****

**MOBILE APP DEVELOPMENT**

***App Name: Savology***

***Name:***

***ID:***

***Module Code: CIS4034-N***

**Abstract**

The Savology app has been developed with the integration of Android programming operations to continuously monitor the health conditions of individuals aged 40 to 60 years as this age has been recognised as the most crucial age for rising health issues. The mobile application has been developed with high accessibility and simpler design to provide strong assistance in promoting user interactions for the targeted users. Besides that, an agile development process has been followed in Savology to include a flexible development approach in the mobile application. In addition to that, the development of Savology app has prioritised data security, user privacy, and legal compliance while also taking social considerations like accessibility and health equality into account. In order to provide flexibility and responsiveness to evolving project requirements, Extreme Programming (XP) has been selected as the agile development approach. The use of digital environments and tools, namely Kotlin DSL, has been implemented to enhance the efficiency of development procedures and produce a superior cross-platform mobile application. Overall, in addition to guaranteeing user happiness, compliance, and efficient project management throughout the development lifecycle, the Savology app has been designed to encourage healthy living.

**Table of Contents**

[1. Introduction 3](#_Toc173521135)

[1.1 Brief Description of the App and Targeted Users 3](#_Toc173521136)

[1.2 Objective of the Report 3](#_Toc173521137)

[2. App demonstration 4](#_Toc173521138)

[3. Software design decisions 7](#_Toc173521139)

[3.1 Design and Layout 7](#_Toc173521140)

[3.2 Hardware Usage 7](#_Toc173521141)

[3.3 User Experience 8](#_Toc173521142)

[3.4 Data Persistence 9](#_Toc173521143)

[3.5 Third Party Library Usage 9](#_Toc173521144)

[3.6 Program Code for Mobile App Development 10](#_Toc173521145)

[4. Social, legal, ethical and security issues 28](#_Toc173521146)

[5. Selection, use, and evaluation of an agile development method 29](#_Toc173521147)

[6. Use of digital environments and tools 32](#_Toc173521148)

[7. Conclusion 33](#_Toc173521149)

[References 34](#_Toc173521150)

# 1. Introduction

## 1.1 Brief Description of the App and Targeted Users

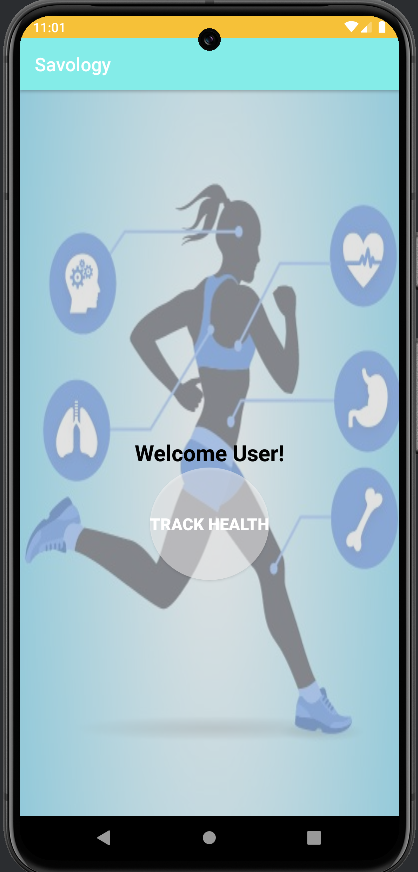
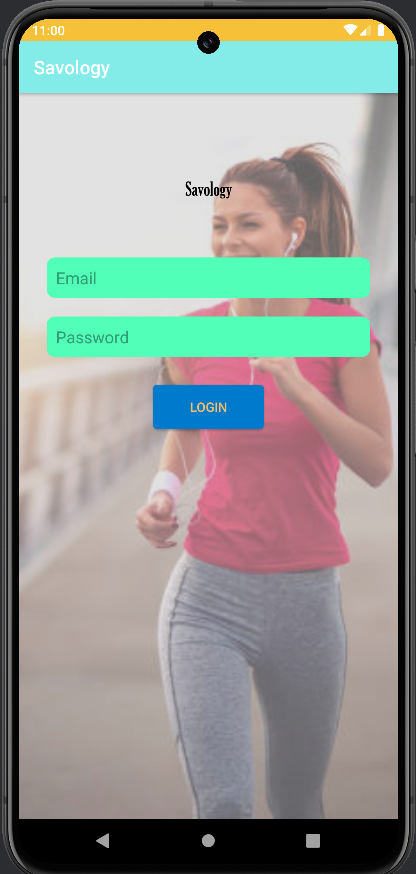
Savology is a mobile application operated on the basis of Android programming operations to keep track of the changes in the human body. The application will estimate changes in human bodies at a regular time interval to ensure an appropriate health condition of individuals. In addition, this application can promote healthy concerns of individuals leading to avoiding any complex health issue. The application will be utilised by the users aged 40 to 60 years irrespective of male or females as individuals can be affected by any disease in this age.

## 1.2 Objective of the Report

The goal of the report is to provide a thorough account of the reasoning and development process used to create an Android application named as Savology. This report will include an in-depth overview of the idea of the app, its targeted user base, technological setup, and compliance with ethical, legal, and professional norms. Social, and security issues, including adherence to GDPR guidelines and licensing requirements, will all be covered in the study. Moreover, it will also cover the use of digital technologies for version control and project management, as well as the selection, use, and assessment of an agile development methodology.

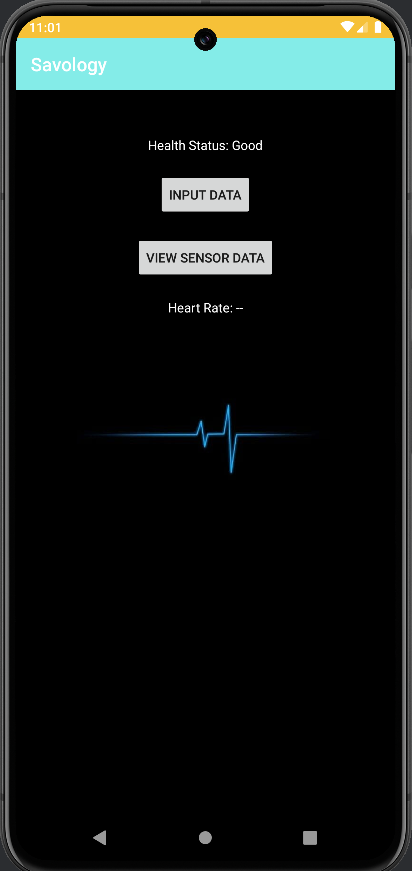
# 2. App demonstration

Viewers are led through the main features and functionalities of the Savology app during the demonstration to get a thorough grasp of its potential.



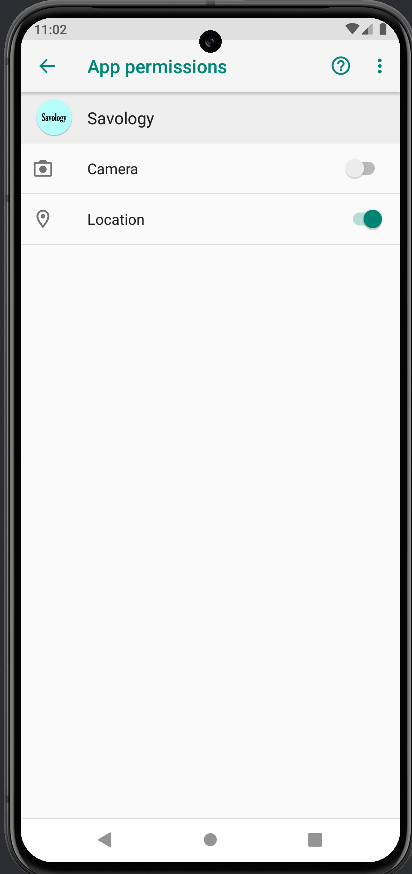
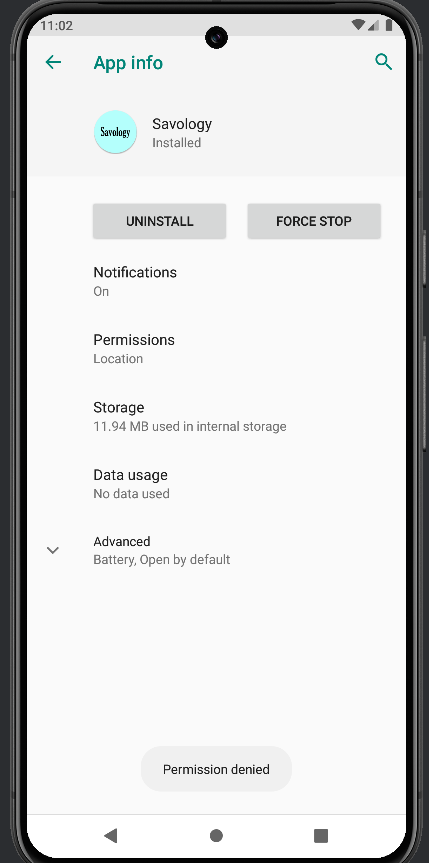
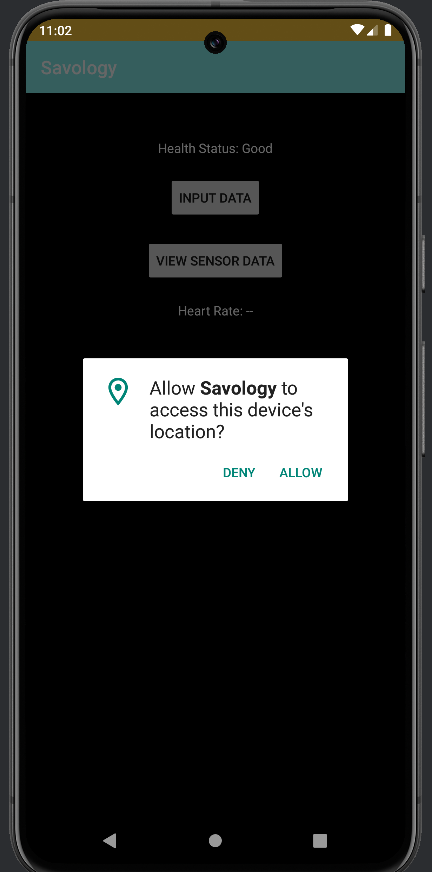
***Figure 1: Login and landing screen of Savology***

