5-1-2022

DOCTOR-PATIENT REAL TIME CHAT APPLICATION

Sri Subhash Pathuri

DOCTOR-PATIENT REALTIME CHAT APPLICATION

by

SRI SUBHASH PATHURI

Presented to the Faculty of the Honors College of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements for the Degree of

HONORS BACHELOR OF SCIENCE IN COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT ARLINGTON
May 2023

ACKNOWLEDGMENTS

I convey my heartful gratitude to Dr. Shawn Gieser for revising my initial topic and coming up with a concrete plan for this project. In addition, I am thankful to him for effectively guiding my team and me patiently throughout the senior design project period. With all his support, I was able to brainstorm effectively and develop a dynamic feature that impacted my senior design project to a new level.

The project could not have been completed without the support of my teammates, Rithik Kapoor and Dhruva Malik. We went through mainly all-nighters and conflicts and sacrificed other aspects to transform the initial pitch into reality. Even though we could not meet regularly for a couple of months due to the pandemic, we were able to keep with the expectations and have detailed documentation with the help of GitHub for tracking our progress and being compatible with a plethora of android devices.

May 1, 2022

ABSTRACT

DOCTOR-PATIENT REALTIME
CHAT APPLICATION

Sri Subhash Pathuri, B. S. Computer Science

The University of Texas at Arlington, 2022

Faculty Mentor: Shawn Gieser

During the pandemic, the majority of the working population spent most of their time doing sedentary desk jobs. As a result, they may have developed inadequate spine posture, which can cause profound implications such as back pain if not corrected at an early stage. Implementation of machine learning and computer vision has allowed us to capture and analyze video input from the user's cellphone to remind them whenever they are not in the correct position. To further support users, a chat feature has been implemented through which the users can quickly get in touch with orthopedic experts for evaluation and feedback. With the help of strategies from kinesiology, a comprehensive analysis has been conducted to find the root cause of abnormality in the spine. As a result, anyone can instantly monitor their posture and make minute corrections to recover swiftly after consultation or a medical operation.

TABLE OF CONTENTS

Appendix

LIST OF ILLUSTRATIONS

CHAPTER 1

INTRODUCTION

1.1 Background

Throughout the pandemic, most of the working population started working from home remotely. After understanding multiple surveys, it has been proven that people are more alert and have better composure in the first half of the day, and as the day passes, these levels begin to drop until they stop working. Many people spent an increasing amount of time doing desk jobs and have a poor seated posture, which could lead to many health issues like back pain, headaches, and spine curvature. Bad posture leads to not having confidence, which further leads to low self-esteem. This decreases the productivity of the employee, which is a great loss for the organization.

We aim to develop a unique application that will support people in improving their posture by using the cellphone's camera to monitor the user while they are seated and notify them whenever they get into a bad posture. Although there are various other posture correction applications on the Google play store or iOS application store, almost all of them involve some expensive tracker placed on the body and do not seem to use the vast capabilities of machine learning and computer vision. The spine is an essential part of the body. Without it, we cannot keep ourselves upright or stand up. It provides us the body structure and support. With the aid of the spinal cord help, we are able to move freely, and it is designed to protect the central nervous system. Without that, we cannot properly move; hence keeping the spine healthy is vital to living a happy life. This is the main idea behind why we wanted to make a profound impact on everyone's lives by developing a product that is immensely beneficial.

1.1.1 Honors Project Responsibilities

As a part of the capstone project, additional components that are essential to provide the users with the best overall experience were worked on. Worked on connecting the doctors available in the local network to users seeking advice on specific therapy practices or post-surgery cool-down procedures through chat. Adding medical support with the help of online physicians can boost the morale of users and serve as an added incentive to use the platform for maintaining a healthy lifestyle. The general tips provided on the internet may not be suitable for all of us. Hence, its essential to provide users with the ability to connect with external resources such as doctors to guide them with respect to their specific health conditions. The real benefit of implementing this feature is that in our application, doctors can readily view the patient's history and view all the analyzed metrics swiftly.

1.1.2 Honors Project Tools

This Honors project utilizes the same tools as the rest of the application. Java was to develop the application in android studio. Firebase is used as the database for the backend. Google's ML Kit Vision Application was used, and it is the backbone of our application. In Java, I used to recycler view for instantly populating the messages in the conversation in real-time based on the internet traffic and flow of data through packets. All the messages would be stored as unique components with their identification tagged with the user in the cloud platform. Accessing real-time data was efficient by incorporating the

Firebase SDK. Once all the instances of firebase applications were set up, reading and updating the database was a seamless process.

