Georgian Technical University

Mohammad Saeid Tavana

Internet of Thing (IoT) in medical informatics and telemedicine with approach to design patient Electronic Health Records (EHRs) and patient alarm system

Submitted for academic degree of master of engineering

Master Program – Biomedical Engineering (English)

Code:

Georgian Technical University

Tbilisi, 0175, Georgia

JULY 2018

Georgian Technical University

Faculty of Informatics and Control Systems

We, the undersigned, certify, that we have examined the Master's Thesis: "Internet of Thing (IoT) in medical informatics and telemedicine with approach to design patient Electronic Health Records (EHRs) and patient alarm system", performed by Mohammad Saeid Tavana recommend it to the Examination Commission of Informatics and Control Systems Faculty" of Georgian Technical University to consider it for awarding the Master Degree in Biomedical Engineering.

Date: ----------

Supervisor: Zviad Gurtskaia
_____

Reviewer: _____ Mamonti Rogava

Head of department of quality assessment
Faculty of informatics and control system: _____ Zurab Baiashvili

Georgian Technical University
Presented to the Board of Examinations in 2018
Mohammad Saeid Tavana

"Internet of Thing (IoT) in medical informatics and telemedicine with approach to design patient Electronic Health Records (EHRs) and patient alarm system"

Georgian Technical University
Faculty of Informatics and Control Systems

Presented for obtaining the Master Degree in Biomedical Engineering

Meeting held on ----------, 20---

Author's signature _____

Abstract

Nowadays Internet of Thing (IoT) has progress wildly in every project. IoT has efficient stage in medical informatics and telemedicine. this thesis provides brief introduction to IoT and related technologies Which is relate to Medical informatics and telemedicine to build fully automatic electronics health records (EHRs) using IOT.

Software has built with knowledge of IoT and software engineering. Software gets the information from the patient by the sensors or devices which is connected to patient body and send information to processors which called server side of software, it can be super computer or single raspberry pi. By using AI (Artificial intelligence) and image processing, software can decide what decision must be taken and provide patient history to specific doctor or nurse.

Software has multi-level permission for accessing resources. This software automates all actions in hospital, clinics and other agency which provide health services. Software provide variety of service for patient, doctor and nurse such as: patient health history, patient electronic prescription, patient doctor appointment, patient electronic diagnosis, patient examination results and all information related to patient health records. with use of this software, patient does not need to carry on his/her health history to clinics or hospitals, this data can be accessible everywhere which this software installed.

Software has many features will describe in thesis. Automed software which is modern electronics health records (EHRs) which can manage full process of patient admintssion and patient operations and some more other task like patient prescriptions, patient life history, patient disgnosis result and patient examinations. Automes also manage appointment between doctor and patient automaticly just by few click by nurse user and nurse can see list and details of doctor appointment as well. Automed has both hardware and software side, hardware side which refere as alaraming device.

Detect alarm from patient monitor and using IR camera taking picture from patient monitor screen and using image processing technique to discover data from image which has been taken from patient monitor screen and send alarm with patient information directly to doctor and nurse station. With this alarm which occurred by automed medical system doctor can order immediately to nurse station to performe some operation on patient.

Also, automed medical system record all trasnsaction which done by all users in system such as doctor, nurse, patient and alarm device, all login logs which shows who loged in and when loged in with some other login details are store as well for future reports. Automed also is modern and intelligent HERs (Electronic health recors) system that hold all information regarding patient life history and patient diagnosis.

All patient health history and examinations related to patient health are store in automed medical system, paitent health data can be restore everywhere which automed medical system has installed. automed medical software can reduce paper work in hospitals and clinics also with such system, health provider can provide good quality of health services very fast.

Automed medical system use high level of security methods to secure patient health data to prevent from any cyber Attack.

რეზიუმე

დღეს დღეობით ინტერნეტმა (IoT) ძლიერი პროგრესი განიცადა ყველა სფეროში. IoT-მ მნიშვნელოვან საფეხურს მიაღწია სამედიცინო ინფორმატიკასა და ტელემედიცინაში. წინამდებარე თეზისი წარმოადგენს მოკლე შესავალს IoT -ის და იმ ტექნოლოგიების შესახებ რომელთაც დაკავშირებულია სამედიცინო ინფორმატიკასა და ტელემედიცინასთან, რათა შექმნას სრულყოფილი ავტომატური ელექტრონული სამედიცინო ჩანაწერი (EHR) IoT-პროგრამის გამოყენებით.

პროგრამული უზრუნველყოფა შეიქმნა IoT-პროგრამისა და პროგრამული ინჟინერიის გამოყენებით. პროგრამა პაციენტის შესახებ ინფორმაციას იღებს სენსორების ან მოწყობილობების საშუალებით რომელთაც შეერთებულია პაციენტის სხეულთან და უგზავნის ინფორმაციას პროცესორებს რომელსაც ეწოდება პროგრამის სერვერული მხარე. ის შეიძლება იყოს სუპერ კომპიუტერული ან ერთიული Raspberri pi. ხელოვნური ინტელექტისა და გამოსახულების დამუშავების საშუალებით, პროგრამას შეუძლია მიიღოს გადაწყვეტილება მიაწოდოს თუ არა პაციენტის ისტორია შესაბამის ექიმსა და ექთანს. პროგრამას აქვს მრავალფუნქციანი ნებართვა წყაროებზე წვდომისათვის. ეს პროგარმა აღდენს ყველა მოქმედების ავტომატიზებას ჰოსპიტალში, კლინიკებში და სხვა ასეთი ტიპის ჯანდაცვის ორგანიზაციებში რომელთაც უზრუნველყოფენ ჯანდაცვითმომსახურებას.

პროგრამა უზრუნველყოფს მრავალფეროვან მომსახურებას, როგორიცაა: პაციენტის ჯანმრთელობის ისტორია, ელექტრონული დანიშნულების გამოწერა, ექიმის ვიზიტის დანიშვნა, პაციენტის ელექტრონული დიაგნოზის დასმა, პაციენტის გამოკვლევების შედეგები და ყველა ინფორმაცია რომელთც დაკავშირებულია პაციენტის ჯანმრთელობის ისტორიასთან. ამ პროგრამის გამოყენებით პაციენტს არსჭირდება ხელახლა მოყვეს თავისი ავადმყოფობის ისტორია სხვადასხვა კლინიკასა თუ ჰოსპიტალში, მისი მონაცემები ხელმისაწვდომი იქნება ყველგან სადაც არის დაინსტალირებული ელიშნული პროგრამა.

პროგრამას აქვს ბევრი მახასიათებლები რომელსაც ალვწერთ თეზისში. ავტომატიზირებული პროგრამა რომელსაც შეუძლია მართოს პაციენტის მიღების, ოპერაციის სრული პროცესი და სხვა საკითხები როგორიცაა რეცეპტის გამოწერა, პაციენტის ისტორია, დიაგნოზი და პაციენტის გამოკვლევები. პროგრამას ასევე შეუძლია მართოს პაციენტისა და ექიმის შეხვედრა, ავტომატურად რამოდენიმე დღით წკაპუნებით ექთანს შეუძლია ნახოს ექიმის შეხვედრების სია და დეტალები. ავტომატურ პროგრამას აქვს ორივე როგორც აპარატურული ისე პროგრამული უზრუნველყოფა.

აპარატურულთ მხარე - რომელთც მუშაობს როგორც სასიგნალო მოწყობილობა.

სიგნალს ალმოჩენა პაციენტის მონიტორიდან და IR კამერის გამოყენება რომელთც გადაიღებს ფოტოს პაციენტის მონიტორის ეკრანიდან და გაგზავნის სიგნალს პაციენტის ინფორმაციის შესახებ ექიმთან და ექთანთან. ამ სიგნალთთ რომელთც ჩნდება ავტომატური სამედიცინო სისტემის საშუალებითექიმს შეუძლთა პირადპირ მისცეს განკარგულება ექთანს რომ პაციენტს ჩაუტარდეს გარკვეულთ საზის პროცედურა.

ავტომატური სამედიცინო სისტემა იწერს ყველა ტრანზაქციას რომელთც ხორციელდება სისტემის ყველა მომხმარებლთს მიერ როგორიცაა ექიმი, ექთანი, პაციენტი, და სასიგნალო მოწყობილობა, ყველა დარეგისტრირება რომელთც აჩვენებს ვინ და რა დროს შევიდა სისტემაში და ასევე სხვა დეტალები ინახება მომავალთ ანგარიშებისათვის. ავტომატიზირებულთა ასევე თანამედროვე და ჭკვიანი ელექტრონულთ სამედიცინო ჩანაწერის (HER) სისტემა რომელთც ფლობს პაციენტის ისტორიისა და დიაგნოზის სრულ ინფორმაციას.

ყველა ინფორმაცია პაციენტის ჯანმრთელობის და გამოკვლევების შესახებ ინახება ავტომატურ სამედიცინო სისტემაში, პაციენტის ჯანმრთელობის მონაცემები შეიძლება აღდგენილიქნეს ნებისმიერი ადგილას სადაც არის დაინსტალირებულთ ავტომატური სამედიცინო სისტემა. ავტომატური სამედიცინო სისტემას შეუძლთა შეამციროს საკანცელარიო სამუშაო საავადმყოფოებსა და კლინიკებში ასევე მსგავსი სისტემითე სამედიცინო დაწესებულებას შეუძლთა უზრუნველყოს ჯანდაცვის ხარისხიანი და სწრაფი სერვისი.

ავტომატური სამედიცინო სისტემა იყენებს უსაფრთხოების მაღალ ზომებს რათა დაიცვას პაციენტის ჯანმრთელობის მონაცემები ნებისმიერი კიბერშეტევისგან.

Table of Contents

List of Figure

Abbreviations:

JVM - Java Virtual machin
JNI - Java Native Interface
JDK -Java Development Kit
JRE - Java Runtime Interface
PDA - Personal Device Assistant
HERs - Electronic Health Records
CPRs - Computer Patient Records
JSP - Java Server Page
JSF - Java Server Face
ASP - Active Server Pages
SSI - Server-Side Includes
SSL - Secure Socket Layer
QR Code - Quick Response Code
MVP - Model View Presenter
SOA- Service Oriented Architecture
DCOM - Distributed Component Object Model
CORBA - Common Object Request Broker Architecture
OMG - Object Management Group
MVC - Model View Controler
IOT - Internet of Things

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my supervisors Professor IrineGotsiridze and Professor Krzysztof Penkala for their continuous support of my Master's thesis.

I am deeply grateful to the head of the Biomedical Engineering Department of Georgian Technical University, Professor Irine Gotsiridze for giving me the opportunity to study at Program on Biomedical Engineering and for the knowledge I acquired, that has provided a good basis for the present thesis and special thanks to the West Pomeranian University of Technology Szczecin, where I spend my IV semester as Erasmus+ Master Student, Faculty of Electrical Engineering, for giving me opportunity to performing my master thesis in their university.

Special thanks to my mother Farahnaz Malak Kohi and my father Mohammad Tavana and to all my friends for every kind of support they provided me also my dear Dr. Mariam Tsiklauri who help me much during writing of this dessirtation. Particularly, very deep thanks for every person who helped me during my master project and support me with materials, equipment and etc.

CHAPTER1
Introduction

1.1 Internet of Things (IoT)

The Internet of things (IoT) is the network of physical devices, vehicles, home appliances and other items embedded with electronics, software, sensors, actuators, and connectivity which enables these objects to connect and exchange data. Each thing is uniquely identifiable through its embedded computing system but is able to inter-operate within the existing Internet infrastructure.

The figure of online capable devices increased 31% from 2016 to 8.4 billion in 2017. Experts estimate that the IoT will consist of about 30 billion objects by 2020. It is also estimated that the global market value of IoT will reach $7.1 trillion by 2020.

The IoT allows objects to be sensed or controlled remotely across existing network infrastructure, creating opportunities for more direct integration of the physical world into computer-based systems, and resulting in improved efficiency, accuracy and economic benefit in addition to reduced human intervention. When IoT is augmented with sensors and actuators, the technology becomes an instance of the more general class of cyber-physical systems, which also encompasses technologies such as smart grids, virtual power plants, smart homes, intelligent transportation and smart cities.

"Things", in the IoT sense, can refer to a wide variety of devices such as heart monitoring implants, biochip transponders on farm animals, cameras streaming live feeds of wild animals in coastal waters, automobiles with built-in sensors, DNA analysis devices for environmental/food/pathogen monitoring, or field operation devices that assist firefighters in search and rescue operations. Legal scholars suggest regarding "things" as an "inextricable mixture of hardware, software, data and service.

These devices collect useful data with the help of various existing technologies and then autonomously flow the data between other devices. The term "the Internet of things" was coined by Kevin Ashton of Procter & Gamble, later MIT's Auto-ID Center, in 1999.[1]

1.2 Why the IoT Is Strategically Sound

Information technology (IT) has been in the forefront in precisely and perfectly automating

and accelerating a variety of business tasks in order to immensely empower businesses, partners, employees, and consumers to accrue the widely circulated and IT-enabled business benefits. IT is being positioned as the best business enabler. IT is constantly evolving to do better and bigger things. These days, IT, apart from being the greatest enabler of simple as well as complicated busi-ness operations, is penetrating powerfully into every tangible industry segment in order to proac-tively ensure newer and nimbler customer-centric business offerings. In short, IT is able to both simplify and amplify business outputs and outlooks significantly. It is absolutely clear that the strategically sound association and alignment between business and IT are on the consistent rise to create and sustain real-time, adaptive, composable, and instant-on enterprises. Having compre-hensively understood the outstanding contributions of IT in keeping up the business expectations in the cut-throat competitive marketplace, business executives and entrepreneurs are striving hard and stretching further to put in more money to conceive, concretize, and deliver next-generation IT-enabled business services and solutions to their worldwide clients and consumers, to devise workable and well-intended mechanisms and methods to understand people's needs, and deliver them with all the quality of service (QoS) and quality of experience (QoE) attributes embedded in through the smart adoption and adaption of all kinds of exquisite advancements in the hot and happening IT field. Decision makers and other stakeholders are constantly looking out for fresh avenues for more revenues.

With businesses achieving the desired success on various fronts, there is a tectonic shift in IT Being intrinsically leveraged for empowering people in their daily activities. That is, the movement toward consumer-centric IT is on the fast track with the emergence of innovative, transformative, and disruptive technologies.[1]

1.3 IoT leads to smart computing

How human life will be on this planet in and around the year 2025? What kind of lasting impacts, cultural changes, and perceptible shifts will be achieved in the human society due to the constant and consistent innovations, evolutions, and inventions in information, communication, sensing, vision, perception, knowledge, engineering, dissemination, and actuation technologies? Today this has become a dominating and lingering question among leading researchers, luminaries, and scientists. Many vouch for a complete and comprehensive turnaround in our social, personal, and professional lives due to a dazzling array of technological sophistications, creativities, and novel-ties. Presumably computing, communication, perception, and actuation will be everywhere all the time. It is also presumed and proclaimed that the ensuing era will be fully knowledge backed. It is going to be a knowledge-driven society. Databases will pave the way for knowledge bases, and there will be specialized engines for producing and persisting with self-managing systems. Knowledge systems and networks will be used for autonomic communication. Cognition-enabled machines and expert systems will become our casual and compact companions. A growing array of smarter systems will surround,

support, and sustain us in our classrooms, homes, offices, motels, coffee houses, airport lounges, gyms, and other vital junctions, eating joints, and meeting points in big numbers. They will seamlessly connect, collaborate, corroborate, and correlate to understand our mental, social, and physical needs and deliver them in a highly unobtrusive, secure, and relaxed fashion. That is, the right information and rightful services will be conceived, constructed, and delivered to the right person, at the right time, and at the right place. Extensively physical artifacts and assets, mechanical and electrical machines, instruments, equipment, utensils, and wares, electronic devices, communication gateways, and IT systems will become the major contributors for this tectonic and tranquil modernization and migration. Every common, casual, and cheap thing will join the mainstream computing toward the world of cognizant, connected, and cognitive computing.[1]

1.4 IoT Delivers Smarter Environments

Our living, relaxing, and working environments are envisioned to be filled up with a variety of electronics including environment monitoring sensors, actuators, monitors, controllers, proces-sors, tags, labels, stickers, dots, motes, stickers, projectors, displays, cameras, computers, com-municators, appliances, robots, gateways, and high-definition IP TVs. Apart from these, all the physical and concrete items, articles, furniture, and packages will become empowered with computation and communication-enabled components by attaching specially made electronics onto them. Whenever we walk into such kinds of empowered and augmented environments lightened up with a legion of digitized objects, the devices we carry and even our e-clothes will enter into a calm yet logical collaboration mode and form wireless ad hoc networks with the inhabitants in that environment. For example, if someone wants to print a document in his or her smartphone or tablet, and if he or she enters into a room where a printer is situated, the smartphone will begin a conversation with the printer automatically and send the document to be printed.Thus, in that era, our everyday spots will be made informative, interactive, intuitive, and invigorative by embedding and imbedding intelligence into their constituents (audio or video systems, cameras, information and web appliances, consumer and household electronics, and other electronic gadgets besides digitally augmented walls, floors, windows, doors, ceilings, and any other physical objects and artifacts). The disappearing computers, communicators, sen-sors, and robots will be instructing, instigating, alerting, and facilitating decision making in a smart way, apart from accomplishing all kinds of everyday needs proactively for human beings. Humanized robots will be extensively used in order to fulfill our daily physical chores. That is, computers in different sizes, looks, capabilities, interfaces, and prizes will be fitted, glued, implanted, and inserted everywhere to be coordinative, calculative, and coherent, yet invisible for discerning human minds. In summary, the IoT technologies in sync up with cloud infra-structures are to result in people-centric smarter environments. Context awareness is the key motivator for business and IT systems to be distinct in their operations, offerings, and

outputs. The days of ambient intelligence (AmI) are not far away as the speed and sagacity with which scores of implementation technologies are being unearthed and nourished by product vendors and system integrators. [2]

1.5 The Connected Device

The device space is fast evolving (implantable, wearable, mobile, portable, nomadic, fixed, etc.). The rough and tough passage from the mainframe and the pervasive PC cultures to trendy and handy portables, handhelds and wearables, disappearing implantables, invisible tags, stickers, labels, and chips, and versatile mobiles subtly and succinctly convey the quiet and ubiquitous transition from the centralization to the decentralization mode. This positive and pathbreaking trend, however, brings the difficult and dodging issues of heterogeneity, multiplicity, and incom-patibility. That is, all kinds of participating and contributing devices, machines, instruments, and electronics in our personal as well as professional environments need to be individually as well as collectively intelligent enough to discover one another, link, access, and use to be competent and distinctive to accomplish bigger and better things for humans.The end result is that constructing and managing cross-institutional and functional applica-tions in this sort of dynamic, disparate, decentralized, and distributed environments is laced with a few unpredictable possibilities. That is, there are chances for risky interactions among varied services, sensors, and systems resulting in severe complications and unwanted implications for the safety and security of the human society. Also, it is envisioned that the future spaces will be highly digitized environments with a fabulous collection of digital devices and digitized artifacts; each is distinct in its face, feature, and functionality. Figure 1.1 succinctly illustrates the prevailing IT trends. The IT evolutions and revolutions are categorized as follows. This compendium of devices will be increasingly interlinked to local as well as the global network transparently. With this sophisticated, yet complicated scenario brewing silently and strongly, it is logical to think about the ways and means of ably and autonomically utilizing, managing, and extracting their inherent capabilities (specific as well as generic) and capacities for:

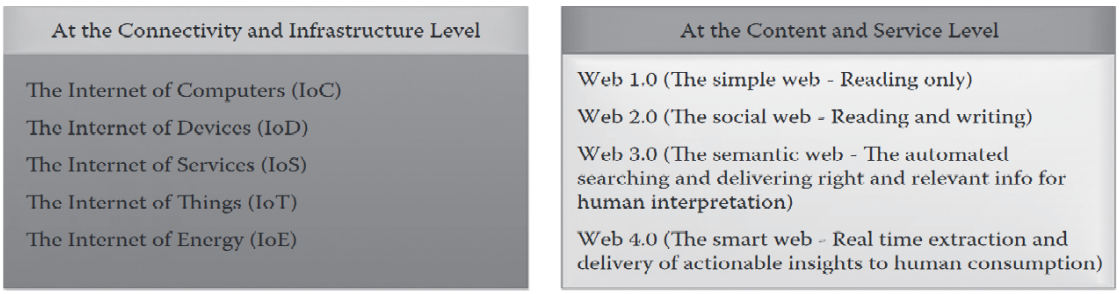


Figure 1.1 The evolution of the Internet paradigm.

arriving at a horde of people-centric, pioneering, and premium services. As we are keenly wait-ing for the paradigm of computing everywhere every timeto cherish and flourish, it is imperative to nourish and nudge any variety of participating devices to be extremely agile and adaptive and to empower them to proactively, preemptively, and purposefully collaborate, correlate, and cor-roborate to figure out the user(s)' contextual needs by dynamically connecting, complementing, and sharing the dynamic resources with one another accordingly and unobtrusively. At the other end, there are a wider variety of input and output devices such as tablets and smartphones to assist people to finish their personal as well as professional assignments effectively and efficiently. That is, devices are becoming device ensembles and clusters through internal as well as external integration

1.6 Smart Energy

Energy has become a scarce commodity, and hence its preservation is very much obligatory. Also, more energy consumption means more heat dissipation into our fragile environment. That is, with efficient usage of precious power energy, the much-feared environmental degradation and global warming can be grossly minimized to achieve environmental sustainability. Smart meteringsolutions (this is an M2M solution connecting every energy-gobbling device in a network with the centralized smart meter) are very much accepted and used in advanced countries in order to accurately understand the usage. In other words, smart electricity meters help energy consumers to decode how energy savings can be achieved based on the readings and alerts being rendered by smart meters. The advanced metering infrastructure (AMI) is an active and ongoing research area to generate solutions for energy efficiency.[1]

1.7 Smart health care

Health care is turning out to be a huge industry in the years to unfold. There are a number of specific devices for measuring and managing a number of health parameters of humans. M2M solutions are capable of reminding the patient and their family members as well as the doctor in case of any emergency arising out of any abnormality in any of the health readings. this is the exact area which this thesis is all about.

