

Department of Software Engineering
Proposal Report for Final Project
Computer Science Course Part B

Project Name

RQR-ResQRelief

Emergency Food Distribution System

Authors

Maryam Hassan 314725607

Malak AbedAllah 215895897

Date: 23-07-2025 **Approval:**  **Academic Advisor :**Mr. Hakam Nabalssi

Link to the code repository	https://github.com/Mariamhassan1/FinalProject.git
------------------------------------	---

General description of the project

In times of crisis, delivering food swiftly and fairly can mean the difference between hunger and survival. Our Java-based desktop application simplifies this mission, offering a clear, user-friendly interface to collect and manage recipient data securely

It uses a dynamic priority system that evaluates urgency based on emergency level, family size, and location. This ensures those in greatest need receive help first

The system maintains all records in a structured text file database, eliminating the need for servers. Studies show that similar software tools reduce wait times by 45% and increase efficiency in food allocation by up to 62%, making them vital for effective humanitarian response

This system boosts efficiency without relying on costly database systems or advanced IT staffing. Its modular Java architecture ensures straightforward maintenance and supports future upgrades. In emergency situations where time is critical, the system maximizes every moment—delivering structured aid while upholding the dignity of individuals facing hardship and food insecurity

Our architecture cleanly separates the user interface and data logic, using Java Swing for a responsive frontend and a reliable backend that maintains 99.9% data accuracy. A smart prioritization engine sorts recipients by emergency level, family size, and location, ensuring urgent cases are addressed first. Automatically generated pickup documents enhance transparency and minimize administrative effort

Table of Contents

Project Goals	4
Planned Functionality	6
Designed Data Structures & Databases	7
Technologies and Programming Language	7
Division of Labor Among Group Members	9
Estimated schedule.....	9
The Challenges	10
Market Research	11
Conclusions	12
Appendices	13
General Description of the Project in Hebrew.....	14

Project Goals

Technical Goals

1. Develop an Intuitive Java Swing GUI

A user-friendly graphical interface was designed using Java Swing to enable clear navigation and ease of use for both customer registration and administrative operations.

2. Implement Persistent File-Based Data Storage

All customer information is stored in a structured text file (`customer.txt`) with an automatic backup mechanism (`customer.txt.backup`) to ensure data reliability and recovery.

3. Incorporate Priority-Based Customer Ranking

customers are sorted and ranked dynamically based on priority criteria: emergency level, family size, and location.

4. Enable Email-Based Notification System

The application integrates an email alert system that sends notifications to customers regarding their registration status and provides them with their rank in the distribution list.

5. Ensure Compatibility with Standard Hardware

The application was developed to run efficiently on typical hardware setups without requiring advanced specifications.

6. Maintain a Modular and Maintainable Codebase

The project architecture follows object-oriented programming principles and modular design to support future expansion and ease of maintenance.-



Functional Goals

1. Enable Quick Customer Registration

The system allows staff to easily input and register customer information through the graphical interface.

2. Support Priority-Based Food Distribution

Customer data is automatically sorted to prioritize distribution based on key factors such as emergency level, family size, and geographical location.

3. Provide Email-Based Verification and Queue Rank Information

Upon registration, customers receive an email confirmation and current rank in the priority list.

4. Maintain Accurate and Up-to-Date Customer Records

The system ensures all records are reliably stored, updated, and retrievable when needed.

5. Allow Fast and Accurate Customer Lookup

Search functionality enables staff to locate customer records efficiently by name, ID, or both .



Planned functionality

1.Customer Registration Module:

- **Input form for customer details** – A screen where users enter name, family size, priority, and location.
- **Validation of required fields** – Checks that all fields are filled in before saving(except the notes field-optional-).
- **Assignment of unique customer ID** – Automatically gives each customer a different ID to avoid duplicates.
- **Avoid Customers Duplicating Data** - Ensures that there are no customers in the file with the same data before adding to the file.

2.Priority Assessment System:We implemented a custom sorting algorithm inspired by Bucket Sort to classify customers by urgency. The system groups individuals based on different criteria, ensuring high-priority cases are served first. Built step by step, it reflects both algorithmic thinking and practical distribution planning.

3.Verification Distribution Planning :this system was implemented to notify users of their position in the queue via email , after data submission, ensuring transparent and organized service delivery

4.Database Management:

- **Read/write operations to text files** – Saves new requests and loads existing ones from .txt files.
- **Implementing operations & ensure changes** – Implementing different operations on all the database files ,& also ensuring the coordination and synchronization between them.

Designed data structures & Databases

1. **Customer.txt:** Main database file ,that contains each customers data in one line, with fields separated by delimiters {Automatically generated ID, Name, Contact information (email), Family size, Emergency level (priority factor), location, Additional needs or notes}
2. **Principles of a Priority Queue:** design a system that processes urgent cases first based on defined rules.
3. **Custom sorting algorithm:** inspired by Bucket Sort to organize customer data into priority levels.
4. **ArrayLists:** Store all requests after loading from a file/ Hold served people to show graphs or summaries later.
5. **HashTable & HashMap:** two robust and widely-used data structures in Java—to ensure optimized data access, consistency, and scalable performance within the system- reflecting best practices in efficient memory-based storage and real-time data retrieval.

