**Chapter 1: Introduction**

In today’s world of increasing dependence on computers and computer systems, it is imperative that we be able to rely on secure and confidential connections to the computers. Traditionally, this has been by authentication with usernames and passwords. “A password is information associated with an entity that confirms the entity’s identity.” The password is a string of characters that can either be automatically generated by the system or selected by the entity or user. Passwords can range from a single character to passphrases, which can be hundreds of characters in length and be comprised of series of words and phrases. The goal of a password is to authenticate a user. It is a piece of information that the user knows. However, this is a weak method of authentication because users tend to generate passwords that are easy to remember but also easy to crack .Passwords provide a first line of defence in most cases, but there is much more. If someone can guess a user’s password, they can impersonate the user.

**1.1 Need**

**1.1.1 Ridiculously Easy Passwords**

The aspect of protecting data has been the primary concern for computer users all over especially those who are entrusted with sensitive data. Windows password policies can be invoked in different ways as deemed necessary by relevant authorities in an organization and it is therefore imperative that these be followed to the book to ensure that none of the data is violated. However, some of the policies are not enough to guarantee security over personal or organizational data, hence the need for other stronger or foolproof solutions to ensure protection. Having (and enforcing) a strong password policy is a basic step that is often overlooked. In random audits of corporate networks, about 30% of user passwords are ridiculously easy to guess and appear in any hacker's dictionary. Strict password policies can also be a double edged sword - make them too strong and your users will begin writing their passwords on sticky notes and keeping them in their desk drawers, under mouse pads, or taped to the bottom of the keyboard. They can also flood your help desk with requests to reset forgotten passwords.

**1.1.2 Windows Password Security**

The windows security feature has been the most important issue ever since the beginning of the windows systems. The password feature is tied to user accounts and users can create or modify accounts, if they have administrator privileges. The general drawback was that there was no guarantee that the set passwords are strong or perhaps follow some guideline t o make sure that they are secure enough. The existence of password policies made certain that the systems were protected using certain guidelines either set in the system itself or imposed by organizations so that users follow these rules to ensure that the systems were secure. Even then the general acceptance of these policies was not wide. Some policies became more and more forced through system configuration and some were based on the user abiding by them, for example, other policies require password change at the end of each work day or even once a week.

Password policy implementation in organization Philip Inglesant explored the use of password policies in organizations and also to understand more about specific problems faced by users in conforming to password policies, and the strategies which they adopt to cope with these issues. In their research they discovered that Conflict between password policies and the capabilities of users, and the problems this creates, the ways users find to cope with this conflict and the impact on security of different contexts of use. The results showed that users tend to feel that some policies imposed by their organizations are too difficult to follow. In the study, they realized that the idea of devising complex passwords was a direct interruption of user’s activities and also the need to change passwords after a short while. Even if users understand the importance of such a policy, the impositions themselves are a burden and will prompt the user to resist. In this light, a more versatile solution must be implemented that will ensure that the needs of the organizations and of the users are met. Users just do not want the burden of having to be alert when conforming to policies that require too much of them. Thus they end up creating simpler solutions and probably create simpler passwords

**Chapter 2: Review of Literature**

Philip Iglesant (2010) says in his paper, Despite a growing number of graphical and biometric authentication mechanisms, passwords remain the most familiar and commonly-used form of user authentication in organisational settings. In this paper, we investigate the impact of passwords, and their associated security policies, on individual users’ productivity and experience. Password policies govern not only construction and lifetime of individual passwords, but also work with other contextual factors to define the numbers of passwords users are expected to remember and the frequency with which they have to use them.

Over 10 years ago, Adams & Sasse [1] found that password policies that do not meet users’ work practices caused high levels of dissatisfaction, and led to insecure practices and low security motivation. Since then, we have seen studies of passwords using controlled [2] and survey methodologies [4], which provide some understanding of the policy factors that make passwords easier to generate, remember, and use, in ways which are appropriate to the situation. So - has this understanding been applied in practice? Has anything changed?

The study in the paper highlighted the contextual issues of passwords:

1. What specific aspects of password policy cause problems for users?

2. What coping strategies do password users adopt to overcome those problems?

3. How do those coping strategies affect productivity, the security of the specific systems, and organisation’s risk management in general?

4. Are there unexpected password issues not covered by existing policy?

The following observations were made,

1. When users cannot cope with the demands of strict password policies, it a) reduces their productivity, and b) leads them to adopt coping strategies - which usually reduce security.

2. Although passwords are usually considered in terms of authentication for a service oar device, today they are encountered in many other ways in the workplace – and existing password policies do not cover these. As a result, users adopt ad-hoc solutions, which are usually insecure.

3. Security depends on the context of use. Context - including virtual workstations, Single Sign-on, and home and mobile working - impacts not only the frequency of password use, but also on the risks associated with it.

**2.1 The Impact of Password Policies**

In this section, we examine the current guidelines on how to select password policies, and the user load results from different policies. Recent guidelines by the US National

Institute of Standards and Technology organise aspects of password policy around identified risks in the compromise of password-based authentication systems.

Among the recommendations which impact on users, they cover:

1. Secure transmission of passwords, to mitigate password capture;

2. Construction of strong passwords and a (high) limit on the frequency of guesses, to mitigate guessing and cracking;

3. Password expiration and avoidance of recently-used passwords, to mitigate the use of compromised passwords; and

4. Use of Single Sign-On and local password management, to enable stronger policies while reducing the load on the user.

Other points which a password policy should consider include:

1. Timeouts and screen locking, to mitigate opportunistic misuse of an unattended desktop; and

2. Rules about sharing passwords - traditional advice suggests that this is to be avoided, but others [1] found situations in which password sharing might be appropriate in the workplace [1] or for personal banking.