1.8 Smart Home Security

Sophisticated home networking, integration, automation, security, and control mechanisms are hitting the market very frequently. M2M solutions for home security are merging with energy management to provide remote alarm controls as well as

remote heating, ventilation, and air con-ditioning (HVAC) controls for homes and businesses through mobile phones.

1.9 Smart Cargo Handling

M2M solutions are being manufactured into a variety of storage or handling containers including cargo containers, money and document bags, and nuclear waste drums. The real-time location of the container, whether it has been opened or closed, and how containers are being handled through motion sensors, can be easily obtained to prevent any possible security and theft risks and to increase recovery capability of stolen or lost material.[1]

1.10 Smart Traffic Management

M2M solutions are able to provide real-time road traffic information to vehicles' drivers viaauto-mobile GPS devices to enable them to contemplate better alternatives.

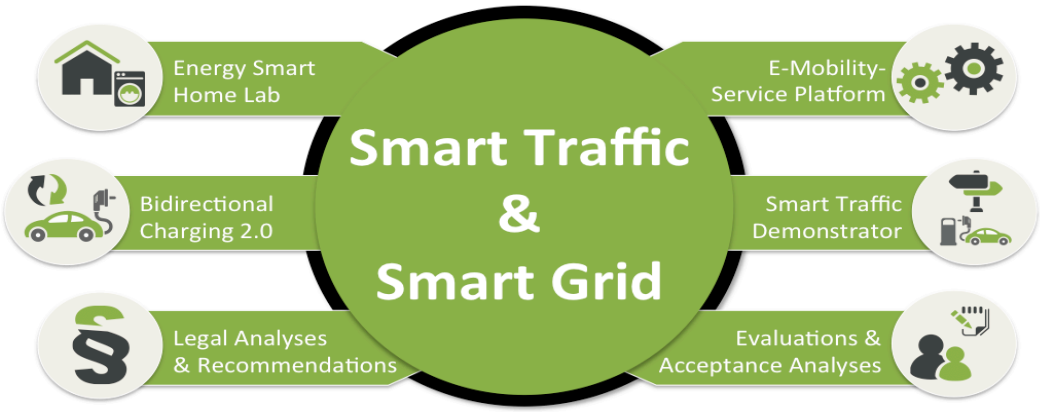


Figure 1.2 Smart trafic managment

1.11 Smart Inventory and Replenishment Management

M2M solutions can be integrated into the sensors measuring the amount of bulk product in a storage bin. This information can be made available to both the supplier and the user, so proactive reorders can be initiated when inventories reach a predetermined level. This is very beneficial for the manufacturing process that does not consume a consistent and predictable amount of product or the transport time of the bulk product results in product run-out.[1] .with use of internet of thing in every day life in any section of works like smart trafic, organization can provide fast and good quality of service to their customer. for instance police can show places with heavy trafic in mobile application instead of announcing in news agancy.

1.12 Smart Cash Payment

M2M solutions allow mobile credit or debit card readers to provide secure and encrypteddata transmissions at the transaction and ticketing counters in hyper malls, hotels, movietheaters, food joints, and so on. Retailing becomes a smooth affair without standing in thequeue for cash payment. The seamless connectivity between tags, tag readers, cash cards, merchant banks, retailers, and so on goes a long way in considerably enhancing the customerexperience.

1.13 Smart Tracking

M2M solutions allow parents to track their children very precisely sitting from the office and empower caregivers to remotely track those with disabilities as well as independently living, disease-stricken, debilitated, and bed-ridden people. Managers can monitor their employees per-forming duties in rough and tough places. Especially those who work in oil wells, fight a forest fire, help out in disaster-struck places, battle in war zones, hike in mountains, and so on are to be immensely benefited through such kinds of technological innovations. The items inside vending machines can connect with their suppliers and provide all the relevant information about the number of bins and bottles inside and how much more are needed to fill up the vendingmachine. This is definitely a sharp improvement over the current practice.

1.14 Smart Displays

All kinds of machines such as ATMs, vending machines, television sets, security video cameras, sign posts, and dashboards can be intertwined together at strategic locations. With such intimate integration through a competent M2M solution, customized video, as well as static images, can be dispatched to these machines to flash time-sensitive and preferred details and displays. A hungry person could order his pizzas on his mobile phone yet see the pizza details and pictures on the larger screen of any one of these machines or with connected projectors showing the images on a white wall to give a clear vision.[1]

1.15 Smarter Manufacturing

A car is driven back to its home garage for the night, and its data port is plugged in. Then, some exciting things start to happen. First off, the car sends diagnostic information back to the manu-facturer to cross check against any system that requires repair, maintenance, or replacement. The manufacturer then downloads a selection of new driver experiences, including a different accel-eration style (choice of sporty or smooth), improved navigation and mapping software, and new stay-in-lane safety features. From an outside perspective, the IoT offers an unlimited selection of

innovations, ranging from an electric toothbrush that monitors correct brushing style through to tire pressure sen-sors in truck fleets to geolocation sensors attached to livestock. The potential for their use is limitless, and this includes on the factory floor. Connected manufacturing is the IoT. Proactive analytics helps a device identify future needs, such as the case when a part might fail, when it requires service, or whensupplies need to be ordered. When the machine itself can dispatch the appropriate commands to a human or another machine, it ensures smooth, safe, and eco-nomical operation.

1.16 Smart Asset Management

Every industry has its own set of specific assets. For example, hospitals should have a number of scanning machines, diagnostic equipment, health care monitors, robots, and other instruments. That is, there are a variety of devices both small and large. The real challenge lies in their effective location identification in case of any emergency, upkeep, management, monitoring, inventory, security, and so on. There are several unique benefits of an M2M solution in this complicated scenario. An advanced M2M solution sharply reduces the time consumed by employees to pinpoint the assets' exact location, considerably increases their utilization, and provides the ability to share high-value assets between departments and facilities. With every asset in a hospital environment integrated with one another and with the remote web or cloud platforms via the M2M product, remote monitoring, repairing, and management are being lavishly facilitated. Through the connectivity established with cloud-hosted health care applications, every machine could update and upload its data to the centralized and cyber applications, thereby getting a number of activities fully auto-mated by avoiding manual intervention, interpretation, and instruction.Professionals and experts are exploring, experimenting, and expounding an increasing array of value-added business and use cases for a variety of industry segments to keep the momentum on the M2M space intact. There is another trend fast picking up these days with the active participa-tion ofacademicians and industry veterans. Cyber-physical systems (CPS) are the new powerful entitiesgainingmomentum. That is, all kinds of physical systems at the ground level are being empowered with scores of generic as well as specific cyber applications, services, and data. That is, not only connectivity but also software-inspired empowerment is being ticked as the next-generation evolution in the machine space.[1]

1.17 Smarter Retailing

McDonald's has gone for a unique experiment in user engagement. A blend of IoT devices and contextual promotions have allowed the restaurant operator to tailor its mobile application offers and advertising to information such as location, weather, purchase habits, and response to promo-tions. For example, if someone is moving

quickly on a hot summer day, the application, which runs on a Vmob contextual analytics platform, shows an offer for a soda at a nearby drive-through. This has received a rousing welcome from customers. For retailers, using the IoT for marketing and sales comes down to creating meaningful experi-ences in order to increase loyalty and customer engagement. Starbucks, the coffee chain, chose to launch a number of remote beacons in its Seattle establishments. For customers with the Starbucks app, the beacons push notifications on the freshest brews and personalized promotions. The idea is to transition the casual customer to premium blends that are offered at Starbucks.Facilitating customer transactions and rewarding brand interactions are effective ways to strengthen customer loyalty. Home Depot is leveraging IoT to increase customer engagement on the second dimension, providing personalized service and information to their customers to help them in their decision-making process. Their mobile app allows shoppers to locate inventory, compare shops, ask experts about projects, and see how products would look in their homes. Once inside the store, the app can guide them through aisles to find the products. The IoT has exciting propositions for different industry verticals including retailers. Interactive touch screens, contextual advertising, geotargeted promotions, personalized in-store environ-ments, and augmented reality are just the beginning. However, it is important to remember that the value of IoT does not lie in technological advances but instead in improving and creating immersive customer experiences.

CHAPTER2
Introduction

2.1 Telemedicine

2.1.1 Information Technology and Healthcare Professionals

The history of modern telemedicine goes back to the invention of the traditional telephone about a century ago. Medical advice was given by physicians over the telephone. The term telemedicine is very simply a description of supporting medical services through the use of telecommunications. 'Tele' is a prefix for distant, originated from ancient Greek. So, telemedicine literally translates to providing medical services over distance. Telecommunica- tions used in medical applications can be categorized as sending medical information between a pair of transmitters and receivers. The so-called 'medical information' can be as simple as a doctor providing consultation to sophisticated data captured from a human body. In its most primitive form, 'The Radio Doctor' first appeared in the Radio News magazine (circa 1924) and is perhaps the earliest documented case of utilizing telecommunication technology for medical application. Although information technology has been used in healthcare since then, (Moore, 1975) was the first scientific literature formally addressing the application of technology in medicine that appeared.

As information technology advanced over the past decades, a wider range of healthcare services could be supported. Indeed, the types of services that can be supported is so vast that any book which tries to provide comprehensive coverage of all areas will most likely contain thousands of pages in several volumes. This book aims to provide an in- depth coverage on how wireless communications and related technologies are used in medical services, we will also look at the challenges and limitations of current technology associated with healthcare information systems.

We will first begin by taking a look at how simple wireless communication networks function and what a telemedicine system consists of. We look at a number of examples that describe how a primitive system supports healthcare services. In the course of the book, more sophisticated systems will be described in more detail.

The context of this chapter is anticipated to give readers an overview on how informa- tion technology is widely used in assisting healthcare without going into technical depth. To begin our discussion, we revisit the term 'information technology', something often associ- ated with computer science. Essentially, it is extensively interpreted as a blend of computing and telecommunications. This leads to the acronym ICT, which stands for Information. [15]

2.1.2 Telemedicine Technologies

Communications Technology, also known as infocomm for short. All these are merely de- scriptions of the use of technology to securely and reliably transmit information between two or more entities. Information Technology (IT) is widely used in many areas that influence our daily life. For example, banking, transportation, manufacturing, etc. This list is seemingly endless. When we see so many information technologies support so many things that we use on a daily basis, it will not be difficult to understand how widely it can be used to support healthcare and medical applications.

Since the information technology and 'dot-com bubble burst' in 2000, the whole IT industry has never quite picked up. Looking at NASDAQ that peaked at its all time high of 5132 in March 2000, then retreated to about a quarter of its peak some nine years later, it does appear that people related to the IT industry have suffered substantial thwack for many years. IT professionals who developed a career in finance enjoyed a few more years of wealth until the subprime mortgage fallout that started in early 2007. So, despite various aspects of IT being widely used in different aspects of daily life, it has a close relationship with the global economic cycle. In contrast, healthcare and medical service is one of the few domains that have consistent high demand very simply because every single one of us understands the fundamental importance of our own well-being. We know as a matter of fact that without quality health nothing will be important to us. For this simple reason, healthcare naturally becomes an essential part of daily life that will continue to be in high demand for many years in the future. [15]

Having realized the prime importance of healthcare, we go further into how IT is applied to healthcare and medical services. Long before the evolution of information technology, herbal medicine practitioner's millennia ago already utilized the most primitive form of information exchange mechanism, namely communication systems to convey messages on medical ser- vices. (Wang, 1999) documented a case where Shen Nong made use of information exchange for treatment of respiratory syndrome as far back as 2735 BC, this may not have been the first case, but it is certain that medicine and communications have been linked together for over 4,000 years. As IT becomes more sophisticated over time, a more diverse range of medical services can be supported. To name a few, IT in medicine involves drug prescription, spread of pandemic modelling, patient monitoring, remote operation, medical database and so on. This is by no means an exhaustive list and we will cover these as well as many

Obviously, healthcare professionals can make use of IT advancements in different ar- eas. Advantages brought by IT include improvement in reliability, efficiency, precision, ease of information retrieval, accomplishing tasks remotely, and better organization. Healthcare therefore becomes more readily accessible and more

efficient. We will look at how technology benefits healthcare professionals, with the assumption that readers have very little prior IT knowledge and know virtually nothing about the underlying technologies.

2.1.3 Providing Healthcare to Patients

In addition to facilitating medical practitioners to perform their tasks, another important issue to address is the healthcare services provided to patients, as they are the end users who must feel comfortable receiving the treatment given. The provision of a technically feasible solution is not the only obstacle to deal with. Other important issues including patients' acceptance and accessibility must also be addressed. We strive to look at providing healthcare solutions to patients using IT from the perspectives of both providers and patients. End users, particularly children and the elderly, may not be too keen on accepting technology as a tool for healing. Convincing patients of the benefits of IT in healthcare may involve liability, security and privacy issues. For example, in the case of monitoring or tracking a patient recovering at home, the patient must be assured that personal information is securely kept and no such information is accessed in any way without consent. Before leaving the topic on providing care to the elderly for now, it is worth briefly noting the advantages brought to this group of users by telemedicine technology. As population ageing is becoming a more significant concern in many countries, it can be widely expected that more care and monitoring will be needed. A significant increase in the application of wireless communications in elderly care has been seen over the past few years as related technologies become more mature. The cost of service becomes more affordable and portable devices become smaller and more user-friendly. As pervasive computing technology advances, more comprehensive and automated services will become available to the ageing population in the years to come (Stanford, 2002). The design of interconnected devices and sensors on the patients' side must ensure non-obtrusiveness and can be comfortably worn. Also, user's movement will not be restricted and reliability will not be affected irrespective of wearing condition. User-friendliness is another important design factor, as absolutely minimal training should be necessary especially for children and the elderly. These should be genuine 'plug- and-play' devices. In this sense, the healthcare system in the patient's home can be installed by a technician during initial deployment. Thereafter, almost everything should be fully automatic except for unavoidable scheduled maintenance such as battery replacement and calibration. [21]

Let's elaborate more on a patient's point of view as an end user. The primary objective of telemedicine is to provide medical services remotely. Amongst numerous advantages brought to patients by telemedicine, an obvious convenience is reducing the need for clinical visits. Through utilization of IT, a patient can rest at home while receiving full medical attention. Reviewing the level of medical support provided over the past two to three decades, IT has certainly provided tremendous benefits to the general public as a whole. The advancement of faster computers and more efficient

bandwidth usage has allowed more types of services to be extended to more users. For example, a few decades ago a simple request for medical advice could be obtained by finding a fixed line telephone and dialing in to the clinic where a physician was stationed. With the availability of mobile Voice over Internet Protocol (VoIP) technology, one can now simply pick up a mobile phone and place a video-enabled call to a physician; the physician does not necessarily have to be situated inside the clinic in order to provide advice. This is just one amongst numerous examples where IT advancements have made healthcare more readily available. More examples will be presented throughout the book. [21]

While the benefits to patients are obvious, there is a wide range of challenges that different parties face in order to serve the patients. These concern people from developers, practitioners, healthcare management and authorities. The subsequent paragraph will highlight challenges that different people face starting from initial planning stage to final rollout and continuing maintenance. From the IT perspective, the fundamental question is feasibility. Primary consideration is whether current technology is capable of doing something. After this comes practicality and cost effectiveness. We begin by considering an example where school children are to enroll into a program that ensures their school bags are ergonomically prepared to minimize issues

2.1.4 Telemedicine Technologies

with back pain. The advantage to participating children is very obvious because the program should reduce their chances of suffering from back pain. However, how viable is the entire program? We need to understand more about the technology involved in order to answer this seemingly simple question.

In this case study, we have the following parties involved: engineers developing the moni- toring system, clinical staff analyzing the captured data, funding bodies providing necessary resources, children participating in the study, and finally, participants' parents giving consent to their children's involvement. We shall look at the case with respect to benefits and concerns from each party's standpoint. [18]

2.1.5 Technical Perspective

Biomedical engineers need to develop a system based on requirements specified by clinical staff, such as that illustrated in Figure 1.1, with the necessary sensors and data communication network. This simple system has a number of sensors forming a sensor network for capturing different types of information about a patient. It is linked to the system for analysis by a workstation and storage in electronic patient record (EPR), and is monitored by necessary system and network administration tools. In this discussion, we shall not go into the technical details whilst giving an insight into what is involved. Engineers analyze this by evaluating technical feasibility and practicality. Digging deeper into technical challenges, one obvious issue to address is

how to ensure whatever captured data is meaningful. There are several factors that influence the validity of data, most notably from what the sensors have picked up followed by what has been transmitted and subsequently received. In this respect, the sensors must be securely attached at the relevant points of the participant's body, and each sensor

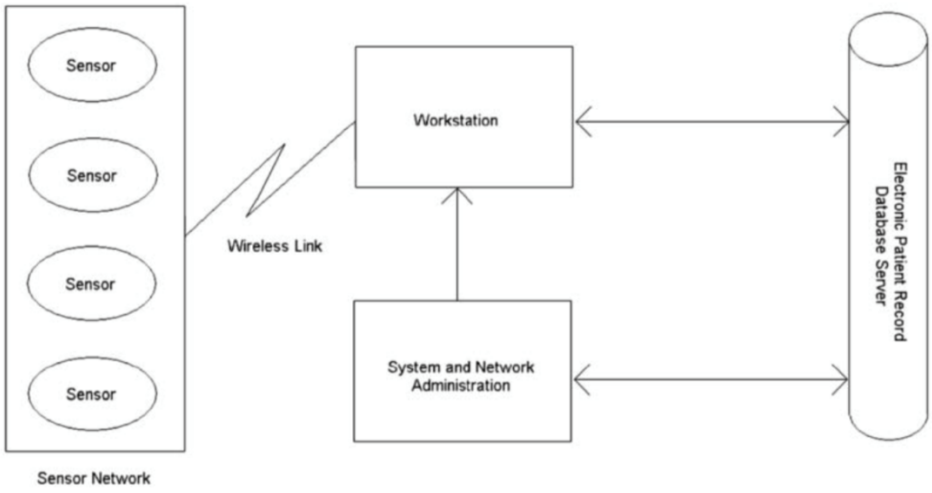


Figure 2.1 A simple biosensor network

must be sensitive enough to pick up any subtle tilting of the body while not too sensitive to pick up any vibration from other sources. Having dealt with these problems, next we must ask whether the sensors are suitable for the specific application, the size may be too large for attaching onto a child, and whether it will cause any discomfort. Are readings affected by any physical obstacles that may be separating the children from the backpack, such as clothing? How is captured data sent out for processing and analysis? Will sensors interfere with each other if placed too close together? Here is just a list of questions related to sensors that need to be dealt with.

We shall proceed by assuming that sensors are well-designed and we manage to overcome all problems listed above. So, we are technically able to capture a set of valid data that tell us something about a child's behaviour when carrying a backpack. We now look briefly at how telemedicine is utilized in a biosensor network; we will come back to this with more details in Section 3.5. In the previous paragraph, we raised a question about how the captured data is sent out, essentially, we have two choices, namely using wireless communications or connecting the sensors with wires. How they compare will depend on the system itself as there is no clear advantage with either option. This is one major topic that we will cover throughout the book.

