IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 8, April 2023

Smart Health: A Data-Driven Approach for Personalized Health Consulting via a Web-Based Platform

Nikhil Tiwari[1], Sarthak Srivastava[2], Dr Ashima Mehta[3]
Under Graduate Students, Department of Computer Science Engineering[1,2]
Head of Department, Department of Computer Science Engineering[3]
Dronacharya College of Engineering Gurgaon, Haryana, India

Abstract: *This system aims at maintaining patient health records and even getting appointments from various doctors for related treatments. The system user must register as a member of this system and keep updating his medical history. Patients can then select from a list of specialized doctors for respective treatments such as (skin specialist, ENT specialist cardiologist etc) at particular locations. Patients may also select suitable appointment timings for their meeting. Health is one of the most important and fundamental things in everyone's life. To maintain a sound health, everyone needs to consult with doctors. For the time being it become difficult to consult with doctors during the time of need. The Smart Health Consulting System will be able to solve this particular problem. The main purpose of SMART HEALTH CONSULTING SYSTEM is to maintain patients' health record, manage appointment for various treatments within short time and patients will get the opportunity to consult with the chosen doctor. In the case of emergency or if the area is unknown for patient, then in this mean situation knowing the right address or contact info of doctor is really important. This website will provide all the current and authentic information that will assist the user to get the right information. On the other hand, this process will be less time consuming from find a doctor to get the treatment by identifying the diseases.*

Keywords: Smart Health, Personalized consulting, Web-based platform, Data-Driven approach.

I. INTRODUCTION

Today's era is the one of informatization. With the advancement of technology and scientific theory, traditional medicine with biotechnology as its core, has gradually begun to digitize and to information. And smart healthcare incorporating a new generation of information technology has emerged. Smart healthcare is not just a simple technological advancement, but also an all-round, multi-level change. This change is embodied in the following: medical model changes (from diseasecantered to patient-cantered care), informatization construction changes (from clinical informatization to regional medical informatization), changes in medical management (from general management to personalized management), and changes in the concept of prevention and treatment (from focusing on disease treatment to focusing on preventive healthcare). These changes focus on meeting the individual needs of people while improving the efficiency of medical care, which greatly enhances the medical and health service experience, and represent the future development direction of modern medicine. This review will start from the concept of smart healthcare, then briefly introduce the key technologies supporting smart healthcare and explain the achievements and challenges of it by reviewing the application status of these technologies in important medical fields, before finally putting forward the future prospects of smart healthcare.

Health is one of the most important and fundamental things in everyone's life. To maintain a sound health, everyone needs to consult with doctors. For the time being it become difficult to consult with doctors during the time of need. The Smart Health Consulting System will be able to solve this particular problem. Smart healthcare was born out of the concept of "Smart Planet" proposed by IBM (Armonk, NY, USA) in 2009. Simply put, Smart Planet is an intelligent infrastructure that uses sensors to perceive information, transmits information through the internet of things (IoT), and processes the information using supercomputers and cloud computing. It can coordinate social systems and integrate

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 8, April 2023

them to realize the dynamic and refined management of human society. Smart healthcare is a health service system that uses technology such as wearable devices, IoT, and mobile internet to dynamically access information, connect people, materials and institutions related to healthcare, and then actively manages and responds to medical ecosystem needs in an intelligent manner. Smart healthcare can promote interaction between all parties in the healthcare field, ensure that participants get the services they need, help the parties make informed decisions, and facilitate the rational allocation of resources. In short, smart healthcare is a higher stage of information construction in the medical field.

