**Smart Healthcare System  
 for Senior Citizen**

**A PROJECT REPORT**

*Submitted by*

Upanshu Bhardwaj

RA2011028010083

Priyam Chakraborty

RA2011003011398

*Under the Guidance of*

(Assistant Professor, Module of)

*In partial fulfillment of the Requirements for the Degree*

*of*

**BACHELOR OF TECHNOLOGY**

## COMPUTER SCIENCE ENGINEERING

## 



**MODULE OF**

**FACULTY OF ENGINEERING AND TECHNOLOGY**

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**KATTANKULATHUR – 603203**

**MAY 2023**

**BONAFIDE CERTIFICATE**

Certified that this project report titled “**Smart Healthcare System for Senior Citizen”** is the bonafide work ofwho carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion for this or any other candidate.

Assistant Professor Head of the Module

Dept. of Dept. of

Signature of Internal Examiner Signature of External Examiner

Module of 

##### SRM Institute of Science & Technology

##### Own Work Declaration Form

##### **Degree/ Course : Bachelor of Technology, Computer Science Engineering**

**Student Name :**

##### **Registration Number : RA20**

**Title of Work : Tuberculosis Detection with Remedials and Air quality analysis**

We hereby certify that this assessment compiles with the University’s Rules and Regulations relating to Academic misconduct and plagiarism, as listed in the University Website, Regulations, and the Education Committee guidelines.

We confirm that all the work contained in this assessment is my / our own except where indicated, and that we have met the following conditions:

* Clearly references / listed all sources as appropriate
* Referenced and put in inverted commas all quoted text (from books, web, etc)
* Given the sources of all pictures, data etc. that are not my own
* Not made any use of the report(s) or essay(s) of any other student(s) either past or present
* Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
* Compiled with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

|  |
| --- |
| **DECLARATION:** |
| I am aware of and understand the University’s policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above. |
| YOUR NAME [REG NO.] \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

**ACKNOWLEDGEMENT**

We express our humble gratitude to **Dr C. Muthamizhchelvan**, Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We extend our sincere thanks to Dean-CET, SRM Institute of Science and Technology, **Dr T.V.Gopal**, for his invaluable support.

We wish to thank **Dr Revathi Venkataraman**, Professor & Chairperson, School of Computing, SRM Institute of Science and Technology, for her support throughout the project work.

We are incredibly grateful to our Head of the Module**,**Professor, Module of Data Science and Business Systems, SRM Institute of Science and Technology, for her suggestions and encouragement at all the stages of the project work.

We want to convey our thanks to our program coordinators **,**Professor, Module of Data Science and Business Systems, SRM Institute of Science and Technology, for her inputs during the project reviews and support.

We register our immeasurable thanks to our Faculty Advisor**, Associate Professor,** Module of Data Science and Business Systems, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to my guide, **, Assistant Professor**, Module of Data Science and Business Systems, SRM Institute of Science and Technology, for providing me with an opportunity to pursue my project under his/her/their mentor-ship. He/She/They provided me with the freedom and support to explore the research topics of my interest. Her/His/Their passion for solving problems and making a difference in the world has always been inspiring.

We sincerely thank the Data Science and Business Systems staff and students, SRM Institute of Science and Technology, for their help during our project. Finally, we would like to thank parents, family members, and friends for their unconditional love, constant support, and encouragement.

**STUDNENT NAME[REG NO.]**

**Abstract**

The proposed healthcare system for senior citizens aims to address the challenges of remote health monitoring without the use of wearable devices. The system utilizes advanced technologies such as mobile applications and a secure cloud infrastructure to collect and analyze real-time health data.

The user-friendly mobile application serves as the central hub for caregivers, providing instant access to their senior family members' health information. This ensures that caregivers can effectively monitor their loved ones' health and respond to any potential health issues promptly.

The system also facilitates proactive healthcare management and communication with healthcare providers. By leveraging advanced technologies, the system can identify potential health issues before they escalate into more severe conditions. This allows healthcare providers to intervene early and prevent complications.

The system's user-friendly interface and seamless integration with healthcare providers enable caregivers to easily access and share their senior family members' health data. This ensures that healthcare providers have the necessary information to provide effective care and support.

The system's secure cloud infrastructure ensures the privacy and security of the collected health data. This is crucial, as the health data of senior citizens is sensitive and must be protected to maintain their trust and confidence.

In summary, the proposed healthcare system for senior citizens offers a promising solution to address the challenges of remote health monitoring without the use of wearable devices. By leveraging advanced technologies and a user-friendly mobile application, the system empowers caregivers to effectively monitor and manage their senior family members' health. This ensures that senior citizens receive the necessary care and support to maintain their well-being.

While the project is still undergoing testing, it showcases a potential advancement in elder care, highlighting the potential benefits of leveraging technology to empower families in ensuring the well-being of their senior loved ones.

**INDEX**

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Topic** | **Page. No.** |
| I | Acknowledgement | 1 |
| II | Abstract | 2 |
| III | List of Abbreviations | 4 |
| 1 | Introduction | 5 |
| 2 | Literature Survey | 18 |
| 3 | Problem Statement and Proposed Solution | 19 |
| 4 | Methodology | 28 |
| 5 | Requirements | 39 |
| 6 | System Design | 40 |
| 7 | Coding and Testing | 41 |
| 8 | Results and Analysis | 71 |
| 9 | Conclusion and future works | 76 |
| 10 | References | 80 |
| 11 | Appendices | 81 |

**List of Abbreviations**

**AI**: Artificial Intelligence

**ANN:** Artificial Neural Network

**BMI:** Body Metabolism Index

**CBTI:** Clinical Multiple Problems in Defect Segmentations Imaging

**RESNET-50:** Convolutional Neural Network

**CRM**: Customer Relationship Management

**DL**: Deep Learning

**EEG**: Electro-encephalogram

**EWS**: Early Warning System

**FED:** Feature Engineering and Designing

**KPI**: Key Performance Indicator

**ML**: Machine Learning

**DEFECT SEGMENTATION:** Magnetic Resonance Imaging

**PET:** Positron Emission Tomography

**R&D:** Research and Development

**SRG:** Smart Report Generation

**VGG:** Visual Geometry Group

**XML:** Extensive Mark-up Language

**Chapter 1**

**Introduction**

A Smart Healthcare System is a comprehensive approach to healthcare delivery that utilizes advanced technologies to improve efficiency, patient outcomes, and overall quality of care. The system is designed to provide real-time health insights, personalized care, and increased patient engagement while maintaining rigorous security and privacy standards.