Briefly summarizing the discussion here, we have seen how many questions need to be addressed in relation to the deployment of such a supposedly simple health monitoring system. So, although the system may appear simple enough to patients,

design and implementation may not be as straightforward and there are so many limitations. [13]

2.1.6 Healthcare Providers

Healthcare professionals should understand that technology is available for making their routine work easier and safer. Many may still prefer traditional practicing methods, just like many people still prefer jotting down notes using pen and paper. Others may find technologies helpful when using a personal digital assistant (PDA) device for the same purpose. There are, of course, many advantages with a PDA although users may need to familiarize themselves with its user interface. Another concern to some is the risk of losing its stored data due to breakdown. We can see that people who are so used to conventional ways of carrying out a task may need to be convinced of the associated benefits technologies bring, in order to impel them into learning to utilize technologies. So, as a practitioner, a simple-to-use interface would be a fundamental design requirement. The entire process should be as automated as possible while maintaining a very high level of reliability. Different applications may have very different demands. For example, tele-surgery requires ultra-high precision for control and crystal-clear imaging details with no time delay, whereas tele-consultation may have much less stringent requirements.

Although technical advancements may be more efficient and fault-free enabling numerous tasks to be accomplished quickly and reliably, the incentive of using IT solutions may not be that compelling unless practitioners have mastered the operations of what is made available to them. Getting used to something new, especially for critical tasks, can be a major challenge. A uniform change to new technology for all applications would be vitally important for a swift switch to making good use of available technology. [25]

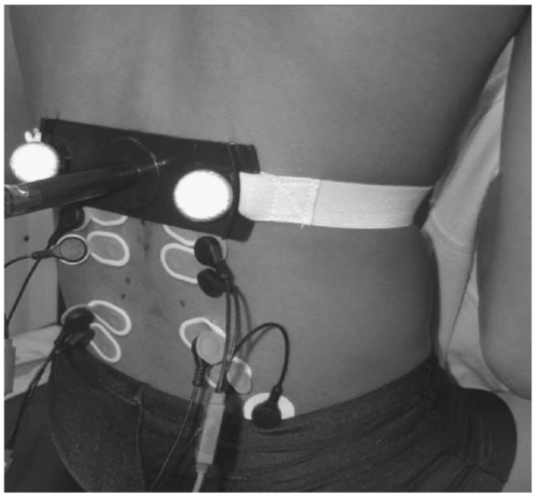


Figure 2.2 Biosensors attached to the back of a patient

2.1.7 End Users

The end users of the system are the patients. The term 'patient' refers to someone who receives medical treatment or service, which includes routine check-up. We should clarify at this point that by definition a person described as a patient may not necessarily be unwell. A perfectly healthy person can be referred to as a patient in this regard. Here, in our case study we have a group of patients who participate in the study of schoolbags on children. They help with the study by having a set of sensors attached to their back while carrying a schoolbag of varying weights. An illustration on how the sensors are attached to the back of a patient is shown in Figure 2.2. We discuss the case study from a patient's point of view by first looking at Figure 2.2. As shown, a number of sensors are attached to the back; each sensor is connected to a data capturing device by a wire. Movement is somewhat affected by the wires so we can readily see the advantages of using wireless sensors as far as the patient is concerned. So, why not wireless? This example exhibits three major technical challenges that make wires extremely difficult to eliminate. First, sensors attached to a child's back must be very small. Powering the sensors can be an issue as installing an internal battery may be a problem. Also, wave propagation issues effectively rule out its use between the body and the bag, as absorption would be a very significant issue. Finally, measurement accuracy given the physical separation of individual sensors and the amount of movement would make the use of wireless solution impractical. For all these reasons, patients have to bear with the wires surrounding them while participating in the experiment.

2.2 Medical informatics

Health informatics is a wide-ranging science incorporating the complex mixture of people,
organizations, illnesses, patient care and treatment. It is a scientific field that deals with the storage, retrieval, sharing, and optimal use of biomedical information, data, and knowledge for problem solving and decision making. The field touches on all basic and applied fields in biomedical science and is closely tied to modern information technologies, notably in the areas of computing and communication. Health informatics looks into ways to optimize clinical knowledge creation, sharing and application to deliver better healthcare and to promote health.

The emergence of medical informatics as a new discipline is due in large part to the rapid advances in computing and communications technologies, an increasing awareness that the knowledge base of biomedicine is essentially unmanageable by traditional paper-based methods, and a growing conviction that the process of informed decision making is as important to modern biomedicine as is the collection of facts on which clinical decisions or research plans are made. A term is currently used is Big Data. The term describes large and exponential growth and availability of data. These data could be structured or unstructured data. This is well defined as data that adhere to the following four articulated criteria. The first criteria is volume, which

is considered to be an ever-increasing amount. With the emergence of various storage devices and the reduction of the storage cost it is made increasingly possible to manage this large volume of data. However, strict evaluation of the large volume of data is very relevant by the analytics. The second criteria is velocity,
as the data is generated in an exceptionally high speed and therefore needs to be managed in a timely manner to be retrieve good analysis. Thirdly, big data have a huge variety as various different types of data are collected e.g. data from laboratory, pharmacy, radiology, financial transaction/billing etc. This will also increase the amount of unstructured data. The fourth
criterion is veracity of the data. The big data are to be gathered from trustable sources well recognized under the healthcare system. There is an information hierarchy that is important in the information sciences, as depicted in the pyramid in Figure 2.3. Data are observations reflecting differences in the world. They are unorganized and unprocessed facts.

Information is aggregation of data that makes decision making easier, thus meaning can be attached and contextualized. Information often answers questions such as what, who, when, where. Knowledge is information that is justifiably believed to be true. It is an understanding gained through experience and it answers the 'how' question. Finally, wisdom represents principles by integrating knowledge and answers 'why' questions. [7]

Figure 2.3 Information Hierarchy Pyramid

In order to do useful computation, one has to segregate some part of the physical world and create a conceptual model. The definitions of what concepts are relevant are defined in the conceptual model and other information is considered as irrelevant. The conceptual model created is used to design and implement a computational model. A conceptual model for a given criteria such as diabetes would include information such as the patient name, weight, blood glucose levels, and HbA1c

values. These data are relevant and are part of the conceptual model while this information will be used in the computational model to perform the analytical representation of the data. Therefore, there should be rules that govern the mapping of the information in terms of numbers or symbols into the computational analysis. This will enable correct capture and preservation of meaning of the data provided. The nation's healthcare system is undergoing a transformation in an effort to improve quality, safety and efficiency of care. On Feb. 17, 2009, President Obama signed the American Recovery and Reinvestment Act (ARRA) of 2009. The act, more known as the Recovery Act, is a critical measure to modernize the nation's infrastructure, one of which is the "Health Information Technology for Economic and Clinical Health (HITECH) Act". The law provides major opportunities for the Department of Health and Human Services (DHHS), its partner agencies, and the States to improve the nation’s healthcare through health information technology (HIT) by promoting the meaningful use of electronic health records via incentives.

The ARRA authorizes the Centers for Medicare & Medicaid Services (CMS) to provide reimbursement incentives for physician and hospital providers who are successful in becoming “meaningful users” of an Electronic Health Records (EHR). Meaningful use is defined by the use of certified EHR technology in a meaningful manner to ensure that the certified EHR technology is connected in a manner that provides for the electronic exchange of health information to improve the quality of care. The concept of meaningful use rests on the “5 pillars” of health outcomes policy priorities, namely:

- Improve quality, safety, efficiency, and reduction of health disparities
- Engage patients and families in their health
- Improve care coordination
- Improve population and public health
- Ensure adequate privacy and security

protection for personal health information in order to encourage widespread EHR adoption, promote innovation and to avoid imposing excessive burden on healthcare providers, meaningful use was showcased as a phased approach, which is divided into three stages that span 2011 (data capture and sharing), 2013 (advanced clinical processes) and 2015 (improved outcomes). The incentive payments range from $44,000 over 5 years for the Medicare providers and $63,750 over 6 years for Medicaid providers (starting in 2011). Participation in the CMS EHR incentive program is totally voluntary, however if physician and hospital providers fail to join in by 2015, there will be negative adjustments to their Medicare/Medicaid fees starting at a 1% reduction and escalating to a 3% reduction by 2017 and beyond. The degree to which health informatics can deliver on the promises to improve healthcare delivery depends greatly on how the emerging information technologies are deployed and managed. This series of articles will shed a light on a few challenges for the implementation of a successful healthcare information system such as regulatory environment and changes in healthcare consumerism and issues facing information technology adoption within various healthcare settings. [5]

CHAPTER3
Technologies used in automed medical software

3.1 Java database connectivity (JDBC)

The JDBC API is the industry standard for database-independent connectivity between the Java programming language and a wide range of databases. The JDBC API provides a call-level API for SQL-based database access. JDBC technology allows you to use the Java programming language to exploit "Write Once, Run Anywhere" capabilities for applications that require access to enterprise data.

3.1.1 JDBC API Overview

The JDBC API makes it possible to do three things:

- Establish a connection with a database or access any tabular data source
- Send SQL statements
- Process the results

3.1.2 JDBC Architecture

The JDBC API contains two major sets of interfaces: the first is the JDBC API for application writers, and the second is the lower-level JDBC driver API for driver writers. JDBC technology drivers fit into one of four categories. Applications and applets can access databases via the JDBC API using pure Java JDBC technology-based drivers, as shown in this figure: [6]

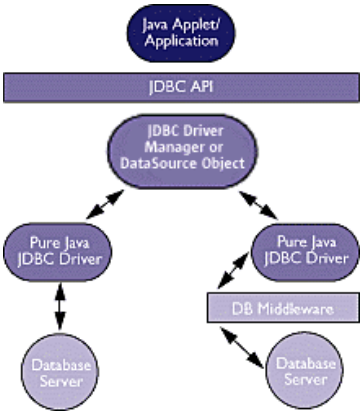


Figure 3.1 jdbc Architecture

Left side, Type 4: Direct-to-Database Pure Java Driver This style of driver converts JDBC calls into the network protocol used directly by DBMSs, allowing a direct call from the client machine to the DBMS server and providing a practical solution for intranet access. Right side, Type 3: Pure Java Driver for Database Middleware

This style of driver translates JDBC calls into the middleware vendor's protocol, which is then translated to a DBMS protocol by a middleware server. The middleware provides connectivity to many different databases. The graphic below illustrates JDBC connectivity using ODBC drivers and existing database client librarie. [6]

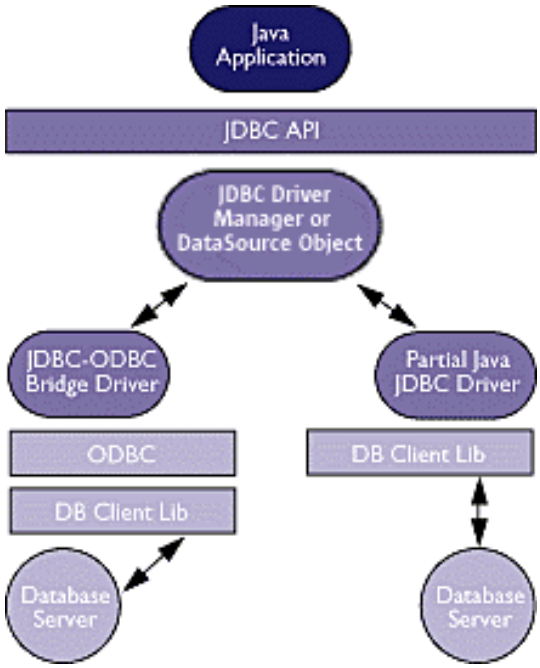


Figure 3.2 jdbc Architecture

Left side, Type 1: JDBC-ODBC Bridge plus ODBC Driver This combination provides JDBC access via ODBC drivers. ODBC binary code -- and in many cases, database client code -- must be loaded on each client machine that uses a JDBC-ODBC Bridge. Sun provides a JDBC-ODBC Bridge driver, which is appropriate for experimental use and for situations in which no other driver is available. Right side, Type 2: A native API partly Java technology-enabled driver This type of driver converts JDBC calls into calls on the client API for Oracle, Sybase, Informix, DB2, or another DBMS. Note that, like the bridge driver, this style of driver requires that some binary code be loaded on each client machine. For comparison of driver types, please see the article published in Computerworld.

3.1.3 Partnering for Progress

Sun worked with an array of companies in the industry to create and rapidly establish the JDBC API as the industry-standard, open interface for Java applications to access databases. [6]

3.1.4 Industry Momentum

Leading database, middleware and tool vendors have been building support for JDBC technology into many new products. This ensures that customers can build portable Java applications while choosing from a wide range of competitive products for the solution best suited to their needs. See the Industry Support page for a list of companies that are shipping products with support for JDBC technology.

3.2 Advantages of JDBC Technology

3.2.1 Leverage Existing Enterprise Data

With JDBC technology, businesses are not locked in any proprietary architecture, and can continue to use their installed databases and access information easily -- even if it is stored on different database management systems. [6]

3.2.2 Simplified Enterprise Development

The combination of the Java API and the JDBC API makes application development easy and economical. JDBC hides the complexity of many data access tasks, doing most of the "heavy lifting"for the programmer behind the scenes. The JDBC API is simple to learn, easy to deploy, and inexpensive to maintain. [6]

3.2.3 Zero Configuration for Network Computers

With the JDBC API, no configuration is required on the client side. With a driver written in the Java programming language, all the information needed to make a connection is completely defined by the JDBC URL or by a DataSource object registered with a Java Naming and Directory Interface (JNDI) naming service. Zero configuration for clients supports the network computing paradigm and centralizes software maintenance.[6]

3.3 Key Features

3.3.1 Full Access to Metadata

The JDBC API provides metadata access that enables the development of

sophisticated applications that need to understand the underlying facilities and capabilities of a specific database connection.

3.3.2 No Installation

A pure JDBC technology-based driver does not require special installation; it is automatically downloaded as part of the applet that makes the JDBC calls.

3.3.3 Database Connection Identified by URL

JDBC technology exploits the advantages of Internet-standard URLs to identify database connections. The JDBC API includes an even better way to identify and connect to a data source, using a DataSource object, that makes code even more portable and easier to maintain.

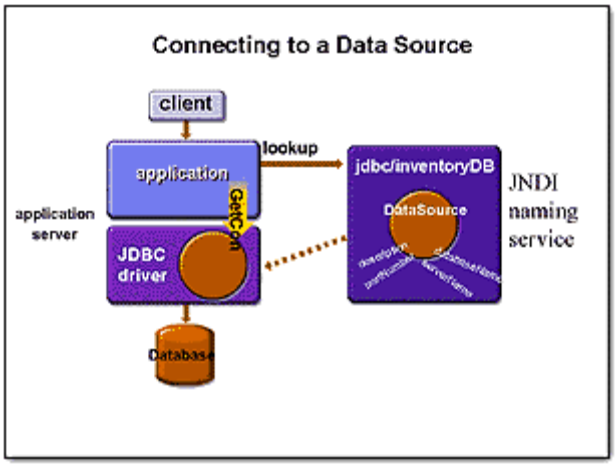


Figure 3.3 jdbc Architecture

In addition to this important advantage, DataSource objects can provide connection pooling and distributed transactions, essential for enterprise database computing. This functionality is provided transparently to the programmer. [6]

3.3.4 Included in the Java Platform

As a core part of the Java 2 Platform, the JDBC API is available anywhere that the platform is. This means that your applications can truly write database applications once and access data anywhere. The JDBC API is included in both the Java 2 Platform, Standard Edition (J2SE) and the Java 2 Platform, Enterprise Edition (J2EE), providing server-side functionality for industrial strength scalability.

An example of a J2EE based architecture that includes a JDBC implementation:

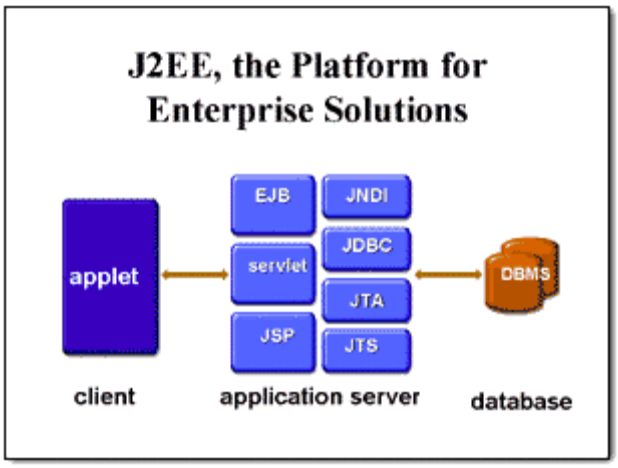


Figure 3.4 jdbc Architecture

3.4 Hibernate Object Relational Mapping (ORM) Framework

Hibernate ORM (Hibernate in short) is an object-relational mapping tool for the Java programming language. It provides a framework for mapping an object-oriented domain model to a relational database. Hibernate handles object-relational impedance mismatch problems by replacing direct, persistent database accesses with high-level object handling functions.

Hibernate is free software that is distributed under the GNU Lesser General Public License 2.1.

Hibernate's primary feature is mapping from Java classes to database tables, and mapping from Java data types to SQL data types. Hibernate also provides data query and retrieval facilities. It generates SQL calls and relieves the developer from the manual handling and object conversion of the result set

3.4.1 Mapping

The mapping of Java classes to database tables is implemented by the configuration of an XML file or by using Java Annotations. When using an XML file, Hibernate can generate skeleton source code for the persistence classes. This is auxiliary when annotations are used. Hibernate can use the XML file or the Java annotations to maintain the database schema.

There are provided facilities to arrange one-to-many and many-to-many relationships between classes. In addition to managing associations between objects, Hibernate can also manage reflexive associations wherein an object has a one-to-many relationship with other instances of the class type.

Hibernate supports the mapping of custom value types. This makes the following scenarios possible:

- Overriding the default SQL type when mapping a column to a property.
- Mapping Java Enums to columns as though they were regular properties.
- Mapping a single property to multiple columns.

Definition: Objects in an object-oriented application follow OOP principles, while objects in the back-end follow database normalization principles, resulting in different representation requirements. This problem is called "object-relational impedance mismatch". Mapping is a way of resolving the object-relational impedance mismatch problem.

Mapping informs the ORM tool of what Java class object to store in which database table.

3.4.2 Hibernate Query Language (HQL)

Hibernate provides an SQL inspired language called Hibernate Query Language (HQL) that allows SQL-like queries to be written against Hibernate's data objects. *Criteria Queries* are provided as an object-oriented alternative to HQL. Criteria Query is used to modify the objects and provide the restriction for the objects. HQL (Hibernate Query Language) is the object-oriented version of SQL. It generates database independent queries so that there is no need to write database-specific queries. Without this capability, changing the database would require individual SQL queries to be changed as well, leading to maintenance issues.

3.4.3 Persistence

Hibernate provides transparent persistence for Plain Old Java Objects (POJOs). The only strict requirement for a persistent class is a no-argument constructor, not necessarily *public*. Proper behavior in some applications also requires special attention to the *equals()* and *hashCode()* methods. Hibernate recommends providing identifier attribute and it's becoming a requirement in an upcoming release.

Collections of data objects are typically stored in Java collection classes such as implementations of the Set and List interfaces. Java generics, introduced in Java 5, are supported. Hibernate can be configured to lazy load associated collections. Lazy loading is the default as of Hibernate 3.

Related objects can be configured to *cascade* operations from one to the other. For example, a parent Album object can be configured to cascade its save and/or delete operation to its child Track objects.

3.4.4 Integration

Hibernate can be used both in standalone Java applications and in Java EE applications using servlets, EJB session beans, and JBI service components. It can also be included as a feature in other programming languages. For example, Adobe integrated Hibernate into version 9 of ColdFusion (which runs on J2EE app servers) with an abstraction layer of new functions and syntax added into CFML.

3.5 MySQL

MySQL is an open-source relational database management system(RDBMS) Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedishcompany MySQL AB, now owned by Oracle Corporation. For proprietary use, several paid editions are available, and offer additional functionality.

MySQL is a central component of the LAMP open-source web application software stack (and other "AMP" stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python". Applications that use the MySQL database include: TYPO3, MODx, Joomla, WordPress, SimpleMachinesForum, phpBB, MyBB and Drupal. MySQL is also used in many high-profile, large-scale websites, including Google (though not for searches), Facebook, Twitter, Flickr, and YouTube.

3.5.1 Deployment

MySQL can be built and installed manually from source code, but it is more commonly installed from a binary package unless special customizations are required. On most Linux distributions, the package management system can download and install MySQL with minimal effort, though further configuration is often required to adjust security and optimization settings. MySQL can be built and installed manually from source code, but it is more commonly installed from a binary package unless special customizations are required. On most Linux distributions MySQL can be built and installed manually from source code, but it is more commonly installed from a

binary package unless special customizations are required. On most Linux distributions.