**2.2 Windows Password Security**

The windows security feature has been the most important issue ever since the beginning of the windows systems. The password feature is tied to user accounts and users can create or modify accounts, if they have administrator privileges. The general drawback was that there was no guarantee that the set passwords are strong or perhaps follow some guideline to make sure that they are secure enough. The existence of password policies made certain that the systems were protected using certain guidelines either set in the system itself or imposed by organizations so that users follow these rules to ensure that the systems were secure. Even then the general acceptance of these policies was not wide. Some policies became more and more forced through system configuration and some were based on the user abiding by them, for example, other policies require password change at the end of each work day or even once a week

**Chapter 3: Existing Systems**

Reliable personal recognition is required by a wide variety of access control systems. Examples of these systems include ATMs, laptops and cellular phones [1].

If these systems fail to meet the demands of reliable and robust authentication potential imposters may gain access to these systems. In order to enhance the security of access control systems two factor authentication (T-FA) has been introduced, wherein two factors are combined in order to authenticate a user. The key idea of T-FA is to sum up the security of two factors. These factors include passwords, representing “something you know”, or physical tokens, such as smart-cards, representing “something you have”. Additionally, biometric traits are applied, representing “something you are”.

**3.1 Biometrics**

An authentication factor is a piece of information used to authenticate or verify the identity of a user. In a T-FA system two different factors are combined in order to authenticate a user. It is claimed that T-FA generally delivers a higher level of authentication assurance compared to using just one factor. Three basic classes of factors can be distinguished: personal factors, such as user-defined passwords, physical factors, such as smart-cards or human factors, such as biometric traits [3]. Combining two factors from two different classes yields T-FA where each factor is applied independently.

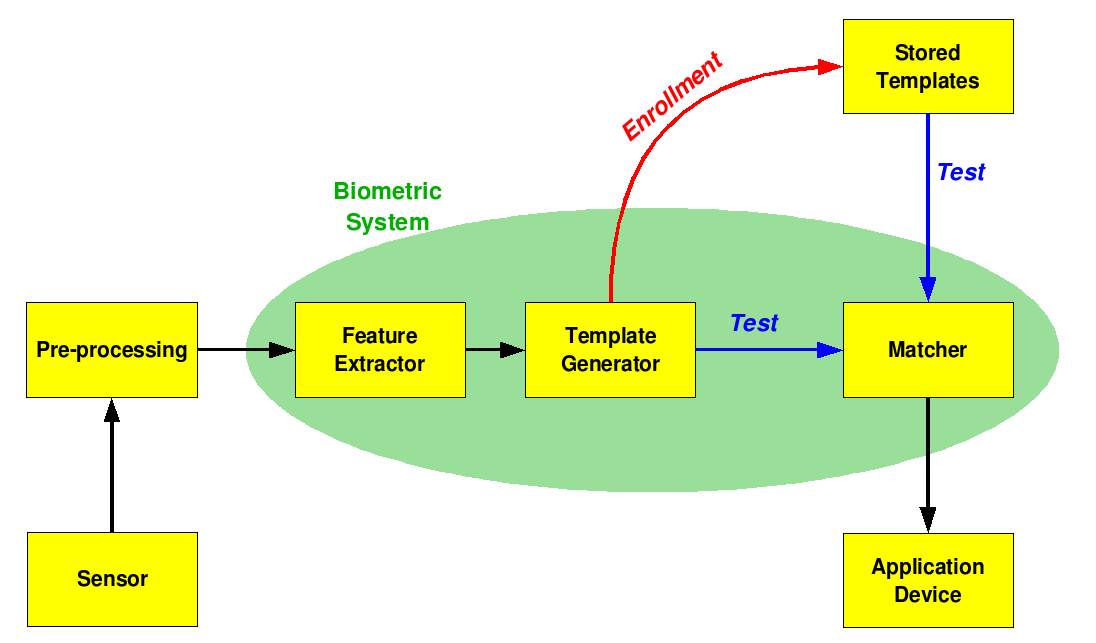


Fig3.1.a: Working of Biometric System

Biometrics means identifying or authenticating the user on the fingerprints, or retina scanner, etc.It requires an authenticating device which needs to be purchased or made.

**3.2 Iris Recognition Systems**

In order to apply biometric authentication we use our own implementation of the algorithm of Ma et al. In their approach the iris texture is treated as a kind of transient signal which is processed using a 1-D wavelet transform. The local sharp variation points, which denote important properties of transient signals, are recorded as features. We always extract an iris texture from eye images as a first step. We assume the texture to be the area between the two almost concentric circles of the pupil and the outer iris. These two circles are found by contrast adjustment, followed by canny edge detection and Hough transformation. After the circles are detected, unwrapping along polar coordinates is done to obtain a rectangular texture of the iris. In our case, we always resample the texture to a size of 512x64 pixels

The texture is subsequently divided into N stripes to obtain N one-dimensional signals, each one averaged from the pixels of M adjacent rows. We used N=10 and M= 5 for our 512x64 pixel textures (only the 50 rows close to the pupil are used from the 64 rows, as suggested in). A dyadic wavelet transform is then performed on each of the resulting 10 signals, and two fixed sub bands are selected from each transform. This leads to a total of 20 sub bands. In each sub-band we then locate all local minima and maxima above some threshold and write a bit code alternating between 0 and 1 at each extreme point. Using 512 bits per signal, the final code is then 512x20 bit.

Once bit codes are obtained, matching can be performed on them and Hamming distance lends itself as a very simple distance measure. For matching to work well, we compensate for eye tilt by shifting the bit-masks during matching by three pixels in each direction.

**3.3 Windows Password Authentication**

Windows has only one step password authentication which asks for the password during the login or the first time boot. After the login, any user from any login account can have access to all the user files stored in all the drives. This poses a very high potential risk to the user as the private and the extremely secret files cannot be stored in the shared system.

**Chapter 4: Aim and Objective**

The main aim of this system is to offer a possible solution to the difficulties being faced by the users who have difficulties abiding by some password policies and who are also conscious on securing their files.

The mention of single factor, two factor and three factor authentication arises. According to the research, authentication can involve something the user knows, for example a password and something the user has possibly a smartcard or any other token and finally something the user “is” for example a fingerprint or voice pattern.

