**AUTOMATIC QUESTION PAPER GENERATOR SYSTEM**

|  |  |
| --- | --- |
| **S.NO** | **TITLE** |
| 1 | Abstract |
| 2 | Introduction |
| 3 | Existing system |
| 4 | Disadvantages |
| 5 | Proposed system |
| 6 | Advantages |
| 7 | Flow Diagram |
| 8 | Use case Diagram |
| 9 | Sequential Diagram |
| 10 | Activity Diagram |
| 11 | Modules |
| 12 | Modules Description |
| 13 | H/W&S/W Requirements |
| 14 | H/W&S/W Description |
| 15 | Testing of Product |
| 16 | Coding & Screenshot |
| 17 | Conclusion |
| 18 | Future Enhancement |

**TABLE OF CONTENT**

**Abstract:**

Information and intelligence are two vital columns on which development of humankind rise and knowledge has significant impact on operating of society. Student assessment is a crucial part of teaching and is done through the process of examinations and preparation of exam question papers has consistently been a matter of interest. Present-day technologies assist the teacher to stock the questions in a computer databases but the problem which emerges is how the present day technologies would also assist the teachers to automatically create the variety sets of questions from every now and then without worry about replication and duplication from the previous exam while the question bank keeps growing, so a non-automatic path for conniving exam paper would not be able to serve to this need so in this paper we introduce an automated way which would permit the operation of conniving exam paper to be further well organized and productive and it would also aid in developing a database of questions which could be further classified for blending of exam question paper, currently there is no systematic procedure to fortify quality of exam question paper. Hence there appears a requirement to have a system which will automatically create the question paper from teacher entered description within few seconds. It describes the utilization of randomization algorithm in an Automatic Question paper Generator System which has been implemented specially for autonomous institutes. The endeavour needed for generating question paper is diminished after the implementation of this advanced system and because of this advanced system there is no obligation for humans to ponder and employ time which can be utilized on some additional important duty instead of designing question paper.

**Introduction:**

In today’s current ambitious world, an examination plays a crucial role in checking the academic development of students and the era of information technology is now substituted by productive application of the technology. So producing utility from knowledge is crucial for development of society into an “Information Society”. For various examinations conducted in a year in any academic course, teachers need to create variation of question papers as per the college guidelines and assessment requirements. It is very difficult for the teachers to cover all features of the course outcomes and evade duplication of questions in the succeeding exams. There is no systematic procedure and hence the quality of the question paper relies entirely on an individual teacher’s experience and proficiency. At times, this entire element may degrade standard of the question paper. As per research, a quality question paper is a real combination of questions supervised by varied criteria such as difficulty level, distribution of marks across the question paper in form of paper pattern and the type of examination. The procedure involved in composition of an equitable examination paper by an independent is challenging and complex. Standard of the examination paper rely on diverse set of specifications so taking into account the distinct levels of learners is also a crucial parameter and the course outcomes also play a vital role in planning a systematic question paper. So associating the learning outcome of the subject to the examination paper is also a great job. With the profound dispersal of technology in the area of education, acquiring technology to smooth the technique of examination paper creation is a pure option and creation of extensively vast question bank and automatic exam paper generation furnishes a key provision to the issue encountered during the manual composition of examination papers. Automatic creation of examination paper yields a stage to create a well-organized examination paper and also the automation would smooth in incorporating many elements determining quality of a question paper. The structure presented in the next module is to automate the activity of examination paper generation. The system would be comprised of a cluster of questions upon which regulation would be implemented to create question paper. The structure is general and is not for any specific branch of learning. It seeks to furnish a generic procedure to the diverse requirement of distinct fields of study. This generic structure can be redesigned to all departments in colleges thus ease the assessment needs. Before the exam could be given to the student, teachers must compose the questions according to the modules covered for individual subject. A proficient question paper is habitually fit for usual students but it also encloses demanding items for clever students. Thus automatically creating question paper from a teacher’s entered description using a semantically labelled question bank is the requirement of the hour in present day. Here we are implementing a system which accordingly creates the question paper from this semantically labelled question bank. Since the prevailing systems are stiff and absence of flexibility of aiding all types of labels, the produced question paper may not be completely range with its given outcomes. Our each stuff is defined with a range demonstrating that value should not be under the bottom value and not surpass the topmost value of the range. Also, it is rule base system which takes all the associations of the labels and creates solution based on the rule relevant.

**Existing System:**

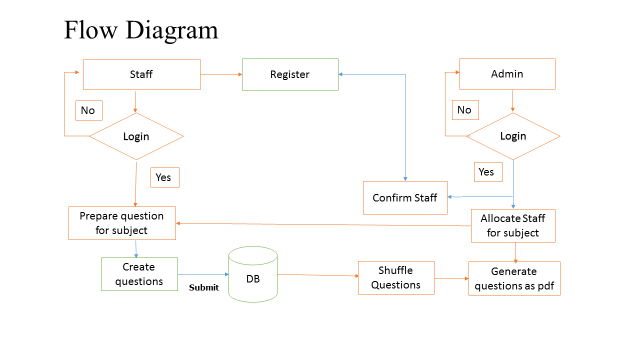
In the existing system the question preparation is a tedious job for the staffs. They need to do lot of paper work to prepare the question paper. It take more time until the final question is generated. Need to contact related staffs to prepare the questions. After all need to select the final question and take the printout.

**Proposed System:**

In our proposed system questions paper format can be created as pdf. Lots of paper works can be reduced. Stress for taking the question can be minimized. To use this system the Staff and the COE first want to login into the system. After staff login they are accepted by COE and then the staffs prepare the question for which pattern they want and questions are submitted. COE generate the last stage of questions.

**Diagrams:**

**Flow Diagram:**

****

**Use Case Diagram (Admin):**

Admin

DB

**Use Case Diagram (STAFF):**

Staff

DB

**Activity Diagram**

**Modules**

**COE**

* Login
* Verify Staff
* Allocate Staff
* Shuffle Questions
* View Questions
* Generate PDF

**STAFF**

* Login OTP
* Select Pattern
* Prepare Questions

**Modules Description:**

**Admin Module:**

The admin is the Controller of the examiner, Here COE has the main responsibility of the generating the question paper and split according to the department. To produce the question the admin first want to login into the system. After the login the admin can able to accept the registered staff. The admin here will check whether the staff is valid or not. If the staff is valid then staffs are accepted. After the acceptance of the staff the admin want to allocate the staff for each subject. The admin can select each staff that which question they need to take.

After the allocation the staff will prepare the questions and uploaded to the database. Now the admin can able to view the questions that are uploaded by the staffs. Admin can select the pattern based on how the question should be generated. After view Questions admin can able to generate the questions in the pdf format and can take the printout.

**Staff Module**

The Staff first want to register into the system. After the registration they want to wait for the approval of the admin. Once admin approved the staff they can login into the system. The staff can see for which subject they have selected to prepare the questions. Then the staff can choose any pattern that he wants. There are total of 5 patters with different questions format. Then the staff will start to prepare the questions. All the questions were uploaded to the database. Two or more staff can prepare the questions for the same subject.

**SYSTEM REQUIREMENTS**

**HARDWARE REQUIREMENTS:**

* System : Pentium IV 2.4 GHz
* Hard Disk : 160 GB
* Monitor : 15 VGA color
* Mouse : Logitech.
* Keyboard : 110 keys enhanced
* Ram : 2GB

**SOFTWARE REQUIREMENTS:**

* O/S : Windows 7
* Language : ASP.NET
* Code Behind : C#
* IDE : VISUAL STUDIO 10
* Data Base : SQL SERVER 2008
* .NET Framework : 4.0

**SYSTEM DESIGN**

**Introduction:**

System design is the process or art of defining the architecture, components, modules, interfaces and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development. Design is the first phase in development phase for any engineer’s product system. Design is the creative process. It deals with the creative ability of the programmer. A good design is the key to effective system. The term “Design” is defined as “The process of applying various techniques and principles for the purpose of defining a process or a system in sufficient details to permit its physical realization”.

**Input design**

The user interface design is very important for any application. The interface design describes how the software communicated within itself, to system that interpreted with it and with humans who use it. The interface is a packing for computer software if the interface is easy to learn, simple to use. If the interface design is very good, the user will fall into an interactive software application.

The input design is the process of converting the user-oriented inputs into the computer-based format. Errors entered by data entry operations can be controlled by input design. The data is fed into the system using simple interactive forms. The forms have been supplied with messages so that user can enter data without facing any difficulty.

The data is validated wherever it requires in the project. This ensures that only the correct data have been incorporated into the system. The goal for designing input data is to make data entry as easy, logical and free from errors.

The objectives of input design are:

* To produce a cost effective method of input
* To make the input forms understandable to the user
* To ensure the validation of data input
* To achieve the highest position level of accuracy

The various activities to be performed for the overall input processors are:

* Data recording at its source.
* Data transfer to input form.
* Data conversation to computer acceptable mode.
* Data validation.
* Data flow control.
* Data correction if necessary.