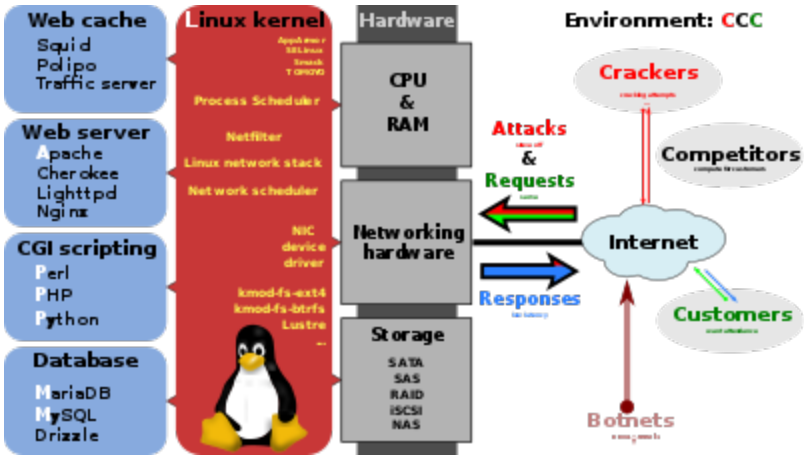


Figure 3.5 software bundle, displayed here together with Squid.

Though MySQL began as a low-end alternative to more powerful proprietary databases, it has gradually evolved to support higher-scale needs as well. It is still most commonly used in small to medium scale single-server deployments, either as a component in a LAMP-based web application or as a standalone database server. Much of MySQL's appeal originates in its relative simplicity and ease of use, which is enabled by an ecosystem of open source tools such as phpMyAdmin. In the medium range, MySQL can be scaled by deploying it on more powerful hardware, such as a multi-processor server with gigabytes of memory.

There are, however, limits to how far performance can scale on a single server ('scaling up'), so on larger scales, multi-server MySQL ('scaling out') deployments are required to provide improved performance and reliability. A typical high-end configuration can include a powerful master database which handles data write operations and is replicated to multiple slaves that handle all read operations. The master server continually pushes binlog events to connected slaves so in the event of failure a slave can be promoted to become the new master, minimizing downtime. Further improvements in performance can be achieved by caching the results from database queries in memory using memcached, or breaking down a database into smaller chunks called shards which can be spread across a number of distributed server clusters.

3.6 JavaServer Pages

avaServer Pages (JSP) is a technology for developing Webpages that supports dynamic content. This helps developers insert java code in HTML pages by making use of special JSP tags, most of which start with <% and end with %>.

A JavaServer Pages component is a type of Java servlet that is designed to fulfill the role of a user interface for a Java web application. Web developers write JSPs as text files that combine HTML or XHTML code, XML elements, and embedded JSP actions and commands.

Using JSP, you can collect input from users through Webpage forms, present records from a database or another source, and create Webpages dynamically.

JSP tags can be used for a variety of purposes, such as retrieving information from a database or registering user preferences, accessing JavaBeans components, passing control between pages, and sharing information between requests, pages etc.

Why Use JSP?

JavaServer Pages often serve the same purpose as programs implemented using the **Common Gateway Interface (CGI)**. But JSP offers several advantages in comparison with the CGI.

- Performance is significantly better because JSP allows embedding Dynamic Elements in HTML Pages itself instead of having separate CGI files.
- JSP are always compiled before they are processed by the server unlike CGI/Perl which requires the server to load an interpreter and the target script each time the page is requested.
- JavaServer Pages are built on top of the Java Servlets API, so like Servlets, JSP also has access to all the powerful Enterprise Java APIs, including **JDBC, JNDI, EJB, JAXP,** etc.
- JSP pages can be used in combination with servlets that handle the business logic, the model supported by Java servlet template engines.

Finally, JSP is an integral part of Java EE, a complete platform for enterprise class applications. This means that JSP can play a part in the simplest applications to the most complex and demanding.

3.6.1 Advantages of JSP

Following table lists out the other advantages of using JSP over other technologies −

vs. Active Server Pages (ASP)

The advantages of JSP are twofold. First, the dynamic part is written in Java, not Visual Basic or other MS specific language, so it is more powerful and easier to use. Second, it is portable to other operating systems and non-Microsoft Web servers.

vs. Pure Servlets

It is more convenient to write (and to modify!) regular HTML than to have plenty of println statements that generate the HTML.

vs. Server-Side Includes (SSI)

SSI is really only intended for simple inclusions, not for "real" programs that use form data, make database connections, and the like.

vs. JavaScript

JavaScript can generate HTML dynamically on the client but can hardly interact with the web server to perform complex tasks like database access and image processing etc.

vs. Static HTML

Regular HTML, of course, cannot contain dynamic information.

3.6.2 JSP – Architecture

The web server needs a JSP engine, i.e, a container to process JSP pages. The JSP container is responsible for intercepting requests for JSP pages. This tutorial makes use of Apache which has built-in JSP container to support JSP pages development.

A JSP container works with the Web server to provide the runtime environment and other services a JSP needs. It knows how to understand the special elements that are part of JSPs.

Following diagram shows the position of JSP container and JSP files in a Web application.

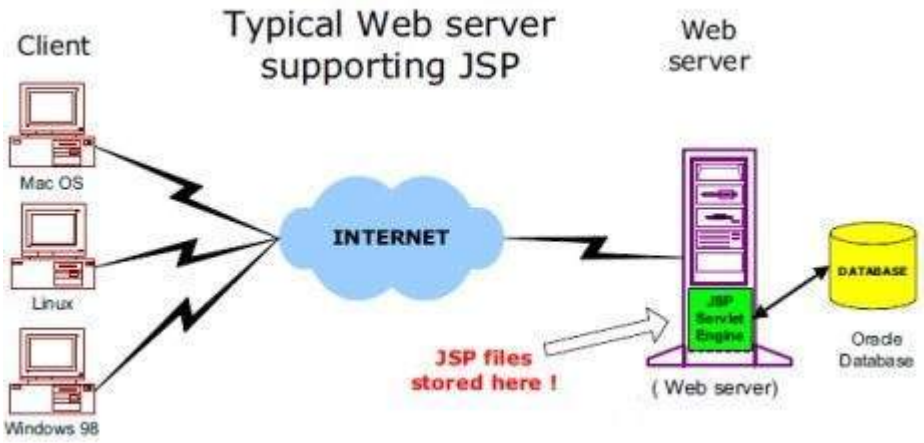


Figure 3.6 JSP – Architecture

3.6.3 JSP Processing

The following steps explain how the web server creates the Webpage using JSP −

- As with a normal page, your browser sends an HTTP request to the web server.
- The web server recognizes that the HTTP request is for a JSP page and forwards it to a JSP engine. This is done by using the URL or JSP page which ends with **.jsp** instead of **.html**.
- The JSP engine loads the JSP page from disk and converts it into a servlet content. This conversion is very simple in which all template text is converted to println( ) statements and all JSP elements are converted to Java code. This code implements the corresponding dynamic behavior of the page.
- The JSP engine compiles the servlet into an executable class and forwards the original request to a servlet engine.
- A part of the web server called the servlet engine loads the Servlet class and executes it. During execution, the servlet produces an output in HTML format. The output is furthur passed on to the web server by the servlet engine inside an HTTP response.
- The web server forwards the HTTP response to your browser in terms of static HTML content.
- Finally, the web browser handles the dynamically-generated HTML page inside the HTTP response exactly as if it were a static page.

All the above mentioned steps can be seen in the following diagram −

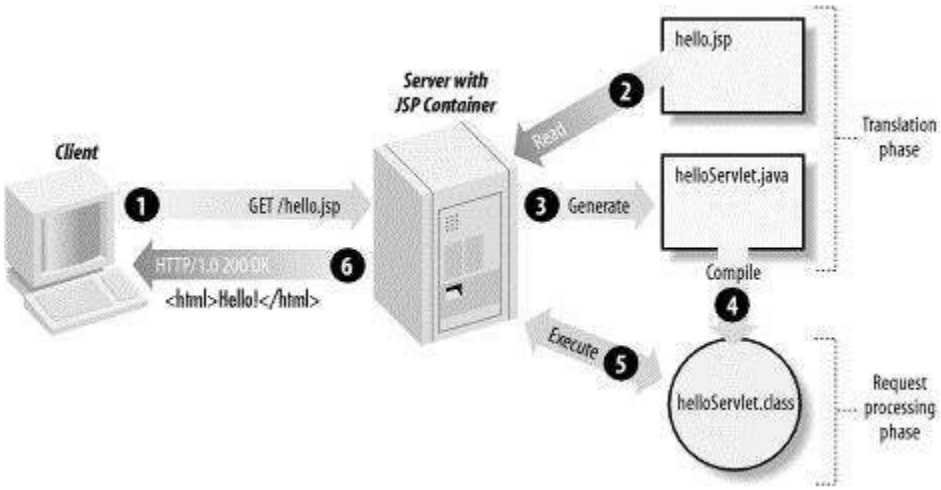


Figure 3.7 JSP – Architecture

Typically, the JSP engine checks to see whether a servlet for a JSP file already exists and whether the modification date on the JSP is older than the servlet. If the JSP is older than its generated servlet, the JSP container assumes that the JSP hasn't

changed and that the generated servlet still matches the JSP's contents. This makes the process more efficient than with the other scripting languages (such as PHP) and therefore faster.[10]

So in a way, a JSP page is really just another way to write a servlet without having to be a Java programming wiz. Except for the translation phase, a JSP page is handled exactly like a regular servlet.

3.7 Apache Tomcat

Apache Tomcat, often referred to as **Tomcat Server**, is an open-source Java Servlet Containerdeveloped by the Apache Software Foundation (ASF). Tomcat implements several Java EEspecifications including Java Servlet, JavaServer Pages (JSP), Java EL, and WebSocket, and provides a "pure Java" HTTP web server environment in which Java code can run.

Tomcat is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation, released under the Apache License 2.0 license, and is open-source software.

3.7 Apache Tomcat Components

Tomcat 4.x was released with Catalina (a servlet container), Coyote (an HTTP connector) and Jasper (a JSP engine).

Catalina

Catalina is Tomcat's servlet container. Catalina implements Sun Microsystems's specifications for servlet and JavaServer Pages (JSP). In Tomcat, a Realm element represents a "database" of usernames, passwords, and roles (similar to Unix groups) assigned to those users. Different implementations of Realm allow Catalina to be integrated into environments where such authentication information is already being created and maintained, and then use that information to implement Container Managed Security as described in the Servlet Specification.

Coyote

Coyote is a Connector component for Tomcat that supports the HTTP 1.1 protocol as a web server. This allows Catalina, nominally a Java Servlet or JSP container, to also act as a plain web server that serves local files as HTTP documents

Coyote listens for incoming connections to the server on a specific TCP port and forwards the request to the Tomcat Engine to process the request and send back a response to the requesting client. Another Coyote Connector, Coyote JK, listens

similarly but instead forwards its requests to another web server, such as Apache, using the JK protocol. This usually offers better performance.

Jasper

Jasper is Tomcat's JSP Engine. Jasper parses JSP files to compile them into Java code as servlets (that can be handled by Catalina). At runtime, Jasper detects changes to JSP files and recompiles them.

As of version 5, Tomcat uses Jasper 2, which is an implementation of the Sun Microsystems's JSP 2.0 specification. From Jasper to Jasper 2, important features were added:

- JSP Tag library pooling - Each tag markup in JSP file is handled by a tag handler class. Tag handler class objects can be pooled and reused in the whole JSP servlet.
- Background JSP compilation - While recompiling modified JSP Java code, the older version is still available for server requests. The older JSP servlet is deleted once the new JSP servlet has finished being recompiled.
- Recompile JSP when included page changes - Pages can be inserted and included into a JSP at runtime. The JSP will not only be recompiled with JSP file changes but also with included page changes.
- JDT Java compiler - Jasper 2 can use the Eclipse JDT (Java Development Tools) Java compiler instead of Ant and `javac`.

Three new components were added with the release of Tomcat 7:

Cluster

This component has been added to manage large applications. It is used for load balancing that can be achieved through many techniques. Clustering support currently requires the JDK version 1.5 or higher.

High availability

A high-availability feature has been added to facilitate the scheduling of system upgrades (e.g. new releases, change requests) without affecting the live environment. This is done by dispatching live traffic requests to a temporary server on a different port while the main server is upgraded on the main port. It is very useful in handling user requests on high-traffic web applications.

Web application

It has also added user- as well as system-based web applications enhancement to add support for deployment across the variety of environments. It also tries to manage sessions as well as applications across the network.

Tomcat is building additional components. A number of additional components may be used with Apache Tomcat. These components may be built by users should they need them or they can be downloaded from one of the mirrors.

3.8 JavaMail API

The JavaMail API provides a platform-independent and protocol-independent framework to build mail and messaging applications. The JavaMail API provides a set of abstract classes defining objects that comprise a mail system. It is an optional package (standard extension) for reading, composing, and sending electronic messages.

JavaMail provides elements that are used to construct an interface to a messaging system, including system components and interfaces. While this specification does not define any specific implementation, JavaMail does include several classes that implement RFC822 and MIME Internet messaging standards. These classes are delivered as part of the JavaMail class package.

Following are some of the protocols supported in JavaMail API:

- **SMTP**: Acronym for **Simple Mail Transfer Protocol**. It provides a mechanism to deliver email.
- **POP**: Acronym for **Post Office Protocol**. POP is the mechanism most people on the Internet use to get their mail. It defines support for a single mailbox for each user. RFC 1939 defines this protocol.
- **IMAP**: Acronym for **Internet Message Access Protocol**. It is an advanced protocol for receiving messages. It provides support for multiple mailbox for each user, in addition to, mailbox can be shared by multiple users. It is defined in RFC 2060.
- **MIME**: Acronym for **Multipurpose Internet Mail Extensions**. . It is not a mail transfer protocol. Instead, it defines the content of what is transferred: the format of the messages, attachments, and so on. There are many different documents that take effect here: RFC 822, RFC 2045, RFC 2046, and RFC 2047. As a user of the JavaMail API, you usually don't need to worry about these formats. However, these formats do exist and are used by your programs.
- **NNTP and Others**: There are many protocols that are provided by third-party providers. Some of them are Network News Transfer Protocol (NNTP), Secure Multipurpose Internet Mail Extensions (S/MIME) etc.

Details of these will be covered in the subsequent chapters.

3.8.1 Architecture

As said above the java application uses JavaMail API to compose, send and receive emails. The following figure illustrates the architecture of JavaMail:

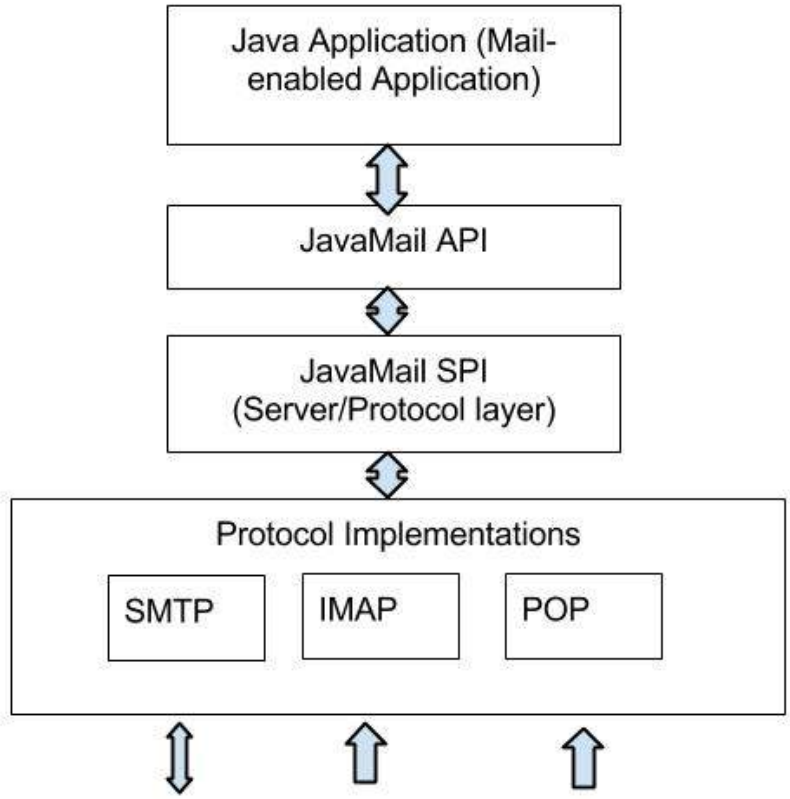


Figure 3.8 java mail API Architecture

The abstract mechanism of JavaMail API is similar to other J2EE APIs, such as JDBC, JNDI, and JMS. As seen the architecture diagram above, JavaMail API is divided into two main parts:

- An application-independent part: An application-programming interface (API) is used by the application components to send and receive mail messages, independent of the underlying provider or protocol used.
- A service-dependent part: A service provider interface (SPI) speaks the protocol-specific languages, such as SMTP, POP, IMAP, and Network News Transfer Protocol (NNTP). It is used to plug in a provider of an e-mail service to the J2EE platform.

3.9 Spring Framework

The **Spring Framework** is an application framework and inversion of control container for the Java platform. The framework's core features can be used by any Java application, but there are extensions for building web applications on top of the Java EE (Enterprise Edition) platform. Although the framework does not impose

any specific programming model, it has become popular in the Java community as an addition to, or even replacement for the Enterprise JavaBeans (EJB) model. The Spring Framework is open source.

The Spring Web MVC framework provides Model-View-Controller (MVC) architecture and ready components that can be used to develop flexible and loosely coupled web applications. The MVC pattern results in separating the different aspects of the application (input logic, business logic, and UI logic), while providing a loose coupling between these elements.

- The **Model** encapsulates the application data and in general they will consist of POJO.
- The **View** is responsible for rendering the model data and in general it generates HTML output that the client's browser can interpret.
- The **Controller** is responsible for processing user requests and building an appropriate model and passes it to the view for rendering.
-

The DispatcherServlet

The Spring Web model-view-controller (MVC) framework is designed around a *DispatcherServlet* that handles all the HTTP requests and responses. The request processing workflow of the Spring Web MVC *DispatcherServlet* is illustrated in the following diagram −

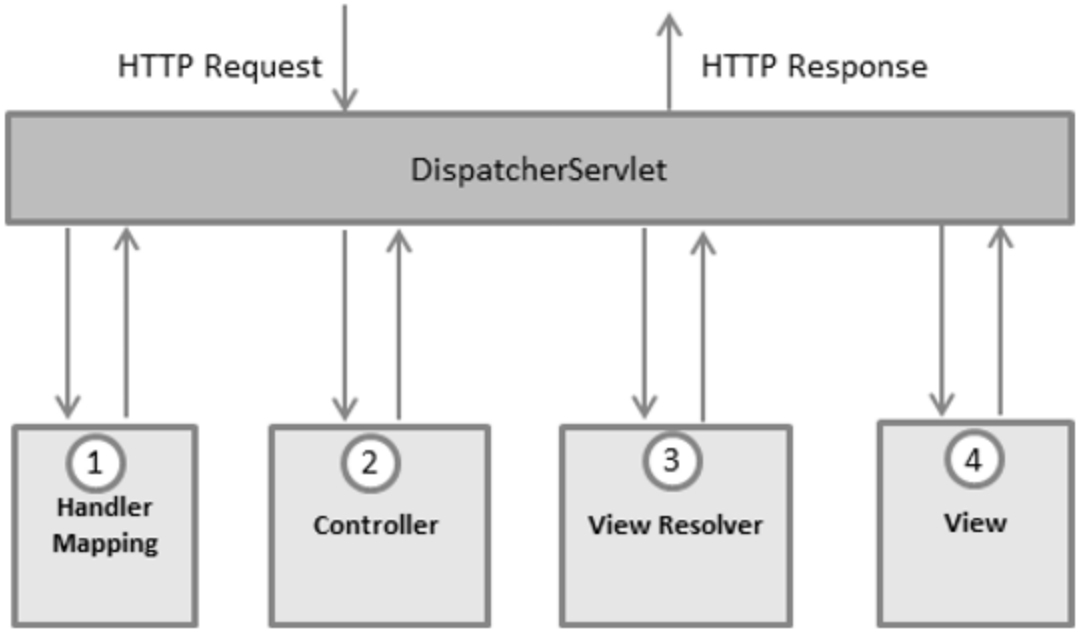


Figure 3.9 Spring Framework Architecture

Following is the sequence of events corresponding to an incoming HTTP request to *DispatcherServlet* −

- After receiving an HTTP request, *DispatcherServlet* consults the *HandlerMapping* to call the appropriate *Controller*.
- The *Controller* takes the request and calls the appropriate service methods based on used GET or POST method. The service method will set model data based on defined business logic and returns view name to the *DispatcherServlet*.
- The *DispatcherServlet* will take help from *ViewResolver* to pickup the defined view for the request.
- Once view is finalized, The *DispatcherServlet* passes the model data to the view which is finally rendered on the browser.