After a successful login, the user can probably access the functionality of the app by clicking on the "Welcome User!" and "Track Health" buttons ***(Refer to figure 1)***. Thus, the login page appears to be simple and easy to use, assisting users in submitting their credentials and gaining access to the features of the app. After then, the emphasis switches to particular features like keeping track of food habits, measuring physical activity, and observing sleep patterns.



***Figure 2: Home Screen***

In this case, the "Health Status: Good" denotes that the health of the user is good as of right now, based on the app's analysis of the app and collection of their health data. The "INPUT DATA" option most likely enables the user to manually enter health-related data into the app, such as blood pressure, weight, and blood sugar levels ***(Refer to figure 2)***. Besides that, the "VIEW SENSOR DATA" button can display information gathered from wearable sensors, like a fitness tracker or smartwatch.



***Figure 3: App permission process***

The Savology app emphasises the value of security and privacy. The application is specifically requesting permission to access the location of the device and the choice to "Deny" or "Allow" location access is presented to the user ***(Refer to figure 3)***. Overall, by giving viewers a thorough rundown of the features, capabilities, and user interface of the Savology app, this highlights the importance of the app in assisting users in tracking and enhancing their wellbeing.

# 

# 3. Software design decisions

## 3.1 Design and Layout

A simpler design has been developed with the integration of Android programming operations for different app pages such as Login, landing, details verification and others of the mobile application, Savology. It would be easily accessible by users of 40 to 60 years to monitor their health conditions.



***Figure 4: Logo of the App***

The mobile application has a unique and definite logo to make differences from other health monitoring applications for individuals.

## 3.2 Hardware Usage

In order to develop mobile applications, proper programming operations can be obtained with the help of engaging hardware features. The interactive assumptions in developing activities of a mobile application can be integrated with the help of including hardware features on mobile applications. In this regard, the desired mobile application has been developed with the application of cameras, breathing sensors and others to check the health condition of individuals.

In the mobile application, different mobile hardware features have been properly integrated. The mobile application has used the integrated sensors and cameras to obtain the necessary real-time data in analysing the health condition of users. As stated by Majumder and Deen (2019), mobile-integrated cameras and sensors can be properly used in data collection processes for monitoring health conditions of individuals such as heart rate (HR), the variability in heart rate (HRV), and others with the help of tracking live movements and images through a mobile application. In this regard, the Savology app has used mobile cameras and integrated sensors to obtain real-time health-related data of users appropriately. As mentioned by Deng *et al.* (2023), mobile sensors such as breathing sensors, Global Positioning System (GPS) receiver, temperature sensor, and others can be used in checking different health factors such as oxygen intakes, location and movements, and human body temperature of the users of a health monitoring mobile application. The Savology app has been linked to a GPS receiver to identify the current location of the engaged users to analyse their movements and utilised energy levels. As a result, the mobile application can analyse the total step counts of users as maintaining their health routine on a regular basis. On the other hand, the checking of oxygen intakes and oxygen content in the blood have been properly monitored by the Savology app through an oximeter integrated in the mobile device. The lung condition of human bodies can be properly measured with the help of estimated oxygen levels in blood contents and air intakes.

In addition to that, sound sensors are also integrated with the mobile application to include proper demonstrations of health conditions of the users. As described by Zeng *et al.* (2020), ultra-wideband noise sensors can be utilised in health monitoring apps to determine heartbeats accurately. Similarly, the Savology app has been effective at analysing heartbeats with the help of sound sensors integrated in the mobile device.

The integration of cameras of mobile devices can be highly beneficial for checking different health conditions such as eye condition, heart rate, health of lungs, and skins with the help of engaging numerous programming operations on analysing images accumulated using those integrated cameras. As per the study conducted by Han, Kang and Lim (2023), different physiological signals related to health conditions can be analysed with the help of captured images of human face and body to understand the current situation of health conditions of individuals. In this regard, the developed mobile application has been directed to measure health entities in terms of heart issues, skin conditions, eye condition and others with the help of captured images through mobile devices. Hence, it can be stated that in the development process of the intended mobile application, sensors, cameras, and other mobile hardware components have been incorporated.

## 3.3 User Experience

The user experience (UX) of the Savology app has been carefully designed to guarantee a smooth and simple interaction for users between the ages of 40 and 60, irrespective of their gender. The design principles prioritise accessibility, simplicity, and clarity to accommodate a wide range of users with different degrees of technical expertise and health consciousness. The simple and clear layout of the app makes it simple for users to switch between its multiple parts, which include settings, reminders, and health tracking along with unique screens including details verification page, login page and others. Users are guided through the features of the app with ease via clear labelling and intuitive symbols, preventing confusion or annoyance. Savology places a high priority on client feedback and assistance, providing quick customer care channels for answering queries, resolving technical problems, and giving advice on the way to use the app. The app will continue to develop and adapt to the changing demands of its user base because of regular updates and improvements based on input from users.

In addition to that, with the extensive health monitoring features of the app, users can enter and track a variety of health-related information, such as symptoms, diet, physical activity, and medication compliance. Users are empowered to make knowledgeable decisions about their lifestyle and health because of interactive charts and graphs that illustrate patterns over time. This approach helps avert complicated health problems in this population by allowing people to make decisions about their own health (Jiang *et al.*, 2021). Furthermore, users get a personalised dashboard with important health parameters like blood pressure, weight, and activity level when they log in. The indicators that are most important to each individual user can be prioritised on this dashboard, giving them a personalised experience that encourages participation and responsibility. Moreover, it can be seen that the user experience in Savology is made to provide consumers the confidence, ease, and clarity to take charge of their health and well-being.

## 3.4 Data Persistence

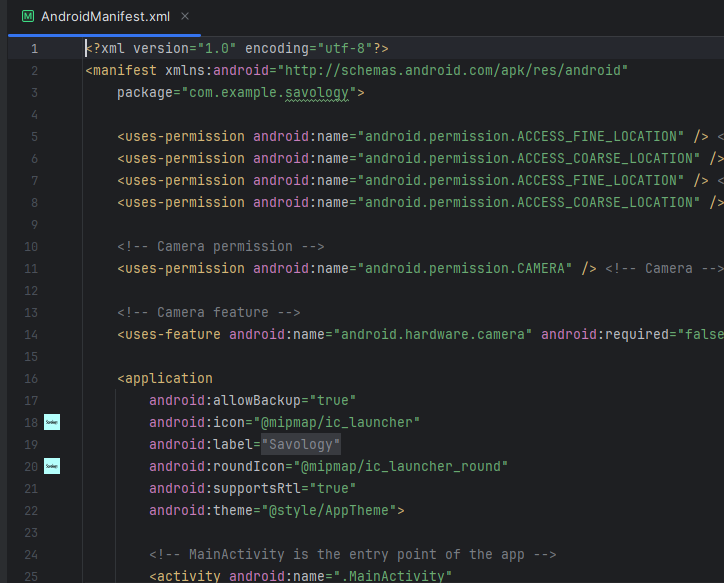
In terms of maintaining appropriate data persistence, the necessary information has been properly stored in a cloud platform integrated in the mobile app development stage. Firebase is an effective platform for storing data and ensuring data integrity appropriately. In this regard, this mobile application has been integrated with the utilisation of the Firebase console, which is also recognised for its backup performance.