**Output Design**

The system output is the most important and direct source of information to the user. So intelligible output design improves the relationship with the user and helps in decision-making. Outputs from the computer systems are required primarily to communicate the results of processing to users. They are also used to provide a permanent copy of these results for later consultation.

A major form of output is a hard copy obtained from the printer. These printouts are designed to include the exact requirements of the user. The outputs required by the end-user are defined during the logical design stages.

Two phases of the output design are:

* Output definition.
* Output specification

Computer outputs are the most important and direct source of information to the user. A quality output is one which meets the requirements of the end user and which presents information in a way which is clear, easy to read and visually attractive. The screens are designed in such a way that the outputs are provided to the user in an understandable form.

The objectives of output design are:

* Design output to serve the indented purpose.
* Provide output on time.
* Assume that output is where it is needed.
* Design output to fit the user

**FEASIBILITY STUDY**

The feasibility study is carried out to test whether the proposed system is worth being implemented. The proposed system will be selected if it is best enough in meeting the performance requirements.

The feasibility carried out mainly in three sections namely.

**•** Economic Feasibility

• Technical Feasibility

• Behavioural Feasibility

**Economic Feasibility**

Economic analysis is the most frequently used method for evaluating effectiveness of the proposed system. More commonly known as cost benefit analysis. This procedure determines the benefits and saving that are expected from the system of the proposed system. The hardware in system department if sufficient for system development.

**Technical Feasibility**

This study center around the system’s department hardware, software and to what extend it can support the proposed system department is having the required hardware and software there is no question of increasing the cost of implementing the proposed system. The criteria, the proposed system is technically feasible and the proposed system can be developed with the existing facility.

**Behavioral Feasibility**

People are inherently resistant to change and need sufficient amount of training, which would result in lot of expenditure for the organization. The proposed system can generate reports with day-to-day information immediately at the user’s request, instead of getting a report, which doesn’t contain much detail.

**SOFTWARE DESCRIPTION**

**C#:**

**C#** is a [multi-paradigm programming language](https://en.wikipedia.org/wiki/Multi-paradigm_programming_language) encompassing [strong typing](https://en.wikipedia.org/wiki/Strong_typing), [imperative](https://en.wikipedia.org/wiki/Imperative_programming), [declarative](https://en.wikipedia.org/wiki/Declarative_programming), [functional](https://en.wikipedia.org/wiki/Functional_programming), [generic](https://en.wikipedia.org/wiki/Generic_programming), [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) ([class](https://en.wikipedia.org/wiki/Class_(computer_science))-based), and [component-oriented](https://en.wikipedia.org/wiki/Component-based_software_engineering) programming disciplines. It was developed by [Microsoft](https://en.wikipedia.org/wiki/Microsoft) within its [.NET](https://en.wikipedia.org/wiki/.NET_Framework) initiative and later approved as a standard by [Ecma](https://en.wikipedia.org/wiki/Ecma_International) (ECMA-334) and [ISO](https://en.wikipedia.org/wiki/International_Organization_for_Standardization) (ISO/IEC 23270:2006). C# is one of the programming languages designed for the [Common Language Infrastructure](https://en.wikipedia.org/wiki/Common_Language_Infrastructure).

C# is a general-purpose, object-oriented programming language. Its development team is led by [Anders Hejlsberg](https://en.wikipedia.org/wiki/Anders_Hejlsberg). The most recent version is C# 7.2, which was released in 2017 along with Visual Studio 2017 version 15.5.

By design, C# is the programming language that most directly reflects the underlying [Common Language Infrastructure](https://en.wikipedia.org/wiki/Common_Language_Infrastructure) (CLI). Most of its intrinsic types correspond to value-types implemented by the CLI framework. However, the language specification does not state the code generation requirements of the compiler: that is, it does not state that a C# compiler must target a Common Language Runtime, or generate [Common Intermediate Language](https://en.wikipedia.org/wiki/Common_Intermediate_Language) (CIL), or generate any other specific format. Theoretically, a C# compiler could generate machine code like traditional compilers of C++ or [FORTRAN](https://en.wikipedia.org/wiki/Fortran).

C# has the ability to utilize LINQ through the Microsoft.NET Framework with the IEnumerable Interface a developer can query any .NET collection class, XML documents, ADO.NET datasets, and SQL databases. There are some advantages to using LINQ in C# and they are as follows: intellisense support, strong filtering capabilities, type safety with compile error checking ability, and brings consistency for querying data over a variety of sources. There are several different language structures that can be utilized with C# with LINQ and they are query expressions, lambda expressions, anonymous types, implicitly typed variables, extension methods, and object initializers.

**Features:**

### **Portability**

By design, C# is the programming language that most directly reflects the underlying [Common Language Infrastructure](https://en.wikipedia.org/wiki/Common_Language_Infrastructure) (CLI). Most of its intrinsic types correspond to value-types implemented by the CLI framework. However, the language specification does not state the code generation requirements of the compiler: that is, it does not state that a C# compiler must target a Common Language Runtime, or generate [Common Intermediate Language](https://en.wikipedia.org/wiki/Common_Intermediate_Language) (CIL), or generate any other specific format.

### **Methods and functions**

Methods in programming language are the members of a class in a project, some methods have signatures and some don't have signatures. Methods can be void or can return something like string, integer, double, decimal, float and bool. If a method is void it means that the method does not return any data type.

**Property**

C# provides [properties](https://en.wikipedia.org/wiki/Property_(programming)) as [syntactic sugar](https://en.wikipedia.org/wiki/Syntactic_sugar) for a common pattern in which a pair of methods, [accessor (getter) and mutator (setter)](https://en.wikipedia.org/wiki/Mutator_method) encapsulate operations on a single [attribute](https://en.wikipedia.org/wiki/Attribute_(computing)) of a class. No redundant method signatures for the getter/setter implementations need be written, and the property may be accessed using attribute syntax rather than more verbose method calls.

### **Namespace**

A C# namespace provides the same level of code isolation as a Java package or a C++ namespace, with very similar rules and features to a package.

### **Memory access**

In C#, memory address pointers can only be used within blocks specifically marked as *unsafe*, and programs with unsafe code need appropriate permissions to run. Most object access is done through safe object references, which always either point to a "live" object or have the well-defined [null](https://en.wikipedia.org/wiki/Nullable_type) value; it is impossible to obtain a reference to a "dead" object (one that has been garbage collected), or to a random block of memory. An unsafe pointer can point to an instance of a value-type, array, string, or a block of memory allocated on a stack. Code that is not marked as unsafe can still store and manipulate pointers through the System.intptr type, but it cannot dereference them.

### **Exception**

[Checked exceptions](https://en.wikipedia.org/wiki/Checked_exceptions) are not present in C# (in contrast to Java). This has been a conscious decision based on the issues of scalability and versionability.

### **Polymorphism**

Unlike [C++](https://en.wikipedia.org/wiki/C%2B%2B), C# does not support [multiple inheritance](https://en.wikipedia.org/wiki/Multiple_inheritance), although a class can implement any number of interfaces. This was a design decision by the language's lead architect to avoid complication and simplify architectural requirements throughout CLI. When implementing multiple interfaces that contain a method with the same signature, C# allows implementing each method depending on which interface that method is being called through, or, like Java, allows implementing the method once, and have that be the one invocation on a call through any of the class's interfaces.

### **Language Integrated Query - LINQ**

C# has the ability to utilize LINQ through the Microsoft.NET Framework with the ienumerable Interface a developer can query any .NET collection class, XML documents, ADO.NET datasets, and SQL databases. There are some advantages to using LINQ in C# and they are as follows: intellisense support, strong filtering capabilities, type safety with compile error checking ability, and brings consistency for querying data over a variety of sources. There are several different language structures that can be utilized with C# with LINQ and they are query expressions, lambda expressions, anonymous types, implicitly typed variables, extension methods, and object initializers.

### **Functional programming**

Though primarily an imperative language, C# 2.0 offered limited support for functional programming through [first-class functions](https://en.wikipedia.org/wiki/First-class_functions) and closures in the form of anonymous delegates. C# 3.0 expanded support for functional programming with the introduction of a lightweight syntax for lambda expressions, extension methods (an affordance for modules), and a [list comprehension](https://en.wikipedia.org/wiki/List_comprehension) syntax in the form of a "query comprehension" language.

**ASP.NET:**

ASP.NET is an [open-source](https://en.wikipedia.org/wiki/Open_source) [server-side](https://en.wikipedia.org/wiki/Server-side_scripting) [web application framework](https://en.wikipedia.org/wiki/Web_application_framework) designed for [web development](https://en.wikipedia.org/wiki/Web_development) to produce [dynamic web pages](https://en.wikipedia.org/wiki/Dynamic_web_page). It was developed by [Microsoft](https://en.wikipedia.org/wiki/Microsoft) to allow [programmers](https://en.wikipedia.org/wiki/Programmer) to build dynamic [web sites](https://en.wikipedia.org/wiki/Web_site), [web applications](https://en.wikipedia.org/wiki/Web_application) and [web services](https://en.wikipedia.org/wiki/Web_service).

