iNote – A Digital Personal Notebook

Project Report submitted in partial

fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

Submitted by

ANUSHKA JAIN (17SCSE101501)
ABU AQUIB (17SCSE101141)
ABHIJEET SHARMA (17SCSE101114)

IN
COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

Under the Supervision of

Mr. Bibhas Kumar Rana (Assistant Professor)

GALGOTIAS UNIVERSITY
(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

APRIL 2021

SCHOOL OF COMPUTER SCENCE AND ENGINEERING

BONAFIDE CERTIFICATE

Certified that this project report "iNote- A Digital Personal Notebook" is the bonafide work of "Anushka Jain, Abhijeet Sharma, Abu Aquib" who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
MUNISH SBHARWAL	**BIBHAS KUMAR RANA**
Dean	**SUPERVISOR**
School of Computer Science and Engineering	**Assistant Professor, SCSE**
	Galgotias University
	Plot No. 2, Yamuna Expy, Opposite, Buddha International Circuit, Sector 17A, Greater Noida, Uttar Pradesh 203201

ABSTRACT

The aim of this project is to provide user a clean and helpful Interface where he/she can store the valuable Data of day to day life.

iNote is a multiuser Application that means a number of user’s can store and access their personal data without been interfering or messing up of the Data. The system provides a login form which authenticates the User and allows it to create a new Account in case the user is new to the System or want to create a new Storage for their secure Storage and retrieval.

This system provide a single password authentication for a number of user data Storage which leads to a hassle free and yet secured way to store the Data.

iNote is a XML based system that also provide a liberty for future upgradability of the System and Data.

The future Scope of the Project is that it can be used on a variety of the OS which allows an easy portability of that Secured User Data. iNote is an XML based system that also provide a liberty for future upgradability of the System and Data. Xamarin claims whatever you used to do with Java, Objective-C or Swift, the same thing you can achieve with C#, a single language used by Xamarin. Getting to all the highlights of C#, and its .Net libraries you can construct Android, iOS and Windows Native applications.

TABLE OF CONTENTS

1. INTRODUCTION

What is .NET Framework-

The **.**NET Framework (pronounced *dot net*) is developed by Microsoft. The **.**NET Framework is a software framework that runs primarily on Microsoft Windows. It includes a large library and provides language interoperability (each language can use code written in other languages) across several programming languages. Programs written for the .NET Framework execute in a software environment (as contrasted to hardware environment), known as the Common Language Runtime (CLR), an application virtual machine that provides important services such as security, memory management, and exception handling. The class library and the CLR together constitute the .NET Framework.

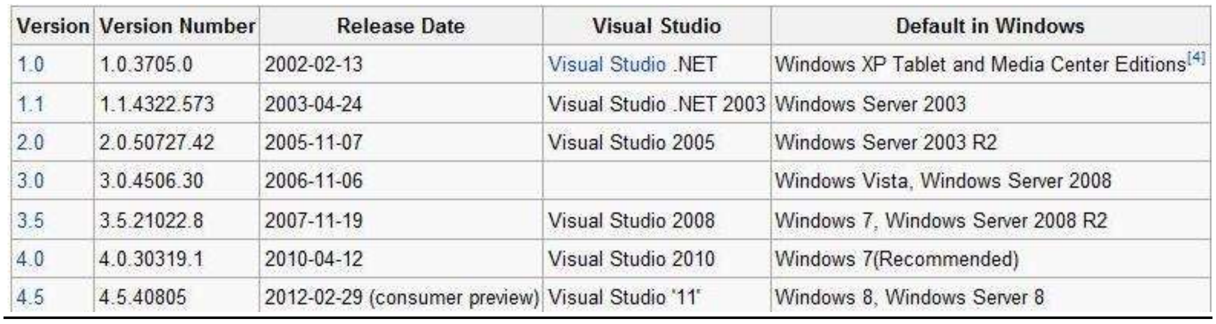
The .NET Framework's Base Class Library provides user interface, data access, database connectivity, cryptography, web application development, numeric algorithms, and network communications. Programmers produce software by combining their own source code with the .NET Framework and other libraries. The .NET Framework is intended to be used by most new applications created for the Windows platform. Microsoft also produces a popular integrated development environment largely for .NET software called Visual Studio.