All the above-mentioned components, i.e. HandlerMapping, Controller, and ViewResolver are parts of *WebApplicationContext* w which is an extension of the plain*ApplicationContext* with some extra features necessary for web applications.

3.10 Jackson JSON Java Parser API

ackson JSON Java Parser is very popular and used in Spring framework too. Java JSON Processing API is not very user friendly and doesn't provide features for automatic transformation from Json to Java object and vice versa. Luckily, we have some alternative APIs that we can use for JSON processing. In last article we learned about Google Gson API and saw how easy to use it.

To use Jackson JSON Java API in our project, we can add it to the project build path or if you are using maven, we can add below dependency.

```
<dependency>
	<groupId>com.fasterxml.jackson.core</groupId>
	<artifactId>jackson-databind</artifactId>
	<version>2.2.3</version>
</dependency>
```

Figure 3.10 jakson json dependency

jackson-databind jar depends on **jackson-core** and **jackson-annotations** libraries, so if you are adding them directly to build path, make sure you add all three otherwise you will get runtime error.

Jackson JSON Parser API provides easy way to convert JSON to POJO Object and supports easy conversion to Map from JSON data. Jackson supports generics too and directly converts them from JSON to object.

Jackson JSON Example

For our example for JSON to POJO/Java object conversion, we will take a complex example with nested object and arrays. We will use arrays, list and Map in java objects for conversion. Our complex json is stored in a file employee.txt with below structure:

```
{
  "id": 123,
  "name": "Pankaj",
  "permanent": true,
  "address": {
    "street": "Albany Dr",
    "city": "San Jose",
    "zipcode": 95129
  },
  "phoneNumbers": [
    123456,
    987654
  ],
  "role": "Manager",
  "cities": [
    "Los Angeles",
    "New York"
  ],
  "properties": {
    "age": "29 years",
    "salary": "1000 USD"
  }
}
```

Figure 3.11 jakson json sample format

3.11 Servlets

Java Servlets are programs that run on a Web or Application server and act as a middle layer between a requests coming from a Web browser or other HTTP client and databases or applications on the HTTP server.

Using Servlets, you can collect input from users through web page forms, present records from a database or another source, and create web pages dynamically.

Java Servlets often serve the same purpose as programs implemented using the Common Gateway Interface (CGI). But Servlets offer several advantages in comparison with the CGI.

- Performance is significantly better.
- Servlets execute within the address space of a Web server. It is not necessary to create a separate process to handle each client request.
- Servlets are platform-independent because they are written in Java.
- Java security manager on the server enforces a set of restrictions to protect the resources on a server machine. So servlets are trusted.
- The full functionality of the Java class libraries is available to a servlet. It can communicate with applets, databases, or other software via the sockets and RMI mechanisms that you have seen already.

3.11.1 Servlets Architecture

The following diagram shows the position of Servlets in a Web Application.

Java Servlets are programs that run on a Web or Application server and act as a middle layer between a request coming from a Web browser or other HTTP client and databases or applications on the HTTP server.

Using Servlets, you can collect input from users through web page forms, present records from a database or another source, and create web pages dynamically.

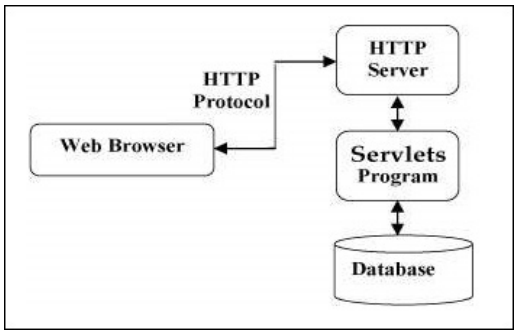


Figure 3.12 java servlert

3.11.2 Servlets Tasks

Servlets perform the following major tasks:

- Read the explicit data sent by the clients (browsers). This includes an HTML form on a Web page or it could also come from an applet or a custom HTTP client program.
- Read the implicit HTTP request data sent by the clients (browsers). This includes cookies, media types and compression schemes the browser understands, and so forth.
- Process the data and generate the results. This process may require talking to a database, executing an RMI or CORBA call, invoking a Web service, or computing the response directly.
- Send the explicit data (i.e., the document) to the clients (browsers). This document can be sent in a variety of formats, including text (HTML or XML), binary (GIF images), Excel, etc.
- Send the implicit HTTP response to the clients (browsers). This includes telling the browsers or other clients what type of document is being returned (e.g., HTML), setting cookies and caching parameters, and other such tasks.

3.11.3 Servlets Packages

Java Servlets are Java classes run by a web server that has an interpreter that supports the Java Servlet specification.

Servlets can be created using the **javax.servlet** and **javax.servlet.http**packages, which are a standard part of the Java's enterprise edition, an expanded version of the Java class library that supports large-scale development projects.

These classes implement the Java Servlet and JSP specifications. At the time of writing this tutorial, the versions are Java Servlet 2.5 and JSP 2.1.

Java servlets have been created and compiled just like any other Java class. After you install the servlet packages and add them to your computer's Classpath, you can compile servlets with the JDK's Java compiler or any other current compiler.

3.12 QR code

QR code (abbreviated from **Quick Response Code**) is the trademark for a type of matrix barcode (or two-dimensional barcode) first designed in 1994 for

the automotive industry in Japan. A barcode is a machine-readable optical label that contains information about the item to which it is attached. A QR code uses four standardized encoding modes (numeric, alphanumeric, byte/binary, and kanji) to efficiently store data; extensions may also be used.

The Quick Response (QR code) system became popular outside the automotive industry due to its fast readability and greater storage capacity compared to standard UPC barcodes. Applications include product tracking, item identification, time tracking, document management, and general marketing.

A QR code consists of black squares arranged in a square grid on a white background, which can be read by an imaging device such as a camera, and processed using Reed–Solomon error correction until the image can be appropriately interpreted. The required data is then extracted from patterns that are present in both horizontal and vertical components of the image. It is now widely used around the world to get to websites quicker, and it can also be used for advertisements.

Figure 3.13 Version 3 (29×29). Content: "Version 3 QR Code"

3.13 Barcode

A **barcode** (also **bar code**) is an optical, machine-readable, representation of data; the data usually describes something about the object that carries the barcode. Traditional barcodes systematically represent data by varying the widths and spacings of parallel lines, and may be referred to as linear or one-dimensional (1D). Later, two-dimensional (2D) variants were developed, using rectangles, dots, hexagons and other geometric patterns, called *matrix codes* or *2D barcodes*, although they do not use bars as such. Initially, barcodes were only scanned by special optical scannerscalled barcode readers. Later application software became available for devices that could read images, such as smartphones with cameras.

Barcode was invented by Norman Joseph Woodland and Bernard Silver and patented in US in 1952 (US Patent 2,612,994). The invention was based on Morse code that

was extended to thin and thick bars. However, it took over twenty years before this invention became commercially successful. An early use of one type of barcode in an industrial context was sponsored by the Association of American Railroads in the late 1960s. Developed by General Telephone and Electronics (GTE) and called KarTrak ACI (Automatic Car Identification), this scheme involved placing colored stripes in various combinations on steel plates which were affixed to the sides of railroad rolling stock. Two plates were used per car, one on each side, with the arrangement of the colored stripes encoding information such as ownership, type of equipment, and identification number. The plates were read by a trackside scanner, located for instance, at the entrance to a classification yard, while the car was moving past.The project was abandoned after about ten years because the system proved unreliable after long-term use.

Barcodes became commercially successful when they were used to automate supermarket checkout systems, a task for which they have become almost universal. Their use has spread to many other tasks that are generically referred to as automatic identification and data capture (AIDC). The very first scanning of the now-ubiquitous Universal Product Code (UPC) barcode was on a pack of Wrigley Company chewing gum in June 1974. QR codes, a specific type of 2D barcode, have recently become very popular.

Other systems have made inroads in the AIDC market, but the simplicity, universality and low cost of barcodes has limited the role of these other systems, particularly before technologies such as radio-frequency identification (RFID) became available after 2000.

Figure 3.14 A UPC-A barcode symbol

3.14 Java Media Framework

The **Java Media Framework** (**JMF**) is a Java library that enables audio, video and other time-based media to be added to Java applications and applets. This optional package, which can capture, play, stream, and transcode multiple media

formats, extends the Java Platform, Standard Edition (Java SE) and allows development of cross-platform multimedia applications.

3.14.1 Versions and licensing

An initial, playback-only version of JMF was developed by Sun Microsystems, Silicon Graphics, and Intel, and released as JMF 1.0 in 1997. JMF 2.0, developed by Sun and IBM, came out in 1999 and added capture, streaming, pluggable codecs, and transcoding. JMF is branded as part of Sun's "Desktop" technology of J2SE opposed to the Java server-side and client-side application frameworks. The notable exceptions are Java appletsand Java Web Start, which have access to the full JMF in the web browser's or appletviewer's underlying JRE.

JMF 2.0 originally shipped with an MP3 decoder and encoder. This was removed in 2002, and a new MP3 playback-only plug-in was posted in 2004.

JMF binaries are available under a custom license, and the source is available under the SCSL.

The current version ships with four JAR files, and shell scripts to launch four JMF-based applications:

- **JMStudio** - A simple player GUI
- **JMFRegistry** - A GUI for managing the JMF "registry," which manages preferences, plug-ins, etc.
- **JMFCustomizer** - Used for creating a JAR file that contains only the classes needed by a specific JMF application, which allows developers to ship a smaller application.
- **JMFInit**

JMF is available in an all-Java version and as platform-specific "performance packs", which can contain native-code players for the platform, and/or hooks into a multimedia engine specific to that platform. JMF 2.0 offers performance packs for Linux, Solaris (on SPARC) and Windows.

In January 2011, Tudor Holton of Bentokit Project released a Debian package for the JMF to alleviate difficulties that had arisen over time when installing the JMF on Debian and Ubuntu GNU/Linux. This package does not contain the JMF, but presents the user with the JMF License, retrieves it from the Oracle website, and then installs it. A similar Debian package installer for the JMF MP3 Plugin was also built in February 2011.

3.15 Multi-factor authentication

Multi-factor authentication (**MFA**) is a method of confirming a user's claimed identity in which a user is granted access only after successfully presenting 2 or more pieces of evidence (or factors) to an authenticationmechanism: knowledge (something they and only they know), possession (something they and only they have), and inherence (something they and only they are).

Two-factor authentication (also known as **2FA**) is a type (subset) of multi-factor authentication. It is a method of confirming a user's claimed identity by utilizing a combination of *two* different factors: 1) something they know, 2) something they have, or 3) something they are.

A good example of two-factor authentication is the withdrawing of money from an ATM; only the correct combination of a bank card (something that the user possesses) and a PIN (personal identification number, something that the user knows) allows the transaction to be carried out.

Two-step verification or **two-step authentication** is a method of confirming a user's claimed identity by utilizing something they know (password) and a second factor **other** than something they have or something they are. An example of a second step is the user repeating back something that was sent to them through an out-of-bandmechanism. Or the second step might be a 6 digit number generated by an app that is common to the user and the authentication system.

3.15.1 Authentication factors

The use of multiple authentication factors to prove one's identity is based on the premise that an unauthorized actor is unlikely to be able to supply the factors required for access. If, in an authentication attempt, at least one of the components is missing or supplied incorrectly, the user's identity is not established with sufficient certainty and access to the asset (e.g., a building, or data) being protected by multi-factor authentication then remains blocked. The authentication factors of a multi-factor authentication scheme may include:

- some physical object in the possession of the user, such as a USB stick with a secret token, a bank card, a key, etc.
- some secret known to the user, such as a password, PIN, TAN, etc.
- some physical characteristic of the user (biometrics), such as a fingerprint, eye iris, voice, typing speed, pattern in key press intervals, etc.

Knowledge factors

Knowledge factors are the most commonly used form of authentication. In this form, the user is required to prove knowledge of a secret in order to authenticate.

A password is a secret word or string of characters that is used for user authentication. This is the most commonly used mechanism of authentication. Many multi-factor authentication techniques rely on password as one factor of authenticationVariations include both longer ones formed from multiple words (a passphrase) and the shorter, purely numeric, personal identification number (PIN) commonly used for ATM access. Traditionally, passwords are expected to be memorized.

Many secret questions such as "Where were you born?" are poor examples of a knowledge factor because they may be known to a wide group of people, or be able to be researched.

Possession factors

Possession factors ("something the user and only the user has") have been used for authentication for centuries, in the form of a key to a lock. The basic principle is that the key embodies a secret which is shared between the lock and the key, and the same principle underlies possession factor authentication in computer systems. A security token is an example of a possession factor.

Figure 3.14 RSA SecurID token, an example of a disconnected token generator.

Disconnected tokens have no connections to the client computer. They typically use a built-in screen to display the generated authentication data, which is manually typed in by the user.

Connected tokens

Connected tokens are devices that are *physically* connected to the computer to be used. Those devices transmit data automatically. There are a number of different types, including card readers, wireless tags and USB tokens.

Software tokens

A software token (a.k.a. *soft token*) is a type of two-factor authentication security device that may be used to authorize the use of computer services. Software tokens are stored on a general-purpose electronic device such as a desktop computer, laptop, PDA, or mobile phone and can be duplicated. (Contrast hardware tokens, where the credentials are stored on a dedicated hardware device and therefore cannot be duplicated (absent physical invasion of the device).)

Inherent factors

These are factors associated with the user, and are usually biometric methods, including fingerprint, face, voice, or iris recognition. Behavioral biometrics such as keystroke dynamics can also be used.

3.16 Speech recognition

Speech recognition is the inter-disciplinary sub-field of computational linguistics that develops methodologies and technologies that enables the recognition and translation of spoken language into text by computers. It is also known as **automatic speech recognition** (**ASR**), **computer speech recognition** or **speech to text** (**STT**). It incorporates knowledge and research in the linguistics, computer science, and electrical engineering fields.

Some speech recognition systems require "training" (also called "enrollment") where an individual speaker reads text or isolated vocabulary into the system. The system analyzes the person's specific voice and uses it to fine-tune the recognition of that person's speech, resulting in increased accuracy. Systems that do not use training are called "speaker independent" systems. Systems that use training are called "speaker dependent".

Speech recognition applications include voice user interfaces such as voice dialing (e.g. "Call home"), call routing (e.g. "I would like to make a collect call"), domotic appliance control, search (e.g. find a podcast where particular words were spoken), simple data entry (e.g., entering a credit card number), preparation of

structured documents (e.g. a radiology report), speech-to-text processing (e.g., word processors or emails), and aircraft (usually termed direct voice input).

The term voice recognition or speaker identification refers to identifying the speaker, rather than what they are saying. Recognizing the speaker can simplify the task of translating speech in systems that have been trained on a specific person's voice or it can be used to authenticate or verify the identity of a speaker as part of a security process.

From the technology perspective, speech recognition has a long history with several waves of major innovations. Most recently, the field has benefited from advances in deep learning and big data. The advances are evidenced not only by the surge of academic papers published in the field, but more importantly by the worldwide industry adoption of a variety of deep learning methods in designing and deploying speech recognition systems. These speech industry players include Google, Microsoft, IBM, Baidu, Apple, Amazon, Nuance, SoundHound, iFLY TEK many of which have publicized the core technology in their speech recognition systems as being based on deep learning.

3.17 Speech synthesis

Speech synthesis is the artificial production of human speech. A computer system used for this purpose is called a **speech computer** or **speech synthesizer**, and can be implemented in software or hardware products. A **text-to-speech** (**TTS**) system converts normal language text into speech; other systems render symbolic linguistic representations like phonetic transcriptions into speech.

Synthesized speech can be created by concatenating pieces of recorded speech that are stored in a database. Systems differ in the size of the stored speech units; a system that stores phones or diphones provides the largest output range, but may lack clarity. For specific usage domains, the storage of entire words or sentences allows for high-quality output. Alternatively, a synthesizer can incorporate a model of the vocal tract and other human voice characteristics to create a completely "synthetic" voice output.

The quality of a speech synthesizer is judged by its similarity to the human voice and by its ability to be understood clearly. An intelligible text-to-speech program allows people with visual impairments or reading disabilities to listen to written words on a home computer. Many computer operating systems have included speech synthesizers since the early 1990s.

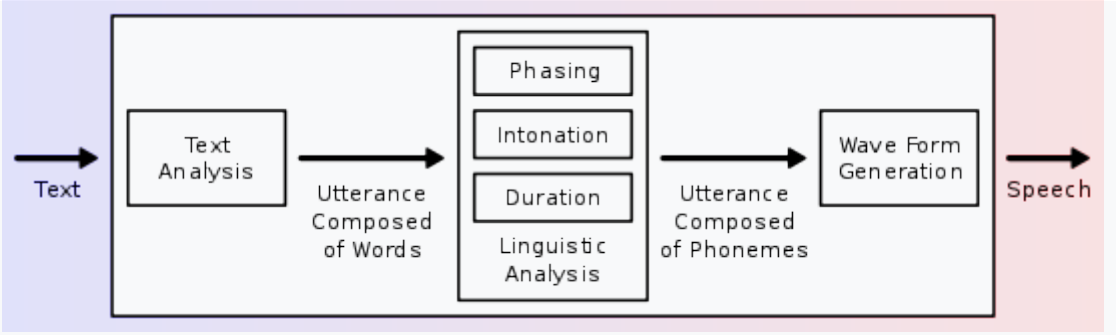


Figure 3.15 Overview of a typical TTS system

A text-to-speech system (or "engine") is composed of two parts: a front-end and a back-end. The front-end has two major tasks. First, it converts raw text containing symbols like numbers and abbreviations into the equivalent of written-out words. This process is often called *text normalization*, *pre-processing*, or *tokenization*. The front-end then assigns phonetic transcriptions to each word, and divides and marks the text into prosodic units, like phrases, clauses, and sentences. The process of assigning phonetic transcriptions to words is called *text-to-phoneme* or *grapheme-to-phoneme* conversion. Phonetic transcriptions and prosody information together make up the symbolic linguistic representation that is output by the front-end. The back-end—often referred to as the *synthesizer*—then converts the symbolic linguistic representation into sound. In certain systems, this part includes the computation of the *target prosody* (pitch contour, phoneme durations), which is then imposed on the output speech.

3.18 File Upload

in computer networks, to upload is to send data to a remote system such as a server or another client so that the remote system can store a copy.