It was first released in January 2002 with version 1.0 of the [.NET Framework](https://en.wikipedia.org/wiki/.NET_Framework), and is the successor to Microsoft's [Active Server Pages](https://en.wikipedia.org/wiki/Active_Server_Pages) (ASP) technology. ASP.NET is built on the [Common Language Runtime](https://en.wikipedia.org/wiki/Common_Language_Runtime) (CLR), allowing programmers to write ASP.NET code using any supported [.NET language](https://en.wikipedia.org/wiki/List_of_CLI_languages). The ASP.NET [SOAP](https://en.wikipedia.org/wiki/SOAP) extension framework allows ASP.NET components to process SOAP messages.

ASP.NET's successor is [ASP.NET Core](https://en.wikipedia.org/wiki/ASP.NET_Core). It is a re-implementation of ASP.NET as a modular [web framework](https://en.wikipedia.org/wiki/Web_framework), together with other frameworks like [Entity Framework](https://en.wikipedia.org/wiki/Entity_Framework). The new framework uses the new open-source [.NET Compiler Platform](https://en.wikipedia.org/wiki/.NET_Compiler_Platform) (codename "Roslyn") and is [cross platform](https://en.wikipedia.org/wiki/Cross_platform). [ASP.NET MVC](https://en.wikipedia.org/wiki/ASP.NET_MVC), ASP.NET Web API, and ASP.NET Web Pages (a platform using only [Razor](https://en.wikipedia.org/wiki/ASP.NET_Razor) pages) have merged into a unified MVC 6.

**Characteristics:**

ASP.NET Web pages, known officially as Web Forms, are the main building blocks for application development in ASP.NET. There are two basic methodologies for Web Forms, a web application format and a web site format. Web applications need to be compiled before deployment, while web sites structures allows the user to copy the files directly to the server without prior compilation. Web forms are contained in files with a ".aspx" extension; these files typically contain static ([X](https://en.wikipedia.org/wiki/XHTML)) [HTML](https://en.wikipedia.org/wiki/HTML) markup or component markup. The component markup can include server-side Web Controls and User Controls that have been defined in the framework or the web page. For example, a textbox component can be defined on a page as <asp:textbox id='myid' runat='server'>, which is rendered into a html input box. Additionally, dynamic code, which runs on the server, can be placed in a page within a block <% -- dynamic code -- %>, which is similar to other Web development technologies such as [PHP](https://en.wikipedia.org/wiki/PHP), [JSP](https://en.wikipedia.org/wiki/JavaServer_Pages), and [ASP](https://en.wikipedia.org/wiki/Active_Server_Pages). With [ASP.NET Framework 2.0](https://en.wikipedia.org/wiki/ASP.NET_3.5), Microsoft introduced a new *code-behind* model that lets static text remain on the .aspx page, while dynamic code remains in an .aspx.vb or .aspx.cs or .aspx.fs file (depending on the programming language used).

**Code Behind Model:**

[Microsoft](https://en.wikipedia.org/wiki/Microsoft) recommends dealing with dynamic program code by using the code-behind model, which places this code in a separate file or in a specially designated script tag. Code-behind files typically have names like "*MyPage.aspx.cs"* or "*MyPage.aspx.vb"* while the page file is *MyPage.aspx* (same filename as the page file (ASPX), but with the final extension denoting the page language). This practice is automatic in [Visual Studio](https://en.wikipedia.org/wiki/Visual_Studio) and other [IDEs](https://en.wikipedia.org/wiki/Integrated_development_environment), though the user can change the code-behind page. Also, in the web application format, the pagename.aspx.cs is a partial class that is linked to the pagename.designer.cs file. The designer file is a file that is autogenerated from the ASPX page and allows the programmer to reference components in the ASPX page from the CS page without having to declare them manually, as was necessary in ASP.NET versions before version 2. When using this style of programming, the developer writes code to respond to different events, such as the page being loaded, or a control being clicked, rather than a procedural walkthrough of the document.

ASP.NET's code-behind model marks a departure from Classic ASP in that it encourages developers to build applications with [separation of presentation and content](https://en.wikipedia.org/wiki/Separation_of_presentation_and_content) in mind. In theory, this would allow a Web designer, for example, to focus on the design markup with less potential for disturbing the programming code that drives it. This is similar to the separation of the controller from the view in [model–view–controller](https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller) (MVC) frameworks.

**Visual Studio:**

Microsoft Visual Studio is an [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) from [Microsoft](https://en.wikipedia.org/wiki/Microsoft). It is used to develop [computer programs](https://en.wikipedia.org/wiki/Computer_program), as well as [web sites](https://en.wikipedia.org/wiki/Web_site), [web apps](https://en.wikipedia.org/wiki/Web_app), [web services](https://en.wikipedia.org/wiki/Web_service) and [mobile apps](https://en.wikipedia.org/wiki/Mobile_app). Visual Studio uses Microsoft software development platforms such as [Windows API](https://en.wikipedia.org/wiki/Windows_API), [Windows Forms](https://en.wikipedia.org/wiki/Windows_Forms), [Windows Presentation Foundation](https://en.wikipedia.org/wiki/Windows_Presentation_Foundation), [Windows Store](https://en.wikipedia.org/wiki/Windows_Store) and [Microsoft Silverlight](https://en.wikipedia.org/wiki/Microsoft_Silverlight). It can produce both [native code](https://en.wikipedia.org/wiki/Native_code) and [managed code](https://en.wikipedia.org/wiki/Managed_code).

Visual Studio includes a [code editor](https://en.wikipedia.org/wiki/Code_editor) supporting [IntelliSense](https://en.wikipedia.org/wiki/IntelliSense) (the [code completion](https://en.wikipedia.org/wiki/Code_completion) component) as well as [code refactoring](https://en.wikipedia.org/wiki/Code_refactoring). [The integrated debugger](https://en.wikipedia.org/wiki/Microsoft_Visual_Studio_Debugger) works both as a source-level debugger and a machine-level debugger. Other built-in tools include a [code profiler](https://en.wikipedia.org/wiki/Profiling_(computer_programming)), forms designer for building [GUI](https://en.wikipedia.org/wiki/GUI) applications, [web designer](https://en.wikipedia.org/wiki/Web_designer), [class](https://en.wikipedia.org/wiki/Class_(computing)) designer, and [database schema](https://en.wikipedia.org/wiki/Database_schema) designer. It accepts plug-ins that enhance the functionality at almost every level—including adding support for [source control](https://en.wikipedia.org/wiki/Source_control) systems (like [Subversion](https://en.wikipedia.org/wiki/Subversion_(software))) and adding new toolsets like editors and visual designers for [domain-specific languages](https://en.wikipedia.org/wiki/Domain-specific_language) or toolsets for other aspects of the [software development lifecycle](https://en.wikipedia.org/wiki/Software_development_lifecycle) (like the [Team Foundation Server](https://en.wikipedia.org/wiki/Team_Foundation_Server) client: Team Explorer).

Visual Studio supports 36 different [programming languages](https://en.wikipedia.org/wiki/Programming_language) and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include [C](https://en.wikipedia.org/wiki/C_(programming_language)), [C++](https://en.wikipedia.org/wiki/C%2B%2B), [C++/CLI](https://en.wikipedia.org/wiki/C%2B%2B/CLI), [Visual Basic .NET](https://en.wikipedia.org/wiki/Visual_Basic_.NET), [C#](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)), [F#](https://en.wikipedia.org/wiki/F_Sharp_(programming_language)), [JavaScript](https://en.wikipedia.org/wiki/JavaScript), [TypeScript](https://en.wikipedia.org/wiki/TypeScript), [XML](https://en.wikipedia.org/wiki/XML), [XSLT](https://en.wikipedia.org/wiki/XSLT), [HTML](https://en.wikipedia.org/wiki/HTML) and [CSS](https://en.wikipedia.org/wiki/Cascading_Style_Sheets). Support for other languages such as [Python](https://en.wikipedia.org/wiki/Python_(programming_language)),[[8]](https://en.wikipedia.org/wiki/Microsoft_Visual_Studio#cite_note-8) [Ruby](https://en.wikipedia.org/wiki/Ruby_(programming_language)), [Node.js](https://en.wikipedia.org/wiki/Node.js), and [M](https://en.wikipedia.org/wiki/MUMPS) among others is available via [plug-ins](https://en.wikipedia.org/wiki/Plug-in_(computing)). [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) (and [J#](https://en.wikipedia.org/wiki/J_Sharp)) were supported in the past.The most basic edition of Visual Studio, the Community edition, is available free of charge.

