Software engineering-1 project

A gift-culture app project

Part one of the report:

1-Introduction:

- **Purpose:** This application encourages a community where people can give away or request items freely, without money, fostering sharing and sustainability, encouraging sustainability and reducing waste. In this way, we reduce consumption, and as a result, our planet becomes cleaner, and we reduce global warming.
- **Project Scope**: The Gift-Economy App will be a community-based web application where users can:
  - Sign up, log in, and manage profiles.
  - Join and search a local group based on their town or neighbourhoods.
  - Share the interests of the members.
  - Post items they wish to give away ("gifts").
  - Browse available gifts and request items they need.
  - Communicate with other users to coordinate exchanges.

- Report inappropriate posts or users (give feedback or review).
- View history of given/received gifts.
- Admin users will be able to manage users and content.
- Donate any items or exchange them with others freely within the community.
- Prohibiting any monetary payment and relying solely on expressions of gratitude.
- Restricting people from different geographic areas from being together or joining the groups.
- Restricting individuals from chatting after the exchange has been made.

Glossary and Abbreviations:

- Gift Post: a public offer of an item or service shared for free within a local group.
- Local Group: a small community-based group (e.g., neighborhood or district) where users can exchange items and services.
- Admin (Group Admin): a user assigned to manage a local group. They approve or reject members, moderate posts, and enforce community rules.
- Report: an action a user takes to flag a post or comment that violates the platform's guidelines.
- Badge: a symbolic token of appreciation, such as a "Thank You" badge, given to other users.
- Session Timeout: a security feature that logs users out automatically after a period of inactivity.
- Accessibility: Design practices that ensure the app is usable by people with disabilities (e.g., screen reader support, high contrast mode).
- Post Moderation: The process of reviewing and controlling user-generated content to maintain community safety and quality.

- REST API: a standard for backend communication using HTTP requests to perform CRUD operations.
- UI / UX (User Interface / User Experience): The design and behavior of the app that affect how users interact with and experience the system.

- The stakeholders:
  - **End Users**: Individuals using the platform to give away or request free items/services, they interact with the core features like posting, searching, requesting, and messaging.
  - **Group Admins**: volunteers or assigned users who manage local groups' responsible for approving new members, moderating posts, and enforcing community rules.
  - **System Administrators (Developers)**: responsible for maintaining the backend, fixing bugs, and ensuring security, manage system-level features, backups, and configuration.
  - **Future Contributors**: developers or maintainers who might extend or improve the system post-course (e.g., adding new features or localization).
- References:

  Buy nothing website and the donation centers

2-**Functional Requirements** (contains the natural language of functional requirements):

1: The system shall allow users to **register and create a profile.**

- **The reason:** to gain the purpose of the website and buy nothing, the user must create an account.

1.1) for the first time using the system, creating a new profile is a must.

1.2) the profile can be created using new account or with Gmail or Facebook.

2: The system shall allow the user to login easily after creating an account.

- **The reason**: there is no need to register again after creating an account.

2.1) The system shows the user a search bar and the world map.

2.2) the user searches for the location by writing the address.

2.3) The location appears on the map and the user confirms the location to finish.

3: The system shall ask the user to **set their location on the map to join.**

- **The reason**: to help the user to see the items and services that are close to him to get to it easily.

4: The system shall allow users to **join a local group** based on their town or neighborhood.

- **The reason**: if the user wants to join a group close to him/her and find posts about items he/she interests in it.

4.1) The system directly finds the user's neighbors and shows their posts on the main page.

5: The system shall **ask the user their interests** after he/she created the account as an extra information.

- **The reason**: to know about the interests of the user and find them in his/her town.

5.1) The system shows a window that asks the user to choose the items categories like (clothes, furniture, sports items, etc.) they are interested in.

5.2) The user can choose one or more choice form the list.

6: The system shall allow users **to search for local groups** using location keywords like (countries, cities or governorates).

- **The reason:** if the user wants to join a specific group in its local area.

7: The system should **provide a "Find a Group" page** with links to all existing groups.

- **The reason:** the system provides this page to list the all groups, if the user wants to see all the groups not a specific group.

7.1) The page has a search bar where the user writes the city or the nearest place to their location.

7.2) When the user presses enter, the system shows the platforms' links on the screen.

7.3) when there are no links for the location entered by the user, a (create your group) button appears.

8: The system should let users **ask to start a new group** if there isn't one in their area. They must also agree to help manage it as an admin.

- **The reason:** If the user wants to create a group to show specific items or services or to create a group more closer to its town.

9: The system shall allow users **to post items** (not sell).

- **The reason**: to show the items that the user wants to give it away.

9.1) The user must be able to choose the kind of post (Give, Ask)

9.2) Uploading an image must be an existing feature in a post.

9.3) the post can be only text or text and image.

10: The system should enable the user **to make the post either local or global** to be visible to many people.

- **The reason**: the user has the right to show the post to people that close to him/her or to the whole world.

11: The system shall allow users **to browse posts in their local group.**

- **The reason:** to show to the user commenlly the posts that is in his/her groups.

12: The system shall allow users **to comment on a post** to show interest.

- **The reason**: if the user wants to take that item , the giver should know by receiver's comment.

13: The system shall allow users to **message other users after commenting.**

- **The reason**: if there is an intention or availability to give the item to the receiver, so they should message together to complete the action.

13.1) After commenting, an icon appears for the user who posted the post besides the comment to start a chat between them.

14: The system should let the **user choose a way for giving or receiving** an item through one way (meeting in a place, meet at one's house or shipping service).

- **The reason:** in the system after messaging, they should choose a way of the receiving and let the system know the way to helping them to do it.

15: The system shall **close the chat** after the item is delivered.

- **The reason**: after receiving, the chat is closed as the action is completed as it's not a chatting app.

16: The system shall allow users to **offer services**.

- **The reason**: the system allows users to offer services as yoga or haircut, ect.

17: The system should let users **create posts** about local community events in their group.

- **The reason**: the user should not only create posts for giving away items or services, but also to make people know about events.

18: The system can let users **share items** that they want to lend (not give away), like a tool or a book under lending posts type.

- **The reason**: if the user wants to lend the item not give it away , there is a type of posts that called lending posts.

19: The system should allow users to **receive notifications** for new posts or updates in their local group based on selected preferences.

- **The reason**: to inform the user by all the updates in his/her groups, preventing losing the item his/her wants.

20: The system shall **enforce rules** such as no buying/selling, and prevent users from messaging unless permitted.

- **The reason**: the purpose of the website is giving away not paying money, so there are rules that the users must follow.

21: The system shall give group admins (only) tools to **remove bad posts**, manage members, and keep the group safe.

- **The reason**: group's admin has the permission to remove posts and members for keeping the group safe and preventing losing the website's purpose.

  21.1) The group admin shall be able to remove posts and comments.

  21.2) The admin shall be able to block users with bad behavior.

