# ABSTRACT

Food Bridge combats the twin problems of food waste and hunger by creating a seamless connection between hotels with excess food and orphanages struggling with food insecurity. This innovative project hinges on a user-friendly web application that acts as a communication bridge. Hotels can leverage this platform to post real-time alerts about their surplus food, specifying details like type, quantity, and preferred pick-up times. Orphanages and other authorized recipients benefit from a notification system tailored to their location and preferences, ensuring they're promptly alerted to available food before it spoils.

The impact of Food Bridge extends far beyond simply connecting donors and recipients. By diverting edible food from landfills, the project plays a crucial role in promoting environmental sustainability. Reduced food waste translates to less methane emissions from decomposition, while also conserving valuable resources used in food production. Furthermore, Food Bridge directly addresses food insecurity by providing orphanages and other vulnerable communities with access to nutritious meals, a critical factor for overall well-being, especially for children.

This platform goes beyond efficiency by fostering collaboration and social responsibility within the community. Hotels play a vital role by donating surplus food, while orphanages and other recipients benefit from this generosity. Food Bridge strengthens the social fabric by encouraging businesses to give back and participate in a more sustainable and equitable food system. This innovative approach has the potential to create a positive ripple effect, promoting environmental responsibility, enhancing food security, and fostering stronger communities.

சுருக்கம்

# CHAPTER 1

# INTRODUCTION

# 1.1 OVERVIEW OF THE PROJECT

The Food Bridge tackles food waste and food insecurity by connecting hotels with excess food to orphanages in need. A web application allows hotels to list their surplus food, while notifying nearby orphanages about the available donations. This approach not only helps feed people but also reduces environmental impact by diverting food from landfills and promoting sustainability. Food Bridge fosters a collaborative community where businesses can give back and contribute to a more equitable food system.

# LITERATURE SURVEY:

# EXISTING SYSTEM:

# DEMERITS:

* + 1. Possible lack of comparative analysis
    2. Complexity and usability.

# PROPOSED SYSTEM:

# MERITS

* + 1. We compared more than two algorithms to get a better accuracy level.
    2. We build a user-friendly web application.

# OBJECTIVE AND SCOPE

Food Bridge's primary objective is to establish a seamless connection between hotels with surplus food and orphanages facing food insecurity. This web-based platform aims to achieve the following:

* **Reduce food waste:** By diverting edible food from landfills, Food Bridge promotes environmental sustainability.
* **Enhance food security:** Orphanages and other authorized recipients gain access to nutritious meals, improving their well-being.
* **Foster community connections:** The platform facilitates collaboration and social responsibility within the community by connecting donors with recipients in need.

The scope of Food Bridge encompasses the development and implementation of a user-friendly web application. This application will cater to both hotels and orphanages, offering the following functionalities:

* **Registration and account management:** Hotels and orphanages can register and manage their accounts, including specifying preferences and updating relevant information.
* **Real-time notification system:** Hotels can post alerts about available surplus food, including details like type, quantity, and preferred pick-up time. Orphanages and other recipients will receive instant notifications based on their location and preferences.
* **Communication and coordination tools:** The platform facilitates communication between donors and recipients, ensuring smooth scheduling and coordination for food pick-up and delivery.

# ORGANIZATION OF THE REPORT

Chapter 1 - This chapter gives a brief introduction about the overview of the entire project, literature survey, existing & proposed system and the objective & scope of the project.

Chapter 2 - This chapter gives a brief introduction about the overall description, product perspective, product functions, user characteristics, operating environment, constraints and system requirements and the tools and technologies involved in the development of the project.

Chapter 3 - This chapter explains the overall system architecture. It explains how a user interacts with the system, how intercommunication happens between various services within the system and the various testing methods used for testing the platform.

Chapter 4 - This chapter gives a detailed explanation of how each module was developed and the various demonstrated images captured of the project.

Chapter 5 - This chapter summarizes the whole project and talks about the future enhancements that can be done in the platform.

# CHAPTER 2

**REQUIREMENT SPECIFICATION**

# OVERVIEW DESCRIPTION

The main purpose of Requirement Specifications is to describe in a precise manner all the capabilities that will be provided by the Software Application “PREDICTION OF STOCK PRICE USING REGRESSION MODEL” which deals with Linear algorithm, Random Forest algorithm and ARD algorithm for predicting the future stock price. The data collection and preprocessing describe the data sources and methods for collecting and preprocessing the data for the prediction task using data cleaning.

# PRODUCT PERSPECTIVE

The system or the project will use regression models, which are statistical techniques that analyze the relationship between a dependent variable (stock price) and one or more independent variables (such as market trends, company performance, economic indicators, etc). The system or the project will find a mathematical function that best fits the data and can be used to estimate the stock price for any given value of the independent variables.

