CS 495 Design Day

Hunter Wallace, Chase Watkins, Luke Holliday, Nathaniel Barber

Business Requirements

- BR1 - Ensure Product Health/Uptime - Ensuring that Accutech’s other applications are healthy and running is an important feature of this software. With the software constantly checking the status of the other applications and alerting an admin if there is a problem, allows for faster response to any potential problem. That way the other applications can get back on track doing what they’re supposed to do and that they are healthy.

- BR2 - 24/7 Monitoring (Especially Night time) - 24/7 monitoring will help in causing less issues when starting off the work day at 8 am. Using a system that will pull information for errors and send them to the Command Console (BR3) there will be a constant view on common issues that go wrong. Instead of having a person watching for these issues manually, there can be automation set in place to save on time. Getting alerts for when things critically go wrong would be a lot more helpful than walking in on a new work day and finding the system went entirely down.

- BR3 - Centralized Command Console - A centralized console to be viewed if needed to check statuses or milestones for applications. This information will be pulled from each tracked application automatically without needing manual requests.

Use Cases

Actors

- Accutech Admins - The admins will be the primary users for this product as they will moderate the system and respond to any warnings it may provide
- Potential Third Party - While Accutech may be the primary user and owner of the product, they have mentioned that if the need arises, they may hire an external team to use the software and help respond with any issues that appear

Use Cases

- UC1 - General System Health - Overall, the software is meant to generally monitor the health of the products and give a quick glance over view as opposed to looking at each one individually. The actor for this would be the admins most likely and the flow would simply be going to the page (most likely with correct credentials) and looking at the dashboard. This is tied to BR1 (Ensure Product Health/Uptime).
- UC2 - Alerts in Response to Errors - Using UC3 information on errors will be collected. These errors will be viewed on criteria on certain levels of severity. These can lead to different types of alerts to be sent out. Such as emails or texts to the Accutech Admins or Third Party. Will update UC1 with general health checks. This is tied to BR2 (24/7 Monitoring (Especially Night time)).
- UC3 - API calls from other applications - This software should be able to get information from other software. The software will automatically be taking information from these other applications 24x7. It's going to get the information from API calls. The information that comes in will be checked for any errors which will result in UC2. The admins at Accutech are the actors for this use case who are the ones who need to view the information coming in.This is tied to BR3 (Centralized Command Console)

Domain Model

- Contacts
  - Contact is used to store all the people that need to be or will be contacted if something goes wrong. Contacts connects to a table with all the groups that contact is in as well as responses that the contact would receive.
- ContactInfos
  - Contact info will hold information for contacting people from health checks. Once a response is given, the users that will need to be contacted will have their info found, such as email or text info.
- ContactGroups
  - Contact groups deal with all the different kinds of groupings that a contact can have. For example a manager or developer. These will help in notifying only certain groups of people to contact.
- Triggers
  - Can parse conditions from different variables and can execute different code based on the variables.
- Variables
  - Constraints used for the different levels of triggers.

Domain Model

- User
  - The Users table is vital because it contains the users that will be able to login to the monitoring software and check on product health. It is important to note that users are not the only ones that will be notified when a product failure happens, thus leading to the contacts table.
- HealthChecks
  - The HealthChecks table is the core feature of this product and will contain the generated scheduled checks that will be done on the programs. It also contains the API calls that will be run.
- Roles
  - Stores roles from information of users.
- Schedule
  - Schedule stores information about different scheduled health checks. It connects to a table with all the different groups of health checks.
- Responses
  - Response is used to store all different responses that we would need to alert to someone. Responses also connects a contact and the table with all the groups.

Requirements

Functional Requirements

- FR1 - 100% Automated - HIGH - BR1
- FR2 - Centralized Command Console - HIGH - BR3
- FR3 - Console to PULL status details from applications - MEDIUM - BR3
- FR4 - Linked/downstream impacts identified - MEDIUM - BR1
- FR5 - Automatic alerts - HIGH - BR1

