

**A. Introduction**

**Introduction:**

The aim of this proposal is to present a solution to the problem of DUT clinics that still uses a manual paper-based system to record and keep track of students and staff who need medical attention. We will develop an app that will assist students and staff to easily access clinic services and book appointments. The app will also have the feature to remind students and staff about their upcoming appointments, reducing the likelihood of missed appointments. The app will be designed with a user-friendly interface, allowing students and staff to book appointments and access clinic services quickly and easily.

**Problem:**

We identified that DUT has no electronic filing systems at their clinics, everything is done manually on piece of paper which sometimes ends up in them losing student files and information of previous visits. It becomes a drag to both student and clinic in going about the process of creating a new file all over again. Bookings are done same day at the clinic which makes it difficult for those who live far to make it early in line for appointment booking, some of these students end up not getting slots for appointments on that day. It would be also easier to identify how severe a student's condition is during booking so that the clinic would know whether to book them to see a doctor or just a nurse.

**Solution:**

We propose the development of a mobile app for a clinic booking system for students in campus. The app will be designed to make it easier for students to access healthcare services on campus by allowing them to schedule appointments, view their health records, and receive notifications about upcoming appointments. the app will help to increase awareness of healthcare services available on campus and improve communication between students and healthcare providers. The app will be available for both iOS and Android platforms. With the proposed solution, we believe that we can provide a convenient way for students to access healthcare services and manage their healthcare needs, while also increasing efficiency and productivity for clinic staff.

**Target Platforms**: The application will be available on both iOS and Android platforms.

**Supported Devices**: smartphones and tablets.

**Features and Objectives:**

**Login** – to allow students who have already visited the clinic to login into their profile and

book further appointments.

• **Register** – to allow students to create a profile or “file” on the clinic’s database so that they

can book appointments at their convenience.

**• Booking appointments/Scheduling** – to allow students to book appointments at ease

without going to the clinic. This avoids long queues and waiting periods.

• **Appointment reminder** – this allows for students to get reminders 1 or 2 days before the

actual appointment date.

• **Availability checking** – this will allow for students to check what dates and times are

available to students as specified by the clinic.

**B. Design**

**User Interface (UI)**

Our mobile application is a clinic booking system which will allow students to schedule appointments with healthcare professionals on DUT Campuses for all their healthcare needs. Our new innovative mobile application will substitute the old traditional method of using a manual paper-based system. The mobile application allows students to book appointments online thus no need for in-person visits and reducing the need for phone calls. This saves time and effort for students and clinic staff and will lead to an overall better healthcare service for students. Our clinic booking system is user friendly, it allows patients to navigate the app and complete the booking process without obstacles. The UI is simple and clean we ensured to not use too many elements, colors and avoided too much information that will overwhelm users. The typography is legible and easy to read, the font size and type are appropriate for each part of our app. For visual appeal we used high-quality images. Most importantly all patient’s data is secured since we created secure login procedures, so sensitive information will be managed using the app.

**Wireframe structure on our booking system:**

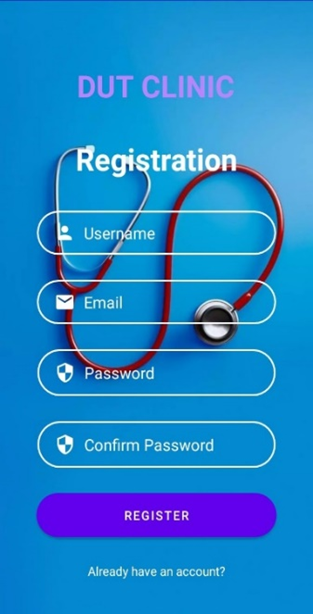
**A screenshot of a login form

Description automatically generated with medium confidence**

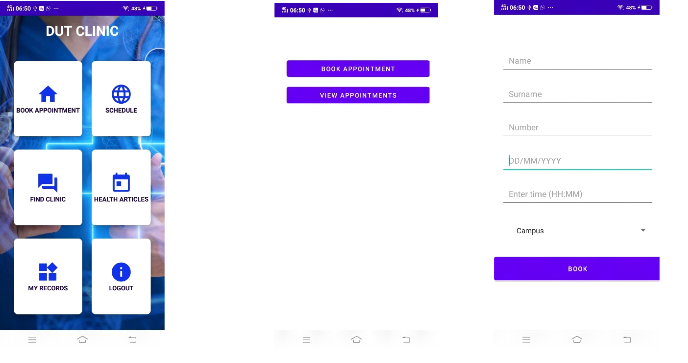
Graphical user interface, diagram

Description automatically generated

**Login** –A login page will allow for DUT students who have already visited the clinic to login into their profile. They will be able to book appointments as well as view their future bookings.