The system incorporates remote monitoring, data analytics, mobile applications, IoT devices, interoperability, and telemedicine services. These components work together to create a seamless, interconnected system that can monitor and manage the health of senior citizens, even when they are not in the presence of healthcare providers.

Remote monitoring involves the use of IoT devices to collect real-time health data from senior citizens. This data is then transmitted to a secure cloud infrastructure for analysis. Data analytics utilize advanced algorithms to identify patterns and trends in the collected health data. This allows healthcare providers to identify potential health issues before they escalate into more severe conditions, enabling them to intervene early and prevent complications.

Mobile applications serve as the central hub for caregivers, providing instant access to their senior family members' health information. This ensures that caregivers can effectively monitor their loved ones' health and respond to any potential health issues promptly. The user-friendly interface of the mobile application allows caregivers to easily access and share their senior family members' health data, ensuring that healthcare providers have the necessary information to provide effective care and support.

Interoperability ensures that the system can seamlessly integrate with existing healthcare systems and services. This allows the system to leverage the full potential of advanced technologies while minimizing disruptions to existing healthcare processes.

Telemedicine services enable healthcare providers to communicate with their patients remotely, even when they are not in the same location. This allows healthcare providers to provide timely care and support to senior citizens, regardless of their geographical location.

In conclusion, a Smart Healthcare System offers a transformative approach to healthcare delivery that utilizes advanced technologies to improve efficiency, patient outcomes, and overall quality of care. By incorporating remote monitoring, data analytics, mobile applications, IoT devices, interoperability, and telemedicine services, the system empowers caregivers to effectively monitor and manage their senior family members' health, ensuring their well-being and overall healthcare experience.

**Chapter 2**

**Literature Survey**

The rise in Smart Healthcare Systems can be attributed to several factors, including the increasing prominence of telemedicine platforms for remote consultations, the popularity of wearable devices like fitness trackers and smartwatches for monitoring health metrics, and the increasing adoption of remote patient monitoring, especially for chronic conditions, which leverages wearables for comprehensive health monitoring.

Telemedicine platforms have gained significant prominence during the COVID-19 pandemic, as they offer a safe and convenient way to access healthcare services remotely. These platforms enable patients to consult with healthcare providers from the comfort of their homes, reducing the risk of exposure to infectious diseases. Telemedicine platforms can also help reduce healthcare costs, increase access to care, and improve patient outcomes.

Wearable devices like fitness trackers and smartwatches have gained popularity in recent years due to their ability to monitor various health metrics, such as heart rate, blood pressure, sleep patterns, and physical activity levels. These devices can help individuals take a more proactive approach to their health, enabling them to make informed decisions about their lifestyle choices. Additionally, healthcare providers can use data from wearable devices to monitor their patients' health and provide more personalized care.

Remote patient monitoring is increasingly adopted, especially for chronic conditions, as it allows healthcare providers to monitor their patients' health in real-time, even when they are not physically present. Remote patient monitoring can help healthcare providers detect potential health issues early, prevent complications, and reduce hospital readmissions. By leveraging wearable devices for comprehensive health monitoring, remote patient monitoring can help individuals with chronic conditions manage their health more effectively, leading to improved quality of life and reduced healthcare costs.

Literature surveys on Smart Healthcare Systems have highlighted several benefits, including improved patient outcomes, increased access to care, reduced healthcare costs, and enhanced patient engagement. However, these surveys have also identified several challenges, such as data privacy and security concerns, the need for standardization and interoperability, and the digital divide between those who have access to technology and those who do not. To address these challenges, it is essential to develop and implement policies and regulations that ensure the secure and ethical use of technology in healthcare, promote standardization and interoperability, and increase access to technology for all individuals.

Integrated Health Information Systems are a comprehensive approach to managing healthcare data that aims to streamline data management in the healthcare landscape. This is achieved through the ongoing development of electronic health records (EHRs) that collect and store patient health information in a centralized and accessible manner.

One of the primary benefits of Integrated Health Information Systems is improved communication among healthcare providers. By integrating health information systems, healthcare providers can access a patient's complete medical history, including their medical records, test results, and medications. This facilitates enhanced communication between healthcare providers, reducing the risk of medical errors and improving patient

**Chapter 3**

**Problem Statement and Proposed Solution**

The development of Smart Healthcare Systems has the potential to transform the healthcare landscape, offering numerous benefits for both healthcare providers and end-users. However, several challenges must be addressed to ensure the successful implementation and adoption of these systems. This essay will explore the challenges of technological adoption, privacy and security, cost management, reliability and accuracy, limited technology access, and scalability in the development of Smart Healthcare Systems.

Technological Adoption

One of the most significant challenges in developing Smart Healthcare Systems is overcoming potential resistance from elderly users to adopt and use new technologies. While the benefits of these systems are clear, many seniors may be hesitant to use them due to a lack of familiarity or comfort with technology. To address this challenge, it is essential to design the system with an intuitive and user-friendly interface that is easy to navigate and understand. This requires careful consideration of the needs and preferences of elderly users, including font size, color contrast, and button placement.

In addition to a user-friendly interface, adequate training and support are crucial for promoting the adoption and use of Smart Healthcare Systems among elderly users. Healthcare providers must provide comprehensive training on how to use the system, including step-by-step instructions and troubleshooting tips. This training should be tailored to the needs and abilities of individual users, ensuring that they feel confident and comfortable using the system. Ongoing support should also be available to address any questions or concerns that arise during use.

Privacy and Security

Managing sensitive health data securely is a critical challenge in the development of Smart Healthcare Systems. With the increasing use of digital health records, there is a growing risk of data breaches and cyber attacks. To address this challenge, healthcare providers must ensure compliance with data protection regulations, such as the Health Insurance Portability and Accountability Act (HIPAA) in the United States. Additionally, implementing robust encryption protocols can help protect user data and build user trust.

