Development of a Mobile Application for Remote Speech Therapy

Guilherme Jorge de Lemos Marques
guilherme.marques@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

October 2019

Abstract

Communication is an essential part of human interaction. Aphasia is an acquired speech disorder that disrupts communication, and depending on the type, it can affect reading, writing and even speech comprehension. With the help of technology, patients suffering from this condition can have an easy and reliable way to complement their therapy. Currently there are two different prototypes available in Portuguese medium, which will be used as a baseline for this project. They already include a raw implementation of the required functionalities, however, they will need to be re-engineered to comply with current trends of software architectures and also suit today’s mobile user experience. The objective of this project is to build an Android mobile application for patients with aphasia (the target audience would be composed mainly by elderly people), so the application developed is user-friendly, responsive, and adaptable to different screen sizes.

Keywords: Mobile Application, Remote Speech Therapy, Aphasia, VITHEA

1. Introduction

Portugal’s resident population has been aging continuously, as a result of a decline in fertility and an increase in longevity [5]. Life expectancy of Portuguese population is increasing due to a rise in advanced age of survival. Life expectancy after 65, in the 2015-17 period, has maintained a positive trend, at 65 the average life expectancy is 19.45 years. Due to the aging population, and a lack of healthy lifestyle habits, there is an increasing percentage of the population suffering from strokes especially those over 65 years old [3], and around 30% of patients who have suffered from stroke are diagnosed as being aphasic [2]. Consequently, aphasia will most likely become more prevalent in aging population.

Aphasia creates an inability to produce or comprehend spoken or written language, although the level in which it is presented differs. Aphasia is a chronic condition and thus cannot be cured. However with speech-language therapy, partial spontaneous recovery is possible. The extend of the recovery can vary, some abilities return after a few days or up to a month, but usually some residual disorders remain. To obtain the best results, frequent and constant speech therapy sessions are recommended for patients with aphasia [1]. People who suffer from aphasia need to rely on some form of gesture or image communication [9], therefore technology could be a valuable resource, to provide a fast and reliable way of complementing their therapy remotely.

This thesis presents a native Android mobile application called Virtual Therapist for Aphasia Treatment (VITHEA) 2.0 developed based on an existing mobile and web applications for aphasic patients and their clinicians. The application developed aims to complement standard speech therapy sessions by providing many speech exercises, to practice with a virtual therapist remotely. The virtual therapist is an animated character, that presents exercises and provides feedback on how the exercises were performed. The application is designed to be simple, user friendly and offer a personalized experience, making it an easy and intuitive speech therapy application for aphasic patients.

There are plenty of web and mobile applications that complement speech therapy, but the majority of them are in English language. VITHEA is a Portuguese project started with the aim of providing easy remote speech therapy through a web application system for Portuguese speaking aphasic patients [7]. However, such web applications cannot leverage device utilities which provide easier user input and output to provide an immersive and personalized user experience, as well as, it is a browser dependent (it takes an extra effort from
the user to type in the URL of the web application) which results in a bad user experience. Therefore, to extent and optimize the productivity of VITHEA’s noble cause, the system was extended to the Android platform via mobile application. Nonetheless, there is an existing mobile application which is regarded as a proof of concept only. It incorporates the basic features that are important for a correct and complete interaction, so it needs re-engineering to keep up with new software architectures and the new functionalities. Hence, a mobile application will be developed based on the existing VITHEA web application to address all the shortcomings, as well as, to add new features to existing prototype of Android mobile application. The new mobile application VITHEA 2.0, intends to be more interactive, intuitive, personalised and run seamlessly by taking advantage of the mobile features, thus encouraging a frequent engagement.

The main objective of this thesis is to develop a native mobile application for Android platform that resembles the functionality offered by an on-line browser dependent system called VITHEA, with the following main functionalities:

- Develop an application with two modules, a patient module, where the patient will be able to exercise and a clinician module, where the clinician will be able to monitor his/her patients.
- Integration of a virtual therapist with speech synthesis capabilities. The virtual therapist is developed as an animated character and is provided as an external library.
- Reproduction of different types of stimuli (audio, video, images) and easy navigation among them.
- Acquisition of the voice signal through the device microphone and communication with the server where the automatic speech recognition engine is located.
- Adaptation to screen rotation and different screen sizes including tablets.
- Add common functionalities such as settings menu, feedback prompts and other features to design a simple and suitable application to meet the current trend of user experience.

