Food Redistribution and Urgency Management System

Sonakshi Patil[1] Sanskruti Gaikwad[2] Ziya Shaikh[3] Arpita Wangari[4]
[1,2,3,4]Department of Artificial Intelligence and Machine Learning
[1,2,3,4]Zeal Polytechnic, Pune, Maharashtra, India

Abstract — Large quantities of edible food are discarded every day while many people remain undernourished. One of the major challenges is the lack of an efficient and coordinated system for redistributing surplus food before it spoils. This paper presents a Food Redistribution and Urgency Management System (FRUMS), designed to connect food donors, receivers, and volunteers through a common platform. The system manages surplus food listings and prioritizes them using an urgency score based on estimated transportation time, food perishability, and available quantity. A dispatch module supports the prioritization of food listings and coordination of deliveries. The proposed system aims to improve food redistribution, reduce avoidable food waste, and support efficient volunteer allocation. This paper describes the system design, urgency-based prioritization approach, and implementation considerations. Future work includes evaluating the system under different operating conditions and conducting pilot testing with local food donors and non-governmental organizations.

Keywords: Food Redistribution, Food Waste, Urgency Based Scheduling, Perishable Food, Volunteer Logistics

I. Introduction

Food loss and waste are significant ethical, environmental, and resource-related problems. The *Food Waste Index Report 2024* estimates that approximately 1.05 billion tonnes of food were wasted in 2022, representing about 19% of the food available to consumers, with households, food service providers, and retailers accounting for a major share [1]. During the same period, hundreds of millions of people experienced hunger [2]. Earlier research by the Food and Agriculture Organization (FAO) had also highlighted the substantial loss and waste of food along the supply chain [3].

A considerable amount of surplus food generated by restaurants, hotels, caterers, wedding halls, and retail shops may still be suitable for consumption when it becomes available for redistribution. However, such food may be wasted because donors lack an efficient way to identify receivers, and volunteer availability is limited. Existing redistribution practices often rely on phone calls, messaging groups, or charitable organizations operating at fixed times. These approaches may not adequately consider differences in food perishability and the time available for safe collection and delivery. For example, cooked meals generally require more urgent handling than properly stored packaged food. Therefore, a system that considers food urgency can help prioritize redistribution more effectively.

This paper proposes a Food Redistribution and Urgency Management System (FRUMS). Its main contributions are: (i) a platform connecting food donors, receivers, and volunteers; (ii) an urgency score based on the remaining safe time after accounting for transportation, food perishability, and available quantity; (iii) a dispatch procedure for prioritizing food listings; and (iv) a proposed evaluation framework for comparing different dispatch policies.

II. Related Work

Several mobile applications, such as OLIO and Too Good to Go, help connect surplus food with potential recipients, while local charitable organizations support food donation and redistribution. In India, the *Food Safety and Standards (Recovery and Distribution of Surplus Food) Regulations, 2019* provide a regulatory framework for the recovery and distribution of surplus food that is safe for consumption [4]. However, coordinating food collection and delivery remains challenging when food perishability, transportation time, and volunteer availability must be considered together.

The food redistribution problem is related to real-time scheduling, where tasks are prioritized according to their deadlines. The earliest-deadline-first (EDF) policy gives priority to tasks with earlier deadlines. A similar concept can be applied to food redistribution by considering the remaining safe time of each food listing. However, food redistribution also involves transportation and volunteer availability, which make it different from conventional scheduling. Multi-criteria decision-making can also help prioritize food listings according to factors such as remaining safe time, perishability, and quantity. Based on these concepts, FRUMS proposes an urgency-based approach to prioritize surplus food listings and coordinate their redistribution.

III. System Design

A. Actors

Donors post food listings that include the food type, quantity in meals, preparation time, packaging details, and pickup location. Receivers, such as NGOs, shelters, and community kitchens, register their locations, receiving capacities, and acceptable food categories. Volunteers register their availability and vehicle details. An administrator verifies donors and receivers and manages complaints.

B. Architecture

Fig. 1 illustrates the architecture of the proposed system. Clients communicate with the server through a REST interface [9]. The Listing and Request Module stores food listings and related events in the database. The Urgency Engine maintains a priority queue of open listings and recalculates their urgency scores at regular intervals. The Matching and Dispatch Module selects suitable receivers and volunteers for delivery. The Notification Service informs the relevant parties through application notifications or SMS. After delivery, the receiver confirms receipt, and the feedback is stored for future analysis.

