Black Bear Foodshare

System Design Document

Olive Food Solutions
Dr. Scott Marzilli
Team Members: Dumas Wesley, Kaulenas Corey, Moody-Broen Makai, Sima Denis, Sholler Jakob

Black Bear Foodshare
System Design Document

Table of Contents

1 Introduction

This is a capstone project for Dr. T Scott Marzilli, Associate Provost of Student Success & Innovation, in partial fulfillment of the computer science BS degree for the University of Maine. Dr. Mazilli places great emphasis on the importance of providing the necessary resources and opportunities for students to succeed at the University of Maine. The impetus of this project comes from the passion Dr. Marzilli and our team share in providing judgement free opportunities for students dealing with food insecurity, as well as furthering the elimination of food waste on our campus. The purpose of this project is to develop an application to allow hosts of catered events to easily inform students of excess food on campus. These posts will include the food, photos of the food, and the location. The goal of this project is to reduce waste and increase sustainability on campus, while also complying with regulations to make sure the excess food is safe to consume.

1.1 Purpose of This Document

The purpose of this document is to inform our team, client, and interested parties of the intended design and functionality of the Black Bear Foodshare application.

1.2 References

- Olive Food Solutions, System Requirements Specification
  System Requirements Specification

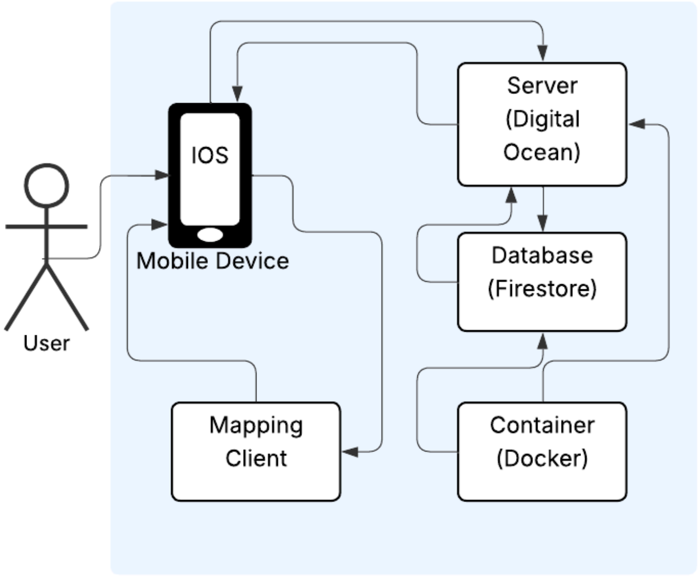
2 System Architecture

This section of the System Design Document contains section 2.1 Architectural Design, and section 2.2, Decomposition Description. This section outlines the necessary technologies that will be implemented in the design of the Foodshare application. Section 2.2 specifies the functions and methods that will be used to develop the application.

2.1 Architectural Design

Our system shall operate as an IOS only mobile application. It will be developed in XCode, using the proprietary IOS object oriented development language, Swift. For the backend we will be using Firestore as our development tool. The backend will be programmed with Python as our language of choice. Digital Ocean will be the hosting service for the corresponding webserver, which will be programmed with JavaScript. Docker will be used as the container service for the application. Depending on time and technological constraint an Android service may be developed alongside the IOS service. If the aforementioned constraints permit development, the Android application will be programmed with Kotlin.

The programming of the Swift application will be responsible for the frontend of the application. This will involve the design of the Home page, Post creation menu, and the structure of the individual posts. The backend will be hosted through Firestore, we will be using Python to code the backend. This will store a limited amount of user information, post information, and survey data.



2.2 Decomposition Description

Our System follows an object-oriented design and utilizes the Model-View-Controller pattern. The pattern is split into three layers.

- Model Layer - Database; utilizing firestore.
  - Handles Data logic

- View Layer - Frontend; utilizing swift for IOS
  - Handles data presentation

- Controller Layer - Backend; utilizing python and firestore.
  - Handles request flow

System Components

- Server - Digital Ocean
  - Digital Ocean is what our system uses to host the webserver for the project.

- Frontend - Swift
  - (Frontend description)