# PRODUCT FUNCTION

The system functions can be described as follows:

**User Interface Design:**

**User Login:**

Orphanage and Hotel User should be able to login into application with their Gmail account. New user should get login access after Admin approval.

**Admin Verification:**

Admin is responsible for verifying registration requests received from Orphanage and Hotels users. Admin needs to verify and approves the request for valid users. And should rejects the request for not valid users.

**Create Food Request:**

Orphanage User creates a Food request with food count. Request sends to nearby Hotels.

**Accepts Food Request:**

Hotel user having enough food can accept the request. When Hotel does not have enough food for the request, they can accept by providing food count they have.

**Declines Food Request:**

In case Hotel users do not have enough food for a request they can Decline it. Declined requests stilled be visible as open request for other hotel users to accept.

**Food Delivery:**

Once food request is accepted for required count, their contacts are shared to each other. Both users can contact each other and plan food delivery accordingly.

**Cancel Request:**

Orphanage user can cancel the raised request in case they got food from other source, they can be able to Cancel the request.

# USER CHARACTERISTICS:

**Admin user:**

Admin user is responsible for verifying and approving new user requests. Should be able to view open and closed requests. Admin can be able to access other user’s data and Food request details. But Admin won’t be able to update any data.

**Orphanage User:**

Orphanage user is responsible for creating food requests in the application. Only orphanage user can be able to create, accept, Close or Cancel the request.

**Hotel User:**

Hotel user is responsible for providing food for the requests by accepting it. They can also decline the request when not having enough foods.

# OPERATING ENVIRONMENT:

System requirements are the required specifications a device must have in order to use certain hardware or software. The hardware and software specification required for the application are listed below.

# Hardware requirements:

The software specifications used for the development of the application:

**Processor:** AMD Ryzen 5 5600H with Radeon Graphics, 3301 MHz

**RAM:** 8 GB

**Hard Disk:** 500 GB (SSD)

# Software Requirements:

The software specifications used for the development of the application:

**Front-End Framework:** Angular 17, Ionic

**Development Environment:** Visual Studio Code

**Database Management:** Firestore

**Hosting:** Firebase

**Login Management:** Google OAuth

# CONSTRAINS:

The system or the project will only use the closing price of the stock as the dependent variable for the prediction task, which may not reflect the intraday fluctuations or volatility of the stock price.

# SPECIFIC REQUIREMENTS

The following subsections of Requirement Specifications should facilitate in providing the entire overview of the “PREDICTION OF STOCK PRICE USING REGRESSION MODEL” under development. This document aims at defining the overall software requirements for developers. Efforts have been made to define the requirements of the prediction system exhaustively and accurately.

# EXTERNAL INTERFACE REQUIREMENTS:

# SYSTEM FEATURES

This system needs to have certain features to enable its round the clock use by multiple stakeholders with similar but not the same requirements. Hence, the set system is designed to have the following elements: -

**High availability** - The system should be available under 24x7 with an acceptable down-time.

**Security** - Only authorized member should be able to access the system as deals with personal and medical data of the patients.

**Usability** - Its user interface should be highly responsive, easy to learn and operate and require less-time to onboard health workers and diagnostic staff.

**Reliability** - As this system deals with life and emergencies of the patients, it is critical that every component of the system functions as expected all the time. This will be proper testing and continuous monitoring post deployment to achieve the goal.

# UML:

UML, short for Unified Modelling Language, is a standardized modelling language consisting of an integrated set of diagrams, developed to help system and software developers for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modelling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems. The UML is a very important part in the software development process and uses mostly graphical notations to express the design of software projects. Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software.

# USE CASE DIAGRAM

Use case diagrams are considered for high level requirement analysis of a system. So when the requirements of a system are analysed the functionalities are captured in use cases. So, it can say that uses cases are nothing but the system functionalities written in an organized manner.