The remaining of this paper is organized as follows - Section 2 discusses the related web applications. Section 3 discusses the mobile applications which addresses aphasia in Portuguese language and Section 4 will explain implementation of client and server application of VITHEA 2.0 in detail. Section 5 examines the evaluation methods that was used used for measuring the performance and quality of the application developed and also presents the results achieved from the evaluation to verify if VITHEA 2.0 application is adequate or not for aphasic patients. Finally Section 6 concludes the work.

2. Related Works
This section presents the related work that addresses aphasia, by providing a remote virtual speech therapy. For this paper, only speech therapy applications in European Portuguese language would be considered and analysed.

2.1. VITHEA Web Application
"VITHEA is the first prototype of an online platform that incorporates Speech and Language Technology (SLT) for the treatment of Portuguese speakers with aphasia [7]." VITHEA is a web-based platform aimed to behave like a virtual therapist to help the user complement their training sessions by leveraging the predetermined exercises online. The user will just have to access the VITHEA online platform through any web browser, thus allowing them to practice exercises at home [7].

The following Figure 1 depicts an overview of the VITHEA platform and how it functions to enhance the speech recognition therapy for Portuguese speaking patients with aphasia. It is based on a client-server architecture, where the client side of VITHEA's platform uses Adobe Flash technology to take inputs from patients as they answer the exercises presented. These answers are then encoded and sent to a web application server (JavaServer Pages (JSP)/Servlet) for processing.

The virtual therapy process with VITHEA platform begins by showing a series of visual and auditory stimuli using the display and the speaker of the user’s device respectively. Then, the user records the answer using the microphone by recognizing and naming the content that is presented to him/her. The recorded input of the user is then encoded and sent to the server side in order to
verify the correctness of the user input. Once the server receives the encoded input, server sends it to Automatic Speech Recognition (ASR) system for decoding and generating a textual representation of the user's encoded answers. Next, it compares the decoded answers with the set of predetermined textual answer for a given question stored in the database system. When the system finishes verifying the answer, it will send the feedback to the user [7]. The ASR system is the backbone for VITHEA, as it can decode the user's encoded input.

Virtual speech therapy in VITHEA platform is done through a three-dimensional (3D) game environment with speech synthesis capabilities in a web-based application where users can try to recognize and name the content they are asked. There are two types of client modules in the web application, one for the patient and another for the clinician to create remote therapy exercises. Clinician’s module is the administrative module where a therapist can create and manage exercises and see how the users are progressing with privileges to choose the resources for the exercises. In the patient's module, there are a set of exercises presented by an animated character, called the virtual therapist. The answer is verified by comparing and matching it to a set of predetermined answers stored in the database, and the feedback (“very good” or “try again”) is sent to the user depending on the correctness of the answer.

Nonetheless, web application technology limits the extension of the framework to mobile devices to scale up. Web applications need constant updating to comply with the browsers updates. When building a mobile application there is a minimum version required, but from that point on, the need for updates is smaller and easier to manage. Web based system is restricted to browser’s features such as the back and the refresh buttons to perform even the most fundamental functionalities, as it cannot leverage the advance mobile gestures such as tab, double tab, swipe, pinch, hold, etc. Therefore, VITHEA extended its web-based system to mobile application which is discussed in next Section 3.

3. VITHEA Mobile Application

A native VITHEA Android mobile application was developed to complement and overcome the shortcomings of the web based platform and also to extend its service to mobile users, taking advantage of mobile based features and functionalities. Since it is a mobile application, the user can take his phone with him and practice speech therapy anytime, and anywhere making the application more accessible. However, the mobile application already developed is still in a early stage of development and not fully operational at this point. It needs software re-engineering to achieve today’s trend of User Experience (UX), which is the exact motivation of this thesis.

The application uses version 2.3.3 of the Android mobile operating system which was released on September of 2011, as of May of 2019 this version is only used by 0.3% of the users [4]. Due to the use of an old version and a lack of functionalities, it is necessary to improve it by producing a new mobile application that could be used by a broader user base and have more functionalities.

The application uses Representational State Transfer (REST) architectural style. In RESTful, application data is exchanged through Hyper Text Transfer Protocol (HTTP). Due to the need of exchanging complex data types that represents the system state information, the data objects need to be serialized into text, usually represented in JavaScript Object Notation (JSON) or Extensive Markup Language (XML). JSON was chosen as the data format for the exchanging the information between client and server. The serialization process takes place upon sending and receiving the data.

