**Vendi**

***A vending machine locator app***

**Project Part 3: Design**

Computer Science, University of Nevada Reno

Team 23

Members: Catherine Wedin, Maxwell Synard, Charles Dunn, Joey Paschke

Instructors: Sergiu Dascalu, Devrin Lee, Vinh Le, Zach Estreito

External Advisor: Clayton Synard

November 15th, 2022

## **Table of contents**

[**2. Abstract**](#_xjwx9mg8rohl) **2**

[**3. Introduction**](#_r0wa2y4o7wsl) **3**

[**4. High-level and medium-level design**](#_a5klz1i1fso) **4**

[**5. Detailed design**](#_11pugh17d17h) **9**

[**6. Initial hardware design**](#_43r09yxt1a0x) **12**

[**7. User interface design**](#_5vuv9ikjrkdz) **13**

[**8. Version control and software management system**](#_isqlbsmxfrm) **23**

[**9. Contributions of team members**](#_7huvlcqt4osn) **23**

## **Abstract**

The focus of the project will be the development of an app that allows for the location of vending machines developed for mobile phones. The base functionality of the app will include: locating nearby vending machines on a map, allowing users to filter between the different kinds of vending machines, and implementing a way to gain data on the vending machines, the current idea being crowdsourcing. The crowdsourcing will be incentivized by implementing a points system that allows rewards to be redeemed by users. This project is worthwhile because vending machines are actively used as quick access to food and drink in many populated places around the world. Through our research, we have yet to find a novel solution to locating them.

## Introduction

The goal of the project is the design and development of a mobile application called “Vendi.” This app will allow users to locate, filter, and participate in the crowdsourcing of data for nearby vending machines. The app will be able to function on touchscreen devices. When it is accessed the users will be greeted with a login screen, and will also be able to continue as a guest. Once in the app, they will be able to see the map with all vending machine locations marked on the map. Clicking on the machines will give information about the machine. On top of this, there will be a sidebar with options including filtering, point redemption, favorite machines, and feedback. Before the deployment of the app sometime next year, we wish to implement a handful of different features to the app beyond our base functionality. This will include a login system, a handful of different data features for the machines beyond just locations and basic filtering, and image analysis, though adding all of these features might be a bit beyond what is likely to be done by next year.

Since the last project assignment, all progress has come in the form of adding new data types to tie to the machines and a better understanding of the design and structure of the data for the project. Through discussion and working on the project assignments, we have a better idea of how the UI for the assignment is going to look, a color scheme for our project, and some of the software technology we are going to be using to help build the app. On top of this, we’ve also created a GitHub repository and added documentation for the project to the repository.

Lastly, we have not made any major changes to project requirements since the last assignment. The interviews and design have caused us to rethink a few more minor features such as a guest system, ideas for expanding the machines that may be in the app, or even adding a chat system to the app, however, all of these ideas go beyond the scope of the basic functionality we would want in the app. However, a bigger focus on simplifying the crowdsourcing to a simple click might be made since it would make people more likely to do it even with a smaller incentive.

## High-level and medium-level design

The Vendi System is a non-oriented solutions project. The context diagram in Fig. 4.1 displays the high level components of the system and how the Vendi system connects to four other systems. The Security System is implemented to ensure the protection and encryption of the user’s data and credentials. The Renderer System displays the map, vending machine, and all related components to that system, the Machine System stores all data pertaining to the vending machine itself and the Account System stores all data pertaining to the user account upon creation or updates.

Figure 4.1: The context model of the Vendi system shows how it will interact with various subsystems to complete its functionalities.