II. KEY TECHNOLOGIES OF SMART HEALTHCARE

Smart healthcare consists of multiple participants, such as doctors and patients, hospitals, and research institutions. It is an organic whole that involves multiple dimensions, including disease prevention and monitoring, diagnosis and treatment, hospital management, health decision-making, and medical research. Information technologies, for example, IoT, mobile Internet, cloud computing, big data, 5G, microelectronics, and artificial intelligence, together with modern biotechnology constitute the cornerstone of smart healthcare. These technologies are widely used in all aspects of smart healthcare. From the perspective of patients, they can use wearable devices to monitor their health at all times, seek medical assistance through virtual assistants, and use remote homes to implement remote services; from the perspective of doctors, a variety of intelligent clinical decision support systems are used to assist and improve diagnosis. Doctors can manage medical information through an integrated information platform that includes Laboratory Information Management System, Picture Archiving and Communication Systems (PACS), Electronic Medical Record, and so on. More precise surgery can be achieved through surgical robots and mixed reality technology. From the perspective of hospitals, radio-frequency identification (RFID) technology can be used to manage personnel materials and the supply chain, using integrated management platforms to collect information and assist decisionmaking. The use of mobile medical platforms can enhance patients' experiences, From the perspective of scientific research institutions, it is possible to use techniques such as machine learning instead of manual drug screening and to find suitable subjects using big data.4 Through the use of these technologies, smart healthcare can effectively reduce the cost and risk of medical procedures, improve the utilization efficiency of medical resources, promote exchanges and cooperation in different regions, push the development of telemedicine and self-service medical care, and ultimately make personalized medical services ubiquitous.

III. PAGE STYLE TESTING AND IMPLEMENTATION

The term implementation has different meanings ranging from the conversation of a basic application to a complete replacement of a computer system. The procedures however, are virtually the same. Implementation includes all those activities that take place to convert from old system to new. The new system may be totally new replacing an existing manual or automated system or it may be major modification to an existing system. The method of implementation and time scale to be adopted is found out initially. Proper implementation is essential to provide a reliable system to meet organization requirement.

3.1 Unit Testing

In computer programming, unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application. In procedural programming, a unit could be an entire module, but it is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method. Unit tests are short code fragments created by programmers or occasionally by white box testers during the development process. It forms the basis for component testing. Ideally, each test case is independent from the others. Substitutes such as method stubs, mock objects, fakes, and test harnesses can be used to assist testing a module in isolation. Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 8, April 2023

Benefits

The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. A unit test provides a strict, written contract that the piece of code must satisfy. As a result, it affords several benefits.

1) Find problems early: Unit testing finds problems early in the development cycle. In test-driven development (TDD), which is frequently used in both extreme programming and scrum, unit tests are created before the code itself is written. When the tests pass, that code is considered complete. The same unit tests are run against that function frequently as the larger code base is developed either as the code is changed or via an automated process with the build. If the unit tests fail, it is considered to be a bug either in the changed code or the tests themselves. The unit tests then allow the location of the fault or failure to be easily traced. Since the unit tests alert the development team of the problem before handing the code off to testers or clients, it is still early in the development process.

2) Facilitates Change: Unit testing allows the programmer to refactor code or upgrade system libraries at a later date, and make sure the module still works correctly (e.g., in regression testing). The procedure is to write test cases for all functions and methods so that whenever a change causes a fault, it can be quickly identified. Unit tests detect changes which may break a design contract.

3) Simplifies Integration: Unit testing may reduce uncertainty in the units themselves and can be used in a bottom-up testing style approach. By testing the parts of a program first and then testing the sum of its parts, integration testing becomes much easier.

4) Documentation: Unit testing provides a sort of living documentation of the system. Developers looking to learn what functionality is provided by a unit, and how to use it, can look at the unit tests to gain a basic understanding of the unit's interface (API). Unit test cases embody characteristics that are critical to the success of the unit. These characteristics can indicate appropriate/inappropriate use of a unit as well as negative behaviours that are to be trapped by the unit. A unit test case, in and of itself, documents these critical characteristics, although many software development environments do not rely solely upon code to document the product in development