Single factor authentication uses one of the mentioned forms, while Two Factor Authentication uses any two and three factor authentication uses all three. The authors acknowledge that using more factors makes it difficult for someone to gain access to the system. This is in contrast to just using single factor in the form of password authentication. This research focuses on two factor authentication (2FA) and attempts to strengthen the already available policies.

The users tend to have difficulties following policies set by their organization manly due to the requirements that they feel, are a constant pressure on their day to day operations. Now this has been a major observation. Users turn to more vulnerable methods that will expose them to numerous attacks and dangers of being hacked. This again confirms that a more robust method of authentication be invoked by organizations as a way to combat such important concerns and make the users a bit more comfortable.

**Chapter 5: Problem Statement**

**5.1 Efficiency of Results**

A.Campilo (2010) in his paper says that several problems may occur when introducing biometric authentication to T-FA systems. Performance gain with respect to recognition rates is often achieved due to the assumption of unrealistic preconditions. Resulting performance distortions may not be recognized at first sight, yet, these could lead to serious security vulnerabilities. In order to shed light on the use of biometrics as additional factors in T-FA schemes we demonstrate a way of how to untruly improve recognition rates in a biometric recognition system by introducing a token-based T-FA scheme. As a consequence, we manifest requirements with respect to performance measurements in T-FA systems including biometrics.

**5.2 Cost**

We tried to find out the cost of the existing password decentralized and 2 factor authenticating systems.

The costs were,

Finger print Scanner:Rs 15,000

Iris Scanner:Rs 30,000

As we can make out, the cost involved in such products are comparatively higher and a common man could not afford these to safeguard his private files.

**5.3 Weak Windows Security**

There is also mention of a program called LC5 that is capable of cracking simple passwords with eight characters in a matter of seconds. In the region of 5-8 characters is where most of the user’s password length resides. “According to classical cognitive research, humans can memorize only seven plus or minus two chunks of information (Miller 1956). Not surprisingly, a set of preliminary interviews showed that most users within the Client organization had difficulty remembering strong passwords. As a result, the users often wrote down their passwords. These written passwords are often kept in insecure locations and were subject to misuse”

**Chapter 6: Proposed System**

We propose a system which will use the Bluetooth of the mobile phones as the 2nd factor for the Two Factor Authentication. Nowadays almost all the cell phones comprise of Bluetooth and can be easily found.

**6.1 Working**

The system should work as follows:

During the first login, the user will log into the system. As soon as the computer will start, the software will start running as a background process. After the successful login, the software will start searching for the nearby Bluetooth devices. The user will then select one the Bluetooth device and the device will be paired with the computer and MAC address of the device will be stored in the registry. The user will then create a profile with the paired Bluetooth device and save it. He will then set the profile active. User will then choose the folder on which he is working or the folder he wants protection. The protection will then be enabled for that folder.

There may be two scenarios once the protection is enabled for that folder. Either the Bluetooth is in range or the Bluetooth is not in range.

When the Bluetooth is in range,

The software will prompt for a password and if the password is correct, all the files will be decrypted and displayed.

When the Bluetooth is not in range,

The software will check if the protection is enabled for that folder. Ifyes, then the software will encrypt all the files immediately the password won’t ever be asked.

**6.2Design of Proposed System**

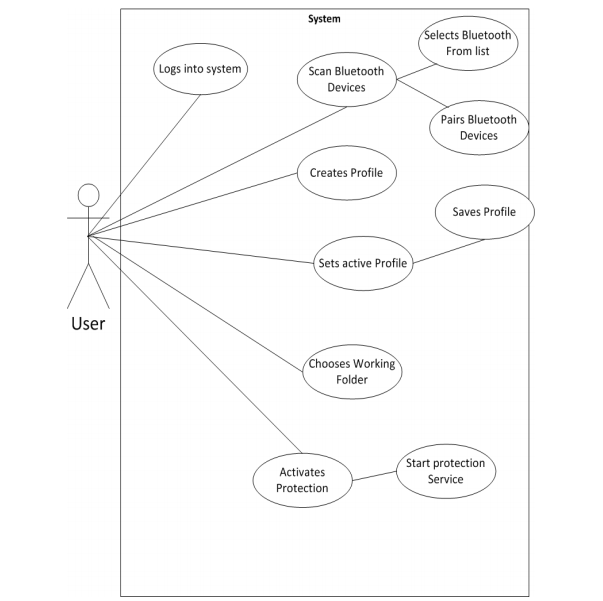


Fig6.2.a: Design of Proposed System

**Chapter 7: Requirement Analysis**

**7.1 Windows Service**

This is the service that runs in the background for the lifetime of the computer system runtime and will automatically be started when the computer boots. This service has functional requirements that it has to fulfill in order to be fully operational.

* + 1. **Lifetime service** – Service must run for the whole time the windows system is running. Scan Bluetooth Proximity – Service should be able to recognize when a particular Bluetooth device is range. This particular device is the one that is saved in the database as the active profile. It should match the service records of the Bluetooth that match the MAC address of that device.
    2. **Encrypt Files** – Should be able to encrypt files in the working folder whenever the Bluetooth device goes out of range.
    3. **Lock workstation** – The service should automatically lock the windows workstation when the authenticated Bluetooth device is out of range.

**Chapter 8: Scope**

In future, we can put a criteria or alternate means of decryption when the phone is lost. When the user reports a loss of phone, then the software will provide a series of keystroke for that specific build of the software by which the password for decryption will be asked. The user will need to change the paired device when he logins next time.

We may also allow the user to add multiple phones in his profile. This will improve the variability in the software supposedly the user owns multiple phones.

To avoid unethical pairing by some intruder other than the user with the users paired phone, an application in android can be developed which will only allow pairing when the password is entered. However this requires the learning of the android programming language.