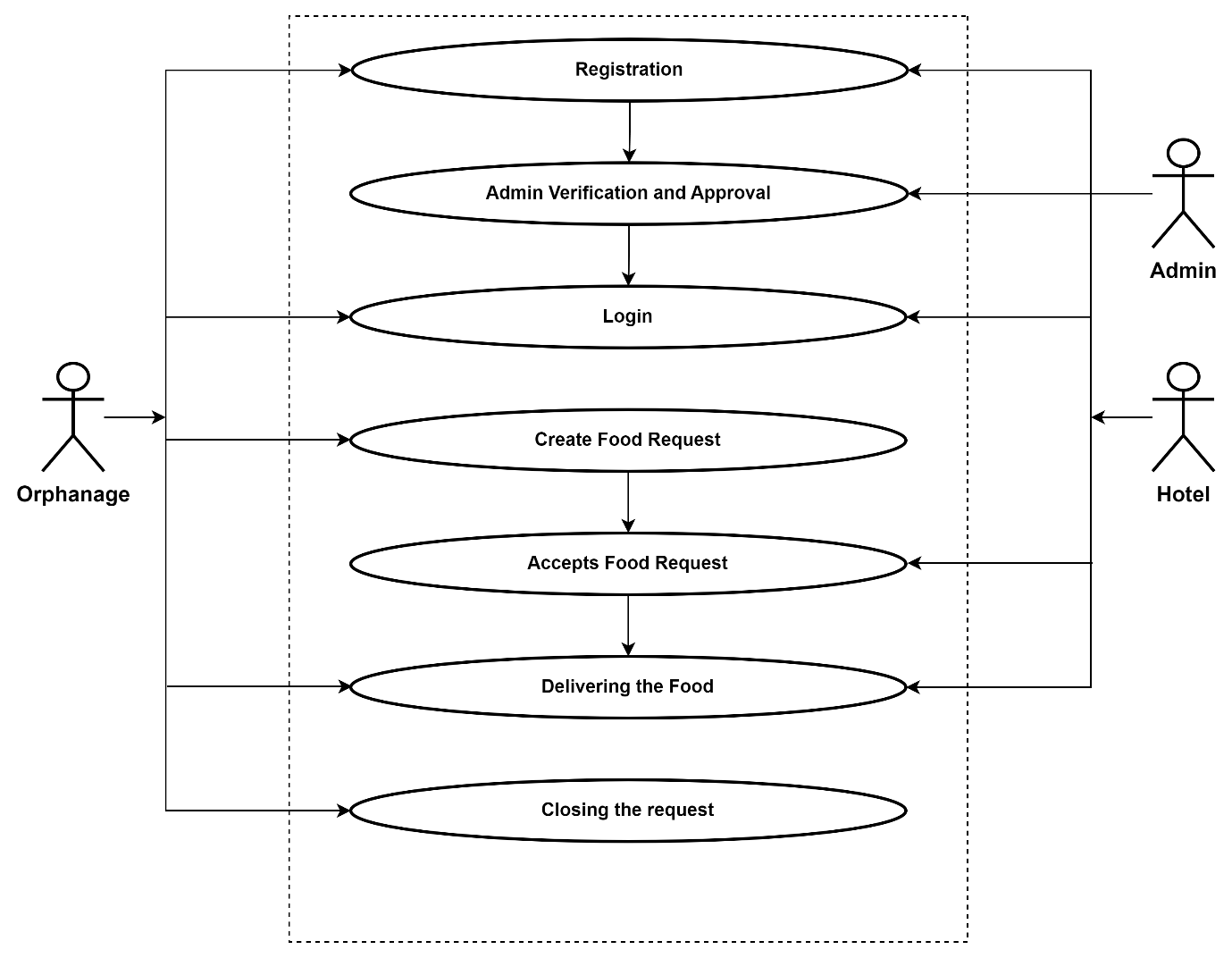


Figure 2.1 Use Case Diagram

# Class Diagram:

Class diagram is basically a graphical representation of the static view of the system and represents different aspects of the application. So a collection of class diagrams represent the whole system. The name of the class diagram should be meaningful to describe the aspect of the system. Each element and their relationships should be identified in advance Responsibility (attributes and methods) of each class should be clearly identified for each class minimum number of properties should be specified and because, unnecessary properties will make the diagram complicated. Use notes whenever required to describe some aspect of the diagram and at the end of the drawing it should be understandable to the developer/coder. Finally, before making the final version, the diagram should be drawn on plain paper and rework as many times as possible to make it correct.

A black background with white text

Description automatically generated

Figure 2.2 Class Diagram

# SEQUENCE DIAGRAM

Sequence diagrams model the flow of logic within your system in a visual manner, enabling you both to document and validate your logic, and are commonly used for both analysis and design purposes. Sequence diagrams are the most popular UML artifact for dynamic modelling, which focuses on identifying the behaviour within your system. Other dynamic modelling techniques include [activity diagramming](http://agilemodeling.com/artifacts/activityDiagram.htm), [communication diagramming](http://agilemodeling.com/artifacts/communicationDiagram.htm), [timing diagramming](http://agilemodeling.com/artifacts/timingDiagram.htm), and [interaction overview diagramming.](http://agilemodeling.com/artifacts/interactionOverviewDiagram.htm) Sequence diagrams, along with [class](http://agilemodeling.com/artifacts/classDiagram.htm) [diagrams](http://agilemodeling.com/artifacts/classDiagram.htm) and [physical data models](http://agiledata.org/essays/dataModeling101.html) are in my opinion the most important design-level models for modern business application development