22: The system shall allow users to **report inappropriate posts or behavior.**

- **The reason**: if there are bad posts that group's admin didn't see, so the users can report them to warn the admins to remove them.

22.1) The system directs reports from users to admins.

22.2) Admins review reports and take the action needed.

23: The system shall show the **main platform rules** and any extra rules written by the local group.

- **The reason**: beside the rules of the no buying and selling, there are rules for every group that are enforced by admin must be shown in the group.

23.1) The system shall has a link with the title "rules" which directs the user to a page where all rules and policies are written.

23.2) the rules page shall be updated when an admin set a new rule.

24: The system shall provide a **customer service** for helping.

- **The reason**: if there is a problem in the website, the user can call or chat with the customer service for helping.

24.1) The system shall show customer service numbers, Facebook accounts for managers, and an Ai model helper as an extra feature.

24.2) customer service accounts and chats shall be available

at working hours from 8 am till 4pm according to Egypt.

25: The system shall send **reminders or notifications** to group admins to monitor activity.

- **The reason**: if the users report a post or something, there are notifications for the admin to warn them and remove posts.

25.1) when a report is done by a user, the system shall notify admins with the report, the bad post or behavior and the number of reports on the same post or behavior.

25.2) The notification is sent every two days if no action (discard the report, block user, remove post or user warning) has been taken.

26: The system should let users **post Thank you messages** to show appreciation for items or services they received.

- **The reason**: after receiving and item or service, it's a good thing to create type of posts called Thank you messages to thank the giver.

26.1) The post can have a gratitude rate (4 stars)which will be collected for business reports.

27: The system should let **show the number of members and gifts** per a year at the Home page.

- **The reason**: For the whole users, it's good to inform them the total numbers of members and gifts to encourage them to make posts.

28: The system shall show a **list of free local resources** (like food banks or community help centers) and let users add to it.

- **The reason**: if the user doesn't want to make post for giving away , he/she can give away in free local resources that are listed in the website.

28.1) the system shall show the bussines partners from food banks and charities on the home page.

28.2) the system shall have a community where users can participate at events like donating or volunteering with these authorities.

29: The system should make **a report with the number of exchanges** at the end of each month.

- **The reason**: for the mangers of the whole website, the system should make a report for them to know the total of receiving process to evaluate the progress of the website purpose.

29.1) the report must have the number of gifts from giving posts and asking posts.

29.2) the report is done at the last day of each month at 11:59 Pm.

29.3) the report can be accessed by managers only.

30: The system should **let users choose their preferred language** to use the website.

30.1) English and Arabic language is a must.

30.2) other languages are optional.

30.3) English language is the default one.

31: The system should let the user say **feedback** on the system.

- **The reason**: feedback helps the developers of the website to know the problems and bugs in the website.

31.1) once a user becomes a volunteer, the system shows a window with a button 'join volunteers community'

31.2) when the volunteer presses join, they are directed to the facebook group for volunteering.

32: The system should **allow the volunteer user to have a special community** to communicate with other volunteers on the website or on any other platform.

- **The reason**: another type of users called volunteers, they collect the items that no one receives or wants it, and try to find receivers in another groups.

33: The system shall **allow the shipping clerk to connect with the sender, receiver** or both as three members chat in case of choosing shipping option.

- **The reason**: if the sender and receiver chose shipping service as a giving way, so the system shall connect shipping clerk to connect with receiver and sender.

34: The system should allow the sender **to choose the way of paying shipping fees** (credit, cash).

- **The reason**: before shipping, the shipping clerk should take the fees from the sender by any of this ways that the sender chooses.

35: The system shall ask volunteers **to agree on sharing advertisements** to show people the purpose of the website.

- **The reason**: volunteers' job is also to advertise for the giving up idea , so the system should offer to them the idea of advertising and the platform to advertise in it.

35.1) Every user profile has a 'saved posts' partition that contains all posts a user saves.

35.2) All posts shall have 'save post' feature available for all users.

36: The system should allow users **to save or bookmark posts of interest** to view or respond to them later.

- **The reason**: if the user liked an item but doesn't need to receive it now, he/she can find it in saved posts corner.

36.1) At the end of each month (28, 30 or 31) on 11:59, the system shall generate a report (pdf file) from the data base.

36.2) The report must contain volunteer id, name , age , date of joining, and location.

37: The system should **make a report for managers only** with the number of new volunteers each month.

- **The reason**: it's good for the managers to know the number of volunteers, as when the number increases, the advertisements increase.

37.1) The system should have a block on the main page of the website with a green background that shows the total number of members, total gifts exchanged, and total number of volunteers.

37.2) this info should be updated each 3 months

3- **Non-functional Requirements** (contains the natural language of non-functional requirements):

Look-and-Feel Requirements.

The system is visually comfortable and will use styles that are suitable for all types of users.

Requirement:

1-The system shall have a clean and intuitive user interface (UI).

b) Specification:

The system will have a clean and intuitive UI with clear fonts and soft colors for an easy and pleasant experience.

c) Fit Criterion:

85% of test users must complete key tasks (e.g., posting a gift) without help within 3 minutes in usability tests.

d) Architectural Impact:

Influences the frontend structure and use of clear, user-friendly interfaces.

Usability & Humanity Requirements.

Users can easily interact with the system and quickly become familiar with it because of its simplicity.

Requirement:

2- The system shall support multiple languages.

b) Specification:

UI available in English, Arabic, and French. Users can select language in settings.

c) Fit Criterion:

Users can select from at least three languages in settings, and UI displays content correctly in each selected language.

d) Architectural Impact:

Requires internationalization (i18n), dynamic content loading, and localization files.

Requirement:

3- The system shall offer a dark mode.

b) Specification:

The system should offer a dark mode option to reduce eye strain and enhance user experiences.

c) Fit Criterion:

A toggle switch in settings enable dark mode. The UI theme changes without errors, and user preference is saved.

d) Architectural Impact:

Separation of styling using CSS variables or themes. May influence frontend component design.

Performance Requirements.

The system performs efficiently, providing fast responses to user commands, handling high loads effectively, and maintaining a suitable storage footprint.

Requirement:

4- The system shall be available 99.9% of the time.

b) Specification:

The website will be highly available by using backup servers and load balancing techniques to ensure 99.9% uptime.

c) Fit Criterion:

The platform must be online at least 99.9% per month. Downtime should not exceed 43 minutes monthly, measured by uptime monitoring tools.

d) Architectural Impact:

Requires redundant servers and failover mechanisms to reduce downtime.

Requirement:

5-The system shall respond to user actions within 2 seconds.

b) Specification:

The system will be fast by optimizing code, compressing files, and minimizing delays to respond within 2 seconds.

c) Fit Criterion:

95% of user actions must load within 2 seconds during normal use, measured by browser dev tools or performance tests.

d) Architectural Impact:

Affects the use of caching, load balancing, and efficient database design.