****

Table 4.1: This table lists the programs units under the Machine Data Entry System

| Machine Data Entry | A node module that contains all of the methods from user and vendor input. |
| --- | --- |
| addItemName(itemID, col) | This method takes object named values as input from the user or vendor and adds it to the database. It will add the newly created item into the column. |
| addPrice(itemID, col) | This method takes a numeric value as an input and will add it to the price column in the database aligned with the item name. |
| addInStock(itemID, col) | This method takes a boolean true or false value where the user will mark it as currently available or unavailable in the system. |
| addPhoto(itemID, col) | This optional method allows the user to take a photo of the vending machine. |
| addReview(itemID, col) | This method is a string input allowing the user to enter a review on the vending machine based on item variety, location, pricing, quality of the physical vending machine and availability of items. |
| addRating(itemID, col) | This method takes a numeric input from a range of 1-5 in which the user will rate the overall quality of the machine based on various metrics. |
| addFlag(itemID, col) | This method gives the option to flag an item if applicable if it is not accurately listed in the machine. |

Table 4.2: This table lists the programs units under the Map Renderer System

| Map Renderer | A node module that contains methods that will render information. |
| --- | --- |
| viewMap() | This method will render all available vending machines with a given area. |
| viewMachineDetails(machineID) | This method will show all available items listed in the vending machine as well as if the machine takes cash, card, or both. If available, it will also show an image of the vending machine. Within this, the reviews and overall average rating will be displayed. |
| viewItem(itemID) | The method will allow the user to select a specific item and view all details pertaining to it such as the price, optional description, and if it is currently in stock or not. |

Table 4.3: This table lists the programs units under the Account System

| Account | A node module that contains methods that store information pertaining to the user’s profile in the database. |
| --- | --- |
| createUsername (user) | This method takes a string input from the user and if the username is not taken, it will store it in the database. |
| createPassword(password) | This method will take string input from the user to create a password to store the username on the account. If the password passes the password test, it will be saved. |
| changeUsername(user) | The method takes user input and allows the user to change their username and if the new username is not in use, it will return true and be stored as the current username. |
| changePassword | The method takes user input and allows the user to change their password and if the new password is not equal to the current password and passes all password requirements, it will return true and be stored as the current password. |
| checkUsername(user) | This method takes the user input and checks to make sure the username is already not in use. If it is already in use, the username will be denied and the user will be prompted to enter another username. |
| checkPassword(password) | This method takes the user input from the password and makes sure it passes all password requirements to ensure the user has used a strong password to decrease the chances of hacking. |

Table 4.4: This table lists the program units under the Security System that will ensure the protection of the user’s data.

| Security | The security system restricts access to data for unauthorized users. It helps prevent user accounts from being breached, stops user data from being leaked, and prevents breaches of data that can’t be accessed from the front-end. |
| --- | --- |
| checkCredentials(password) | Once an account for the requested email/username has been identified the password is checked against the encrypted password on hand for the account. |
| verifyPermissions() | Begins another login process whenever sensitive information is attempting to be accessed such as personal data or point redemption on a user account. Once the login has been submitted, checkCredentials will be called to confirm this login. |
| encryptData(data) | Encrypts any data passed into the method. This encrypted information is returned. It will be used to encrypt user information when tied to machine images posted to the app, for user location security. |
| hashPassword(password) | On either creation of an account or during verification of credentials, the given password will be hashed and either compared to the information stored in the database or associated with the account in the database. |

Table 4.5: The User table stores all the user login credentials into the database as well as information pertaining to the account/profile that will be used for account verification purposes.

| User |  |
| --- | --- |
| userName<string> | The username of the user that is stored in the database. |
| userID<int> | Unique user ID to distinguish between users. |
| userEmail<string> | The email that the user used to sign up for the app. This will be stored in the database for user login. |
| userPassword<string> | A password tied to an account. It is stored as a hashed password for user security. |
| userLocation<int> | Encrypted location data on the user. Without encryption it would be read as the user's geographical coordinates. |
| favoritesList<object array> | A list of the user’s favorite vending machines. Can be interacted with in the app. |