Microsoft started the development on the .NET Framework in the late 1990s originally under the name of Next Generation Windows Services (NGWS). By late 2000 the first beta versions of .NET 1.0 were released.

Version 3.0 of the .NET Framework is included with Windows Server 2008 and Windows Vista. Version 3.5 is included with Windows 7, and can also be installed on Windows XP and the Windows Server 2003 family of operating systems .On 12 April 2010, .NET Framework 4 was released alongside Visual Studio 2010.

The .NET Framework family also includes two versions for mobile or embedded device use. A reduced version of the framework, the .NET Compact Framework, is available on Windows CE platforms, including Windows Mobile devices such as smartphones. Additionally, the .NET Micro Framework is targeted at severely resource-constrained devices.

The .NET Framework is an integral Windows component that supports building and running the next generation of applications and Web services. The key components of the .NET Framework are the common language runtime (CLR) and the .NET Framework class library, which includes ADO.NET, ASP.NET, Windows Forms, and Windows Presentation Foundation (WPF). The .NET Framework provides a managed execution environment, simplified development and deployment, and integration with a wide variety of programming languages.



Version	Version Number	Release Date	Visual Studio	Default in Windows
1.0	1.0.3705.0	2002-02-13	Visual Studio .NET	Windows XP Tablet and Media Center Editions[4]
1.1	1.1.4322.573	2003-04-24	Visual Studio .NET 2003	Windows Server 2003
2.0	2.0.50727.42	2005-11-07	Visual Studio 2005	Windows Server 2003 R2
3.0	3.0.4506.30	2006-11-06		Windows Vista, Windows Server 2008
3.5	3.5.21022.8	2007-11-19	Visual Studio 2008	Windows 7, Windows Server 2008 R2
4.0	4.0.30319.1	2010-04-12	Visual Studio 2010	Windows 7(Recommended)
4.5	4.5.40805	2012-02-29 (consumer preview)	Visual Studio '11'	Windows 8, Windows Server 8

Design features-

Interoperability-

Because computer systems commonly require interaction between newer and older applications, the .NET Framework provides means to access functionality implemented in programs that execute outside the .NET environment. Access to COM components is provided in the System.Runtime.Interop Services and System. Enterprise Services namespaces of the framework; access to other functionality is provided using the P/Invoke feature.

Common Language Runtime Engine-

The Common Language Runtime (CLR) is the execution engine of the .NET Framework. All .NET programs execute under the supervision of the CLR, guaranteeing certain properties and behaviors in the areas of memory management, security, and exception handling.

Language Independence-

The .NET Framework introduces a Common Type System, or CTS. The CTS specification defines all possible datatypes and programming constructs supported by the CLR and how they may or may not interact with each other conforming to the Common Language Infrastructure (CLI) specification. Because of this feature, the .NET Framework supports the exchange of types and object instances between libraries and applications written using any conforming .NET language.

Base Class Library-

The Base Class Library (BCL), part of the Framework Class Library (FCL), is a library of functionality available to all languages using the .NET Framework. The BCL provides classes that encapsulate a number of common functions, including file reading and writing, graphic rendering, database interaction, XML document manipulation, and so on.

Simplified Deployment-

The .NET Framework includes design features and tools which help manage the installation of computer software to ensure it does not interfere with previously installed software, and it conforms to security requirements.

Security-

The design is meant to address some of the vulnerabilities, such as buffer overflows, which have been exploited by malicious software. Additionally, .NET provides a common security model for all applications.

