



DEPARTMENT OF  
COMPUTER SCIENCE AND ENGINEERING

---

**Title: System implementation (system setup,  
tools and techniques selection) for the given  
project**

---

INTEGRATED DESIGN PROJECT II  
CSE 406



GREEN UNIVERSITY OF BANGLADESH

---

## 1 Objective(s)

- For defining how the information system should be built.
- For ensuring that the information system is operational and used.
- For ensuring that the information system meets quality standard (i.e., quality assurance).

## 2 Systems implementation

Systems implementation is a set of procedures performed to complete the design (as necessary) contained in the approved systems design document and to test, install, and begin to use the new or revised Information System. Figure 7.1 depicts systems implementation as the fifth major step in the development of an Information System.

The systems implementation goals are as follows::

- Complete as necessary the design contained in the approved systems design document. For example, the detailed contents of new or revised documents, computer screens, and database must be laid out and created.
- Write, test, and document the programs and procedures required by the approved systems design document.
- Ensure, by completing the preparation of user manuals and other documentation and by training personnel, that the organization's personnel can operate the new system.
- Determine, by thoroughly testing the system with users, that the system satisfies the users' requirements.
- Ensure a correct conversion by planning, controlling, and conducting an orderly installation of the new system.

### 2.1 Information System implementation Techniques

In this section we describe implementation approaches that can be taken to install the new or modified system. The four most common implementation approaches.

**1. parallel approach:** Figure 1, the parallel approach, provides the most control of the three. When the new system is used at the same time as the old system the two systems are said to be running in parallel. During this period, time x to time y (which is usually one operating cycle, such as one month or one quarter), the outputs of the two systems are compared to determine whether the new system is operating comparably to the old. At time y, management makes a decision, based on the comparison of the two systems' outputs, whether to terminate the operation of the old system. The parallel approach provides more control because the old system is not abandoned until users are satisfied that the new system adequately replaces the old. Although this approach makes good intuitive sense, in practice it frequently alienates users who perceive parallel operations as doubling their workload.

#### **Advantages:**

1. Users can compare the output of the old system with the output of the new system, to ensure correctness
2. There is little risk of data loss because the known-good system is running

#### **Disadvantages:**

1. Users must take more time to enter data into two different systems
2. Data could be different in two different systems if there is intensive data entry.

#### **Example:**

A medical system that tracks patient heart rates is being replaced. A new system is attached while the old system is still working. The two systems are used in parallel to ensure the new system produces the exact same data as the old system.

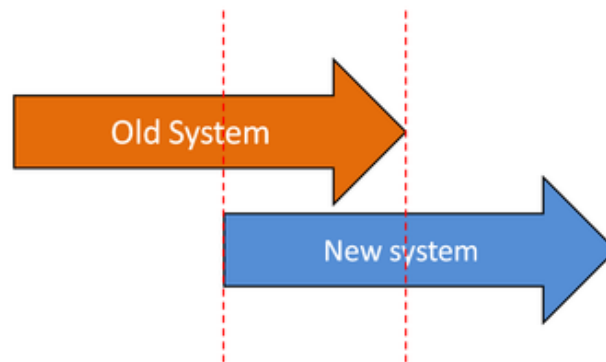


Figure 1: Parallel Approach

**2.Direct approach:** Figure 2, the direct approach, is often called the “Big Bang” approach and is the riskiest of the three approaches. At time x the old system is stopped and the new system cuts in with no validation that the new system operates comparably to the old. While we will see a little later that it need not be so, enterprise systems are often implemented using this approach. Sometimes, as you’ll see in the Hershey story in Chapter 11, direct implementations can lead to disaster.

**Advantages:**

1. If the system is not critical, this can be a good method for implementation

**Disadvantages:**

1. If you are not sure the system will work, this method of implementation may not be a good idea

**Example:**

A store is implementing a new electronic system for employees to leave suggestions for improvement. There is no existing system. The store uses direct method because they are very sure the new system will work, there is a low cost if the system fails, and the store wants to make a "big splash" with the new system.

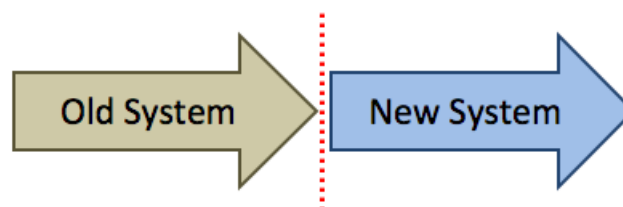


Figure 2: Direct Approach

**3.Phased:** Figure 3, When small parts of the new system gradually replace small parts of the old system, the implementation method is said to be phased.

