

Experience Sampling – remote survey system

This application and backend interface allows researchers to conduct surveys remotely using the mobile phone on Android and iOS.

This app is developed by BOSONIC.design in assignment of the department of Human-Technology Interaction @ Eindhoven, University of Technology.

info@bosonic.design - <http://www.bosonic.design/>

hti@tue.nl - <https://www.tue.nl/en/university/departments/industrial-engineering-innovation-sciences/research/research-groups/human-technology-interaction/>

App files overview

In “www/js/func.js” you find the most important function that you might want to edit to customize the app to your needs. This file includes for example the “renderQuestion” function in which you can provide logic to show certain question depending on specific parameters. For example, you can hide a question in the weekend (Saturday and Sunday) and show only throughout workdays (example provided). You can also show specific question only in the morning, or the evening.

Furthermore, this file allows you to schedule local notifications, to remind your participants to fill in the questionnaire on specific times. This function is called “scheduleNotifications”. This part is based on the local-notifications plugin of Cordova, so more information about how to set-up notifications is found here: <https://github.com/katzer/cordova-plugin-local-notifications>.

Lastly, you might want to edit is “www/js/pages.js”. This file contains the navigation logic of the app, allow specific function to fire when you enter or leave a specific page. The app is based on Framework7 framework, so check out <https://v1.framework7.io/docs/page-callbacks.html> to see what navigation functions are possible. You can also look at the general documentation to see what else is possible: <https://v1.framework7.io/docs/get-started.html>

In order to change the styling of the app, change the CSS in “www/css/style.css” accordingly.

Survey Questions

Type	Description	Label	Label Example
ShortText	Single line text input	-	
LongText	Multi line text input	-	
Select	Select a single option	Write down individual option, separated by ‘;’. Write ‘-s’ behind an option to automatically select it.	Mouse;Duck-s;Bee Note that ‘Duck’ will be automatically selected.
MultiSelect	Select multiple option	Write down individual option, separated by ‘;’.	Mouse;Duck;Bee

Likert	Likertscale (similar as Select, however the options will be displayed horizontally).	Write individually cases like Select. However, one can also write only '5', '7' or '9' to automatically generate likert scale with corresponding options.	No;Neutral-s;Yes Note that 'Neutral' will be automatically selected. OR 5 Note that automatically a 5-point Likert scale will be generated labelled '1' to '5'.
Slider	Select a value between a maximum and minimum number.	Write in the form of "minimum-selected-maximum", where minimum is the lowest value, selected the default value and maximum the maximum value.	0<50<100
Dropdown	Select a option from a dropdown list	Write down individual option, separated by ';'. ;	Mouse;Duck;Bee
Date	Select a date	-	
DateText	Generates 3 specific fields for date input in the form dd/mm/yyyy	-	
Time	Select a time	-	
ShareLocation	Generates a checkbox, for accepting to share the device's current location and saves the latitude and longitude.	-	
ChooseLocation	Allows the user to select a location on a map from Google. Saves latitude and longitude of the specified location	-	
Recording	Record audio through the microphone	-	
Photo	Select photo from phone or take a photo with the camera	-	

Setting up the system

Web-interface set-up

phpMyAdmin

1. Navigate to your phpMyAdmin environment, databases and import a new database. In the main folder you can find ExperienceSampler.sql, that has to be imported.

2. Navigate to the new created database and select the questions table. Edit here the questions according to the table found under "Survey Questions" in this document.

Web-interface files

1. In the Sampling-WebInterface folder open the file "php/Config.php".
2. In this document give up credentials that will be used by the app to identify itself (remember these, as you will need them for the app), as well as the credentials to access the database.
3. That is all! Upload all the files to your host.