**Chapter 9: Methodology**

The concept behind this proposed system may be divided into two categories:

* 1. Connection Between The Laptop And The Bluetooth Device
  2. Encryption Method

**9.1 Connection between the Laptop and the Bluetooth Device**

The software will start as a background process as soon as the laptop will start.

It should start discovering Bluetooth devices. All the Bluetooth devices have a specific MAC (Mobile Authentication Code) Address in it. The discovery of the Bluetooth devices will be done using the MAC address. The software will then pair with the Bluetooth device.

JSR-82(Also known as Blue Cove) is a java specification for defining APIs for communicating with Bluetooth devices.

Handshake protocol will be implemented by the software with the Bluetooth device every 5 seconds. If the device is not in range, then all the files for which protection has been enabled will be encrypted.

**9.2** **Encryption:**

As an extra added feature but highly effective, to enhance security, encryption of files seems best as this preserves the confidentiality of most sensitive files. They are many competitive encryption algorithms out there and they differ on their efficiency, speed and complexity.

The encryption method adopted for this research is the Rijndael cipher which is an Advanced Encryption Standard (AES). AES is based on a design principle known as a substitution permutation network, and is fast in both software and hardware (Bruce Schneider et al, 2000).

Unlike its predecessor DES, AES does not use a Festal network. AES is a variant of Rijndael which has a fixed block size of 128 bits, and a key size of 128, 192, or 256 bits. By contrast, the Rijndael specification per se is specified with block and key sizes that may be any multiple of 32 bits, both with a minimum of 128 and a maximum of 256 bits. The Rijndael encryption in overall performance, based on the speed of encryption and decryption and on key set-up time, the algorithm has attained top scores in tests conducted by National Institute of Standards and Technology (NIST). The belief is that almost all US government agencies will shift to the AES algorithm for their data security needs in the next few years. Also, that the algorithm will find its way in smart cards and other security-oriented applications used for safely storing private information about individuals.

In terms of performance, High speed and low RAM requirements were criteria of the AES selection process. Thus AES performs well on a wide variety of hardware, from 8 -bit smart cards to high-performance computers. On a Pentium Pro, AES encryption requires 18 clock cycles / byte, equivalent to a throughput of about 11 MiB/s for a 200 MHz processor. On a Pentium M1.7 GHz throughput is about 60 MiB/s. On Intel i3/i5/i7 CPUs supporting AES-NI instruction set extensions, throughput can be over 700MiB/s per thread.

If files are encrypted and the cipher used is commendable, then we can at least ensure that if we are hacked, the hackers will have a difficult time cracking the algorithm. At the end of the day files are in a secure state and the most important factor to almost all organizations, Confidentiality, is preserved.