**Advantages:**

1. Training can be completed in small parts
2. A failure of the new system has minimal impact because it is only one small part
3. Issues around scale can be addressed without major impact.

**Disadvantages:**

1. This implementation method takes more time to get the new system fully online than other methods.
2. There is a possibility of data loss if part of the new system fails.

**Example:**

A school has a new system to manage student athletics. The old system is paper and pencil. Slowly, over time, a new system is introduced to manage students, their teams, seasons, and their coaches. At first, the new system simply manages teams. Then the new system manages seasons (and school years), slowly, the new system is increased to manage coaches, players and finally events. At the end of implementation, the new system is managing everything related to student athletics and the old paper and pencil system isn’t being used any longer.

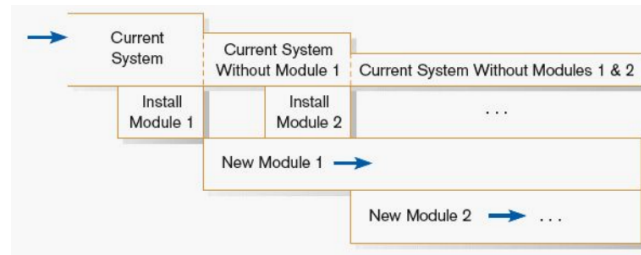


Figure 3: Phased Approach

**4. Single-Location or Pilot:** Figure 4, When a small group of users within an organization uses a new system prior to wider use, the system is said to be piloted.

**Advantages:**

1. Training can be supported by pilot group
2. Failure or problems can be identified and addressed without wide-spread impact to the organization

**Disadvantages:**

1. In a pilot, issues of scale can cause problems. For example, the system might work well for 10 users, but not for 1000.

**Example:**

A bakery is implementing a new system for customers to order online. They choose 50 customers and ask them to try the new system, and provide feedback. The bakery can then identify issues and address them prior to implementing systems for thousands of users.

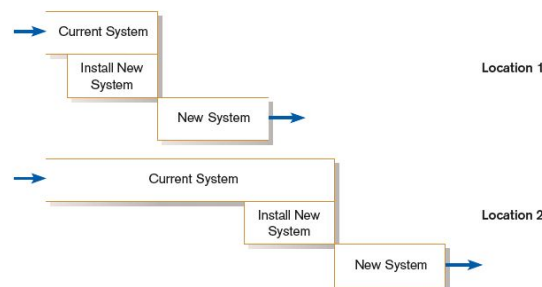


Figure 4: Pilot Approach

### 3 Software Implementation steps

Implementing new software is always a challenge, even for the most flexible companies. Succeeding demands planning transitions smoothly and providing your team with relevant training material on the new software. If you want to avoid hitches in your implementation process, follow these steps:

**1. Planning Ahead**

Always start with a plan! Identify which processes and teams are going to be affected by the implementation and create a shared timeline for carrying the execution gradually. At this stage, you should try and question every possible thing that may get affected by the new software:

- Is the implementation going to be done on stages or altogether?
- How long will it last?
- Will it require new security measures?
- Who is going to deliver training for the members of your team?

Remember, planning ahead will always make for a smoother implementation.

**2. Process Design**

Process design is a technique that allows you to organize and run things more efficiently, no matter whether it's a business, software or a team. Most of the commercially available software nowadays has been created

---

using process design methods. This is why when implementing new software you will probably need to adjust many of your new processes to the logic that was used for building that software. If you are unsure how to deal with process design, you may decide to work with a Design Analysts or hire a software company to advise you on this topic.

### **3. Solution Design**

Once you have created the process design, it is time to work on the solution design, which is essentially a roadmap of business requirements and processes. During this step, you will map the software to your business requirements so that everything is working in tandem. Remember that these initial requirements are just a starting point. You may realize that other methods work better after having used the software for a few months. The ultimate aim of solution design is to identify which features of your new software can be used just as they are and which aspects will need to be customized to fit the needs of your business and teams.

### **4. Configuration and Customization**

It is now time to install the software and proceed to configure those features that can be used immediately. As a matter of fact, this step should always come first, before defining any processes or rushing to customize any module. You will easily realize that some modules and functionality of this software meet your company's needs out-of-the-box, while other aspects will require to be configured and connected to the roadmap you designed previously. Once installed, you'll be able to customize the software. What's the purpose of customizing the software? Companies customize the software so that it meets their specific business environment, processes, and needs. Let's imagine you are implementing sales and marketing software: you are very picky with the way you create new leads and require your team to add details such as lead name, last name, email address, phone number, LinkedIn profile URL, vertical market and so on. However, the default configuration in your software doesn't have a field to add URLs. So you decide to customize it

### **5. Integration**

Integration is a critical step within software implementation and it involves migrating data from one system to another. With proper integrations, you can save your team from having to copy data between systems manually. They will thank you for having thought about them! When developing integrations remember about data synchronization! Do you want to run the integrations in real-time or behind the scenes —which means outside of normal business hours? If accessing updated data is a critical thing for your team, you may want to consider running integrations continuously. Otherwise, you may wish to consider doing this outside regular working hours.

