Design change to capture app activity

RETAINED FOR BACKGROUND INFORMATION

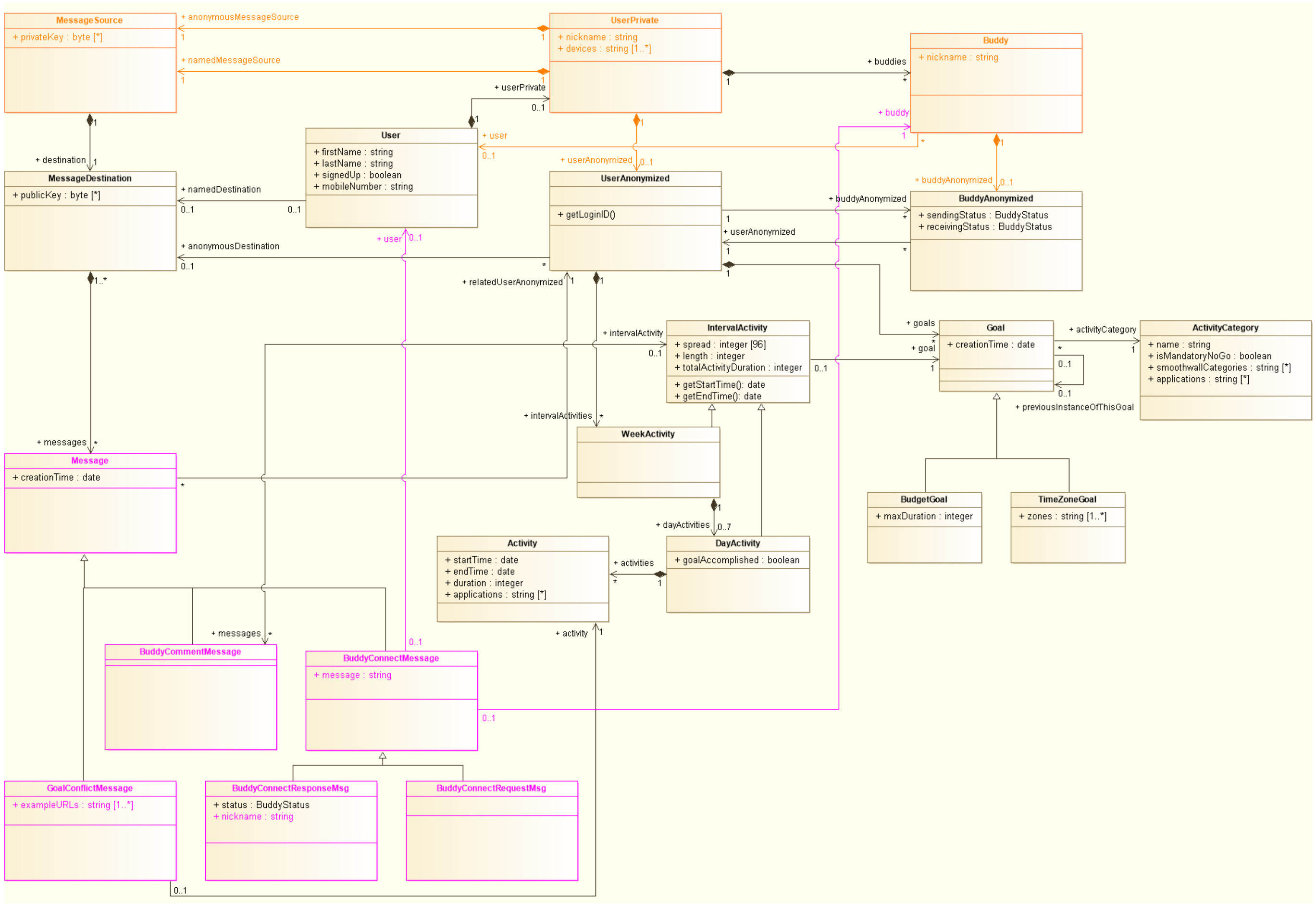
Introduction

During the Development kick-off meeting on 2015-11-11, the functional team introduced a direction change for the Yona app, to focus it as a lifestyle application. This change has implications for the design that are explained on this page.

In the meanwhile, the changes described on this page are incorporated in the Architecture page. This page is retained for background information.

Domain model changes

This is how the new design proposal looks like (a change list is given below the diagram):



1. Removed the association from `Buddy` to `Goal` (see ✓ ~~YD-105~~ - Remove association from Buddy to Goal DONE)

2. Rename `Goal` into `ActivityCategory` and rename all associations accordingly. ✓
   1. Added `isMandatoryNoGo` method. Mandatory no-go categories will return true while other categories return false. When a new user signs up, a budget goal (see below) with a 0 budget will be added for every mandatory no-go goal ~~YD-158~~ DONE . The user cannot remove or change such goals.

