ToGather

System-Wide Requirements Specification

# Introduction

In this system-wide requirements specification document, the project team briefly introduced system-wide functional requirements, system qualities, system interfaces, business rules, system constraints, system compliance, and system documentation. Under all topics, related requirements are defined explicitly to capture different stakeholders’ needs and to establish a clear understanding that will guide the development process.

# System-Wide Functional Requirements

**2.1. Authentication, Authorization and User Management:**

* The system must provide secure authentication mechanisms to verify user identity and enforce role-based access control.
* The system must offer user account management functionalities, including registration, password management, and account deactivation. Access control mechanisms must be configurable.
* Social sign-up integration with third party institutions such as Google must be introduced to ensure sign up with one click.

**2.2 Auditing, Logging and Backup:**

* The system must maintain comprehensive logs of all user activities, system events, and security-related incidents. These logs must be securely stored and readily accessible for auditing and analysis purposes.
* To ensure data integrity in the system failures, the system must implement regular data backup procedures, providing safety against data loss.

**2.3. Integration:**

* The system should integrate with third-party services, such as event organizations, social media platforms, and external data sources, to expand the system's functionality.

**2.4. Scalability and Performance:**

* The system must be designed to scale for ensuring optimal performance when user loads and data volumes grow.
* The system must provide efficient search and query capabilities, enabling users to locate and retrieve specific information from the application.

**2.5. Social Interactions:**

* The system must offer notification and messaging services to keep users informed about important events or updates.
* The system must enable users to create profile pages and share personal information.
* The system must facilitate actions such as adding friends, forming communities, and sending chat invitations.
* The system must provide event-specific chat and collaboration tools to allow event attendees to discuss event details, share photos, and coordinate plans.
* The system should integrate with users' contact lists and social media networks.

**2.6. User Onboarding and Training:**

* The system should offer onboarding materials, tutorials, and help documentation to assist new users in becoming familiar with the system's features and functionality.
* The system should support an online user manual that contains Product Description, Explanation of Product Features, Use Case, Potential Product Risks, How to Articles, Frequently Asked Questions.
* The system should offer user feedback mechanisms, such as surveys or in-app feedback forms, to collect suggestions and improvements from users.

**2.7. Mobile Responsiveness:**

* The system's user interfaces must be responsive and mobile-friendly, ensuring availability across various devices and screen sizes.
* The system should be accessible on multiple platforms, including iOS, Android, and web.

**2.8. Event Scheduling and Recommendations:**

* The system must provide automated event scheduling.
* The system must provide recommendations based on users' interests, past activities, and social connections.
* The application must allow users to create, schedule, and manage events, with features for defining event details, dates, times, and locations.
* The application should support the creation and sending of event invitations including accepted, declined, and pending responses.

**2.9. Localization and Globalization:**

* The system must support English language.
* The system must support local date formats, regional settings and time zones.

# System Qualities

## Usability

* The system shall show the event times according to the user’s time zone.
* The UI shall include distinctive and recognizable action buttons to direct users more easily.
* The UI shall comply with WCAG 2.1 accessibility standards to ensure the usability by individuals having sensitivity and disabilities.
* The application should offer interactive getting started tutorials for all the new users.
* Users under 40 years old shall be able to create a simple task within 2 minutes after sign-up.
* Users older than 40 years old shall be able to create a simple task within 4 minutes after sign-up.
* The UI shall provide consistency to reduce cognitive load. UI elements such as buttons, icons, and navigation shall behave predictably across different sections of the application.
* The system shall inform users about the status of their actions. Confirmation messages, error messages and loading indicators should be provided to provide timely and understandable feedback to the users.
* The system shall provide informative error messages to the users. These messages shall include corrective action suggestions.
* The UI shall be adaptable to various resolutions, screen sizes and orientations.
* The UI shall be usable on both desktop and mobile platforms. Users should be able to switch devices seamlessly.
* For readability, the UI font size shall be 14px at minimum.
* The application shall support both Turkish and English languages.
* The UI shall be compatible with commonly used web browsers such as Chrome, Safari, Opera, Mozilla Firefox, Edge, and their stable versions, and iOS and Android platforms according to the time of this document.
* The UI color palette shall provide a contrast ratio of 4.5:1 for normal text and 3:1 for large text. (WCAG guidelines)
* The application shall include online user manuals to provide guidance to the users.