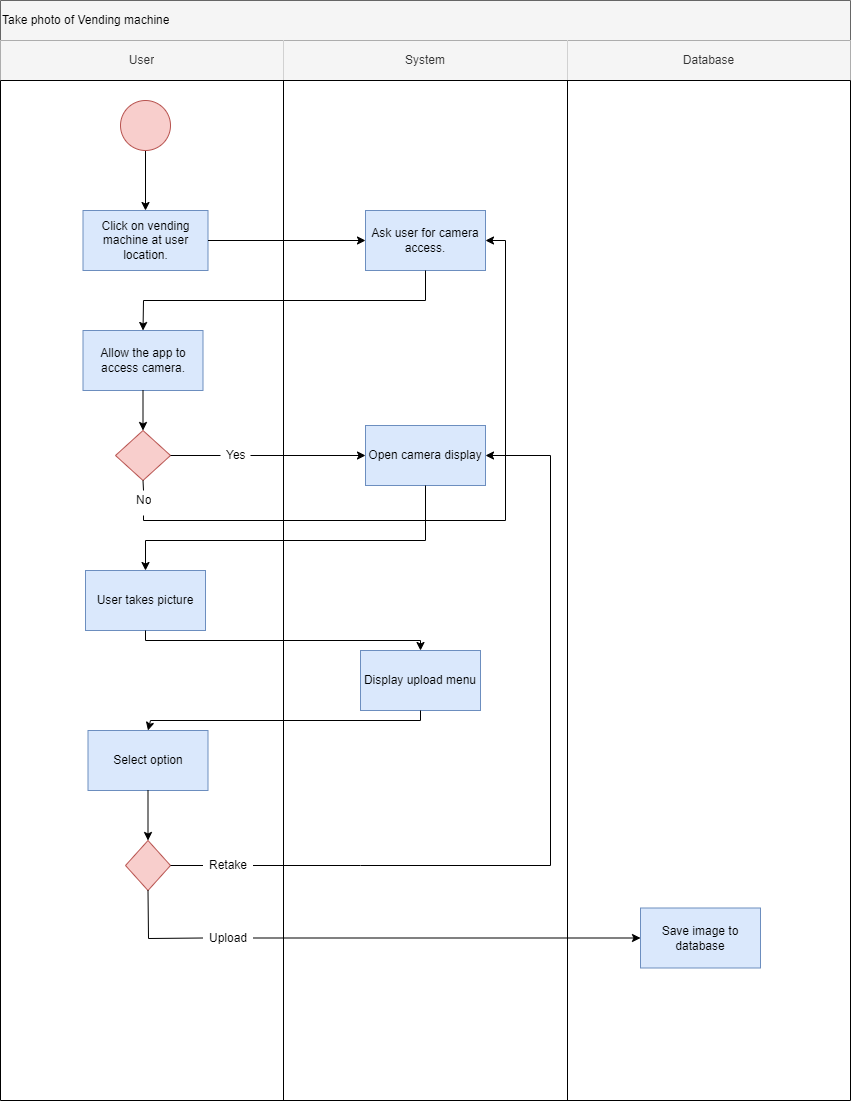
Table 4.6: The Vendor table stores all the vendor login credentials into the database for the back end as well as information pertaining to the account/profile that will be used for account verification purposes.

| Vendor |  |
| --- | --- |
| vendorName<string> | The username associated with the vendor’s account in the database. |
| vendorID<int> | Unique vendor ID to distinguish between similar vendors. |
| vendorEmail<string> | The email that the vendor used to create his vendor account. This is stored in the database for vendor login. |
| vendorPassword<string> | A password tied to a vendor account. It is stored as a hashed password for security purposes. |
| machineList<object array> | A list of the vending machines verified to be owned by the vendor. It allows updating the official details of the machines in the app. |

Table 4.7**:** The Machines table stores all the data pertaining to a given vending machine in which is identified by the machineID number.

| Machines |  |
| --- | --- |
| **machineID<int>** | The primary key and unique key used to differentiate each vending machine |
| itemID<int> | A unique key assigned to each individual item within a vending machine. |
| itemName<string> | The name of the item. |
| price<double> | The price of the item given in dollars and cents. |
| quantity<int> | The total amount of the item in stock. |
| inStock<bool> | Displays a yes or no if the item is currently in stock. |
| machineName<string> | Displays the name of the machine |
| location<string> | Displays the location of machine |

## **Detailed design**

Figure 5.1: Activity diagram of the process taken when the user wants to log a picture of a vending machine.