### **6. Reporting**

This phase is about understanding what information is valuable for your teams in order to improve their decision-making process on a daily basis. The good thing is most software nowadays includes data dashboards where you visualize your business performance — in case it doesn't come with a reporting module, it is something that you may consider to build and customize! It is a good idea to list the most important metrics for your organization, but think smart: nobody will use these reports if they are not actionable. Avoid trying to include every stat out there and just focus on insight which is really useful. Finally, try to come up with a standard reporting layout, so everyone in your organization can digest the information easily.

### **7. Training Testing**

Last but not least comes training and testing. Training may come in different forms and to different groups, from educating your project team on the new software to teaching the end-user how it works. You should always plan and deliver training systematically, even if people may have been already exposed to the same software somewhere else. And don't forget that learning is a continuous process, so best practice suggests training sessions should be ongoing and planned according to actual needs. We have finally arrived at the final step of implementing software successfully: testing. There are four different types of testing: unit, project team, integration, and end-user testing. Each of them should be carried at different stages of the software implementation process to guarantee that everything works according to the plan. Like training, testing is also an ongoing process that should be done throughout the software implementation process, but also after the team has been using the software for a while.

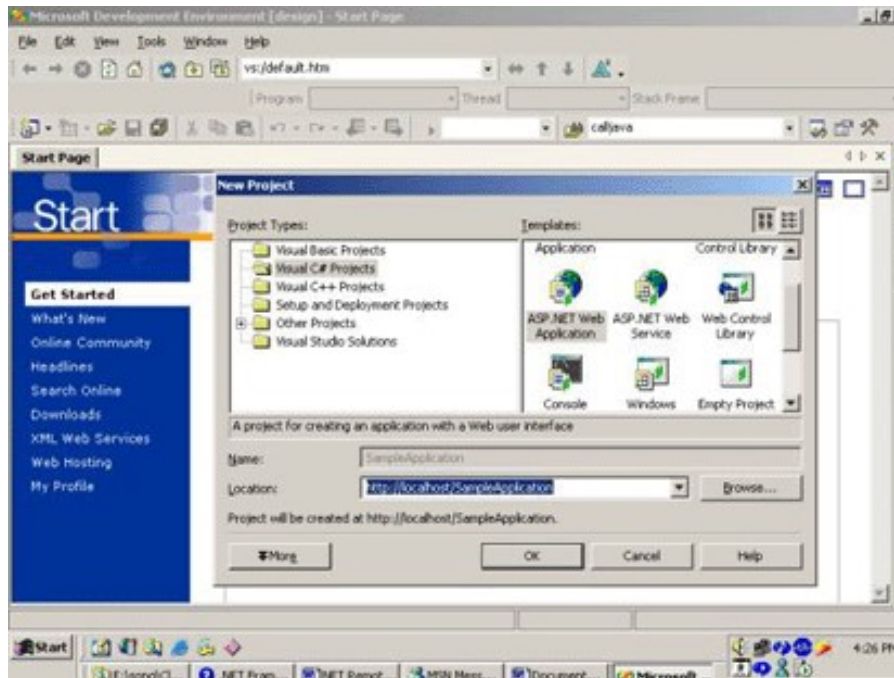
## **4 System Setup**

Deployment of an application is a big issue in application software development. Normally you have to check each and every possible scenario before deploying your application so that your current deployment doesn't affect the existing application.