1.2 Value Proposition

There are various telehealth services available. However, the benefit of incorporating that feature into our application is that physicians can readily view the patient history and view the analyzed metrics of his back posture from any period of time, based on the patient's consent. This vastly reduces the amount of documentation and evidence needed for physicians to decide on the treatment right away and solve it in the early stages itself. Several times in society, the presence of spine and posture problems in younger people might have incorrect interpretations from their friends and family circle as to not exercising properly. With the help of this feature, the users can share all their feelings confidentially. At-home, post-operation therapy has not only proven to be clinically efficacious, but patients feel more comfortable communicating through the computer screen and smartphone than in person. Another important aspect is that it is crucial for the patient to feel at ease in their environment while performing the necessary exercises. And for that, there is nothing more comfortable than being in the place where you want to be. Even the expenses for gasoline are reduced with stress from traffic, making it a viable option to pursue.

CHAPTER 2

TECH STACK

2.1 Android Studio & Java

Although many excellent tools are available for mobile application development, Android Studio and Java stood out for several reasons. Android Studio was used to take advantage of certain performance advantages immediately. Java is considered one of the best programming languages for mobile applications. It is an object-oriented language developed by Sun Microsystem in 1995. In all other coding systems, the code is first translated by a compiler into instructions. In contrast, Java, on the other hand, turns code into bytecode, which is then interpreted by Java runtime environment, and is more efficient than the former. This is an easy-to-learn language with fluent English-like syntax and less complex jargon. It has a rich API, XML (Extensible markup language) parsing, database connection, networking, and utilities and provides almost everything that a developer can expect. This is open source and is also freely available.

Android has succeeded in keeping Java at the forefront in the last couple of years. For Android applications, you can use android APIs as well as Java to write code. These applications are called native apps, developed by utilizing native tools. Android Studio is the official integrated development environment for Android application development. It is based on IntelliJ IDEA, a Java integrated development for software, and incorporates its code editing and developer tools. This is much easier compared to cross-platform frameworks such as React and Angular, where state management becomes extremely complex. IntelliJ's company initially developed Android Studio, it offers a diverse set of features that enhance productivity when building apps, such as the flexible Gradle-based build system, that is a fast and feature-rich emulator and a unified environment where you can develop for all varieties of android devices. It has built-in support for the Google Cloud platform, making it easy to integrate Firebase and Google Cloud Messaging and App Engine.

2.2 Firebase

Firebase is one of the optimal services to utilizes in the backend domain due to its seamless integration and user-friendly interface. This platform provides services at multiple tiers to support all types of customers. First of all, it has its own proprietary backend platform to securely store all the user data, and secondly it has multiple services such authentication to effectively synchronize the backend platform and the application. This platform supports everything from databases and analytics to crashing reports; therefore, it is easy to debug applications multiple times to solve all errors. Throughout this app, Firebase was incorporated into this application, it is being used for secure authentication, storing chat in a real-time database and storing images in the Cloud Storage part. The best part is that one can instantly implement all the features in real time by connecting all data points together with a single unique UserID for each separate entry in the database.

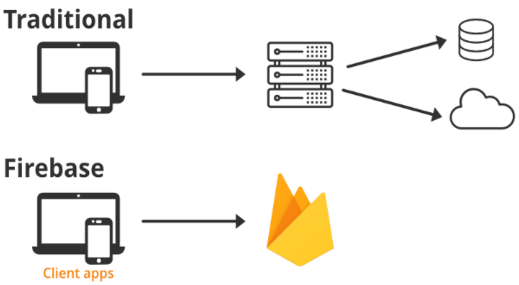


Figure 2.1: Firebase removes the need for special servers

CHAPTER 3

METHODOLOGY

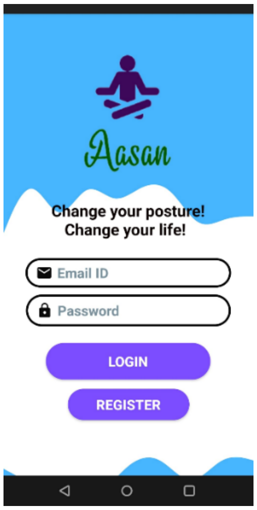
In addition to utilizing the machine learning models for posture detection, deep learning and neural networks were used to formulate the various back problems with the help of Kinesiology and other practices. A variety of situations were researched were where issues can occur and collect information also to provide the users during the consultation so that doctors can provide treatments by looking to the resources present from our side, without any further research and mainly using their profound experience.