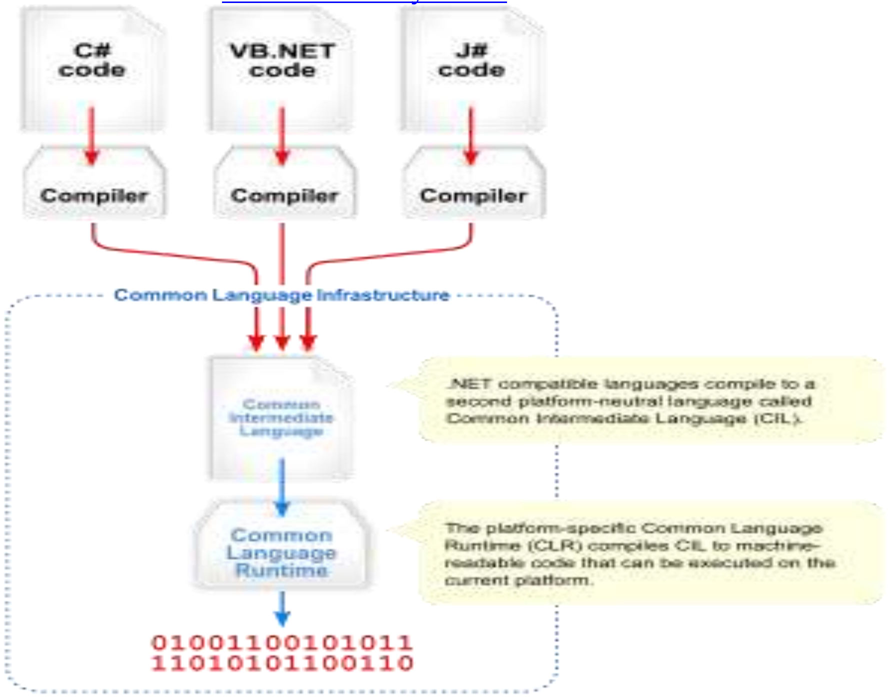
Portability-

While Microsoft has never implemented the full framework on any system except Microsoft Windows, the framework is engineered to be platform agnostic, and cross-platform implementations are available for other operating systems (see Silverlight and the Alternative implementations section below). Microsoft submitted the specifications for the Common Language Infrastructure (which includes the core class libraries, Common Type System, and the Common Intermediate Language),- the C# language- and the C++/CLI language- to both ECMA and the ISO, making them available as official standards. This makes it possible for third parties to create compatible implementations of the framework and its languages on other platforms.

Architecture-

Common Language Infrastructure (CLI)-

The purpose of the Common Language Infrastructure (CL) is to provide a language-neutral platform for application development and execution, including functions for Exception handling, Garbage Collection, security, and interoperability. By implementing the core aspects of the .NET Framework within the scope of the CL, this functionality will not be tied to a single language but will be available across the many languages supported by the framework. Microsoft's implementation of the CLI is called the Common Language Runtime, or CL.
The CIL code is housed in .NET assemblies. As mandated by specification, assemblies are stored in the Portable Executable (PE) format, common on the Windows platform for all DLL and EXE files. The assembly consists of one or more files, one of which must contain the manifest, which has the metadata for the assembly. The complete name of an assembly (not to be confused with the filename on disk) contains its simple text name, version number, culture, and public key token. Assemblies are considered equivalent if they share the same complete name, excluding the revision of the version number. A private key can also be used by the creator of the assembly for strong naming. The public key token identifies which public key an assembly is signed with. Only the creator of the keypair (typically the .NET developer signing the assembly) can sign assemblies that have the same strong name as a previous version assembly, since he is in possession of the private key. Strong naming is required to add assemblies to the Global Assembly Cache.


C# code
VB.NET code
J# code
Compiler
Compiler
Compiler
Common Language Infrastructure
Common Intermediate Language
.NET compatible languages compile to a second platform-neutral language called Common Intermediate Language (CIL).
Common Language Runtime
The platform-specific Common Language Runtime (CLR) compiles CIL to machine-readable code that can be executed on the current platform.
01001100101011
11010101100110

Security-

.NET has its own security mechanism with two general features: Code Access Security (CA), and validation and verification. Code Access Security is based on evidence that is associated with a specific assembly. Typically the evidence is the source of the assembly (whether it is installed on the local machine or has been downloaded from the intranet or Internet). Code Access Security uses evidence to determine the permissions granted to the code. Other code can demand that calling code is granted a specified permission. The demand causes the CL to perform a call stack walk: every assembly of each method in the call stack is checked for the required permission; if any assembly is not granted the permission a security exception is thrown.

Class library-

The .NET Framework includes a set of standard class libraries. The class library is organized in a hierarchy of namespaces. Most of the built-in APIs are part of either `System.*` or `Microsoft.*` namespaces. These class libraries implement a large number of common functions, such as file reading and writing, graphic rendering, database interaction, and XML document manipulation, among others. The .NET class libraries are available to all CLI compliant languages. The .NET Framework class library is divided into two parts: the Base Class Library and the Framework Class Library

The Base Class Library (BC) includes a small subset of the entire class library and is the core set of classes that serve as the basic API of the Common Language Runtime. The classes in `mscorlib.dll` and some of the classes in `System.dll` and `System.core.dll` are considered to be a part of the BCL. The BCL classes are available in both .NET Framework as well as its alternative implementations including .NET Compact Framework, Microsoft Silverlight and Mono.