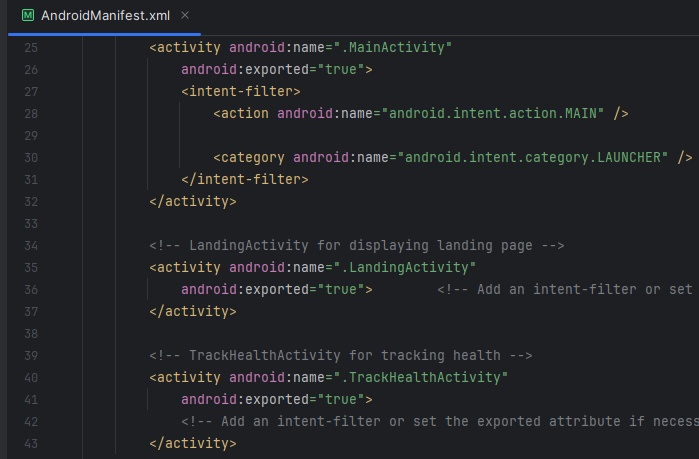
## 3.5 Third Party Library Usage

Third Party Libraries are any third-party software, material, features, functionality, and other components that are utilised in conjunction with the application and are controlled by third parties. These libraries offer pre-built tools, modules, or features that can be added to a software project to increase its capabilities (He *et al.*, 2019). The incorporation of the library into the project environment is indicated by its specification as a component with a key labelled "Savology". Third-party libraries, which offer capabilities and permissions necessary for the Savology application to work, have been used here. The ACCESS\_FINE\_LOCATION and ACCESS\_COARSE\_LOCATION permissions are used by the <uses-permission> tags to request access to the location services of the device. These rights provide the programme access to the exact and approximate position of the device, which is necessary for functions like health tracking and location-based services.

Third-party libraries frequently have community support and copious documentation, making integration and troubleshooting simpler. It has been found from the study of Wang and Guo (2018), developers can take advantage of the knowledge that the designers of the library and other users possess, in order to overcome obstacles and maximise their implementation. In this regard, as stated by Wang and Guo (2018), advanced features and specialised capabilities that might not be possible to build internally are frequently provided by third-party libraries. Hence, without devoting a large amount of funds to creation and research, it is possible to add cutting-edge features to the apps by using these libraries. Overall, it can be inferred that in order to keep the project stable and intact, it is imperative to make sure that these libraries are thoroughly examined for compatibility, security, and licensing issues.

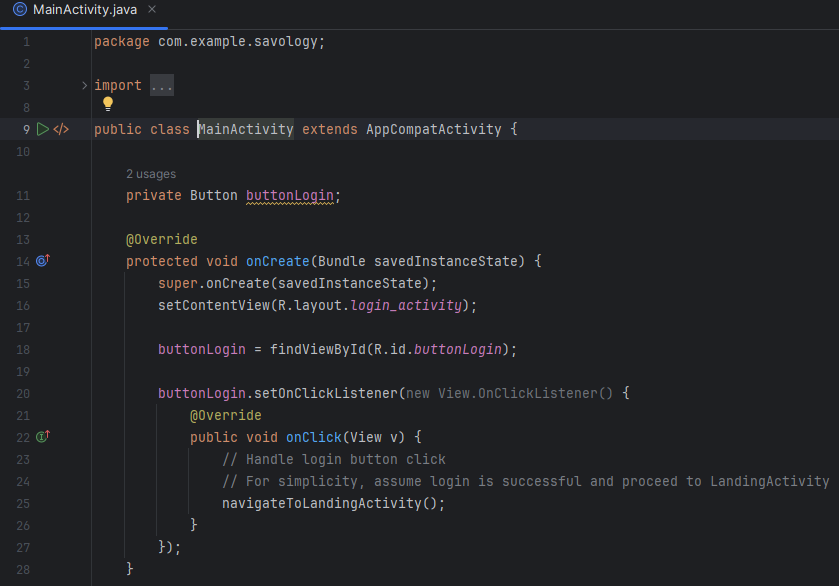
## 3.6 Program Code for Mobile App Development

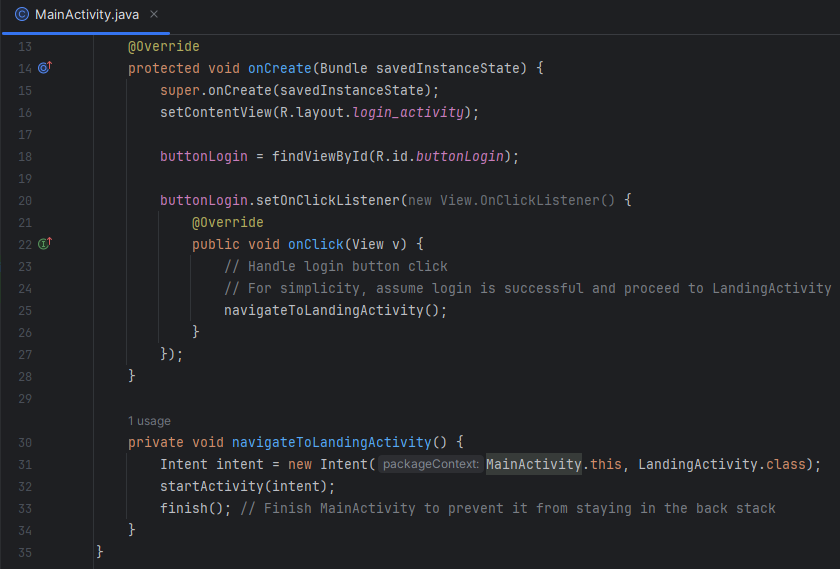


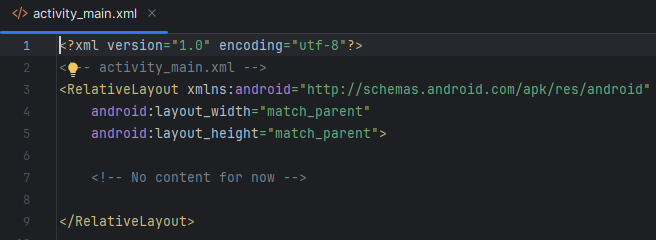


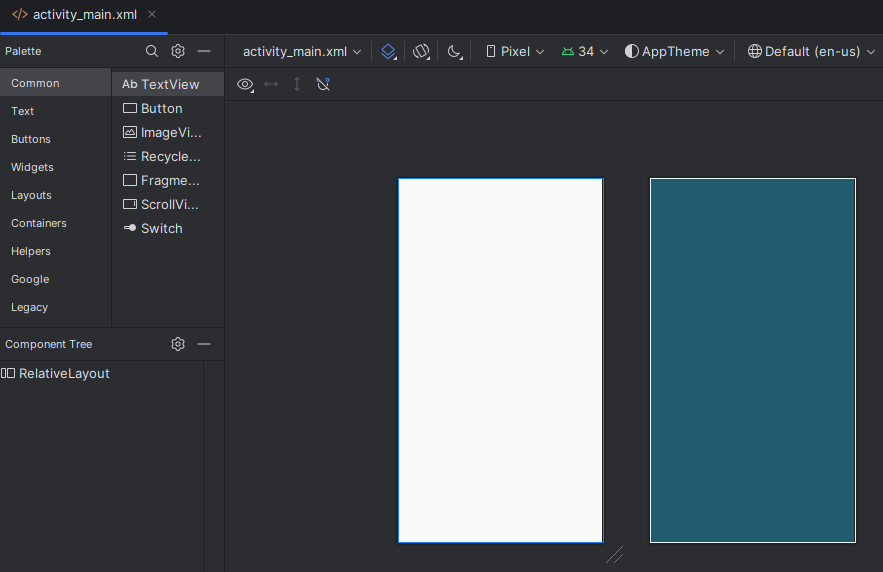
***Figure 5: Android Manifest***

The ACCESS\_FINE\_LOCATION and ACCESS\_COARSE\_LOCATION permissions are used by the app to request access to the location services on the device ***(Refer to figure 5)***. The ability to get the exact and approximate position of the user is made possible by these permissions, and it might be utilised for functions like health tracking and location-based services. In addition, the requirements for the android.hardware.camera feature are specified in the <uses-feature> tag. It suggests that operation of the application depends on the existence of a camera hardware feature. The CAMERA permission is used by the app to request access to the camera of the device. The application can make use of the camera hardware on the device with this permission, which enables features like taking pictures and reading QR codes. Moreover, it can be seen that the core functions of the application, including recording health data and using camera-based features, depend on certain features and permissions.