Requirement:

6-The system shall be able to handle at least 10,000 concurrent users.

b) Specification:

The system will support at least 10,000 concurrent users by using scalable server architecture and load distribution.

c) Fit Criterion:

The platform must support 10,000 concurrent users without crashes or significant slowdowns, verified through load testing.

d) Architectural Impact:

Requires horizontally scalable server architecture and effective load distribution mechanisms.

Requirement:

7- All important actions must be logged and time-stamped for audit and traceability.

b) Specification:

Actions like deleting posts or banning users logged with timestamps.

c) Fit Criterion:

Action logs are automatically created in the system database with user ID, action, and timestamp. Verified in log files.

d) Architectural Impact:

Requires a logging service or system that supports immutable logs and audit trails.

Operational & Environmental Requirements.

The system runs efficiently on different operating systems and devices, ensuring consistent performance in diverse technical environments.

Requirement:

8- The system shall be mobile-friendly and work on all modern browsers.

b) Specification:

The system will be mobile-friendly and compatible with modern browsers using responsive design techniques.

c) Fit Criterion:

The website must work properly on all major browsers and common mobile screen sizes, tested via responsive design checks.

d) Architectural Impact:

Requires responsive frontend design and cross-platform compatibility.

Maintainability & Support Requirements.

The system is designed to be easily maintainable, enabling efficient updates and prompt resolution of any potential issues.

Requirement:

9- The system must perform full automatic backups every 24 hours.

b) Specification:

Backup jobs scheduled daily, to ensure data is not lost in case of failure.

c) Fit Criterion:

A scheduled backup job runs every 24 hours and generates a backup log file with success status.

d) Architectural Impact:

Requires automated task scheduling (e.g., CRON), backup storage, and disaster recovery plans.

Requirement:

10-The system code shall be modular.

b) Specification:

Code organized in independent modules for easier enhancement (e.g., gamification).

c) Fit Criterion:

Verified via code reviews; new features can be added without modifying core logic.

d) Architectural Impact:

Promotes modular or microservices architecture with clear APIs and separation of concerns.

Cultural Requirements.

The system is designed to be culturally inclusive, supporting a wide range of languages and catering to diverse cultural preferences.

Requirement:

11- The system shall support localization for different towns and neighborhoods.

b) Specification:

Localization will be implemented to display content in the local language and format based on the user's region.

c) Fit Criterion:

The system must support item listings by location and allow adding new towns/neighborhoods without affecting system performance.

d) Architectural Impact:

Calls for modular architecture and possible use of micro services.

Legal Requirements.

The system complies with all relevant laws and regulations, including data protection.

Requirement:

12- The system must comply with data protection regulations.

b) Specification:

Data is encrypted; no sharing without consent; users can request data deletion.

c) Fit Criterion:

Personal data only stored with consent; deletions occur within 72 hours.

d) Architectural Impact:

Architecture must enforce strict data access control, encryption at rest and in transit, and data anonymization.

Security Requirements.

The system Focus on protecting user data, preventing unauthorized access, and ensuring privacy.

Requirement:

13- The system shall protect user data and communications.

b) Specification:

All sensitive data encrypted; user messages are end-to-end encrypted.

c) Fit Criterion:

TLS used for message transfer, all personal data must be encrypted and stored securely. Compliance with local privacy laws (e.g., GDPR) must be confirmed.

d) Architectural Impact:

Leads to implementing secure communication (HTTPS), encryption methods, and access control.

Requirement:

14- Users must be logged out after 15 minutes of inactivity.

b) Specification:

Inactive sessions are automatically ended.

c) Fit Criterion:

After 15 minutes of no input, the session ends, and the user is redirected to the login screen.

d) Architectural Impact:

Requires session management, idle timer detection on frontend, trigger logout events, and token expiration on backend.

4- **Design & Implementation Constraints:**

1. **Platform Constraint:**
   The system must be accessible from multiple devices, including mobile phones, laptops, and PC. Therefore, the design should be fully responsive and compatible across different screen sizes and input methods (touch and keyboard/mouse).

2. **Technology Constraint:**
   The application will be developed using HTML, CSS (Bootstrap), JavaScript for the front-

end, and a SQL- based relational database (such as MySQL or PostgreSQL) for backend data storage and management.

3. **Geographical Limitation:**

   Users must only be able to interact with other users in their **local area** or **community**. This requires implementing location-based filtering and regional access controls.

4. **No Monetary Transactions:**
   The application must strictly **prohibit any kind of financial exchange**. All item exchanges must be done voluntarily and with gratitude only. This constraint influences the business logic by removing pricing, bidding, or payment options.

5. **Post-Exchange Communication Restriction:**
   After a successful exchange of an item, users must **no longer be able to chat** with each other. This constraint will be enforced by automatically disabling or hiding chat threads after the exchange is marked as complete.

6. **Budget Constraint:**
   The project must use **free and open-source technologies only**, as it is developed in an academic setting with no budget allocated.

7. **User Simplicity Constraint:**
   The app targets **non-technical users**, so the interface must be clean, intuitive, and extremely user-friendly, minimizing the need for training or instructions.

5- System Evolution:

- Anticipated changes:
  - Increase in the number of users and expansion of Local Groups and Communities over time.

- Changes in how users interact with and what they expect from the system.
- Includes features such as online payment, renting, or borrowing.
- Technological Advancements.

- How should any anticipated changes in the future affect the system design?

  - Scalability and Load Handling: The system should be designed to easily handle growing numbers of users, ensuring seamless user experience under high traffic conditions.
  - Flexible and Customizable User Interface: The system should support a flexible UI that can be easily updated.
  - Integration with New Technologies and APIs: The platform should be designed to integrate easily with new technologies such as AI.
  - Integration of Electronic Payment System: The system should be prepared to integrate an electronic payment feature in the future, enabling users to make transactions securely for gifting, borrowing, or renting services.

6- **What are the requirements discovery approaches that you'll rely on?**

- **Brainstorming Sessions**:
  Example of that: The team discussed the whole project from the description of it, trying to configure all the requirements that is written in the description besides the requirements that we created after deep thinking to make the website different and complete.
- **Observation of Similar Platforms:**
  Example of that: We observed another platform which is close to our website (BuyNothingProject.org) , we tried to know the main titles and ideas in it and added to it many different ideas that we got from brainstorming sessions.

7- **What requirements validation techniques will you employ/use? (Give examples) ?**

1. **Team-Based Walkthroughs**: We will conduct regular walkthrough sessions within our team, where each requirement (functional and non-functional) is reviewed out loud and discussed.

This will help:

- Identify vague or duplicated requirements.
- Ensure everyone has a shared understanding of what is expected.

Example:

While reviewing the requirement “Users can request posts,” we might discuss:

– What if two users request the same post?

– Can a user cancel the request?

– How will the post owner choose who gets the item?

2. **Low-Fidelity Prototyping**: We will create simple wireframes or HTML mock ups of the main user interfaces (such as the home page, create post page, and join group page). These mock ups will be shared with classmates and potential users to gather feedback before development.