Once all the information is collected, a comprehensive analysis was performed and came up with a well-defined flow as to which problem can lead to other serious issues if not taken care of and the respective side effects for each circumstance. In this way, we initially plan to learn and become well-versed with the basics of the various medical practices so that we can truly understand and build systems suited to those needs.

3.1 Authentication

The authentication part of the application consists of 3 parts, which is the SignIn page, SignUp page, and the Forget password page. All these pages contain the basic instances of all the design components, such as buttons from the aesthetic point of view. The functionality in Android studio works through the way of listening, each component waits to listen to the user interaction, and we created a step-by-step process informing the application what to do based on the type of interaction with the user. Hence, once the user types in all credentials, the application will observe their interaction.

When the submit button is pressed, the Listener function for that button will be activated. Once the button is activated, it will send a snapshot of the entered credentials and verify it with already stored details in the cloud and confirms if the user is legitimate or not. The passwords are not stored, as it is in the application. Every time a new user creates a password, the application uses an internally modified version of script to hash account passwords. Even when an account is uploaded with a password using a different algorithm, Firebase Auth will rehash the password the first time that the account logs successfully in. Accounts downloaded from Firebase Authentication will only ever contain a password hash if one for this version of the script is available or contain an empty password hash otherwise. They have mainly incorporated this to make sure the personal data of each user is in safe hands. In addition, they have created a progress bar that displays the loading symbol when the authentication process is going on. This process gets activated when a user clicks the submit button, and they have programmed it so that it only stops showing once the entire procedure with Firebase is complete as a way to be transparent with the user.



Change your posture!
Change your life!

Figure: 3.1: Login Screen

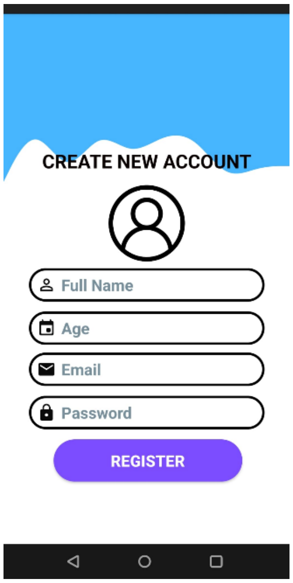


Figure: 3.2: SignUp Screen

3.2 Real Time Database

Once the authentication is complete, Firebase stores a copy of the credentials in the auth section of the platform, however, the central database where all the user and chat data is stored is the real-time database. This database was chosen because in any chat application, there is a need for bi-directional transfer of data. Firebase Realtime database allows applications to approach cross-platform data within real-time after joining NoSQL cloud-storage. Even piece is data is still being cached in the memory of a device when offline and start synchronizing after getting into a local café that has a Wireless internet connection. They can set customized data permissions at different levels so that all the data points are sealed, to prevent any malicious attacks online. The data in the cloud is arranged in a JavaScript Object notation (JSON) fashion. The system goes through multiple layers and obtain the required data by chaining the commands as shown below:

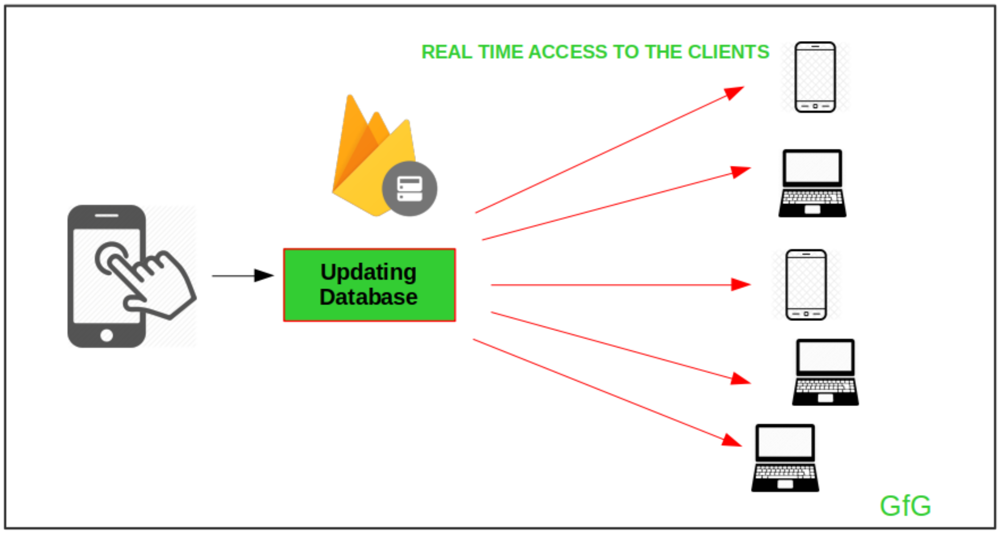
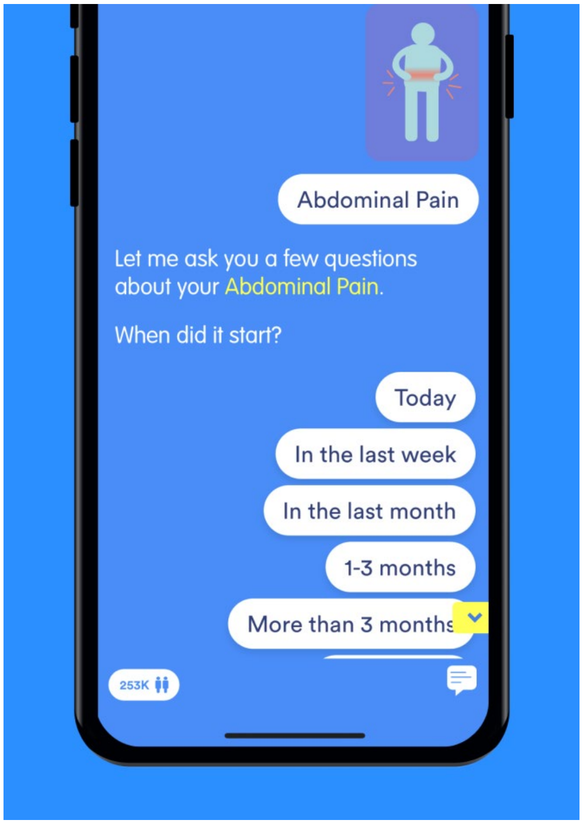


Figure 3.3: Functionality of the real time database

3.3 Chat Feature Implementation

It took an entire week to create a logical flow as to create an initial design of the software architecture. Once, the flowchart was ready, a development mechanism was chosen where any two users can stay online in the application at the same time concurrently. In addition, a couple of design components were created of each populated message icon and box so that each aligns to the respective edge based on each user's perspective. For example, in the first user's screen, their messages should appear on the right side of the application and the second user's messages should appear on the left side of the screen to differentiate.



Abdominal Pain

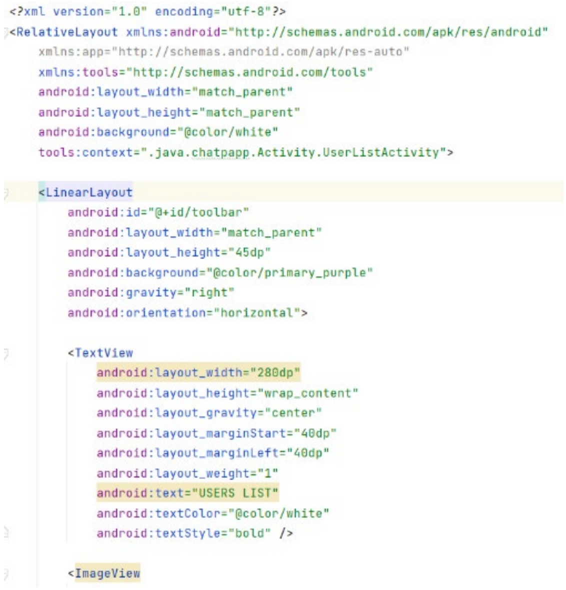
Let me ask you a few questions about your Abdominal Pain.

When did it start?