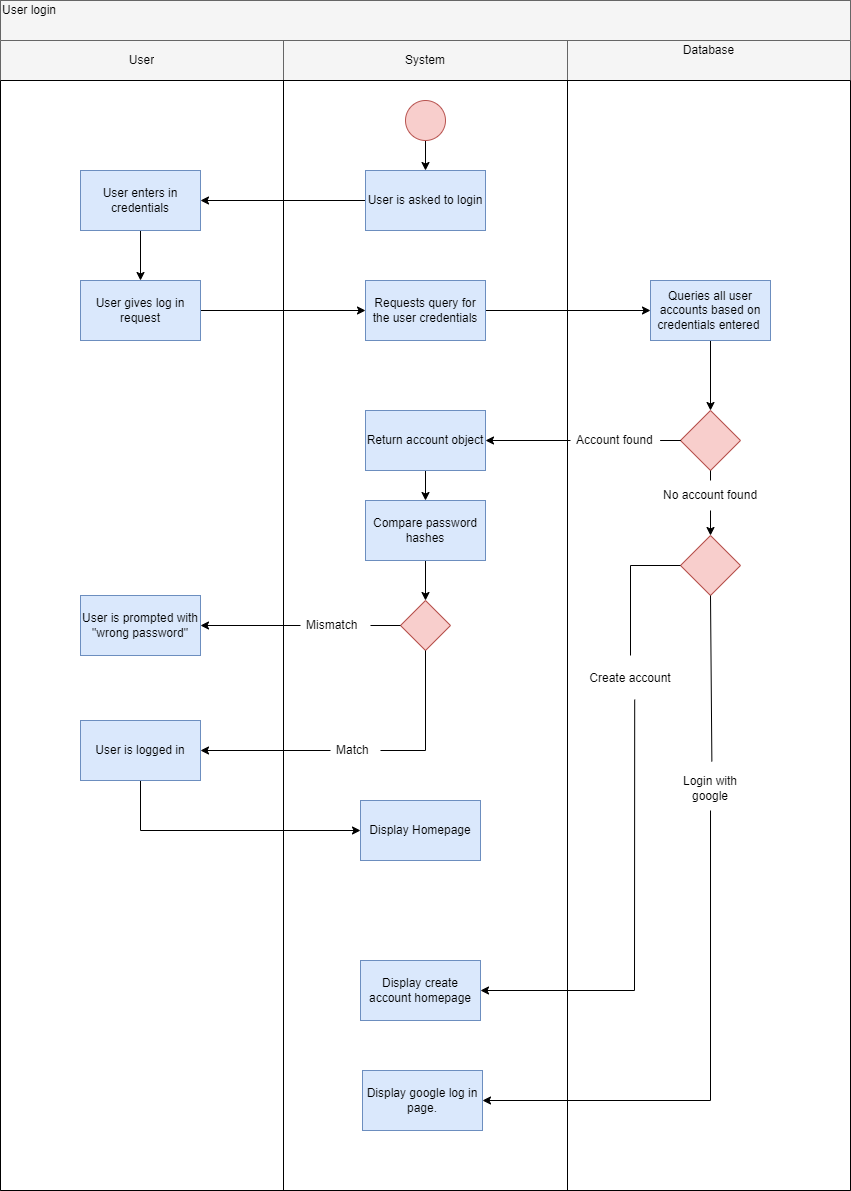
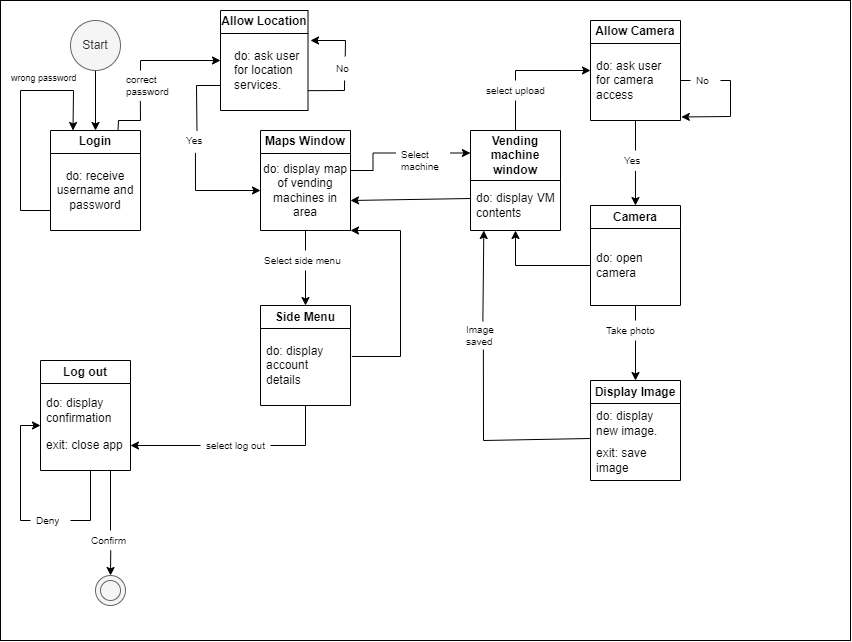