Example: A user prototype showing the “Create a Post” screen will be tested to ensure that the fields for item name, description, and pickup location are clear and easy to fill out.

3. **Writing Use Cases and Basic Test Scenarios**: Before implementation, we will write basic use cases and test scenarios to check whether each requirement can actually be tested and implemented.

Example:

For the requirement “users receive notifications,” we will ask: What triggers the notification? Is it real-time? Is it via email or in-app?

4. **Walkthroughs of Use Cases**: The team will conduct step-by-step reviews of each use case scenario to check for logical consistency and coverage of success and failure paths.

Example: For the use case “Request a Gift Post,” we will verify what happens if multiple users request the same item, and whether the original poster can choose who to give it to.

5. **Checklist-Based Review**: We will validate the requirements against a checklist derived from course guidelines and the template (e.g., Have all user roles been considered? Are all main use cases covered? Is each non-functional requirement testable?).

6. Traceability Matrix: A simple traceability matrix will be created to ensure each requirement is mapped to:

- At least one use case
- A design element
- A test case

Example:

Requirement: “User must be able to report a post”

Mapped to:

- UC8
- moderation logic in the backend.
- validation via test case TC08.

These techniques will help catch missing, conflicting, or ambiguous requirements early, reduce rework, and ensure that the final system matches both user expectations and technical constraints.

Structured Natural Language.

1. Structured Specification (Functional Requirement)

For defining core functions like "Post a Gift" or "Request an Item":

Field	Description
Function	Post a Gift
Description	Allows users to list an item they wish to give away.
Inputs	Item details (title, description, condition, location, image).
Source	User input via mobile app/web form.
Outputs	Published gift post visible to local community.
Destination	BuyNothing platform database and community feed.
Pre-condition	User must be logged in and located in a participating community.
Post-condition	Item is listed; user receives confirmation.
Side Effects	None.

2. Tabular Specification (Decision Logic)

For rules like "Item Approval Criteria":

Condition	Action
Item description contains prohibited words.	Auto-flag for moderator review.
User is new (joined <24 hours).	Hold post for manual approval.
Item is legal/safe and follows guidelines.	Publish immediately to community feed.

1-functinal requirements:

1-User Registration (Structured Specification)

Function	User Registration
Description	Allows new users to create an account on the platform by taking the user information.
Inputs	Email, password, username.
Source	User input via registration form

Outputs	New user account created in database.
Destination	User database
Action	If this is the user's first time on the website, they will need to register an account by entering their email, password, and then choosing a username.
Requirement	-That this is the user's first time. -User must consent to location sharing
Pre-condition	No existing account with same email.
Post-condition	User profile created; user can log in.
Side effects	Creates default privacy settings

2. User Login (Structured Specification)

Function	User Login
Description	It allows the user to log into their account by entering their email or username and password.
Inputs	Email/username and password.

Source	User input via login form.
Outputs	Access to user account home page.
Destination	session management system
Action	If the user already has an account, they can log in by entering either their email or username along with their password.
Requirement	Matching the sent data with the stored data
Pre-condition	Valid registered account.
Post-condition	User session initiated.
Side effects	Updates last login time

3. Location Setup (Tabular Specification)

Condition	**Action**
User completes registration	System prompts to set location on map/GPS
User skips location setup	Restrict group access until location set

4. Join Local Group (Structured Specification)

Function	Join Local Group
Description	Automatically adds user to group based on location.
Inputs	User's location coordinates.
Source	User manual location ظGPS
Outputs	Membership in local (BuyNothing) group.
Destination	Group membership database
Action	When the user selects their location, they are added to group of people nearby within the same area so they can post and interact with them.
Requirement	Group must exist in the area and user must consent to location sharing
Pre-condition	That the GPS has matched the city or town name entered, as they are nearby
Post-condition	User is active member of local group. Group posts become visible. User appears in member list.
Side effects	- Updates group member count

5. User Interests (Tabular Specification)

Condition	Action
After account creation	Show optional interest selection screen
User selects interests	The system saves the user's preferences so it can later suggest posts, groups, or items that match their interests and needs.
User skips	Proceed without interest data

6. Search Groups (Structured Specification)

Function	Search Local Groups
Description	Enables finding groups by location keywords.
Inputs	Text input (city/town names).
Source	User search query and location database
Outputs	List of matching groups.
Destination	Search results page (web/mobile view)

Action	When a user wants to find local groups, they simply type in the name of their city or town, and the system will show them a list of nearby groups they can join
Requirement	location services enabled and valid groups exist in search area
Pre-condition	That the GPS has matched the city or town name entered, as they are nearby
Post-condition	Search history recorded. Cached results for faster future searches.
Side effects	none

7. Find Group Page (Structured Specification)

Function	Find Group Directory
Description	Displays all existing groups with join links.
Inputs	User location (GPS coordinates)
Source	Group database and user preferences (if logged in)
Outputs	Paginated list of groups with locations.
Destination	group discovery interface.

Action	When the group search is completed, the groups are displayed in a way that allows the user to choose the one they want based on their needs and interests.
Requirement	Accurate group location data
Pre-condition	That the GPS has matched the city or town name entered, as they are nearby.
Post-condition	Join selected groups & share group links
Side effects	none

8. Create New Group (Tabular Specification)

Condition	**Action**
No local group exists in this space of aera	Show "Start New Group" option
User accepts admin role	Create group after verification
User declines admin role	Prevent group creation

9. Post Items (Structured Specification)

Function	Create Giveaway Post
Description	the user wants to post in order to offer a gift in spacific group.
Inputs	Title, description, photos, condition.
Source	- User input form - Device camera/gallery
outputs	Published post visible in: - Group feed - User profile - Search results
Destination	Group posts database
Action	The user targets the appropriate group and then publishes their post.
Requirement	- Title must contain item type - At least 1 clear photo required - No prohibited items (per group rules) - Must specify holding period (1-7 days)
Pre-condition	Must be group member.

Post-condition	Post appears in group feed & user's "active posts" counter increments
Side effects	Updates group activity metrics

10. Post Visibility (Tabular Specification)

Condition	Action
User selects "Local"	Show only in user's nearby group.
User selects "Global"	Show in all nearby groups.

11. Browse Posts (Structured Specification)

Function	View Local Posts
Description	Shows available items in user's group.
Inputs	User location &that they are a member of the group.
Source	Group posts database.

