[**Proposal link**](https://revaturetech-my.sharepoint.com/:w:/g/personal/kyle_hill_revature_net/ERSvHvwJeKNOg3khJJMXxQIBpERPEQAvAl4SORGkB22Ymg?e=o0pMbm)

**Team:** CafeAPI - Kyle Hill (Team Leader), Gulomjon Saidov, Joshua Posada, Christie Thammavong

**Kanban Board:** We will be using the Kanban service provided by Github. This can be found under the “Projects” tab in our repo.

**GitHub Repo:** [P2\_CafeAPI Repo](https://github.com/08162021-dotnet-uta/P2_CafeAPI)

**Section 1 - Sales Description**

We will create an online shop where users can choose from a large assortment of items. The functionality we will add to the skeleton provided by Rainforest API includes customers creating an account, making purchases with the account, and viewing order history. Customers will also be able to contribute content to the site. For instance, they can leave product reviews, product ratings, and submit product subscriptions. A consequence of customer-generated content is that an administrator must monitor it for quality assurance. Because our product listings will be so bare (see API description), we will incentivize users to write professional descriptions by giving them vouchers to use on our site.

**Section 2 - Minimum Viable Product (MVP)**

* Users can sign-in to the website
* Users can look up products by category
* Users can filter products by price and ratings
* Users can purchase products

**Section 3 - 3rd-party API**

Rainforest API – [Categories API](https://www.rainforestapi.com/docs/categories-api/overview) and [Product Data API](https://www.rainforestapi.com/docs/product-data-api/overview)

All of the inventory, inventory categories, and product search capabilities will be provided by Rainforest API. The Product Data API will be used to provide lists of items for the customers to choose from. The Categories API is for identifying the subcategories within the more general categories, e.g., refrigerators and microwaves are subcategories of kitchen appliances. We will not be calling the endpoint which provides the details for individual products because the response time is too long in our opinion (10s). A consequence of this decision is that the only product information from the API will be an image link, name, and price.

**Section 4 - User Stories**

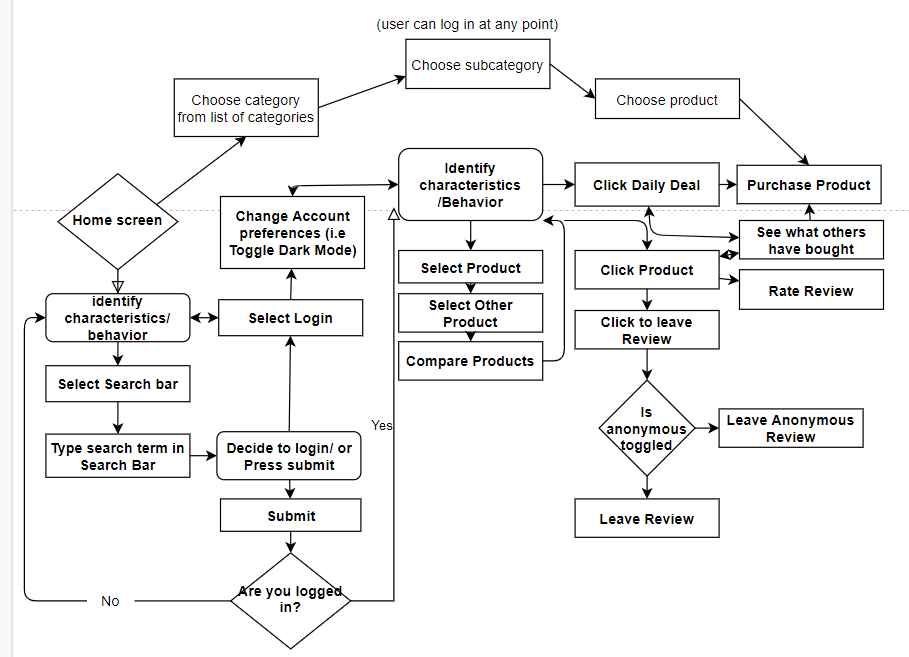
1. Customers will be able to search for and purchase products. They can search with the search bar or by using the category tree.
2. Users will be able to switch from light mode to dark mode.
3. Customers will be able to submit product descriptions and be rewarded with vouchers if an administrator approves their submission.
4. Customers will be able to login and utilize a dashboard that shows voucher balance, order history, and their personal information.
5. Customers will be able to leave anonymous reviews.
6. Customers will be able to see daily deals on front page.
7. Sign in as an administrator and be able to evaluate description submissions by users.

**Section 5 - Stretch Goals**

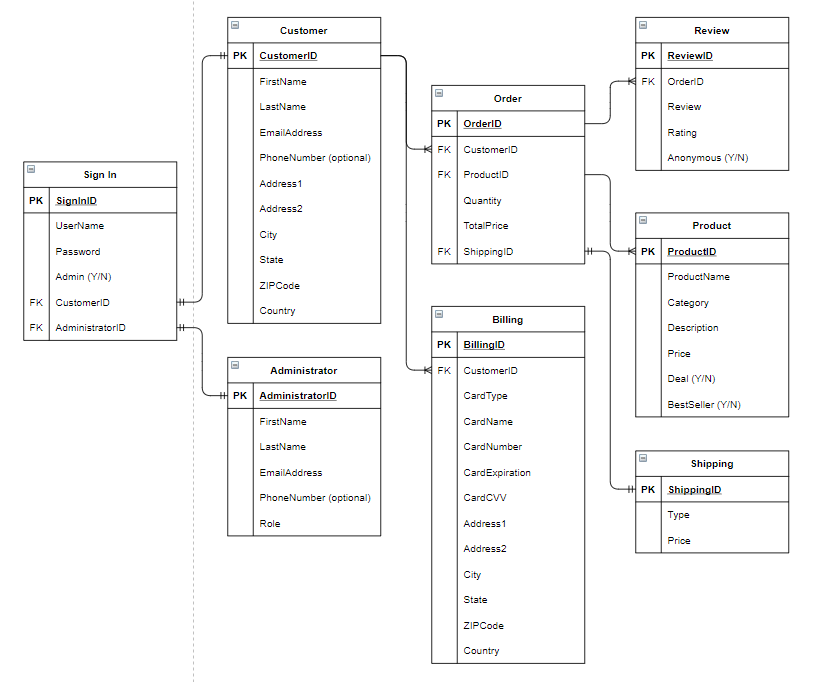
1. Reward system for providing acceptable description and good reviews.
2. Users to be rated based on their reviews by other users.
3. Compare two products side by side based on their specifications.
4. Customer returns an item and it shows up in their order history.
5. Upload images and videos to accompany reviews and descriptions.
6. Give user the option of not seeing sponsored items when searching.

Both model diagrams can be found [here.](https://drive.google.com/file/d/1NWsxd9Xcfu3xzU_qMH74Wam16WZKUws2/view?usp=sharing) Click on “Open with diagrams.net” at the top of the page to open. The two diagrams are in separate tabs which can be chosen at the bottom of the page.

**Section 6 - User Interaction Model**



**Section 7 - Database Model**



**CI/CD pipeline**

We will use Azure Pipelines to automatically build and test the project code. The pipeline will combine continuous integration (CI) and continuous delivery (CD). This will ensure us to automate merging and testing code, catching bugs in the development cycle, and deploying the code. While we set up our Azure Pipeline to configure the CI/CD pipeline, we will use a YAML file at the root of the project source code, and it will be triggered by the [project repo](https://github.com/08162021-dotnet-uta/P2_CafeAPI) created in the Batch organization repo. This will help us to control the build, test, and deploy stages of the application.