

Meter ActiApp

Data documentation

Rev 0.1 - 28 Jun 2016

1 Introduction

ActiApp is a Cordova based android app, which allows Meter project participants to submit activity and individual information through simple interface. The app can also be used to code up hand written activity booklets.

This document explains

- The ActiApp structure
- The tree structure of activity selection
- The navigation logic between activities and screens

2 ActiApp structure

The app is build around a html5 user interface consisting of the following sections:

1. **Navigation bar** (navbar): Three buttons at the top of the screen.
 - **Home**: navigate to the ActivityList where current activities can be seen and new ones added.
 - **Survey**: navigate to the Survey ('About Me'), where information about the individual should be entered.
 - **Help**: navigate to a selection of video and written instructions.
2. **Header** (header): The region where the question (e.g. "What are you doing?") is displayed
3. **Activity List** (activity_list_pane): Shown on 'home' screen.
 - **Add new activity** (btn-addnew): Buttons to add new activity (time stamped as now) and 'recent' activity, which opens a 'how long ago' dialogue.

- **Catch up list** (catchup-list): List of past times for which an entry is encouraged (times like 5:30pm, 6pm, 8am).
 - **Lists of activities** (activity-list): Currently live activities. Can be removed when completed.
4. **Choices** (choices): A 2 by 3 array of buttons populated by the `navigateTo` function. Each button navigates to a new screen.
 5. **Footer** (footer-nav): Two buttons to aid navigation. Left for 'back', right for 'done'.