**Register** – A registration page will for allow DUT students to create a profile or “file” on the clinic’s database so that they are able book appointments at their convenience.



**Booking appointments/Scheduling** –A booking/scheduling page will allow students to book appointments at ease without visiting the clinic. This avoids long queues and waiting periods. There are various clinics pertaining to different healthcare professionals, students will be able to access all clinics via the app. By doing so, the problem of locating each clinic will be eliminated.

**Appointment reminder** – This page will allow for DUT students to receive reminders 1 or 2 days before the actual appointment date. These reminders will help students by alerting them of their appointment soon. Previously students would not have been reminded about their allocated appointment, now the likelihood of a missed appointment is reduced.

**Find Clinic** - This is where all DUT Clinics are found, by clicking you will be able to see Ritson Medical Clinic, Steve Medical Clinic, Dental Clinic, Somatology Clinic and Outsource Clinics along with their experiences and contact details.

**Health Articles**- This page allows for students to read up on heath articles and further information relating to their problem.

**Appointment reminder** – This page will allow for DUT students to receive reminders 1 or 2 days before the actual appointment date. These reminders will help students by alerting them of their appointment soon. Previously students would not have been reminded about their allocated appointment, now the likelihood of a missed appointment is reduced.

**C. Technical Architecture**

1. **Front-end:**

* Responsible for rendering the user interface and handling user interactions.
* Developed using Java/Kotlin for Android.
* The front-end communicate with the back-end components to fetch data and update the user interface accordingly.
* Front end function:

1. **Back-end :**

The back-end of the application would handle the logic and data processing.

* It would be responsible for managing user accounts, handling authentication and interacting with the database.
* All the server functions are written in Java.

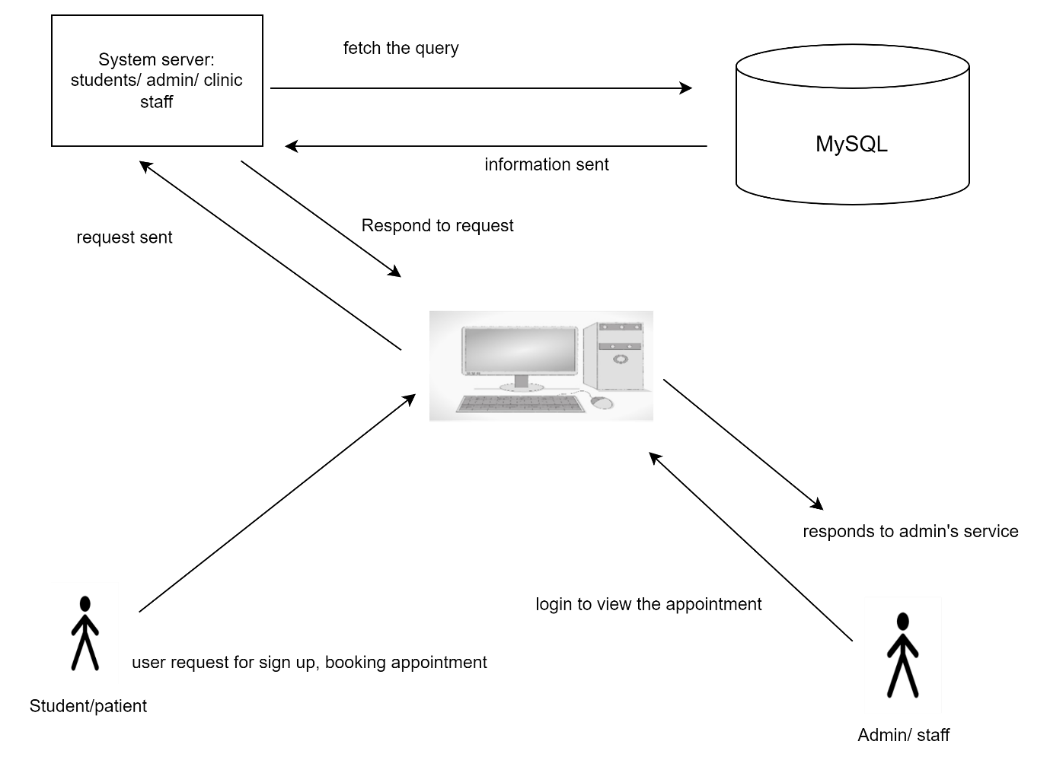
**3.Database:**

A database is used to store clinic related data such as students, appointments, and other relevant information.