**Chapter 10: Design Details**

**10.1 Context Level Diagram**

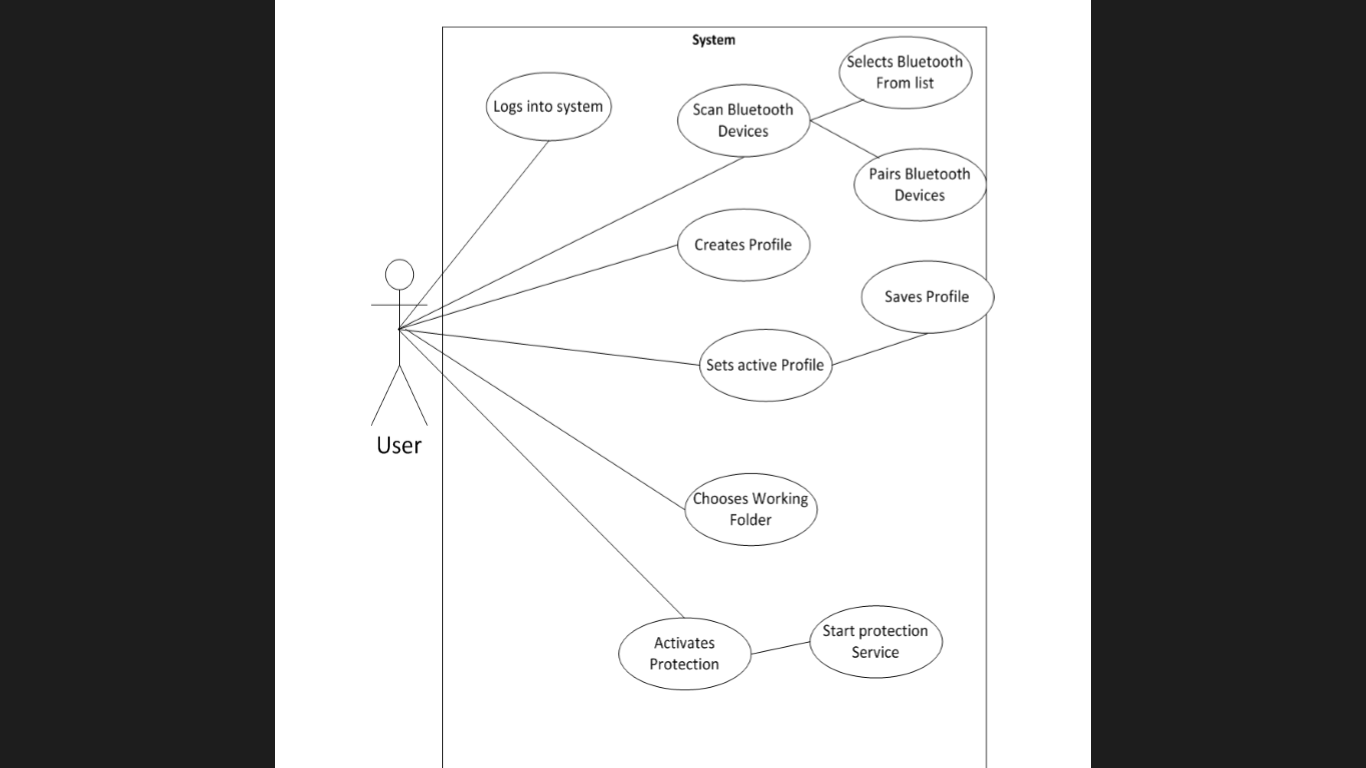


Fig 10.1.a: Context level diagram

**10.2 DFD Diagram**

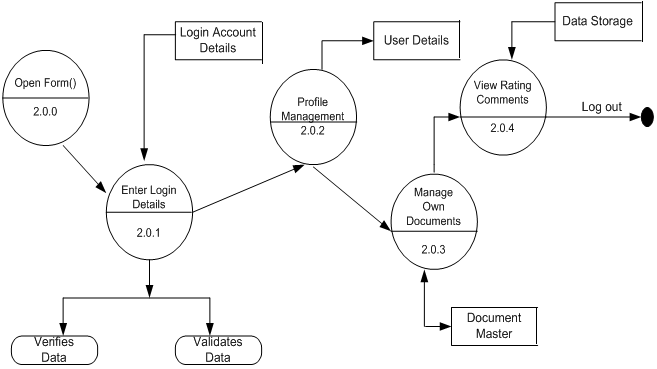


Fig 10.2.a: DFD Diagram

**10.3 Sequence Diagram**

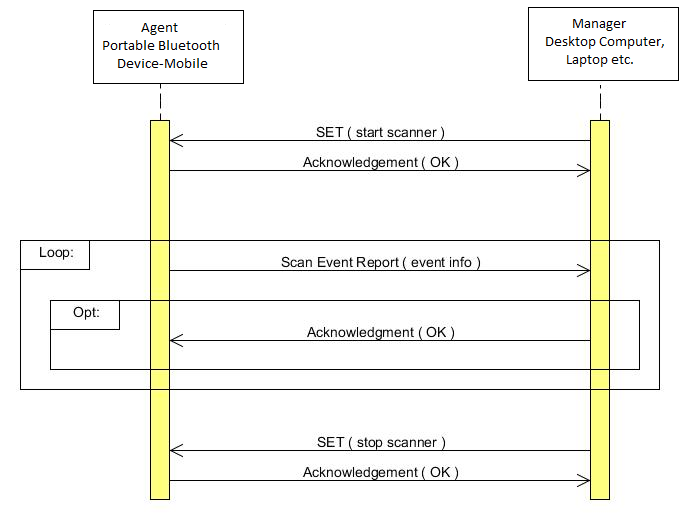


Fig 10.3.a: Sequence Diagram

**10.4 ER Diagram**

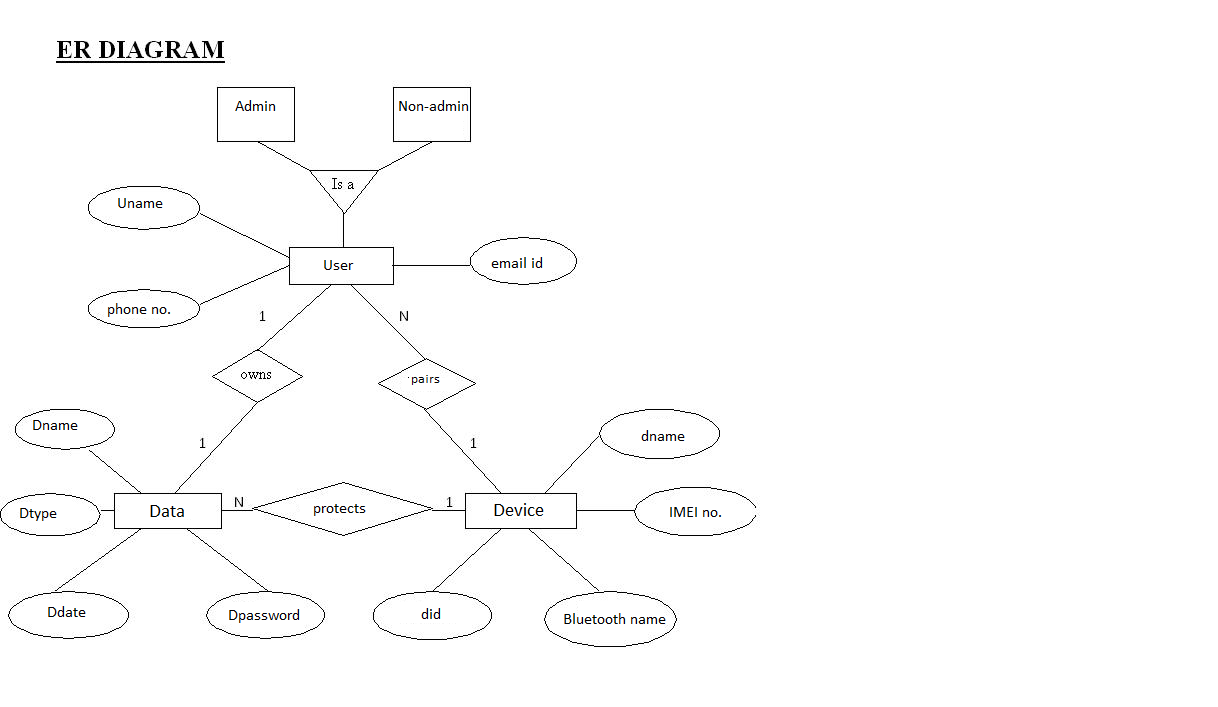