Healthcare providers must also be transparent about how they collect, use, and share user data. Clear and concise privacy policies should be developed and communicated to users, outlining the types of data collected, how it is used, and with whom it is shared. Users should also have the ability to opt-out of data sharing or delete their data at any time.

Cost Management

Balancing the financial implications of developing and maintaining a Smart Healthcare System is crucial to ensure affordability for both healthcare providers and end-users. While the benefits of these systems are clear, the cost of development and maintenance can be prohibitive for many healthcare providers. To address this challenge, healthcare providers must carefully consider the costs and benefits of implementing a Smart Healthcare System, including the potential return on investment.

Additionally, healthcare providers must consider the costs and benefits of different system features and functionality. While some features may be essential for the effective use of the system, others may be nice-to-have but not necessary. By prioritizing essential features, healthcare providers can reduce the overall cost of the system while still providing value to end-users.

Reliability and Accuracy

Ensuring the reliability and accuracy of Smart Healthcare Systems is critical to maintaining credibility and user trust. Healthcare providers must ensure that the system consistently provides reliable and accurate health data, reducing the risk of medical errors and improving patient outcomes. This requires careful consideration of the types of data collected, the accuracy of the data, and the methods used to analyze and interpret the data.

Healthcare providers must also ensure that the system is regularly updated and maintained, addressing any bugs or issues that arise during use. Regular system audits and testing can help identify potential issues and ensure that the system is functioning optimally.

Limited Technology Access

Limited access to smartphones or reliable internet connections among seniors can be a significant barrier to the adoption and use of Smart Healthcare Systems. To address this challenge, healthcare providers must consider alternative methods of access, such as through a web-based platform or a dedicated device.

Additionally, healthcare providers must consider the needs and abilities of users with different levels of technology literacy. Providing simple and clear instructions, as well as ongoing support, can help promote inclusivity and ensure that all users can access and use the system.

Scalability

Designing Smart Healthcare Systems to accommodate a growing number of users and evolving healthcare needs is essential for long-term success. Healthcare providers must ensure that the system architecture is scalable and flexible, allowing for the addition of new features and functionality as needed.

Additionally, healthcare providers must consider the potential for growth and expansion of the system, including the ability to integrate with other healthcare systems and services. By designing the system with scalability in mind, healthcare providers can ensure that it can evolve and adapt to meet the

One way to ensure scalability is to adopt a modular system architecture that allows for the addition of new features and functionality as needed. This approach enables healthcare providers to start with a basic system and add new modules over time, as the needs and demands of their users evolve. This approach also allows for greater flexibility in the system design, enabling healthcare providers to adapt to changing regulations, technologies, and user needs.

Another critical factor in ensuring scalability is the use of open standards and interoperability. By adopting open standards, healthcare providers can ensure that their Smart Healthcare Systems can easily integrate with other systems and services, both within and outside the healthcare industry. This approach can help reduce vendor lock-in, promote innovation, and enable healthcare providers to leverage the latest technologies and best practices.

Interoperability is also essential for ensuring that Smart Healthcare Systems can share data and information across different healthcare providers, settings, and devices. By enabling seamless data exchange, healthcare providers can improve care coordination, reduce medical errors, and enhance patient outcomes.

However, achieving interoperability is not without its challenges. Healthcare providers must navigate complex regulatory and technical landscapes, ensuring compliance with data protection regulations, such as HIPAA and the General Data Protection Regulation (GDPR) in the European Union. They must also consider the potential for data fragmentation, ensuring that data is accurate, complete, and up-to-date, even as it flows across different systems and devices.

To address these challenges, healthcare providers must adopt a holistic approach to interoperability, considering the technical, semantic, and organizational aspects of data exchange. This approach requires careful consideration of data standards, such as Fast Healthcare Interoperability Resources (FHIR) and Health Level Seven International (HL7), as well as the use of application programming interfaces (APIs) and other integration technologies.

Moreover, healthcare providers must establish clear policies and procedures for data exchange, ensuring that all stakeholders understand their roles and responsibilities in sharing and using health data. This approach requires ongoing collaboration and communication among healthcare providers, patients, and other stakeholders, promoting a shared understanding of the benefits and challenges of data exchange.

Smart Healthcare Systems have the potential to transform the healthcare landscape, offering numerous benefits for both healthcare providers and end-users. However, the successful implementation and adoption of these systems require careful consideration of several challenges, including technological adoption, privacy and security, cost management, reliability and accuracy, limited technology access, and scalability.

By addressing these challenges, healthcare providers can ensure that Smart Healthcare Systems provide value to end-users, improve patient outcomes, and promote operational efficiency. However, this requires a holistic and collaborative approach, involving healthcare providers, patients, policymakers, and other stakeholders in the design, implementation, and evaluation of these systems.

The future of Smart Healthcare Systems is bright, with emerging technologies, such as artificial intelligence, machine learning, and the Internet of Things (IoT), providing new opportunities for innovation and improvement. By embracing these technologies and addressing the challenges outlined in this essay, healthcare providers can unlock the full potential of Smart Healthcare Systems, promoting better health and well-being for all

Our proposed idea is to develop an innovative healthcare application specifically designed for senior citizens that prioritizes ease of use, accessibility, and advanced features to enhance their health and well-being. This application will include unique features such as voice or manual input for medicine timers using Natural Language Processing (NLP), an emergency button, a periodic alarm, a medicine tracker for doctors, and a booking feature for appointments.

Voice or Manual Input for Medicine Timer using NLP The application will allow senior citizens to input their medication schedules using voice or manual input. The use of NLP technology will enable the app to understand and interpret user input accurately, providing reminders at the appropriate times. This feature will help seniors remember to take their medication as prescribed, reducing the risk of medication non-adherence and associated health complications.

Emergency Button An emergency button will be included in the application, allowing seniors to alert their caretakers in case of an emergency. When the emergency button is activated, the application will send a notification to the caretaker, along with the senior's live location. This feature will provide seniors with a sense of security and ensure that they can quickly receive help in case of an emergency.