**Ultimate 2010:**

Visual Studio Ultimate 2010 replaces Visual Studio 2008 Team Suite. It includes new modeling tools, such as the Architecture Explorer, which graphically displays projects and classes and the relationships between them. It supports [UML](https://en.wikipedia.org/wiki/Unified_Modeling_Language) activity diagram, component diagram, (logical) class diagram, sequence diagram, and use case diagram. Visual Studio Ultimate 2010 also includes Test Impact Analysis which provides hints on which test cases are impacted by modifications to the source code, without actually running the test cases. This speeds up testing by avoiding running unnecessary test cases.

Visual Studio Ultimate 2010 also includes a [historical debugger](https://en.wikipedia.org/wiki/Historical_debugger) for [managed code](https://en.wikipedia.org/wiki/Managed_code) called IntelliTrace. Unlike a traditional debugger that records only the currently active stack, IntelliTrace records all events, such as prior function calls, method parameters, events and exceptions. This allows the code execution to be rewound in case a breakpoint was not set where the error occurred. Debugging with IntelliTrace causes the application to run more slowly than debugging without it, and uses more memory as additional data needs to be recorded. Microsoft allows configuration of how much data should be recorded, in effect, allowing developers to balance the speed of execution and resource usage. The Lab Management component of Visual Studio Ultimate 2010 uses virtualization to create a similar execution environment for testers and developers. The [virtual machines](https://en.wikipedia.org/wiki/Virtual_machine) are tagged with checkpoints which can later be investigated for issues, as well as to reproduce the issue. Visual Studio Ultimate 2010 also includes the capability to record test runs that capture the specific state of the operating environment as well as the precise steps used to run the test. These steps can then be played back to reproduce issues.

**SQL Server**

Microsoft SQL Server is a [relational database management system](https://en.wikipedia.org/wiki/Relational_database_management_system) developed by [Microsoft](https://en.wikipedia.org/wiki/Microsoft). As a [database server](https://en.wikipedia.org/wiki/Database_server), it is a [software product](https://en.wikipedia.org/wiki/Software_product) with the primary function of storing and retrieving data as requested by other [software applications](https://en.wikipedia.org/wiki/Software_application)—which may run either on the same computer or on another computer across a network (including the Internet).

Microsoft markets at least a dozen different editions of Microsoft SQL Server, aimed at different audiences and for workloads ranging from small single-machine applications to large Internet-facing applications with many [concurrent users](https://en.wikipedia.org/wiki/Concurrent_user).

**SQL**

[Microsoft Visual Studio](https://en.wikipedia.org/wiki/Microsoft_Visual_Studio) includes native support for data programming with Microsoft SQL Server. It can be used to write and [debug code](https://en.wikipedia.org/wiki/Debug_code) to be executed by SQL CLR. It also includes a data designer that can be used to graphically create, view or edit database schemas. Queries can be created either visually or using code. SSMS 2008 onwards, provides [intelli sense](https://en.wikipedia.org/wiki/Intellisense) for SQL queries as well.

SQLCMD is a command line application that comes with Microsoft SQL Server, and exposes the management features of SQL Server. It allows SQL queries to be written and executed from the command prompt. It can also act as a [scripting language](https://en.wikipedia.org/wiki/Scripting_language) to create and run a set of SQL statements as a script. Such scripts are stored as a .sql file, and are used either for management of databases or to create the database schema during the deployment of a database.

SQLCMD was introduced with SQL Server 2005 and this continues with SQL Server 2012, 2014 and 2016. Its predecessor for earlier versions was OSQL and ISQL, which is functionally equivalent as it pertains to TSQL execution, and many of the command line parameters are identical, although SQLCMD adds extra versatility.

**Platforms and interfaces**

Many programming languages with language-specific APIs include libraries for accessing SQL databases. These include SQL Connector/Net for integration with Microsoft's Visual Studio languages such as C# and VB are most commonly used and the JDBC driver for Java. In addition, an ODBC interface called Modoc allows additional programming languages that support the ODBC interface to communicate with a SQL database, such as ASP or ColdFusion. The SQL server and official libraries are mostly implemented in ANSI C/ANSI C++.

**Bootstrap Framework:**

Bootstrap is a [free and open-source](https://en.wikipedia.org/wiki/Free_and_open-source_software) front-end [web framework](https://en.wikipedia.org/wiki/Web_framework) for designing [websites](https://en.wikipedia.org/wiki/Website) and [web applications](https://en.wikipedia.org/wiki/Web_application). It contains [HTML](https://en.wikipedia.org/wiki/HTML)- and [CSS](https://en.wikipedia.org/wiki/CSS)-based design templates for [typography](https://en.wikipedia.org/wiki/Typography), forms, buttons, navigation and other interface components, as well as optional [javascript](https://en.wikipedia.org/wiki/JavaScript) extensions. Unlike many web frameworks, it concerns itself with [front-end development](https://en.wikipedia.org/wiki/Front-end_web_development) only.

Bootstrap is the second most-starred project on [github](https://en.wikipedia.org/wiki/GitHub), with more than 111,600 stars and 51,500 forks.

**Features:**

Bootstrap 3 supports the latest versions of the [Google Chrome](https://en.wikipedia.org/wiki/Google_Chrome), [Firefox](https://en.wikipedia.org/wiki/Firefox), [Internet Explorer](https://en.wikipedia.org/wiki/Internet_Explorer), [Opera](https://en.wikipedia.org/wiki/Opera_(web_browser)), and [Safari](https://en.wikipedia.org/wiki/Safari_(web_browser)) (except on Windows). It additionally supports back to [IE8](https://en.wikipedia.org/wiki/Internet_Explorer_8) and the latest [Firefox](https://en.wikipedia.org/wiki/Firefox) Extended Support Release (ESR).[[12]](https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)#cite_note-supported-browsers-12)

Since 2.0, Bootstrap supports [responsive web design](https://en.wikipedia.org/wiki/Responsive_Web_Design). This means the layout of web pages adjusts dynamically, taking into account the characteristics of the device used (desktop, tablet, mobile phone).

Starting with version 3.0, Bootstrap adopted a [mobile-first design](https://en.wikipedia.org/wiki/Mobile-first_design) philosophy, emphasizing responsive design by default. The version 4.0 alpha release added [Sass](https://en.wikipedia.org/wiki/Sass_(stylesheet_language)) and [flex box](https://en.wikipedia.org/wiki/CSS_Flex_Box_Layout) support.

**Structure and Function:**

Bootstrap is modular and consists of a series of [less stylesheets](https://en.wikipedia.org/wiki/Less_(stylesheet_language)) that implement the various components of the toolkit. These stylesheets are generally compiled into a bundle and included in web pages, but individual components can be included or removed. Bootstrap provides a number of configuration variables that control things such as color and padding of various components.

Since Bootstrap 2, the Bootstrap documentation has included a customization wizard which generates a customized version of Bootstrap based on the requested components and various settings.

As of Bootstrap 4, [Sass](https://en.wikipedia.org/wiki/Sass_(stylesheet_language)) is used instead of [less](https://en.wikipedia.org/wiki/Less_(stylesheet_language)) for the stylesheets.

Each Bootstrap component consists of an HTML structure, CSS declarations, and in some cases accompanying javascript code.

Grid system and responsive design comes standard with an 1170 pixel wide [grid layout](https://en.wikipedia.org/wiki/CSS_Grid_Layout). Alternatively, the developer can use a variable-width layout. For both cases, the toolkit has four variations to make use of different resolutions and types of devices: mobile phones, portrait and landscape, tablets and pcs with low and high resolution. Each variation adjusts the width of the columns.

### **Stylesheets**

Bootstrap provides a set of stylesheets that provide basic style definitions for all key HTML components. These provide a uniform, modern appearance for formatting text, tables and form elements.

### **Re-usable components**

In addition to the regular HTML elements, Bootstrap contains other commonly used interface elements. The components are implemented as CSS classes, which must be applied to certain HTML elements in a page.

### **Javascript components**

Bootstrap comes with several javascript components in the form of [jquery](https://en.wikipedia.org/wiki/JQuery) plugins. They provide additional user interface elements such as dialog boxes, tooltips, and carousels. They also extend the functionality of some existing interface elements, including for example an auto-complete function for input fields. In version 1.3, the following javascript plugins are supported: Modal, Dropdown, Scrollspy, Tab, Tooltip, Popover, Alert, Button, Collapse, Carousel and Typeahead.

**Jquery:**

Jquery is a cross-platform javascript library designed to simplify the client side scripting of HTML. It is [free, open-source software](https://en.wikipedia.org/wiki/Free_and_open_source_software) using the permissive [MIT License](https://en.wikipedia.org/wiki/MIT_License). [Web](https://en.wikipedia.org/wiki/World_Wide_Web) analysis indicates that it is the most widely deployed JavaScript library by a large margin.