Fig 10.4.a: ER Diagram

**10.5 Control Flow Diagrams**

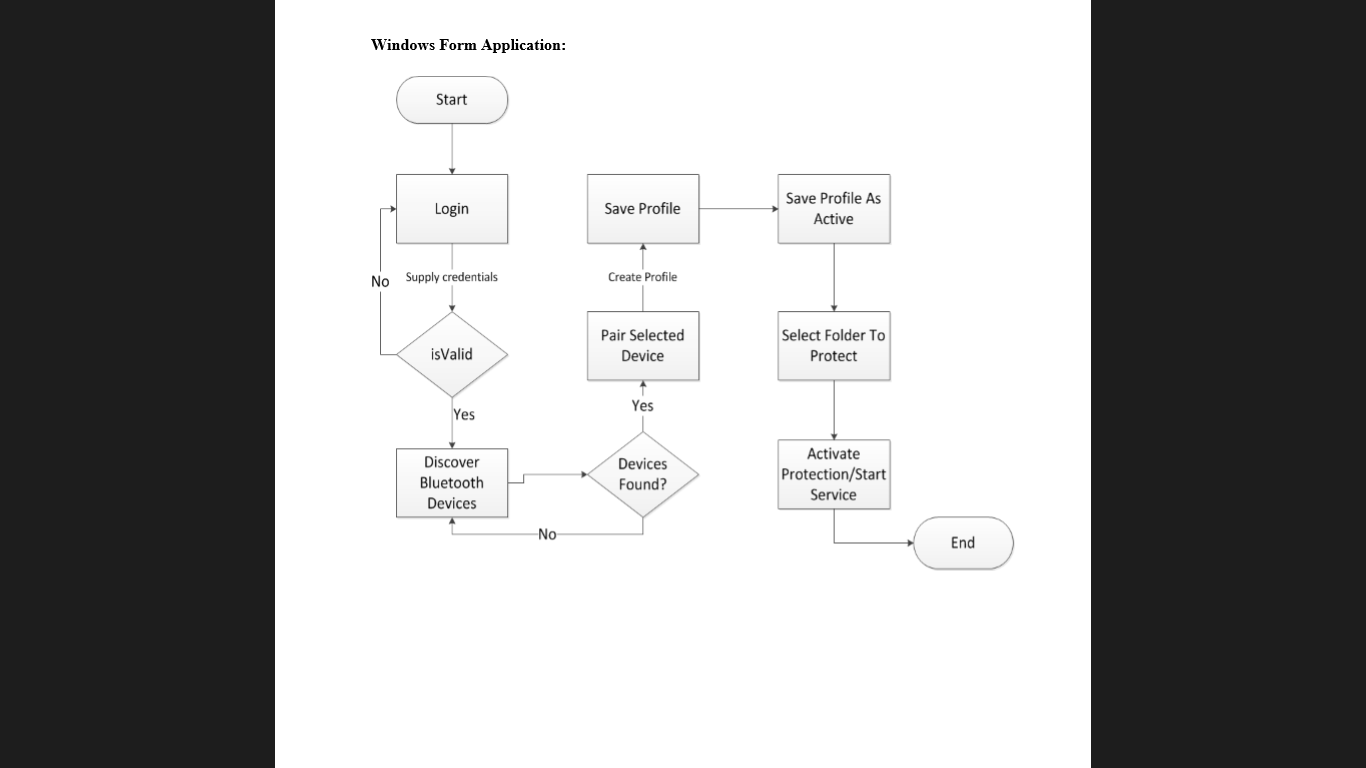
**10.5.1 Windows Form Application**

Fig 10.5.1.a: Windows Form Application

**10.5.2 Windows Service**

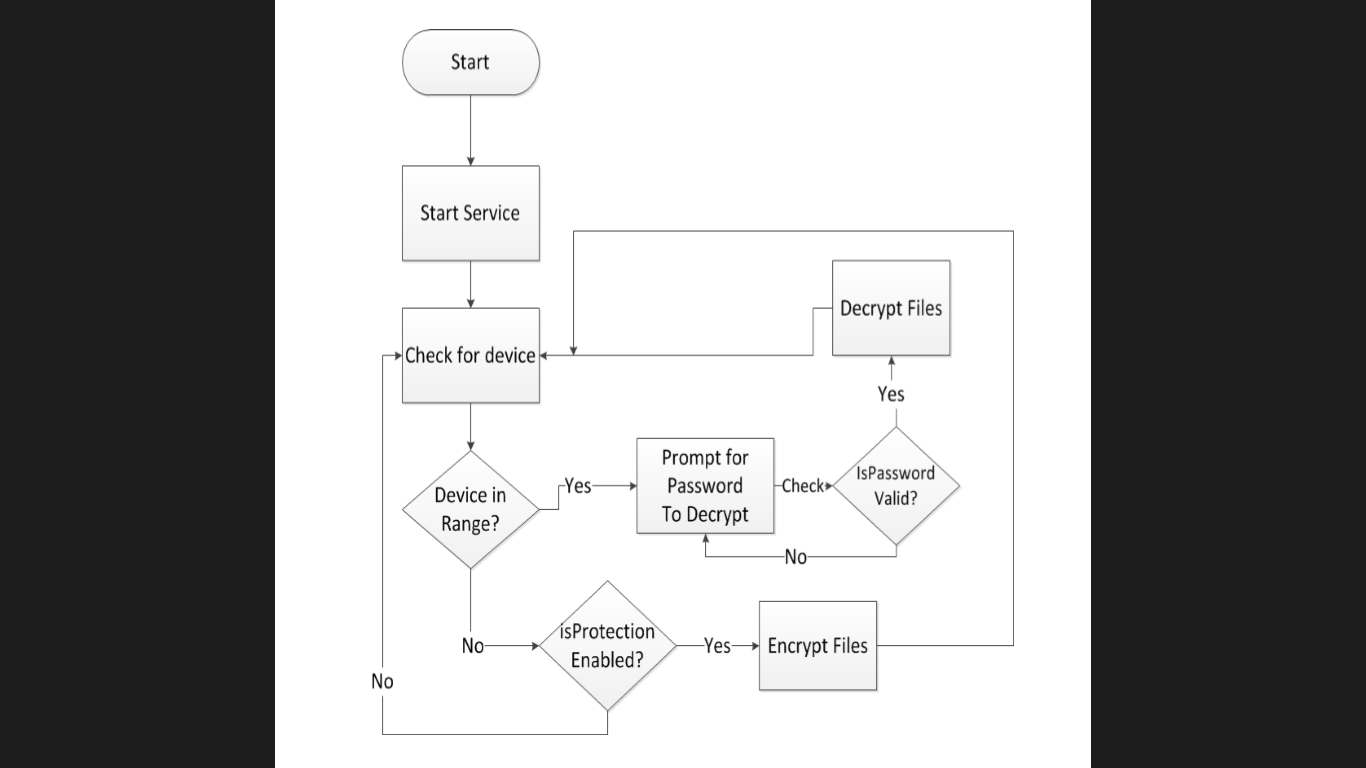


Fig 10.5.2.a: Windows Service

**Chapter 11: Implementation Plan**

The research will be implemented in 4 stages:

* 1. Analysis
  2. Design
  3. Coding
  4. Testing

**11.1 Analysis**

Requirements analysis in , encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product, taking account of the possibly conflicting [requirements](http://en.wikipedia.org/wiki/Requirement) of the various [stakeholders](http://en.wikipedia.org/wiki/Stakeholder_(corporate)), analysing, documenting, validating and managing software or system requirements.

Requirements analysis is critical to the success of a systems or software project. The requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design.

**11.2 Design**

**11.2.1 Architectural Design**

Architectural design represents the structure of data and program components that are required to build a computer-based system.

**11.2.2 Component Level Design**

Component-level design defines the data structures, algorithms, interface characteristics, and communication mechanisms allocated to each software component.

**11.2.3 User Interface Design**

User interface design creates an effective communication medium between a human and a computer.

**11.3 Coding**

**11.3.1 Interface coding**

An interface is a shared boundary across which two separate components of a [computer system](http://en.wikipedia.org/wiki/Computer_system) exchange information. The exchange can be between [software](http://en.wikipedia.org/wiki/Software), [computer hardware](http://en.wikipedia.org/wiki/Computer_hardware), [peripheral](http://en.wikipedia.org/wiki/Peripheral) devices, humans and combinations of these.

**11.4 Testing**

**11.4.1 Unit Testing**

Unit testing is a [software testing](http://en.wikipedia.org/wiki/Software_testing) method by which individual units of [source code](http://en.wikipedia.org/wiki/Source_code), sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures are tested to determine if they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application. In [procedural programming](http://en.wikipedia.org/wiki/Procedural_programming), a unit could be an entire module, but it is more commonly an individual function or procedure. In [object-oriented programming](http://en.wikipedia.org/wiki/Object-oriented_programming), a unit is often an entire interface, such as a class, but could be an individual method. Unit tests are short code fragments created by programmers or occasionally by [white box testers](http://en.wikipedia.org/wiki/White-box_testing) during the development process.

**11.4.2 Integration Testing**

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in [software testing](http://en.wikipedia.org/wiki/Software_testing) in which individual software modules are combined and tested as a group. It occurs after [unit testing](http://en.wikipedia.org/wiki/Unit_testing) and before [validation testing](http://en.wikipedia.org/wiki/Verification_and_validation_(software)). Integration testing takes as its input [modules](http://en.wikipedia.org/wiki/Module_(programming)) that have been [unit tested](http://en.wikipedia.org/wiki/Unit_testing), groups them in larger aggregates, applies tests defined in an integration [test plan](http://en.wikipedia.org/wiki/Test_plan) to those aggregates, and delivers as its output the integrated system ready for [system testing](http://en.wikipedia.org/wiki/System_testing).

**11.4.3 Validation Testing**

Verification and validation (V&V) is the process of checking that a software system meets specifications and that it fulfils its intended purpose. It may also be referred to as [software quality control](http://en.wikipedia.org/wiki/Software_quality_control). It is normally the responsibility of [software testers](http://en.wikipedia.org/wiki/Software_testing) as part of the [software development lifecycle](http://en.wikipedia.org/wiki/Software_development_process)

**11.4.4 Alpha Testing**

Alpha testing is simulated or actual operational testing by potential users/customers or an independent test team at the developers' site. Alpha testing is often employed for off-the-shelf software as a form of internal acceptance testing, before the software goes to beta testing.

**11.4.5 Beta Testing**

Beta testing comes after alpha testing and can be considered a form of external [user acceptance testing](http://en.wikipedia.org/wiki/User_acceptance_testing). Versions of the software, known as [beta versions](http://en.wikipedia.org/wiki/Beta_version), are released to a limited audience outside of the programming team. The software is released to groups of people so that further testing can ensure the product has few faults or [bugs](http://en.wikipedia.org/wiki/Computer_bug). Sometimes, beta versions are made available to the open public to increase the [feedback](http://en.wikipedia.org/wiki/Feedback#In_organizations) field to a maximal number of future users.

**Chapter 12: Hardware and Software Requirements**

**12.1 Software: For Fluent Communication between Bluetooth Devices**

**12.1.1 Java SE 6**

Java is a [computer programming language](http://en.wikipedia.org/wiki/Computer_programming_language) that is [concurrent](http://en.wikipedia.org/wiki/Concurrent_computing), [class-based](http://en.wikipedia.org/wiki/Class-based), [object-oriented](http://en.wikipedia.org/wiki/Object-oriented_programming), and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "[write once, run anywhere](http://en.wikipedia.org/wiki/Write_once,_run_anywhere)" (WORA), meaning that code that runs on one platform does not need to be recompiled to run on another. Java applications are typically [compiled](http://en.wikipedia.org/wiki/Compiler) to [bytecode](http://en.wikipedia.org/wiki/Java_bytecode) that can run on any [Java virtual machine](http://en.wikipedia.org/wiki/Java_virtual_machine) (JVM) regardless of [computer architecture](http://en.wikipedia.org/wiki/Computer_architecture). Java is, as of 2014, one of the most popular programming languages in use. Java was originally developed by [James Gosling](http://en.wikipedia.org/wiki/James_Gosling) at [Sun Microsystems](http://en.wikipedia.org/wiki/Sun_Microsystems) (which has since [merged into Oracle Corporation](http://en.wikipedia.org/wiki/Sun_acquisition_by_Oracle)) and released in 1995 as a core component of Sun Microsystems' [Java platform](http://en.wikipedia.org/wiki/Java_(software_platform)). The language derives from [C](http://en.wikipedia.org/wiki/C_(programming_language)) and [C++](http://en.wikipedia.org/wiki/C%2B%2B), but it has fewer [low-level](http://en.wikipedia.org/wiki/Low-level_programming_language) facilities than either of them.