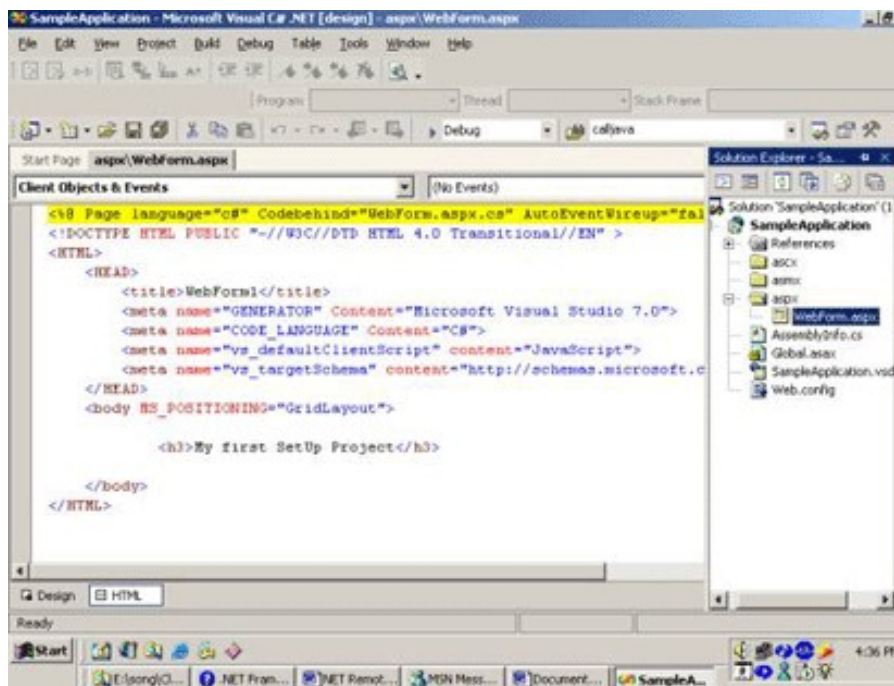
Through Microsoft Visual studio .NET we can do it in a better way. I would like to discuss creating a setup project for a web application.

Let us discuss the step-by-step process of creating an installer for a web application. In this I am mainly concentrating on the Setup project. You can follow the same steps for creating an installer for any complicated project.

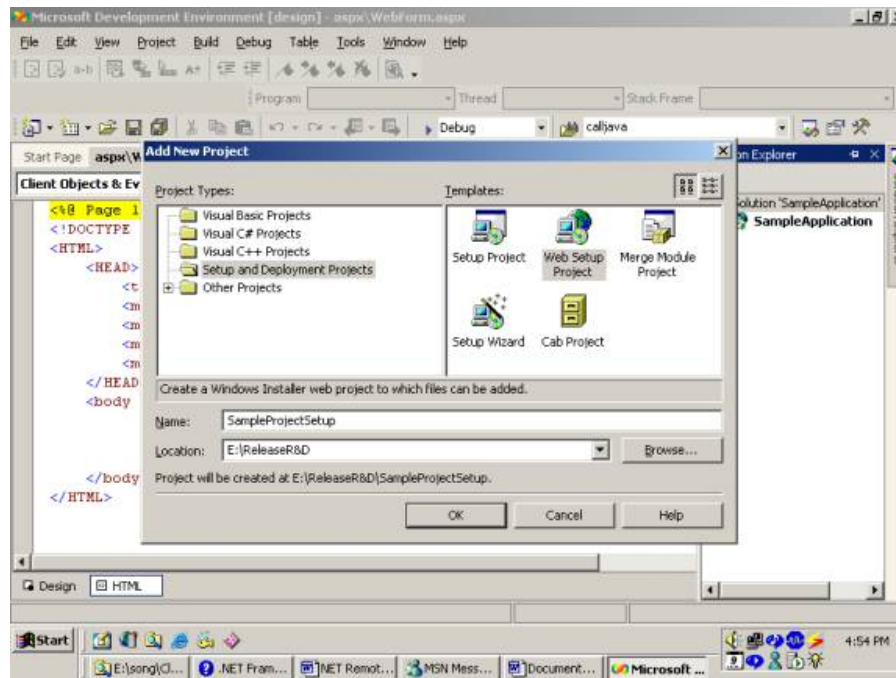
- **Step I:** Create a Simple web application using Microsoft Visual Studio .NET