Figure 3.4: Chat feature where a patient can ask about any problem

This application used a recycler view in the application, as it is a flexible layout whenever a user opens the application, all the previous messages need to be re-populated every time. And this application has a loop such that the text extracted from the cloud will be enclosed in the message box components, and will be packaged together, to be displayed sorted by time from the bottom in the descending order. A special library called Picasso is utilized, to store the images and for downloading and caching library for android. Further, to update the database each time, a Database Reference instance was created to generate the most updated snapshot on demand to analyze where to update in the database based on the local timestamp of the device. To facilitate, the process of each message being properly arranged, a User class that consists of all the parameters, so that during each cycle, all essential details such as name, email, age, and all messages are updated together every time.

To decide which person to chat with, an addition screen was developed which was called the UserListActivity. This page consists of a Linear Layout which displays all user profiles and their images as a list of User Cards. In this way, we are able to understand which all users are online and offline. These cards are designed to appear dynamically if the number of users does not fit the screen at once, the user will be able to scroll infinitely.



```xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/white"
    tools:context=".java.chatpapp.Activity.UserListActivity">

    <LinearLayout
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="45dp"
        android:background="@color/primary_purple"
        android:gravity="right"
        android:orientation="horizontal">

        <TextView
            android:layout_width="280dp"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:layout_marginStart="40dp"
            android:layout_marginLeft="40dp"
            android:layout_weight="1"
            android:text="USERS LIST"
            android:textColor="@color/white"
            android:textStyle="bold" />

        <ImageView
```

Figure 3.5: A snapshot of how the user list activity is designed

CHAPTER 4

CONCLUSION

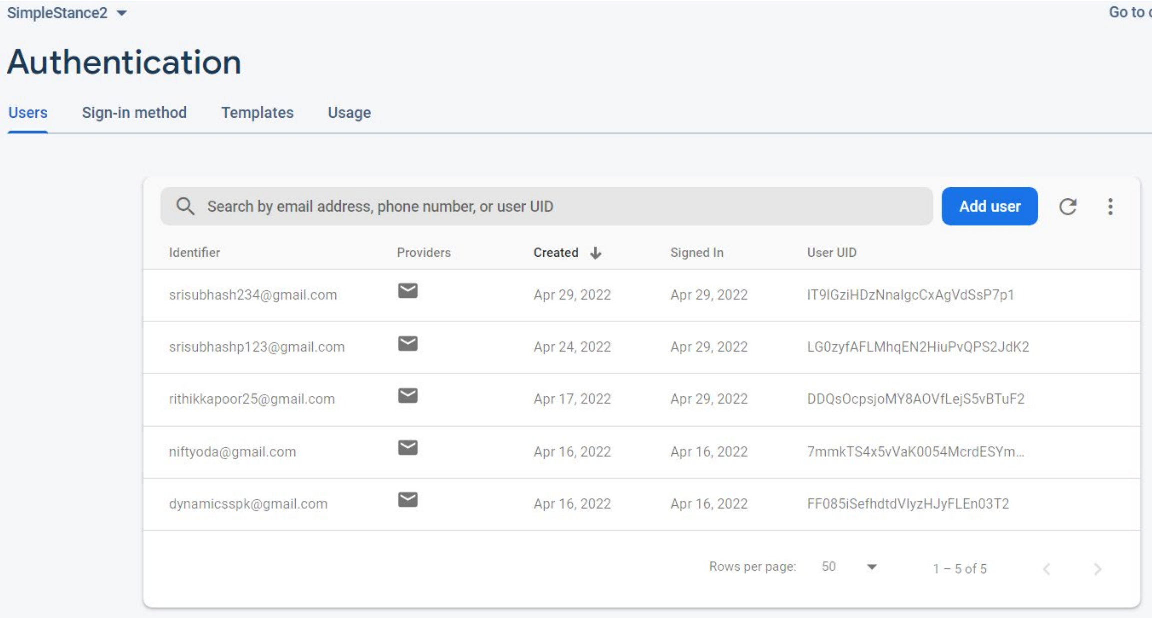
This project allowed me to realize my great potential which was untapped till now. Honors College faculty played a big role in encouraging me throughout the process. Android Studio and Firebase are both incredible tools that made this honors project manageable. The members of our team were quite familiar with the workload required to achieve all the required goals. We were surprised by our efficient co-ordination. Although, there was an initial learning curve with syntax and the logic of these tools, this quickly outweighed its benefits.