3.2 Integration Testing

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

Purpose

The purpose of integration testing is to verify functional, performance, and reliability requirements placed on major design items. These "design items", i.e., assemblages (or groups of units), are exercised through their interfaces using black-box testing, success and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and inter-process communication is tested and individual subsystems are exercised through their input interface. Test cases are constructed to test whether all the components within assemblages interact correctly, for example across procedure calls or process activations, and this is done after testing individual modules, i.e., unit testing. The overall idea is a "building block" approach, in which verified assemblages are added to a verified base which is then used to support the integration testing of further assemblages. Software integration testing is performed according to the software development life cycle (SDLC) after module and functional tests. The crossdependencies for software integration testing are: schedule for integration testing, strategy and selection of the tools used for integration, define the cyclomatic complexity of the software and software architecture, reusability of modules and life-cycle and versioning management. Some different types of integration testing are big-bang, top-down, and bottom-up, mixed (sandwich) and risky-hardest. Other Integration Patterns are: collaboration integration, backbone integration, layer integration, client-server integration, distributed services integration and high-frequency integration.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 8, April 2023

Big Bang

In the big-bang approach, most of the developed modules are coupled together to form a complete software system or major part of the system and then used for integration testing. This method is very effective for saving time in the integration testing process. However, if the test cases and their results are not recorded properly, the entire integration process will be more complicated and may prevent the testing team from achieving the goal of integration testing. A type of big-bang integration testing is called "usage model testing" which can be used in both software and hardware integration testing. The basis behind this type of integration testing is to run user-like workloads in integrated user-like environments. In doing the testing in this manner, the environment is proofed, while the individual components are proofed indirectly through their use. Usage Model testing takes an optimistic approach to testing, because it expects to have few problems with the individual components. The strategy relies heavily on the component developers to do the isolated unit testing for their product. The goal of the strategy is to avoid redoing the testing done by the developers, and instead flesh-out problems caused by the interaction of the components in the environment. For integration testing, Usage Model testing can be more efficient and provides better test coverage than traditional focused functional integration testing. To be more efficient and accurate, care must be used in defining the user-like workloads for creating realistic scenarios in exercising the environment. This gives confidence that the integrated environment will work as expected for the target customers.

Top-down And Bottom-up

Bottom-up testing is an approach to integrated testing where the lowest level components are tested first, then used to facilitate the testing of higher-level components. The process is repeated until the component at the top of the hierarchy is tested. All the bottom or low-level modules, procedures or functions are integrated and then tested. After the integration testing of lower-level integrated modules, the next level of modules will be formed and can be used for integration testing. This approach is helpful only when all or most of the modules of the same development level are ready

3.3 Software Verification and Validation

In software project management, software testing, and software engineering, verification and validation (V&V) is the process of checking that a software system meets specifications and that it fulfills its intended purpose. It may also be referred to as software quality control. It is normally the responsibility of software testers as part of the software development lifecycle. Validation checks that the product design satisfies or fits the intended use (high-level checking), i.e., the software meets the user requirements. This is done through dynamic testing and other forms of review. Verification and validation are not the same thing, although they are often confused. Boehm succinctly expressed the difference between

- Validation: Are we building the right product?
- Verification: Are we building the product, right?

According to the Capability Maturity Model:

- Software Verification: The process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. Software verification ensures that "you built it right".
- Software Validation: The process of evaluating software during or at the end of the development process to determine whether it satisfies specified requirements. Software validation ensures that "you built the right thing".

From Testing Perspective

- Fault – wrong or missing function in the code.
- Failure – the manifestation of a fault during execution.
- Malfunction – according to its specification the system does not meet its specified functionality

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 8, April 2023

Both verification and validation are related to the concepts of quality and of software quality assurance. By themselves, verification and validation do not guarantee software quality; planning, traceability, configuration management and other aspects of software engineering are required.

Classification of Methods

In mission-critical software systems, where flawless performance is absolutely necessary, formal methods may be used to ensure the correct operation of a system. However, often for non-missioncritical software systems, formal methods prove to be very costly and an alternative method of software V&V must be sought out. In such cases, syntactic methods are often used.