Jquery's syntax is designed to make it easier to navigate a document, select dom elements, create animations, handle events, and develop ajax applications. jQuery also provides capabilities for developers to create [plug-ins](https://en.wikipedia.org/wiki/Plug-in_(computing)) on top of the JavaScript library. This enables developers to create [abstractions](https://en.wikipedia.org/wiki/Abstraction_(computer_science)) for low-level interaction and animation, advanced effects and high-level, theme able widgets. The modular approach to the jQuery library allows the creation of powerful [dynamic web pages](https://en.wikipedia.org/wiki/Dynamic_web_page) and Web applications.

The set of [jQuery core features](https://en.wikipedia.org/wiki/JQuery#Features)—DOM element selections, traversal and manipulation—enabled by its selector engine (named "Sizzle" from v1.3), created a new "programming style", fusing algorithms and DOM data structures. This style influenced the architecture of other [JavaScript frameworks](https://en.wikipedia.org/wiki/Comparison_of_JavaScript_frameworks) like [YUI v3](https://en.wikipedia.org/wiki/YUI_Library) and [Dojo](https://en.wikipedia.org/wiki/Dojo_Toolkit), later stimulating the creation of the standard Selectors API.

[Microsoft](https://en.wikipedia.org/wiki/Microsoft) and [Nokia](https://en.wikipedia.org/wiki/Nokia) bundle jQuery on their platforms. Microsoft includes it with [Visual Studio](https://en.wikipedia.org/wiki/Microsoft_Visual_Studio) for use within Microsoft's [ASP.NET AJAX](https://en.wikipedia.org/wiki/ASP.NET_AJAX) and [ASP.NET MVC](https://en.wikipedia.org/wiki/ASP.NET_MVC) frameworks while Nokia has integrated it into the Web Run-Time widget development platform.

Jquery, at its core, is a [Document Object Model](https://en.wikipedia.org/wiki/Document_Object_Model) (DOM) manipulation library. The DOM is a tree-structure representation of all the elements of a Web page. Jquery simplifies the syntax for finding, selecting, and manipulating these DOM elements. For example, jquery can be used for finding an element in the document with a certain property (e.g. All elements with an [h1](https://en.wikipedia.org/wiki/HTML_element#heading) tag), changing one or more of its attributes (e.g. Color, visibility), or making it respond to an event (e.g. A mouse click).

Jquery also provides a paradigm for event handling that goes beyond basic DOM element selection and manipulation. The event assignment and the event callback function definition are done in a single step in a single location in the code. Jquery also aims to incorporate other highly used javascript functionality (e.g. Fade ins and fade outs when hiding elements, animations by manipulating [CSS](https://en.wikipedia.org/wiki/Cascading_Style_Sheets) properties).

The principles of developing with jquery are:

* Separation of javascript and HTML: The jquery library provides simple syntax for adding [event](https://en.wikipedia.org/wiki/Event_(computing)) handlers to the [DOM](https://en.wikipedia.org/wiki/Document_Object_Model) using javascript, rather than adding [HTML event attributes](https://en.wikipedia.org/wiki/HTML_attribute#Event_attributes) to call javascript functions. Thus, it encourages developers to [completely separate](https://en.wikipedia.org/wiki/Separation_of_concerns) javascript code from HTML markup.
* Brevity and clarity: jquery promotes brevity and clarity with features like chainable functions and shorthand function names.
* Elimination of cross-browser incompatibilities: The javascript engines of different browsers differ slightly so javascript code that works for one browser may not work for another. Like other javascript toolkits, jquery handles all these cross-browser inconsistencies and provides a consistent interface that works across different browsers.
* Extensibility: New events, elements, and methods can be easily added and then reused as a plugin.

**TESTING OF PRODUCT**

**Testing of Product:**

System testing is the stage of implementation, which aimed at ensuring that system works accurately and efficiently before the live operation commence. Testing is the process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an error. A successful test is one that answers a yet undiscovered error.

Testing is vital to the success of the system.  System testing makes a logical assumption that if all parts of the system are correct, the goal will be successfully achieved.  The candidate system is subject to variety of tests-on-line response, Volume Street, recovery and security and usability test.  A series of tests are performed before the system is ready for the user acceptance testing.  Any engineered product can be tested in one of the following ways.  Knowing the specified function that a product has been designed to from, test can be conducted to demonstrate each function is fully operational.  Knowing the internal working of a product, tests can be conducted to ensure that “al gears mesh”, that is the internal operation of the product performs according to the specification and all internal components have been adequately exercised.

**UNIT TESTING:**

Unit testing is the testing of each module and the integration of the overall system is done.  Unit testing becomes verification efforts on the smallest unit of software design in the module.  This is also known as ‘module testing’.  The modules of the system are tested separately.  This testing is carried out during the programming itself.  In this testing step, each model is found to be working satisfactorily as regard to the expected output from the module.  There are some validation checks for the fields.  For example, the validation check is done for verifying the data given by the user where both format and validity of the data entered is included.  It is very easy to find error and debug the system.

**INTEGRATION TESTING:**

Data can be lost across an interface, one module can have an adverse effect on the other sub function, when combined, may not produce the desired major function.  Integrated testing is systematic testing that can be done with sample data.  The need for the integrated test is to find the overall system performance.

There are two types of integration testing. They are:

1. Top-down integration testing.
2. Bottom-up integration testing.

**WHITE BOX TESTING:**

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

White-box test design techniques include the following [code coverage](https://en.wikipedia.org/wiki/Code_coverage) criteria:

* [Control flow](https://en.wikipedia.org/wiki/Control_flow) testing
* Data flow testing
* Branch testing
* Statement coverage
* Decision coverage
* [Modified condition/decision coverage](https://en.wikipedia.org/wiki/Modified_condition/decision_coverage)
* Prime path testing
* Path testing

**BLACK BOX TESTING:**

* + Black box testing is done to find incorrect or missing function
  + Interface error
  + Errors in external database access
  + Performance errors
  + Initialization and termination errors

In ‘functional testing’, is performed to validate an application conforms to its specifications of correctly performs all its required functions. So this testing is also called ‘black box testing’.  It tests the external behaviour of the system.  Here the engineered product can be tested knowing the specified function that a product has been designed to perform, tests can be conducted to demonstrate that each function is fully operational.

**Black box testing techniques**

Following are some techniques that can be used for designing black box tests.

* Equivalence partitioning: It is a software test design technique that involves dividing input values into valid and invalid partitions and selecting representative values from each partition as test data.
* Boundary Value Analysis: It is a software test design technique that involves determination of boundaries for input values and selecting values that are at the boundaries and just inside/ outside of the boundaries as test data.
* Cause Effect Graphing: It is a software test design technique that involves identifying the cases (input conditions) and effects (output conditions), producing a Cause-Effect Graph, and generating test cases accordingly.

**Black box testing advantages**

* Tests are done from a user’s point of view and will help in exposing discrepancies in the specifications.
* Tester need not know programming languages or how the software has been implemented.
* Tests can be conducted by a body independent from the developers, allowing for an objective perspective and the avoidance of developer-bias.
* Test cases can be designed as soon as the specifications are complete.

**VALIDATION TESTING:**

After the culmination of black box testing, software is completed assembly as a package, interfacing errors have been uncovered and corrected and final series of software validation tests begin validation testing can be defined as many, but a single definition is that validation succeeds when the software functions in a manner that can be reasonably expected by the customer. In [software project management](https://en.wikipedia.org/wiki/Software_project_management), [software testing](https://en.wikipedia.org/wiki/Software_testing), and [software engineering](https://en.wikipedia.org/wiki/Software_engineering), verification and validation (V&V) is the process of checking that a software system meets specifications and that it fulfills its intended purpose. It may also be referred to as [software quality control](https://en.wikipedia.org/wiki/Software_quality_control). It is normally the responsibility of [software testers](https://en.wikipedia.org/wiki/Software_testing) as part of the [software development lifecycle](https://en.wikipedia.org/wiki/Software_development_process). In simple terms, software verification is: "Assuming we should build X, does our software achieve its goals without any bugs or gaps?" On the other hand, software validation is: "Was X what we should have built? Does X meet the high level requirements?"

# **USER ACCEPTANCE TESTING:**

User acceptance of the system is the key factor for the success of the system.  The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system at the time of developing changes whenever required. In software development, user acceptance testing (UAT) - also called beta testing, application testing, and [end user](http://whatis.techtarget.com/definition/end-user) testing - is a phase of software development in which the software is tested in the "real world" by the intended audience. UAT can be done by in-house testing in which volunteers or paid test subjects use the software or, more typically for widely-distributed software, by making the test version available for downloading and free trial over the Web. The experiences of the early users are forwarded back to the developers who make final changes before releasing the software commercially.