Outputs	Filterable list of posts.
Destination	User's feed view (mobile app/web)
Action	Retrieves posts from joined groups and renders optimized feed (paginated)then updates in real-time for new posts.
Requirement	Posts must be active (not expired) & accurate location data for distance sorting
Pre-condition	User belongs to >= 1 group & location permissions granted
Post-condition	User can: View full post details
Side effects	none

12. Comment on Posts (Tabular Specification)

Condition	**Action**
User views post	Show comment option
User comments	Display comment publicly on post

13. Private Messaging (Structured Specification)

Function	Initiate Private Message
Description	Enables communication after commenting.
Inputs	Message content (text/attachments)
Source	Comment thread or user profile
Outputs	Private chat window opens.
Destination	Encrypted messaging database
Action	User clicks "Message" button on comment then the system verifies comment history
Requirement	Minimum 1 prior public comment & no blocked user status
Pre-condition	Must have commented on post.
Post-condition	Chat thread created.
Side effects	none

14. Close Chat After Delivery

Function	Close the chat conversation after item delivery confirmation.
Description	Automatically terminates the messaging session between two users once the item or service has been marked as delivered to avoid further communication on completed exchanges.
Inputs	Delivery confirmation (e.g., user clicks "Mark as Delivered").
Source	User input through the item post or chat interface.
Outputs	Chat thread status is updated to "Closed".
Destination	Messaging module (affecting both users in the conversation).
Action	The system checks for delivery status. Once confirmed, it disables message input fields and marks the chat as closed.
Requirements	The item or service must have a valid post and a corresponding chat thread with a delivery option available.
Pre-condition	An active chat session must exist between giver and receiver.
Post-condition	Chat is marked as closed, users can no longer send messages.
Side effects	None; chat history remains viewable for reference.

15. Offer Services

Function	Allow users to post offers for free services.
Description	Enables users to create posts offering their help or skills (e.g., tutoring, dog walking) to other members in the community.
Inputs	Service title, description, category, availability.
Source	User input via the "New Post" form with the "Offer Service" type selected.
Outputs	New service post displayed in the group feed.
Destination	Group feed under the “Services” category.
Action	The system publishes the post and makes it visible to group members based on location.
Requirements	User must be logged in and a member of the group.
Pre-condition	User is authenticated and selects the "Offer Service" post type.
Post-condition	The service post appears in the group feed.
Side effects	Post may trigger notifications for interested users.

16. Post Local Community Events

Function	Allow users to share local event announcements.
Description	Lets users post about community activities like gatherings, workshops, or clean-up events.
Inputs	Event title, date, time, description, and location.
Source	User input through event post form.

Outputs	A new event post is created and displayed.
Destination	Events section within the group feed.
Action	The system validates the data and publishes the event.
Requirements	Valid event information and group membership.
Pre-condition	User is a verified group member and selects the "Community Event" option.
Post-condition	Event appears in the feed and can be viewed by group members.
Side effects	Interested users may receive event reminders or alerts.

17. Share Lending Items

Function	Let users post items available for lending.
Description	Users can offer to temporarily lend personal items (e.g., books, tools) instead of giving them away.
Inputs	Item name, photo, description, and lending duration.
Source	User submission via "Lending" post type.
Outputs	Lending post published in group feed.
Destination	Lending section in the local group feed.
Action	The system tags the post as “Lending” and stores return terms if applicable.
Requirements	User must select “Lending” as the post type.
Pre-condition	User must be logged in and provide complete item details.
Post-condition	Post is published and available to group members.

Side effects	May prompt users to negotiate borrowing terms via chat.

18. Notifications for New Posts

Function	Notify users about new posts based on preferences.
Description	Sends alerts when new posts match a user's interests or categories they follow.
Inputs	New post data, user notification settings.
Source	New post created by another user.
Outputs	Notification (email, push, or in-app alert).
Destination	User's notification center or linked email.
Action	The system filters new posts and checks against user preferences.
Requirements	User must have enabled notifications.
Pre-condition	User preferences are set in the system.
Post-condition	User receives alert with a link to the post.
Side effects	Increased engagement with relevant posts.

19. Enforce Group Rules

Function	Enforce platform and group-specific rules.
Description	Blocks forbidden actions like selling or unauthorized messaging and shows warnings when users attempt them.
Inputs	User-generated content (posts/messages).

Source	System monitors post and chat submissions.
Outputs	Warning or block message.
Destination	User interface (form or chat box).
Action	The system checks content against predefined rules.
Requirements	Rules must be defined and stored in the system.
Pre-condition	User attempts to perform an action (e.g., post or message).
Post-condition	If action violates rules, it is blocked and a warning is shown.
Side effects	User may be temporarily restricted or flagged for admin review.

20. Admin Tools for Moderation

Function	Provide admins with moderation controls.
Description	Allows group admins to remove posts, manage members, and take actions to maintain safety and order.
Inputs	Admin actions (delete, block, approve).
Source	Admin dashboard or group tools interface.
Outputs	Post/user status updated.
Destination	Group feed, user status.
Action	The system executes admin commands in real time.
Requirements	User must have admin role.
Pre-condition	Admin is logged in and viewing group tools.

Post-condition	Post is removed or user is restricted.
Side effects	Changes visible to members; notifications may be triggered.

21. Report Inappropriate Posts

Function	Allow users to report inappropriate behavior or content.
Description	Gives users the option to flag posts or users that violate community guidelines.
Inputs	Report reason, optional comment.
Source	User clicks "Report" on a post or profile.
Outputs	Report submitted to moderators/admins.
Destination	Admin dashboard or moderation queue.
Action	The system logs the report and alerts moderators.
Requirements	Reporting feature must be active and visible.
Pre-condition	A visible post or user must exist.
Post-condition	Report is stored and available for admin review.
Side effects	May lead to post removal or user warning.

22. Display Platform and Group Rules

Function	Show general and group-specific rules to users.
Description	Makes platform guidelines and any extra local group rules easily visible to users.
Inputs	Stored rule content.

Source	System settings and group admin input.
Outputs	Rules page displayed to users.
Destination	Rules section (profile menu or group info).
Action	The system fetches and displays rules on request.
Requirements	Rules must be created and assigned.
Pre-condition	User visits the rules section or tries to post.
Post-condition	Rules are shown clearly to the user.
Side effects	User is more likely to comply with platform policies.

23. Provide Customer Service

Function	Offer customer service support to users.
Description	Users can contact support for help with issues related to posts, messages, or system use.
Inputs	Support request form (text, category, optional screenshot).
Source	User submission from "Help" or "Support" page.
Outputs	Ticket created and assigned to support staff.
Destination	Customer support system.
Action	The system stores the request and notifies the support team.
Requirements	Support channel must be integrated.
Pre-condition	User initiates a request via the app.
Post-condition	Support ticket is logged and pending response.

Side effects	Response delay may affect user satisfaction.

24. Admin Notifications for Group Activity

Function	Notify group admins about recent activity.
Description	Sends reminders or reports to admins about recent posts, flagged content, or new members to help them monitor their group.
Inputs	Group activity logs.
Source	System tracking of posts, flags, member joins.
Outputs	Notification or summary report.
Destination	Admin dashboard or inbox.
Action	System compiles activity data and sends it at set intervals.
Requirements	Group must have an assigned admin.
Pre-condition	Group has had activity in a defined time range.
Post-condition	Admin receives update and may take action.
Side effects	Encourages regular admin engagement.