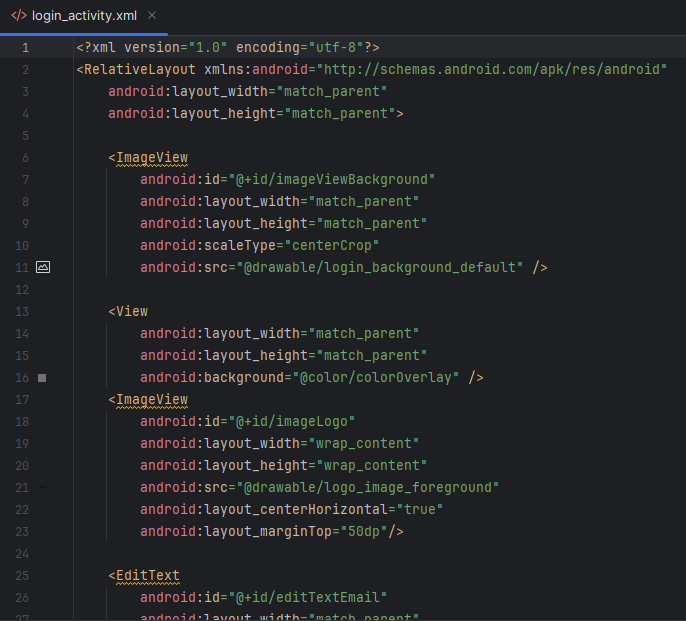


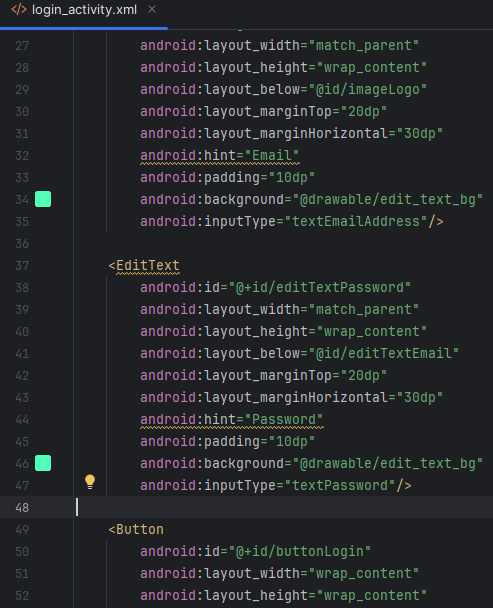


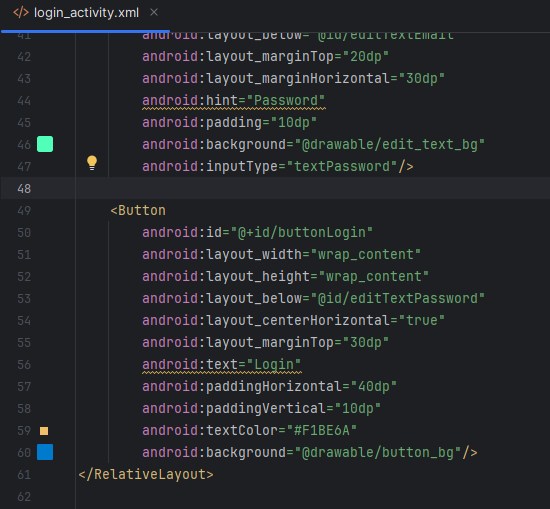


***Figure 6: Main activity***

The MainActivity class is an essential part of the programme as it acts as the entry point of the interface and coordinates user interactions. The setContentView() function is used in the onCreate() method to set the layout of the activity to the login\_activity layout file. Furthermore, the button with the id buttonLogin from the layout file is located in order to initialise the buttonLogin variable. Next, the buttonLogin is configured to have a click listener, which when clicked, causes the onClick() function to be called ***(Refer to figure 6)***. An intent to start the LandingActivity class is created within this method by calling the navigateToLandingActivity() function. The user is then successfully navigated to the landing page of the application by means of startActivity(), which initiates this intent. The MainActivity is finally closed by using the end() function, which stops it from being in the back stack. This illustrated the flow of the Savology application, with the MainActivity managing user interactions like checking in and then navigating to other activities based on actions taken by the user. In addition to that, the root layout container is a RelativeLayout element, which enables child views to be positioned in relation to the parent layout or each other. In order to design the user interface of the main activity of the application, different UI components, including buttons, text fields, or graphics have been added to layout files like activity\_main.xml. It is possible to design the visual aspects that people will interact with on their devices by utilising the layout editor and the many UI elements that are accessible. Moreover, using a RelativeLayout, it establishes the fundamental framework for the layout of the primary activity.

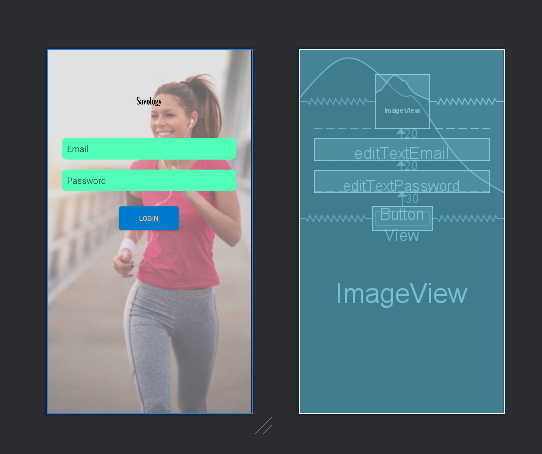






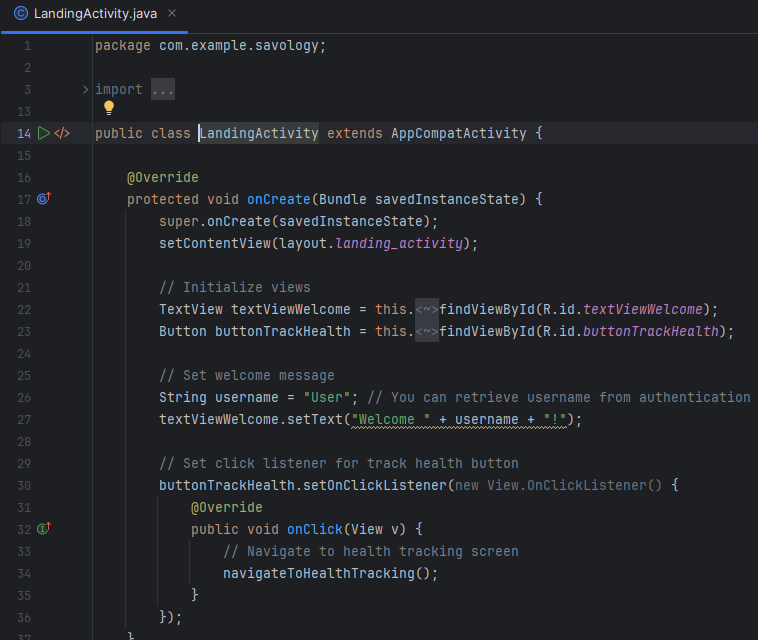
***Figure 7: Programme Code for Layout of Login Page***

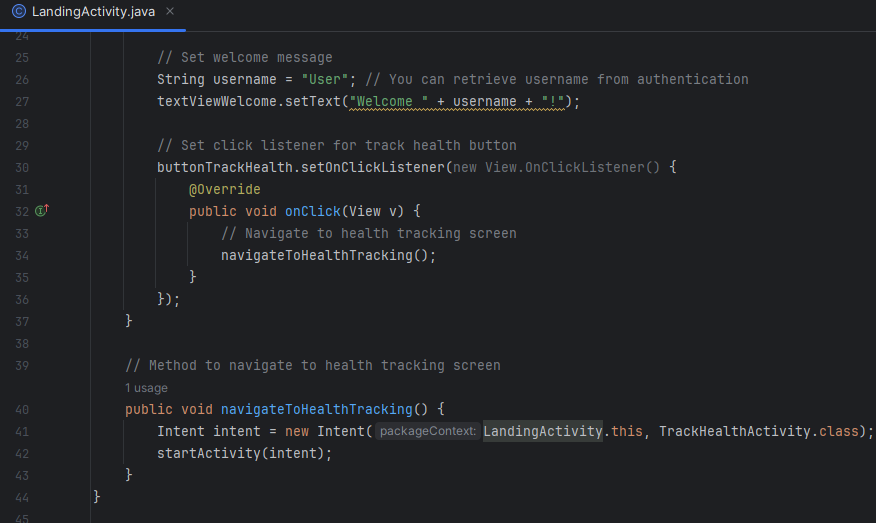
The layout details of the login page of the Savology app have been included in the login\_activity.xml file stating different attributes and visual elements. The programme file for login layout of the Savology app contains the details of imageview, view, edittext, and button in terms of their id, background, textcolor, padding, inputType, and other visual aspects ***(Refer to Figure 7)***. The included details of each component have properly defined the visual aspect of the login page of Savology, which would be beneficial for users to get authenticated logins to the health monitoring app. The <RelativeLayout></RelativeLayout> tags have included the visual details of the login page for users of the Savology app ***(Refer to Figure 7)***.