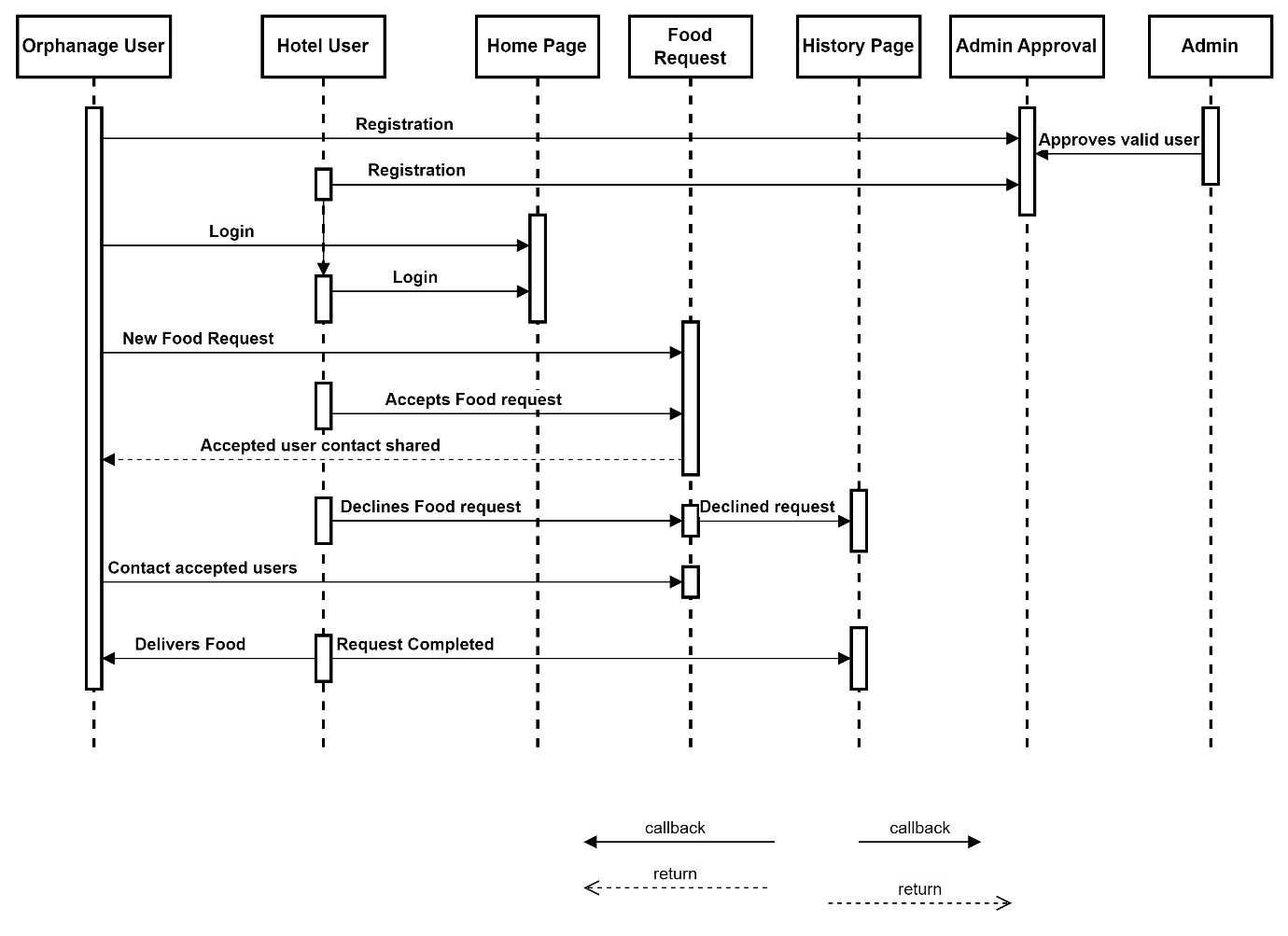


Figure 2.3 Sequence Diagram

# WORKFLOW DIAGRAM

A workflow diagram (also known as a workflow) provides a graphic overview of the business process. Using standardized symbols and shapes, the workflow shows step by step how your work is completed from start to finish.