25. Thank You Messages

Function	Allow users to post appreciation messages.
Description	Users can thank others publicly for items or services they received to promote kindness and gratitude.
Inputs	Thank you text, optional image.
Source	User input via “Thank You” post form.

Outputs	Public appreciation post visible to group.
Destination	Group feed under “Thanks” section.
Action	System publishes the post with a special “Thank You” badge.
Requirements	The user must have interacted in a past exchange.
Pre-condition	Completed exchange or confirmed receipt.
Post-condition	Message appears in the feed.
Side effects	Builds community trust and positive engagement.

26. Show Group Stats on Home Page

Function	Display stats about group size and gifting activity.
Description	Shows the number of active members and total gifts shared in the group over the past year.
Inputs	System analytics and logs.
Source	System-generated data.
Outputs	Stats displayed in the homepage dashboard.
Destination	Group’s Home Page.
Action	The system calculates and updates statistics periodically.
Requirements	Group must have logged data from past interactions.
Pre-condition	System has valid statistics to show.
Post-condition	Stats are updated and visible to all group members.
Side effects	Encourages participation and transparency.

27) Free Local Resource Listing and Contribution(Structured Specification).

Function	Free Local Resource Listing and Contribution.
Description	The system shall display a list of free local resources (e.g., food banks, community help centers) and allow users to contribute by adding new resources to the list. This helps users find local support and contribute to the community network.
Inputs	Existing list of free local resources from the database. User-Added Resource (Name, address, contact, type, description).
Source	Existing database of local resources. User input (form) or external API (if auto-importing).
Outputs	Displayed list of free local resources on the relevant page. Confirmation message after a new resource is successfully added. Validation messages if input is incomplete or incorrect.
Destination	Local Resources page UI (where the list and form are displayed). Database (where new resources are stored).
Action	1-When the user opens the Local Resources page, the system retrieves and displays the list of existing resources. 2-The user can fill out a form to add a new resource. 3-Upon submission: • The system validates the input fields.

	• If valid, the system stores the new resource in the database. • The new entry appears in the resource list.
Requirements	1-The system must persist submitted data to a backend database. 2-Display must update to show the latest list without requiring a page reload. 3-User-friendly UI/UX for both viewing and submitting.
Pre-condition	The user accesses the Local Resources page. The system can connect to the database to retrieve or store resource data.
post-condition	New resource appears in the list after submission. Database is updated with the new entry.
Side effects	Spam/fake entries if moderation is lacking. May increase database size over time with user submissions. Outdated information if no expiry/update mechanism exists. Duplicate entries if validation fails.

28) Monthly Exchange Report Generation (Structured Specification).

Function	Monthly Exchange Report Generation.
Description	The system shall automatically generate a report at the end of each month containing the total number of exchanges that occurred during that month. This helps track platform usage and trends over time.

Inputs	Exchange records with timestamps (date and time of each exchange). Current system date and time (for monthly trigger).
Source	Exchange records in the database. System date/time service.
Outputs	A monthly report containing: • Month and year. • Total number of exchanges. Report stored in a designated location (e.g., database table or file system).
Destination	Monthly reports table in the database or File system (CSV, PDF, etc., depending on implementation). Admin dashboard or notification service (if alerts are used).
Action	1-At the end of each month the system: • Queries the database for all exchanges that occurred within that month. • Counts the total number of exchanges. • Creates a new report entry with the results. • Stores the report data for later access. • Optionally, sends a notification or email to administrators.
Requirements	1-Accurate time-stamped records of all exchanges. 2-Automated job or scheduler to trigger the report generation.

	3-Storage mechanism for saving reports (database table or file system). 4-Secure access for viewing reports.
Pre-condition	Exchange data exists in the system. The system is running and can access time/date functions and the database.
post-condition	Report is generated and delivered to designated recipients. Database/logs record that the report was run.
Side effects	Increased database activity at the time of report generation, especially with large datasets. Disk or storage usage increases over time as reports accumulate.

29) Language Preference Selection (Tabular Specification).

Condition	**Action**
User visits website for the first time.	Display website in default language (e.g., English).
User selects a preferred language from language menu.	Reload or update the website UI to display in the selected language.
Translation file loads successfully.	Update all UI text dynamically.
Translation file fails to load.	Fallback to default language and console warning.
User is logged in and has previously saved language preference.	Automatically load website in the saved preferred language.

User selects a new language.	Update the current session's language setting and store the new preference.

30) User Feedback Submission (Structured Specification).

Function	User Feedback Submission.
Description	The system shall allow users to submit feedback about the system. Feedback can include comments, suggestions, complaints, or praises. This helps the organization improve the system based on user experiences.
Inputs	User-entered feedback text. (Optional) User's contact information (e.g., email address). (Optional) Feedback category (e.g., Bug report, Suggestion).
Source	User input (feedback text and optional fields).
Outputs	Confirmation message that feedback was received. Stored feedback record in the database or sent to an administrator for review.
Destination	Feedback storage database or admin notification system for review.
Action	1-User accesses the "Feedback" form/page. 2-User enters feedback and optional information. 3-User submits the feedback form. 4-The system: • Validates that required fields (at least the feedback text) are filled.

	• Stores the feedback in the database or forwards it to a feedback management system. • Shows a confirmation message to the user.
Requirements	1-Feedback form on the system interface. 2-Validation to ensure at least feedback text is entered. 3-Database or storage for saving feedback records. 4-Anonymous submissions allowed. 5-Admin dashboard or notification system to review feedback.
Pre-condition	User has access to feedback feature The feedback form is correctly loaded and accessible.
post-condition	Feedback is stored securely in the system. User is notified that the feedback was successfully submitted.
Side effects	Spam or malicious submissions if there is no input validation or CAPTCHA. Database growth over time as feedback accumulates, possibly affecting performance if not managed.

31) Volunteer Community for Communication (Tabular Specification).

Condition	**Action**
User is registered and verified as a volunteer.	Display access to the special volunteer community (website or external platform).

User is not registered as a volunteer.	Hide or deny access to the volunteer community features.
Volunteer user accesses the community section.	Allow the user to view, post, and respond to messages or discussions.
Volunteer user submits a message / post.	Validate and publish the message to the community.
Admin/moderator identifies inappropriate behavior or spam.	Remove the post/message and optionally issue a warning or suspend the user.
Volunteer user accepts an invitation to an external platform (e.g.,Discord)	Redirect the user to the external platform.
Volunteer user leaves or is removed from the volunteer program.	Revoke access to the volunteer community and external platform links.

32) Three-Member Chat for Shipping Coordination (Structured Specification).

Function	Three-Member Chat for Shipping Coordination.
Description	The system shall enable a three-way chat between the shipping clerk, sender, and receiver when shipping is selected, facilitating real-time coordination about delivery details, issues, or changes.
Inputs	hipping request with sender, receiver, and shipping clerk assignment. Chat messages from sender, receiver, and shipping clerk. Attachments (e.g., shipping labels, addresses, tracking info).
Source	Shipping transaction data (sender and receiver info).