General app set-up

1. Navigate to "Mobile-Experience-Sampling-Master/Sampling-App" and open the config.xml file.
2. On line 2, give your app an appropriate package name, in the widget's attribute id (default value is "com.tue.experienceSampler") and app name (default value is "Experience Sampling").
3. You might also want to edit additional parameters like description and author information found on line 4 to 8.
4. On line 18 and 25 you find the access tag with default value "<http://yourweb.com>", change this value to your own host where you uploaded the web-interface of the app.
5. Next, open the file config.js found in "www/js".
6. In this file adjust the parameters found on line 18 to 20, with the credentials you set in the WebInterface.

Android Specific

1. Go to firebase (<https://console.firebase.google.com/>), create a project and download the "google-services.json" file for the android SDK.
2. Place this file in the root folder of the app ("Mobile-Experience-Sampling-Master/Sampling-App").
3. Go to Google maps Android SDK (<https://developers.google.com/maps/documentation/android-api/>) and get your google maps key.
4. Using command line navigate to the root folder of the app, and execute the following code:
 - a. Cordova plugin add cordova-plugin-firebase
 - b. Cordova plugin add cordova-plugin-googlemaps --variable API_KEY_FOR_ANDROID="<insert google maps key here>"
 - c. **!-important Depending on the cordova-plugin-firebase you added, you might need to replace the "after_prepare.js" file found in the plugins/cordova-plugin-firebase, with the file in _pluginTweaks folder.**
 - d. Cordova platform add android
5. Run the following code in the command line: "Cordova build android" to create your apk file for testing purposes or upload the apk file directly to your phone over USB with the Android Debug Bridge (ADB). Look at cordova's documentation for more information about building a release version of the app.

iOS Specific

1. **!- important, you need a PAID Apple Developer account in order to make push notifications work.**
2. Go to firebase (<https://console.firebase.google.com/>), create a project and download the "GoogleService-Info.plist" file for the iOS SDK.

3. Place this file in the root folder of the app ("Mobile-Experience-Sampling-Master/Sampling-App").
4. Go to Google maps iOS SDK (<https://developers.google.com/maps/documentation/ios-sdk/>) and get your google maps key.
5. Using command line navigate to the root folder of the app, and execute the following code:
 - a. Cordova plugin add cordova-plugin-firebase
 - b. Cordova plugin add cordova-plugin-googlemaps --variable API_KEY_FOR_IOS="<insert google maps key here>"
 - c. Cordova platform add ios
6. Then open the file <app_name>.xcodeproj in the folder "platforms/ios" and follow the Xcode process to push the app to your device.
7. Make sure you turn on "inter-app audio" and "push notifications" to make certain plugins work.

Firestore Usage

Go to grow > notifications to start sending push notifications.

For sending a notification to everyone, use for target topic with the text 'Announcement'.

For sending it to a specific user use for target single device and place a token from the database (Users table) in the field.

!- important: this is a bit annoying, but the plugin sadly works like this: under advanced options > custom data you MUST add some information to make sure the message is correctly received:

For the field key use 'content' and for value copy or rewrite the initial message you wrote for the notifications on the top of the page. If you do not do this, the notification will not be received correctly.

If you do this correctly, you will receive a push notifications and/or a message on the message page.

Considerations

When deploying your app to the 'wild' consider adding <https://github.com/tkyaji/cordova-plugin-crypt-file> to encrypt your app and increase safety.

Plugins versions

This are the plugin versions which should work with this system:

- cordova-plugin-badge 0.8.7 "Badge"
- cordova-plugin-camera 4.0.2 "Camera"
- cordova-plugin-device 2.0.1 "Device"
- cordova-plugin-file 6.0.1 "File"
- cordova-plugin-file-transfer 1.7.1 "File Transfer"
- cordova-plugin-firebase 0.1.25 "Google Firebase Plugin"
- cordova-plugin-geolocation 4.0.1 "Geolocation"
- cordova-plugin-googlemaps 2.2.5 "cordova-plugin-googlemaps"
- cordova-plugin-local-notification 0.9.0-beta.3 "LocalNotification"
- cordova-plugin-media 5.0.2 "Media"
- cordova-plugin-network-information 2.0.1 "Network Information"
- cordova-plugin-whitelist 1.3.3 "Whitelist"