Periodic Alarm In addition to medication reminders, the application will include a periodic alarm feature that enables caretakers to be sure that the patient is safe and has responded by turning off the alarm. This feature will help caretakers monitor the safety and well-being of senior citizens, even when they are not physically present.

Medicine Tracker for Doctors The application will include a medicine tracker feature that allows doctors to check when the patient did not take their medication. This feature will help healthcare providers monitor medication adherence and identify potential issues, allowing them to intervene early and provide additional support to seniors who are struggling with medication management.

Appointment Booking Feature The application will include a booking feature for appointments, allowing seniors to schedule appointments with healthcare providers easily. This feature will help seniors manage their healthcare needs and ensure that they receive the necessary care and support.

Additional Features The application will include several additional features to enhance the user experience and provide value to senior citizens. These features include:

Customizable Reminders The application will allow users to set up customizable reminders for each medication, providing notifications or alarms as a reminder to take their pills at the specified times. Users will be able to set multiple reminders and adjust their preferences.

Refill Reminders The application will remind users to refill their medications before they run out, helping prevent gaps in medication supply and ensuring continuous therapy.

Medication Information The application will provide comprehensive information about each medication, including its purpose, instructions, potential side effects, and interactions. This will empower users to have a better understanding of their medications.

Medication Adherence Tracking The application will allow users to track their medication intake by letting them mark doses as taken within the app. This feature will help users keep a record of their medication adherence and provide insights into their compliance.

Medication History The application will maintain a log of medication history, recording when doses were taken, missed, or rescheduled. This feature will allow users to review their medication adherence over time and identify any patterns or issues.

User Accounts The application will allow users to create profiles or accounts to store their medication information securely. User accounts can also enable data synchronization across multiple devices and facilitate data backup and restoration.

Visual Data Representation The application will present medication adherence data in visual formats, such as charts or graphs, to help users track their progress and identify trends. Generating reports that summarize medication adherence can also be beneficial.

Integration with Health Tracking Devices The application will integrate with health tracking devices or platforms, such as fitness trackers or smartwatches, to gather additional health data that can inform medication management and provide a holistic view of a user’s health.

Data Security The application will ensure that generated data is safely saved in the cloud and synchronized with any devices users choose to access the service. The constant availability of current information on a patient’s adherence to their medical care is another benefit of this data synchronization. Additionally, cloud sync is a requirement if the app ever needs to link with a pharmacy or other healthcare providers.

Medication Interaction Checker The application will include a feature that checks for potential interactions between different medications and provides warnings or alerts if any potential issues are detected.

Additional Reminders The application will include reminders for tasks related to medications, such as scheduling

**Chapter 4**

**Methodology**

Developing an application for senior citizens with the proposed features requires careful planning, design, and implementation. Here is a high-level methodology for developing the application:

**Requirements Gathering and Analysis**

Conduct user research and interviews with senior citizens, caretakers, and healthcare providers to understand their needs, preferences, and pain points.

Define the application's objectives, scope, and functionality.

Create user stories, use cases, and personas to guide the design and development process.

Determine the technical requirements and infrastructure needed to support the application.

Design

Create wireframes and mockups of the application's user interface, including font sizes, color contrast, and button placement.

Design the application's flow and navigation, ensuring it is intuitive and easy to use.

Develop a prototype of the application to test and validate the design.

Development

Choose a cross-platform development framework such as React Native, Flutter, or Xamarin to build the application.

Implement the voice or manual input for medicine timer using Natural Language Processing (NLP) libraries such as Google's Dialogflow or IBM Watson.

Integrate the emergency button feature with location services and real-time communication APIs such as Firebase Cloud Messaging or Twilio.

Implement the periodic alarm feature to remind the caretaker to check on the patient.

Develop the medicine tracker feature for doctors to monitor when the patient did not take the medicine.

Integrate the booking feature for appointments with calendar APIs such as Google Calendar or Outlook Calendar.

Implement robust security measures such as data encryption, secure user authentication, and access controls.

Ensure the application is scalable, reliable, and maintainable.

**Testing**

Conduct functional testing to ensure the application meets the requirements and specifications.

Perform usability testing with senior citizens, caretakers, and healthcare providers to validate the user experience and identify any issues or bugs.

Test the application on various devices and platforms to ensure compatibility and performance.

Perform security testing to ensure the application's data and infrastructure are secure.

**Deployment and Maintenance**

Deploy the application to app stores such as Google Play or Apple App Store.

Monitor the application's performance, usage, and feedback.

Provide ongoing support, maintenance, and updates to the application as needed.

1. Create a new agent in Dialogflow and define the intents and entities required for the medicine timer. For example, the intent can be "Set Medicine Timer" and the entities can be "Medicine Name", "Dosage", and "Frequency".
2. Implement a webhook to handle the intent and extract the parameters from the user input using Dialogflow's fulfillment API.
3. Convert the user input into a timestamp and set a reminder using the device's notification API.
4. Save the medicine timer data in a database or cloud storage for future reference.
5. Create a Firebase project and configure the Firebase Cloud Messaging API.
6. Implement a button in the application that triggers a Firebase Cloud Messaging request when pressed.
7. Send a push notification to the caretaker's device using the Firebase Cloud Messaging API.
8. Include the patient's location in the push notification using the device's location API.
9. Create a Firebase Realtime Database and define the medicine tracker schema.
10. Save the medicine timer data in the database when the user sets a reminder.
11. Allow the doctor to access the medicine tracker data through a web interface or mobile app.
12. Highlight any missed doses or non-compliance issues to the doctor and the caretaker.
13. Authenticate the user with Google Sign-In and request permission to access the Google

**Chapter - 5**

**Technical Requirements**

**HARDWARE REQUIREMENTS:**

1. Computer System

* A modern computer system with a minimum of 8GB RAM, 2.5 GHz processor speed, and 500GB hard drive storage.
* A reliable internet connection with a minimum download speed of 10 Mbps and upload speed of 5 Mbps.

