Iteration Five

Logan Coleman, Hannah Sadikovic, Matthew Roth, Faysal Bulbul, Chris Humphrey

Table of Contents

1. Class Diagrams 3

2. User Interface Navigation Diagram 5

3. Database Design and Data Definitions 6

4. User Interface Prototypes 14

5. Gantt Chart 27

Class Diagram

 A class diagram is a representation of the use cases and describes the structure of a system by showing the use cases classes, attributes, methods, and their relationship among objects. It shows a developer how the system should work and run smoothly. With that being said, the attributes and methods must be specified exactly how the designer would want the system to look like and behave. For the class diagram, we used a method called verb-noun analysis where we named the class and put that name in the first row of the class in the form of a noun from the appropriate use case for the class being made. We then identified all the nouns and listed them in the second row of the class as attributes. Finally, we identified all the verbs and listed them in the third column as methods. After we had all our classes, we had to find relationships between them. There are two main classes that all the sub classes go back to. They are the Login class and the Load site class. An arrow is connected from all sub classes to either the Login Class or the Load Site Class. The Login class is a prerequisite for all sub classes that people who deal with the backend of the website, staff members. The Load site class is a prerequisite for all sub classes that deal with the front-end of the website, guests as well as staff members.

A picture containing timeline

Description automatically generated

Windows Navigation Diagram

The windows navigation diagram shows the basic structure of the website. It starts with the broadest part, the home page, and becomes more specific from there. The diagram shows how the entire website is connected to itself.

Diagram, engineering drawing

Description automatically generated

Database Design and Data Definitions

The following diagram is a relational database diagram designed in 3NF (third normal form). Below that are the data descriptions about the attributes of each class, displayed in table format.

Timeline

Description automatically generated

**Scheduling table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Attribute Definition** | **Attribute Size** | **Attribute Type** | **Primary/Foreign Key** |
| **AmountHours** | The amount of hours that were volunteered. | Max 40 bytes | string | **Primary Key** |
|  |  |  |  |  |
|  |  |  |  |  |
| **Name** | The name of the volunteer who performed the service | Max 40 bytes | string | N/A |
| **Date** | The date the service was performed | Max 40 bytes | string | N/A |

**Login Table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Attribute Definition** | **Attribute Size** | **Attribute Type** | **Primary/Foreign Key** |
| **UserID** | A username that uniquely identifies a student or staff member | Max 40 bytes | string | **Primary Key** |
| **Password** | A password that uniquely identifies a student or staff member | Max 40 bytes | string | N/A |

**Drop off Table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Attribute Description** | **Attribute Size** | **Attribute Type** | **Primary/Foreign Key** |
| **PhoneNumber** | Phone number of a guest | 11 bytes | Long unsigned int | **Primary Key** |
| **Email** | Email address of a guest | Max 40 bytes | string | N/A |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| **Items** | Food items being dropped off | Max 255 bytes | string | N/A |
| **FirstName** | First name of guest | 40 bytes | string | N/A |
| **LastName** | Last name of guest | 40 bytes | string | N/A |

**RSO Table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Attribute Description** | **Attribute Size** | **Attribute Type** | **Primary/Foreign Key** |
| **PhoneNumber** | Phone number of RSO contact | Max 11 bytes | Long unsigned int | **Primary Key** |
| **RSOName** | Name of RSO | Max 40 bytes | string | N/A |
| **EventName** | Name of an RSO event | Max 40 bytes | string | N/A |
| **Email** | Email of Head of RSO event | Max 40 bytes | string | N/A |

**Subscription Table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Attribute Description** | **Attribute Size** | **Attribute Type** | **Primary/Foreign Key** |
| **Email** | Email of the Guest User that is subscribing to the mailing list | Max 255 bytes | string | **Primary Key** |
| **FirstName** | First name of user | Max 40 bytes | string | **N/A** |
| **LastName** | Last name of user | Max 40 bytes | string | **N/A** |

**Inventory Table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Attribute Description** | **Attribute Size** | **Attribute Type** | **Primary/Foreign Key** |
| **FoodName** | Description of type of food item | Max 40 bytes | string | **Primary Key** |
| **FoodItems** | Food items the pantry currently has in stock | Max 255 bytes | string | **N/A** |
| **NonFoodItems** | Non-Food Items that the pantry has in stock | Max 255 bytes | string | N/ A |
| **Quantity** | Quantity of inventory item | Max 255 bytes | string | N/A |
| **ExpirationDate** | Expiration date of food items | Max 20 bytes | string | N/A |