Figure 5.2: Above, user login requirements are shown with an activity diagram.

Figure 5.3: State diagram outlining the different states the user will encounter when using the app.

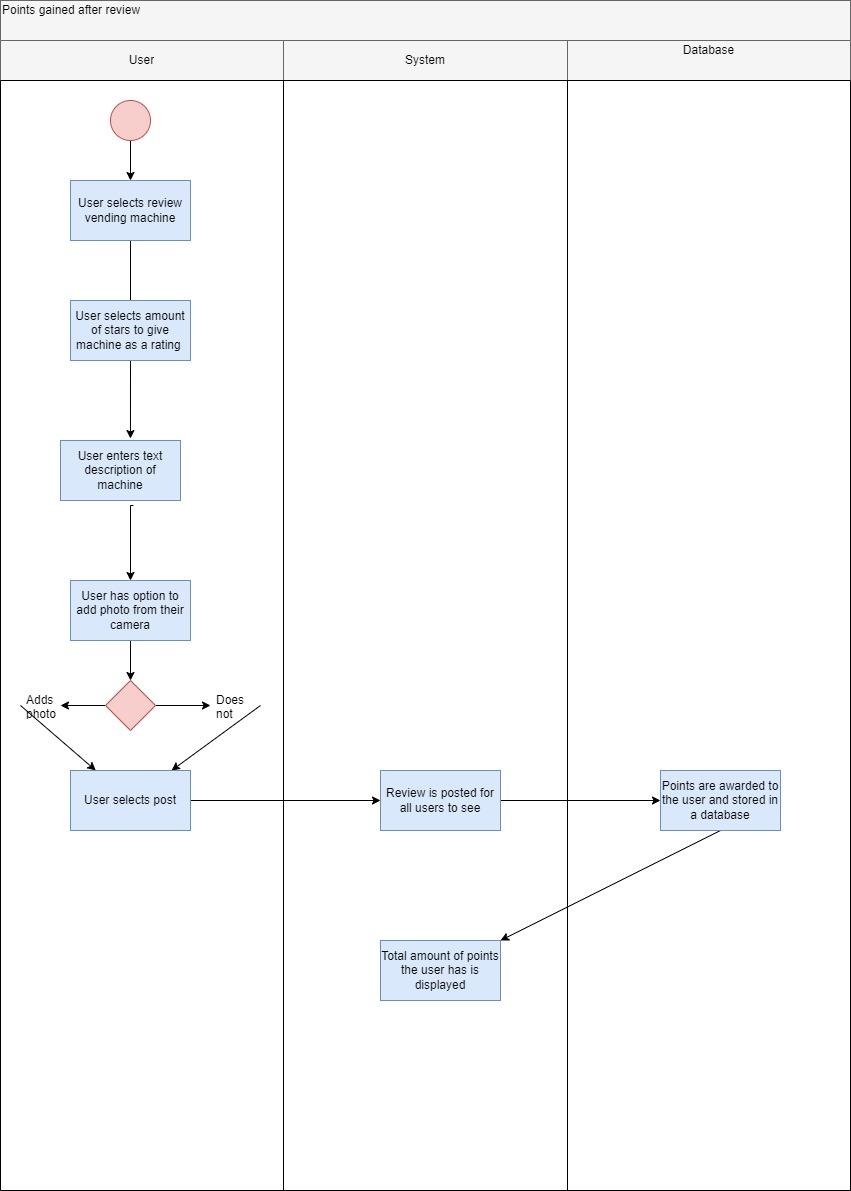


Figure 5.4: Activity diagram of the process taken when the user writes a review and is rewarded points.

## Initial hardware design

* + Our product does not require hardware components. All that is required is a smartphone and an internet connection.

## User interface design

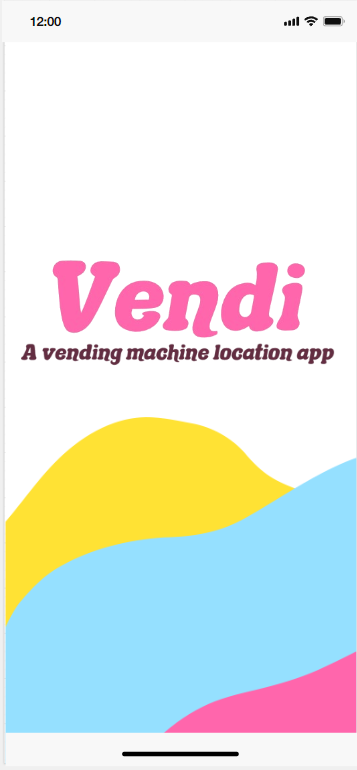