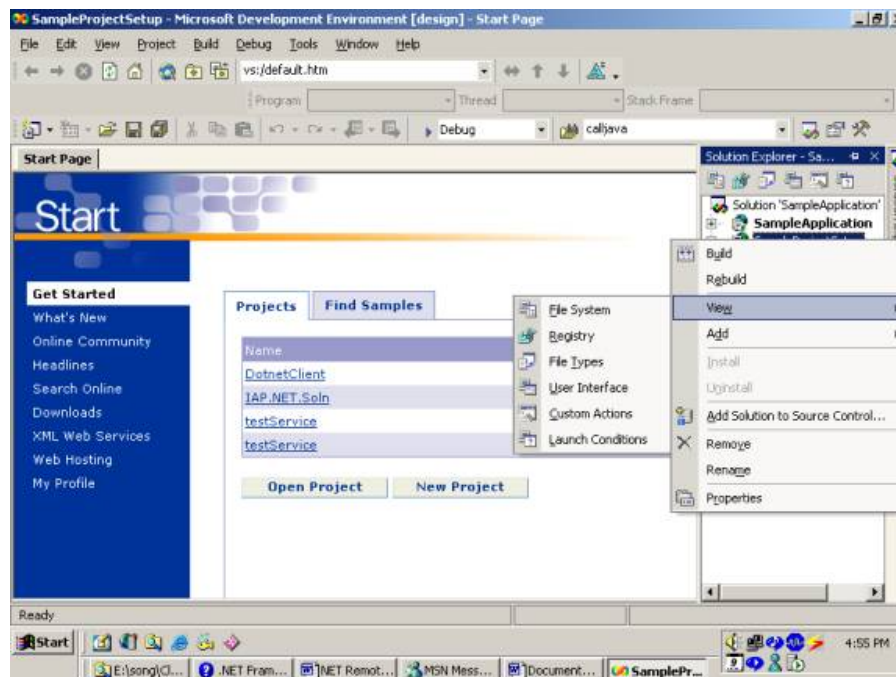
- **Step II:** Create a Simple WebForm in the web project. You can also add some more Webform or other control in this project by creating a separate directory or in the same directory.



- **Step III:** Select Setup and Deployment Projects and then select Web Setup Project also specify the location and name of your project.



After creating setup project right click on the project and then select view, it will show different possible operations, which you can perform with this setup project.



These options are

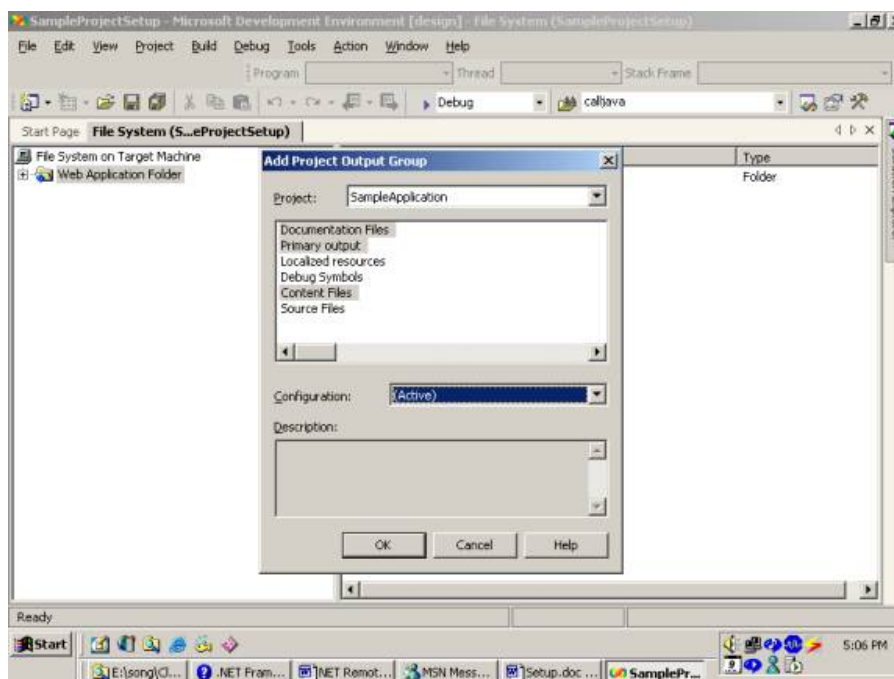
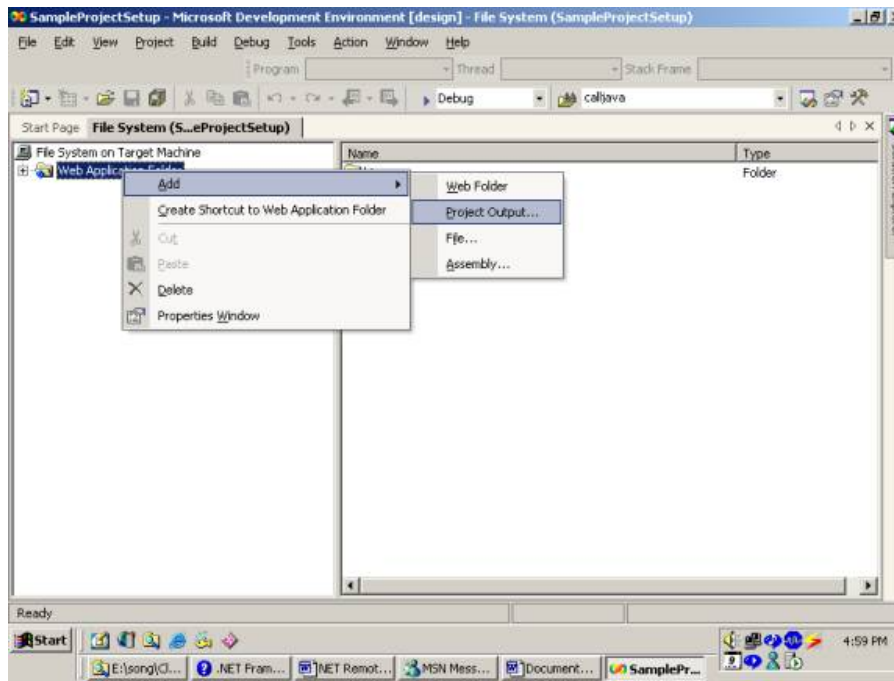
- File System
- Registry
- File Types
- User interfaces
- Custom Actions