In the Client application, Spring for Android has been used. This is an extension of the Spring Framework that aims at simplifying the development of native Android applications. It includes a REST client that provides higher level functions and several conversion functionalities for the various data representations supported. It also provides support for integrating Spring Social functionality, which includes an OAuth based authorization client, although this part was not explored. The audio response from the user is acquired and recorded using the microphone; when the recording stops this audio is sent to the server through a RESTful POST request. Then the speech engine AUDIMUS processes the file and the system verifies if the answer provided was correct or not. Finally the result is returned to the user [8].

The developed prototype only incorporates the functionalities that are important for a correct and complete interaction, such as the integration of the recognition process, the virtual therapist character, and authentication. The native module of the virtual therapist, which is done in the game engine Unity, is exported and then integrated into the Android application by plugins. The exercises include some video, audio and text, but the application cannot display some of the videos and audio files due to them not being supported by the Android platform.

The following Figure 2 depicts the User Interface (UI) of the application. On the top of Figure 2 is the main menu where a patient would select which
category of exercises he would like to do, while the bottom shows an example of an exercise from the visual category.

![Image](image.jpg)

Figure 2: Stimuli for exercises (top), and visual exercise (bottom) [7]

The Authentication is implemented by a simple basic authentication over Hyper Text Transfer Protocol Secure (HTTPS). When accessing the system, the user writes his credentials which are then sent to the server. In each of the following requests, data is then stored in the client application for the entire execution time. The information sent to the server is encrypted, and added to the authorization header field. On the server side it is compared with the encrypted version of the same data that resides in the persistent storage support. If the credentials are correct, the user is granted access. The access restriction to a given resource is done at the configuration file level.

In the Server Side, Spring Security and Spring Web Model View Controller (MVC), were implemented. Spring Security is a non-intrusive framework that focuses on providing both authentication and authorization to Java applications, it is easily extendable and can meet custom requirements. Spring MVC is a framework that helps in the development of web applications and REST services.

Although the previous version of VITHEA mobile application was just a prototype, it included the possibility for the user to perform exercises for speech rehabilitation. Overall, the application was functioning, however there have been many developments in Android over the years and by today’s standards the application lacks features to improve the UX and functionalities to personalise it, like for example showing the progress to engage the user.

4. Implementation of VITHEA 2.0

VITHEA 2.0 mobile application aims to provide a native re-engineered mobile application (Android) to support the current functionalities of VITHEA’s web platform, as well as extend the features of the existing VITHEA Android prototype mobile application to provide a personalized remote virtual speech therapy for patients with aphasia in Portuguese medium.

VITHEA 2.0 client application includes two modules - one for patient and one for his clinician, these modules communicate with the server to access the therapy exercises. All the data from both the modules, as well as, the exercises are stored in a database. All the components of the server and client application will be discussed in up coming sections.

4.1. Client Application

The client application will be built with Android Studio, the official Integrated Development Environment (IDE) for the operating system. Some of the features of the IDE include, a fast and featured packed emulator, and a single environment where it is possible to develop code for all Android devices without having to build an Android Package (APK).

The Android mobile application developed has two modules: a patient’s module to help aphasic patients to recover their speech with VITHEA therapy exercises, and a clinician’s module to assist the patients and to keep track of their progresses enabling the clinician to guide their patients properly. Each modules has multiple activities; an activity is a crucial component of an Android application to provide the window in which the application draws its UI. The Android system initiates code in an Activity, making it the entry point for the interaction with the user.

To successfully access any service from VITHEA Web Services (VITHEAWS), a user must be authenticated. When a user (patient or clinician) launches the application, if an account already exists in the accounts list, the application will send the token to the web server, this request is usually fast. While the application is waiting for the authentication process to finish, an activity will be shown to the user with the logo illustrated in Figure 3(left). This is done in order to let the user know that the application is running, otherwise only a black screen would appear. It only takes few seconds to verify and load the next activity.
If the user does not have an account created with VITHEA already, in the Sign-In activity there is a link to the Register activity as shown in Figure 3(right). The link will take the user to create account activity from where it is possible to create an account as a patient or clinician as illustrated in Figure 4. When the account is successfully created, it will automatically save the email address and authentication token in accounts.

All the requests and responses are sent using HTTPS to ensure security. Sending the users password in every request to the server makes it vulnerable to possible malicious attacks or eavesdropping in non-secure connections. Therefore, to improve security and the UX without compromising the users' credentials, an authentication process was developed. The Account Manager is the built-in class for account management, it enables the application to remember and recognize a user. With this feature the username and password would only be asked once, removing the need for the user to input the email and password every time the application is opened. These Application Programming Interfaces (APIs) provide an easy way to fetch the user’s credentials. For security reasons, the password is never saved in the phone, just a token with which the user can access the services, so even if the phone is compromised the attacker would not know the users password.