***Figure 8: Output of Login page of Savology***

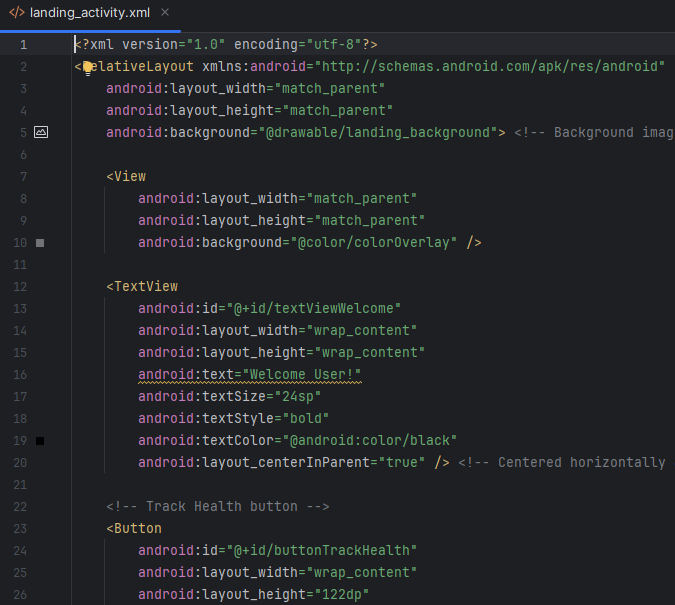
The login page of the Savology app consists of two definite text areas with an image background for including definite tasks such as taking input from users in terms of email and password to redirect to the landing page of the mobile app ***(Refer to Figure 8)***. There is the login button with proper middle orientation for clicks by users with a striking blue colour. The blueprint of the login page of Savology consisting of two text areas for email and password and the logo and button for login operations has been properly depicted using the login\_activity.xml file ***(Refer to Figure 8)***.

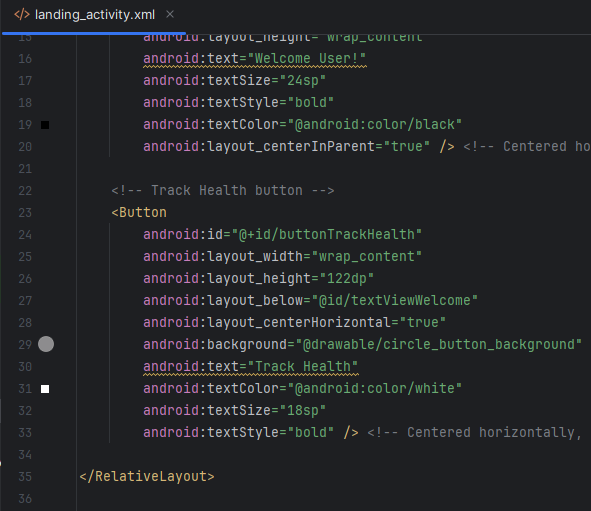




***Figure 9: Programme Code for Operations of Landing Page***

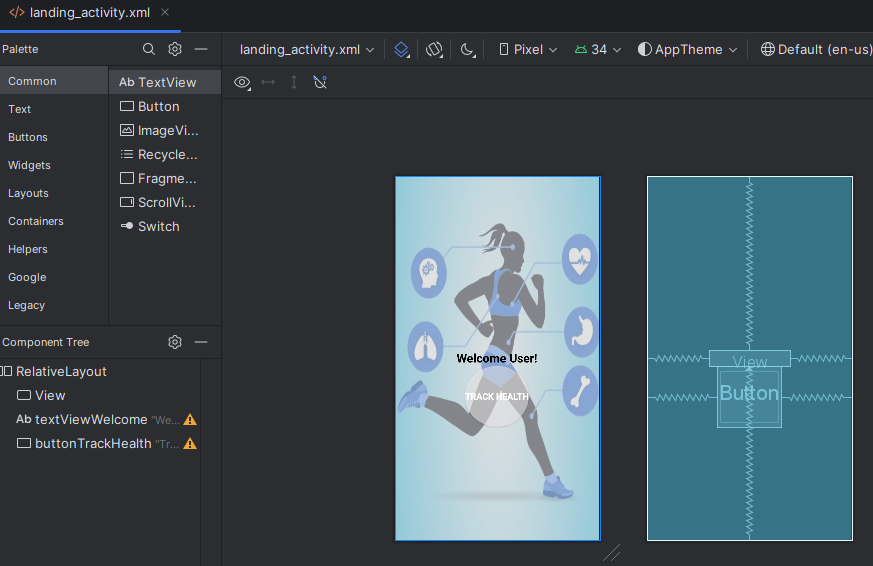
In the landing page of the Savology app, Java programming operations have been applied to include overriding for different functions such as onCreate() and onClick() ***(Refer to Figure 9)***. Additionally, a public void function, navigateToHealthTracking(), has been defined to include a new intent properly.





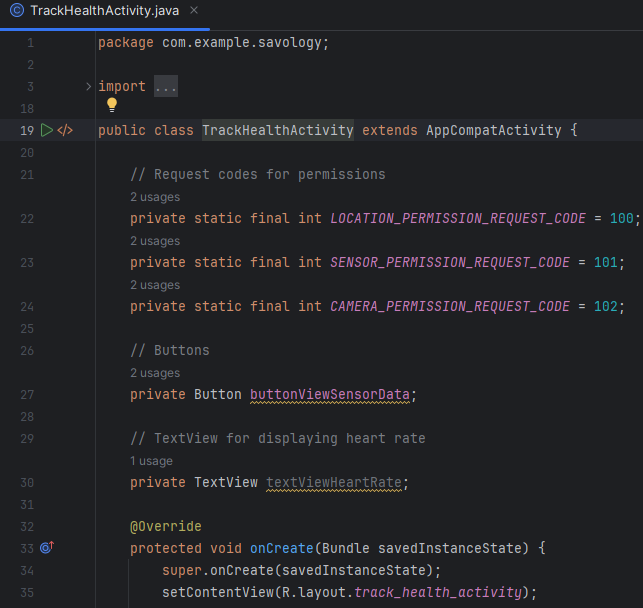
***Figure 10: Programme Code for Layout of Landing Page***

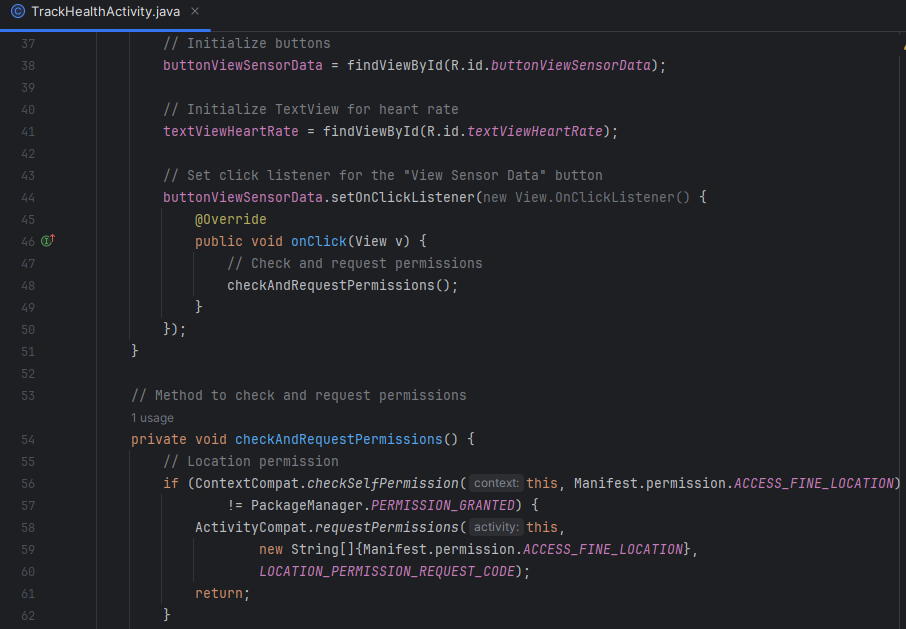
The layout overview of the landing page of the Savology app has been demonstrated in the landing\_activity.xml file with the definition of different attributes and visual elements. The coded file for landing layout of the Savology app comprises the visual orientation of imageview, view, edittext, and button in terms of their size, background, padding, textcolor, inputType, and other visual factors within the <RelativeLayout></RelativeLayout> tags ***(Refer to Figure 10)***.