All rights reserved by www.ijsrd.com

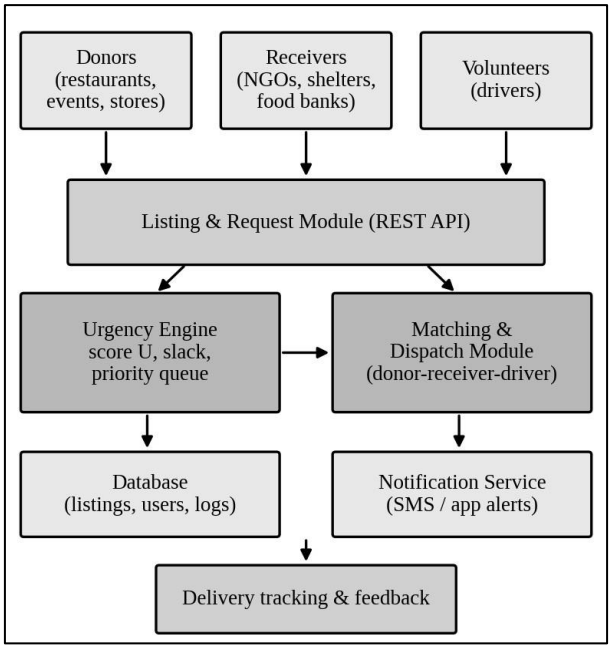


Fig. 1: Architecture of the proposed Food Redistribution and Urgency Management System (FRUMS).

C. Listing Life Cycle

A food listing passes through the following states: posted, matched, picked up, delivered, and expired. A listing automatically moves to the expired state when its defined safe-time limit is reached, preventing further collection and notifying the donor. If a volunteer cannot complete an assigned task, the volunteer may release it. The listing can then return to the priority queue, provided it is still eligible for redistribution, and its urgency score is recalculated.

IV. Urgency Management Model

A. Safe Time and Slack

Let listing i be posted at time t with an expiry time e_i. The expiry is obtained from the safe window s_c of its food category c and the preparation time. The remaining safe time is $r_i = e_i - t$. If the estimated trip time for pickup and delivery is d_i, the slack is

$$L_i = r_i - d_i, \quad (1.1)$$

where $d_i = (k_p + k_d) / v + h$. Here k_p is the distance from the volunteer to the donor, k_d is the distance from the donor to the receiver, v is the average speed and h is the handling time. A listing with $L_i < 0$ cannot be delivered safely and is not dispatched. This feasibility check is applied in every policy compared in Section VII.

B. Urgency Score

The urgency score of a feasible listing is

$$U_i = w_1 F_i + w_2 P_c + w_3 Q_i \quad (1.2)$$

The urgency score is a weighted combination of time pressure, food perishability, and food quantity. The weights are set to 0.60, 0.25, and 0.15, respectively, based on the assumed importance of each factor. A higher score indicates a higher dispatch priority. These weights are illustrative and require validation using real-world data.

where $F_i = 1 - \min\{\max(L_i / s_c, 0), 1\}$ is the time-pressure term, P_c is the perishability factor of the category, and $Q_i = q_i / q_{max}$, with q_i the quantity in meals and q_{max} the largest quantity allowed in one listing. The weights satisfy $w_1 + w_2 + w_3 = 1$. The first term grows as slack shrinks relative to the safe window. The second term gives cooked food an advantage over packaged food. The third term gives a small preference to larger quantities. Table I lists the category values used in this study, which are illustrative and should be replaced with values from local food safety guidance before any real deployment.

Categories	Safe window s (h)	Perishability P
Cooked meals	4	1.0
Bakery	8	0.6
Fresh produce	14	0.4
Packaged	30	0.1

Table I: Food Category Parameters Used in the Model

The weights were set to $w_1 = 0.60$, $w_2 = 0.25$ and $w_3 = 0.15$, reflecting the judgement that time pressure matters most, followed by perishability and then quantity. They can be derived more formally by pairwise comparison [8] and tuned with data from a pilot.

C. Dispatch Procedure

Whenever a volunteer becomes free, the dispatch module runs the following steps. (1) Remove expired listings and those with $L_i < 0$. (2) Compute U_i for all remaining listings. (3) Select the listing with the highest score. (4) Choose the nearest receiver that accepts the category and has spare capacity. (5) Notify the volunteer, donor and receiver. If no listing is feasible, the volunteer waits. Each run takes time proportional to the number of open listings, which is small for a city-level service.