Figure 7.1: A mockup of the loading screen when the user first opens the app.

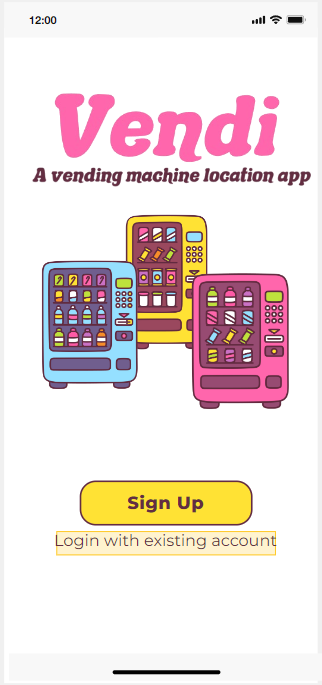


Figure 7.2: A mockup of the first screen that will appear after the loading screen. Here the user can either click the link to take them to the sign-up page or the login page.

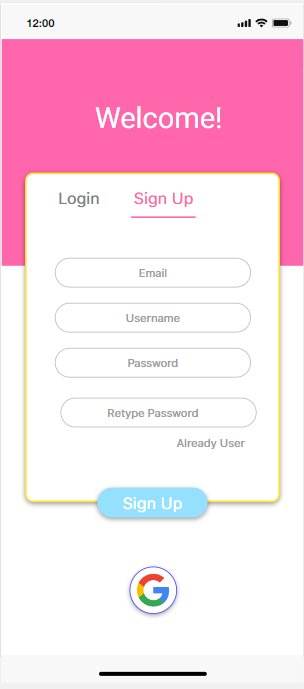


Figure 7.3: A mockup of the screen that has the user enter the information needed to create an account. There is also an option to sign in with a Google account with our mockup.

