Blind Quiz: Android Application for Visually Impaired People to Access Moodle Quizzes

Sudharma Priyadarshani Subasinghe
Department of Computer Science and Engineering
University of Moratuwa, Sri Lanka
suba.11@cse.mrt.ac.lk

***Abstract*—Although the Moodle facilitates the Accessibility feature for visually impaired people it is not much user friendly for them. The requirement of third party software is the main limiting factor in the existing solution. But Android provides rich accessibility service for different able people. Therefore the application uses the Android platform to give access to Modular object oriented dynamic learning environment (Moodle). This application provides support to access quiz course material on Moodle. People who are visually impaired cannot access Moodle quizzes due to their disability as normal ones. Necessarily they need an external support from login to finish accessing Moodle. But they can access the mobile devices using Android Accessibility service. Thus the project is aimed to build Android Interface to Moodle quizzes for visually impaired people using Android Accessibility service.**

Keywords—accessibility, Learning Management System, Android;

I. INTRODUCTION

A. *Overview of Moodle*

Moodle is an Open-Source, Course Management System (CMS) or a Learning Management System (LMS) which has been widely spread among many institutes in the World. Moodle supports people to interact with the learning material, construct new material for others, and interact with other students about the material. Moodle allows adding several kinds of static course material such as:

- Web pages
- Links to anything on the Web
- A directory of files
- A label that displays any text or image

Moreover, Moodle allows adding interactive course materials such as:

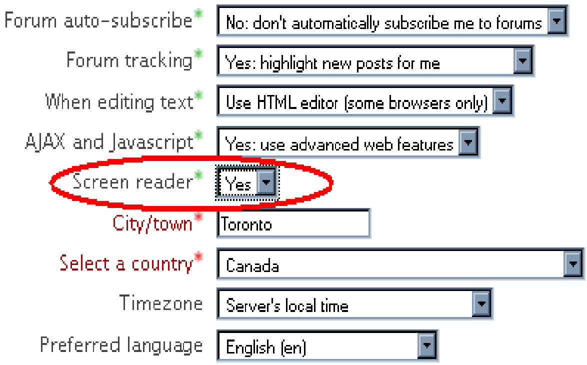
- Assignment
- Choice (a single question)
- Lesson
- Quiz

Offering activities where the students can interact is another functionality provided by Moodle. Some of such functionalities are as follow.

- Chat
- Forum
- Glossary
- Wiki
- Workshop

B. *Moodle for Visually Impaired People*

Since Moodle is an effective LMS it can be used by institutes for visually impaired people. Moodle has added accessibility features to make accessible for visually impaired people. From Moodle 2.7(latest Moodle version) [1] onwards it supports screen reader feature [1]. But there is a limitation with the existing feature where it requires third party software as the screen reader to make accessible for visually impaired people. The user profile has an option to enable the screen reader as Fig.1, which makes Moodle pages screen reader friendly.



Forum auto-subscribe*	No: don't automatically subscribe me to forums
Forum tracking*	Yes: highlight new posts for me
When editing text*	Use HTML editor (some browsers only)
AJAX and Javascript*	Yes: use advanced web features
Screen reader*	Yes
City/town*	Toronto
Select a country*	Canada
Timezone	Server's local time
Preferred language	English (en)

Fig. 1. Enable Screen Reader.

C. *Overview of Android*

Android is the first complete, open, and free mobile platform [2]. It provides a comprehensive software development kit, with powerful developing tools for feature-rich applications. There are no costly barriers to entry for developers, which is no required fees [2]. (A modest fee is required to publish on third-party distribution mechanisms such as the Android Market)

With smartphones and tablets becoming more popular, the operating systems for those devices have evolved as well. Android is such an operating system for low powered devices that run on battery and are full of hardware like Global Positioning System (GPS) receivers, cameras, light and orientation sensors, Wi-Fi and Universal Mobile Telecommunications System (UMTS) connectivity and touchscreen.

Android applications are written in Java and run in virtual machines [2]. For this purpose Android features the Dalvik virtual machine which executes its own byte code. Dalvik is a core component, as all Android user applications and the application framework are written in Java and executed by Dalvik.

D. Android for Visually Impaired People

Android facilitates the Accessibility Service which enables access the mobile devices for different ability people. Therefore visual impaired people can easily access mobile devices within the enabled accessibility service. The Accessibility Application Progamming Interface (API) is available since release 1.6 of the Android platform.

II. LITERATURE REVIEW

A. Android Accessibility

The accessibility APIs are light-weight with no performance impact if accessibility is disabled [3]. On Android devices each application runs in a separate process. Hence, the application a user is interacting with and the accessibility services which provide feedback based on that user's actions run in separate processes.