V. Proposed Evaluation

The proposed Food Redistribution and Urgency Management System (FRUMS) can be evaluated by comparing four dispatch policies: First-In-First-Out (FIFO), Nearest Pickup First, Earliest Deadline First (EDF), and the proposed urgency-based policy. The evaluation should consider scarce and moderate volunteer availability. The main performance metrics are the percentage of meals delivered before expiry, the percentage of cooked meals delivered before expiry, and the total quantity of food rescued. Simulation experiments will be conducted to measure and compare the performance of these policies.

The system was evaluated by simulation, because no field data were available. Listings arrived as a Poisson process with a rate of six per hour for 12 hours. Categories were drawn with probabilities 0.35 (cooked), 0.20 (bakery), 0.25 (produce) and 0.20 (packaged). Quantity was uniform between 20 and 200 meals, the donor-to-receiver distance was uniform between 1 and 10 km, and the volunteer-to donor distance was uniform between 0.5 and 6 km. The speed was 20 km/h, the handling time was 15 minutes, and volunteers stopped taking tasks after hour 14.

Four dispatch policies were compared: first-in-first-out (FIFO), nearest pickup first (Nearest), earliest expiry first (EDF) and the proposed urgency score. The number of volunteers was 3 (scarce) or 4 (moderate). Each configuration was run for 300 random seeds and the mean values are reported. The main measures were the percentage of all meals

All rights reserved by www.ijsrd.com

delivered before expiry and the percentage of cooked meals delivered, since cooked food has the shortest safe window.

VI. Implementation Considerations

A. Data Model

The proposed system can use a relational database to manage user and food redistribution information. Four main tables are suggested: Users, Listings, Deliveries, and Feedback. The Users table can store user roles and verification status. The Listings table can store food category, quantity, preparation time, expiry time, location, and listing status. The Deliveries table can record listing, receiver, volunteer, and delivery timestamps. The Feedback table can store ratings and remarks. These records can support future analysis of waiting times and listings that expire without being delivered.

B. Food Safety and Trust

The proposed system should require donors to provide food preparation details and information about appropriate handling and storage. Receivers should be able to reject food that appears unsuitable for consumption, with each rejection recorded in the system. An administrator should verify donors and receivers before allowing them to post or accept listings. Food handling, storage, and redistribution should follow applicable food-safety requirements and guidance from the relevant national food-safety authority [4].

C. Usability and Privacy

The system should provide a simple listing process requiring minimal information, such as a photograph, food category, and quantity. Local-language interfaces and SMS notifications could improve accessibility for users with limited internet connectivity or smartphone access. Contact details should be shared only when necessary for a confirmed match. Location data should be collected and retained only as needed for delivery coordination, with appropriate privacy safeguards.

VII. Expected Evaluation and Discussion

The proposed urgency-based policy is designed to prioritize food listings with limited remaining safe time and high perishability. Its performance will be compared with FIFO, Nearest Pickup First, and EDF. The evaluation will examine whether urgency-based dispatch improves the delivery of highly perishable food while maintaining the total quantity of food rescued. Actual simulation results are required to determine the effectiveness of the proposed policy.

A. Limitations

The results come from a simplified simulation. It assumes one listing per trip, no traffic variation, no refusals by receivers and known expiry times. The category windows and weights are illustrative. Food safety also depends on storage temperature and handling, which the model does not capture. The findings should be treated as evidence of feasibility, not as a measure of real-world performance.

VIII. Technical Extension and Research Status

This extension develops the FRUMS design without altering the preceding manuscript. It adds implementation specifications, a receiver-aware dispatch procedure, failure recovery, and a reproducible evaluation protocol. These additions are proposed engineering developments, not evidence of a completed deployment. References [1]-[9] retain the numbering and bibliographic entries given on page 3.

A. Clarification of the Evidence Base

Sections VI and VII contain inconsistent descriptions of evaluation status: some passages describe a completed simulation, while others state that experiments remain future work. No numerical results, executable implementation, or per-seed records accompany the supplied manuscript. Accordingly, this extension treats the simulation settings as a proposed protocol. The statements about 300 seeds and volunteer configurations specify intended experimental conditions; they do not establish measured performance. No superiority over FIFO, Nearest, or EDF is claimed.