4.1.1 Patient Module

Patient module comprises of different types of exercises such as visual (image + video), audio and text as shown in Figure 5(left), which are chosen depending on the patient’s needs or preferences. After selecting the type of exercise, a list of stimuli (different categories of exercises to choose from) is presented to the patient as illustrated in Figure 6(top). Selecting any stimuli would initiate and elicit the exercises accordingly, which are presented to patients by the chosen virtual therapist, as shown in Figure 6(bottom).

For any category (visual, text or audio) and stimulus, the UI for the patient module would display the exercise on the left and a chosen virtual therapist on the right as shown in Figure 6(top). There are three virtual therapists a user can choose from, namely Catarina, Filipe and Edgar. The response is then recorded using the patient device microphone, a file containing the audio will be sent to the server, in order to verify if the answer matches one of the multiple correct answers defined by the clinician. When a set of exercises is completed, a summary statistics of total number of stimuli, correct or incorrect, number of exercises, and the number of not attempted exercises are displayed as shown in the Figure 6(bottom) below.
Figure 6: UI of VITHEA 2.0 from visual category (top) and (bottom) Summary Statistics (bottom)

After the patient delivers a response and the server validates his/her answer, a green or red background is prompted to give an immediate visual feedback on the patient’s performance as correct or incorrect as shown in Figure 7 (top) and Figure 7 (bottom) respectively.

Figure 7: Feedback background when the response is correct (top) and incorrect (bottom)

The patient’s module contains new functionalities, each one has a different activity; this is done in order to improve UX. The following different buttons and functionalities were developed to enhance the user experience, they are common to all the categories:

1. Home : Home button allows users to return to the main menu, if they wish to change the category or stimulus of the exercise, or just want to change any of the settings. It will save all the exercises data and prompt a summary of the statistics.

2. Forward : This button allows users to go to the next exercise.

3. Repeat/Refresh : This button appears only at the end of a set of exercises, and it allows users to repeat that particular set of exercises again, as illustrated in Figure 6 (bottom).

4. Recording : The user can tap and hold the microphone button to record the responses, or slide left to cancel the current recording.

5. Number of Attempts : This is an optional setting, the patient can either turned it off or customize the number of attempts (1 to 5) by accessing the settings menu depending on how they want to challenge themselves. When the number of attempts is deactivated, it means the patient has an unlimited number of attempts to finish the exercises.

From the home activity as shown in Figure 5 (left), a patient can access the settings activity to configure, manage and customise the application to user’s preference as illustrated in Figure 8.

Figure 8: Settings Activity

Settings activity provides a useful set of options to improve user retention by helping them personalise the application and record their progresses. The functionalities of each menu are categorised into the following menus:
User: In this sub menu, the patient can manage their existing profile by accessing the Edit profile activity. This activity allows the user to change their password, full name, date of birth or sex. Before saving any of the changed profile settings, the user will be prompted with a message asking whether he really want to do it or not. In case of changing the password, the user will get another authentication token, which will be stored into the Account Manager replacing the old authentication token. This is done to maintain the consistency with the user authentication process, removing the need to input the email and password.

In this sub menu, the patient can also view their statistics, illustrating their performance. If the patient wants to start from scratch it is possible to erase all the statistics, this will only affect the user, the clinician will still be able to see all the statistics from his monitored patients.

Next in the same user sub menu, the patient can also change the number of attempts. The patient can either deactivate the number of attempts, or set number of attempts from one to five according to their preference, by sliding on the seek-bar as depicted in Figure 8 (left).

Sound: This sub menu bundles features such as microphone testing, where users can test if their microphone is working properly or not by recording audio, which will be played back. The other option in this sub menu as shown in Figure 8 (left) is used to activate or deactivate play back the voice which was recorded, while doing exercises. This way the users can analyse their response and verify the audio that was recorded.

Text: In this sub menu, the user can change the text size as per their liking by sliding on the seek-bar as depicted in Figure 8 (right). This was added due to the target audience of the application being elderly patients, with age the eyesight gets worse so this option helps mitigate that, by proving a huge font as needed.

Under Text sub menu, the user can choose their preferred language, English or Portuguese, this feature was added in case VITHEA ever expands to new languages.

Choose Virtual Therapist: This application is updated to accommodate three virtual therapists. The users can personalise their application by changing the virtual therapist to their favourite one illustrated in Figure 8 (right).