Therefore, to propagate the information of user interactions to interested accessibility services the Android User Interface (UI) layer is instrumented to fire accessibility events if something noticeable happens e.g.:(say clicking a button). These events are dispatched by the system to the accessibility services which are registered as such and managed by the system. Enabling or disabling an accessibility service happens only upon an explicit user action in the settings menu.

Accessibility services send accessibility service info to the system with information about how they would like to be handled. An accessibility service can provide a specific feedback type and the system invokes only one accessibility service per feedback type, thus avoiding two accessibility services speaking at the same time. Android devices come with three accessibility services [4]:

- TalkBack - provides spoken feedback
- KickBack - provides haptic feedback
- SoundBack - provides audible feedback

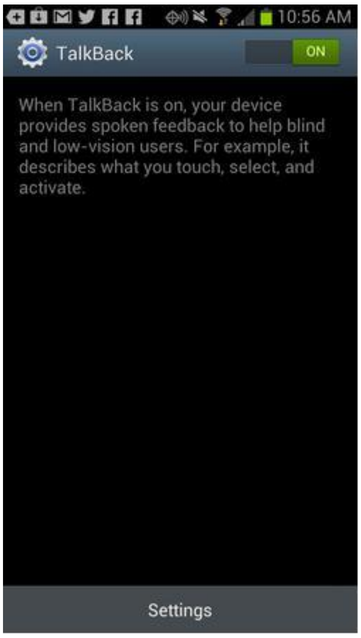
B. TalkBack Accessibility Service

TalkBack is a pre-installed screen reader service provided by Google for visually impaired users. It uses spoken feedback to describe the results of actions, such as opening an app, and events such as notifications. TalkBack cannot be uninstalled [3]. The "Blind Quiz" application completely uses the TalkBack Service to be accessible for visually impaired people.

How to Enable TalkBack

- Tap the menu button on device
- Tap Settings
- Tap Accessibility
- Tap TalkBack
- Switch the slider to On
- Adjust the volume on the device to suit the user

There are numerous settings that can be managed with TalkBack. There is a setting called Explore by Touch. On devices running Android 4.0 and higher, you can touch your device's screen to hear the content under your finger spoken aloud. It provides a special manner to the blind user to work with the touch screen. This is the way how it responds to the user touch.



When TalkBack is on, your device provides spoken feedback to help blind and low-vision users. For example, it describes what you touch, select, and activate.

Fig. 2. TalkBack application.

- Place the finger on the screen and move it about
- When the finger lands on a screen item, TalkBack will speak the name of the icon
- Once the user find the item he/she want, lift the finger and tap in the same spot to enable it
- To scroll through screens do so with two fingers

C. Web Accessibility

The "Blind Quiz" application needs to access the Moodle web content. Therefore it is not sufficient that only application content responds to TalkBack service. Furthermore the web content should be accessible to the user as the application content and Moodle web content are two separate contexts. Android provides Web Accessibility Service under this purpose. It allows apps to install scripts from Google to make the web content accessible.

Every web browser does not support to TalkBack service e.g. Opera. Chrome OS supports to TalkBack service as it is a Google product. So to provide impartial service from the "Blind Quiz" app, it is using Android WebView as the web browser. Android WebView allow displaying web pages inside the application. It is powerful enough to execute JavaScript. It allows loading HTML content as String and also displaying Web Page from domain/URL.

Prior to Android L Preview, WebViews with JavaScript enabled required an internet connection for spoken accessibility to work correctly. This is because WebView relied on a JavaScript (JS)-based screen reader (ChromeVox) that may not always be available in WebView cache. But in L Preview and later have a native implementation to be spoken the web content by the TalkBack service.

D. Jsoup

The "Blind Quiz" application needs to access the web content elements of the Moodle to retrieve the information of Moodle quizzes. Jsoup library was used for that purpose. Jsoup is an open source java library which provides a very convenient API for extracting and manipulating data, using DOM, CSS, and jQuery-like methods. Jsoup allows scraping and parsing HTML from a URL, file, or string and many more. It establishes a non-solid connection to retrieve data from a URL.

The major strength of jsoup is how HTML is parsed. When scraping and parsing an HTML page, jsoup provides a method for connecting a URL, file or string. The "Jsoup.connect (String url).get ()" method was used to fetch and parse a HTML document.