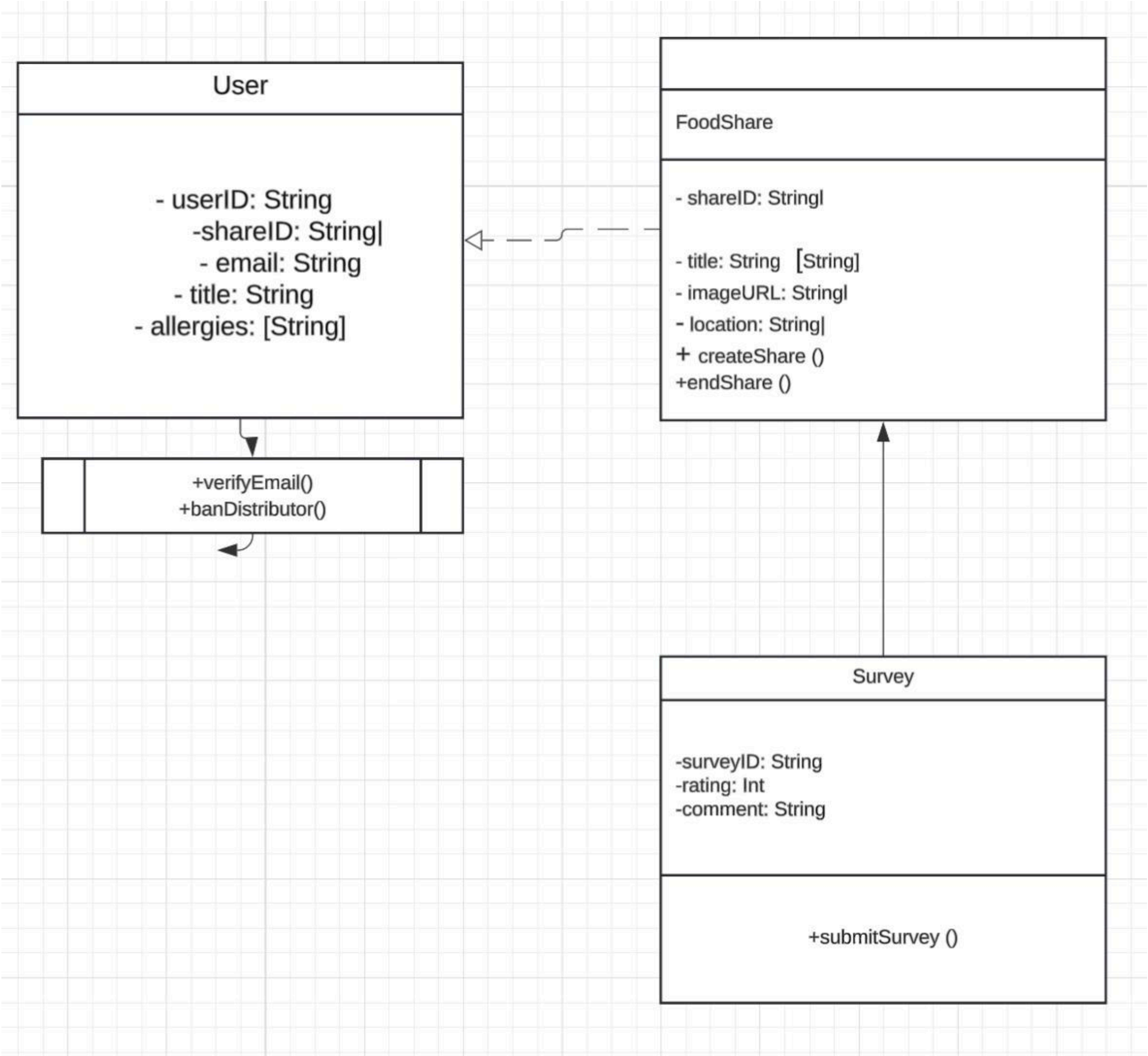
- Backend - python firestore

    - (Backend description)

- Database - firestore
    - (Database description)

Class Diagrams (See last three pages for guidelines)

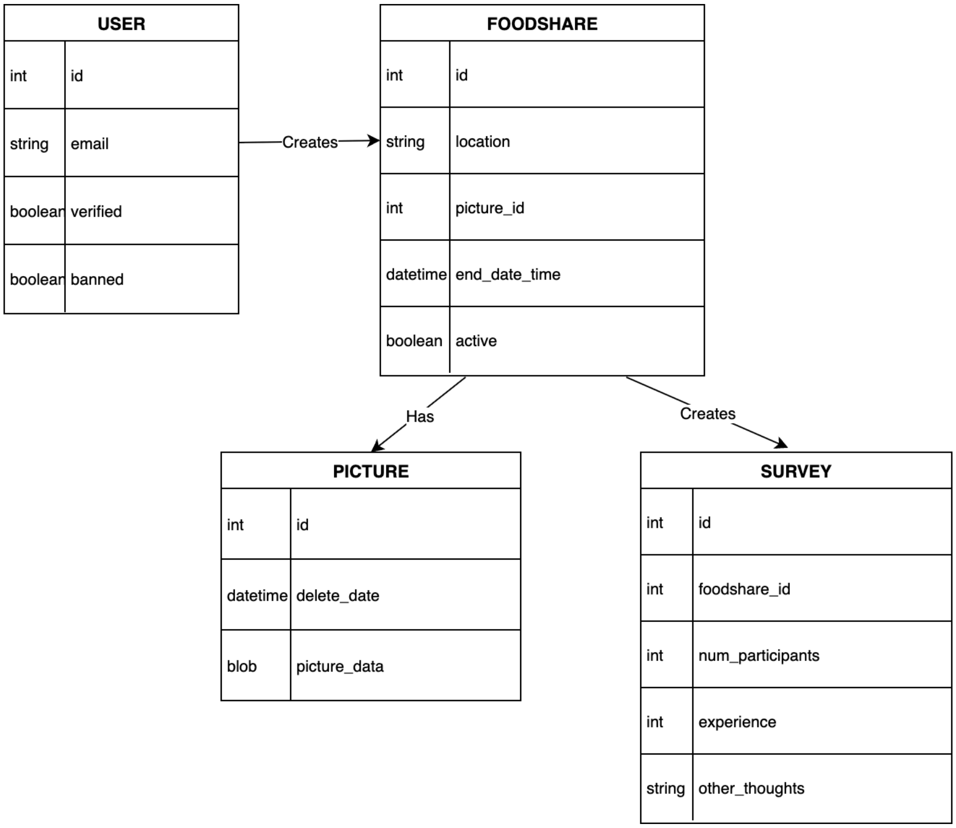
Class: User
Attributes: Email → string SMS notifications → boolean Push notifications → boolean Phone number → string Verified → boolean Banned → boolean
Operations:



3 Persistent Data Design

3.1 Database Descriptions (if you use a database)

Our system will use a nosql database. We will have fields for the User, Foodshare, Picture, and Survey. The User, Foodshare and Survey will be permanent, while the foodshare and pictures will be deleted from the database after a certain amount of time.



3.2 File Descriptions

Our system uses no unique file structure. The database will use standard file types like jpeg, zip, json, and csv.

4 Requirements Matrix

The requirements matrix shows how each functional requirement in the System Requirements Specification corresponds with the components in the System Design Document. Its purpose is to ensure that every requirement is addressed in the system design and has methods or functions to complete the requirements. It will be referenced during implementation to determine the details of how each of these different requirements should be implemented in practice to work together to create the overall product.

Functional Requirement use case number	Functional Requirement name	System Component

UC-001	Users shall be able to sign up for push notifications.	subscribePush()
		subscribeSMS()

UC-002	Create food share	createShare()
		updateDatabase()

UC-003	Hosts shall be able to upload and post images inside their posts	uploadImage()
UC-004	View current Food Shares	getActiveShares()

UC-005	Ending food shares	endShare()
		presentSurvey()

UC-006	User Verifies Email	sendVerificationEmail()
		verifyEmail()

UC-007	User Filters Food Shares by Allergy.	filterShares()

UC-008	Host Adds Allergen warning.	addAllergenWarning()
		saveToDatabase()

UC-009	User accesses location map service	openMappingSoftware()
UC-010	View Food Share	getShare()

UC-011	Admin Views Food Share Data	getAllShares()
		exportShares()
		downloadShares()

UC-012	Admin Deletes Food Share	deleteShare()
		updateDatabase()

UC-013	Admin Bans Food Distributor	banDistributor()
		updateDatabase()

Appendix A – Agreement Between Customer and Contractor

The customer and team agree that the document represents a finalized version that meets all agreed upon criteria and requirements. Both parties have thoroughly reviewed the document and all of its contents and agree that the information and structure meet the standards of the requirements. Any feedback or revisions have been addressed and remedied. The document is considered complete and ready for use.

If changes to the System Design Document are required, the team will begin by preparing a draft of the document with the changes. After the draft has been prepared, each team member will review the draft and note any problem with it they might have. Once all the problems have been addressed, the team will review the document once more and sign off on it, to show that they are satisfied with the state of the document. Once that has been completed, we will submit the draft to the client for review and approval.

Wesley Dumas
Signature: ________________________Date:

Corey Kaulenas
Signature: _____________________Date:

Makai Moody-Broen
Signature: ______________________Date:

Denis Sima
Signature: ____________________________Date:

Jakob Sholler
Signature: ________________________Date:

Dr. Scott Marzilli
Signature: ______________________________________Date:
Comments:

Appendix B – Team Review Sign-off

All team members have thoroughly reviewed this document and reached full agreement on all of its content. No major concerns have been raised, and any minor concerns or clarifications raised by individuals are noted in the comments below their signature. Additionally, all members of the team have agreed upon the formatting and presentation of this document with no major complaints.

Wesley Dumas
Signature: ________________________Date:
Comments:

Corey Kaulenas
Signature: ________________________Date:
Comments:

Makai Moody-Broen
Signature: ______________________Date:
Comments:

Denis Sima
Signature: __________________________Date:
Comments:

Jakob Sholler
Signature: _______________________________Date:
Comments:

Appendix C – Document Contributions

Sholler Jakob, Contributions:
- Partial Decomposition description / outline

Dumas Wesley, Contributions:
- Requirements Matrix
- Appendix A, B

Moody-Broen Makai, Contributions:
- Introduction
- Purpose
- Architectural design

Kaulenas Corey, Contributions:
- Database Description
- File Description

Sima Denis, Contributions: