

MOBILE TEST AUTOMATION FOR TAMIL NATIVE MOBILE APPS IN ANDROID & IOS

Karthik Bharathi, IT Professional <mobility.karthik@gmail.com>

This paper is intended to demonstrate on how to plan and execute development and automation testing of Native apps addressing the areas of Internationalization and Localization for TAMIL in Android & iOS Devices.

Internationalization

Today's enterprises operate across the globe catering to various customers speaking different languages. With the advent of Mobile phones, enterprises have the opportunity to internationalize their reach across the globe for their customers in their native language through Native application. Internationalization helps us to develop a Native Application once and use the same Application for multiple languages.

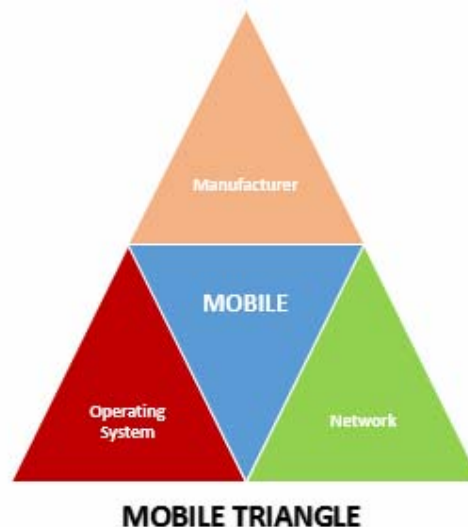
Localization

In order to reach customers across different countries speaking different languages, Native application developers use the locale feature of the mobile to identify the local language preference of the customer and tailor the native application to dynamically display the application contents in the local language of choice for the customer.

Mobile Testing Challenges

Mobile applications operate within three major constraints imposed by the market dynamics which I call it as "Mobile Triangle". Multiple Operating Systems, Multiple Manufacturers and Multiple Networks form the Mobile Triangle. Mobile Applications needs to work within the rigid constraints imposed by the three edges of the Mobile Triangle.

Mobile Test Automation is the answer to the myriad issues of mobile environment to deliver quality assurance to the native mobile application being developed.



Mobile Test Automation open source tools have built in mechanism to test and validate native applications written in any language which is supported by Unicode. Robotium is one such mobile test automation open source tool which automates the functional testing of the

native application including internationalization and localization functionality. Robotium tool is built on the instrumentation libraries provided by Google as part of Android instrumentation libraries. These instrumentation libraries have the ability to interact with the native application running either in emulator or real device. Robotium tool can be run in conjunction with the JUNIT libraries to produce the test results validating the functionality of the mobile native application.

Robotium mobile automation tool has the ability to check the localized strings, images, application names depending on the language used by the customer.

Custom Locale

Locale Settings is at the crux of the Internationalization and Localization initiative. Tamil language locale information is **ta_IN**. Locale information is set under Custom Locale property of the Settings information in the emulator and device. Once set, the device and the emulator customizes all applications to reflect contents in the language set in the custom locale.

Designing Native Application for Internationalization and Localization Needs



Uniqueness of non-European languages is the major criteria when designing native applications. Non-European language alphabets (including Tamil) tend to be 25% larger in size than their European counterparts. Size is one major attribute for the characters to be displayed in the mobile application and sufficient space needs to be provided when the application is designed. For example, Apple provides Auto Layout feature in the Xcode IDE during the design process. Auto Layout feature adapts the space according to the language used in the application.

| Internationalization | Localization |
|--|---|
| <ul style="list-style-type: none"> • Ability to handle application input, output in the user’s native language. • Ability to handle date, time and number formats • Use appropriate Calendar and time zone for processing data. | <ul style="list-style-type: none"> • Ability to render content depending on the user locale. <ul style="list-style-type: none"> • Localizing Images • Localizing Strings • Localizing Dynamic Strings • Localizing App Name |

XLIFF – XML LOCALIZATION INTERCHANGE FILE FORMAT

XLIFF is a single, uniform and consistent format used to store information to display in different languages. Native application contents which are to be customized in multiple languages are pulled into XLIFF format and translated accordingly. The Presentation and

Business logic embedded within the native application remains untouched. This provides the advantage of separating the Presentation and Business layer of a native application.

