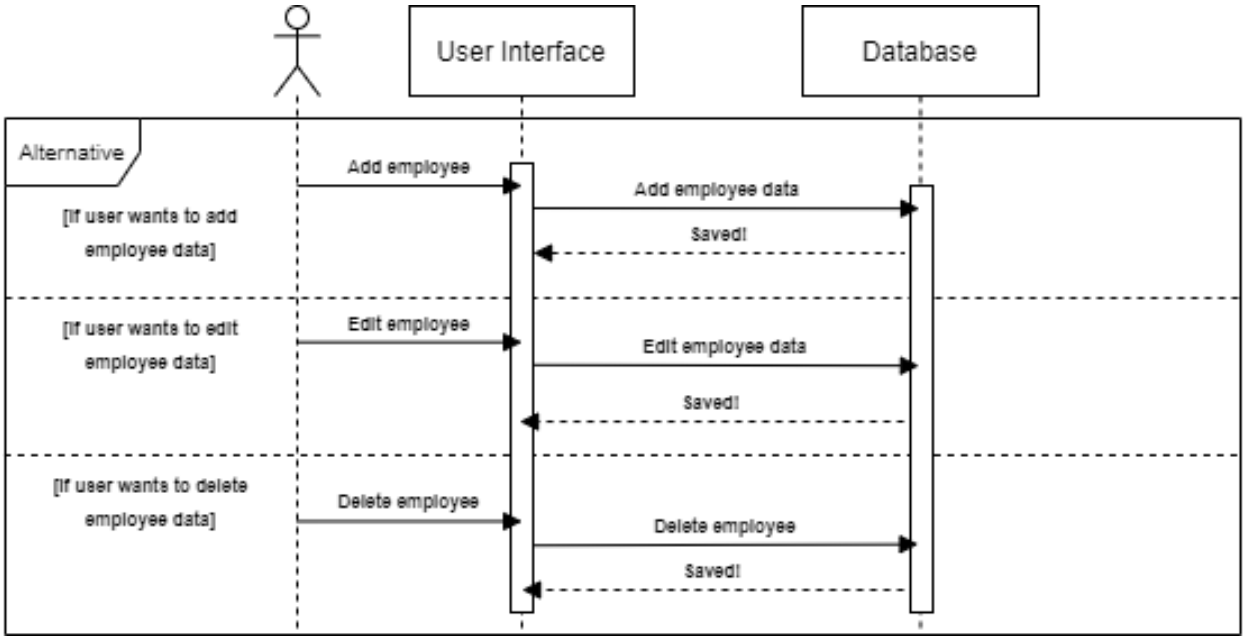
6. Detailed Design

Create a set of interaction models (i.e. sequence diagrams) to capture low level design view of the system