4.1.2 Clinician Module

In this section, the clinician module will be discussed thoroughly. As in the patient module, if a clinician wants to access the application, he will have to authenticate. After a successful authentication, the clinician will be presented with the following home activity, as depicted in Figure 9 below.

From the Home Menu activity of the clinician module, there are 4 activities (Send Exercises To Patients, See Patients Statistics, Patient Management and Edit Profile) that the clinician can select from.

The first two items (Send Exercises To Patients and See Patients Statistics) will prompt the next activity, as shown in Figure 10, in order to select one, many or all patients. Only the patients that accepted being monitored by the clinician will appear on this selection.

Choose Virtual Therapist: This application is updated to accommodate three virtual therapists. The users can personalise their application by changing the virtual therapist to their favourite one illustrated in Figure 8 (right).

If the activity selected was "Send Exercises To Patients", after identifying the patients the clinician will now be able to select which exercises he wants his patients to focus on. The exercises are divided into categories and subcategories shown in Figure 11 (left). In each subcategory it is possible to select how many exercises the clinician wants to suggest his patients to perform. If the clinician does not want to specify an exercise he can simply
write a number in the label near "Qualquer Exercício" ("Any Exercise") and an algorithm will choose. The selection of both the exercises and the number of exercises in each will be random. For example if the clinician decides to send 50 random exercises, the algorithm will pick a random sub category and assign a random number of exercises between 1 and 50, it will end when there are no more exercises to distribute.

After sending the exercises, the clinician will be prompted with the result of the action, i.e. if the server was able to execute and store all the values or it failed. Using the home button (on the right corner) the application will return the clinician to main menu.

On the other hand, if after selecting the patients, the original activity selected was "See Patients Statistics", then the activity will appear as shown in Figure 11 (right). The clinician will be able to now see how many exercises the patients have been doing, and what is the ratio between right and wrong answers.

The option “Patient Management” allows to access to the activity shown in Figure 12. As with the previously explained activity, in this one the patients shown are those whom have accepted to be monitored. If the clinician ever wants to remove any of his patients, there is a button next to each of the patients’ names that will allow him to do so. In order to add new patients, the clinician must know the email address that the patient registered with.

4.2. Server Application
For the server application, in order to implement web services REST was used.

REST is an architectural style, its architecture is based on resources and interfaces. Any information can be a resource and they share a uniform interface to transfer the state between the client and server. The results from REST is easier to be read by humans, and it is also flexible and easier to build. In terms of security, since the requests go through HTTP or HTTPS, the firewall can verify each message by analysing the HTTP command used.

Regarding data representation with REST, two standards were evaluated, JSON and XML. Both the data representations can describe and facilitate the transport and consumption of hierarchical data structures. However, JSON has a simpler syntax than XML, because XML is a markup language and it requires more data for the same amount of information, therefore creating an unnecessary overhead. Another advantage of JSON is that, it is significantly faster than XML [6]. Hence, for implementation of VITHEA 2.0, JSON is chose for data representation.

4.2.1 Server Architecture
The implemented architecture remained the same as the previous VITHEAWS, as shown in Figure 13.
because the main components (Apache Tomcat, Spring, Hibernate, and MariaDB) are still one of the industry standard for web servers. Only the version of the components were updated.

Apache Tomcat is used to host Java web applications, in this case VITHEAWS. Maven helps to deploy the web server to Tomcat, Maven is a project management tool that builds the server application, and afterwards produces a .war file, which is then deployed to Tomcat. VITHEAWS uses Spring - to develop REST APIs, to add security, and is also responsible for the communication with AUDIMUS. Spring can also integrate with Hibernate. Hibernate is an Object Relational Mapping (ORM) used to map an object-oriented domain model to a relational database, it is responsible for the communication with the database, in this case MariaDB, this is where all the data is stored.

All the main components will be discussed in detail in this section.

4.2.2 Re-engineering

VITHEA previous architecture was developed around 6 years ago, making it a big span of time in which all the technologies used are now outdated, due to the rate at which all the technology is progressing today.

Table 1 shows the list of updated libraries, in this project. Apache Tomcat was also updated to version 7.0.96, this version increases the stability of Tomcat and adds security. Other minor libraries were also updated.