**Itemized List Table:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attribute** | **Attribute Description** | **Attribute Size** | **Attribute Type** | **Primary/Foreign Key** | |
| **FoodName** | Description of type of food item | Max 40 bytes | string | **Foreign Key** |
| **NeededFoodItem** | Food item Cardinal Cupboard needs | Max 255 bytes | string | **N/A** | |
| **NeededNonFoodItem** | Non-food item Cardinal Cupboard needs | Max 255 bytes | string | N/A |

**Applications table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Attribute Definition | Attribute Size | Attribute Type | Primary or Foreign Key? |
| UserID | Username of staff Member | **Max 10 bytes** | string | Primary Key |
| Password | Password of staff member | **Max 10 bytes** | string | N/A |
| Applications | The applications that staff can add, modify, or delete. | **Max 1MB** | string | N/A |

**Forms table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Attribute Definition | Attribute Size | Attribute Type | Primary or Foreign Key? |
| FormID | The identifier for the form being submitted. | **Max 20 bytes** | Long unsigned int | **Primary Key** |
| RecipeForm | The recipe form users can submit. | **Max 100 bytes** | string | **N/A** |
| ApplicationForm | The volunteer application form users can submit. | **Max 100 bytes** | string | N/A |
| FirstName | First name of user. | **Max 40 bytes** | string | N/A |
| LastName | Last name of user. | **Max 40 bytes** | string | N/A |
| Phone Number | Phone number of user. | **Max 11 bytes** | Long unsigned int | Foreign Key |
| StudentID | Student ID number | **Max 10 bytes** | Long unsigned int | Foreign Key |

**Recipe table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Attribute Definition | Attribute Size | Attribute Type | Primary or Foreign Key? |
| RecipeID | The Recipe Name | **Max 40 bytes** | String | Primary Key |
| Inventory | The inventory that both guest users and staff can view. | **Max 10GB** | String | N/A |
| Recipes | The recipes that staff can add, modify, or delete. | **Max 5 GB** | String | N/A |

**Contact Table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Attribute Definition** | **Attribute Size** | **Attribute Type** | **Primary or Foreign Key?** |
| **PhoneNumber** | Student’s phone number | **Max 11 bytes** | long unsigned Int | **Primary Key** |
| **FirstName** | Student’s first name | **Max 40 bytes** | String | N/A |
| **LastName** | Student’s last name | **Max 40 bytes** | String | N/A |
| **Email** | Student’s email | **Max 40 bytes** | String | N/A |

**Amazon Wishlist Table:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Attribute** | **Attribute Definition** | | **Attribute Size** | | **Attribute Type** | | **Primary or Foreign Key?** | |
| **AmazonWishlist** | Cardinal Cupboard’s wishlist with items people can donate from. | | **Max 255 bytes** | | string | | **Primary Key** | |
| **ItemSelected** | Items selected from the wishlist | **Max 40 bytes** | | string | | N/A | |
| **Quantity** | Number of items on wishlist | **Max 255 bytes** | | Long unsigned int | | N/A | |
| **Item** | Description of item on the wishlist | **Max 255 bytes** | | string | | N/A | |

**ItemDonation Table:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attribute** | **Attribute Definition** | **Attribute Size** | **Attribute Type** | **Primary or Foreign Key?** | |
| **Email** | Contact Information for the guest donating. | **Max 40 bytes** | String | Primary Key | |
| **FoodDonation** | Food that is donated to the cupboard. | **Max 255 bytes** | String | N/A |
| **NonFoodDonation** | Items that are not food donated to the cupboard | **Max 255 bytes** | String | N/A |
| **PerishableID** | Determines if a FoodDonation is perishable or non-perishable. If perishable, this is used to find an expiration date | **2 bytes** | Boolean | N/A |
| **ExpirationDate** | The date a food item expires | **Max 16 bytes** | String | N/A |

**MonetaryDonation Table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Attribute Definition** | **Attribute Size** | **Attribute Type** | **Primary or Foreign Key?** |
| **MonetaryDonation** | Money donated to the cupboard. | **Max 40 MB** | Long Unsigned Int | N/A |
| **Email** | Contact Information for the guest donating. | **Max 40 bytes** | String | Primary Key |

**Dynamic Calendar Table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Attribute Definition** | **Attribute Size** | **Attribute Type** | **Primary or Foreign Key?** |
| **Event Name** | Name of an event | **Max 255 bytes** | String | N/A |
| **Event Description** | Describes the event | **Max 255 bytes** | String | N/A |
| **Event Time** | Time event starts and ends | **8 bytes** | int | N/A |
| **Event Date** | Day(s) event is on | **Max 255 bytes** | String | N/A |