– Launch conditions

- **Step IV:** Select File system, it is used to create a file system on the target machine. Through this, you can specify what details you want to provide at the target machine.

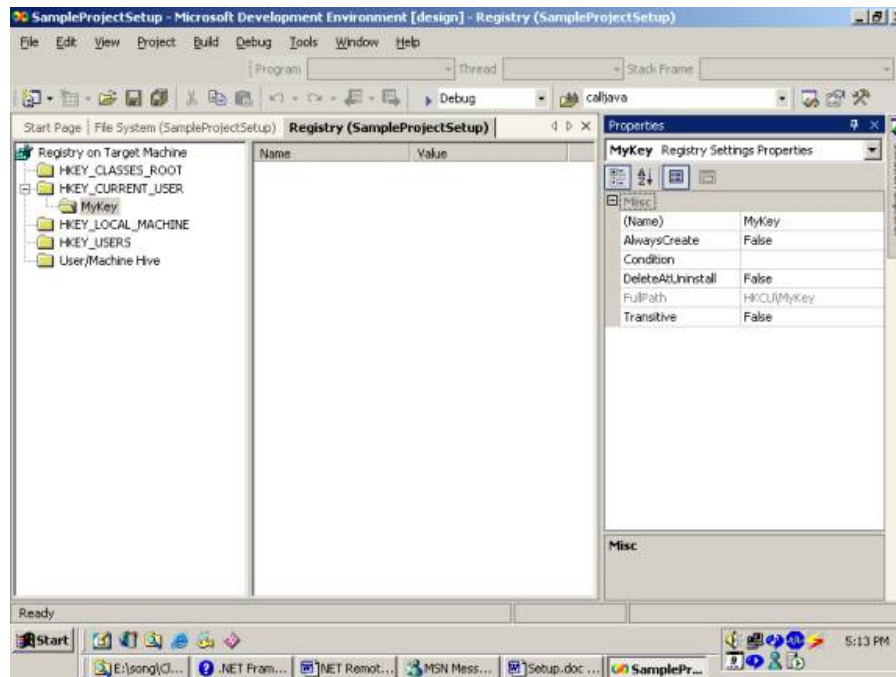
For providing right-click on web application project then select add, it will display certain options out of that select project output. After that select all details you want to supply for the target machine.



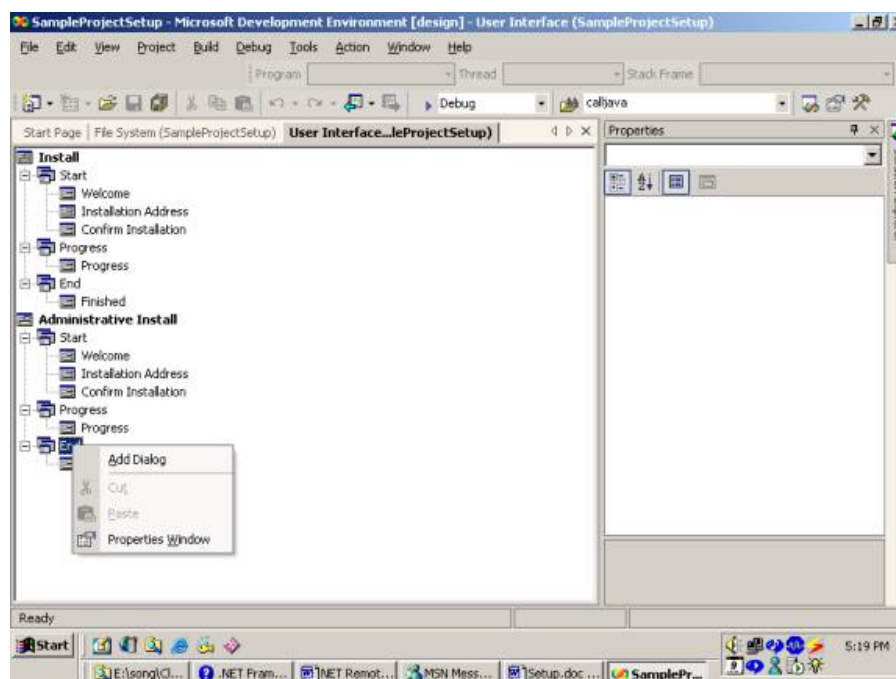


- **Step V:** Second Selection is Registry Selection in this selection you can specify the dll, which you want to install at client registry. For that, you have to create an exact structure of registry in the registry folder structure and then you can specify the required DLL for installation in that structure.