1. Mobile Devices

* One or more mobile devices with Android or iOS operating system for testing the application.
* The mobile devices should have a minimum of 2GB RAM, 1.5 GHz processor speed, and 16GB internal storage.

1. External Devices

* A high-quality scanner or a smartphone camera with a minimum of 12MP resolution for scanning prescriptions and reports.
* A printer for printing medication schedules, appointment bookings, and other relevant documents.

1. Additional Equipment

* A set of headphones or earphones with a built-in microphone for voice input and testing.
* A USB flash drive or external hard drive for storing and transferring data and files.

**SOFTWARE REQUIREMENTS**

1. Flutter Framework

* Use the Flutter framework to create a cross-platform mobile application for both iOS and Android.
* Utilize Flutter's built-in widgets and tools to create a user-friendly interface for senior citizens.
* Implement state management techniques such as Provider or BLoC to manage the application's state and data flow.

1. Firebase Backend

* Use Firebase to manage the application's backend, including user authentication, data storage, and real-time communication.
* Implement Firebase Authentication for user registration, login, and authorization.
* Use Firebase Cloud Firestore as the application's primary database to store user data, medication schedules, and appointment bookings.
* Implement Firebase Cloud Functions for server-side logic and automation.

1. Ocr.space API

* Integrate the Ocr.space API to scan and extract text from prescription forms and medical reports.
* Use the API's OCR (Optical Character Recognition) technology to extract text from images and convert it into editable text.
* Implement the API's RESTful API endpoints to send and receive data from the application.

1. spaCy ML Framework

* Integrate the spaCy open source ML framework to analyze and extract meaningful insights from the scanned prescription data.
* Train a custom NLP (Natural Language Processing) model using spaCy to extract medication names, dosages, and frequencies from the prescription text.
* Use the trained model to analyze the scanned prescription data and extract relevant information.

1. Open SSI Encryption

* Implement end-to-end encryption using Open SSI to ensure the security and privacy of user data.
* Utilize Open SSI's open source encryption libraries to encrypt and decrypt user data, including medication schedules, appointment bookings, and scanned prescription data.
* Implement a secure key management system to ensure the safety and confidentiality of encryption keys.

1. Data Testing and Validation

* Test the application's conversion algorithm using 100 prescription samples to ensure its accuracy and reliability.
* Implement unit testing and integration testing to validate the application's functionality and performance.
* Use automated testing tools and frameworks such as Flutter's built-in testing tools to ensure the application's stability and robustness.

1. Additional Technical Specifications

* Use version control tools such as Git to manage the application's source code and track changes.
* Implement continuous integration and continuous delivery (CI/CD) pipelines to automate the build, test, and deployment process.
* Utilize cloud infrastructure and services such as Firebase Cloud Functions and Google Cloud Storage to manage the application's infrastructure and resources.
* Implement accessibility features such as large font sizes, high contrast colors, and screen reader support to ensure the application is accessible to all users, including those with disabilities.
* Follow best practices for mobile application development, including code organization

**Chapter – 6**

**System Designs**

1. Architecture The healthcare application will be designed as a client-server architecture, where the client-side will be developed using Flutter for mobile devices and the server-side will be developed using Firebase for backend and cloud infrastructure.
2. User Interface The user interface will be designed using Flutter's built-in widgets and tools. The interface will be simple, intuitive, and user-friendly, with large font sizes, high contrast colors, and easy-to-use navigation. The user interface will include the following features:

* Medicine Timer: A feature that allows users to set reminders for their medication schedules using voice or manual input. The feature will use NLP to extract relevant information from the user input and store it in the Firebase database.
* Emergency Button: A feature that allows users to send an emergency signal to their caretakers with their live location using Firebase Cloud Messaging.
* Periodic Alarm: A feature that allows caretakers to set periodic alarms to check on the user and ensure their safety.
* Medicine Tracker: A feature that allows doctors to monitor the user's medication intake and identify any issues using Firebase Cloud Functions.
* Appointment Booking: A feature that allows users to schedule appointments with healthcare providers using Google Calendar API.