**STRESS TESTING:**

Stress testing (sometimes called torture testing) is a form of deliberately intense or thorough testing used to determine the stability of a given system or entity. It involves testing beyond normal operational capacity, often to a breaking point, in order to observe the results. Reasons can include:

* to determine breaking points or safe usage limits
* to confirm mathematical model is accurate enough in predicting breaking points or safe usage limits
* to confirm intended specifications are being met
* to determine [modes of failure](https://en.wikipedia.org/wiki/Failure_cause) (how exactly a system fails)
* to test stable operation of a part or system outside standard usage

[Reliability engineers](https://en.wikipedia.org/wiki/Reliability_engineering) often test items under expected stress or even under accelerated stress in order to determine the operating life of the item or to determine modes of failure. The term "[stress](https://en.wikipedia.org/wiki/Stress_(mechanics))" may have a more specific meaning in certain industries, such as material sciences, and therefore stress testing may sometimes have a technical meaning – one example is in [fatigue testing](https://en.wikipedia.org/wiki/Fatigue_(material)) for materials.

**PERFORMANCE TESTING:**

In [software engineering](https://en.wikipedia.org/wiki/Software_engineering), performance testing is in general, a [testing](https://en.wikipedia.org/wiki/Software_testing) practice performed to determine how a [system](https://en.wikipedia.org/wiki/System) performs in terms of responsiveness and stability under a particular workload. It can also serve to investigate, measure, validate or verify other [quality](https://en.wikipedia.org/wiki/Quality_(business)) [attributes](https://en.wikipedia.org/wiki/Attribute_(computing)) of the system, such as [scalability](https://en.wikipedia.org/wiki/Scalability), [reliability](https://en.wiktionary.org/wiki/reliability) and resource usage.

Performance testing, a subset of [performance engineering](https://en.wikipedia.org/wiki/Performance_engineering), is a [computer science](https://en.wikipedia.org/wiki/Computer_science) practice which strives to build performance standards into the implementation, design and architecture of a system.

## **Testing types**

### **Load testing**

[Load testing](https://en.wikipedia.org/wiki/Load_testing) is the simplest form of performance testing. A load test is usually conducted to understand the behaviour of the system under a specific expected load. This load can be the expected concurrent number of users on the [application](https://en.wikipedia.org/wiki/Application_software) performing a specific number of [transactions](https://en.wikipedia.org/wiki/Transaction_processing) within the set duration. This test will give out the response times of all the important business critical transactions. The [database](https://en.wikipedia.org/wiki/Database), [application server](https://en.wikipedia.org/wiki/Application_server), etc. are also monitored during the test, this will assist in identifying [bottlenecks](https://en.wikipedia.org/wiki/Bottleneck_(software)) in the application software and the hardware that the software is installed on.

### **Stress testing**

[Stress testing](https://en.wikipedia.org/wiki/Stress_testing) is normally used to understand the upper limits of capacity within the system. This kind of test is done to determine the system's robustness in terms of extreme load and helps application administrators to determine if the system will perform sufficiently if the current load goes well above the expected maximum.

### **Soak testing**

[Soak testing](https://en.wikipedia.org/wiki/Soak_testing), also known as endurance testing, is usually done to determine if the system can sustain the continuous expected load. During soak tests, memory utilization is monitored to detect potential leaks. Also important, but often overlooked is performance degradation, i.e. to ensure that the throughput and/or response times after some long period of sustained activity are as good as or better than at the beginning of the test. It essentially involves applying a significant load to a system for an extended, significant period of time. The goal is to discover how the system behaves under sustained use.

### **Spike testing**

Spike testing is done by suddenly increasing or decreasing the load generated by a very large number of users, and observing the behaviour of the system. The goal is to determine whether performance will suffer, the system will fail, or it will be able to handle dramatic changes in load.

### **Configuration testing**

Rather than testing for performance from a load perspective, tests are created to determine the effects of configuration changes to the system's components on the system's performance and behaviour. A common example would be experimenting with different methods of [load-balancing](https://en.wikipedia.org/wiki/Load_balancing_(computing)).

### **Isolation testing**

Isolation testing is not unique to performance testing but involves repeating a test execution that resulted in a system problem. Such testing can often isolate and confirm the fault domain.

### **Internet testing**

This is a relatively new form of performance testing when global applications such as Facebook, Google and Wikipedia, are performance tested from load generators that are placed on the actual target continent whether physical machines or cloud VMs. These tests usually requires an immense amount of preparation and monitoring to be executed successfully.

# **OUTPUT TESTING:**

After performing the validation testing, the next step is output asking the user about the format required testing of the proposed system, since no system could be useful if it does not produce the required output in the specific format.  The output displayed or generated by the system under consideration.  Here the output format is considered in two ways.  One is screen and the other is printed format.  The output format on the screen is found to be correct as the format was designed in the system phase according to the user needs.  For the hard copy also output comes out as the specified requirements by the user. Hence the output testing does not result in any connection in the system.

**System Implementation:**

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended users and the operation of the system. The people are not sure that the software is meant to make their job easier.

* The active user must be aware of the benefits of using the system
* Their confidence in the software built up
* Proper guidance is impaired to the user so that he is comfortable in using the application

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not running on the server, the actual processes will not take place.

**User Training:**

To achieve the objectives and benefits expected from the proposed system it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for education and training is more and more important. Education is complementary to training. It brings life to formal training by explaining the background to the resources for them. Education involves creating the right atmosphere and motivating user staff. Education information can make training more interesting and more understandable.

**Training on the Application Software:**

After providing the necessary basic training on the computer awareness, the users will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design, type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the data entered. This training may be different across different user groups and across different levels of hierarchy.

**Operational Documentation:**

Once the implementation plan is decided, it is essential that the user of the system is made familiar and comfortable with the environment. A documentation providing the whole operations of the system is being developed. Useful tips and guidance is given inside the application itself to the user. The system is developed user friendly so that the user can work the system from the tips given in the application itself.

**System Maintenance:**

The maintenance phase of the software cycle is the time in which software performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is to make adaptable to the changes in the system environment. There may be social, technical and other environmental changes, which affect a system which is being implemented. Software product enhancements may involve providing new functional capabilities, improving user displays and mode of interaction, upgrading the performance characteristics of the system. So only thru proper system maintenance procedures, the system can be adapted to cope up with these changes. Software maintenance is of course, far more than “finding mistakes”.

**Corrective Maintenance:**

The first maintenance activity occurs because it is unreasonable to assume that software testing will uncover all latent errors in a large software system. During the use of any large program, errors will occur and be reported to the developer. The process that includes the diagnosis and correction of one or more errors is called Corrective Maintenance.

**Adaptive Maintenance:**

The second activity that contributes to a definition of maintenance occurs because of the rapid change that is encountered in every aspect of computing. Therefore Adaptive maintenance termed as an activity that modifies software to properly interfere with a changing environment is both necessary and commonplace

**Perceptive Maintenance:**

The third activity that may be applied to a definition of maintenance occurs when a software package is successful. As the software is used, recommendations for new capabilities, modifications to existing functions, and general enhancement are received from users. To satisfy requests in this category, Perceptive maintenance is performed. This activity accounts for the majority of all efforts expended on software maintenance.

**Preventive Maintenance:**

The fourth maintenance activity occurs when software is changed to improve future maintainability or reliability, or to provide a better basis for future enhancements. Often called preventive maintenance, this activity is characterized by reverse engineering and re-engineering techniques.

**Coding & Screenshots:**

View Question.aspx.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data.SqlClient;

using System.Data;

public partial class viewques : System.Web.UI.Page

{

SqlConnection con = new SqlConnection("Data Source=EGC-PC\\SQLEXPRESS;Initial Catalog=AutomaticquesSystem;Integrated Security=True");

protected void Form\_Load(object sender, EventArgs e)

{

}

protected void Button1\_Click(object sender, EventArgs e)

{

}

protected void DropDownList1\_SelectedIndexChanged(object sender, EventArgs e)

{

}

protected void b1\_Click(object sender, EventArgs e)

{

con.Open();

string a = d2.SelectedItem.ToString();

string b = d1.SelectedItem.ToString();

Session["Code"] = d1.SelectedItem.ToString();

if (a.Equals("Pattern 1"))

{

string sql, sql1,sql2,sql3,sql4,sql5;

string sq = "truncate table p1\_2marks";

SqlCommand cmd1 = new SqlCommand(sq, con);

cmd1.ExecuteNonQuery();

sql = "insert into p1\_2marks (question,mark,type,code) select Questions, Mark\_Allocation, Question\_Type, Subject\_Code from StaffQuestions1 where Mark\_Allocation = '2' And Subject\_Code = '" + b + "'";

SqlCommand cmd = new SqlCommand(sql, con);

cmd.ExecuteNonQuery();

string sq1 = "truncate table p1\_10mark";

SqlCommand cmd2 = new SqlCommand(sq1, con);

cmd2.ExecuteNonQuery();

sql1 = "insert into p1\_10mark (question,mark,type,code) select Questions, Mark\_Allocation, Question\_Type, Subject\_Code from StaffQuestions1 where Mark\_Allocation = '10' And Subject\_Code = '" + b + "'";