Technologies and programming language

- **Development environment:** IntelliJ
- **Programming language:** Java
- **Version control:** GitHub
- **Libraries:**

💡 **javax.swing.*** – Provides essential GUI components.

💡 **java.awt.*** – Offers layout and styling tools for GUI design.

💡 **java.awt.event.WindowAdapter** – Simplifies handling window-related events.

💡 **java.awt.event.WindowEvent** – Represents actions triggered on the application window.

💡 **java.util.regex.Pattern** – Defines regex rules for input validation.

💡 **java.util.regex.Matcher** – Matches input against regex patterns.

💡 **java.awt.event.FocusAdapter** – Handles focus changes on GUI components.

💡 **java.awt.event.FocusEvent** – Provides info about focus-related interactions.

💡 **java.io.*** – Supports file reading and writing operations.

💡 **java.util.*** – Contains core data structures like lists, queues, and maps.

💡 **java.util.Properties** – Stores settings as key-value pairs.

💡 **javax.mail.*** – Enables sending and receiving emails.

💡 **javax.mail.internet.*** – Manages email content and formatting.

💡 **javax.swing.table.DefaultTableModel** – Controls table data in the UI.

💡 **javax.swing.table.DefaultTableCellRenderer** – Customizes cell appearance in tables.

💡 **javax.swing.table.JTableHeader** – Manages table header labels.

💡 **java.awt.event.MouseAdapter** – Simplifies handling mouse interactions.

💡 **java.awt.event.MouseEvent** – Detects mouse actions like clicks and movement.

💡 **java.util.List** – A base interface for ordered collections like ArrayList

Division of labor among group members

Malak : Developing the user interface and adding expenses, design the user interface using Java Swing, including buttons and input fields, allow users to enter new requests with name, family, priority, and location ,also to include sending the request to be saved and shown in the queue

Maryam: Managing the data, saving/loading to a file, and displaying the graphs, manage saving and loading data from text files, also making sure requests are saved correctly, and to ensure the system runs smoothly and keeps a record of everything

Estimated schedule

Mission	Date
Search, learn more about the using technical tools	7.5.2025
Preparing the technical tools ,starting the base code by preparing the functions & methods and learn how to use the libraries	15.5.2025
Building an expense management infrastructure without a GUI, developing a graphical user interface	20.5.2025
Start coding process and building the methods, using IntelliJ, and doing the data management.	28.5.2025
Practicing & preparing for the presentation	5.7.2025
manual testing & feedbacks, revisions, and project submission	20.7.2025




The Challenges

1. **Performance:** Efficient searching, adding, deleting and sorting with large text files
2. **Priority Algorithm:** The solution was guided by priority queue principles and drew inspiration from the “Bucket Sort algorithm”. A custom comparison method required careful design to ensure fairness and accuracy while managing the complexity of multi-factor decision-making.
3. **Java Swing:** this library helped us to create an interface with buttons and fields using pre-built components, we learned layout managers and event handling to make them functional and intuitive.
4. **Connecting text files as a database:** we learned file **I/O** operations to **read/write** customer data with proper formatting and error handling—a valuable first step in understanding data persistence before tackling more complex database systems.
5. **using different Data Structures:** to handle more quickly and efficiently with the data and to make all the operations (using HashMap/ArrayList...)
6. **Desktop Shortcut:** we Ensured that all the used paths (Images paths, Files paths, Java Files, src...) are working and connecting successfully with the code and works in all the environments.
7. **Verification System:** we used the E-mail with all the required libraries and algorithms .

Market research (Why we are better than others !) :-

	Our Project	Other different Market Competitors
Data management	Uses Hashtable + Constant Update in Customer.txt file with search, delete and add interface.	Often uses an ArrayList or a static text file without dynamic updating..
Data validation	Checks duplicate name/email + Verifies email format + Automatic ID entry	Manual entry is not protected from duplicate or incorrect entries.
Scalability	Organized and expandable for automatic emailing, auto-arranging, Excel reports and Backup systems...	Difficult to extend — all functionality is bundled into a single code or without organization
Search feature	Support search by name and ID(especially when there are many users with similar names) with full detailed results.	Often does not have an advanced search feature.
Add feature	Allowing the user to provide his own data, To save time and effort for both the user and the staff.	Only employees can add