**Drop Box Table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Attribute Definition** | **Attribute Size** | **Attribute Type** | **Primary or Foreign Key?** |
| **Student ID** | The ID U of L gives to each student. | **7 bytes** | Long unsigned Int | **Primary Key** |
| **Student Email** | Student’s email | **Max 255 bytes** | String | N/A |
| **ItemsSelected** | The items a student can select to pick up | **Max 255 bytes** | string | N/A |

**Compost Table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Attribute Definition** | **Attribute Size** | **Attribute Type** | **Primary or Foreign Key?** |
| **UserID** | The ID of staff Member | 7 bytes | Long unsigned Int | **Primary Key** |
| **CompostQuantity** | The amount of compost created | Max 20 bytes | Long unsigned int | **N/A** |

User Interface Prototypes

The following images are basic html prototypes of what the system must look like in order to complete each use case.

1. Submit Recipe

*In order to submit a recipe, the user must navigate the landing page and choose “Resources”, then “Recipes.”*

1. View Recipe

*In order to view a recipe, the user must login through the staff login, where they will be directed to the member page. The user will choose “Recipes” and will be directed to a page with the submitted recipes, where the user can view the recipe.*

1. Add Recipe

*In order to add a recipe, the user must login through the staff login, where they will be directed to the member page. The user will choose “Recipes” and will be directed to a page with the submitted recipes, where the user can add the recipe to a list of “Accepted Recipes.”*

1. Modify Recipe

*In order to modify a recipe, the user must login through the staff login, where they will be directed to the member page. The user will choose “Recipes” and will be directed to a page with the submitted recipes, where the user can modify the recipe.*

1. Delete Recipe

*In order to delete a recipe, the user must login through the staff login, where they will be directed to the member page. The user will choose “Recipes” and will be directed to a page with the submitted recipes, where the user can delete the recipe.*

1. View Application

*In order to view an application, the user must login through the staff login, where they will be directed to the member page. The user will use “Applications” and will be directed to a page with the submitted applications, where the user can view the application.*

1. Add Application

*In order to add an application, the user must login through the staff login, where they will be directed to the member page. The user will use “Applications” and will be directed to a page with the submitted applications, where the user can add the application to a list of “Accepted Applications.”*

1. Delete Application

*In order to delete an application, the user must login through the staff login, where they will be directed to the member page. The user will use “Applications” and will be directed to a page with the submitted applications, where the user can delete the application.*

*Landing Page*

Table

Description automatically generated

*(1.)*

*Member Login Page*

A picture containing table

Description automatically generated

*Member Page*

Graphical user interface, application

Description automatically generated

*(2., 3., 4., 5., 6., 7., 8.)*

*Recipes Page*

Table

Description automatically generated

*(2., 3., 4., 5.)*

*Applications Page*

Table

Description automatically generated

*(6., 7., 8.)*

These HTML prototypes do not connect to any databases, and only reflect the minimum of programming. **Due to the backup system, and restore system use cases requiring external programs, no diagrams need to be made for them.**

This page covers the view dynamic updates, contact Cardinal Cupboard,  view additional information, and view social media use cases. **The view additional information use case is covered because it is accessed by clicking the *about*headline.** This about headline displays the additional information page.

Graphical user interface, application, table

Description automatically generated

*This page is the additional information page. It covers the view history, and view mission statement use cases as well.*

Table

Description automatically generated

*This page covers the access grab n go use case.*

Table

Description automatically generated

The first prototype below covers many of the use cases. In the prototype, you can see where you would search the website in the upper righthand corner. The sustainability emphasis will be covered in the section that is labeled “Stats on Composting and Sustainability. Days of operation is covered along with the hours of operation.

Graphical user interface, application, table

Description automatically generated

The next prototype will cover the donation page which can be accessed by clicking on the donate tab seen above. The donation page will cover the Amazon Wishlist, the types of donations, and donation emphasis. You can also see a link to monetary donation which will be covered in the next prototype.

Graphical user interface, application

Description automatically generated

The monetary donation prototype is a mockup of what a payment form would look like, which is where the monetary donation use case ends.

Graphical user interface

Description automatically generated

Table

Description automatically generated

*Communicate Drop-Off Box (Above)*

Table

Description automatically generated

*Communicate with RSO’s (Above)*

Table

Description automatically generated

*Itemized List (Above)*

Table

Description automatically generated

*Student Login (Above)*

Table

Description automatically generated

*Submit Volunteer Application (Above)*

Table

Description automatically generated

*Subscribe Mailing List (Above)*

Table

Description automatically generated

*View Dynamic Calendar (Above)*

Table

Description automatically generated

*View Inventory (Above)*

*Graphical user interface, application

Description automatically generated*

*Donation Page (Above)*