The four category windows in Table I are illustrative model inputs, not universal safe holding periods. A nonnegative slack value establishes only mathematical timing feasibility under the stated assumptions. It cannot establish that food is safe to consume. Food condition, temperature history, packaging, and handling must be checked separately. The regulatory basis remains the FSSAI surplus-food regulations [4], which address safe donation, handling, transport, and distribution before expiry.

B. New Development Objectives

The proposed update separates three decisions: whether food is eligible for redistribution, whether a particular delivery is feasible, and which feasible delivery receives priority. This separation prevents a large urgency score from overriding a safety rejection. It also makes decisions explainable: a listing may be eligible but temporarily unmatched because no receiver or suitably equipped volunteer is available.

The extension focuses on auditable decisions rather than adding an unvalidated predictive model. Every recommendation should retain its input timestamps, policy version, selected receiver, assigned volunteer, and exclusion reason. These records allow administrators to reconstruct why a listing was dispatched or left waiting. They also support future comparison of alternative weights without implying that any weight set is already optimal.

IX. Receiver-Aware Feasibility and Dispatch

A. Candidate-Specific Travel and Slack

The original slack equation depends on a donor-to-receiver distance, although the receiver is selected later in the dispatch sequence. The proposed procedure resolves this dependency by evaluating candidate combinations before ranking. For listing i, volunteer j, receiver k, and decision time t, let travel duration include the volunteer-to-donor leg, the donor-to-receiver leg, and expected handling time.

$$d(i,j,k,t) = a(j,i,t) + b(i,k,t) + h(i,j,k)$$

$$L(i,j,k,t) = e(i) - t - d(i,j,k,t)$$

All durations must use the same unit, such as hours. The quantities a and b represent travel-time estimates rather than straight-line distances. A candidate is timing-feasible only when L is nonnegative. Receiver opening hours, vehicle

All rights reserved by www.ijsrd.com

capacity, category acceptance, and available receiving capacity must also be satisfied. These are hard constraints, not additional urgency weights.

B. Proposed Assignment Sequence

At each dispatch event, the server first reads an availability snapshot and removes cancelled, expired, already reserved, or safety-rejected listings. It then constructs compatible volunteer-listing-receiver combinations and evaluates travel feasibility. For each feasible combination, it calculates the original urgency score using the combination-specific slack. The highest-scoring combination is considered first.

For equal scores, a deterministic tie-break rule uses earlier expiry, then shorter delivery duration, then the listing identifier. The last rule ensures repeatable tests. Before confirmation, the server checks the selected records again inside a transaction. If the candidate is no longer available, it discards that candidate and evaluates the next one. Notifications are sent only after the reservation is committed.

C. Computational Scope

With n listings, m free volunteers, and r receivers, exhaustive candidate evaluation can inspect up to nmr combinations. The earlier linear-time statement therefore applies only when volunteer and receiver selection are fixed or precomputed. Category filters, capacity filters, and spatial candidate searches can reduce practical work, but their effect should be measured. A city-scale deployment must report dispatch latency under load rather than assume that its workload is always small.

X. Proposed Safety and Reliability Updates

A. Safety Gate Before Priority Scoring

A proposed safety gate records preparation time, declared use-by time, packaging condition, storage declaration, and the donor responsible for the information. Missing or inconsistent records place the listing in a review state rather than automatically inferring an expiry from its category. Where information is uncertain, an administrator or authorized food-safety operator must determine whether redistribution is appropriate.

The gate is intended to support, not replace, applicable food-safety procedures. FSSAI requires appropriate handling and transport, safe surplus food, and distribution within shelf life [4]. Accordingly, a volunteer must not be encouraged to collect a listing merely because its score is high. An unsafe or unverifiable listing remains excluded even when a receiver is nearby and the modeled delivery time is short.

B. Uncertainty and Consumption Allowance

The original model considers arrival before expiry. The proposed update also reserves time for receiver handling and distribution to beneficiaries. Let u represent a travel uncertainty allowance and c represent receiver processing and consumption allowance. A conservative planning slack is defined below; neither allowance is a validated food-safety threshold.

$$Lsafe(i,j,k,t) = e(i) - t - d(i,j,k,t) - u - c$$

The dispatch service may require Lsafe to be nonnegative. Values for u and c must be documented and tested in sensitivity analysis. Where no reliable estimates exist, the system should flag the limitation instead of presenting a precise guarantee. A later implementation may estimate travel uncertainty from historical trips, but such a model requires calibration and separate validation before operational use.