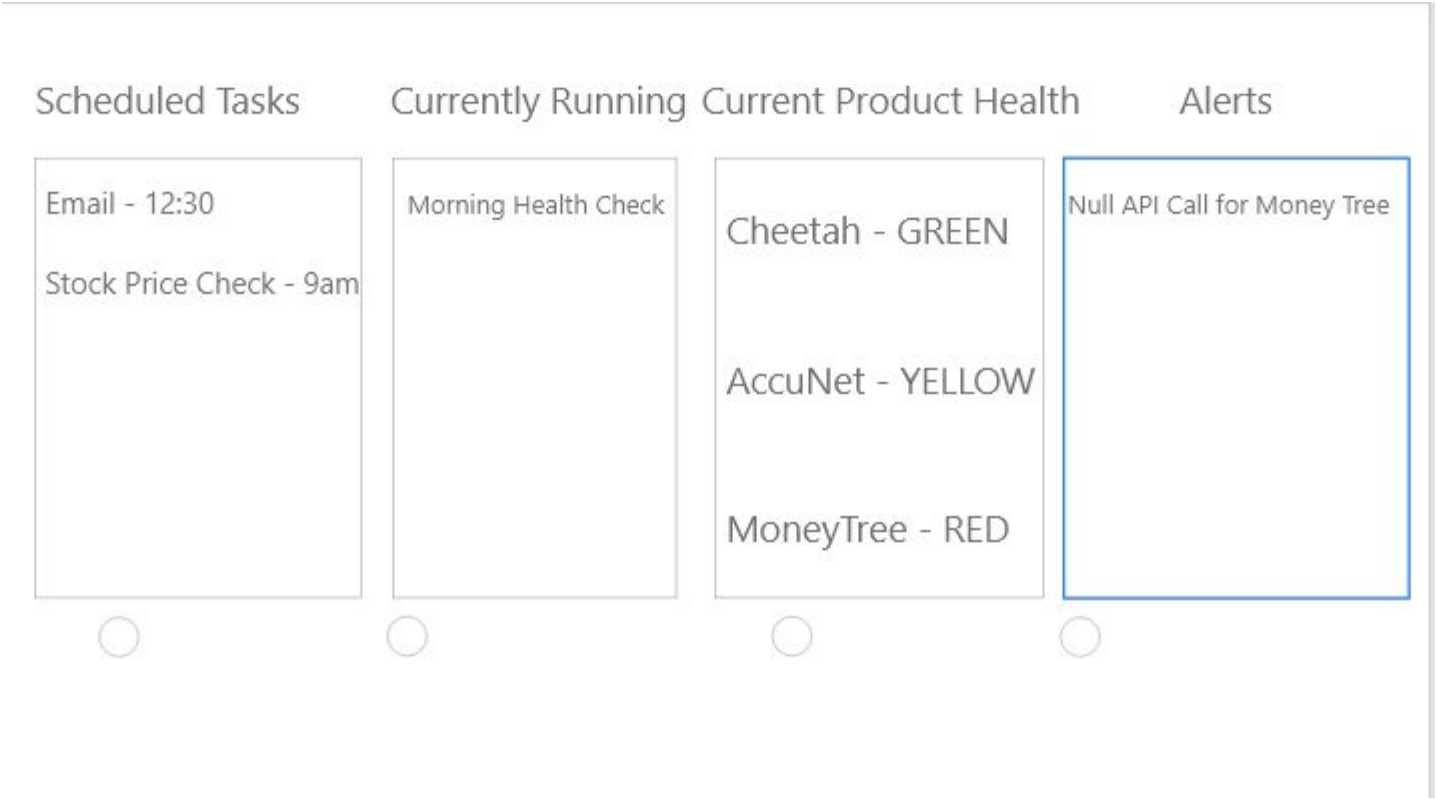
Non-Functional Requirements

- NR1 - Use C#.NET - HIGH - BR3
- NR2 - Use Twillio and Mailgun for notifications - HIGH - BR1
- NR3 - Will be hosted with AWS - HIGH - BR2
- NR4 - 24/7 Uptime - HIGH - BR2
- NR5 - History Page - MEDIUM - BR3

TechStack

- Language: C# and .NET
- Database: Postgres , EF CORE
- Hosting: AWS
- Services: Twillio and Mailgun

Prototype



Scheduled Tasks

Email - 12:30

Stock Price Check - 9am

○

Currently Running

Morning Health Check

○

Current Product Health

Cheetah - GREEN

AccuNet - YELLOW

MoneyTree - RED

○

Alerts

Null API Call for Money Tree

○

First Iteration Features

- A .NET built webpage

- The capability to call and utilize APIs and Data

- Notifications on a scheduled basis with acknowledgement to disable repeats

Client Feedback

- A command console is in other terms a web dashboard (was initially interpreted as a command terminal)

- Customizability is the greatest focus

- Designing the database to prepare for external clients to manage the monitoring software is appreciated and great future thinking

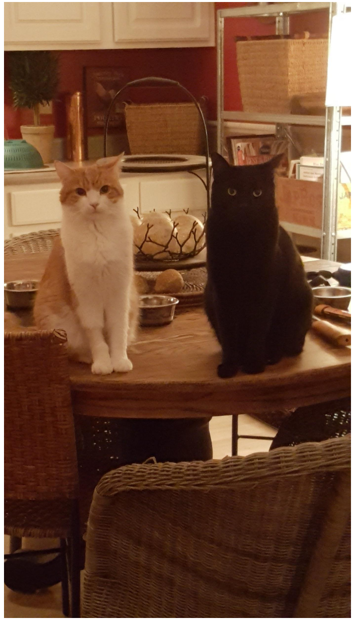
Mentor Feedback

- Teamwork makes the dream work, constant communication helps a lot (We helped mitigate this issue by switching more over to Discord which is easier for quick communication and voice chats as opposed to making Zoom rooms every meeting)
- Gave lots of feedback regarding knowledge to C# and how to best start a project foundationally and useful libraries.
- Lots of inspiration for how to tackle difficult parts of the project such as a game inspired customization system for making alerts/specific status checks for the products.

Fun Slide - Quarantine Hobbies

Luke - Home Cooking

Chase - Cats



Nathaniel - Dogs

Hunter - Plants