## Reliability

**Availability**

* The availability target of the system shall be 99.9% uptime at minimum.
* To provide redundancies and minimize service interruptions in case of any failures, backup servers, load balancing, failover mechanism shall be applied.
* The system performance and availability shall be monitored continuously.
* Alerting mechanisms shall be used by customizing thresholds according to the system’s needs and performance.
* Load balancing shall be implemented to distribute the traffic across multiple servers or replicates of services according to the capacity at that time.
* Network connectivity shall be redundant to decrease downtime in case of network failures.

**Frequency of Severity of Failures**

* Scheduled maintenance shall be limited to a maximum of two hours per month.
* The scheduled maintenance shall be in non-prime time slots.
* For critical incidents, the response time shall be 1 hour at maximum.
* For the non-critical incidents, the response time shall be up to 6 hours.
* All incidents and failures shall be reported including the impact and urgency.
* The system shall be tested in each quarter to control its behavior in case of failover and heavy load.

**Recoverability**

* The customer data backups shall be taken to a separate storage location daily.
* The backup retention time shall be a minimum of 30 days. After this retention time, the data shall be archived.
* Data recovery from backups shall be completed in 6 hours.
* Cloud services shall be used to provide geographical redundancy of the system.
* Critical system components such as application servers, database, load balancers shall be deployed by following at least N+1 redundancy.
* The system shall provide automatic failover in case of any failure. The automatic failover shall be deployed within at most 5-10 minutes.
* Data integrity checks shall be implied during backups.
* Users who have lost access to their accounts shall regain access to their accounts within at most three hours by ensuring a secure account recovery process.

## Performance

* The web application's web pages shall load within 2-3 seconds for at least 90% of users under typical user loads.
* The mobile application's pages shall load within 1-2 seconds for at least 90% of users under typical user loads.
* The application shall support minimum of 1000 concurrent users.
* The application shall handle a minimum of 1000 transactions per second.
* Search queries shall return results in at most 500 milliseconds for more frequent searches.
* Database queries shall execute in at most 2 seconds for complex queries.
* The standard deviation of page load time shall be at most 300 milliseconds.
* For repeated database queries, caching mechanisms shall achieve a hit rate of 95 %.
* The transaction time for the integration with third party services and applications shall be in at most 1 second when third party service is accessible, and connection is stable.
* The users from different geographic locations shall access the application regardless of the distance to the server. CDN (Content Delivery Network) shall be used to provide seamless accessibility.
* Progressive loading techniques shall be implemented to make the page loading process more efficient by prioritizing the loading according to content classification.
* The application shall be capable of scaling under varying user loads.

## Supportability

* The application shall be designed and built in a modular perspective to provide isolation between components.
* The source code of the application shall be provided in a version control system such as Git. In case of deployment, source code should be delivered from the version control system continuously.
* System documentation and end-user documentation shall be maintained.
* To identify and resolve issues, the system shall log every incident, errors, exceptions with detailed information, subject, action, and timestamp.
* The system shall include troubleshooting and changelog mechanisms to debug features for development and testing environments.
* There should be an isolated test environment apart from production environment for mitigating security vulnerabilities and deploy bug fixes.
* Changes and updates on the used third-party libraries, APIs, etc. shall be monitored regularly.
* The application shall provide contact information for technical support.

# System Interfaces

## User Interfaces

### Look & Feel

* The application should implement a modern looking and simple interface.
* Some components of the user interface should be similar to the popular calendar and social media applications to ensure intuitive usage and easy onboarding of users.
* 70-20-10 rule which suggests usage of three main colors (background color, information color, and action color respectively for our case) must be used through the UI.
* Colors must be chosen in a way that provides a user-friendly experience to color-blind users too.