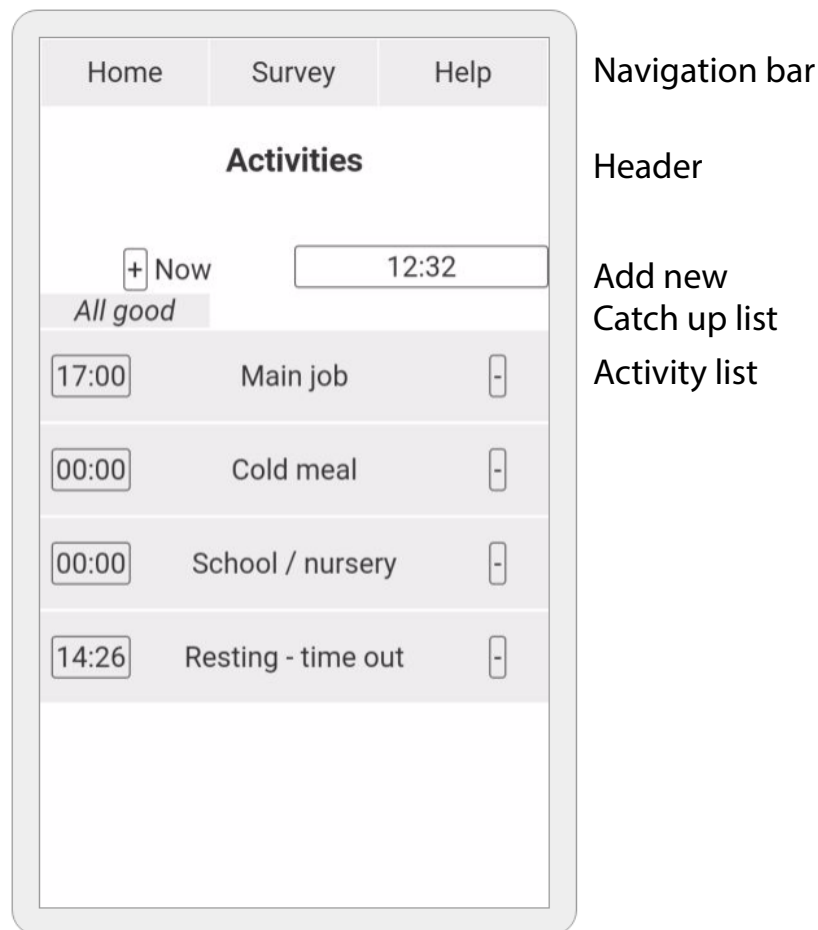


Figure 1: Home screen

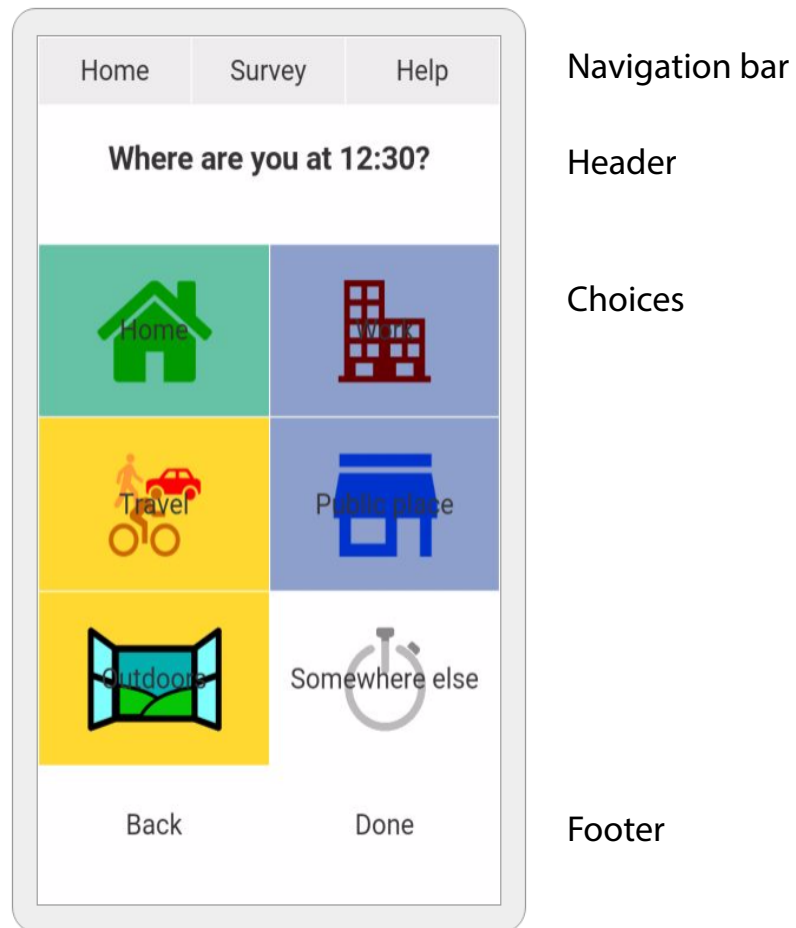


Figure 2: Activity choice screen

3 Activity navigation

Activity entries follow a common sequence shown in Figure 3.

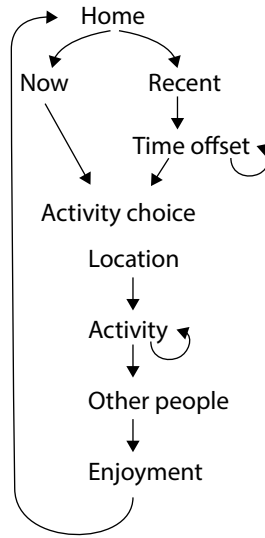


Figure 3: Entry sequence for activities

Each screen, other than the *home screen* (Figure 1) is structured as shown in Figure 2 with six choice buttons. Screens are populated from the file `screens.json`.

```
"activity root": {
  "title": "Where are you at ${time}?",
  "help": "Your location.",
  "activities": [
    "Loc home",
    "Loc work",
    "Loc travelling",
    "Loc friends",
    "Loc outdoors",
    "Loc other location"
  ]
}
```

"activity root" is the `screen_id` by which the screen is identified. The Meter naming convention is for screens to be lower case, to tell them apart more easily from `activities`, which are preferably upper case.

The "title" string is displayed in the Header, "help" text is displayed when Help is toggled on in the Navigation bar.

The six buttons in Choices are populated based on the array "activities". Each of those strings ("Loc home", "Loc work"...) are keys for activities.json. In this file activities are represented as follows:

```
"Loc home": {  
  "title": "meter: location home",  
  "icon": "home",  
  "caption": "Home",  
  "help": "In or around your home / garden",  
  "ID": 30011,  
  "value": 1,  
  "category": "care_self",  
  "next": "activity main"  
}
```

"Loc home" is the unique lookup key for this activity. The button is populated with "caption" text and optionally can have an "icon" with .png as the default extension. The "title" string is the text that will be stored as the activity description in the database and can be more expansive than the caption, for instance to explain the 'path' by which the activity was selected.

As with screen.json, the "help" text is only displayed when Navigation bar > Help is toggled on.

Time use code values are stored under "ID". These values are based on HETUS (the Harmonised European Time Use Survey), but several extensions have been made. Code ranges are explained in Section XXX Code Ranges.

Not all activities have a "value". These are used for numerical coding in some cases (location, enjoyment and for time adjustments where values represent minutes).

The "category" value identifies the main types of activity:

```
var CATEGORIES = [  
  "care_self",  
  "care_other",  
  "care_house",  
  "recreation",  
  "travel",  
  "food",  
  "work",  
  "other_category"];
```

These are used for the colour coding of buttons.

The **"next"** value identifies the **screen_id** to be displayed if this button is pressed. This key identifies a **screen** in **screens.json**, which in turn looks up another set of 6 buttons.

In the above case **"activity main"** will produce a screen with the following activities:

```
"activity main": {  
  "title": "What are you doing?", "help": "at this moment",  
  "activities": [ "care", "work", "Leisure", "food", "Appliances", "more recent" ]  
}
```

(Note: the upper case convention is not observed yet here)

At the end of a typically 3-5 screen deep tree structure follows in most cases **"Other people"**, which asks for a count of people someone was with, followed by the **"enjoyment"** screen. The **"next"** field in all activities displayed points to **"home"**, which displays the list of activities as per Figure 1.

4 Code Ranges

The **"ID"** fields in **activities.json** demarcate several ranges with different purposes:

Code	Function
0 - 10000	Time use codes
10000 - 10100	Relative time adjustment
10100 - 11000	Absolute time adjustment
20000 - 20010	Enjoyment
30000 - 30040	Location
90000 - 91000	Survey answers