Mobile Health EMA
Project Description and Clarification

Erin Mullen | Micah Jenkins | Kily Nhan | Kim Nguyen | Edwin Zheng

Mentors:
Porismita Borah
Nicole O'Donnell

Instructor:
Behrooz Shirazi

Project Description

The Ecological Momentary Assessment (EMA) application will collect data from participants in real time. The app can store predefined questions. The Admin is able to send short messages to remind the participants to take the survey. The participant can use the app to respond to multiple surveys in a day.

Unlike other survey applications, such as SurveyMonkey or Qualtrics, this app will allow admins to send push notifications, messages, and pictures. Although a survey application may be embedded in the program, we will provide the admins with more features as well as giving the users the ability to fill out surveys offline.

Framework[1]

For the framework, Xamarin will be useful for the following reasons:

- It is cross-platform, and so you can completely write the code in C#, and deploy it for Android, iOS, and Windows.
- We can share as much as 75% of the code across the varying operating systems. It will help us shorten the development cycle and leave less room for bugs.
- Xamarin binds the same APIs and UI controls that are used to build iOS, Android and Mac apps in their respective platform specific languages. For Windows development, Xamarin with Microsoft Visual Studio offers Windows Phone and Windows 8 applications.
- Xamarin has TestCloud which allows you to test your apps automatically.
- Anything you can do in Objective-C, Swift, or Java you can do in C# with Xamarin.

Features

The system includes: mobile app (for users), database, web portals (for clinicians/study coordinators), error detection and software update.

The web portals will allow administrators to create new surveys and perform multiple surveys/studies by different clinicians/study coordinators at same time. It will include the option to send push notifications to the user. Notifications should include a confirmation that the notification was successfully sent, and the ability to set exact message timing. It should have different user levels including Admin, Clinician, and Participant. Admins will be able to delete data sets after it's completed and saved in secure space.

The mobile app will be available for iOS, Android, and Windows phones in their respective app stores. Users will be able to create their own accounts and passwords. These will be stored by the app and will allow the user to remain logged in. Users should have some form of ID# to track their responses. Push notifications sent to the user will include the survey name and a timestamp. Completed surveys will include start date/time and time spent on completing the survey. Completed surveys will be stored on the user's device when offline using the minimum space required and data will be deleted from the user's phone immediately after uploading to the server.

---

[1] http://www.softwebsolutions.com/resources/5-reasons-why-xamarin-is-cross-platform-development-king.html

The database will have options for researchers to easily export data to an excel file, query the database, create reports, and download the results. It also can have data visualization, show number of data points collected in some time intervals(hourly, weekly, monthly), and show mean, standard deviation, and variance for different variables.

Limitations

Some of the challenges we can foresee for this project include building the application so that it is cross platform, making sure it is robust enough to handle many users logging/sending data at once, designing in a way that will be easily maintainable and will last as long as it can, and balancing space on the devices. As stated above, we think using Xamrin C# is a viable option for solving the cross platform challenge. However, we may need to focus on one platform at first and then port it across to the others. Traffic from many users to the database could potentially cause issues. Maintenance is a concern as we will be graduating soon after completion. An article on the subject by Savvy says mobile apps should be updated 1-4 times a month.[2] We will need to discuss a plan to solve this issue. Lastly, we will need to set up space requirements and work to balance the amount of space on the user's' phone that we occupy. This can be done by sending data frequently and then removing it after it has been sent successfully.

Client Identification and Preferences

The clients for this project are Porismita Borah, assistant professor for the Murrow College of Communication at WSU, and Nicole O'Donnell, Murrow College phD student. Their preferences include two main interfaces - one for admins and another for users. The admin interface should be usable on a desktop and cross platform, while the user interface should be a cross platform app. The admins should be able to set up surveys and messages for the users. Data and responses provided by the users should be stored in a database accessible by admins.

Stakeholder Identification & Considerations

Stakeholders for this application include users and administrators (researchers, professors, graduate students, etc). Users are participants that meet a certain criteria selected by administrators. Considerations we must make for the users and administrators are accessibility to the app (is the application usable on Macs and PCs) and data (only admins should be able to view and access the survey results). We must also consider the situation where users are offline and admins can't send new messages or collect data.

[2] https://savvyapps.com/blog/how-often-should-you-update-your-app