The database is used to:

1. Store information about clinics, such as their location, operating hours, contact information, and availability.
2. Store information about students, such as their name, contact information, medical history, and appointment history.
3. Manage appointment scheduling, including tracking available appointment slots,
4. scheduling appointments, and managing cancellations or rescheduling.

**The following diagram represent the system architecture:**



Activity Diagram   
use case:

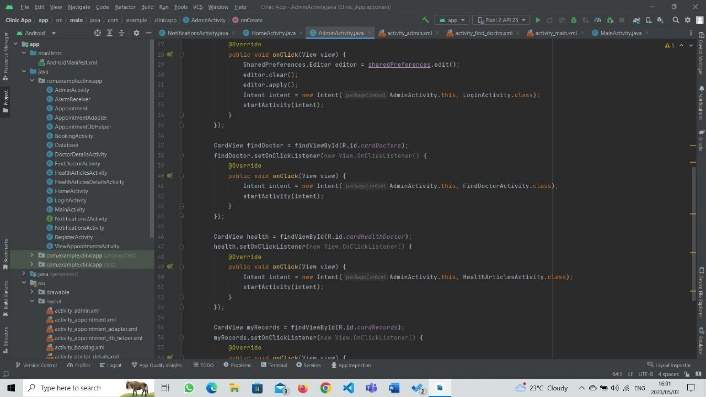
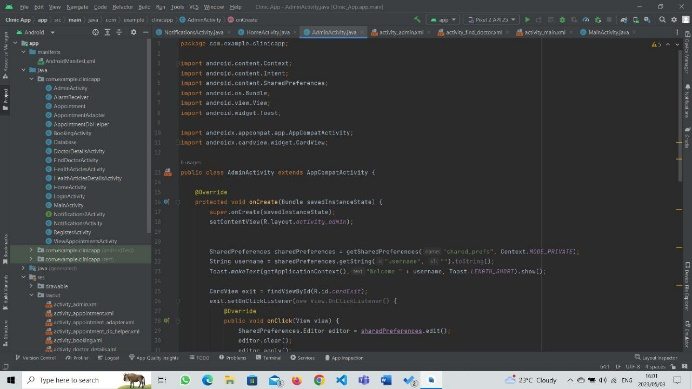


Database Schema

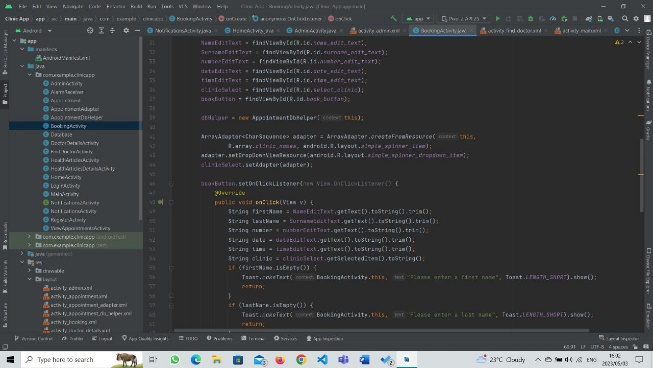


**D. Code**

The main code highlighted is the booking appointment. The following contains the highlighted code that will help another developer understand how the booking process works.

Admin Activity Code

A screenshot of a computer

Description automatically generated  Booking code

A screenshot of a computer

Description automatically generated Admin Dashboard xml Code

A screenshot of a computer

Description automatically generated Booking xml Code

A screenshot of a computer program

Description automatically generated with medium confidence A screenshot of a computer program

Description automatically generated with medium confidence Database

**E. Testing**

**Test plan**

Functional testing:

Verify that the app allows users to log in or sign up and create a profile.