### Layout and Navigation Requirements

* User interface must be responsive which means it should be adaptive to different screen sizes from mobile to desktop.
* At least 5% whitespace from both left and right sides should be introduced. Also, enough whitespace should be used between different components.
* Font types and sizes should be selected properly to ensure readability. Minimum font size should be 14px. Headers and important texts should be bold and at least 20px.
* The action button color should be 10% in the 70-20-10 rule and color must be distinguishable.
* Users should be able to view their calendars in three different modes which are monthly view, weekly view, and daily view.
* User interface should consist of two main parts which are a navigation bar on the right and body component that spans to the center and left parts of the screen. For devices like tablets and phones navigation bar should be invisible until the menu icon is touched, and the body component should span the whole screen.
* The navigation bar should include all the necessary links to the important pages such as Home, Profile, Communities, Events, Settings, etc.
* For mobile devices, the UI should implement Back buttons on all pages.
* For all the action windows, Cancel option should be introduced anytime.

### Consistency

* The application should use a consistent color scheme which is aligned with the branding throughout all the pages and fields.
* If a particular style is assigned to a UI component, the same type of components must be styled the same to ensure consistency of user experience.
* Language and terminology should be consistent in all the pages and the fields.
* User preferences such as Task Type Groups apply the same color and styling options for all the components applicable.

### User Personalization & Customization Requirements

* Users should choose whether to use the application in dark or light mode.
* Users should be able to customize their profiles with personal information, profile pictures, and areas of interests.
* Event feed should be displayed based on users’ areas of interests.
* Notifications should be customizable, notification frequency, and sound should be adjustable.
* Users should be able to define styling options for community tasks based on their types.

## Interfaces to External Systems or Devices

### Software Interfaces

* To provide a single-click and secure sign-up and log-in, third-party authentication services (such as Google OAuth) should be used.
* RESTful APIs and secure web requests should be used to gather information from different event organizers’ websites or APIs.
* To integrate with third-party calendar applications (such as Outlook or Google Calendar), official integration methods provided by these applications should be preferred.
* To integrate with third-party mail and SMS services official integration methods provided by these services should be preferred.

### Hardware Interfaces

* To minimize server-side hardware requirements, cloud services should be used.
* Client-side computations should be kept to a minimum to ensure that any device with a simple browser is able to run the application.

### Communications Interfaces

* AWS, Django and React utilities will be responsible for providing basic communication interfaces to our system.

# Business Rules

## <Task Management Rules>

### <Task Addition (TASK-MNG-001)>

If a task is specified with all the necessary information and the addition process is finished, the task will be visible on the user’s calendar.

### <Task Update (TASK-MNG-002)>

If a task’s information is updated by creator or any authorized user, the task information will be updated on all the related calendars.

### <Task Deletion (TASK-MNG-003)>

If a task is deleted by creator or any authorized user, the task will be removed from all the related calendars.

### <Shared Task (TASK-MNG-004)>

A task will be assigned to a user who is not creator if the user accepts the invitation.

### <Task Reminder (TASK-MNG-005)>

A task will be created with a reminder automatically unless the user closes reminder of the related task.

## <Event Management Rules>

### <Event Recommendations (EVNT-RCM-001)>

Events will be recommended by an algorithm to the users based on their interests, location, and past activity. Users can force the algorithm by choosing “Recommend more events like this” or “Do not recommend events like this” from the event feed.

### <Event Report (EVNT-RPR-001)>

Users can report an event that is subject to moderation because of harmful content or abuse of community rules.

### <Event Invitation from User (EVNT-INV-001)>

A user can be invited to an event by another user if and only if users are friends. The event will not be shown in the calendar until the invitation is accepted.

### <Event Invitation from Community (EVNT-INV-002)>

If a user shares an invitation within a community, this invitation will be sent to all the members of the community, and it will be shown in the community calendar but not in personal calendars until the invitation is accepted.

## <Social Networking Rules>

### <Connection Request (SOCL-CON-001)>

A connection between users can be built only if one of them sends a request and the other is accepted.

### <Connection Cancellation (SOCL-CON-002)>

A connection between users can be cancelled if one of the connected users discards the other user from his/her connections.

### <Community Request (SOCL-COM-001)>

Communities can be built within the application by community invitations. To add a user to a community, a member of the community must be connected to that user and send an invitation. The invited user must accept the invitation to join the community. Then, community events and calendars will be visible to that user.