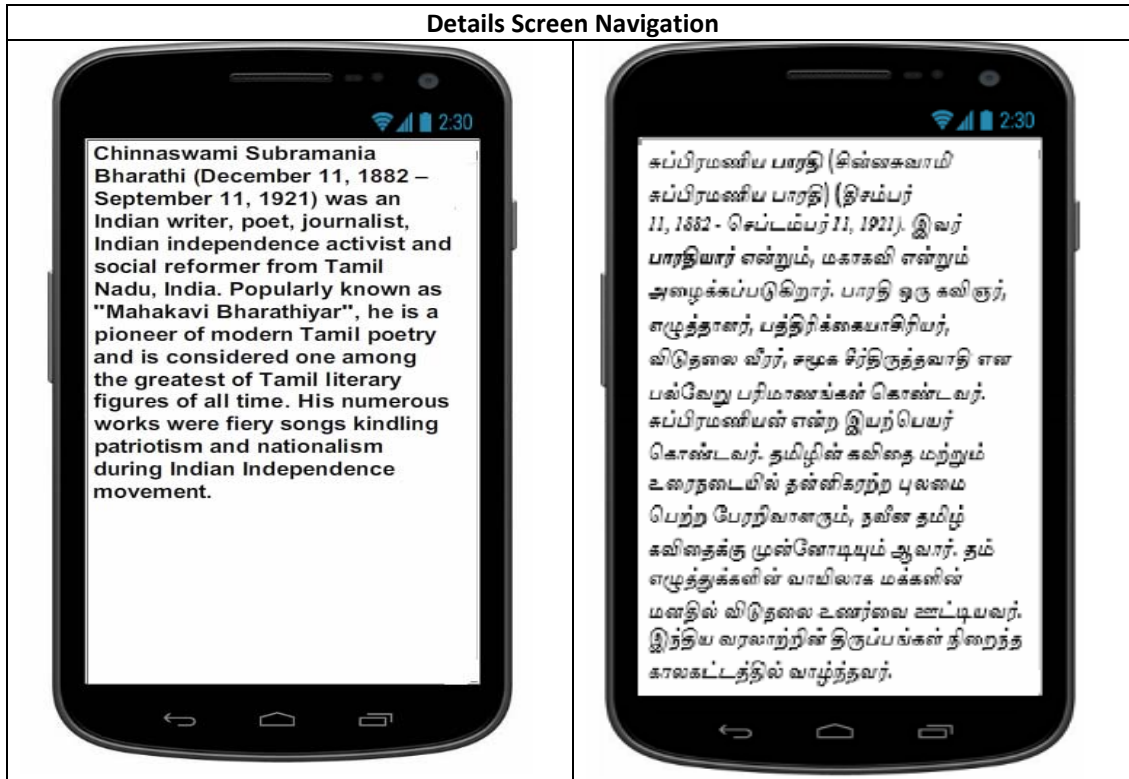
| English Screen Listing | Tamil Screen Listing | | | | | | | | | | | | | | | | | | |
|---|----------------------|-----------|---------------|---------|------------|------------------------|--------------------|---------------------|---------------|---|-----------|----------|------------|--------|-----------|----------------------|------------------|-----------------|--------------|
|  <p>LIST OF TAMIL POETS</p> <table border="1"><tr><td>Agastiayar</td></tr><tr><td>Avvaiyyar</td></tr><tr><td>Bharathidasan</td></tr><tr><td>Kapilar</td></tr><tr><td>Kannadasan</td></tr><tr><td>Desigavinayagam Pillai</td></tr><tr><td>Ramalinga Swamigal</td></tr><tr><td>Subramanya Bharathi</td></tr><tr><td>Thiruvalluvar</td></tr></table> | Agastiayar | Avvaiyyar | Bharathidasan | Kapilar | Kannadasan | Desigavinayagam Pillai | Ramalinga Swamigal | Subramanya Bharathi | Thiruvalluvar |  <p>தமிழ் கவிஞர்கள் பட்டியல்</p> <table border="1"><tr><td>அகத்தியர்</td></tr><tr><td>ஓளவையார்</td></tr><tr><td>பாரதிதாசன்</td></tr><tr><td>கபிலர்</td></tr><tr><td>கண்ணதாசன்</td></tr><tr><td>தேசிகவிநாயகம் பிள்ளை</td></tr><tr><td>இராமலிங்க அடிகள்</td></tr><tr><td>சுப்ரமணிய பாரதி</td></tr><tr><td>திருவள்ளுவர்</td></tr></table> | அகத்தியர் | ஓளவையார் | பாரதிதாசன் | கபிலர் | கண்ணதாசன் | தேசிகவிநாயகம் பிள்ளை | இராமலிங்க அடிகள் | சுப்ரமணிய பாரதி | திருவள்ளுவர் |
| Agastiayar | | | | | | | | | | | | | | | | | | | |
| Avvaiyyar | | | | | | | | | | | | | | | | | | | |
| Bharathidasan | | | | | | | | | | | | | | | | | | | |
| Kapilar | | | | | | | | | | | | | | | | | | | |
| Kannadasan | | | | | | | | | | | | | | | | | | | |
| Desigavinayagam Pillai | | | | | | | | | | | | | | | | | | | |
| Ramalinga Swamigal | | | | | | | | | | | | | | | | | | | |
| Subramanya Bharathi | | | | | | | | | | | | | | | | | | | |
| Thiruvalluvar | | | | | | | | | | | | | | | | | | | |
| அகத்தியர் | | | | | | | | | | | | | | | | | | | |
| ஓளவையார் | | | | | | | | | | | | | | | | | | | |
| பாரதிதாசன் | | | | | | | | | | | | | | | | | | | |
| கபிலர் | | | | | | | | | | | | | | | | | | | |
| கண்ணதாசன் | | | | | | | | | | | | | | | | | | | |
| தேசிகவிநாயகம் பிள்ளை | | | | | | | | | | | | | | | | | | | |
| இராமலிங்க அடிகள் | | | | | | | | | | | | | | | | | | | |
| சுப்ரமணிய பாரதி | | | | | | | | | | | | | | | | | | | |
| திருவள்ளுவர் | | | | | | | | | | | | | | | | | | | |