1. Data Storage The healthcare application will use Firebase Cloud Firestore as the primary database to store user data, medication schedules, appointment bookings, and other relevant information. The database will be designed to ensure data consistency, availability, and scalability.
2. Data Processing The healthcare application will use Ocr.space API to extract text from scanned prescriptions and reports. The extracted text will be analyzed using spaCy ML framework to extract relevant information such as medication names, dosages, and frequencies. The analyzed data will be stored in the Firebase database.
3. Data Security The healthcare application will implement end-to-end encryption using Open SSI to ensure the security and privacy of user data. The encryption mechanism will be applied to all data transfers, including voice input, medication schedules, appointment bookings, and scanned prescription data.
4. Cloud Infrastructure The healthcare application will use Firebase for backend and cloud infrastructure, including Firebase Authentication, Firebase Cloud Functions, Firebase Cloud Messaging, and Firebase Cloud Storage. The cloud infrastructure will ensure scalability, availability, and reliability of the application.
5. Testing and Validation The healthcare application will be tested and validated using 100 prescription samples to ensure the accuracy and reliability of the conversion algorithm. The application will undergo unit testing, integration testing, and functional testing to ensure its stability and robustness.
6. Deployment and Maintenance The healthcare application will be deployed to app stores such as Google Play and Apple App Store. The application will be continuously monitored and maintained to ensure its performance, security, and availability.
7. User Interface The user interface of the healthcare application will be designed using Flutter's built-in widgets and tools. The interface will be simple, intuitive, and user-friendly, with large font sizes, high contrast colors, and easy-to-use navigation. The interface will be optimized for senior citizens, taking into account their specific needs and limitations.
8. The home screen of the application will display the user's medication schedules, appointment bookings, emergency button, and periodic alarm. The user can navigate to different sections of the application using the bottom navigation bar.
9. The medicine timer feature will allow users to set reminders for their medication schedules using voice or manual input. The voice input will use NLP to extract relevant information from the user input and store it in the Firebase database. The manual input will allow users to enter the medication name, dosage, frequency, and start and end dates manually.
10. The emergency button feature will allow users to send an emergency signal to their caretakers with their live location using Firebase Cloud Messaging. The caretaker will receive a notification with the user's location and can take appropriate action.
11. The periodic alarm feature will allow caretakers to set periodic alarms to check on the user and ensure their safety. The alarm will display a message on the user's device, prompting them to respond to the alarm.
12. The medicine tracker feature will allow doctors to monitor the user's medication intake and identify any issues using Firebase Cloud Functions. The feature will use Ocr.space API to extract text from scanned prescriptions and reports, and spaCy ML framework to analyze the data and extract relevant information.
13. The appointment booking feature will allow users to schedule appointments with healthcare providers using Google Calendar API. The feature will allow users to select the appointment date and time, enter the healthcare provider's name and contact information, and add notes or comments.
14. Data Storage The healthcare application will use Firebase Cloud Firestore as the primary database to store user data, medication schedules, appointment bookings, and other relevant information. The database will be designed to ensure data consistency, availability, and scalability.
15. The database schema will be designed to minimize data redundancy and ensure data normalization. The schema will include collections for users, medication schedules, appointment bookings, and healthcare providers.
16. The database will ensure data consistency and availability using Firebase's real-time synchronization and offline support. The database will be optimized for mobile devices, with low latency and high throughput.
17. Data Processing The healthcare application will use Ocr.space API to extract text from scanned prescriptions and reports. The extracted text will be analyzed using spaCy ML framework to extract relevant information such as medication names, dosages, and frequencies.
18. The Ocr.space API will be integrated into the application using RESTful API endpoints. The API will allow the application to send and receive data in JSON format, with support for various image formats such as JPEG, PNG, and PDF.
19. The spaCy ML framework will be integrated into the application using Python scripts. The framework will be trained on a dataset of 100 prescription samples to ensure the accuracy and reliability of the conversion algorithm.
20. The analyzed data will be stored in the Firebase database, with support for real-time synchronization and offline access.
21. Data Security The healthcare application will implement end-to-end encryption using Open SSI to ensure the security and privacy of user data. The encryption mechanism will be applied to all data transfers, including voice input, medication schedules, appointment bookings, and scanned prescription data.
22. The encryption mechanism will use a combination of symmetric and asymmetric encryption algorithms, such as AES and RSA. The encryption keys will be generated and managed using Open SSI's open source encryption libraries.
23. The encryption mechanism will ensure data confidentiality, integrity, and authenticity, with support for forward secrecy and perfect forward secrecy.
24. Cloud Infrastructure The healthcare application will use Firebase for backend and cloud infrastructure, including Firebase Authentication, Firebase Cloud Functions, Firebase Cloud Messaging, and Firebase Cloud Storage.
25. Firebase Authentication will be used for user registration, login, and authorization. The authentication mechanism will support various authentication methods such as email and password, phone number, and third-party providers.
26. Firebase Cloud Functions will be used for server-side logic and automation, with support for real-time data processing and event-driven triggers.
27. Firebase Cloud Messaging will be used for push

Here is a high-level diagram of the proposed system design:

A diagram with text on it

Description automatically generated A diagram with text and circles

Description automatically generated A diagram of a diagram

Description automatically generated

Fig. Architecture diagram for the project.

**Chapter-7**

**Coding and Testing**

**Login page**

**Activity\_Login.xml**

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".LoginActivity">

<EditText

android:id="@+id/editTextUsername"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Username"

android:inputType="text"/>

<EditText

android:id="@+id/editTextPassword"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@id/editTextUsername"

android:layout\_marginTop="16dp"

android:hint="Password"

android:inputType="textPassword"/>

<Button

android:id="@+id/buttonLogin"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@id/editTextPassword"

android:layout\_marginTop="16dp"

android:text="Login"/>

</RelativeLayout>

**LoginActivity.java:**

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

public class LoginActivity extends AppCompatActivity {

private EditText editTextUsername, editTextPassword;

private Button buttonLogin;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_login);

editTextUsername = findViewById(R.id.editTextUsername);

editTextPassword = findViewById(R.id.editTextPassword);

buttonLogin = findViewById(R.id.buttonLogin);

buttonLogin.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

String username = editTextUsername.getText().toString().trim();

String password = editTextPassword.getText().toString().trim();

// Here you can add your login logic, such as checking credentials

// For simplicity, let's just show a toast message indicating successful login

if (isValidCredentials(username, password)) {

// Login successful

Toast.makeText(LoginActivity.this, "Login Successful", Toast.LENGTH\_SHORT).show();

} else {

// Login failed

Toast.makeText(LoginActivity.this, "Invalid Credentials", Toast.LENGTH\_SHORT).show();

}

}

});

}

// Example method to validate credentials

private boolean isValidCredentials(String username, String password) {

// Add your validation logic here, such as checking against a database

return username.equals("admin") && password.equals("password");

}

}

**activity\_home.xml**

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".HomeActivity">

<TextView

android:id="@+id/textViewWelcome"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Welcome to Home Page!"

android:textSize="24sp"

android:layout\_centerInParent="true"/>

<Button

android:id="@+id/buttonLogout"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Logout"

android:layout\_below="@id/textViewWelcome"

android:layout\_centerHorizontal="true"

android:layout\_marginTop="24dp"/>

</RelativeLayout>

**JavaScript (NLP back end server)**

const express = require('express');

const bodyParser = require('body-parser');

const { SessionsClient } = require('@google-cloud/dialogflow').v2; // Import SessionsClient from @google-cloud/dialogflow

const { v4: uuidv4 } = require('uuid');

const schedule = require('node-schedule');

const app = express();

const port = process.env.PORT || 3000;

// Initialize Dialogflow session client

const sessionClient = new SessionsClient();

// Middleware to parse JSON requests

app.use(bodyParser.json());

// Function to set a timer

function setTimer(medicine, timeOfDay) {

// Initialize variables to store hour and minute

let hour = 0;

let minute = 0;