III. SYSTEM DESIGN

A. Basic System Architecture

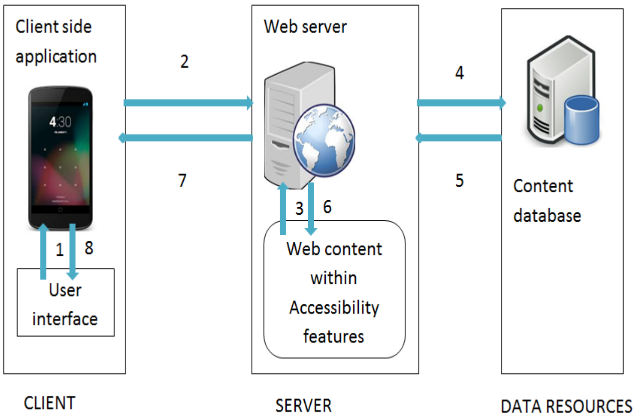


Fig. 3. Basic System Design.

Fig. 3 provides basic system architecture of the system. The "Blind Quiz" app provides the front-end, client side application of the system. It provides the Android Interface to access Moodle quizzes. Accessing the "Blind Quiz" app, the web accessibility enabled web content scripts can be installed from the sever side. Then the blind user can attempt the quizzes. As the Android WebView facilitates application built-in web browser, the user can directly connect to the Moodle quiz through a solid connection. So connection to the Moodle database also is done through a solid connection. Since the user can attempt the quiz following normal procedure.

B. Design Details

1) Static Structure: The "Blind Quiz" app is designed to run as a background service as it is easy to launch the application for blind users. When the device is boot-up the BootReceiver identify the signal and start the OpenAppHandler service. The OpenAppHandler service launches the "Blind Quiz" application's Main Activity which handles the main part of the system.

2) Behavior Structure: The behavior structure represents how the user interaction with the system goes on. After establishing connection the user can interact with the system to access the Moodle quiz.

IV. IMPLEMENTATION

A. Validating and Ensuring URLs related to Moodle

It is necessary to validate the user entering URLs are in standard format of a URL and ensure that it consists of Moodle URL. If not, the user will tend to enter incorrect URLs as input. So it is very important condition to be checked.

If the input is acceptable as a URL, then it is checked to ensure as Moodle URL or not. To ensure it, the input URL is connecting using jsoup and loads the HTML document and check the meta- data of the page.

If the variable state value is set to true, then the input URL can be accepted and stored in the internal device storage for further process for user. So the user does not have to enter a URL which has been added earlier.

B. Check Network Connection

The "Blind Quiz" application requires a network connection for all the time as it needs to connect Moodle. Therefore it is very important to check the network connectivity when connecting to the Moodle. It informs the user whether there is a network connection or not. So the user can take the correct action upon provided details.

C. Login to Moodle

After validating correct URL and ensuring presence of the network connection, the next step is login to the Moodle to retrieve correct data. As the jsoup library was used to retrieve necessary data, jsoup was used to implement login function as well.

First the necessary data which need to login was retrieved using JavaScript Console. Then those data was used to authenticate the app with Moodle through jsoup connection.

As jsoup does not build a solid connection it requires the session id to keep up the session until retrieve the necessary data about quizzes.

To check whether the user successfully logged in or not used inspecting elements of login page. The user credentials can be validated using that method.

D. Retrieve Quiz Data

After successfully login to the Moodle, the next step is separate quiz details and displays them to user. Jsoup HTML parser was used for that purpose as well. First a Jsoup connection is built to the Calendar page of the Moodle and then retrieves the details about events for the selected date.

After getting the event list related to selected data, the next step is separating the quiz events from the list. If there is a quiz it will be displayed in the next activity.

E. Attempting Quiz

Up-to now, jsoup did not create and establish a solid connection, instead it grabbed the data that we needed, and quit. A solid connection is needed to attempt the quiz. The "Blind Quiz" app goes to quiz page navigating through pages on the WebView by using JavaScript. When the WebView is launched initially, the user has not yet logged in , therefore it arrives at the login page. As valid credentials can be obtained using jsoup, those credentials can be used to login programmatically to attempt the quiz using a solid connection.

At this point it clears the cache and history of the webview, as well as the cookies of the present activity. This is important because it ensures no traces of the web activity are left behind. This prevents potential security errors arising from stored usernames and passwords over staying their welcome for other people to use.

V. EVALUATION

A. Unit Testing

1) MonkeyTalk IDE: For unit testing, test automation tool called MonkeyTalk IDE was used. MonkeyTalk is an integrated environment for recording, customizing, running and managing test suites. These are some features of MonkeyTalk IDE.