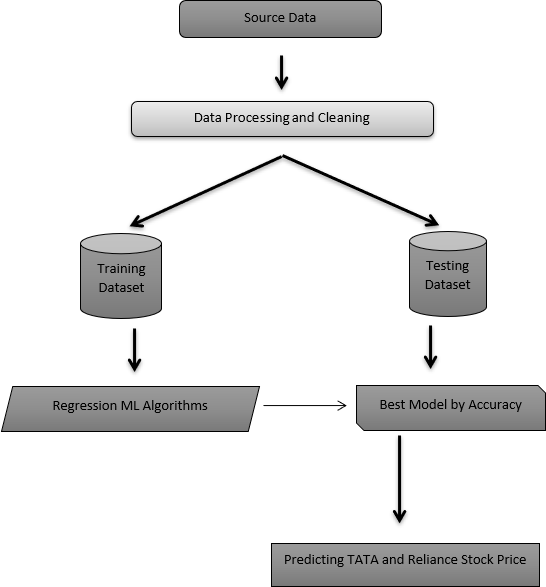


Figure 2.4 WorkFlow Diagram

# ENTITY RELATIONSHIP DIAGRAM

An entity relationship diagram (ERD), also known as an entity relationship model, is a graphical representation of an information system that depicts the relationships among people, objects, places, concepts or events within that system. An ERD is a [data modeling](https://searchdatamanagement.techtarget.com/definition/data-modeling) technique that can help define business processes and be used as the foundation for a [relational database](https://searchdatamanagement.techtarget.com/definition/relational-database). Entity relationship diagrams provide a visual starting point for database design that can also be used to help determine information system requirements throughout an organization. After a relational database is rolled out, an ERD can still serve as a referral point, should any debugging or business process re-engineering be needed later.

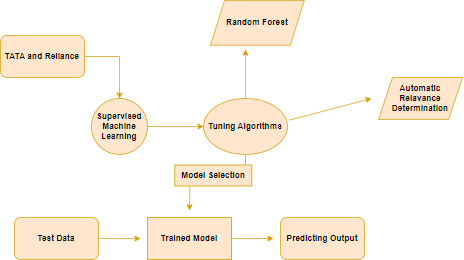


Figure 2.5 Entity Relationship Diagram

# ACTIVITY DIAGRAM

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing dynamic nature of a system but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in activity diagram is the message part. It does not show any message flow from one activity to another. Activity diagram is some time considered as the flow chart. Although the diagrams looks like a flow chart but it is not. It shows different flow like parallel, branched, concurrent

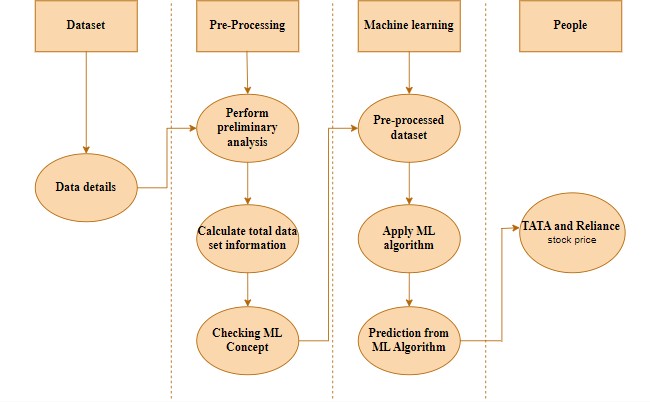


Figure 2.6 Activity Diagram

# DATA FLOW DIAGRAM

Machine learning needs data gathering have lot of past data. Data gathering have sufficient historical data and raw data. Before data pre-processing, raw data can’t be used directly. It’s used to pre-process then, what kind of algorithm with model. Training and testing this model working and predicting correctly with minimum errors. Tuned model involved by tuned time to time with improving the accuracy.



Collecting the Data

Pre-Process the Collected Data

Choose the model



Train model

Test model

Tune model

Prediction

Figure 2.7 Process of dataflow diagram

# SOFTWARE QUALITY ATTRIBUTES

**Reliability** – Reliability is the probability and percentage of the software performing without failure for a specific number of uses or amount of time.

**Availability** - This feature defines the amount of time the system is running, the time it takes to repair a fault, and the time between lapses. We plan a system to be available 24x7.

**Maintainability** - This feature indicates the average time and ease and rapidity with which a system can be restored after a failure.

**Security** - Security measures ensure your software’s safety against espionage or sabotage. As prediction system contains lot of sensitivity data especially personal records of clients.

**Data integrity** - Data integrity refers to maintaining and assuring data accuracy and consistency over its entire lifecycle. If this factor is corrupted, data is lost due to a database error.

**Usability** - This feature concerns the users; it indicates how effectively they can learn and use a system.