Verify that the app displays a list of available healthcare professionals, their specialties, and the times they are available.

Verify that the app allows users to select a healthcare professional and book an appointment at a convenient time.

Verify that the app sends a confirmation email or notification to the user and the healthcare professional after an appointment is booked.

Verify that the app allows users to view and modify their upcoming appointments and cancel appointments if necessary.

Verify that the app sends reminders to users about upcoming appointments, based on their preferences.

Usability testing:

Conduct user testing to ensure that the app is easy to navigate and understand.

Collect feedback from users on the design, layout, and overall usability of the app.

Analyse the feedback and make changes to improve the user experience.

Performance testing:

Test the app's performance under various conditions, such as low internet connectivity, multiple users accessing the app simultaneously, and heavy usage.

Measure the app's response time and loading speed.

Ensure that the app does not crash or freeze during regular use.

Security testing:

Verify that the app securely stores user data and sensitive information, such as login credentials and medical history.

Test the app's authentication and authorisation mechanisms to ensure that only authorised users can access the app's features and data.

Perform penetration testing to identify potential security vulnerabilities and mitigate them.

Compatibility testing:

Test the app on different Android devices with various screen sizes, resolutions, and operating system versions.

Ensure that the app's layout and functionality are consistent across different devices.

Test the app's compatibility with popular web browsers and operating systems.

Accessibility testing:

Verify that the app complies with accessibility guidelines.

Test the app's accessibility features, such as text-to-speech, color contrast, and font size.

Regression testing:

Conduct regression testing to ensure that new updates or changes to the app do not introduce new bugs or affect the app's existing functionality.

Test the app's compatibility with new devices, operating system versions, and web browsers.

Test the app's security and performance after updates or changes are made.

**Test Cases**

| Test Case# | Description | Test Steps | Conditions | Test Data | Expected Result | Actual Result | Pass/Fail |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | Check User Login with valid data | 1. Enter username  2. Enter password  3. Click Login | 1. Username must be 8 characters or longer.  2. Password must consist of 8 characters, numbers, special characters (@, #, \*), uppercase and lowercase letters. | Username = sfolly99  Password = Sfolly123@ | User logged in to app | As expected, | Pass |
| 2 | Check User Login with invalid data | 1. Enter username  2. Enter password  3. Click Login | 1. Username must be 8 characters or longer.  2. Password must consist of 8 characters, numbers, special characters (@, #, \*), uppercase and lowercase letters. | Username = sfolly  Password = Sfolly123 | User log in unsuccessful. Error message displayed. Re-enter password. | As expected, | Fail |
| 3 | Register new user with valid data | 1. Enter username  2. Enter email  3. Enter password  4. Confirm password | 1. Email must contain @ symbol.  2. Password must consist of 8 characters, numbers, special characters (@, #, \*), uppercase and lowercase letters.  3. Username must be 8 characters or longer. | Username = sfolly99  Email = [sfolly@gmail.com](mailto:sfolly@gmail.com)  Password = Sfolly123@ | Registration successful. Redirected to login. | As expected, | pass |
| 4 | Register new user with invalid data | 1. Enter username  2. Enter email  3. Enter password  4. Confirm password | 1. Email must contain @ symbol.  2. Password must consist of 8 characters, numbers, special characters (@, #, \*), uppercase and lowercase letters.  3. Username must be 8 characters or longer. | Username = sfolly  Email = [sfollygmail.com](mailto:sfolly@gmail.com)  Password = 123qwerty | Registration unsuccessful. Error message displayed to enter valid username, email and password. | As expected, | fail |
| 5 | Entering valid mobile number | 1. Go to booking  2. Enter mobile number | 1. Must be 10 digits  2. Must begin with 0 | 098 1234 567 | Mobile number saved and loaded to database | As expected, | Pass |
| 6 | Entering invalid Number | 1. Go to booking  2. Enter mobile number | 1. Must be 10 digits  2. Must begin with 0 | 756382 | Mobile number unaccepted. Enter valid mobile number. | As expected, | Fail |
| 7 | Book appointment with valid data | 1. Enter Name  2. Enter Surname  3. Enter Mobile  4. Number  5. Enter Date  6. Enter Time  7. Choose Clinic  8. Book appointment | 1. All fields need to be filled in  2. Mobile number must be 10 digits and must begin with 0  3. Date must be the current date and in the format DD/MM/YYYY  4. Time must be according to digital  5. Must choose a clinic from the dropdown list | Name = Shristi  Surname = Folly  Number = 087 234 5678 Date = 03/05/2023  Time = 14:00  Clinic = Ritson Clinic | Data Loaded to Database.  Appointment Booked Successful. | As expected, | Pass |
| 8 | Book appointment with invalid data | 1. Enter Name  2. Enter Surname  3. Enter Mobile  4. Number  5. Enter Date  6. Enter Time  7. Choose Clinic  8. Book appointment | 1. All fields need to be filled in  2. Mobile number must be 10 digits and must begin with 0  3. Date must be the current date and in the format DD/MM/YYYY  4. Time must be according to digital  5. Must choose a clinic from the dropdown list | Name = Shristi  Surname = Folly  Number = 5463829  Date = 03/2023  Time = 2pm  Clinic = Ritson Clinic | Data not Loaded to Database.  Booking is Unsuccessful.  Re-enter valid data | Data Loaded to database.  Booking successful | Fail |