**12.1.2 NetBeans IDE 7.3 Debugger**

The Java Platform Debugger Architecture is a collection of [APIs](http://en.wikipedia.org/wiki/API) to [debug](http://en.wikipedia.org/wiki/Debugging) [Java](http://en.wikipedia.org/wiki/Java_(programming_language)) code. Java Debugger Interface (JDI) - defines a high-level Java language interface that developers can easily use to write remote debugger application tools. [Java Virtual Machine Tools Interface](http://en.wikipedia.org/wiki/Java_Virtual_Machine_Tools_Interface) (JVMTI), a native interface that helps to inspect the state and to control the execution of applications running in the [Java Virtual Machine](http://en.wikipedia.org/wiki/Java_Virtual_Machine) ([JVM](http://en.wikipedia.org/wiki/JVM)). Java Virtual Machine Debug Interface (JVMDI)- JVMDI was deprecated in J2SE 5.0 in favor of JVM TI, and was removed in Java SE 6.Java Debug Wire Protocol (JDWP) - defines communication between debugger (a Java application) and [debugger](http://en.wikipedia.org/wiki/Debugger) processes.

**12.1.3 Bluecove is a JSR-82 implementation.**JSR-82 is a java specification for defining APIs for communicating with Bluetooth devices.

* + 1. **Windows 7/8 OS**

**12.2 Hardware:**

* + - Mobile Phone with Bluetooth
    - Bluetooth dongle Plug-n-Play(External Bluetooth Device)
    - PC or Laptop

**Chapter 13: GANTT CHART**

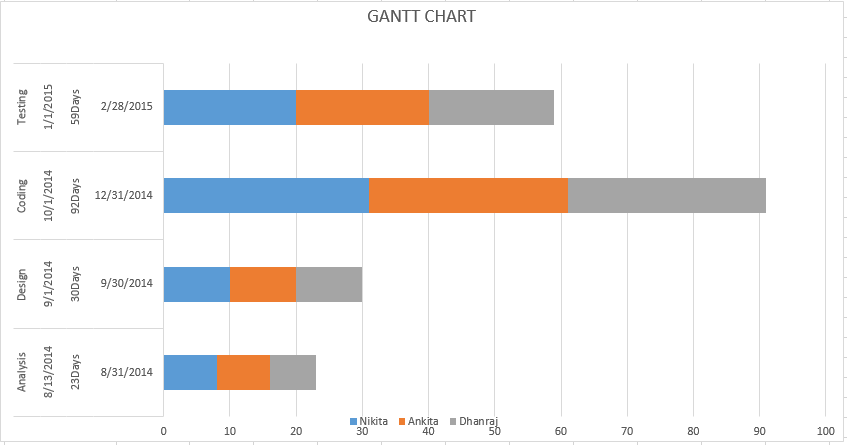


Fig 13.a: Gantt chart

**Chapter 14: Appendix**

Rijndael Encryption: The Rijndael cipher which is an Advanced Encryption Standard (AES). AES is based on a design principle known as a substitution permutation network, and is fast in both software and hardware

Bluecove: Bluecove (JSR-82) is a java specification for defining APIs for communicating with Bluetooth devices.

Java SE 6: Java is a [computer programming language](http://en.wikipedia.org/wiki/Computer_programming_language) that is [concurrent](http://en.wikipedia.org/wiki/Concurrent_computing), [class-based](http://en.wikipedia.org/wiki/Class-based), [object-oriented](http://en.wikipedia.org/wiki/Object-oriented_programming), and specifically designed to have as few implementation dependencies as possible.

NetBeans IDE 7.3 Debugger/Notepad/Lite-Pro Edition (LE-PRO): The Java Platform Debugger Architecture is a collection of [APIs](http://en.wikipedia.org/wiki/API) to [debug](http://en.wikipedia.org/wiki/Debugging) [Java](http://en.wikipedia.org/wiki/Java_(programming_language)) code.

Java Debugger Interface (JDI): It defines a high-level Java language interface that developers can easily use to write remote debugger application tools.

Bluetooth dongle: A personal computer that does not have embedded Bluetooth can be used with a Bluetooth adapter that will enable the PC to communicate with other Bluetooth devices. While some [desktop computers](http://en.wikipedia.org/wiki/Desktop_computer) and most recent laptops come with a built-in Bluetooth radio, others will require an external one in the form of a [dongle](http://en.wikipedia.org/wiki/Dongle).

**Chapter 15: References**

[1] “The True Cost of Unusable Password Policies: Password Use in the Wild “Philip Inglesant & M. Angela Sasse Department of Computer Science University College London Gower Street, London WC1E 6BT, UK @ 2010

[2] “2CAuth: A New Two Factor Authentication Scheme Using QR-Code” N. Harini 1 and Dr. T.R Padmanabhan2Department of Computer Science and Engineering, Amrita VishwaVidyapeetham, Coimbatore, India. @2013

[3] “Two-Factor Authentication or How to Potentially Counterfeit Experimental Results in

Biometric Systems” Christian Rathgeb and Andreas Uhl

University of Salzburg, Department of Computer Sciences, A-5020 Salzburg, Austria@2010

[4] “The First 10 Years of Advanced Encryption” Joan Daemen STMicro­electronics Vincent RiJmen Katholieke Universiteit Leuven@2010

[5] “E-Learning Concept Design Of Rijndael Encryption Process” Dr. Zahir Zainuddin, M.Sc. Information Technology Hasanuddin University Makassar, Indonesia Evanita V Manullang, ST Information Technology Science and Technology University of Jayapura Makassar, Indonesia@2013