## <Privacy Rules>

### <Profile Visibility (PRIV-PRF-001)>

A user can set his/her profile as private or public. If public, all the content of the profile is visible to all users except the content that is set to private explicitly by the user. If private, only the username and profile picture will be visible.

### <Task/Event Visibility (PRIV-TSK-001)>

A user can set a task or event as private or public. If private, all the content of the task even name will be invisible to all users, only a placeholder on the calendar will be visible to the authorized users (based on the PRIV-PRF-001). If public, event details will be visible to the connected users.

# System Constraints

* Open-source software implementation languages and frameworks will be used in this project. Based on the time of the preparation of this document, the stable versions will be used.
* As backend, software implementation language will be Python.
* As a backend framework Django will be used.
* As frontend, software implementation language will be Javascript (HTML and CSS will be used too).
* As frontend framework, React will be used.
* As a relational database management program, MySQL will be used.
* As NoSQL database management program, MongoDB will be used.
* For development, testing and production environment, cloud services (AWS, GCP etc.) will be used. For development purposes, PaaS or IaaS services will be received according to use cases.

# System Compliance

## Licensing Requirements

* Users who want to use ToGather Application, must sign up with an agreement to KVKK (Kişisel Verilerin Korunumu Kanunu) and Accepting Cookies to enhance site navigation, analyze site usage, and assist in our marketing efforts.
* ToGather Application will not charge any of the customers for licensing in the future.

## Legal, Copyright, and Other Notices

* There shall be defined Intellectual Property, and Copyright Notices.
* ToGather is a free software which will not promise any warranties for the users who are desired to use.
* For the application, domain name claim will be performed by the team as a part of the project, and it will be provided.

## Applicable Standards

* ISO Basic Latin Alphabet

*“The ISO basic Latin alphabet is an international standard (beginning with ISO/IEC 646) for a Latin-script alphabet that consists of two sets (uppercase and lowercase) of 26 letters, codified in various national and international standards and used widely in international communication. In ToGather App, the language and the alphabet will be compatible with ISO standard.”*

* ANSI’s ASCII Character Code of Standard

*“ASCII abbreviated from American Standard Code for Information Interchange, is a character encoding standard for electronic communication. ASCII codes represent text in computers, telecommunications equipment, and other devices. Because of technical limitations of computer systems at the time it was invented, ASCII has just 128 code points, of which only 95 are printable characters, which severely limited its scope. Modern computer systems have evolved to use Unicode, which has millions of code points, but the first 128 of these are the same as the ASCII set.”*

* HTML and Pantone Color Standard

*“Pantone color standards are vibrant, solid colors that can be converted to RGB or CMYK (cyan, magenta, yellow, key) when necessary. HTML color codes are hexadecimal triplets representing the colors red, green, and blue (#RRGGBB). For example, in red, the color code is #FF0000, which is '255' red, '0' green, and '0' blue. There are 16,777,216 possible HTML color codes, and all are visible on a 24-bit display. These color codes can change the color of the background, text, and tables on a web page.”*

* Web Content Accessibility Guidelines (WCAG) 2.1

*“Web Content Accessibility Guidelines (WCAG) 2.1 covers a wide range of recommendations for making Web content more accessible. Following these guidelines will make content more accessible to a wider range of people with disabilities, including accommodations for blindness and low vision, deafness and hearing loss, limited movement, speech disabilities, photosensitivity, and combinations of these, and some accommodation for learning disabilities and cognitive limitations; but will not address every user need for people with these disabilities. These guidelines address accessibility of web content on desktops, laptops, tablets, and mobile devices. Following these guidelines will also often make Web content more usable to users in general.”*

# System Documentation

* The application interface must have “Help” button with a question mark on it.
* The system must direct the user to the online user manual page once he/she presses “Help” button. The system must display “Frequently Asked Questions”, “Product Description”, “Explanation of the Product Features”, “Use Case”, “Potential Product Risks”, “How-to Articles”, “Send Your Question” buttons.
* In “Frequently Asked Questions” part, user must see most frequently asked questions about how to use the application, and their answers. These questions must be updated every month to show the most recent and frequent ones.
* In “Send Your Question” part, user must see a field to type his/her question and a “Send” button. These questions will be directed automatically to Customer Support teams.