**ShareBox:**

Optimizing the Donation Process for Donor, Receivers and Donation Centers

Lead Developer: Nini Ola

**1.1 Project Overview**

ShareBox aims to create an online platform for sharing food and resources to reduce waste and combat resource scarcity. This will be accomplished by integrating a software extension into government websites, such as *michigan.gov* or *detroit.com*, enabling resource sharing. The app will feature two types of users: **Donors** and **Receivers**. Donors, which could include restaurants, homeowners, and individuals, will have the option to reduce food waste by sharing surplus food and other resources. Receivers will have access to these shared resources, fostering a more sustainable waste management system. To ensure the safety and health of all users, ShareBox will implement strict security measures. All users will be required to provide a photo ID, a verified Google account, and a phone number for accountability. Additionally, all edible donations must include an ingredients list and a date of preparation or expiration to guarantee safety and compliance with health standards. The current design of donation boxes also raises concerns about safety and accessibility, as they often fail to protect users or prevent misuse. ShareBox will address these issues by introducing innovative and secure solutions to create a safe and effective resource-sharing system.

**1.2 Project Purpose, Scope, Objective**

The purpose of this project is to develop a software extension that promotes the sharing of excess resources. The extension will seamlessly integrate with government-issued websites, ensuring accessibility for both donors and receivers. Through this platform, users can locate secure donation boxes or hubs to either donate excess items or acquire needed resources. By encouraging resource sharing, the software aims to reduce waste and address issues like food insecurity and the unequal distribution of resources.

To prioritize user safety, the app will require all users to log in using their Google account or phone number, ensuring all activities are tracked and deterring malicious actions. Donors will be provided with a list of acceptable items and conditions, along with a form to confirm the validity of their contributions. They will also need to upload photos of their items and will be given a unique code to access the designated donation box. A timer will start once the item is deposited. Receivers will log in with their phone number and gain access to a list of available donation boxes within a 50-mile radius. To prevent misuse, receivers can claim only one food box every four days and one resource box every three months. Both donors and receivers will have access to a map, reporting services, and the app will require an internet connection to function. Illegal activities will be monitored and punishable by law. Additionally, a library card will be required to use the platform.

The developer will handle all aspects of frontend, backend, and database development. The frontend will include a simple login feature, submission and receiving forms, and a confirmation page with a QR code for successful transactions. The database will maintain a record of all users, including their phone numbers, donation history, and item details, ensuring accountability and transparency throughout the system.

**1.3 Team Organization (Roles and Responsibilities)**

The developer will serve as the primary point of contact for all communications and administrative duties related to the project. Due to the unique nature of this project, the developer will independently reach out to external stakeholders, including government organizations, food banks, and local stores, to gather feedback on the app after phases 1 and 3. Through surveys and trial testing, community members will have the opportunity to provide their opinions and constructive criticism on the app's features and functionalities. This feedback will be instrumental in refining the platform to better meet the needs of users and stakeholders.

Developer responsibilities:

The developer is solely responsible for completing tasks in all entirety. The developer will focus on understanding the functionality and frameworks of the software utilized, as well as documenting their uses in the Extension. When the developer experiences technical issues, they have been informed to solve it on their own as the client (professor) and sub client (GTA) have been instructed to not give the developer any feedback by the Manager (Dean of student). Therefore issues will be resolved using google. The developer is responsible for all sections of the related documentations and will review them for grammatical errors, proper format and appropriate style.

**1.4 Project Plan (iterations, project schedule)**

Due to the absence of an official client, the developer has taken the initiative to identify potential clients by reaching out to local food banks, restaurants, and grocery stores. The developer will conduct surveys to gauge their willingness to participate and to gather insights on the features they would find valuable in the app. While the app's core features are already predetermined, the unique nature of this project allows for client feedback to be incorporated. This feedback will help enrich the app and enhance the overall user experience.

Phase 1 Deadline – Solidifying and Initial development- Monday 02/26/25

* Complete UI design of the software and create powerpoint mockups for them
* Database – design structure of database and visualize flow of data from user to database
* Server: Research different options to Integrate the software extension seamlessly into the already functioning government websites
* Solidify features and functionality

First Client input– Friday 02/23/25 3:00 PM

* Creating survey to send to local and government agencies for their feedback on the project objectives and design
* Receive feedback on app features and functionalities
* Receive ui design and user experience
* determine focus of next week's development activities

Phase 2 Deadline – Monday 03/24/25

* Receive feedback on app features and functionalities
* Solidify ui design and user experience
* Solidify database schema
* Solidify integration techniques and software for extension and external websites
* UI Coding and Documentation for Login, Launch page, create account and home page
* Create Databases from finalized database schema
* Determine focus of next week’s Hackathon
* Implement software engineering principles

Phase 3 Deadline – Unit testing and finalizing

* Finalize the required UI and Database to require users to login and create account
* Develop required UI and controllers to allow users to either Donate or receive items
* Enable machine learning techniques to ensure resource quality before donating
* Develop UI and controllers to save and track all donor and receiver activity and ensure guidelines are being followed.
* Implement unit testing and user trial testing

Final client meetings and survey – 04/18/25

* Give clients and users the opportunity to test and give feedback on software on last time before final deployment.
* Work on issues related to user feedback
* determine focus of next week hackathon

Final Deadline – Wednesday 04/23/25

* All features are in working condition and thoroughly tested
* All UI components are in the final revision and should allow users to donate, receive and view their history, status and benefits.
* Integration of extension and external website has been tested and is successful

**1.5 Configuration management plan**

* VCS Technologies – the developer will use a github repository as they have experience navigating this software. Additionally, they have received no direction from the client (professor) and subclient (GTA). Due to the significance of safety, some machine learning algorithms are recommended to verify the quality of donated resources.
* VCS Methodology – the developer will be following the Gitflow Workflow, a framework on branches in Git. When code is to be committed to the central repository a pull request is to be initiated and should include two reviewers: the developer and the GTA. external client inputs will be taken into account also using information from the surveys.

**1.6 Technologies**

* All of the technologies listed were selected due to the sudden nature of this project and team removal. The technologies are chosen by the developer because they have experience in navigating them and building projects with them. As the initial stages of the project progresses technologies are subject to change until research is finalized
* Client-side Technologies - react native and node.js will be used to complete the front-end portion of the application,
* Database Technologies - Firebase will be used to track users and their activity
* Server-side Technologies - Flask Microframework for Wrapping and integrating Machine learning models
* Machine Learning models: The Quality Assurance models are written entirely in python. The image classification model is a CustomConvNet is a deep convolutional neural network designed for image classification tasks. It extracts features from input images using stacked convolutional blocks and outputs class scores using a global average pooling layer followed by flattening. While the label classification model is a Rnn\_Lstm is a bidirectional Long Short-Term Memory-based neural network designed for sequence classification tasks, especially those involving textual data. The model consists of an embedding layer to convert input tokens into dense numerical representations, followed by multiple layers of a bidirectional Long Short-Term Memory network. After processing the sequence data, the model uses two fully connected layers to make the final prediction. Both models are pre trained and licensed to use and modify.