### Table 1: List of updated libraries

<table>
<thead>
<tr>
<th>Library</th>
<th>Previous</th>
<th>Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>Spring Framework</td>
<td>3.2.0</td>
<td>5.1.9</td>
</tr>
<tr>
<td>Spring Security</td>
<td>3.1.3</td>
<td>5.1.6</td>
</tr>
<tr>
<td>Hibernate</td>
<td>3.3.2</td>
<td>5.3.10</td>
</tr>
<tr>
<td>Jackson</td>
<td>1.9.11</td>
<td>2.9.9</td>
</tr>
</tbody>
</table>

In the original VITHEA platform, there were two different web applications, one for the patients (VitheA), one for the clinicians (VitheAdmin). In order to not have all the code duplicated and have to manage two code bases, VITHEAWS would use a library with code from VitheAdmin, this way if the project changed, the developer would only have to create a new library and import it into VITHEAWS.

However, with VITHEA 2.0 the patients and the clinicians both share the same mobile application, while also sharing more business logic, for example, with the clinicians being able to add patients. Having two separate projects would be harder to maintain, and it would lead to duplicated code in both the projects. This approach would have been very time consuming as well, especially while building the new APIs, due to the substantial development in both the clinicians and the patients business logic. Due to all the aforementioned advantages, both project were merged into VITHEAWS.

Updating the above mentioned libraries and merging both projects required considerable refactoring of the code. While updating the libraries there were also deprecated methods, these are methods annotated with @Deprecated. The developer is advised to change these methods. Some of the best practices with the main components also changed, especially the integration between Spring and Hibernate.

5. Evaluation

This chapter discusses the evaluation of VITHEA 2.0. This project evaluation, will determine if the new VITHEA 2.0 mobile application developed has achieved its intended goals and outcomes to ensure that it performs as it is suppose to. To evaluate the performance of the application, user testing were performed to provide evidence on whether the UX of the mobile application was improved as intended or not.

Tests with system users is a crucial part in the development to validate the adequacy of VITHEA 2.0, as well as modify and improve different aspects of the application as per their feedback.

There were a total of 20 participants (6 out of 20 participants were over 65 years old) and they were divided into pairs- each one testing a module (patient or clinician) and interacting through the application. The tests gave feedback on how real users uses the system. The following lists show the different tasks performed by participants as clinician and patients respectively.

**As a Clinician:** (i) Register in VITHEA 2.0 as a clinician; (ii) Register a patient to monitor; (iii) Send exercises, including 10 random exercises, to the patient; (iv) Verify the statistics of the patient after he finished the exercises; and (v) Change the
name, and another attribute in the profile.

**As a Patient:** (i) Register in VITHEA 2.0 as a patient; (ii) Select a category of exercises; (iii) Select a subcategory of exercises, and complete 15 exercises; (iv) Change the virtual therapist; (v) Change the number of attempts; (vi) Select a subcategory of exercises that the clinician sent; (vii) Check the statistics after finishing the exercises; and (viii) Change the password, and another attribute in the profile.

The results of this survey were remarkably good as shown in Figure 14. The application was rated as excellent by more than 50% of the participants, on four questions out of five. However, the main area of discontent is the virtual therapist which is out of scope for this thesis. The main complaints of the participants were that in some exercises it was not clear what was the intended answer, while others complain that the virtual therapist was hard to understand.

6. Conclusions

VITHEA 2.0 successfully provides a native re-engineered Android mobile application to support the current functionalities of VITHEA platform, as well as extend the features of the existing VITHEA Android prototype mobile application. It also re-engineered all the outdated software and technologies, and added new functionalities to provide users more leverage to personalize UX making remote virtual speech therapy more immersive for patients.

VITHEA 2.0 mobile application kept the same client-server architecture like its predecessor. The client application was developed in Android studio and has two modules for patients and their clinician respectively, and the server application is based on previous VITHEA mobile Web Server which is a Java web application. The main components of the server application was kept as it is and only updated the versions, as those components are still relevant and one of today’s standard.

The client application presents therapy exercises as well as render patients an immediate feedback prompts after each exercise and summary statistics after each round of exercises set to keep track of patient’s therapy progresses. These statistics can be viewed and managed by patient’s clinician on his module so that the clinician can review his patients work and progresses. Clinician is also capable of refining patient’s therapy program by sending personalised exercises to patient’s modules in the area where they might need extra effort. At the server side, Apache Tomcat was used to host the VITHEA Web Server using Spring framework and Sprint was integrated with Hibernate to communicate with Maria DB (a relational database).

The evaluation of the mobile application with 20 participants has shown that the results of client application were remarkably good as the application was rated as excellent by more than 50% of the participants, on four questions out of five. However, the main area of discontent was the virtual therapist.

**References**