In the screens above the English version and the Tamil version differs only in the content and not in terms of presentation or the business logic. The English version and the Tamil version were not developed separately. Instead the language information is separately created and stored in XLIFF format. The business logic and the views are created separately. Robotium test tool handles the information verification and validation and compares it with the expected information to be displayed. The entire test process is completely automated and is executed without manual intervention.

Clearly seen is the difference in the size of the language characters. Tamil characters tend to be bigger than its English counterparts. Robotium tool is intelligent enough to compare only the text and not distracted by the size of the text. The Robotium tool also tests on how the mobile application is navigated due to the customer interaction with the native application.

On customer touching any of the Tamil poet's name, the details about the poet are displayed in a new screen. Test automation tools navigate to different screen simulating a

touch event and tests the information displayed in the new screen as well. This process is continued until all the screens and contents in the native application are tested and validated.



Robotium

Robotium being an open source tool is available to anyone for free. The proprietary tools available in the market for automation testing also support all Unicode compliant languages including Tamil. Robotium provides a wrapper which exists on top of the Google Android Instrumentation libraries thereby simplifying the art of automation testing for the native application functionality.

Solo is the Robotium library which provides the necessary functions for mobile native application automation. It has the ability to simulate all the user interactions with the native application. The localized information placed in resources section of the application are retrieved and compared with the expected values in the test data section at runtime. The results of the comparison are documented using JUNIT libraries and stored in the test results section.

Internationalization and Localization ensure that the concept of **“Develop Once, Deploy it for any Language”** is fulfilled for the mobile native application. Test Automation tools ensure that the native application written for whatever language is tested and validated.

References

- Robotium – Open Source Test Automation Tool
 - <http://code.google.com/p/robotium>
 - <http://www.robotium.com>
- Information on Android from <http://developer.android.com>
- Information on iOS and Apple from <http://developer.apple.com>
- Wikipedia <http://wikipedia.com>
-

About the Author:

Bhararathi Karthik is an IT professional with 16 years of the experience in the field of software development in mobile android and iOS applications, enterprise resource planning applications, client server and web applications.