   2. Renamed `categories` into `smoothwallCategories` ✓
   3. Added `applications` (an array of strings). Note that the same application might feature in multiple activity categories. ✓
3. Added `Goal`:
   1. The `creationTime` specifies when this goal was created. Updating the specifics of the goal for the same activity category implies creating a new goal. ✓ ~~YD-159~~ DONE
   2. The association `previousGoal` is use to maintain a history of goals (for the same activity category). That way, we can link an interval aggregation to the goal that was applicable at that moment in time. A cleanup job needs to remove the historical goals when no aggregated activity record refers to it anymore. ✓ ~~YD-160~~ DONE
4. There are two types of goals:
   1. `BudgetGoal`. For a goal of this type, the user assigns themselves a budget of a certain amount of time for this activity category. ~~YD-155~~ - Implement BudgetGoal DONE
   2. TimezoneGoal ~~YD-156~~ DONE . This type ofgoal, implies that the user sets time zones for this type of activity.
5. `UserAnonymized` now associates with `Goal` instead of `ActivityCategory` (FKA `Goal`). ✓
6. `GoalConflictMessage` now associates with `Activity` instead of `ActivityCategory` (FKA `Goal`). ✓
7. Added `WeekActivity` and `DayActivity`. These represent the activity related to a certain goal, in a specific day or week. The `DayActivity` initially, it holds a collection of individual `Activity` items. ~~YD-161~~ DONE On `DayActivity` and `WeekActivity` we will also store whether the goal is accomplished and perhaps things like "time outside budget/timezone", etc..
8. A daily job will: ~~YD-162~~ DONE
   1. Aggregate the `totalActivityDuration` and spread in 15 minute intervals on `DayActivity` and `WeekActivity`
   2. Remove the individual `Activity` instances.
   3. Remove the `DayActivity` instances that are older than a configurable time
9. Added `Activity` . Activities are owned by `DayActivity`, have a start and end time, a duration and the apps used for that activity. Activities are either explicitly created based on app activity posted by the app, or implicitly by the analysis engine based on web activity. A new activity is created after a period of 15 minutes (configurable) of inactivity in that category. The end time is updated for every application start/end event and and for every web request, provided the end time moves by at least 5 seconds (configurable). ✓
10. Added `BuddyCommentMessage` to facilitate buddies sending messages to one another. Such a message can be related to an `IntervalActivity` . As the messages are encrypted with the target user public key, two copies of every message are to be retained, one from the sender and one from the receiver. These two are linked through the association `relatedMessage`. ~~YD-163~~ DONE

    A couple of notes:
    1. The design currently does not accommodate commenting on week activities. This would require adding another level of aggregation. Is that really necessary? Or can we simply list all day-target messages in the week overview, with the option to comment there as of related to the first (or last) day of the week?
    2. Due to the encryption, it's really buddy-to-buddy messaging. Say that Richard and John are buddies of Bob and Richard adds a message to an activity of Bob. In that case, John is not able to read that. Richard and Bob have their own copies, but John doesn't have one.
11. For no-go goals, we create a goal conflict message time the same way we do it today, except we now link it to the activity instead of the `Goal`. With that, we don't need an end time on a goal conflict message anymore. The apps are aggregated on the activity and a selection of the URLs is aggregated on the goal conflict message, to enable disclosure. As the URLs can contain information that allows identifying the user, these need to be encrypted and thus stored in the message. ✓

REST service changes

1. Add POST on `/users/{id}/appActivity/` to capture a set of application activity entries (start/end time with app name). ~~YD-164~~ DONE
2. Add collection `/users/{id}/goals/` to maintain the goals of the user. ~~YD-165~~ DONE Embed these goals in the user DTO, to enable fetching everything in one round trip (see ✓ ~~YD-46~~ - Embed goals as objects in User DONE)
3. Add `/users/{id}/activity/` to provide the activity information: ~~YD-166~~ - Implement /users/{id}/activity/ collection DONE
   1. `/users/{id}/activity/weeks/`
      This is a pageable resource. It starts with the current week and the week before (default page size is 2, in accordance with Me -