	User input (messages typed by sender, receiver, or clerk).
Outputs	Real-time chat interface between sender, receiver, and shipping clerk. Stored chat history for reference. Notifications for new messages to participants.
Destination	Chat server/database (for storing and retrieving messages). Chat UI component (for displaying the conversation).
Action	1-Sender selects the shipping option during a transaction. 2-System automatically creates a private group chat including: -Sender. -Receiver. -Assigned shipping clerk. 3-All three parties can: -Send and receive messages. -Share shipping information (e.g., addresses, instructions). -Ask and answer questions in real time. 4-System logs the conversation securely for future reference.
Requirements	1-Real-time messaging system (chat server). 2-Group chat creation logic involving three specific users. 3-Notifications for new messages. 4-Access control to ensure only the assigned three members can see the conversation.

	5-Message storage and retrieval system. 6-User-friendly UI for chat (mobile and desktop compatible).
Pre-condition	Sender has chosen the shipping option during transaction setup. Sender, receiver, and clerk are registered users in the system. Clerk has permissions to initiate chats.
post-condition	A chat room is created and accessible only by the three involved members. All communication between sender, receiver, and shipping clerk is recorded securely.
Side effects	Increased server load due to real-time messaging traffic. Data storage growth over time as chat histories accumulate. Privacy concerns if access controls are not properly enforced (risk of unauthorized viewing). Potential delays in communication if users are offline (may require fallback notifications via email or SMS).

33) Choosing Shipping Payment Method (Tabular Specification).

Condition	**Action**
Sender is setting up a shipment.	Display payment method options (Credit Card, Cash).
Sender selects "Credit Card" as payment method.	Redirect to credit card payment gateway for processing.

Sender selects "Cash" as payment method.	Mark the shipment as "Cash payment on delivery" and notify the shipping clerk.
Sender does not select any payment method.	Prompt sender with a validation error message to choose a payment method.
Credit payment fails.	Show error message and allow the sender to retry or select another payment method.
Credit payment succeeds.	Confirm payment and update shipping status to "Paid".

34) Volunteer Consent for Sharing Website Advertisements (Structured Specification).

Function	Volunteer Consent for Sharing Website Advertisements.
Description	The system shall request consent from volunteers to allow the platform to share promotional advertisements (e.g., on social media or other networks) using their activity or content (with permission), in order to show the website's purpose and increase public awareness.
Inputs	Consent response (Agree / Disagree). Advertisement examples (images/text). Consent expiration period (e.g., annual re-approval).
Source	Volunteer's consent decision. Ad content database.
Outputs	Recorded consent status (agreed or declined). Message confirming choice.

	(If agreed) Volunteer content/identity made available for inclusion in ads. (If declined) No content used in advertisements.
Destination	Consent status database field. Marketing/advertising system (only for volunteers who agreed).
Action	1-When a user is marked as a volunteer, the system displays a one-time prompt requesting their consent for promotional use. 2-The volunteer selects either "Agree" or "Disagree". 3-The system stores their response in the database. 4-If agreed, the system may use appropriate user-generated content or activity for promotional materials. 5-If declined, the system ensures that no such data is used in advertising campaigns. 6-Schedules annual re-consent reminder.
Requirements	1-Consent dialog or form built into the volunteer onboarding or profile section. 2-Database field for storing volunteer's consent status. 3-Functionality to check consent status before including user data in ads. 4-Privacy and data use policy visible to the user.
Pre-condition	User must be registered and identified as a volunteer. User must be logged in. Advertisement examples exist in system.

post-condition	Volunteer’s choice regarding ad participation is stored. The system respects that choice when preparing or sharing advertisements.
Side effects	Potential trust issues if volunteers feel pressured or unclear about what is shared. Legal/data compliance risks if consent is not properly logged or if content is used without explicit permission. Reputational impact if advertisements include personal information without consent.

35) Save or Bookmark Posts (Structured Specification).

Function	Save or Bookmark Posts.
Description	The system shall allow users to save or bookmark posts of interest so they can view or respond to them later. This enhances user engagement and helps users keep track of content they care about.
Inputs	User clicks a “Save” or “Bookmark” button on a post. User's existing saved items.
Source	User action on the UI (clicking the save/bookmark button). Post database.
Outputs	Post is added to the user's saved/bookmarked list. Toast notification: "Post saved to your library". List view with filters (date saved, post type).
Destination	Saved bookmarks database tied to the user’s profile.
	1-User clicks the save/bookmark icon on a post.

Action	2-The system records the post ID in the user's saved list in the database. 3-A confirmation is displayed (e.g., "Post saved"). 4-User can access a "Saved Posts" section from their profile or dashboard. 5-User may later remove the post from bookmarks.
Requirements	1-UI element (icon or button) on each post for saving/bookmarking. 2-Saved items database or list associated with user profiles. 3-Page or section to view saved posts. 4-Mechanism to remove saved posts.
Pre-condition	User is logged into the system. Post exists and is viewable.
post-condition	The selected post is saved and listed in the user's saved posts section. User can revisit or remove the bookmark later.
Side effects	May increase database size due to storing saved post data. Potential for duplicate saves if not properly handled.

36) Monthly Volunteer Registration Report for Managers (Tabular Specification).

Condition	**Action**
System reaches the end of a calendar month.	Count the number of new volunteers registered during that month.

User with role = "Manager" requests the report.	Display or allow download of the monthly new volunteer report.
User without "Manager" role requests the report.	Deny access and display an unauthorized access message.
System has no new volunteers for the month.	Generate a report indicating zero new volunteer registrations.

37) Show Key Statistics on Main Page (Structured Specification).

Function	Show Key Statistics on Main Page.
Description	The system shall display the total number of members, number of gifts exchanged per month, and total number of volunteers directly on the main page to inform users and visitors of the platform's current activity and reach.
Inputs	Data from: Member registration table. Gift exchange records (including timestamps). Volunteer user role or status records.
Source	User database (for member and volunteer counts). Gift exchange records (with date fields).
Outputs	Total number of registered members. Number of gifts exchanged in the current (or recent) month. Total number of volunteers. Optionally visualized using counters, icons, or simple charts.
Destination	Main page display section (frontend/UI component).

Action	1-On page load (or at set intervals), the system retrieves: • Total member count from the user database. • Gift exchange count filtered by the current month. • Total users marked with volunteer status. 2-Formats these numbers. 3-Renders them into designated areas on the home page.
Requirements	1-Efficient database queries or cache layer for real-time counts. 2-Public display components (UI) for presenting stats. 3-Scheduled update mechanism if not done in real time. 4-Responsive design for proper display on different devices.
Pre-condition	Users and data (members, gifts, volunteers) already exist in the system. Home page is accessed or refreshed.
post-condition	Updated statistics are visible to any site visitor on the main page.
Side effects	Possible data lag if statistics are not refreshed frequently. Risk of information inconsistency if cached data isn't updated properly. May require frontend styling/space management for clean layout on small screens.