Test Cases

A test case is a tool used in the process. Test cases may be prepared for software verification and software validation to determine if the product was built according to the requirements of the user. Other methods, such as reviews, may be used early in the life cycle to provide for software validation

3.4 Black-Box Testing

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied virtually to every level of software testing: unit, integration, system and acceptance. It typically comprises most if not all higher-level testing, but can also dominate unit testing as well.

Test Procedures

Specific knowledge of the application's code/internal structure and programming knowledge in general is not required. The tester is aware of what the software is supposed to do but is not aware of how it does it. For instance, the tester is aware that a particular input returns a certain, invariable output but is not aware of how the software produces the output in the first place.

Test Cases

Test cases are built around specifications and requirements, i.e., what the application is supposed to do. Test cases are generally derived from external descriptions of the software, including specifications, requirements and design parameters. Although the tests used are primarily functional in nature, non-functional tests may also be used. The test designer selects both valid and invalid inputs and determines the correct output, often with the help of an oracle or a previous result that is known to be good, without any knowledge of the test object's internal structure

3.5 White-Box Testing

White-box testing (also known as clear box testing, glass box testing, transparent box testing, and structural testing) is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality (i.e., black-box testing). In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. This is analogous to testing nodes in a circuit, e.g., in-circuit testing (ICT). White-box testing can be applied at the unit, integration and system levels of the software testing process. Although traditional testers tended to think of white-box testing as being done at the unit level, it is used for integration and system testing more frequently today. It can test paths within a unit, paths between units during integration, and between subsystems during a system–level test. Though this method of test design can uncover many errors or problems, it has the potential to miss unimplemented parts of the specification or missing requirements.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 8, April 2023

Levels

Unit testing:

White-box testing is done during unit testing to ensure that the code is working as intended, before any integration happens with previously tested code. White-box testing during unit testing catches any defects early on and aids in any defects that happen later on after the code is integrated with the rest of the application and therefore prevents any type of errors later on.

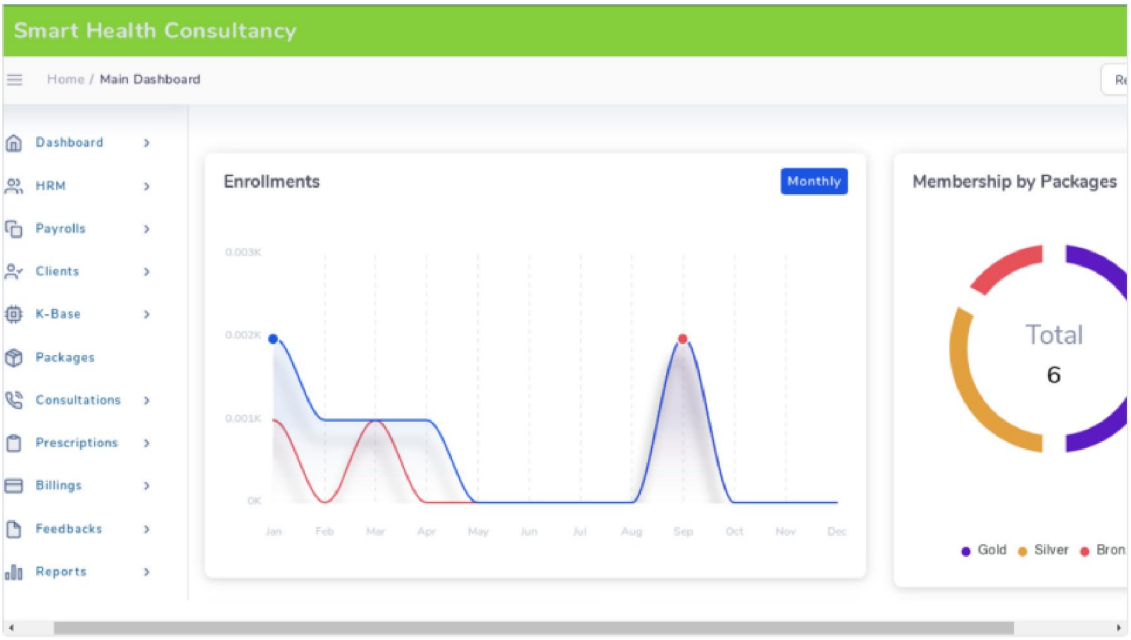
Integration testing:

White-box testing at this level is written to test the interactions of each interface with each other. The Unit level testing made sure that each code was tested and working accordingly in an isolated environment and integration examines the correctness of the behaviour in an open environment through the use of white-box testing for any interactions of interfaces that are known to the programmer.

Regression testing:

White-box testing during regression testing is the use of recycled white-box test cases at the unit and integration testing levels

IV. IMPLEMENTATION



4.1 Functional Requirements

4.1.1 Log In

User story:

As a registered user, I am required to log in so that I can access the system.
As a forgetful user, I can request a password reminder so that I can log in if I forget mine.
Confirmation:

Success: Valid user can log in and refer to home page.
a. "Save password" - ticked - store cookie/ automatic log in next time.
b. "Save password" - not ticked - force log in next time.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 8, April 2023

Failure:
Display message:
a."Email Address is incorrect."
b."Incorrect user name. Please try again."
c."Incorrect Password.Please try again."

4.1.2 Sign Up
User story:
As a new user, I want to register in new account so that I can access the system.
Confirmation:

Success:
New user can sign in and create new account.

Failure:
a. Invalid email address.
b. Mismatched confirmation password.

4.1.3 Home
User story:
As a registered user, I want to visit the main page of the system so that I can get an overall idea and go through the other features.
Confirmation:

Success:
a. Logged in successfully.
b. Can access all the features.

Failure:
Unsuccessful to logged in.

4.1.4 Search Doctor
User story:
As a registered user, I am required to search a specific doctor according to the diseases so that I can get proper treatment.
As a registered user, I want to search specific doctor so that I can get the information about doctors and their schedule.
Confirmation:

Success:
a. View the information of doctors.
b. Can fixed appointment.

Failure:
a. Typing error.
b. Can not make an appointment.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 8, April 2023

4.1.5 Diseases Prediction

User story:

As a registered user, I want to search about my disease so that I can get confirmation about the symptoms and disease.

As a registered user,I need to get information about the related doctors so that I can fix an appointment with them.

Confirmation:

Success:

a. Able to find the disease's symptoms.

b. View related doctors.

c. Can make appointment.

Failure:

a. Typing error.

b. Unable to find any result.

4.1.6 View Appointment

User story:

As a registered user, I want to see the list of patients so that if I would like to see the records of those old patients who have been treated before.

As a registered user, I want to chat online and make video conference so that I can interact with doctors in real time.

Confirmation:

Success:

Able to view all previous record of old patients.

Failure:

a. Page not found.

b. No database connection.

4.1.7 Pharmacy Area

User story:

As a registered user, I want to order medicine so that I can get the medicine in emergency.

Confirmation:

Success:

a. Order medicine in emergency.

b. Save time.

Failure:

a. Shortage of medicine.

b. Unavailable.

c. Delay delivery of medicine.

4.1.8 Bill/ Transaction

User story:

As a registered user, I want to transfer money so that I can pay the bills.

Confirmation:

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 8, April 2023

Success:
a. Can pay bills through online.
b. Save time.
c. Secured transaction process.

Failure:
a. Page not found.
b. No database connection.

4.1.9 Feedback

User story:
As a registered user, I want to give some feedback so that other users can get ideas and benefit from this system.
Confirmation:
Success:
Assume ranking of system.

Failure:
a. Page not found.