4.1 Android Studio Conclusion

The mechanisms of Android Studio are slightly different than the plain Java programming language. Each visible page of the application is divided into the two components, which are the functional and design components. This gave us additional flexibility to create custom connections between components to make the best user interface for the customers. The Picasso library was useful in easily displaying profile photos for each message in the chat screen for easy differentiation.

4.2 Firebase Conclusion

Firebase developed by Google is a marvelous product, which was extremely useful for our team. With other backend platforms, the developer needs to setup the location of the servers and their maintenance costs, and also mange load balancers. These all are taken care of automatically in Firebase. Hence, we were able to merge the Firebase Database, Authentication, and Storage platforms to facilitate in organizing all the messages based on each pair of users. Firebase authentication empowered our team to develop a unique project that was not only functional but was also secure and safe. Based on experience, everyone should kickstart their Cloud journey by taking full advantage of Firebase.



Search by email address, phone number, or user UID

Identifier	Providers	Created ↓	Signed In	User UID
srisubhash234@gmail.com	✉	Apr 29, 2022	Apr 29, 2022	IT9IGziHDzNnalgcCxAgVdSsP7p1
srisubhashp123@gmail.com	✉	Apr 24, 2022	Apr 29, 2022	LG0zyfAFLMhqEN2HiuPvQPS2JdK2
rithikkapoor25@gmail.com	✉	Apr 17, 2022	Apr 29, 2022	DDQsOcpsjoMY8AOVfLejS5vBTuF2
niftyoda@gmail.com	✉	Apr 16, 2022	Apr 16, 2022	7mmkTS4x5vVaK0054McrdESYm...
dynamicsspk@gmail.com	✉	Apr 16, 2022	Apr 16, 2022	FF085iSefhdtdVlyzHJyFLEn03T2

Rows per page: 50 1 – 5 of 5

Figure 4.1: Users would populate this way in Firebase when they sign up

APPENDIX A

ANDROID STUDIO CODE MANAGEMENT PRACTICES

1. XML layouts and AppActivity Screens**:** Creating of mobile applications involves two steps. The first part is the creation of screen filled with basic components with no functionality present, such as adding scroll bars, buttons, text views, image views, etc. The second part, which is the App Activity screens involve the Java code that dictates all the behavior of the activity screens based on the haptic feedback of the user.
2. OnClickListeners and User Adapter**:** All applications function based on our touch response on the screen with respect to the needs on demand. To facilitate the instant response when the user clicks anything, each widget in Android Studio has a system similar to a timer, which keeps on listening (waiting) for a specific way of touch. Once the user touches the button, and if his gesture suits the requirements, then that specific widget will direct you to the next task. In order to display a list of users, a card-based system was created whereby all data from Firebase gets downloaded and packaged in the form of independently combined widgets, which can be stacked side by side in real time. This flexible card-based system is known as the User Adapter.

REFERENCES

Kim, DeokJu et al. “Effect of an exercise program for posture correction on musculoskeletal pain.” Journal of physical therapy science vol. 27,6 (2015): 1791-4. doi:10.1589/jpts.27.1791

Jonaitis, Jenna. “12 Benefits of Good Posture - and How to Maintain It.” Healthline, Healthline Media, 14 Apr. 2020, https://www.healthline.com/health/fitness-exercise/posture-benefits.

“Why Posture Matters.” Harvard Health, Harvard Medical School, 24 Jan. 2017, https://www.health.harvard.edu/staying-healthy/why-good-posture-matters.

“Why Android Developers Should Prefer Android Studio over Eclipse?” *Digital Transformation Agency*, https://www.brainvire.com/android-developers-prefer-android-studio-eclipse/.

BIOGRAPHICAL INFORMATION

Sri Subhash Pathuri is an undergraduate student studying Computer Science at the University of Texas at Arlington. He had participated in two internships throughout his college career; one at Animal Cloud, which was a startup company, and the second one at Humana, as a software reliability intern. These internships enabled Sri Subhash to gain valuable experience and knowledge as how to truly contribute to a company in a healthy manner. Sri Subhash has also worked as an undergraduate research assistant at the Innovative Data Intelligence Research Laboratory. Working on multiple projects in the fields of machine learning, block chain, and computer vision, were useful in improving the skillsets and helped in winning the HackTX (UT Austin hackathon) in 2021. Sri Subhash plans to work for a couple of years to explore all areas and find the optimal field of interest. Further, Sri Subhash would return to obtain a master's degree in that field and try to advance further and contribute significantly.