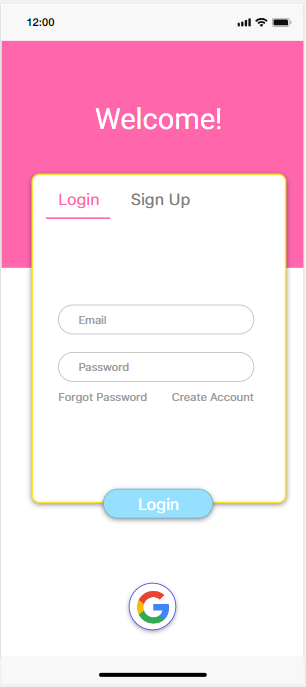


Figure 7.4: A mockup of the page that allows the user to log in with a pre-existing account. From here they can also navigate to the “forgot password” or back to the “create account” page.

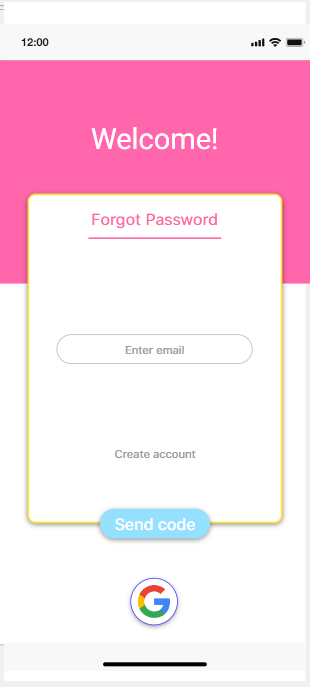


Figure 7.5: A mockup of the forgot password page. The user will enter the email address associated to their account and an email with a link to update their password will be sent.

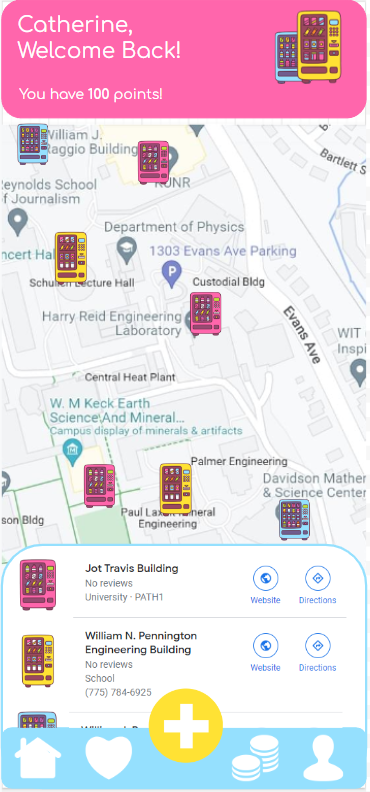


Figure 7.6: A mockup of the page you will see upon sign-in. It will provide you with your name, current number of points, and the closest vending machines near you.

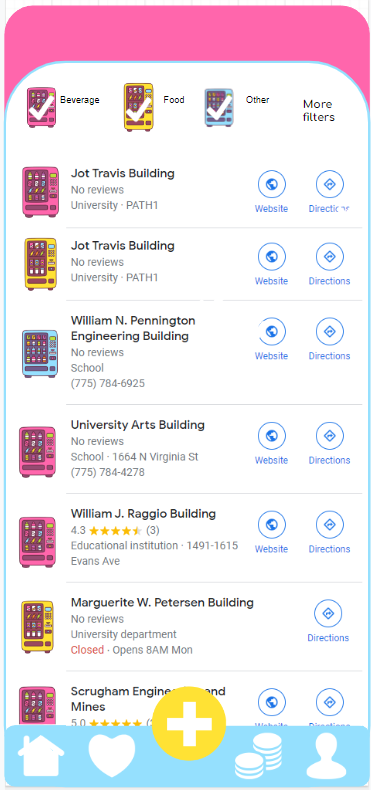


Figure 7.7: A mockup of how the user can scroll through a list view of the machines near them, along with a way to filter the machines. Readily available are filters for machine types, but the user can select “more filters” for more options in their search.