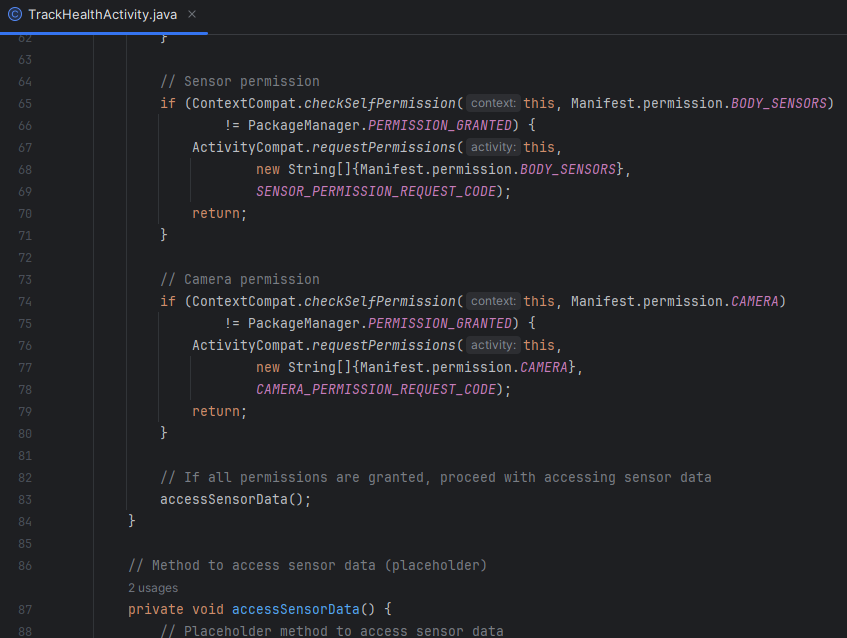


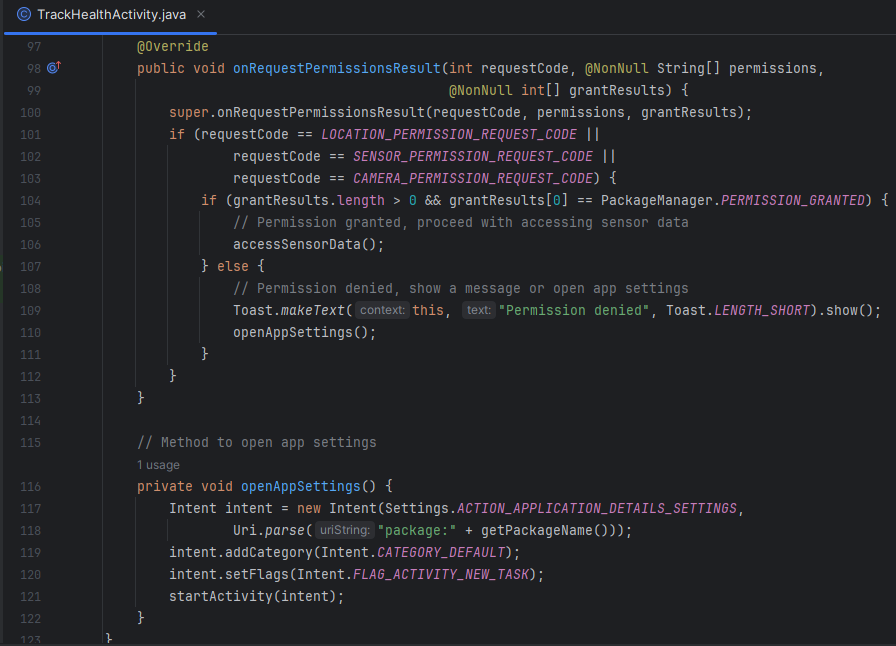
***Figure 11: Output of Layout of Landing Page of Savology***

The landing page of the Savology app comprises an image background with a “Welcome User” message and the "Track Health'' button for taking input from users ***(Refer to Figure 11)***. In addition, the entire layout of the landing page of Savology has been shown as the blueprint of the mobile application containing a button and text area in the middle orientation of the mobile application.



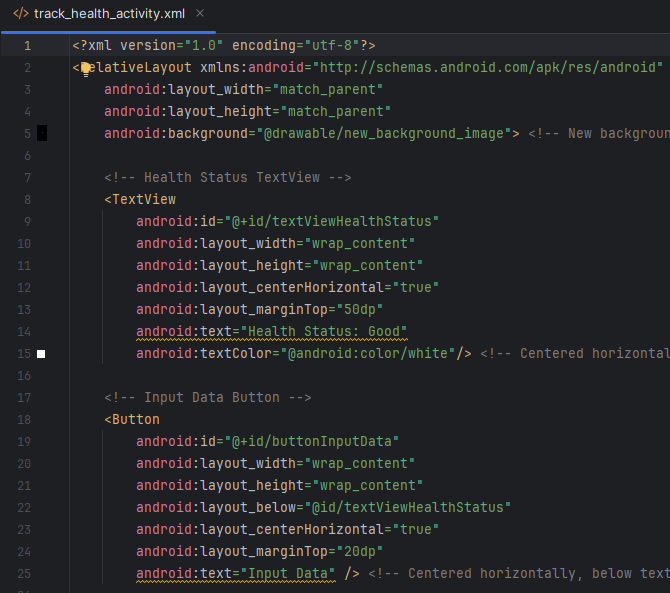


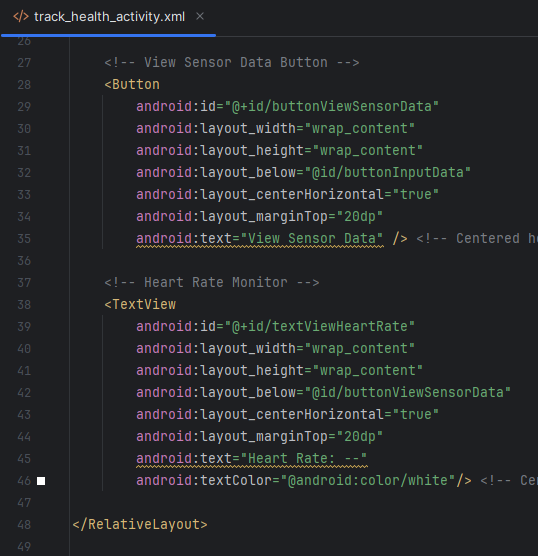




***Figure 12: Programme Code for operations of Health Checking Activity of Savology***

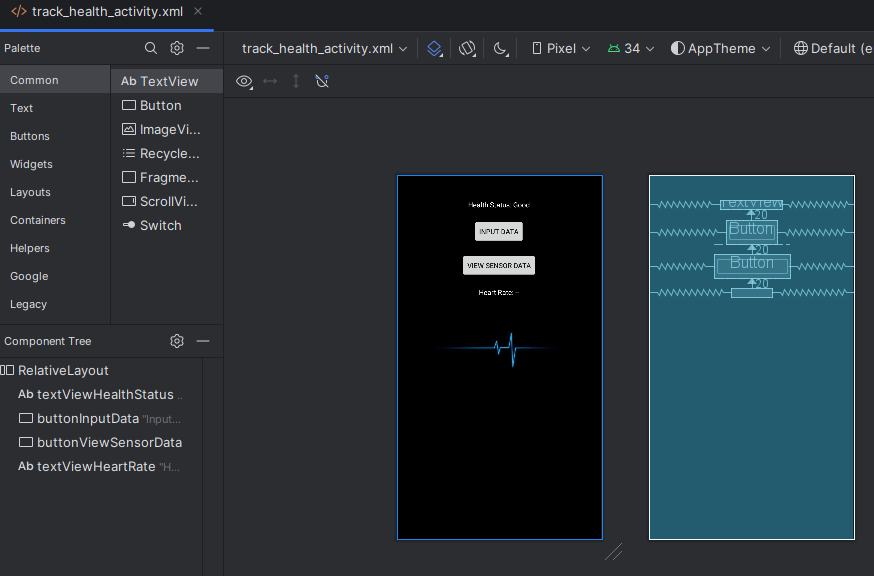
The TrackHealthActivity.java file contains different Java functions to include overriding operations for onCreate(), onClick(), and onRequestPermissionsResult() functions ***(Refer to Figure 12)***. The private void openAppSettings() function is defined in the programme file to create new intent in health checking operations.





