ToGather   
Brief Use-Cases

1. **Introduction**

In this document, use cases are introduced briefly for the Inception. Primary actors and use cases are identified.

1. **Overview**

ToGather Next generation Calendar App is a scheduling program which helps people to be more effective and stay connected to their friends. The main purpose of this application is to make its user more organized while enjoying their personal life, increase effectiveness of their time management skill and decrease chaos among the multitude of their tasks. ToGather App is an opportunity cost savior for its users.

ToGather App requires that the users have a stable internet connection and a device to use the app.

1. **Use-Case Diagram**

The figure below shows the use-case diagram for the ToGather App.

metin, tasarım, çizgi, taslak içeren bir resim

Açıklama otomatik olarak oluşturuldu

1. **Actors**

### Users

### Professionals

### This actor represents a person who wants to use the app to bring balance between their personal and professional life.

### University Students

### This actor represents a person who wants to use the app to bring balance between their personal and university life.

### Calendar Applications

### This actor represents third-party entity which provides synchronization data based on integration request.

### Use Cases

* 1. **Create a Task**

The User wants to create a task. System displays the schedule. User selects manual task creation option. System displays task creation form. User fills the required fields for manual task creation in the form. System creates task and displays created task.

* 1. **Update the Task**

The User wants to update a task. System displays the schedule. User selects the task to be updated from the schedule. System displays task detail in editable form. User updates the fields of the selected task. System updates the task and displays the updated information to the User.

* 1. **Remove the Task**

The User wants to remove a task. System displays the schedule. User selects the task that he/she wants to remove from the schedule. System asks for confirmation. User confirms removal. System removes the task and displays the updated schedule.

* 1. **Handle Friendship Request**

The User wants to check friendship requests. The system displays friendship requests if there are any. User accepts requests. System adds friends and displays confirmation message.

* 1. **Review Existing Task**

The user wants to review an existing task. System displays the task list. User selects task to review from the list. System displays detailed information about the selected task.

* 1. **Set User Profile**

The User wants to set his/her user profile. System displays profile to the User. User selects update option. System displays user profile creation form. User fills the personal information fields. System updates user profile and displays updated User information to the User.

* 1. **Add Friend**

The User wants to add friends. System displays all users. User selects users to add as friends. System adds friends and displays confirmation message of sent request.

* 1. **Remove Friend**

The User wants to remove friend. System displays list of friends. User selects the friend to be removed. System asks for confirmation. User confirms removal. System removes friend and displays removal message.

* 1. **Invite a Person**

The User wants to invite a person to the application. System displays invitation form. User enters the e-mail address into the form. System invites the person and displays invitation message.

* 1. **Create Community**

The User wants to create community for a specific area of interest. System displays communities. User selects the community creation option. System asks for community details (community name, description). User fills in the information. System creates community.

* 1. **Update Community**

The User wants to update a community. System displays User’s communities. User selects the community to be updated. System displays community’s information. User updates community information. System updates community.

* 1. **Sync the Calendars**

The User wants to sync calendars with another User. System displays page of friends. User selects the friend to be synced with. System asks for confirmation. User confirms syncing. The system syncs the calendars and displays the synced calendars in schedule view.

* 1. **View Calendar**

The User wants to view calendar of another User. System displays the friend list of the User. User selects a friend from the list. System displays the profile of the selected friend. User selects view calendar option from the profile. System displays the calendar of the friend.

* 1. **Discover New Events**

The User wants to discover new events. System displays events. System prompts a ranked feed of events based on User’s interests, location, past activities, and User’s social connections’ events and interests. User can filter the feed with respect to filtering options. User chooses the event to see the information about it. System shows event details. User adds the event to their calendar. System displays success message.

* 1. **View Event Recommendations**

User wants to view event recommendations. System displays schedule. User selects a timeslot from schedule. System displays recommended events based on given time interval, User’s and related people’s (if any) interests, past activities, and location. User views the recommended events and chooses one of them if he/she wants to view details. System shows event details. User adds event to his/her calendar. System displays success message.

* 1. **Integrate Own Calendar with Third Party Calendars**

User wants to sync an external calendar with ToGather. System displays User’s profile. User selects integrations option. System displays integration options. User selects one external application (from Supported Calendar Applications) to integrate with. System asks for credentials of related application. User shares credentials. The system retrieves data from the application and asks for syncing confirmation. User confirms syncing. System displays success message and synced calendar.

* 1. **Manage Notifications**

The User wants to manage notifications. System displays settings. User selects notification options. System displays notification options. User sets notification options. System asks for confirmation. User confirms. System saves notification options and displays success message.

* 1. **Provide Feedback**

The User wants to share some feedback. System displays guidelines and customer support links. User writes and sends his/her feedback. System displays success message.

* 1. **View User Manual**

The User wants to see the User manuals. System displays guidelines and customer support links. User selects related User manual. System displays the manual.

Revision Table

|  |  |  |
| --- | --- | --- |
| **Revision** | **Description** | **Date** |
| 1.0 | First revision | 01/11/2023 |
| 1.1 | Changes applied according to ToGather\_REVIEW document. | 11/11/2023 |
| 1.2 | - In section 5, 5.4 Create Subtask use case is discarded.  - In section 5, 5.4 Handle Friendship Request use case is added.  - In section 3, Use Case Diagram is updated accordingly.  -In section 5, brief use cases are written in UI free format. | 08/12/2023 |
| 1.3 | Changes applied according to ToGather document. | 22/12/2023 |
| 1.4 | Create Community use case is updated. | 02/01/2024 |