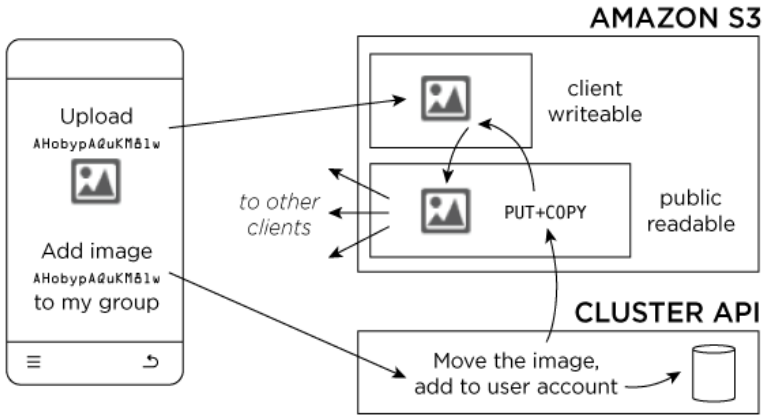


Figure 3.14 sample file upload architecture

CHAPTER4
Automed Software, Hardware Design and Architecture

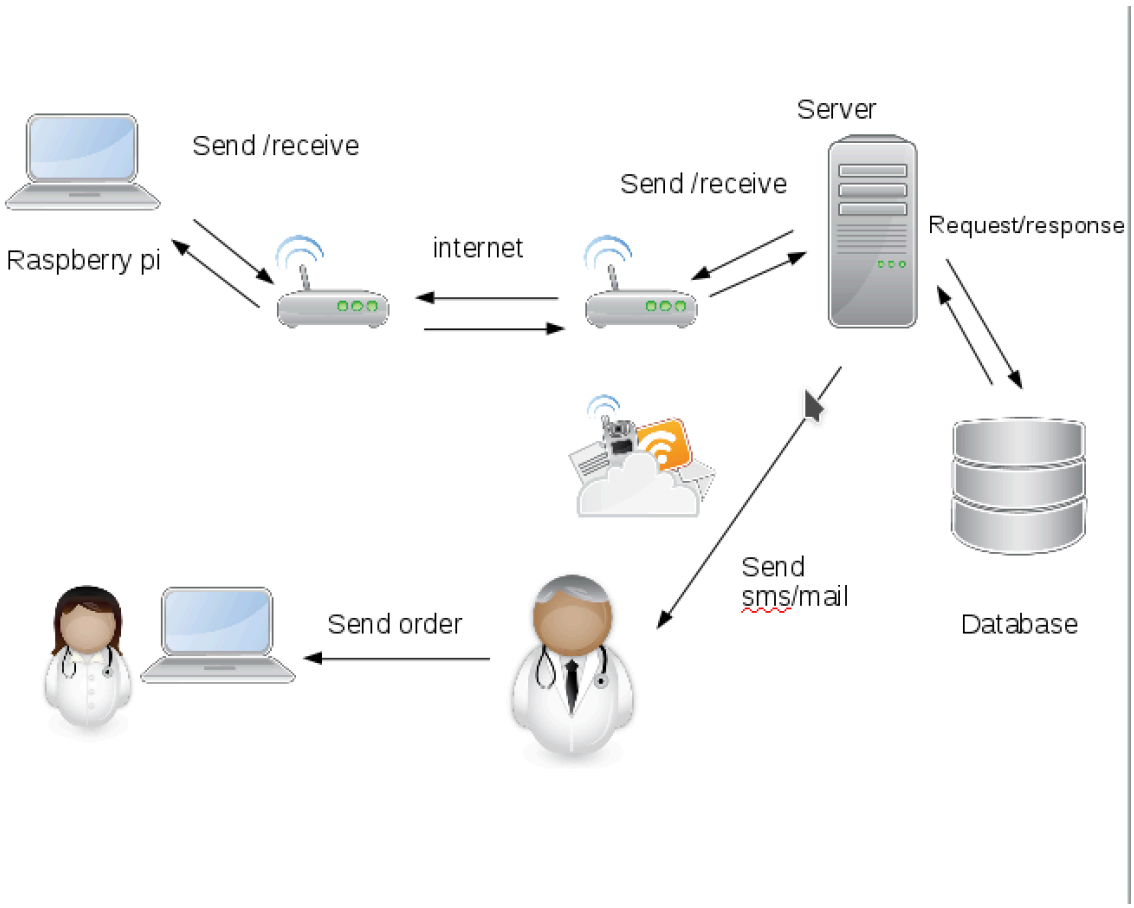


Figure 4.1 Automed project schema

4.1 MVC architecture

Model View Controller (MVC) is a software architecture pattern, commonly used to implement user interfaces: it is therefore a popular choice for architecting web apps. In general, it separates out the application logic into three separate parts, promoting modularity and ease of collaboration and reuse. It also makes applications more flexible and welcoming to iterations.

To make this a little more clear, let's imagine a simple shopping list app. All we want is a list of the name, quantity and price of each item we need to buy this

week. Below we'll describe how we could implement some of this functionality using MVC.

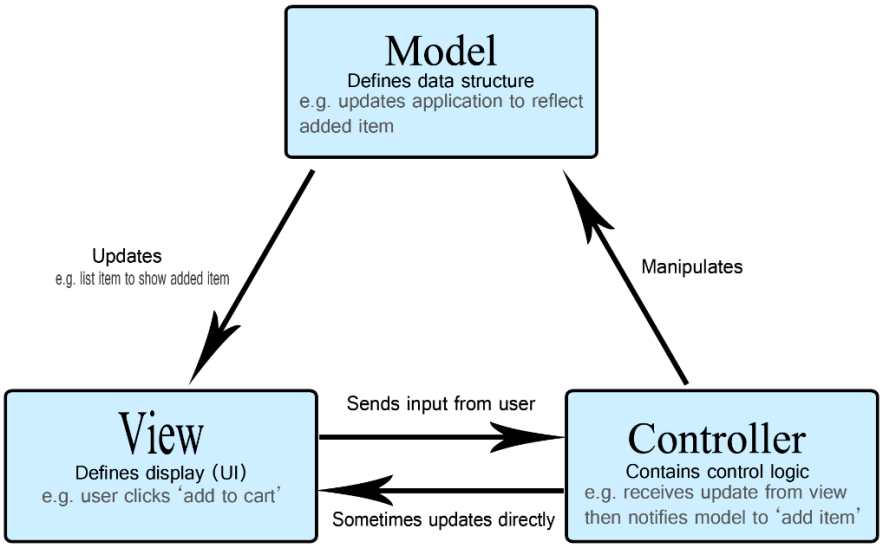


Figure 4.2 Automed project schema

4.1.1 The Model

The model defines what data the app should contain. If the state of this data changes, then the model will usually notify the view (so the display can change as needed) and sometimes the controller (if different logic is needed to control the updated view).

Going back to our shopping list app, the model would specify what data the list items should contain — item, price, etc. — and what list items are already present.

4.1.2 The View

The view defines how the app's data should be displayed. In our shopping list app, the view would define how the list is presented to the user, and receive the data to display from the model.

4.1.3 The Controller

The controller contains logic that updates the model and/or view in response to input from the users of the app.

So for example, our shopping list could have input forms and buttons that allow us to add or delete items. These actions require the model to be updated, so the input is sent to the controller, which then manipulates the model as appropriate, which then sends updated data to the view. You might however also want to just update the view to display the data in a different format, e.g., change the item order to alphabetical, or lowest to highest price. In this case the controller could handle this directly without needing to update the model.

4.1.4 Evolution of MVC on the web

As a web developer, this pattern will probably be quite familiar even if you've never consciously used it before. Your data model is probably contained in some kind of database (be it a traditional server-side database like MySQL, or a client-side solution such as IndexedDB [en-US].) Your app's controlling code is probably written in HTML/JavaScript, and your user interface is probably written using HTML/CSS/whatever else you like. This sounds very much like MVC, but MVC makes these components follow a more rigid pattern.

In the early days of the Web, MVC architecture was mostly implemented on the server-side, with the client requesting updates via forms or links, and receiving updated views back to display in the browser. However, these days, more of the logic is pushed to the client with the advent of client-side data stores, and XMLHttp Request allowing partial page updates as required.

4.2 Service-Oriented Architecture

A service-oriented architecture is essentially a collection of services. These services communicate with each other. The communication can involve either simple data passing or it could involve two or more services coordinating some activity. Some means of connecting services to each other is needed.

Service-oriented architectures are not a new thing. The first service-oriented architecture for many people in the past was with the use DCOM or Object Request Brokers (ORBs) based on the CORBA specification. For more on DCOM and CORBA,

4.2.1 Services

If a service-oriented architecture is to be effective, we need a clear understanding of the term service. A service is a function that is well-defined, self-contained, and does not depend on the context or state of other services.

4.2.2 Connections

The technology of Web Services is the most likely connection technology of service-oriented architectures. The following figure illustrates a basic service-oriented architecture. It shows a service consumer at the right sending a service request message to a service provider at the left. The service provider returns a response message to the service consumer. The request and subsequent response connections are defined in some way that is understandable to both the service consumer and service provider. How those connections are defined is explained in Web Services Explained. A service provider can also be a service consumer.

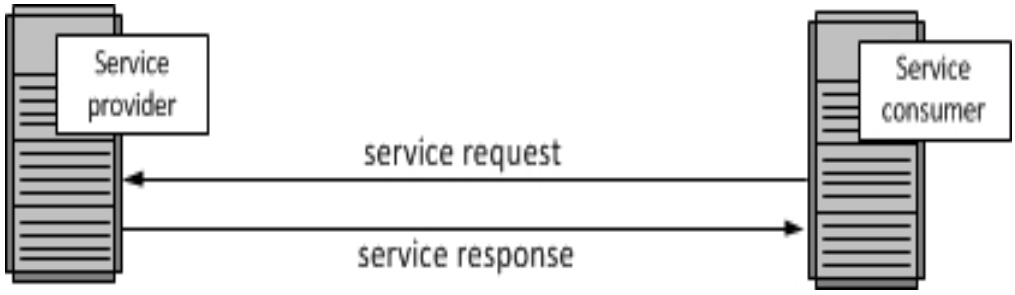


Figure 4.3 service oriented architecture

4.3 CORBA

CORBA is the acronym for Common Object Request Broker Architecture. It was developed under the auspices of the Object Management Group (OMG). It is middleware. A CORBA-based program from any vendor, on almost any computer, operating system, programming language, and network, can interoperate with a CORBA-based program from the same or another vendor, on almost any other computer, operating system, programming language, and network.

The first service-oriented architecture for many people in the past was with the use of Object Request Brokers (ORBs) based on the CORBA specification. The CORBA specification is responsible for really increasing the awareness of service-oriented architectures. CORBA enables communication between software written in different languages and running on different computers. Implementation details from specific operating systems, programming languages, and hardware platforms are all removed from the responsibility of developers who use CORBA. CORBA normalizes the method-call semantics between application objects residing either in the same address-space (application) or in remote address-spaces (same host, or remote host on a network). Version 1.0 was released in October 1991.

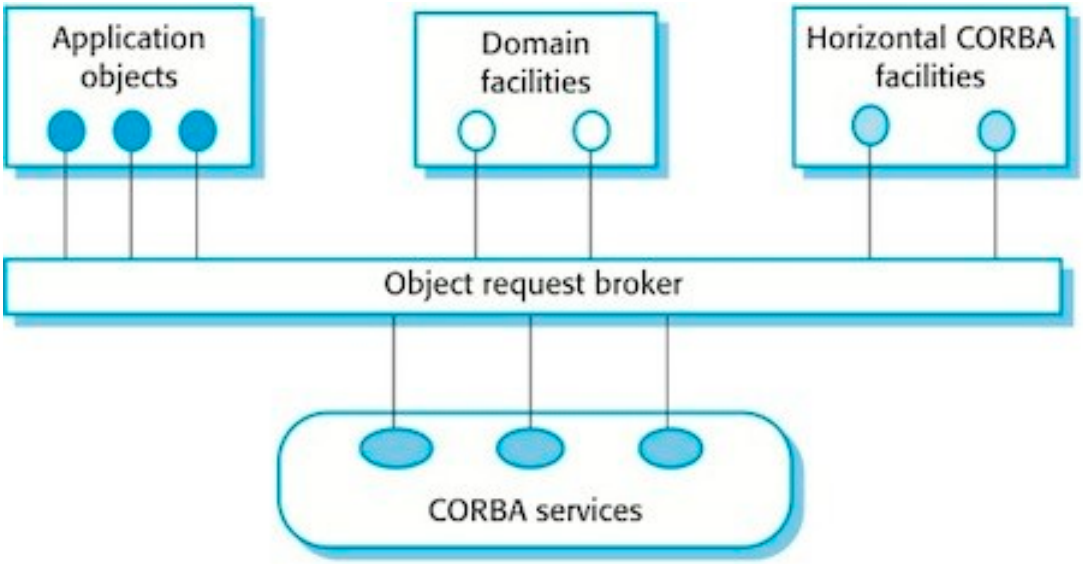


Figure 4.3 corba architecture

4.4 DCOM

DCOM is the acronym for the Distributed Component Object Model, an extension of the Component Object Model (COM). DCOM was introduced in 1996 and is designed for use across multiple network transports, including Internet protocols such as HTTP. DCOM is based on the Open Software Foundation's DCE-RPC spec and will work with both Java applets and ActiveX components through its use of the Component Object Model (COM). It works primarily with Microsoft Windows.

4.5 Web service

The term *web service* is either

- (generic) a service offered by an electronic device to another electronic device, communicating with each other via the World Wide Web, or
- (specific) a web service implemented in the particular technology or brand, W3C Web Services.

In a web service, the Web technology such as HTTP—originally designed for human-to-machine communication—is utilized for machine-to-machine communication, more specifically for transferring machine-readable file formats such as XML and JSON.

In practice, a web service commonly provides an object-oriented web-based interface to a database server, utilized for example by another web server, or by a mobile app, that provides a user interface to the end user. Many organizations that

provide data in formatted HTML pages will also provide that data on their server as XML or JSON, often through a web service to allow syndication, for example Wikipedia's Export. Another application offered to the end user may be a mashup, where a web server consumes several web services at different machines, and compiles the content into one user interface.

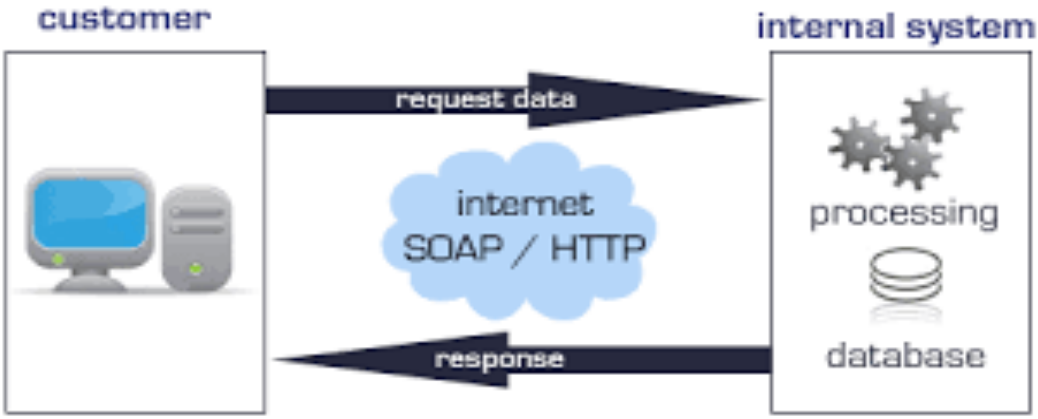


Figure 4.4 web service architecture

4.6 Servlet Architecture

A Servlet is a class, which implements the javax. servlet.Servlet interface. However instead of directly implementing the javax. servlet.Servlet interface we extend a class that has implemented the interface like javax.servlet.GenericServlet or javax.servlet.http.HttpServlet.

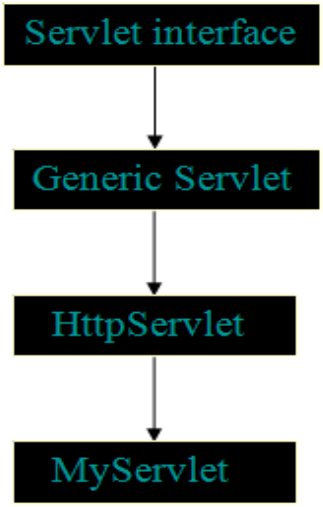


Figure 4.5 servlet architecture

4.6.1 Servlet Exceution

This 67 show a servlet execution takes place when client (browser) makes a request to the webserver.

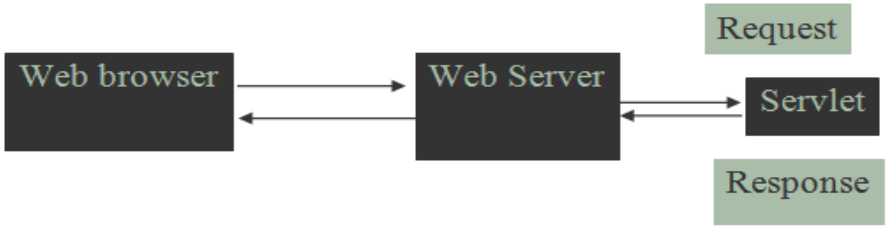


Figure 4.6 Servlet Exceution

4.7 raspberry pi hardware

The Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools and in developing countries. The original model became far more popular than anticipated, selling outside its target market for uses such as robotics. It does not include peripherals (such as keyboards, mice and cases). However, some accessories have been included in several official and unofficial bundles.

According to the Raspberry Pi Foundation, over 5 million Raspberry Pis were sold by February 2015, making it the best-selling British computer. By November 2016 they had sold 11 million units, and 12.5m by March 2017, making it the third best-selling "general purpose computer" In July 2017, sales reached nearly 15 million. In March 2018, sales reached 19 million.

Most Pis are made in a Sony factory in Pencoed, Wales; some are made in China or Japan. Processor speed ranges from 700 MHz to 1.4 GHz for the Pi 3 Model B+; on-board memory ranges from 256 MB to 1 GB RAM. Secure Digital (SD) cards are used to store the operating system and program memory in either SDHC or MicroSDHC sizes. The boards have one to four USB ports. For video output, HDMI and composite video are supported, with a standard 3.5 mm phono jack for audio output. Lower-level output is provided by a number of GPIO pins which support common protocols like I²C. The B-models have an 8P8C Ethernet port and the Pi 3 and Pi Zero W have on-board Wi-Fi 802.11n and Bluetooth. Prices range from US$5 to $35.

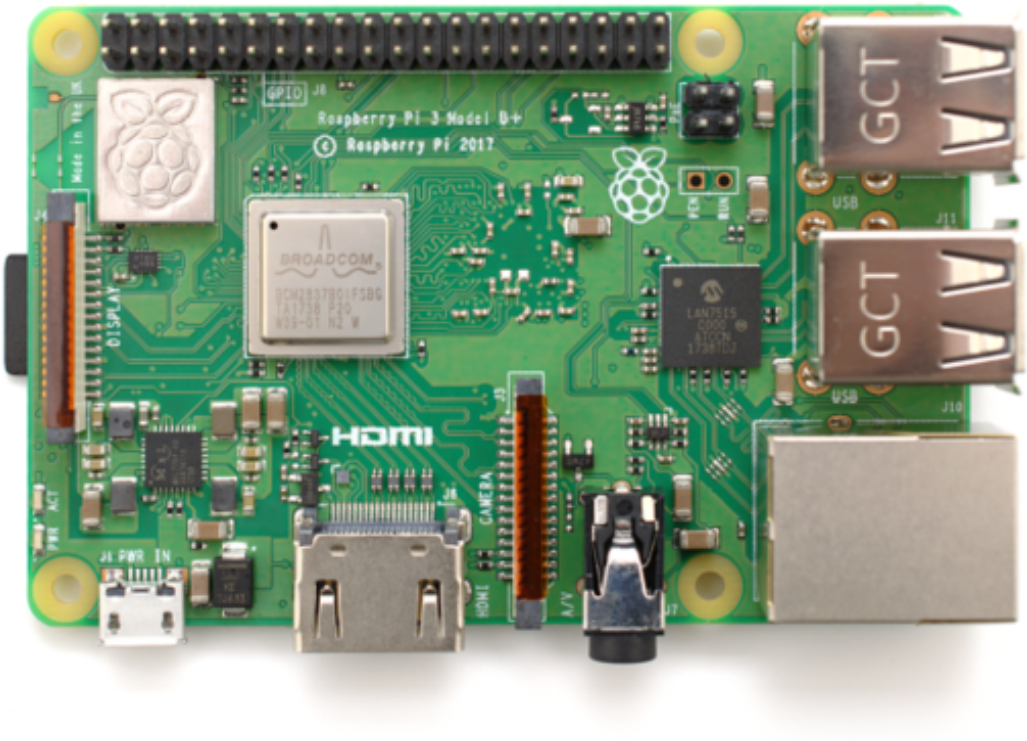
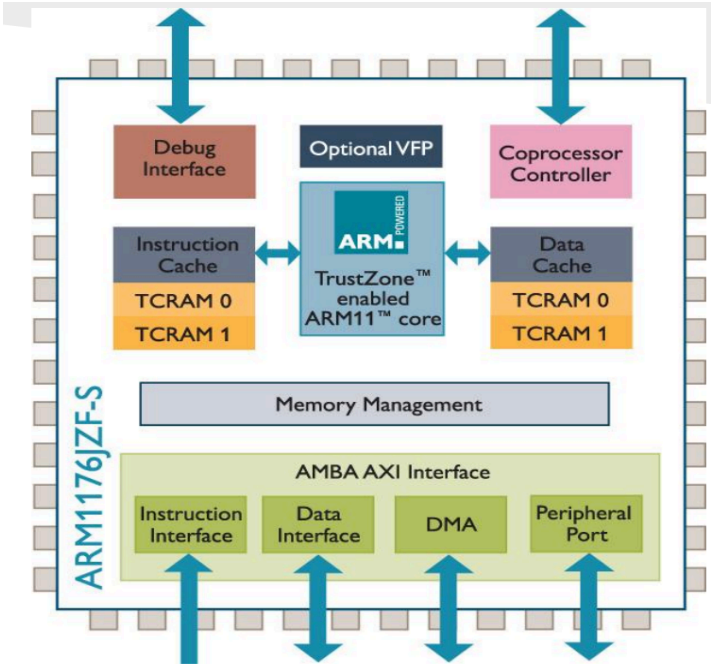


Figure 4.7 raspberry pi as hardware side of automed system



ARM 1176 Processor

Figure 4.8 raspberry pi internal architecture

4.8 Raspberry pi camera

The Raspberry Pi Camera Module is an official product from the Raspberry Pi Foundation. The original 5-megapixel model was released in 2013, and an 8-megapixel Camera Module v2 was released in 2016. For both iterations, there are visible light and infrared versions.