***Figure 13: Programme Code for Layout of Health Checking Activity of Savology***

The track\_health\_activity.xml file includes the exact framework for different elements on the occasion of tracking the health condition of individuals in the Savology app. Different attributes such as text, textColor, layout\_below, id, and others of each component have been properly defined with definite values for depicting visual aspects of the health checking page of Savology ***(Refer to Figure 13)***. The entire visual details of the health checking page have been mentioned in the track\_health\_activity.xml file for describing text view, button, and other components for users. The <RelativeLayout></RelativeLayout> tags have properly addressed the visual aspect of the health checking page of Savology.



***Figure 14: Results of Testing of Health Activity***

The health checking page of the Savology app contains a black background with two buttons for definite tasks such as taking input from users and viewing sensor data ***(Refer to Figure 14)***. These two buttons are at the same distance from each other and the other two text areas. However, the heart rate of each individual has been shown below the last text area, which is 20 units apart from its above elements ***(Refer to Figure 14)***. Besides that, the entire layout of the health tracking page of Savology has been shown as the blueprint of the mobile application.

# 

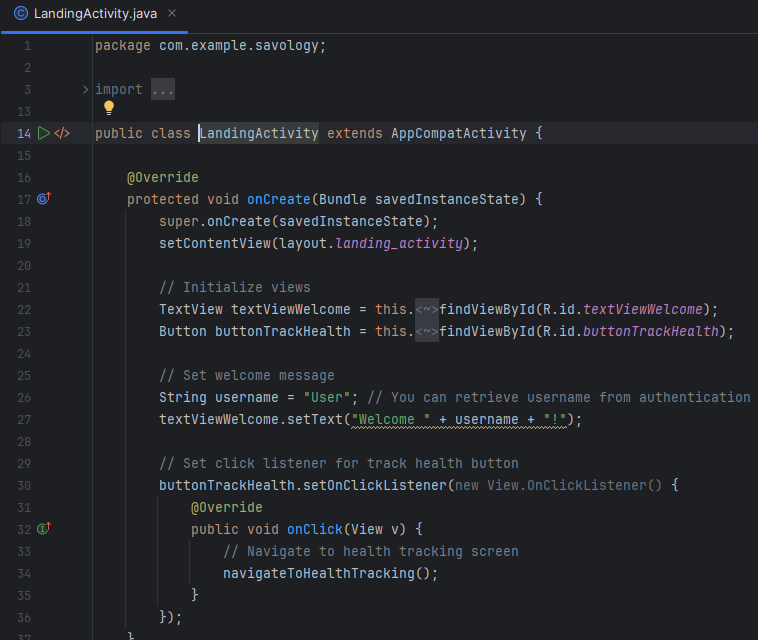
# 4. Social, legal, ethical and security issues

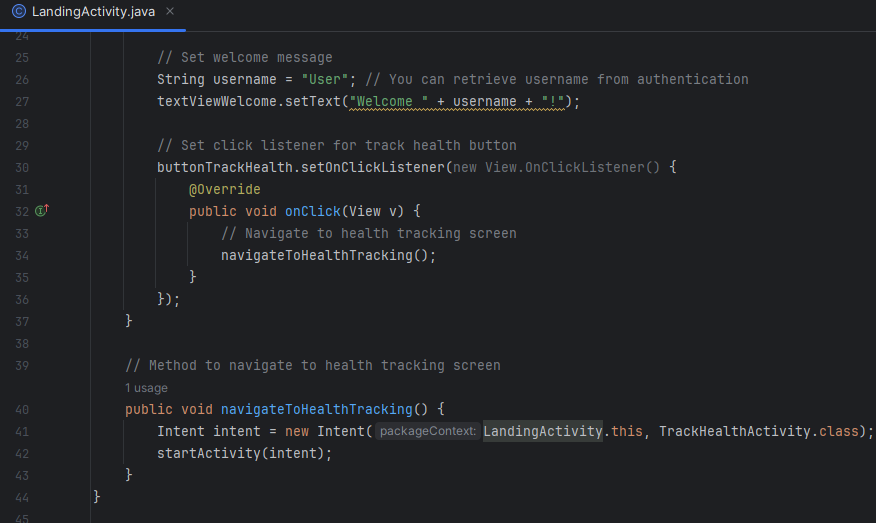
From a legal standpoint, adherence to licensing requirements for both the Android platform and third-party libraries is essential. This entails being aware of the licence agreements governing the usage, distribution, and attribution of the software components that make up the application. Legal consequences, such as accusations of copyright infringement or licence breaches, can result from noncompliance with these regulations (Pallocci *et al.*, 2023). In addition, ethically, data security and user privacy need to be given top priority in the app. It might be possible that the app may gather sensitive personal data because it monitors and evaluates changes in users' bodies. Therefore, it is essential to strictly follow data privacy laws, such as the General Data privacy Regulation (GDPR) (Park *et al.*, 2018). This means getting users' express agreement before collecting any personal data, making sure it is encrypted both during transmission and storage, putting strong access controls in place, and informing users in a clear and understandable way about the way their data will be used.

Socially, for users between the ages of 40 and 60, the app encourages illness prevention and health knowledge***.*** Savology places a high priority on user privacy and data security in order to foster adoption and foster confidence.In addition, getting informed permission, maintaining data confidentiality, and protecting against illegal access are a few ethical factors (Pallocci et al., 2023).Sensitive user data is protected through the use of strong security mechanisms including encryption and secure authentication (Weippl and Schrittwieser, 2023). Apart from that, the application has to take social factors like accessibility and health equity into account. The focus of the app is to encourage healthy living among those between the ages of 40 and 60, however it has been crucial to make sure that everyone can use it, regardless of their financial situation, geography, or degree of digital proficiency. In order to mitigate these issues and ensure equal access to health information and services, it might be helpful to offer resources catered to different cultural backgrounds, create an accessible user experience, and provide multilingual assistance (Sharma *et al.*, 2023). Additionally, from a security perspective, the app implements measures in place to protect against malicious attacks, unauthorised access, and data breaches. This entails putting in place safe authentication procedures, encrypting confidential information, upgrading software components on a regular basis to fix known vulnerabilities, and carrying out exhaustive security audits and assessments (Shukla *et al.*, 2022). Ultimately, it can be said that in order to maintain compliance, safeguard user privacy and data, advance health equity, and reduce cybersecurity threats, it is crucial that social, legal, ethical, and security problems be taken into consideration during the creation and implementation of the Savology app.

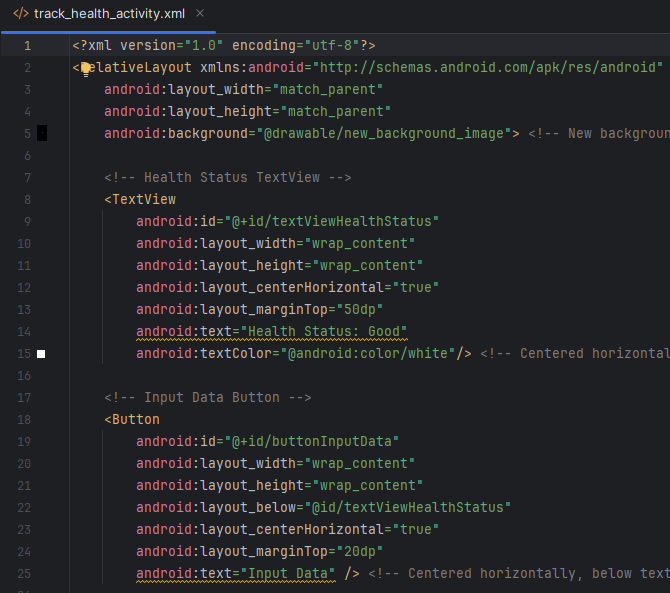
# 5. Selection, use, and evaluation of an agile development method