C. Pickup and Receipt Verification

Pickup confirmation should record time, quantity accepted, packaging observations, and any deviation from the listing. Receipt confirmation should record arrival time, accepted quantity, rejected quantity, and a reason code. Receiver acceptance is evidence of a handover, not laboratory confirmation of food safety. Suspected contamination or an exceeded use-by limit triggers rejection and documented disposal guidance rather than reassignment to another receiver.

XI. Transactional Workflow and Data Design

A. Reservation and Concurrent Requests

The proposed lifecycle adds reserved, cancelled, rejected, and failed states to the original operational states. A reservation has a short configurable acknowledgement deadline. A database transaction must ensure that only one active delivery assignment owns a listing. The same transaction reserves receiver capacity and marks the volunteer unavailable. A failed transaction leaves all three resources unchanged.

An acknowledgement timeout can release an uncollected listing for reconsideration. However, the service must not return a picked-up listing to the donor queue: physical custody has already changed. A post-pickup failure requires an incident workflow with the current custodian, remaining usable time, and an authorized resolution. This distinction prevents the system from dispatching a second volunteer to food that is no longer at the pickup site. *B. Additional Records and Constraints*

The proposed database extension introduces Reservations, SafetyChecks, and DeliveryEvents. A reservation stores listing, receiver, volunteer, creation time, acknowledgement deadline, and status. SafetyChecks store the declared evidence and review outcome without claiming automatic certification. DeliveryEvents record state transitions, actor identifiers, timestamps, and reason codes. Quantity fields must identify whether they represent meals, portions, or another unit; conversions must not be assumed.

For reproducibility, a DispatchDecisions record should retain the candidate score, slack, travel estimate, weight version, and exclusion reasons. Expiry timestamps should include an unambiguous timezone convention. The application may display local time while preserving consistent server-side comparisons. Client-provided timestamps require validation so that a device clock error cannot silently extend the declared usable period. *C. Idempotency and Notification Recovery*

Every assignment request should carry a unique idempotency key. Repeated submissions with the same key return the existing result instead of creating another delivery. A transactional notification queue can retain committed alerts until delivery is attempted. Notification failure must not silently undo an assignment; the application should show

All rights reserved by www.ijsrd.com

acknowledgement status and escalate unresolved contact attempts.

Role-based access limits donors to their own listings and volunteers to assigned delivery details. Receiver contact and precise pickup coordinates should be disclosed only when operationally necessary. Retention periods, access logging, and deletion procedures must be specified before a pilot. These are proposed controls, not a claim of legal compliance or completed security assessment.

XII. Reproducible Evaluation Protocol

A. Common Inputs and Fair Comparisons

The proposed simulation retains the arrival rate, category probabilities, quantity range, travel assumptions, and volunteer counts described in Section VI. Before running any policy, the evaluator generates and saves each complete scenario. All four policies receive the same arrivals, quantities, preparation ages, travel inputs, and volunteer availability for each seed. This paired design separates policy differences from differences in random workloads.

Preparation age at listing time must be specified because it determines remaining usable time. A scenario in which every listing is newly prepared may underrepresent delayed donation. The experiment should therefore include a documented preparation-age distribution and a separate delayed-posting stress test. A listing cannot be assigned a positive expiry solely because it has just been posted.

B. Event and End-of-Run Rules

The simulator processes listing arrival, volunteer release, reservation timeout, pickup, delivery, and expiry events. The order of simultaneous events must be fixed and reported. A delivery completed exactly at the expiry boundary requires an explicit convention; operational allowances should avoid relying on that boundary. Scenarios should distinguish the end of new arrivals from the end of dispatch and the final observation time.

The stated hour-14 dispatch cutoff means that listings may remain undelivered even if their expiry is later. Such listings should be reported as unserved at cutoff, not automatically classified as spoiled. Follow-up observation can determine later expiry, but its duration must be identical across policies. Volunteer location after delivery must also be modeled consistently, because the next pickup leg depends on where the preceding trip ended.

C. Outcomes and Statistical Reporting

Meal rescue rate = 100 x delivered meals / offered meals

Only meals delivered within the defined usable period enter the numerator. Cooked-meal rescue uses offered cooked meals as its denominator. Additional outcomes include total rescued meals, expired meals, unserved-at-cutoff meals, median waiting time, volunteer busy time, and receiver rejection frequency. A claim of improved performance must identify the particular outcome and volunteer condition.