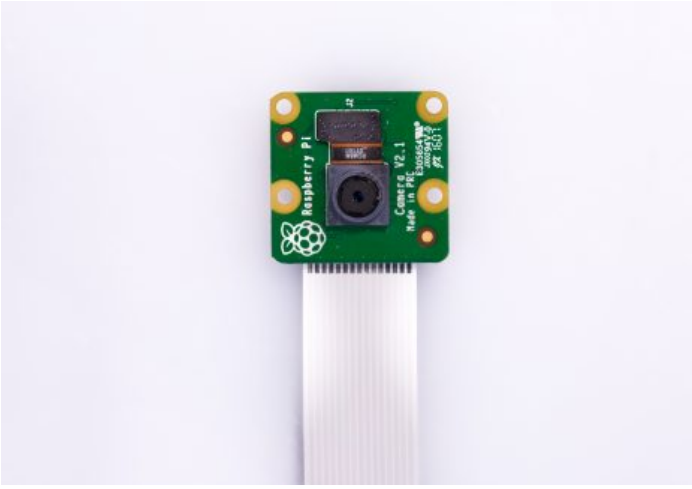


Figure 4.9 raspberry pi camera

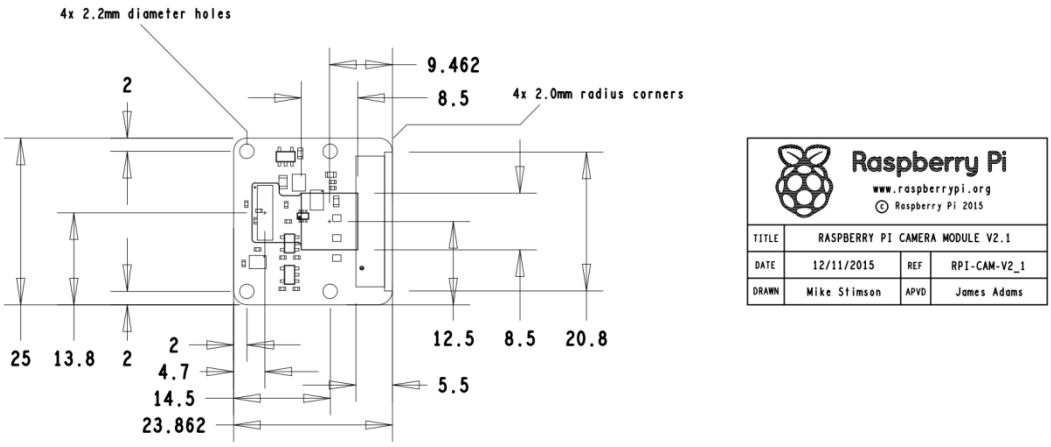


Figure 4.10 raspberry pi camera architecture

Figure 4.11 automed hardware connections

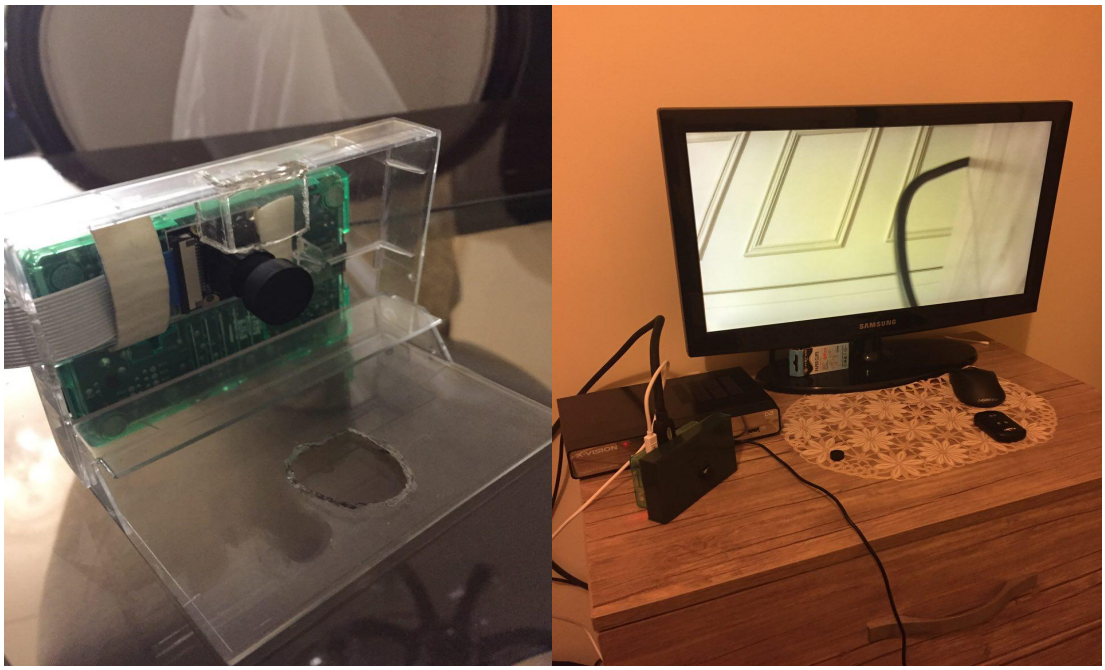


Figure 4.12 automed hardware patient monitor

CHAPTER5
Software capabilities and features

5.1 Send email

This software send email to doctor, nurse and patient in case of nesecery informations such as make appointment, add new patient to system and send alarm to doctor.

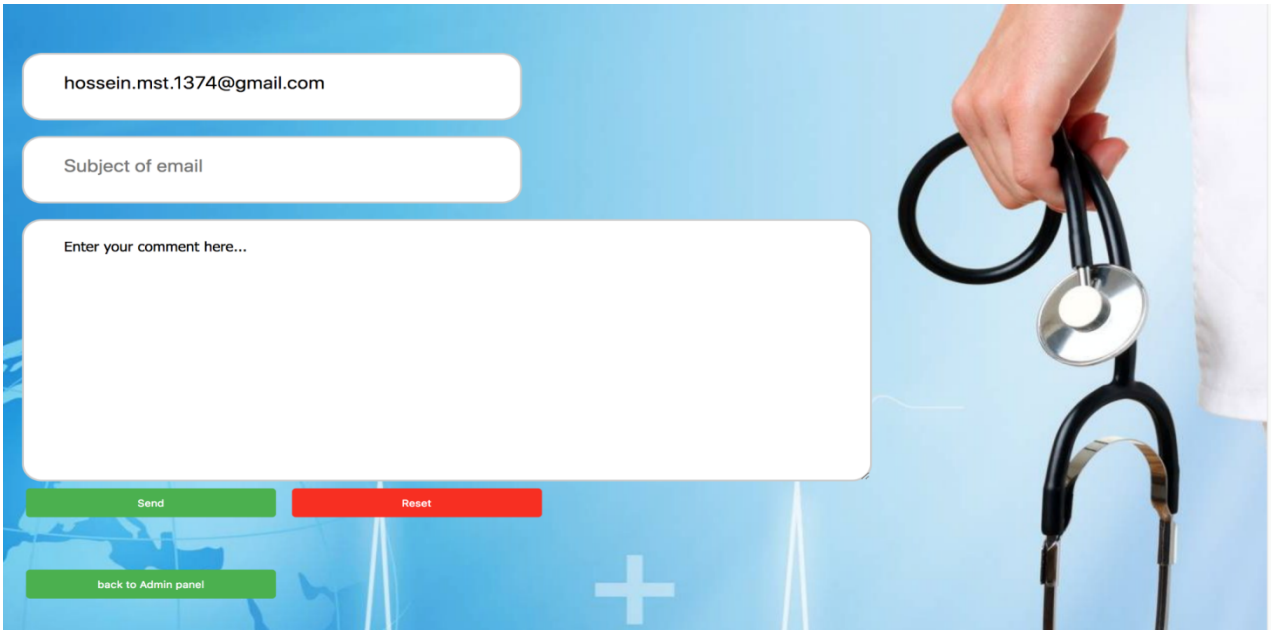


Figure 5.1 automed system send mail

5.2 Send Sms

This software sends sms to doctor, nurse and patient in case of nesecery informations such as make appointment, add new patient to system and send alarm to doctor. System use multi factor authentication to verify admin user login. Automed medical system use nextmo API which offers good platform so sending sms in distributed system such as bank system, medical system, insurance system and more systems with high number of sms in a single day.

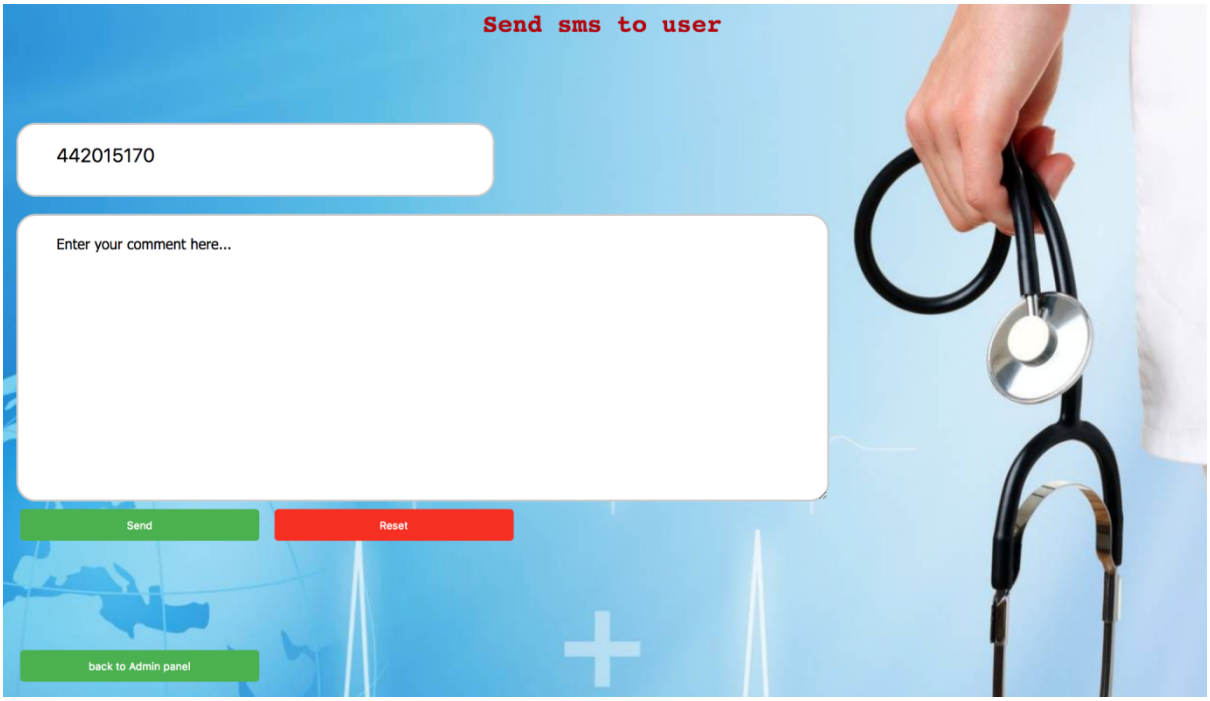


Figure 5.2 automed system send sms

5.3 Send notification

This software sends notification to users. admin can send notification to doctor and nurse panel. admin must first search user and in search result admin can send notification to user panel. when users' login to system, he/she can see all list of notification from admin.

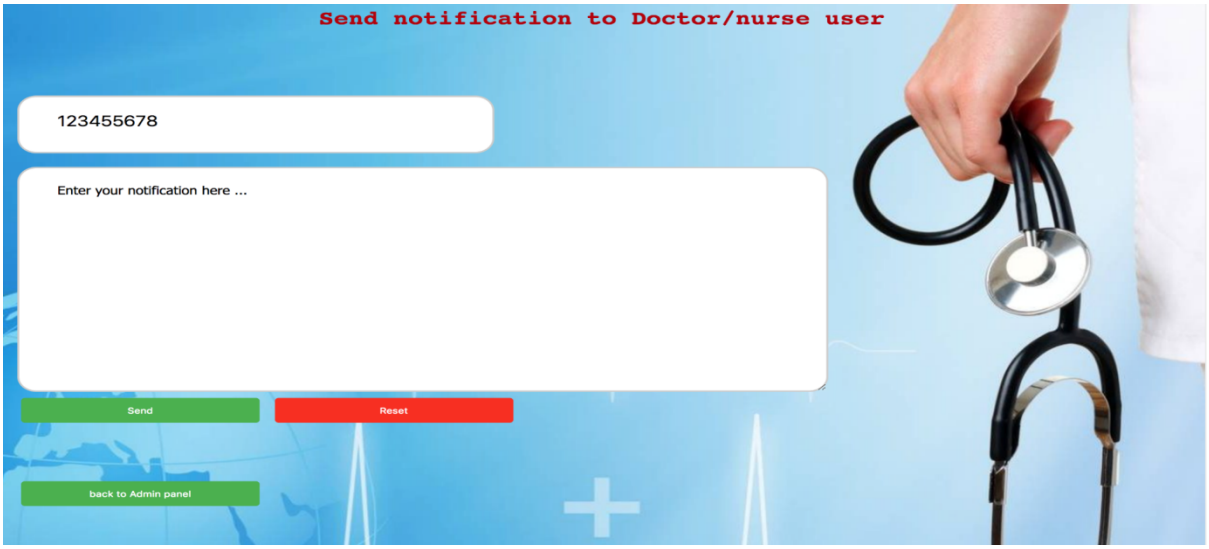


Figure 5.3 automed system send sms

5.4 Print informations

automed system can print all data from system as below:

- print patient diagnosis
- patient prescription
- print user data login
- print system data reports such as total number of doctor, nurs
- List of all user login logs
- list of all doctors with information
- list of connected devices ip
- list of system user logs
- list of online users

Figure 5.4 automed system print patient prescription

5.5 data statistics

Automed system shows system data statistics.

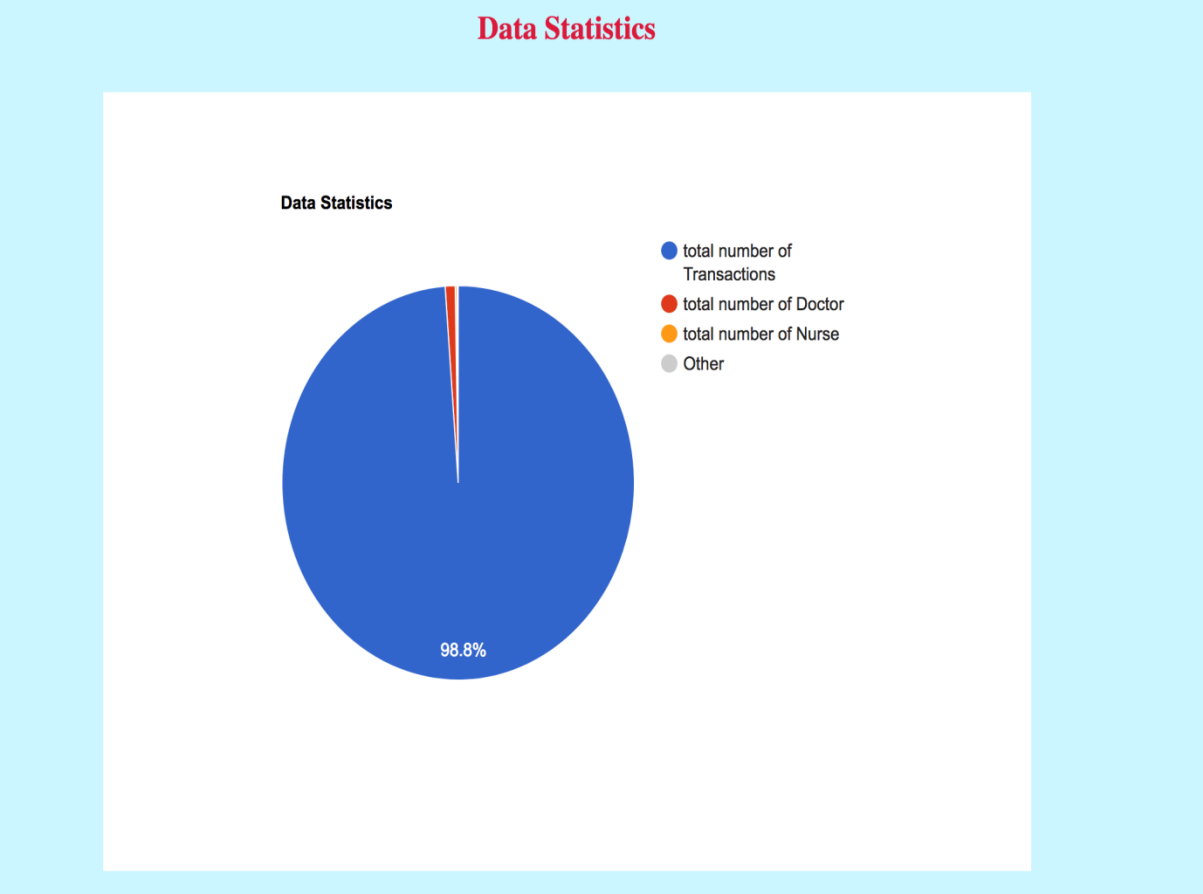


Figure 5.5 automed system data statistics

5.6 Server status

Automed admin panel shows status of server with all server informations. This section of admin panel in automed system shows all infotmation related to server status such as

- database status: indicate database is running or not
- server status: indicate server is running or no it means application is ready to serve
- apache status: indicate the status of apache where application is deployed
- online users: show total number of online users whos are currently working with system
- swap memory usage: shows total number of ram in MB which JVM is used.

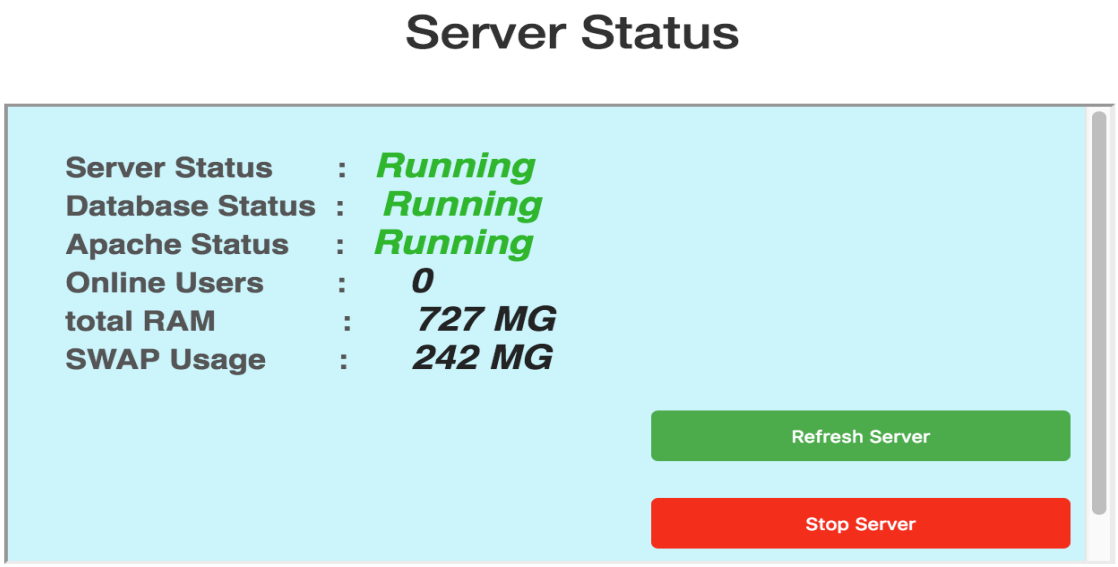


Figure 5.6 automed system server status

5.7 text to speech

Doctor or nurse can listen to result of examination and prescription and diagnosis by using automed system. also, user can see uploaded image or file of examinations by using upload center.