SqlCommand cmd3 = new SqlCommand(sql1, con);

cmd3.ExecuteNonQuery();

sql2 = "truncate table p1\_dist2";

SqlCommand cmd4 = new SqlCommand(sql2, con);

cmd4.ExecuteNonQuery();

sql3 = "truncate table p1\_dist10";

SqlCommand cmd5 = new SqlCommand(sql3, con);

cmd5.ExecuteNonQuery();

sql4 = "insert into p1\_dist2 (question,mark,type,code) select distinct(question),mark,type,code from p1\_2marks";

SqlCommand cmd6 = new SqlCommand(sql4, con);

cmd6.ExecuteNonQuery();

sql5 = "insert into p1\_dist10 (question,mark,type,code) select distinct(question),mark,type,code from p1\_10mark";

SqlCommand cmd7 = new SqlCommand(sql5, con);

cmd7.ExecuteNonQuery();

Response.Redirect("Q1.aspx");

}

else if (a.Equals("Pattern 2"))

{

string sql, sql1,sql2,sql3,sql4,sql5;

string sq = "truncate table p2\_2marks";

SqlCommand cmd1 = new SqlCommand(sq, con);

cmd1.ExecuteNonQuery();

sql = "insert into p2\_2marks (question,mark,type,code) select Questions, Mark\_Allocation, Question\_Type, Subject\_Code from StaffQuestions2 where Mark\_Allocation = '2' And Subject\_Code = '" + b + "'";

SqlCommand cmd = new SqlCommand(sql, con);

cmd.ExecuteNonQuery();

string sq1 = "truncate table p2\_12mark";

SqlCommand cmd2 = new SqlCommand(sq1, con);

cmd2.ExecuteNonQuery();

sql1 = "insert into p2\_12mark (question,mark,type,code) select Questions, Mark\_Allocation, Question\_Type, Subject\_Code from StaffQuestions2 where Mark\_Allocation = '12' And Subject\_Code = '" + b + "'";

SqlCommand cmd3 = new SqlCommand(sql1, con);

cmd3.ExecuteNonQuery();

sql2 = "truncate table p2\_dist2";

SqlCommand cmd4 = new SqlCommand(sql2, con);

cmd4.ExecuteNonQuery();

sql3 = "truncate table p2\_dist12";

SqlCommand cmd5 = new SqlCommand(sql3, con);

cmd5.ExecuteNonQuery();

sql4 = "insert into p2\_dist2 (question,mark,type,code) select distinct(question),mark,type,code from p2\_2marks";

SqlCommand cmd6 = new SqlCommand(sql4, con);

cmd6.ExecuteNonQuery();

sql5 = "insert into p2\_dist12 (question,mark,type,code) select distinct(question),mark,type,code from p2\_12mark";

SqlCommand cmd7 = new SqlCommand(sql5, con);

cmd7.ExecuteNonQuery();

Response.Redirect("Q2.aspx");

}

else if (a.Equals("Pattern 3"))

{

string sql, sql1,sql2,sql3,sql4,sql5;

string sq = "truncate table p3\_2marks";

SqlCommand cmd1 = new SqlCommand(sq, con);

cmd1.ExecuteNonQuery();

sql = "insert into p3\_2marks (question,mark,type,code) select Questions, Mark\_Allocation, Question\_Type, Subject\_Code from StaffQuestions3 where Mark\_Allocation = '2' And Subject\_Code = '" + b + "'";

SqlCommand cmd = new SqlCommand(sql, con);

cmd.ExecuteNonQuery();

string sq1 = "truncate table p3\_15mark";

SqlCommand cmd2 = new SqlCommand(sq1, con);

cmd2.ExecuteNonQuery();

sql1 = "insert into p3\_15mark (question,mark,type,code) select Questions, Mark\_Allocation, Question\_Type, Subject\_Code from StaffQuestions3 where Mark\_Allocation = '15' And Subject\_Code = '" + b + "'";

SqlCommand cmd3 = new SqlCommand(sql1, con);

cmd3.ExecuteNonQuery();

sql2 = "truncate table p3\_dist2";

SqlCommand cmd4 = new SqlCommand(sql2, con);

cmd4.ExecuteNonQuery();

sql3 = "truncate table p3\_dist15";

SqlCommand cmd5 = new SqlCommand(sql3, con);

cmd5.ExecuteNonQuery();

sql4 = "insert into p3\_dist2 (question,mark,type,code) select distinct(question),mark,type,code from p3\_2marks";

SqlCommand cmd6 = new SqlCommand(sql4, con);

cmd6.ExecuteNonQuery();

sql5 = "insert into p3\_dist15 (question,mark,type,code) select distinct(question),mark,type,code from p3\_15mark";

SqlCommand cmd7 = new SqlCommand(sql5, con);

cmd7.ExecuteNonQuery();

Response.Redirect("Q3.aspx");

}

else if (a.Equals("Pattern 4"))

{

string sql, sql1,sql2,sql3,sql4,sql5;

string sq = "truncate table p4\_2marks";

SqlCommand cmd1 = new SqlCommand(sq, con);

cmd1.ExecuteNonQuery();

sql = "insert into p4\_2marks (question,mark,type,code) select Questions, Mark\_Allocation, Question\_Type, Subject\_Code from StaffQuestions4 where Mark\_Allocation = '2' And Subject\_Code = '" + b + "'";

SqlCommand cmd = new SqlCommand(sql, con);

cmd.ExecuteNonQuery();

string sq1 = "truncate table p4\_16mark";

SqlCommand cmd2 = new SqlCommand(sq1, con);

cmd2.ExecuteNonQuery();

sql1 = "insert into p4\_16mark (question,mark,type,code) select Questions, Mark\_Allocation, Question\_Type, Subject\_Code from StaffQuestions4 where Mark\_Allocation = '16' And Subject\_Code = '" + b + "'";

SqlCommand cmd3 = new SqlCommand(sql1, con);

cmd3.ExecuteNonQuery();

sql2 = "truncate table p4\_dist2";

SqlCommand cmd4 = new SqlCommand(sql2, con);

cmd4.ExecuteNonQuery();

sql3 = "truncate table p4\_dist16";

SqlCommand cmd5 = new SqlCommand(sql3, con);

cmd5.ExecuteNonQuery();

sql4 = "insert into p4\_dist2 (question,mark,type,code) select distinct(question),mark,type,code from p4\_2marks";

SqlCommand cmd6 = new SqlCommand(sql4, con);

cmd6.ExecuteNonQuery();

sql5 = "insert into p4\_dist16 (question,mark,type,code) select distinct(question),mark,type,code from p4\_16mark";

SqlCommand cmd7 = new SqlCommand(sql5, con);

cmd7.ExecuteNonQuery();

Response.Redirect("Q4.aspx");

}

else if (a.Equals("Pattern 5"))

{

string sql, sql1,sql2,sql3,sql4,sql5;

string sq = "truncate table p5\_2marks";

SqlCommand cmd1 = new SqlCommand(sq, con);

cmd1.ExecuteNonQuery();

sql = "insert into p5\_2marks (question,mark,type,code) select Questions, Mark\_Allocation, Question\_Type, Subject\_Code from StaffQuestions5 where Mark\_Allocation = '2' And Subject\_Code = '" + b + "'";

SqlCommand cmd = new SqlCommand(sql, con);

cmd.ExecuteNonQuery();

string sq1 = "truncate table p5\_16mark";

SqlCommand cmd2 = new SqlCommand(sq1, con);

cmd2.ExecuteNonQuery();

sql1 = "insert into p5\_16mark (question,mark,type,code) select Questions, Mark\_Allocation, Question\_Type, Subject\_Code from StaffQuestions5 where Mark\_Allocation = '16' And Subject\_Code = '" + b + "'";

SqlCommand cmd3 = new SqlCommand(sql1, con);

cmd3.ExecuteNonQuery();

sql2 = "truncate table p5\_dist2";

SqlCommand cmd4 = new SqlCommand(sql2, con);

cmd4.ExecuteNonQuery();

sql3 = "truncate table p5\_dist16";

SqlCommand cmd5 = new SqlCommand(sql3, con);

cmd5.ExecuteNonQuery();

sql4 = "insert into p5\_dist2 (question,mark,type,code) select distinct(question),mark,type,code from p5\_2marks";

SqlCommand cmd6 = new SqlCommand(sql4, con);

cmd6.ExecuteNonQuery();

sql5 = "insert into p5\_dist16 (question,mark,type,code) select distinct(question),mark,type,code from p5\_16mark";

SqlCommand cmd7 = new SqlCommand(sql5, con);

cmd7.ExecuteNonQuery();