   Week) and allows pagination into the history.
   It returns a collection of week activities:
   1. Week 2015-W51
      1. Goal Social
         1. Sunday:
            1. Date: 2015-12-06
            2. Total time: 35
            3. Time beyond objective: 5
            4. Day details: <link>
         2. Monday:
            1. Date: 2015-12-07
            2. Total time: 25
            3. Time beyond objective: 0
            4. Day details: <link>
         3. Goal: <link>
      2. Goal Games
         1. Sunday:
            1. Date: 2015-12-06
            2. Total time: 73
            3. Time beyond objective: 29
            4. Day details: <link>
         2. Monday:
            1. Date: 2015-12-07
            2. Total time: 78
            3. Time beyond objective: 0
            4. Day details: <link>
         3. Goal: <link>
      3. Week details: <link>
   2. Week 2015-W52
      1. Similar as above
2. `/users/{id}/activity/days/`
   This is a pageable resource. It starts with the current day and the two days before (default page size is 3, in accordance with Me - Day) and allows pagination into the history..
   It returns a collection of day activities:
   1. Day 2015-12-06
      1. Date: 2015-12-06
      2. Goal Social
         1. Total time: 35
         2. Time beyond objective: 5
         3. Day details: <link>
         4. Goal: <link>
      3. Goal Games
         1. Total time: 73
         2. Time beyond objective: 29
         3. Day details: <link>
         4. Spread: Spread over the day (this is a time zone goal)
         5. Goal: <link>
   2. Day 2015-12-07
      1. Similar as above
3. `/users/{id}/activity/weeks/{week}/{goal-id}`
   The {week} path fragment is an ISO 8601 week, e.g. 2015-W51
   It returns a single week:
   1. Date: 2015-W51
   2. Sunday:
      1. Date: 2015-12-06
      2. Total time: 35
      3. Time beyond objective: 5
      4. Day details: <link>
   3. Monday:
      1. Date: 2015-12-07
      2. Total time: 25
      3. Time beyond objective: 0
      4. Day details: <link>
   4. Spread: <Average spread over the day>
   5. Goal: <link>
4. `/users/{id}/activity/days/{date}/{goal-id}`
   The {date} path fragment is an ISO 8601 date, e.g. 2015-12-06.
   It returns a single day:
   1. Date: 2015-12-06
   2. Total time: 35
   3. Time beyond objective: 5
   4. Spread: <Spread over the day>
   5. Goal: <link>
   6. Collection of related messages

Day variants:

In week overview	In week detail	Day overview	Day detail
1. Date: 2015-12-06 2. Total time: 35 3. Time beyond objective: 5 4. Day details: <link>	1. Date: 2015-12-06 2. Total time: 35 3. Time beyond objective: 5 4. Day details: <link>	Day 2015-12-06 1. Date: 2015-12-06 2. Goal Social 1. Total time: 35 2. Time beyond objective: 5 3. Day details: <link> 4. Goal: <link> 3. Goal Games 1. Total time: 73 2. Time beyond objective: 29 3. Day details: <link> 4. Spread: Spread over the day (this is a time zone goal) 5. Goal: <link>	1. Date: 2015-12-06 2. Total time: 35 3. Time beyond objective: 5 4. Spread: <Spread over the day> 5. Goal: <link> 6. Collection of related messages

4. Add `/users/{id}/buddies/{buddyID}/activity/`, in the same way as provided for `/user/`. ~~YD-168~~ DONE The comments will only include the comments made by the current user.
5. Add `/users/{id}/activity/withBuddies/days/` similar to `/user/activity/days/`, but now include the activities of the buddies with goals on the same category:
   This is a pageable resource. It starts with the current day and the two days before (default page size is 3, in accordance with Me - Day) and allows pagination into the history..
   It returns a collection of day activities:
    1. Day 2015-12-06
        1. Date: 2015-12-06
        2. Goal Social
            1. Activity category: <link>
            2. Buddy A:
                1. Total time: 35
                2. Time beyond objective: 5
                3. Day details: <link>
                4. User: <link>
                5. Buddy: <link>
            3. Buddy B:
                1. Total time: 48
                2. Time beyond objective: 0
                3. Spread: Spread over the day (this is a time zone goal)
                4. Day details: <link>
                5. User: <link>
                6. Buddy: <link>
            4. <self>
                1. Total time: 27
                2. Total time beyond objective: 0
                3. Day details: <link>
                4. User: <link>
        3. Goal Games
            1. Similar as above
    2. Day 2015-12-07
        1. Similar as above
6. Rename `analysisEngine/relevantCategories/` into `analysisEngine/relevantSmoothWallCategories/`. ✓

Open topics

1. If we want the analysis engine to be sensitive to the active app, the app would need to send each app open and close event immediately to the server. Do we want that? Is it helpful for devices that support concurrent activities? The alternative is to keep the analysis engine unaware of app activity and have it ignore the URLs that we know to be app polling URLs