**Test Results**

Functional Testing: Users can schedule appointments, view their appointments, and receive appointment reminders. It also identifies any defects, such as incorrect data display or unresponsive user interface.

Usability Testing: Usability testing could show that the app is easy to use and navigate. Users can quickly book appointments, access medical information, and receive reminders.

Performance testing the app performs well under different conditions, such as high and low network connectivity. The app is responsive, fast, and reliable.

Security Testing: the app is secure and meets the necessary authentication and authorisation protocols. Users' data is encrypted and protected from unauthorised access. However, security testing may also reveal vulnerabilities, such as unprotected data transmission or weak passwords.

Compatibility Testing: the app is compatible with various devices, operating systems, and screen sizes. The app works seamlessly across different platforms, ensuring that all users can access and use it.

**Conclusion**

A clinic booking app can be useful tool for patients to book appointments with healthcare providers and. It can help streamline the scheduling process and reduce wait times for students (patients), in addition it can give patients more control over their health care experience and allow them to easily view and manage their appointments. For health care providers, a clinic booking app can help improve efficiency of their practice by reducing time spent on administrative tasks such as scheduling appointments. It can also help reduce no-shows and cancelations by providing patients with automated reminders and the ability to easily reschedule appointments.

**References**

* 1. Nilesh Technology. 2022. Android Healthcare Project. Available: <https://youtu.be/9CkpMm-n5iA> (Accessed 19 April 2023)
  2. Barasa, M. (2021, February 7). How To Consume Data from an API in Android. Available: <https://www.section.io/engineering-education/how-to-consume-data-from-an-api-in-android/> (Accessed 29 April 2023)
  3. Simplilearn. 2021. Java API Tutorial for beginners. Available: <https://www.youtube.com/watch?v=BzJpjejk9u8>) (Accessed 29 April 2023).
  4. Hamilton. 2023. How to write test cases in software testing with examples. Available: <https://www.guru99.com/test-case.html> (Accessed 3 March 2023).
  5. Unadkat. 2023. How to learn Software application testing. Available: <https://www.browserstack.com/guide/learn-software-application-testin> (Accessed 3 March 2023).
  6. Vertabelo. 2016. A database model to manage. Available: <https://vertabelo.com/blog/a-database-model-to-manage-appointments-and-organize-schedules/> (Accessed 12 April 2023)
  7. Drkusic, E. 2016. A Database model to manage appointment and organize schedules. Available: <https://vertabelo.com/blog/a-database-model-to-manage-appointments-and-organize-schedules/> (Accessed 28 April 2023).
  8. freeCodeCamp (2022). Android App Development Tutorial for Beginners - Your First App. Available: <https://www.youtube.com/watch?v=FjrKMcnKahY> (Accessed 17 April 2023)
  9. Davies, A. (n.d.). How To Build A Doctor Appointment App. Available: <https://www.devteam.space/blog/how-to-build-a-medical-appointment-app/> (Accessed 19 April 2023).
  10. SkillsBuild Training. (2022). How to Create Your First Android Application with Android Studio | Tutorial for Beginners Available: https://www.youtube.com/watch?v=uPkjgVv5Ioc (Accessed 21 April 2023).