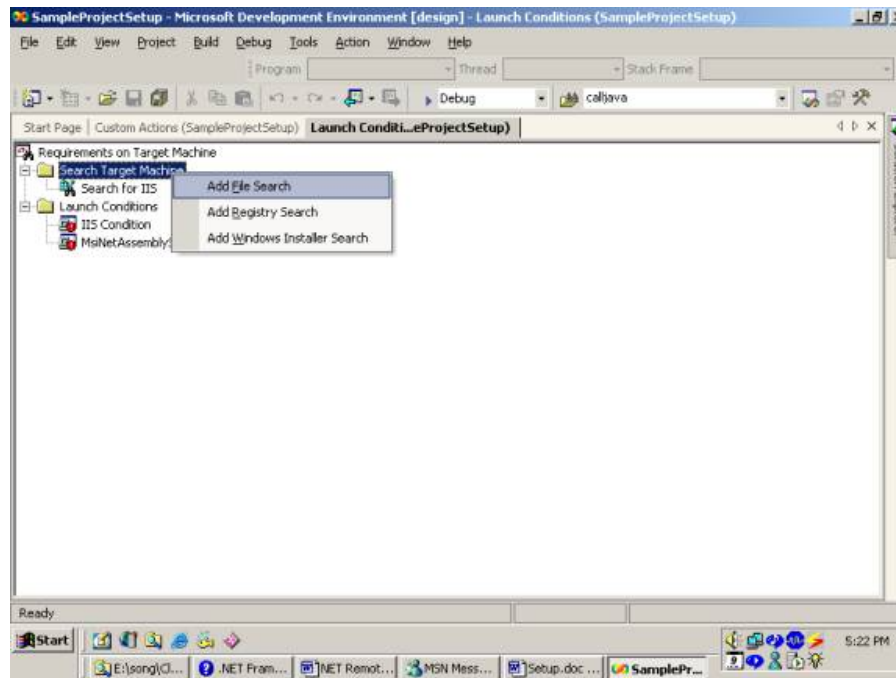
You can create a new folder by right-clicking on that folder and then select a new key and define key Name. Each key is associated with a property window where you can specify which dll you want to install in the registry.



- **Step VI:** Select User Interface from the view in this selection you can specify what are the screens you want to display during installation process. You can change it, remove it or you can define new screen by right-clicking on the screen header.



- **Step VII:**Select Launch Conditions through this condition you can specify your criteria, which are pre-requisites for your application installation.



- **Step VIII:**Now the final step is to compile the setup project. After compilation, you will notice that it has generated SampleProjectSetup.msi at the applications bin/debug folder.

You can supply this MSI the target machine, when you run this MSI at the target machine it will create a virtual directory as well as create the same folder structure, which you have specified in File System. This installer will also install the specified libraries in the registry which are specified in the Registry.

Now you can browse that application at the target machine in the same way as you have done at your own machine.

## 5 System Development Tools

System development is the process of defining, designing, testing and implementing a software application. System development includes the management of the entire process of the development of computer software.

A system development project includes a number of different phases, such as feasibility analysis, requirements analysis, software design, software coding, testing and debugging, installation and maintenance the all process can be maintain by different tools. **List of System Development Tools:**

- Engine Yard
- DevDocs
- Public Apis
- Divide.io
- Codeship
- Codefresh
- Node
- Wercker
- Stamply
- Github

- 
- Codekit

- **Engine Yard**

Engine Yard is a platform as a service provider that automates, configures and deploys applications in the cloud. Engine Yard to deploy and manage their apps in the cloud so they can continue doing what they do best writing beautiful apps.

Website: [www.engineyard.com](http://www.engineyard.com)

Pricing model: Paid

- **DevDocs**

Fast, offline, and free documentation browser for developers. Devdocs.io is a very slick and fast reference manual aggregator for popular front-end languages and libraries.