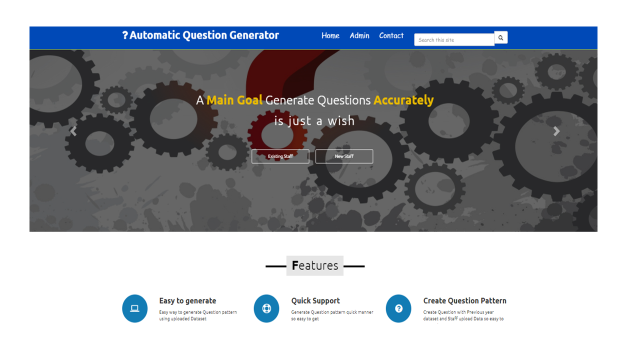
Response.Redirect("Q5.aspx");

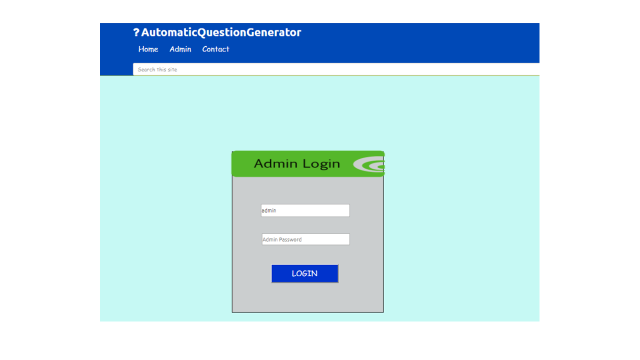
}

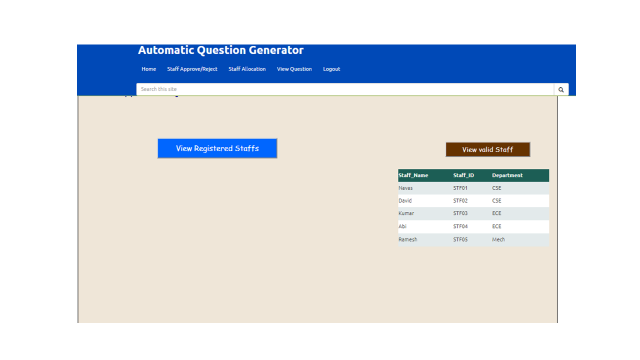
}}

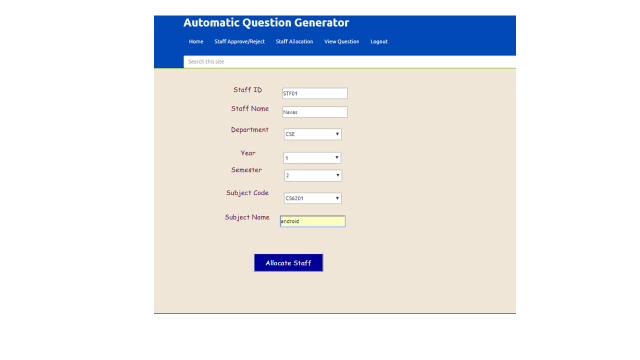
**Screen-Shots:**

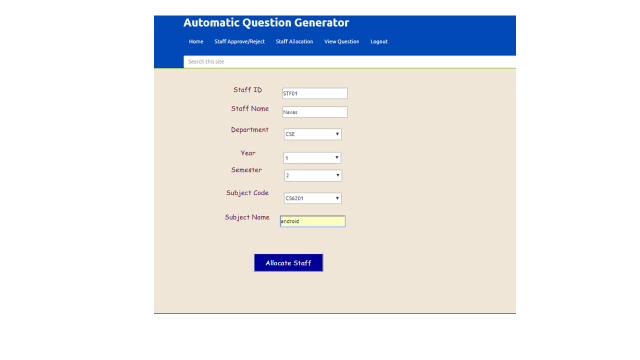
**Automatic Question Generator System**

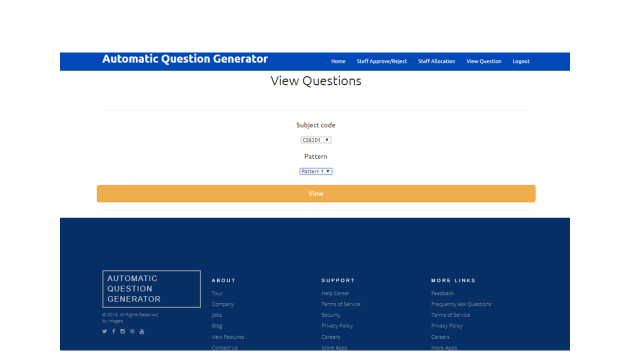
****

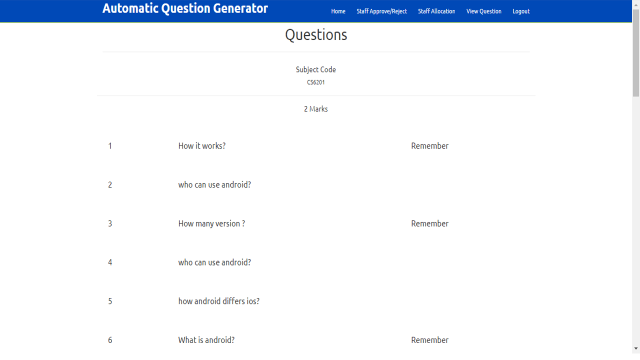
****

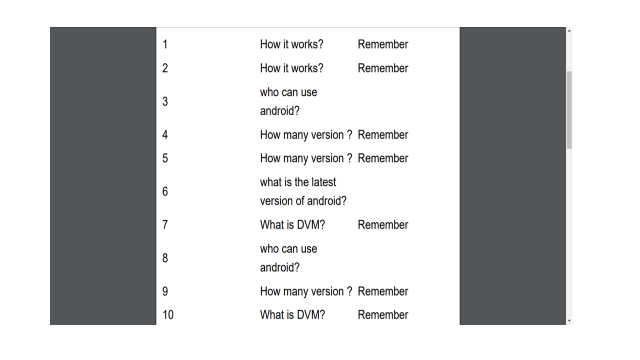
****

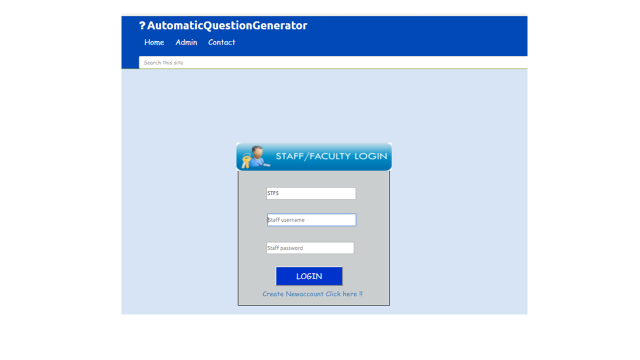
****

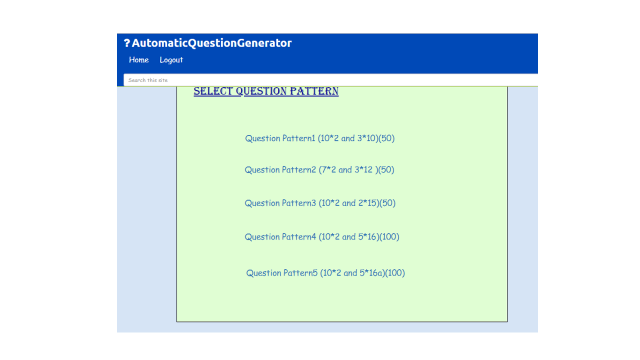
****

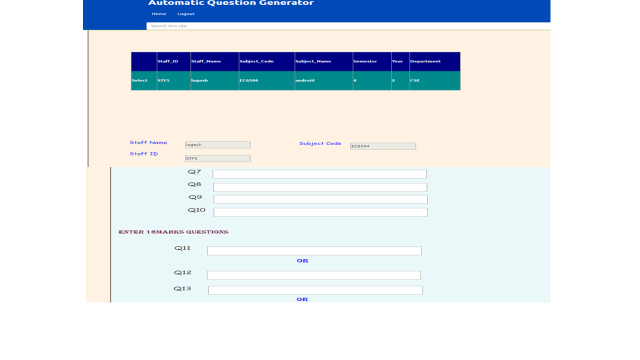
****

****

****

****

****

****

**Conclusion:**

The Automatic Question Generator system helps the staffs to prepare the questions in the easiest way and in the simple steps. It reduces the lot of man work and helps to save the times of the staffs to prepare the questions. Questions are generated as pdf so that there is no need to do any extra works and print out of the questions can also be done easily. The questions that which are generated are unique and it generates the important questions taken by two or more staffs. It makes very easier for preparing the questions. From this system we conclude that, this system helps the staffs in every college to prepare the question easily without any hassle.

**Future Enhancement:**

* In Future we can able to send the questions as the mail to particular departments directly.
* We can also have the One Time password system for staffs’ login, so that the system is more secured.
* We can add the model questions and previous year question for the reference for the students.