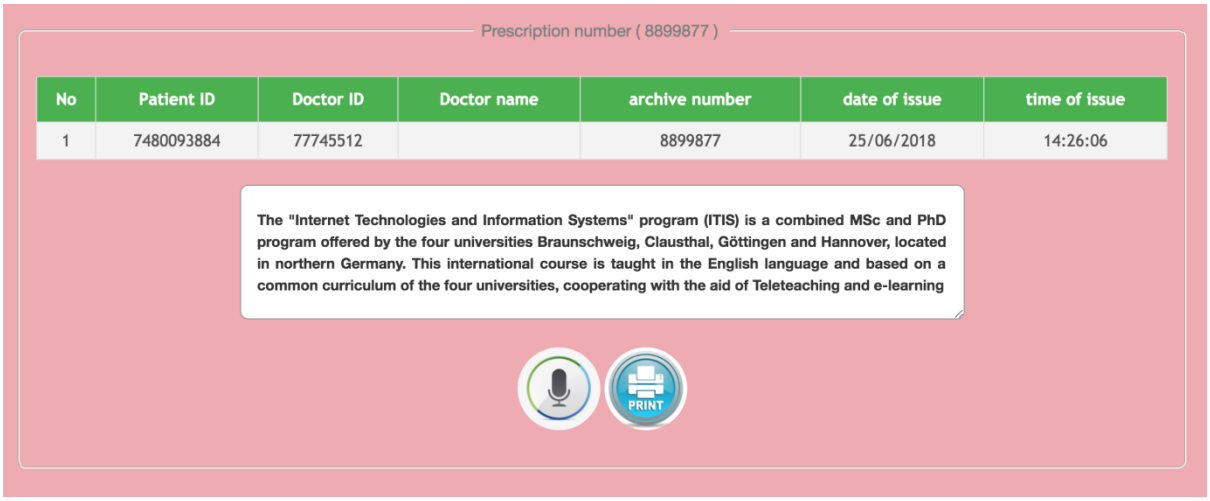


Figure 5.7 automed text to speech system

5.8 Speech to text

Doctor can use voice to create prescription and diagnosis result instead of typing it in textbox.

System support multiple languages to convert speech to text and store in to our database.

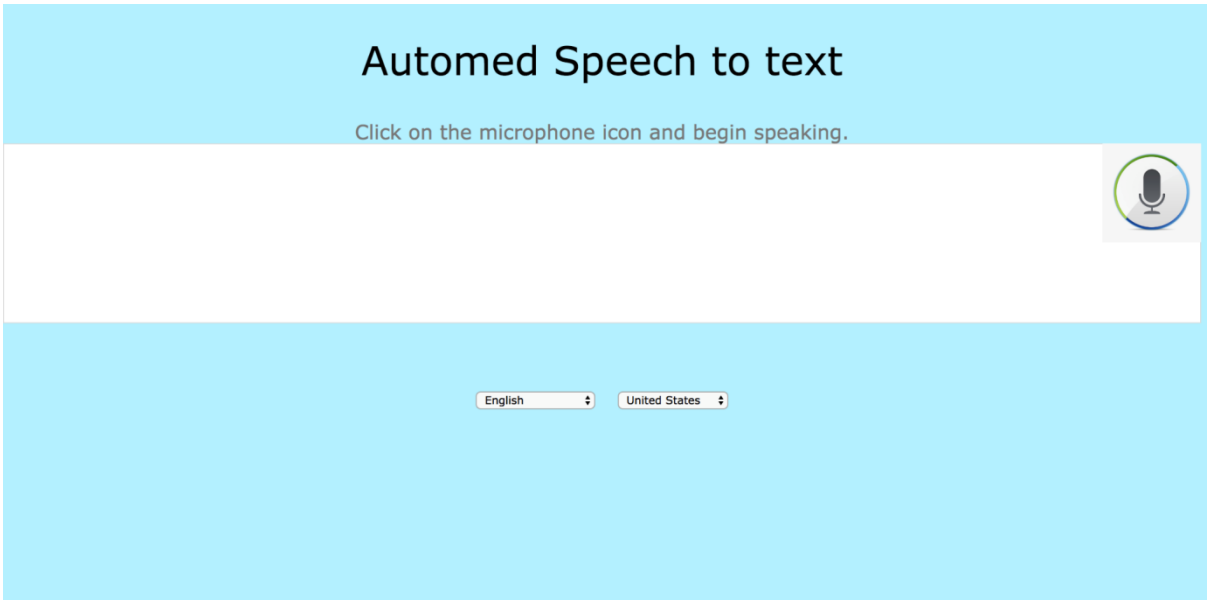


Figure 5.8 automed speech to text system

5.9 IOS mobile app for doctor

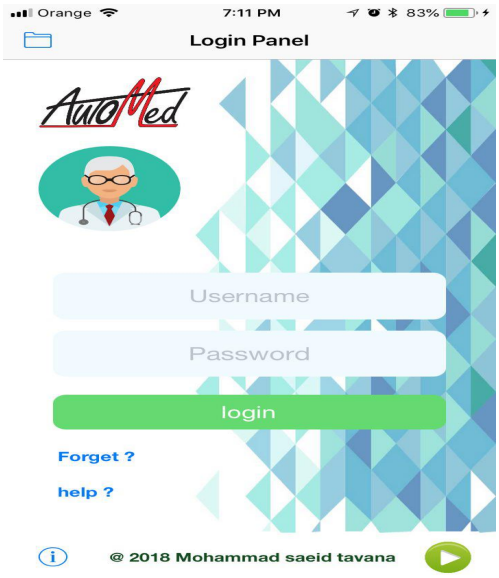


Figure 5.9 automed IOS app

CHAPTER6

Software security and user data protection

6.1 java standard security realms

A *realm* is a security policy domain defined for a web or application server. A realm contains a collection of users, who may or may not be assigned to a group. Managing users on the GlassFish Server is discussed in Managing Users and Groups on the GlassFish Server.

An application will often prompt for a user name and password before allowing access to a protected resource. After the user name and password have been entered, that information is passed to the server, which either authenticates the user and sends the protected resource or does not authenticate the user, in which case access to the protected resource is denied. This type of user authentication is discussed in Specifying an Authentication Mechanism in the Deployment Descriptor.

In some applications, authorized users are assigned to roles. In this situation, the role assigned to the user in the application must be mapped to a principal or group defined on the application server. Figure 24–6 shows this. More information on mapping roles to users and groups can be found in Setting Up Security Roles.

The following sections provide more information on realms, users, groups, and roles.

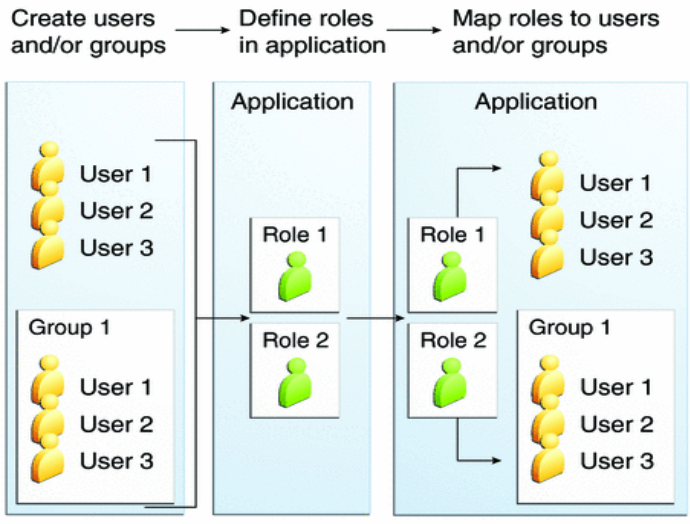


Figure 24–6 Mapping Roles to Users and Groups

What Is a Realm?

A realm is a security policy domain defined for a web or application server. The protected resources on a server can be partitioned into a set of protection spaces, each with its own authentication scheme and/or authorization database containing a collection of users and groups. For a web application, a realm is a complete database of users and groups identified as valid users of a web application or a set of web applications and controlled by the same authentication policy.

The Java EE server authentication service can govern users in multiple realms. The file, admin-realm, and certificate realms come preconfigured for the GlassFish Server.

In the file realm, the server stores user credentials locally in a file named keyfile. You can use the Administration Console to manage users in the file realm. When using the file realm, the server authentication service verifies user identity by checking the file realm. This realm is used for the authentication of all clients except for web browser clients that use HTTPS and certificates.

In the certificate realm, the server stores user credentials in a certificate database. When using the certificate realm, the server uses certificates with HTTPS to authenticate web clients. To verify the identity of a user in the certificate realm, the authentication service verifies an X.509 certificate. For step-by-step instructions for creating this type of certificate,

The admin-realm is also a file realm and stores administrator user credentials locally in a file named admin-keyfile. You can use the Administration Console to manage users in this realm in the same way you manage users in the file realm.

What Is a User?

A *user* is an individual or application program identity that has been defined in the GlassFish Server. In a web application, a user can have associated with that identify a set of roles that entitle the user to access all resources protected by those roles. Users can be associated with a group.

A Java EE user is similar to an operating system user. Typically, both types of users represent people. However, these two types of users are not the same. The Java EE server authentication service has no knowledge of the user name and password you provide when you log in to the operating system. The Java EE server authentication service is not connected to the security mechanism of the operating system. The two security services manage users that belong to different realms.

What Is a Group?

A *group* is a set of authenticated users, classified by common traits, defined in the GlassFish Server. A Java EE user of the file realm can belong to a group on the GlassFish Server. (A user in the certificate realm cannot.) A group on the GlassFish Server is a category of users classified by common traits, such as job title or customer profile. For example, most customers of an e-commerce application might belong to the CUSTOMER group, but the big spenders would belong to the PREFERRED group. Categorizing users into groups makes it easier to control the access of large numbers of users.

A group on the GlassFish Server has a different scope from a role. A group is designated for the entire GlassFish Server, whereas a role is associated only with a specific application in the GlassFish Server.

What Is a Role?

A *role* is an abstract name for the permission to access a particular set of resources in an application. A role can be compared to a key that can open a lock. Many people might have a copy of the key. The lock doesn't care who you are, only that you have the right key.

Some Other Terminology

The following terminology is also used to describe the security requirements of the Java EE platform:

- **Principal**: An entity that can be authenticated by an authentication protocol in a security service that is deployed in an enterprise. A principal is identified by using a principal name and authenticated by using authentication data.
- **Security policy domain**, also known as **security domain** or **realm**: A scope over which a common security policy is defined and enforced by the security administrator of the security service.
- **Security attributes**: A set of attributes associated with every principal. The security attributes have many uses: for example, access to protected resources and auditing of users. Security attributes can be associated with a principal by an authentication protocol.
- **Credential**: An object that contains or references security attributes used to authenticate a principal for Java EE services. A principal acquires a credential upon authentication or from another principal that allows its credential to be used.

6.2 md5 algorithm

The **MD5 algorithm** is a widely used hash function producing a 128-bit hash value. Although MD5 was initially designed to be used as a cryptographic hash function, it has been found to suffer from extensive vulnerabilities. It can still be used as a checksum to verify data integrity, but only against unintentional corruption.

One basic requirement of any cryptographic hash function is that it should be computationally infeasible to find two non-identical messages which hash to the same value. MD5 fails this requirement catastrophically; such collisions can be found in seconds on an ordinary home computer.

The weaknesses of MD5 have been exploited in the field, most infamously by the Flame malware in 2012. The CMU Software Engineering Institute considers MD5 essentially "cryptographically broken and unsuitable for further use".

MD5 was designed by Ronald Rivest in 1991 to replace an earlier hash function MD4. The source code in RFC 1321 contains a "by attribution" RSA license. The abbreviation "MD" stands for "Message Digest."

6.3 SHA 2

SHA-2 (**Secure Hash Algorithm 2**) is a set of cryptographic hash functions designed by the United States National Security Agency (NSA). They are built using the Merkle–Damgård structure, from a one-way compression functionitself built using the Davies-Meyer structure from a (classified) specialized block cipher.

Cryptographic hash functions are mathematical operations run on digital data; by comparing the computed "hash" (the output from execution of the algorithm) to a known and expected hash value, a person can determine the data's integrity. For example, computing the hash of a downloaded file and comparing the result to a previously published hash result can show whether the download has been modified or tampered with. A key aspect of cryptographic hash functions is their collision resistance: nobody should be able to find two different input values that result in the same hash output.

SHA-2 includes significant changes from its predecessor, SHA-1. The SHA-2 family consists of six hash functions with digests (hash values) that are 224, 256, 384 or 512 bits: **SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256**.

SHA-256 and SHA-512 are novel hash functions computed with 32-bit and 64-bit words, respectively. They use different shift amounts and additive constants, but their structures are otherwise virtually identical, differing only in the number of rounds. SHA-224 and SHA-384 are simply truncated versions of the first two, computed with different initial values. SHA-512/224 and SHA-512/256 are also

truncated versions of SHA-512, but the initial values are generated using the method described in Federal Information Processing Standards (FIPS) PUB 180-4. SHA-2 was published in 2001 by the National Institute of Standards and Technology (NIST) a U.S. federal standard (FIPS). The SHA-2 family of algorithms are patented in US patent 6829355. The United States has released the patent under a royalty-free license.

Currently, the best public attacks break preimage resistance for 52 out of 64 rounds of SHA-256 or 57 out of 80 rounds of SHA-512, and collision resistance for 46 out of 64 rounds of SHA-256.

SHA-256 and SHA-512, and, to a lesser degree, SHA-224 and SHA-384 are prone to length extension attacks, rendering it insecure for some applications. It is thus generally recommended to switch to SHA-3 for 512 bit hashes and to use SHA-512/224 and SHA-512/256 instead of SHA-224 and SHA-256. This also happens to be faster than SHA-224 and SHA-256 on x86-64, since SHA-512 works on 64 bit instead of 32 bit words.

Conclusion

With automed system which is modern EHRs (Electronic Health Records), hospitals and clinics can provide hight quality of health service to patient. This enterprise E-health application covers all aspects of patient health. with using of technology in hospitals and clinics, patient life would be saved and procedures of admission and other task in hospital can be done as fast as possible. Also, this system is usefull for save paper because this system reduces paper work in hospitals and clinics. Also, this system can serve hight quality of health services to patient very fast. last but not least is patient log history which is very usefull in Forensic Medicine. Hope such systems grows fast in every country to provide good quality of health service to patient.

References and web links

Project book references:

[1] J. Graba," Remote Method Invocation, java database conectivity" in introduction to Network programming in java, south torkshire, UK, Sheffield hallam university, extera pub,2013

[2] V. Sarcar," singleton pattern, factory pattern, builder pattern" in java design patterns, New York, USA, Appress pub, 2015

[3] H. Deitel and p. Deitel, "java programming "in Java daitel & daitel 10th edition, California, USA, deitel pub, 2014

[4] M. weiss," sorting" in data structures and algorithm analysis in java 3rd edition, florida, USA, florida international university, pearson pub 2011,

[5] A. B. Downey and C. Mayfield, Think Java, California, USA, orelliy pub ,2016

[6] R. S. Pressman," modeling, the software process, agile development" in Software engineering Apractitioner s Approach 7th edition, New York, NY 10020, USA, McGraw-hill companies' Inc ,2010

[7] Basics of object –oriented programming, relation between classes"in Object-Oriented Analysis, Design and Implementation, 2nd Edition, New York, USA, springer pub, 2016

[8] R. Wootton, J. Craig, Introduction to telemedicine, London: Royal Soc. Medicine Press, 1999.

[9] eHealth 2003 High Level Conference*, 22 May 2003.*

[10] G. Eysenbach, "What is e-health?", J Med Internet Res, vol. 3, no. 2, pp. e20, 2001

[11] F. Lievens, M. Jordanova, "An approach to the global vision about Telemedicine/eHealth" in Keynote presentation Med-e-Tel, Luxembourg: pp. 21-23, April 2004

[12] S. Schug, "eHealth and telemedicine for sustainable health services across Europe", *II. SEE Conf.*, 2 2011.

[13] A. Boyle, "Five-Tech forecast", *Retrieved*, September 2009

[14] R. A. Clark, S. C. Inglis, F. McAlister, J. G. F. Cleland, S. Stewart, "Telemonitoring or structured telephone support programmes for patients with chronic heart failure: systematic review and metaanalysis", *British Medical Journal*, vol. 334, pp. 942-953, 2007.

[15] G. Pare, M. Jaana, C. Sicotte, "Systematic review of home telemonitoring for chronic diseases the evidence base", *Journal of the American Medical Informatics Association*, vol. 14, pp. 269-277, 2007.

[16] R. Hofmann-Wellenhof, W. Salmhofer, B. Binder, A. Okcu, H. Ked, H. Soyer, "Feasibility and acceptance of telemedicine for wound care in patients with chronic leg ulcers", *Journal of Telemedicine and Telecare*, vol. 12, no. 1, pp. 15-17, 2006.

[17] D. Mucic, "Telepsychiatry in Denmark: Mental health care in rural and remote areas", *J. of eHealth Technology and Application*, vol. 5, no. 3, pp. 277-281, 2007.

[18] D. Mucic, M. Jordanova, F. Lievens, "International telepsychiatry patient acceptability" in Global Telemedicine/ellealth Updates: Knowledge Resources, Luxembourg: Publ. Luxexpo, vol. 1, pp. 383-384, 2008.

[19] M. Stojakovic, M. Jordanova, F. Lievens, "Compare psychiatric consultation face-to-face and telepsychiatry", *Global telemedicine/ellealth Updates: Knowledge Resources*, vol. 4, pp. 130-134, 2011.
[20] A. Giannone, P. P. Visentin, M. Jordanova, F. Lievens, "Monitoring treatment and prediction of bipolar disorder episodes MONARCA" in Global telemedicine/elIealth Updates: Knowledge Resources, Luxembourg: Publ. ISfTeH, vol. 4, pp. 199-203, 2011.
[21] M. Jordanova, L. Vasileva, A. Vladimirova, A. Gencheva, S. Shtereva-Katsarova, B. Krendeva, M. Rasheva, R. Bojinova, "Telepsychology: Lessons learned from 4 years of experience", *Journal of eHealth Technology and Application*, vol. 7, no. 2, pp. 105-108, 2009.
[22] M. Nayeemuddin, M. A. Majeed, A. Muneer, A. Misra, "An outpatient survey of plastic surgery patients" in paper presented at Med-e-Tel 2007, Luxembourg: G.D. of Luxembourg, Retrieved, no. 21, January 2009
[23] B. von Niman, "User experience guidelines for eHealth telecare services — ETSI Industry consensus workshop", *Med-e-Tel 2007 Luxembourg G.D. of Luxembourg Retrieved*.
[24] G. Bestente, A. Frisiello, F. Scullino, M. Jordanova, F. Lievens et al., "Affordable location-based services (LBS) to assist Alzheimer's disease patients" in Global telemedicine/ellealth Updates: Knowledge Resources, Luxembourg: Publ. ISfTeH, vol. 4, pp. 120-125, 2011
[25] M. Jordanova, F. Lievens, "Chapter in global Telemedicine and eHealth Updates" in Knowledge Resources, Luxembourg: Publ. ISfTeH, vol. 4, pp. 269-297, 2011
[26] ITU-D Study Group 2 4th Study Period International Telecommunication Union Telecommunication Development Bureau, Switzerland: May 2011
[27] R. M. Figueira, M. B. M. Alkmim, M. P. Abreu, M. Jordanova, F. Lievens et al., "Operational costs in a large-scale telehealth service" in Global Telemedicine and eHealth Updates:
[28] M. B. Alkmim, M. C. Nunes, J. C. Jesus, M. Jordanova, F. Lievens et al., "Success factors for telehealth service implementation" in Global Telemedicine and eHealth Updates: Knowledge Resources, Luxembourg: Publ. ISfTeH, vol. 4, pp. 102-107, 2011.
[29] M. Jordanova, Stefane M. Kabene, "Closing the gap: eHealth and optimization of patient care" in Healthcare and the Effect of Technology: Developments Challenges and Advancements, USA:IGI Global, pp. 38-59, 2010.
[30] M. Mars, R. E. Scott, "Global e-health policy: A work in progress", *Health Affairs*, vol. 29, no. 2, pp. 239-245, 2010.
[31] Core Standards for Telemedicine Operations American Telemedicine Association*, February 2008*
[32] Telehealth Practice Recommendations for Diabetic Retinopathy American Telemedicine Association*, May 2004*
[33] K. Belan, "Electronic health records will replace the paper records by 2011", *The Epoch Times Published*, October 2010

website references:

1- www.codeproject.com
2- www.oracle.com
3- www.stackoverflow.com
4- www.stachexchange.com
5- www.p30download.com
6- www.github.com
7- www.tutorialspoint.com
8- www.javaworld.com
9- www.dzone.com
10- www.wikipedia.com
11- www.coursera.org
12- www.apple.com
13- www.java.net
14- www.java2s.com
15- www.javaos.com
16- www.lynda.com
17- www.javacoffeebreak.com
18- www.javapoint.com
19- www.w3schools.com
20- www.mysql.com
21- www.msdn.microsoft.com
22- www.allitbooks.com
23- www.bookboon.com
24- www.onlineprogrammingbooks.com
25- www.sourcebaran.com
26- www.barnamenevice.com
27- www.maktabkhooneh.org
28- www.javacup.ir
29- www.docs.oracle.com
30- www.jetbrain.com
31- www.netbeans.com
32- www.planetcassandra.org
33- www.nosql-database.org
34- http://www.hpkclasses.ir
35- http://hibernate.org
36- https://www.apache.org
37- http://www.jboss.org
38- https://www.nexmo.com