The Framework Class Library (FCL) is a superset of the BCL classes and refers to the entire class library that ships with .NET Framework. It includes an expanded set of libraries, including Windows Forms, ADO.NET, ASP.NET, Language Integrated Query, Windows Presentation Foundation, Windows Communication Foundation among others. The FCL is much larger in scope than standard libraries for languages like C++, and comparable in scope to the standard libraries of Java.

Memory management-

The .NET Framework CL frees the developer from the burden of managing memory (allocating and freeing up when done); it handles memory management itself by detecting when memory can be safely freed. Memory is allocated to instantiations of .NET types (objects) from the managed heap, a pool of memory managed by the CL. As long as there exists a reference to an object, which might be either a direct reference to an object or via a graph of objects, the object is considered to be in use. When there is no reference to an object, and it cannot be reached or used, it becomes garbage, eligible for collection.

NET Framework includes a garbage collector which runs periodically, on a separate thread from the application's thread, that enumerates all the unusable objects and reclaims the memory allocated to them.

The .NET Garbage Collector (GC) is a non-deterministic, compacting, mark-and-sweep garbage collector. The GC runs only when a certain amount of memory has been used or there is enough pressure for memory on the system. Since it is not guaranteed when the conditions to reclaim memory are reached, the GC runs are non-deterministic. Each .NET application has a set of roots, which are pointers to objects on the managed heap (managed objects). These include references to static objects and objects defined as local variables or method parameters currently in scope, as well as objects referred to by CPU registers. When the GC runs, it pauses the application, and for each object referred to in the root, it recursively enumerates all the objects reachable from the root objects and marks them as reachable. It uses .NET metadata and reflection to discover the objects encapsulated by an object, and then recursively walk them. It then enumerates all the objects on the heap (which were initially allocated contiguously) using reflection. All objects not marked as reachable are garbage. This is the mark phase. Since the memory held by garbage is not of any consequence, it is considered free space. However, this leaves chunks of free space between objects which were initially contiguous. The objects are then compacted together to make used memory contiguous again. Any reference to an object invalidated by moving the object is updated by the GC to reflect the new location. The application is resumed after the garbage collection is over.

The GC used by .NET Framework is actually generational. Objects are assigned a generation; newly created objects belong to Generation 0. The objects that survive a garbage collection are tagged as Generation 1, and the Generation 1 objects that survive another collection are Generation 2 objects. The .NET Framework uses up to Generation 2 objects. Higher generation objects are garbage collected less frequently than lower generation objects. This helps increase the efficiency of garbage collection, as older objects tend to have a larger lifetime than newer objects. Thus, by removing older (and thus more likely to survive a collection) objects from the scope of a collection run, fewer objects need to be checked and compacted.

What is iNote aka Daily Assistant-

A Personal Diary used to Store a Varity of User Data in a Fast, Reliable and Secure Manner.

Review of Literature-

- iNote is a Portable Daily assistant which was developed by keeping two objective in mind that are the Data must be available where we want and the User must feel pleasure when using the System.
- Allow user to Save their Important Contacts and retrieve them back when required.
- Personal Diary:-Allow to create a Personal Diary for the Daily Writers.
- Expense Manager:-Help's to get Grip upon the Expense by tracking them.

What is XML?

- XML stands for Extensible Markup Language
- XML is a markup language much like HTML
- XML was designed to carry data, not to display data
- XML tags are not predefined. You must define your own tags
- XML is designed to be self-descriptive
- XML is a W3C Recommendation

The Difference between XML and HTML

XML is not a replacement for HTML.

XML and HTML were designed with different goals:

- XML was designed to transport and store data, with focus on what data is
- HTML was designed to display data, with focus on how data looks

HTML is about displaying information, while XML is about carrying information.

Problem Formulation

In our Daily life we find, generate and stores an amount of data which is need to be securely kept permanently on a medium until we want to dispose it, but what we do we store it on a paper pocket–dairy, or as a draft on mobile phone and off course these mediums are very portable but what if we lose them on the move.
Mobility gives data on move but if we lose them, we lose the Information also. We need to store these information's such as our contacts, Address ,Notes, To-Do's, Reminder's etc. on a Much more permanent Medium.

iNote just perfectly perform this task in a user-friendly manner with Data portability by using a custom file type based on XML data structure as MS office does. This system just not only Stores the Information but also represent it in a very easy and fluent manner so that the user gets whatever   Information needed.