// Determine the time of day and set hour accordingly

switch (timeOfDay) {

case 'morning':

hour = 8; // 8 AM

break;

case 'afternoon':

hour = 12; // 12 PM

break;

case 'night':

hour = 22; // 10 PM

break;

default:

console.error('Invalid time of day:', timeOfDay);

return; // Exit function if time of day is invalid

}

// Schedule the timer

const job = schedule.scheduleJob({ hour: hour, minute: minute }, function () {

console.log(Time to take ${medicine}!);

});

}

// Route handler for POST requests to /webhook

app.post('/webhook', async (req, res) => {

try {

// Extract parameters from the request body

const { medicine, timeOfDay } = req.body.queryResult.parameters;

// Perform logic based on parameters (e.g., set timer for medicine at specified time)

setTimer(medicine, timeOfDay);

// Send a request to Dialogflow

const sessionId = uuidv4();

const sessionPath = sessionClient.projectAgentSessionPath('your-project-id', sessionId); // Replace 'your-project-id' with your actual Dialogflow project ID

const request = {

session: sessionPath,

queryInput: {

text: {

text: req.body.queryResult.queryText,

languageCode: 'en-US',

},

},

};

const responses = await sessionClient.detectIntent(request);

const fulfillmentText = responses[0].queryResult.fulfillmentText;

// Construct response to send back to Dialogflow

const response = {

fulfillmentText: fulfillmentText,

};

// Send response to Dialogflow

res.json(response);

} catch (error) {

console.error('Error processing intent:', error);

res.status(500).send('Internal Server Error');

}

});

// Route handler for GET requests to /webhook (just for testing, you might remove this in production)

app.get('/webhook', (req, res) => {

res.send('Hello, this is the webhook endpoint!');

});

// Start the server

app.listen(port, () => {

console.log(Server is running on port ${port});

});

A screenshot of a computer

Description automatically generated

A computer screen shot of a program

Description automatically generated A computer screen with text on it

Description automatically generated A computer screen with text on it

Description automatically generated A computer screen with many colorful lines

Description automatically generated

**Chapter-8**

**Results and Analysis**

Smart Healthcare Systems highlights several key themes, including the use of advanced technologies for improved patient outcomes, the integration of health information systems, and the challenges of technological adoption, privacy and security, cost management, reliability and accuracy, limited technology access, and scalability.

One study found that the use of remote monitoring and data analytics in Smart Healthcare Systems can lead to improved patient outcomes, including better medication adherence and reduced hospital readmissions. The integration of health information systems can also facilitate enhanced communication among healthcare providers, leading to improved care coordination and patient outcomes.

However, several challenges must be addressed to ensure the successful implementation and adoption of Smart Healthcare Systems. Technological adoption can be a significant barrier, particularly among elderly users who may be hesitant to use new technologies. To address this challenge, it is essential to design the system with an intuitive and user-friendly interface that is easy to navigate and understand. Adequate training and support are also crucial for promoting the adoption and use of Smart Healthcare Systems among elderly users.

Privacy and security are critical challenges in the development of Smart Healthcare Systems. With the increasing use of digital health records, there is a growing risk of data breaches and cyber attacks. To address this challenge, healthcare providers must ensure compliance with data protection regulations and implement robust encryption protocols to protect user data and build user trust.

Cost management is also a significant challenge in the development of Smart Healthcare Systems. While the benefits of these systems are clear, the cost of development and maintenance can be prohibitive for many healthcare providers. Healthcare providers must carefully consider the costs and benefits of implementing a Smart Healthcare System, including the potential return on investment.

Reliability and accuracy are critical to maintaining credibility and user trust. Healthcare providers must ensure that the system consistently provides reliable and accurate health data, reducing the risk of medical errors and improving patient outcomes. Regular system audits and testing can help identify potential issues and ensure that the system is functioning optimally.

Limited technology access can be a significant barrier to the adoption and use of Smart Healthcare Systems. Healthcare providers must consider alternative methods of access, such as through a web-based platform or a dedicated device, and provide simple and clear instructions to promote inclusivity and ensure that all users can access and use the system.

Scalability is essential for long-term success. Healthcare providers must ensure that the system architecture is scalable and flexible, allowing for the addition of new features and functionality as needed. The potential for growth and expansion of the system must also be considered, including the ability to integrate with other healthcare systems and services.

To develop the proposed application, a Flutter framework can be used for cross-platform mobile application development. Firebase can be used for backend services, including user authentication and data storage. Ocr.space API can be used for scanning prescriptions and reports, while spaCy ML framework can be used for analyzing the data. Encryption mechanisms can be implemented using open SSI to ensure the security and privacy of user data.

The hardware requirements for the proposed application include a modern computer system with a minimum of 8GB RAM, 2.5 GHz processor speed, and 500GB hard drive storage. A reliable internet connection with a minimum download speed of 10 Mbps and upload speed of 5 Mbps is also required. Mobile devices with Android or iOS operating systems are required for testing the application. A high-quality scanner or a smartphone camera with a minimum of 12MP resolution is required for scanning prescriptions and reports. A printer is also required for printing medication schedules, appointment bookings, and other relevant documents.

The system design for the proposed application includes a client-server architecture, with the client-side developed using Flutter for mobile devices and the server-side developed using Firebase for backend and cloud infrastructure. The user interface will be designed using Flutter's built-in widgets and tools, with large font sizes, high contrast colors, and easy-to-use navigation. The medicine timer feature will allow users to set reminders for their medication schedules using voice or manual input, with NLP technology used to extract relevant information from the user input and store it in the Firebase database. The emergency button feature will allow users to send an emergency signal to their caretakers with their live location using Firebase Cloud Messaging. The periodic alarm feature will allow caretakers to set periodic alarms to check on the user and ensure their safety. The medicine tracker feature will allow doctors to monitor the user's medication intake and identify any issues using Firebase Cloud Functions. The appointment booking feature will allow users to schedule appointments with healthcare providers using Google Calendar API.

To ensure the reliability and accuracy of the proposed application, the medication timer feature will

**Chapter-9**

**Conclusion and Future Developments**

In conclusion, Smart Healthcare Systems have the potential to transform the healthcare landscape, offering numerous benefits for both healthcare providers and end-users. However, several challenges must be addressed to ensure the successful implementation and adoption of these systems.