2-Non-Functional requirements:

1) High Availability System (Structured Specification).

Function	High Availability System.
Description	The system shall be available 99.9% of the time, ensuring that users can reliably access it with minimal interruptions. This level of availability equates to a maximum of approximately 43 minutes of downtime per month.
Inputs	System uptime and performance monitoring tools. Infrastructure health (servers, networks, services). Peak usage patterns (e.g., holiday spikes).
Source	Application/DB servers. CDN/load balancer logs.
Outputs	System is accessible to users 99.9% of the time in a given month. Downtime logs or alerts (for system administrators). Failover to backup systems.
Destination	Administrators for alerting and reports.
Action	1-System must maintain high uptime through: • Load balancing and server redundancy. • Failover mechanisms and backup services. • Ongoing performance monitoring. 2-Automatically alert system admins in the event of unplanned downtime.

	3-Schedule updates/maintenance during non-peak hours to avoid impacting availability.
Requirements	1-Cloud or high-availability server infrastructure. 2-Load balancers, backup servers, and fault-tolerant design. 3-Monitoring tools. 4-Automated alerts and incident management workflows.
Pre-condition	System is fully deployed and serving real users. Monitoring and recovery mechanisms are in place.
post-condition	Uptime is maintained at or above 99.9% over a calendar month. Downtime incidents are logged and resolved quickly.
Side effects	Increased infrastructure cost to support redundancy and failover. Complex deployment architecture required to avoid single points of failure. May require automated scaling and real-time monitoring tools.

2) Fast System Response(Tabular Specification).

Condition	**Action**
User initiates any action (e.g., form submit, page load).	Process the request and return a visible response within 2 seconds.
Action response time > 2 seconds.	Log the event, optionally show a loading indicator or timeout warning.

Backend or database query is slow.	Use optimization techniques (e.g., caching, query tuning, asynchronous load).
Response is successfully delivered in ≤ 2 seconds.	Show updated UI content or success message.

3- User Data Privacy (Structured Specification)

Function:	User Data Privacy
Description:	Ensure privacy of user data and compliance with laws.
Input:	User personal and account information.
Source:	User registration and usage.
Output:	Secure and private data storage.
Destination:	System database.
Action:	Apply encryption and access control policies .
Requirement:	The system shall ensure user data privacy and follow data protection laws.

Pre-condition:	User provides personal information .
Post-condition:	Data is securely stored and protected from unauthorized access.
Side Effects:	System must maintain updated legal compliance .

4 – Mobile & Browser Support (Tabular Specification)

Function:	Mobile & Browser Support
Description:	System should work smoothly on all modern devices.
Requirement:	The system shall be mobile-friendly and work on all modern browsers.
Condition:	User accesses the app via browser or mobile device.
Action:	Detect device type and adjust UI layout responsively.

5 – Location-Based Grouping (Structured Specification)

Function:	Location-Based Grouping
Description:	Group users based on their local town or neighborhood.
Input:	User-provided or auto-detected location .
Source:	User profile or GPS .
Output:	User assigned to a local group .
Destination:	Group database.
Action:	Match user location to existing group database .
Requirement:	The system shall support localization for different towns and neighborhoods.
Pre-condition:	User location is available .
Post-condition:	User joins the appropriate local group.
Side Effects:	Location mismatch may affect content relevance.

6 – UI Clarity (Tabular Specification)

Function:	UI Clarity
Description:	Maintain a clean, intuitive interface.
Requirement:	The system shall have a clean and intuitive UI for ease of navigation.
Condition:	User accesses any part of the interface.
Action:	Render a simplified and clear UI layout.

7– Concurrent User Handling (Structured Specification)

Function:	Concurrent User Handling
Description:	Support thousands of simultaneous users.
Input:	Concurrent requests .
Source:	Multiple users.
Output:	Stable system response .
Destination:	System resources.

Action:	Use server scaling and queue management .
Requirement:	The system shall be able to handle at least 10,000 concurrent users.
Pre-condition:	System is deployed in production environment .
Post-condition:	Users can access the system without lag or errors.
Side Effects:	Increased infrastructure cost.

8 – Multi-language Support (Structured Specification)

Function:	Multi-language Support
Description:	Allow users to choose between multiple languages.
Input:	Language selection from user .
Source:	User preferences .
Output:	Localized interface .
Destination:	User UI.

Action:	Load language-specific resources.
Requirement:	Multi-language support (e.g., English, Arabic, French).
Pre-condition:	Language files are available .
Post-condition:	User interface appears in selected language .
Side Effects:	Slight delay in loading translated content.

9 – Code Maintainability (Tabular Specification)

Function:	Code Maintainability
Description:	Design modular code for future upgrades.
Requirement:	Code should be modular to support future enhancements.
Condition:	New feature or module is being developed.
Action:	Update the related module only without changing the whole system.

10 – Dark Mode Interface (Structured Specification)

Function:	Dark Mode Interface
Description:	Provide optional dark mode for better UX.
Input:	User toggle for theme.
Source:	User settings .
Output:	Dark-themed UI .
Destination:	User interface .
Action:	Apply dark theme CSS/styles .
Requirement:	The system should offer an optional dark mode interface.
Pre-condition:	User enables dark mode.
Post-condition:	UI changes to dark color palette.
Side Effects:	Minor increase in CSS complexity .

11 – Action Logging (Tabular Specification)

Function:	Action Logging

Description:	Log critical system and user events.
Requirement:	All important actions must be logged and time-stamped for audit and traceability.
Condition:	User deletes a post or admin bans a user.
Action:	Record the action with timestamp and user ID in logs.

12 – Session Timeout (Structured Specification)

Function:	Session Timeout
Description:	Log out users after inactivity for security.
Input:	User inactivity.
Source:	Session timer .
Output:	Terminated session.
Destination:	Login page.
Action:	Check for inactivity and trigger logout .

Requirement:	Users must be automatically logged out after 15 minutes of inactivity.
Pre-condition:	User is logged in .
Post-condition:	User is logged out and redirected.
Side Effects:	User may lose unsaved changes .

13 – Data Backup (Tabular Specification)

Function:	Data Backup
Description:	Backup the database automatically every 24 hours.
Requirement:	The system must perform a full automatic backup of the database every 24 hours.
Condition:	System is online and time reaches 24-hour cycle.
Action:	Execute backup script and store the file securely.

14 – Privacy Compliance (Structured Specification)

Function:	Privacy Compliance
Description:	Comply with GDPR and similar regulations.
Input:	User data and consent status.
Source:	User settings and registration .
Output:	Legally-compliant data usage .
Destination:	User database .
Action:	Check consent and restrict unauthorized access .
Requirement:	The system must comply with data protection regulations .
Pre-condition:	User has registered and consented to data policy .
Post-condition:	System uses data within legal limits .
Side Effects:	Audit and policy tracking required.