This system also provides a way to recover contacts back to the Cell phone by using Mobile contact formats widely adopted by the Cellular phone manufactures.

Requirement Analysis

Technologies used:-

Requirement:-

.NET framework version 3.5:-

Version 3.5 is included with Windows 7, and can also be installed on Windows XP and the Windows Server 2003 family of operating systems. On 12 April 2010, .NET Framework 4 was released alongside Visual Studio 2010.

Details of the Software Used:-

Operating System: -	Microsoft windows operating system (Windows XP or Higher, Windows Seven Recommended)
Framework: -	Microsoft .NET framework v3.5
Frontend: -	Xamarin forms for windows Environment.
Backend: -	DBMS XML
Editors: -	Notepad, Visual studio 2019

Details of the Hardware Used:-

Processor: - Intel Pentium III or Higher(Intel Core i3 Recommended, or equivalent)

Ram: - 512 MB of DDR II (1 GB of DDRII is recommended)

Display: - 1024x786 resolution in 24 bit Color Depth (1366x786 recommended)

Optical Bay:- Standard CD Reader with 24x Read Speed

User requirements:-

Functional requirements:- Person should have a account on the system.

External interface requirements:- The user interface of software is responsible for all the interaction with the user. Almost every software has a user interface. Many users often judge a software product based on its user interface. An interface that is difficult to use leads to higher level of users errors and ultimately leads to user dissatisfaction. Users become particularly irritated when a system behaves in unexpected ways i.e. issued commands do not carry out actions according to the intuitive exceptions of the user. Therefore, sufficient care and attention should be paid to the design of the user interface of any software product. Development of a good user interface usually takes significant portion of the total system development effort.

This software is developed keeping in mind the basic characteristics of a goog user interface.

Conclusion

These days when the amount of the data is increasing very rapidly the security and integrity of user Data is one of the main Goal of any Desktop or a web Application. iNote just meet the Requirements of the Data Security, Integrity and Portability as well because this system uses a file structured Storage system that provides Incredible Flexibility to the System when it comes to the deployment or Accessing the Stored data to a Foreign Environment like as Macintosh, Linux etc.

Xamarin guarantees up to 100% code offering to Xamarin.Forms. Xamarin.Forms can be use with full Advantage where code sharing is given more significance and require less Stage explicit usefulness. For User Interface plan Xamarin. Structures Uses XAML. When you plan the design Connect all the Controls with your common backend code and get completely local Android, iOS and Windows Phone Apps. During runtime each control are planned to stage - explicit local UI components like Textbox on Windows, UI Text View for iOS and Edit Text for Android.

This System provides a Cool and Clean yet Productive environment to Store the User’s important Data while keeping him away from the hassel of remembering a tons of passcodes.

SCOPE OF THE FURTHER ENHANCEMENT:-

The system maintenance is the inevitable process. System requirements always change ans so a system must evolve if it is to remain useful.

The object oriented system is easier to maintain, as the objects are independent. They may be understood and modified as stand-alone entities. Changing the implementation of an object should not affect other system objects. This improves the understandability of design.

CONTACT MANAGER:-In this module a user can currently store its contact and simply retrieve it back either in the system view or can export for the mobile format but it the future a user can also locate the location of the contact it the system using the Google maps API.

VAULT MANAGER:-This module allows simply cryptographic storage of the user data that is protected by the main password of the system but it the future there is a possibility of previewing data before actual extraction of the crypt one.

SCHEDULER:-Present scheduler modular allows user to set there custom appointments or schedules with only one type of alarming sound but in future system capable to do desired function on appointment arrival i.e. openings a specific file application or turning of the system.

DIARY MANAGER: -Diary manager module allows basics text editor functionality with the short of advanced text processing functionalities like as insertion of image graphics etc with the certain parameters of daily human behavior. In future our system allows to create a portable file for daily dairy that can be used in mobiles another portable mediums.

MONEY MANAGER:-This module currently allows to store the daily expensive incomes payables and receivables of the user and allows to view them in a normal grid view. In future our system allows to provide statically view of our expenditure and incomes in a chart and bar representations.

