


## Main Page + Add Collection & Add Event

```

classDiagram
    class Header {
        {familyName}
        {userName}
        {avatarUrl}
        Notification
        Menu
    }
    class UpcomingEvents {
        {activityName}
        {Weather Info Icon}
        {activityDate}
        {participate (Y/N)}
    }
    class Collection {
        {collectionName}
        {collectionIconUrl}
        {toEdit}
        {inProgress}
        {closed}
    }
    class EmptyClass1
    class EmptyClass2
  
```

The diagram illustrates the structure of an event management system. It features a **Header** class at the top, which contains attributes for user information (`{familyName}`, `{userName}`, `{avatarUrl}`) and methods for notifications and menu (`Notification`, `Menu`). Below the header is the **Upcoming Events** class, which lists event details like `{activityName}`, `{Weather Info Icon}`, `{activityDate}`, and a participation status `{participate (Y/N)}`. A **Collection** class is positioned below that, managing event groups with attributes `{collectionName}` and `{collectionIconUrl}`, and methods `{toEdit}`, `{inProgress}`, and `{closed}`. At the bottom, there are two empty classes represented by blue rectangles.

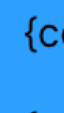



Add Collection

{collectionName}

Type something...

Collection Icon


  


{collectionIconList}

{collectionIconUrl}


Create

add {collectionItem} to {collectionList}

Add Event


{activityName}

Activity Image



{activityImageList}

{activityImageUrl}



{activityDate}

{activityTime}

{activityDescription}


{activityLocation}

{activityUrl}

Create

add {event} to {eventList}

See Event / Collection Overview



See Event

{activityName}

{activityImageUrl}


{activityDate}

{activityTime}

{Weather Info Icon}

{Weather Infos}

{participate}



{participantsList}

{activityOwner}

{activityCreated}

{activityDescription}

{activityLocation}

{activityUrl}

Comments

{commentText}

{commentOwner}

{commentDate}

{commentTime}

{familyName}

Header

{userName}

{avatarUrl}

Notification

Menu

Collection

To Edit

{taskName}

{taskOwner}

{taskResponsibles}

{taskKanban}


{taskPriority}

{taskDue}

In Progress

Closed

## See Task / Add Task

 See Task

{taskName}

{taskPriority}

LowNormalHigh

{taskDescription}

{taskDue}

{taskOwner}

{taskResponsibles}

{userName} ▾

{taskKanban}

To EditIn ProgressClosed

{taskImage}

{taskUrl}


Comments

{commentText}

{commentOwner}

{commentDate}

{commentTime}

 Add Task

{taskName}

Type something...

{taskPriority}

LowNormalHigh

{taskDescription}

{taskDue}

{taskResponsibles}

{userName} ▾

{taskKanban}


{taskImage}

{taskUrl}

Create

add {taskItem} to {taskItemList}

## Family Menu / User Menu / Login

Family Menu

{familyName}

Type something...

{familyMemberList}

{userEmail} {isAdmin}  
{userName}


Invite

{userEmail}

Type something...

Send



add {userEmail} to {familyMemberList}

User Menu

{userEmail}  
{userName}

Type something...

Avatar

{avatarList}  
{avatarUrl}  
{userAvatar}

Login

{userEmail}

Type something...

{userPassword}

Type something...

Login