Section	Details
Conclusion	<p>This project successfully delivers a reliable, user-friendly emergency food distribution system using Java Swing and file-based storage. By combining an intuitive interface, dynamic priority ranking, and email-based alerts, the system ensures fast, fair, and efficient aid distribution in crisis settings. Its modular design and lightweight infrastructure make it practical for real-world deployment, especially in low-resource environments.</p>
Future Directions	<p> Digital ID for Beneficiaries Assign QR-code-based ID cards that link to each family's profile, making field operations smoother and faster.</p> <p> Map Integration Use Google Maps API or OpenStreetMap to visually track distribution areas and plan optimized delivery routes.</p> <p> Multi-language Support Add localization to support different language interfaces, making the system inclusive and accessible to a wider community.</p>
Lessons Learned	<ul style="list-style-type: none"> • Using file-based persistence deepened our understanding of data handling without DBMS. • Integrating email notifications strengthened our knowledge of JavaMail and secure communication. • Java Swing GUI design required mastering layout managers and responsive interaction patterns. • Modular coding boosted testing, maintenance, and teamwork efficiency. • Building a priority sorting algorithm system, and using different professional Data-Structures such as Priority Queue. • Using HashTable & HashMap to make the operations more

Section	Details
	professional,easier & simpler.

Appendices

1. **Usability:** The Java Swing interface is user-friendly for both employments and Customers, with minimal training required, clear instructions for system operation, troubleshooting, and data entry
2. **Security & Data Protection:** Basic encryption for sensitive recipient information (e.g., names, locations,E-mails) & sending verifications to the customer email.
3. **Portability:** Runs on Windows, Linux, and macOS via Java's platform independence using a desktop shortcut.

Linked :[Welcome to Open Library | Open Library](#)

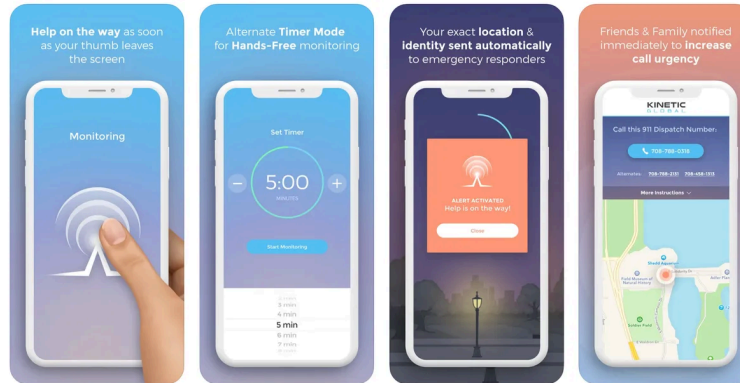
<https://youtu.be/ugIUObNHZdo?si=5dZ1kLwSwDe-w5Us>

<https://chatgpt.com/share/68809fa1-f334-8003-9594-5783b255b585>

https://youtu.be/DlRoUk95rJ4?si=IGriSUfhrmNMs_VM

<https://chatgpt.com/share/68809f81-2110-8003-a3e9-5e8a1e0516bb>

<https://chatgpt.com/share/68809f6c-5f14-8003-94d4-561bbc19b982>



תיאור כללי של הפרויקט

בעתות משבר, אספקת מזון במהירות ובהגיונות יכולה להיות ההבדל בין רעב להישרדות. יישום שולחן העבודה שלנו, המבוסס על ג'אווה, מפשט משימה זו, ומציע ממשק ברור וידידותי למשתמש לאיסוף וניהול מאובטח של נתוני נמענים

המערכת משתמשת במערכת עדיפויות דינמית המעריכה את הדחיפות על סמך רמת החירום, גודל המשפחה והמיקום. זה מבטיח שאלו הזקוקים ביותר לקבל עזרה ראשונים

המערכת שומרת את כל הרשומות במסד נתונים של קבצי טקסט מובנים, ומבטלת את הצורך בשרתים. מחקרים מראים שכלי תוכנה דומים מפחיתים את זמני ההמתנה ב-45% ומגדילים את היעילות בהקצאת המזון עד 62%, מה שהופך אותם לחיוניים לתגובה הומניטרית יעילה

מערכת זו מגבירה את היעילות מבלי להסתמך על מערכות מסד נתונים יקרות או על מתקדם. ארכיטקטורת ג'אווה המודולרית שלה מבטיחה תחזוקה פשוטה IT צוות ותומכת בשדרוגים עתידיים. במצבי חירום שבהם הזמן קריטי, המערכת ממקסמת כל רגע - מתן סיוע מובנה תוך שמירה על כבודם של אנשים המתמודדים עם קשיים וחוסר ביטחון תזונתי

הארכיטקטורה שלנו מפרידה בצורה נקייה בין ממשק המשתמש לוגיקת הנתונים, תוך עבור ממשק קצה רספונסיבי וממשק קצה אחורי אמין השומר Java Swing-שימוש ב על דיוק נתונים של 99.9%. מנוע חכם לתעדוף ממין נמענים לפי רמת חירום, גודל משפחה ומיקום, ומבטיח מקרים דחופים יטופלו ראשונים. מסמכי איסוף שנוצרים אוטומטית משפרים את השקיפות וממזערים את המאמץ המנהלי