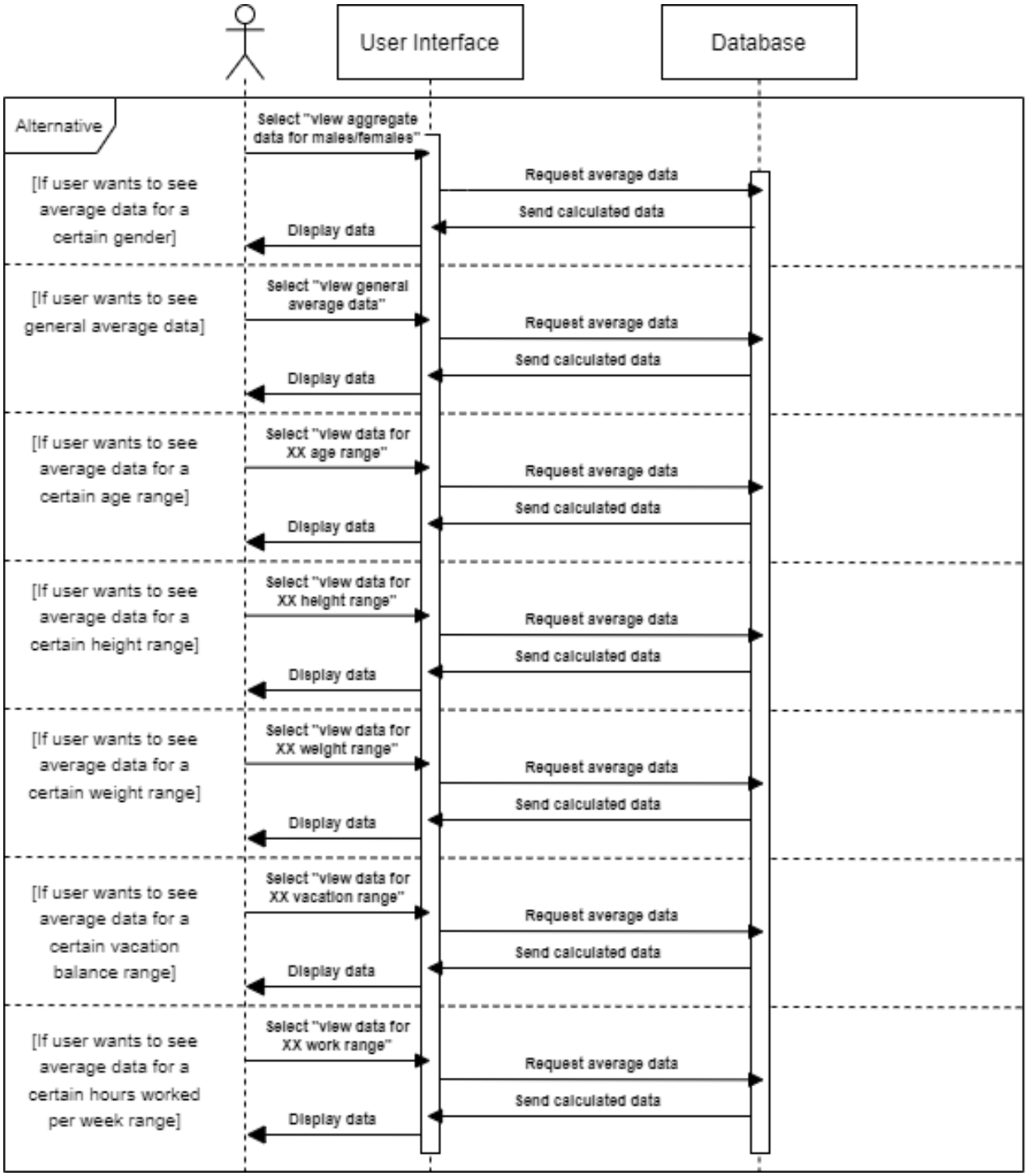
a) For logging in



b) For adding/editing/deleting employee data

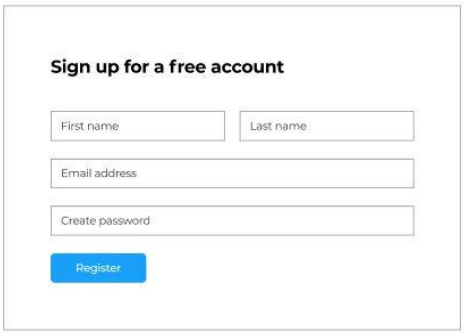


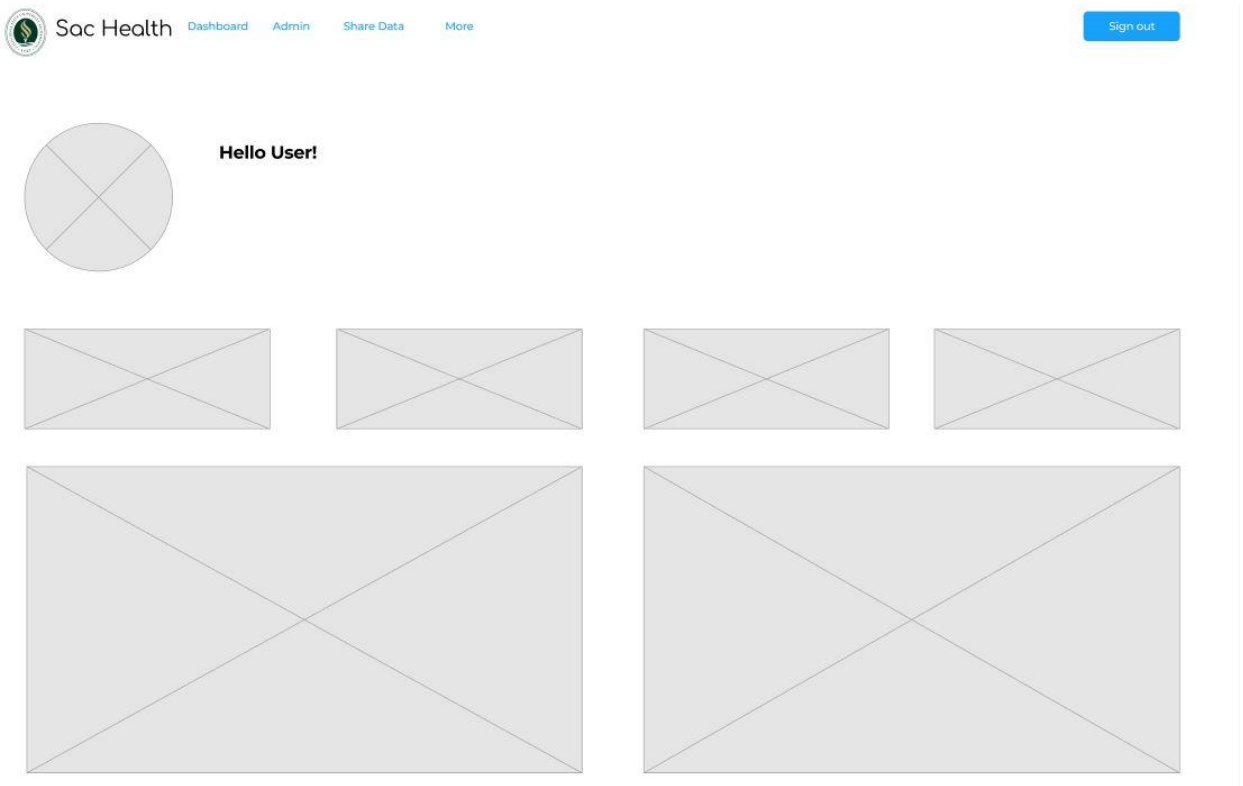
c) For viewing aggregate data



7. User Interface Design

The Interface Design describes internal and external program interfaces. Interface designs are based on the information obtained from the analysis models. Use Cases, User Stories, and Sequence diagrams to capture the interface design.


Sign up for a free account
First name
Last name
Email address
Create password
Register


Sac Health
Dashboard
Admin
Share Data
More
Sign out
Hello User!

8. Technology and Tools

- Frontend: React JS
- Backend: Vendia
- Database: Vendia
- Version control: GitHub
- Javascript
- Discord
- Visual Studio Code (VS Code)
- Google Docs

9. Assumption and Constraints

Any relevant assumptions and any special design issues, which impact the design or implementation of the software, are noted here

- Assume that a second WebApp is not needed due to time constraints.
- Use a minimum number of features on the WebApp in order to reach your goal.
- In order to keep the project in scope, the focus will be on creating the dashboard and the ability to share selected fields of the employee health data with an insurance company.