- Free & Open Source
- Powerful and robust record and playback functions for mobile applications.
- Fully cross platform
- Run on real hardware
- Readable scripts
- JavaScript scripts
- Data-driven scripts
- JUnit-compatible XML reports
- HTML reports

2) Unit Test Plan: Each single unit of the "Blind Quiz" system was tested using MonkeyTalk. It provided an easiest test plan for unit testing. It provides a table view to enter data to test for each case.

B. Accessibility Testing

In addition to Accessibility service, the app was tested against user interaction controls. At that time, the fact that the size of UI controls should be increased as the app is aimed to blind users was revealed. Also the app was tested within TalkBack service and Web Accessibility service.

1) Testing with TalkBack: The TalkBack accessibility service works by speaking the contents of user interface controls as the user moves focus onto controls. This service should be enabled as part of testing focus navigation and audible prompts.

To enable the TalkBack accessibility service:

- Launch the Settings application.
- Navigate to the Accessibility category and select it.
- Select Accessibility to enable it.
- Select TalkBack to enable it.

The whole application was tested within each UI control against the TalkBack service whether it gives correct feedback or not for each user interaction.

2) Testing with Web Accessibility: After attempting the quiz the user should be able to access the quiz content via TalkBack service. Following are the steps which are used to test that capability.

a) Build Moodle

As the Moodle is an open source Learning Management System, the source code could be obtained from official web site [8] and using it, a Moodle could be hosted in the machine which was used for development. As the developer was the administrator of the hosted Moodle site few course modules and quizzes to those course modules could be added. Then the user profiles were created where those users can attempt those quizzes under the given permissions.

b) Connect to the local host Moodle via Android device

As the Moodle which was built for testing purpose is hosted in local host, the android device cannot load the Moodle test through internet. Therefore it was required to build a network via Wi-Fi of local machine and android device. To create a Wi- Fi hotspot a third party software named mHotspot was used. It provides a convenient way to establish the network connection between two devices.

As mentioned earlier, to enable web accessibility it requires internet connection and it was necessary to share the internet connection through the Wi-Fi hotspot. For this purpose also mHotspot was used. Using these techniques, web accessibility could be enabled as it can install scripts from Google using shared internet connection. Finally a usability test was carried out by attempting quiz using Accessibility service successfully.

VI. CONCLUSION

The Blind-Quiz App is an application that provides the accessibility to Moodle for the visually impaired users. In order to reach this objective proposed web crawler was used to obtain specific types of information from Moodle website and to load in an android-friendly format. More specifically jsoup java HTML parser was used to retrieve the public information available on Moodle web page. Jsoup allows an easy navigation and a better data structure as presented in this project.

During the implementation of this project several limitations were encountered. As Jsoup does not provide a solid connection, sessionId called MoodleSession on the server side had to be used. This was done by creating a variable that saves sessionId when the URL is fetched in Jsoup connect and using that sessionId navigated through the web pages. This app is build aiming to the 2.7 version of the Moodle. As the future work the app is supposed to be improved so that this can be used for the old versions of Moodle.

In general, the main goal of this project was accomplished. A visually impaired user can attempt the Moodle quiz enabling Accessibility service in a convenient manner. Rather than using bit complex Moodle accessibility, a user can attempt quiz in an efficient manner using the Blind Quiz application.

VII. REFERENCES

[1] Docs.moodle.org, “MoodleDocs”. [Online]. Available: https://docs.moodle.org/27/en/Main_page.

[2] Developer.android.com, “Develop Apps | Android Developers”. [Online]. Available: http://developer.android.com/develop/index.html.

[3] Developer.android.com, “Accessibility | Android Developers”. [Online]. Available: http://developer.android.com/design/patterns/accessibility.html.

[4] “Text-To-Speech and Eyes-Free UI - Google Developer Codelabs” [Online]. Available: https://sites.google.com/site/gdevelopercodelabs/android/text-to-speech-and-eyes-free-%20ui.

[5] J. Hedley, “Cookbook: jsoup Java HTML parser”, Jsoup.org. [Online]. Available: http://jsoup.org/cookbook/.

[6] Cloudmonkeymobile.com, “MonkeyTalk | Mobile App Testing Tool”. [Online]. Available: https://www.cloudmonkeymobile.com/monkeytalk.

[7] A. Checklist, “Accessibility Testing Checklist | Android Developers”, Developer.android.com. [Online]. Available: http://developer.android.com/tools/testing/testing_accessibility.html.

[8] Moodle.org, “Moodle - Open-source learning platform | Moodle.org”. [Online]. Available: https://moodle.org/.