The proposed idea for the development of an application for senior citizens addresses several of these challenges, including technological adoption, privacy and security, cost management, reliability and accuracy, limited technology access, and scalability. The proposed application includes unique features such as voice or manual input for medicine timer using NLP, emergency button which alerts the caretaker by sending live location, periodic alarm which enables caretaker to be sure that the patient is safe and has responded by turning off the alarm, medicine tracker for doctor to check when the patient did not take the medicine, and booking feature for appointments.

The proposed application can be developed using the Flutter framework for cross-platform mobile application development and Firebase for backend services. Ocr.space API can be used for scanning prescriptions and reports, while spaCy ML framework can be used for analyzing the data. Encryption mechanisms can be implemented using open SSI to ensure the security and privacy of user data.

The hardware requirements for the proposed application include a modern computer system with a minimum of 8GB RAM, a reliable internet connection, mobile devices with Android or iOS operating systems, a high-quality scanner or a smartphone camera with a minimum of 12MP resolution, and a printer.

The system design for the proposed application includes a client-server architecture, with the client-side developed using Flutter for mobile devices and the server-side developed using Firebase for backend and cloud infrastructure. The user interface will be designed using Flutter's built-in widgets and tools, with large font sizes, high contrast colors, and easy-to-use navigation.

The proposed application has the potential to improve medication adherence, emergency response, and care coordination for senior citizens. However, there are also several areas for future development, including the integration of additional health data sources, such as wearable devices, and the development of predictive analytics models to identify potential health issues before they become serious.

In summary, the development of Smart Healthcare Systems offers a promising avenue for improving healthcare delivery and outcomes, and the proposed application has the potential to make a significant contribution to this field. However, ongoing research and development are necessary to ensure the continued success and adoption of these systems.

**References**

Exploring the benefits and challenges of smart healthcare systems: A literature survey. International Journal of Medical Informatics, 133, 104288.

The rise of telemedicine in the era of COVID-19: A systematic review. Frontiers in Public Health, 9, 575244.

Wearable technology and remote patient monitoring: A systematic review. Journal of Medical Internet Research, 21(5), e13988.

Smart healthcare: A review of the literature. Journal of Ambient Intelligence and Humanized Computing, 10(2), 379-396.

The future of telehealth: A review of emerging technologies and trends. Frontiers in Public Health, 8, 1-11.

Remote patient monitoring: A review of the literature. Journal of Medical Systems, 43(6), 155.

Wearable devices and mobile health: A review of current trends and future directions. Journal of the American Medical Association, 316(18), 1863-1864.

Telemedicine and eHealth: A review of current trends and future directions. Telemedicine and e-Health, 26(7), 705-712.

Smart healthcare systems: A literature review and future directions. IEEE Access, 7, 49357-49369.

Wearable technology in healthcare: A review of the literature. Journal of Healthcare Informatics Research, 1(1), 78-93.

Remote patient monitoring for chronic disease management: A review of the literature. Journal of Healthcare Informatics Research, 3(2), 149-167.

Telemedicine and patient outcomes: A systematic review and meta-analysis. Medical Care, 53(5), 417-425.

Chen, L., & Xu, W. (2019). Research on the health service system and health information perception terminal design of the APP health service platform. Journal of Intelligent & Fuzzy Systems, 37(5), 3915-3923.

Li, X., Li, Y., & Wang, Y. (2020). Exploration and application of mobile medical devices in smart healthcare. Journal of Medical Systems, 44(8), 245.

Martínez-Pérez, B., García-Ceberino, E., & Fernández-Carrasco, N. (2017). A systematic review of mobile medical applications for chronic disease management. Journal of Medical Systems, 41(9), 188.

Mohammadzadeh, M., & Safdari, R. (2019). Design and development of a mobile health monitoring and diagnosis system based on IoT. Journal of Ambient Intelligence and Humanized Computing, 10(5), 1799-1810.

Sezgin, M. (2019). Mobile health applications: A critical review of the literature. Health Informatics Journal, 25(2), 838-854.

Ahmedani, B. K., Peterson, E. L., Jin, L., Caverly, T. J., Cui, X., Creed, M. F., ... & Simon, G. E. (2016). Adherence to evidence-based recommendations for self-management behaviors among top-rated diabetes applications. Journal of Medical Internet Research, 18(8), e225.

Kvedar, J. C., Coye, M. J., Everett, A., Patel, V., Pizziferri, L., & Shpple, K. (2017). Mobile health: A catalyst for engaging patients, improving outcomes, and transforming healthcare. Health Affairs, 36(11), 1960-1965.

Chau, K. W., Liu, X., & Hu, P. J. (2010). Mobile healthcare: transforming healthcare services in China. China Review, 10(1), 25-45.

Wang, J., Liu, X., & Guo, Y. (2019). Design and implementation of a mobile healthcare system based on cloud computing. Journal of Ambient Intelligence and Humanized Computing, 10(5), 1757-1766.

Lin, H., Chang, C., & Chen, Y. (2017). A review of mobile health technologies for patient self-management of chronic diseases. Journal of Medical Systems, 41(5), 113.

Lee, Y. S., & Lee, S. W. (2018). A study on the application of smart healthcare technology to chronic disease management. Sustainability, 10(11), 3934.

Chatterjee, S., & Dolui, S. (2020). Smart healthcare: A review on enabling technologies, applications and challenges. Journal of Ambient Intelligence and Humanized Computing, 11(5), 2209-2228.

Li, Y., Li, X., & Wang, Y. (2020). Research on the application of artificial intelligence technology in smart healthcare. Journal of Intelligent & Fuzzy Systems, 39(5), 4551-4557.

Chen, L., & Xu, W. (2019). Research on the health service system and health information perception terminal design of the APP health service platform. Journal of Intelligent & Fuzzy Systems, 37(5), 3915-3923.

Li, X., Li, Y., & Wang, Y. (2020). Exploration and application of mobile medical devices in smart healthcare. Journal of Medical Systems, 44(8), 245.

Martínez-Pérez, B., García-Ceberino, E., & Fernández-Carrasco, N. (2017). A systematic review of mobile medical applications for chronic disease management. Journal of Medical