Website: [www.devdocs.io/](http://www.devdocs.io/)

Pricing model: Free

DevDocsPublic Apis Public APIs, A collective list of free JSON APIs for use in web development. Application programming interfaces, sets of technologies that enable applications to interact with each other. A public API is probably what first comes to mind when you think about APIs, but these are only a small portion of the APIs that exist around the web.

Pricing model: Free

- **Divideio**

Quickly build a backend for your mobile app. Divide.io is an open source backend library. It's easy, secure, and efficient. Deploy anywhere. Keep control of your data. Check us out on Github. Free open source.

Website: [www.divide.io/](http://www.divide.io/)

Pricing model: Free

- **Codeship**

Code ship helps to release software quickly, automatically and multiple times a day. It shortens the development cycles thus reducing the risk of bugs and increasing innovation.

Website: [www.codeship.com/](http://www.codeship.com/)

Pricing model: Free

- **Codefresh**

Codefresh is a continuous delivery and collaboration platform for Docker. Codefresh helps agile teams improve quality and increase speed-to-market by facilitating rapid feedback and testing cycles.

Website: [www.codefresh.io](http://www.codefresh.io)

Pricing model: Free

- **Nodejs**

Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. Node.js' package ecosystem, npm, is the largest ecosystem of open source libraries in the world.

Website: [www.nodejs.org](http://www.nodejs.org)

Pricing model: Free

- **Wercker**

Wercker is a platform and tool to rapidly develop, build, and deploy containerized applications and micro services for the modern cloud. This platform was built in order to make developers' lives easier and enabling them to move fast and make things

Website: <http://www.wercker.com/>

Pricing model: Free

---

- **Stamplay**

Stamplay is an API-based development platform, empowering web developers to build and launch full-featured web apps. Stamplay is a low code platform that enables organizations to build high value business process spanning multiple cloud services.

Website: [www.stamplay.com](http://www.stamplay.com) Pricing Model: Paid

- **Github**

GitHub is a Web-based Git version control repository hosting service. It is mostly used for computer code. It offers all of the distributed version control and source code management functionality of Git

Website: [www.github.com](http://www.github.com)

Pricing Model: Paid

- **Codekit**

Codekit is more than just a compiler for Less Sass. It also compiles JavaScript, Markdown and more, plus it continuously monitors for file changes. Modified files are auto compiled and it also automatically reloads the browser. Codekit is well documented and simple

Website: <https://codekitapp.com/>

Pricing model: Free

## 6 Discussion & Conclusion

When a system implementation is poorly planned, there is an increased chance the system will fail. Don't just think "hey, I'll just make sure the system works". You must be responsible for ensuring there is a good plan for implementation. I have seen perfectly good systems fail because implementation was done poorly. It comes down to good planning.

There is another real-world issue you should be aware of. There is a wide range of emotional reactions to change. Some users will welcome change, others will be uncertain, some will be openly hostile whilst others will attempt to sabotage it. Training issues may require organizations to restructure their workforce. New systems often accompany a wide range of emotional reactions. Communication about the change, and sensitivity about users change should be well-managed.

### 6.1 Problem analysis

A library database system is an infrastructure that allows users to search books and book content, add/remove, and download selected books. The problem faced is that library users require an efficient method to find a specific book or keyword(s) within a book given a continuously expanding library. Some scenarios of the 'Library Management System' are as follows:

1. User who registers himself as a new user initially is regarded as staff or student for the library system.
  - (i) For the user to get registered as a new user, registration forms are available that is needed to be fulfilled by the user.
  - (ii) After registration, a library card is issued to the user by the librarian. On the library card, an ID is assigned to cardholder or user.
2. After getting the library card, a new book is requested by the user as per their requirement.
3. After, requesting, the desired book or the requested book is reserved by the user that means no other user can request for that book.
4. Now, the user can renew a book that means the user can get a new due date for the desired book if the user has renewed them.
5. If the user somehow forgets to return the book before the due date, then the user pays fine. Or if the user forgets to renew the book till the due date, then the book will be overdue and the user pays fine.
6. User can fill the feedback form available if they want to.
7. Librarian has a key role in this system. Librarian adds the records in the library database about each student or user every time issuing the book or returning the book, or paying fine.

- 
8. Librarian also deletes the record of a particular student if the student leaves the college or passed out from the college. If the book no longer exists in the library, then the record of the particular book is also deleted.
  9. Updating database is the important role of Librarian.

## **7 Lab Task (Please implement yourself and show the output to the instructor)**

1. Describe the above (Library Management System) system implementation steps
2. Justify, which system implementation technique is more suitable of the given project 'Library Management System'.

## **8 Policy**

Copying from internet, classmate, seniors, or from any other source is strongly prohibited. 100% marks will be *deducted* if any such copying is detected.