For each comparison, calculate the policy difference within each seed and report its mean and a 95% confidence interval using a documented method. Preserve per-seed records rather than reporting only an overall average. If simulation is later completed, the paper should distinguish implemented behavior from the additional features proposed in this extension.

XIII. Validation, Pilot Plan and Limitations

A. Sensitivity and Ablation Experiments

The proposed sensitivity study varies preparation age, travel delay, volunteer availability, receiver capacity, and the uncertainty allowance. Weight experiments should keep nonnegative weights summing to one and include multiple settings around the illustrative values. An ablation removes the perishability term, another removes quantity preference, and a third replaces urgency ranking with expiry ordering. These comparisons test which component contributes to an observed effect.

Training or tuning scenarios must be separate from final evaluation scenarios. The final report should identify parameter ranges before examining test outcomes. A policy may rescue more cooked meals while rescuing fewer total meals; that trade-off must be reported rather than hidden in one combined score. No particular result is assumed in advance.

B. Functional and Failure Tests

Required tests include expired food exclusion, duplicate reservation attempts, insufficient receiver capacity, volunteer cancellation before pickup, failure after pickup, missing preparation time, and repeated notification delivery. Boundary cases include zero slack, equal scores, unavailable receivers, and a listing whose expiry changes during reservation. Each test specifies input state, expected transition, and the audit record required to explain the outcome.

C. Proposed Local Pilot

A small pilot with consenting donors, receivers, and trained volunteers should begin in shadow mode. In shadow mode, FRUMS records recommendations while existing coordinators retain operational authority. Disagreements are reviewed to identify missing constraints and unrealistic travel estimates. Only after these checks should an authorized team consider supervised dispatch. Participation does not remove normal food-safety responsibilities.

Pilot records should distinguish meals offered, collected, accepted, and distributed. Delivery counts alone do not establish nutritional benefits, avoided emissions, or reduced hunger; those outcomes require additional measurements.

D. Originality and Remaining Work

The added wording was prepared for this extension, not copied from another paper. Established scheduling and database methods are not claimed as novel. Existing sources remain acknowledged. Author review, reference verification, and an independent similarity check are still required; no absolute copyright-clearance guarantee is asserted.

The next research milestone is an executable, tested prototype with reproducible scenarios and transparent results. Until then, this six-page manuscript presents a design and evaluation plan, not a validated operational system.

All rights reserved by www.ijsrd.com

XIV. Conclusion and Future Work

This paper presents FRUMS, a proposed platform for redistributing surplus food using an urgency-based priority score. The system aims to prioritize highly perishable food and coordinate donors, receivers, and volunteers. Future work includes implementing and evaluating the system, conducting simulation experiments, integrating travel-time estimates, and testing the platform with local restaurants and non-governmental organizations.

Acknowledgment

The authors thank their institute and the mentors who provided guidance and support throughout this work.

References

[1] United Nations Environment Programme, *Food Waste Index Report 2024*. Nairobi, Kenya: UNEP, 2024.
[2] FAO, IFAD, UNICEF, WFP, and WHO, *The State of Food Security and Nutrition in the World 2023*. Rome, Italy: FAO, 2023.
[3] J. Gustavsson, C. Cederberg, U. Sonesson, R. van Otterdijk, and A. Meybeck, *Global Food Losses and Food Waste: Extent, Causes and Prevention*. Rome, Italy: FAO, 2011.
[4] Food Safety and Standards Authority of India, *Food Safety and Standards (Recovery and Distribution of Surplus Food) Regulations, 2019*. Government of India, 2019.
[5] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard-real-time environment," *Journal of the ACM*, vol. 20, no. 1, pp. 46–61, 1973.
[6] D. Gale and L. S. Shapley, "College admissions and the stability of marriage," *The American Mathematical Monthly*, vol. 69, no. 1, pp. 9–15, 1962.
[7] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, nos. 1–2, pp. 83–97, 1955.
[8] T. L. Saaty, "A scaling method for priorities in hierarchical structures," *Journal of Mathematical Psychology*, vol. 15, no. 3, pp. 234–281, 1977.
[9] R. T. Fielding, "Architectural styles and the design of network-based software architectures," Ph.D. dissertation, University of California, Irvine, CA, USA, 2000.

All rights reserved by *www.ijsrd.com*