4.1.10 Contact

User story:
As a registered user, I want to view contact area so that I can view admins' information and can give message directly to Admin about anything.

Confirmation:
Success:
View admins' details.

Failure:
a. Wrong phone number.
b. Invalid email address.

4.1.11 Log Out

User story: As a registered user, I want to log out myself from the system so that I can exit the system.
Confirmation:

Success:
a. "Exit system"- ticked- store cookies/ automatic log out.
b. "Exit system"- not ticked- stay logged in.

Failure:
a. Poor server service.

4.2 Non-Functional Requirements

Non-functional requirements do not affect the basic functionality of the system. It is basically the overall attributes of the resulting system. User visible aspects of the system not directly relates to the functional behaviour include quality and constraints. There are a lot of software requirements specifications included in the non-functional requirements of the Smart Health Consulting System which contains performance, safety, security, quality maintainability, availability reliability and usability.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 8, April 2023

4.2.1 Performance

- Response Time: The system provides response in just one second once the patient's information checked.
- Capacity: The system needs to maintenance at least 1000 people at once.
- User-Interface: The user interface accepts within five seconds. Conformity: The system needs to make sure that the rules of Microsoft conveniences are followed.

4.2.2 Security Patient Identification:

The system needs to identify himself/herself.

- Log In ID: User need to hold a Log in ID and password.
- Modifications: Some changes like insert, delete, update for the database can be matched quickly and executed by admin.
- Administrator Rights: The admin can view as well as change any information in this system. If the system fails, this system will be able to recover very quickly.

4.2.3 Maintainability

- Back-Up: The system offers the effectiveness for data backup.
- Error: The system will track every mistake.

4.2.4 Reliability

Availability: The system is accessible all the time.

Other than all of those, we have to take care of the following aspects:

- Resource management
- Documentation
- Reliability
- Security
- Quality
- Safety
- Verification
- Recover ability
- Maintainability

REFERENCES

[1]. Chen, M., Hao, Y., & Lin, Y. (2020). Personalized health consulting using machine learning and internet of things: a systematic review. Journal of medical systems, 44(3), 1-15.

[2]. Li, C., Chen, Y., Li, M., Huang, Y., Li, H., & Li, Y. (2021). A personalized health consulting system based on data mining and machine learning. Journal of medical systems, 45(3), 1-12.

[3]. Jiang, J., & Song, J. (2018). A web-based personalized health consulting system based on deep learning. International Journal of Information Management, 39, 80-87.

[4]. Zhang, Y., Wang, J., Li, Y., & Zhang, B. (2019). A web-based personalized health consulting system using big data analytics and machine learning. Journal of medical systems, 43(11), 1-10.

[5]. Cui, Y., Li, Y., & Chen, S. (2020). Design of a smart health consulting system based on deep learning and internet of things. Future Generation Computer Systems, 105, 27-37.

[6]. Luo, G., & Stone, B. L. (2019). A web-based personalized health consulting system for chronic disease management. Journal of medical systems, 43(2), 1-10.

[7]. Hu, X., Wu, H., Wu, J., Zhu, Y., & Lu, Y. (2020). Design and implementation of a personalized health consulting system based on big data and cloud computing. Journal of medical systems, 44(3), 1-10.

[8]. Kim, M., & Kim, H. J. (2019). Personalized health consulting system based on big data and machine learning. Healthcare informatics research, 25(3), 177-185.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 8, April 2023

[9]. Wang, S., & Zeng, X. (2019). Design and implementation of a web-based personalized health consulting system. Journal of medical systems, 43(8), 1-11.

[10]. Wu, X., Wu, L., & Zhu, X. (2020). A personalized health consulting system based on machine learning and big data analytics. Journal of medical systems, 44(11), 1-11.

BIOGRAPHY

Hello readers, we are undergraduate students of computer science engineering. We have deep interest in smart health care services and are keen towards researching in this field.