REFERENCES

1. Knight, D. (2015). Why use digital interactive notebooks? 21st century learning. [Blog post]. http://www.studyallknight.com/2015/11/interactivedigitalnotebooks.html
2. Miller, B., & Martin, C. (2016). Digital Notebooks for Digital Natives. Science & Children,53(5), 84-89.
3. Robinson, C. (2018). Interactive and digital notebooks in the science classroom. Science Scope, 41(7), 20-25.
4. Sprenger, M. (2017). 101 Strategies to Make Academic Vocabulary Stick. Alexandria, Virginia: ASCD.
5. Transform your classroom with Google Classroom. (2018). Retrieved from https://edu.google.com/k-12-solutions/classroom/?modal_active=none
6. Weimer, M. (2013). Learner-centered teaching: Five key changes to practice. [E-book]. Weimer, M. (2018). Retrieved from https://www.facultyfocus.com/author/maryellen-weimer-phd/
7. An Introduction to Xamarin.Forms, [online] Available: https://docs.microsoft.com/en-us/xamarin/xamarin-forms/get-started/introduction-to-xamarin-forms.
8. Pulkit Sethi, Xamarin Application Architecture, [online] Available: https://blog.kloud.com.au/2018/01/17/xamarin-application-architecture.
9. Architecture, [online] Available: https://docs.microsoft.com/en-us/xamarinlcross-platform/app-fundamentals/building-cross-platform-applications/architecture.
10. Vimal Maheedharan, *Xamarin vs React Native: Which is Better for Cross-Platform App Development?*, [online] Available: https://www.cabotsolutions.com/xamarin-vs-react-native-which-is-better-for cross-platform-app-development.
11. Xamarin vs React Native vs Ionic: Cross-platform Mobile Frameworks Comparison, [online] Available: https://www.altexsoft.com/blog/engineering/xamarin-vs-react-native-vs-ionic-cross-platform-mobile-frameworks-comparison/.
12. https://docs.microsoft.com/enus/visualstudio/cross-platform/visual-studio-andxamarin.
13. https://forums.xamarin.com/.

14. https://msdn.microsoft.com/enin/library/mt488768.aspx
15. https://docs.microsoft.com/en-us/xamarin/crossplatform/macios/native-references.
16. https://docs.microsoft.com/enus/xamarin/xamarin-forms/

Statement of Project Report Preparation

1. Thesis title: iNote – A Digital Personal Notebook

2. Degree for which the report is submitted: B.Tech (Computer Science and Engineering)

3 Project Supervisor was referred to for preparing the report.

4. Specifications regarding thesis format have been closely followed.

5. The contents of the thesis have been organized based on the guidelines.

6. The report has been prepared without resorting to plagiarism.

7. All sources used have been cited appropriately.

8 The report has not been submitted elsewhere for a degree.

(Signature of the Student)
Name: Anushka Jain

Statement of Project Report Preparation

3. Thesis title: iNote – A Digital Personal Notebook

4. Degree for which the report is submitted: B.Tech (Computer Science and Engineering)

3 Project Supervisor was referred to for preparing the report.

8. Specifications regarding thesis format have been closely followed.

9. The contents of the thesis have been organized based on the guidelines.

10. The report has been prepared without resorting to plagiarism.

11. All sources used have been cited appropriately.

8 The report has not been submitted elsewhere for a degree.

(Signature of the Student)
Name: Abu Aquib

Statement of Project Report Preparation

5. Thesis title: iNote – A Digital Personal Notebook.

6. Degree for which the report is submitted: B.Tech (Computer Science and Engineering)

3 Project Supervisor was referred to for preparing the report.

12. Specifications regarding thesis format have been closely followed.

13. The contents of the thesis have been organized based on the guidelines.

14. The report has been prepared without resorting to plagiarism.

15. All sources used have been cited appropriately.

8 The report has not been submitted elsewhere for a degree.

(Signature of the Student)

Name: Abhijeet Sharma

Approval Sheet

This thesis/dissertation/report entitled iNote – A Digital Personal Notebook by Anushka Jain, Abu Aquib, Abhijeet Sharma is approved for the degree of Bachelor of Technology (Computer Science and Engineering).

Examiners

____

____

Supervisor (s)

____

____

____

Chairman

____

Date:__________

Place:__________