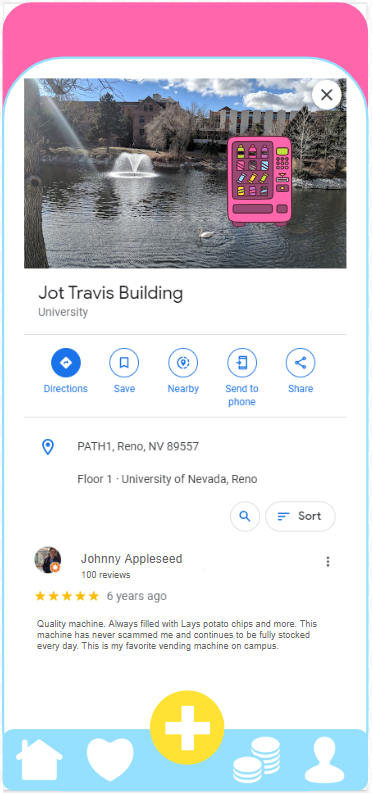


Figure 7.8: A mockup of how it will look when a certain machine is selected. The ratings and reviews for the machine will show, along with any other information such as what floor the machine is located on.

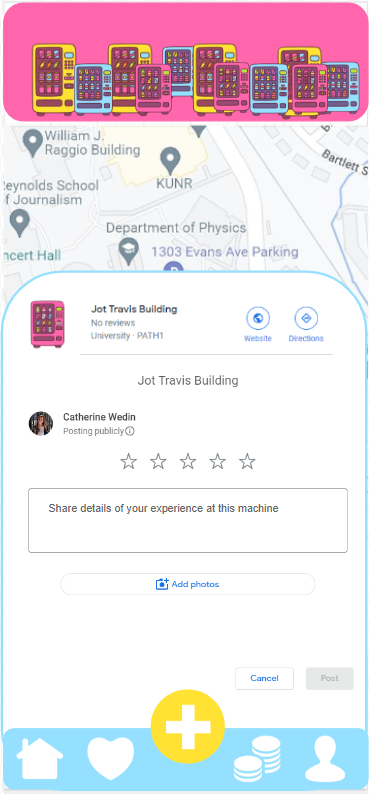


Figure 7.9: A mockup of the screen that the user will see if they want to write their own review. There will be options to rate the machines with stars, add text, and add photos.

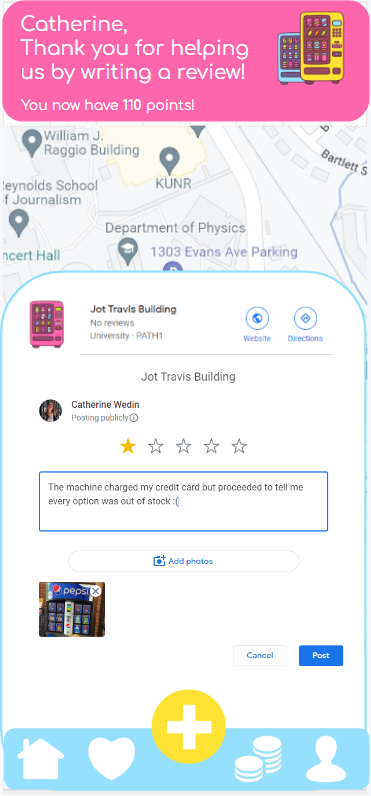


Figure 7.10: A mockup of the page when a user posts a new review. As you can see, a rating was given along with a description and photo. The user is praised with pointed added to their account and a thank you message.

## **Version control and software management system**

[Vendi Github](https://github.com/DuckieOnQuacks/Vendi)

## Contributions of team members

**Catherine Wedin: 6 hours**

Made the table of contents, formatted the entire design document, created one activity diagram, and created all the user interface design mockups.

**Joey Paschke: 5.5 hours**

Made two activity diagrams and one state diagram. Created the GitHub repository and added team members as collaborators.

**Charles Dunn: 5 hours**

Created six non object-oriented solution tables with descriptions and procedures and one system-level diagram for section 4 high-level and medium-level design. Proofread document and made grammar revisions.

**Maxwell Synard: 5 hours**

Wrote the abstract and the introduction. Also added the data structures for users and vendors on top of doing the table for the security system. Lastly, added documentation and updated the readme file in the github.