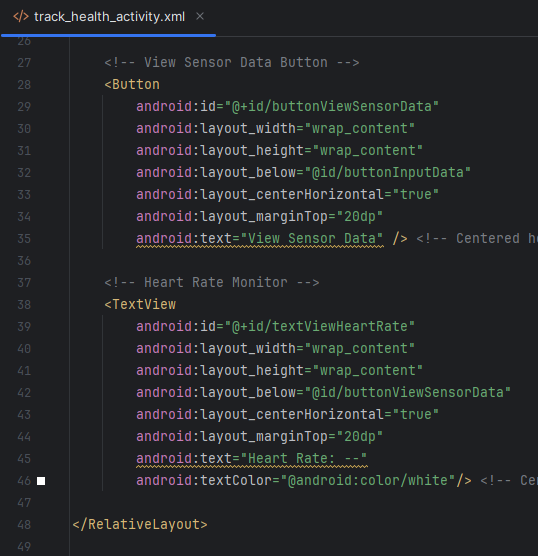
Extreme Programming (XP) is an agile development strategy that has been chosen, used, and assessed for the Savology app in order to ensure successful and effective software delivery and project management. As per the view of Shrivastava et al. (2021), extreme programming is a popular choice for dynamic and changing project needs because of its emphasis on adaptation, flexibility, and quick feedback cycles. The iterative methodology of XP enables continual modifications and upgrades based on user feedback in Savology, where user demands and health patterns may alter over time. Savology integrates XP ideas into every stage of the development process to increase output and quality. This includes methods like Test-Driven Development (TDD), which writes tests before implementing code to ensure robustness and dependability (Ivo et al., 2018). Besides that, regular releases and ongoing integration provide the gradual delivery of features, giving stakeholders early access to project status and enabling prompt modifications.





***Figure 15: Programme Code for Operations of Landing Page***





***Figure 16: Programme Code for Layout of Health Checking Activity of Savology***

Furthermore, throughout the development life cycle of Savology, the effectiveness of extreme programming is continually reviewed and analysed to find areas for improvement and make sure the project objectives are met. In order to measure productivity and quality, metrics like defect rate, cycle time, and velocity are monitored. This enables data-driven decision-making and process optimisation. It ensures that the app fulfils end users' and stakeholders' changing needs and expectations by requesting and incorporating their feedback into subsequent revisions (Butt, Naaranoja and Savolainen, 2018).

Moreover, it can be determined that selecting, implementing, and assessing Extreme Programming during the Savology development process has been crucial in fostering teamwork, flexibility, and excellence all the way through the lifecycle of the project.

# 

# 6. Use of digital environments and tools

Digital environments and tools are essential for promoting collaboration, optimising development processes, and guaranteeing the delivery of a high-calibre result in the Savology app development process. The open-source UI software development kit ***Kotlin DSL***, created by Google, is one of the primary digital environments selected for this project. Kotlin DSL is the perfect option for Savology since it provides multiple kinds of benefits for developing cross-platform mobile applications. Kotlin DSL reduces development time and effort while guaranteeing consistency and uniformity across many devices by allowing developers to create code once and publish it across several platforms, including iOS and Android (Kotlin DSL, 2024b). The Savology app can maximise its impact and reach a wider audience. In this regard, null safety and extension functions, two contemporary language features of Kotlin, improve code quality and lower runtime errors to guarantee a stable and dependable program (Wood *et al.*, 2020). The development process is streamlined by its robust interaction with Android Studio and smooth compatibility with Java. Additionally, the expressiveness and simple syntax of Kotlin DSL increase developer productivity and code maintainability (Wood *et al.*, 2020). All of these characteristics combine to make Kotlin DSL the perfect framework for creating an excellent, safe, and intuitive application, similar to Savology. Hence, this maximises code reuse while cutting down on development time and effort by doing away with the requirement for distinct codebases for each platform. Aside from that, with the abundance of pre-built widgets and components offered by Kotlin DSL, it can be possible to easily design stunning and highly engaging user interfaces. This improves user happiness and retention by enabling the Savology app to offer a smooth and captivating user experience (Kotlin DSL, 2024a).

In addition to that, the remarkable speed and performance of Kotlin DSL allow the Savology app to provide seamless and responsive user experiences. As per the study conducted by Sattar *et al.* (2023), fast app launch speeds and fluid animations are made possible by the highly optimised rendering engine of Kotlin DSL, which also helps to increase user engagement. Thus, it is possible to create a stable, feature-rich, and intuitive mobile application for Savology because of the smart use of digital environments and technologies, especially Kotlin DSL. Moreover, the app can successfully accomplish its goals while upholding high standards of effectiveness and quality by using these technologies.

# 

# 7. Conclusion

As per the above discussion, it can be summarised that the mobile application has been developed with a simpler design and accessibility operation to engage its targeted users successfully. As the mobile application targets aged individuals, the simpler design has promoted the success to enhance user experience in health monitoring operations. In addition, the mobile application has integrated different hardware components such as camera, sensors, and others to check health conditions of the users. Overall, the demonstration illustrates the way users can obtain precise information on their health and fitness by navigating between different areas of the app.

# 

# References

Deng, Z. *et al.* (2023) ‘Smart Wearable Systems for Health Monitoring’, *Sensors*, 23(5), p. 2479. doi: https://doi.org/10.3390/s23052479.

Han, E.-G., Kang, T.-K. and Lim, M.-T. (2023) ‘Physiological Signal-Based Real-Time Emotion Recognition Based on Exploiting Mutual Information with Physiologically Common Features’, *Electronics*, 12(13), p. 2933. doi: https://doi.org/10.3390/electronics12132933.

He, Y. *et al.* (2019) ‘Dynamic privacy leakage analysis of Android third-party libraries’, *Journal of Information Security and Applications*, 46, pp. 259–270. doi: https://doi.org/10.1016/j.jisa.2019.03.014.

Jiang, F. *et al.* (2021) ‘Understanding health empowerment in the context of online information seeking: the perspective of information processing (Preprint)’, *Journal of Medical Internet Research*, 24(1). doi: https://doi.org/10.2196/27178.

Kotlin DSL (2024a) *Material Components widgets*, *docs.Kotlin DSL.dev*. Available at: https://docs.Kotlin DSL.dev/ui/widgets/material (Accessed: 26 April 2024).

Kotlin DSL (2024b) *Multi-Platform*, *Kotlin DSL.dev*. Available at: https://Kotlin DSL.dev/multi-platform (Accessed: 26 April 2024).

Majumder, S. and Deen, M. J. (2019) ‘Smartphone Sensors for Health Monitoring and Diagnosis’, *Sensors*, 19(9), p. 2164. doi: https://doi.org/10.3390/s19092164.

Pallocci, M. *et al.* (2023) ‘Informed Consent: Legal Obligation or Cornerstone of the Care Relationship?’, *International Journal of Environmental Research and Public Health*, 20(3), p. 2118. doi: https://doi.org/10.3390/ijerph20032118.

Park, S. *et al.* (2018) ‘A comparative study on data protection legislations and government standards to implement Digital Forensic Readiness as mandatory requirement’, *Digital Investigation*, 24, pp. S93–S100. doi: https://doi.org/10.1016/j.diin.2018.01.012.

Sattar, A. Md. *et al.* (2023) ‘Accelerating Cross-platform Development with Kotlin DSL Framework’, *Journal of open source developments*. doi: https://doi.org/10.37591/joosd.v10i2.580.

Sharma, A. *et al.* (2023) ‘Designing Inclusive Learning Environments’, *Advances in educational technologies and instructional design book series*, pp. 24–61. doi: https://doi.org/10.4018/978-1-6684-8208-7.ch002.

Shukla, A. *et al.* (2022) ‘System security assurance: A systematic literature review’, *Computer Science Review*, 45, p. 100496. doi: https://doi.org/10.1016/j.cosrev.2022.100496.

Wang, H. and Guo, Y. (2018) ‘Understanding Third-Party Libraries in Mobile App Analysis’, *IEEE/ACM 39th International Conference on Software Engineering Companion*. doi: https://doi.org/10.1109/icse-c.2017.161.

Weippl, E. and Schrittwieser, S. (2023) ‘Introduction to Security and Privacy’, *springer*, pp. 397–414. doi: https://doi.org/10.1007/978-3-031-45304-5\_26.

Wood, B. *et al.* (2020) ‘Remote Method Delegation: a Platform for Grid Computing’, *Journal of Grid Computing*, 18(4), pp. 711–725. doi: https://doi.org/10.1007/s10723-020-09525-z.

Zeng, X. *et al.* (2020) ‘Investigation of an Ultra Wideband Noise Sensor for Health Monitoring’, *Sensors*, 20(4), p. 1034. doi: <https://doi.org/10.3390/s20041034>.