### Web based application

**Webforms Applications**
- accessed anywhere in the world through the internet
- run on Clients
- GUI put together by controls in the System.Windows.Forms namespace
- depend on the .Net Framework but not on the Windows" operating system so they could (hypothetically) run on other platforms and .net-implementations
- Physically mostly a 2 tier
- although in a good design it has multi layers
  - Client (the window forms)
  - server (the database).
- authentication/authorization mechanism partially built
- web apps have validation controls

**When to use**
- Primarily have a browser user interface
- Public access such as e-commerce, employee handbooks, etc.
- No distribution costs
- Platform independent
- Take advantage of features built into most recent browsers
- Recently a little quicker than before
  - AJAX, JSON, Web API, Jquery
  - Develop full stack in one language JavaScript w/Node.js
  - Typically 2-4x time and expense as Windows app dev

### Window based application

**Webapplications**
- need to be install on your machine to access.
- run on webservers (usually IIS)
- Webservices & Websites (ASP.NET) are typical web applications
- Physically ASPNet applications uses 3 tiers:
  - Client (the webbrowser)
  - middle Tier (the webapplication)
  - The server (the database).
- desktop applications (windows forms) need the entire mechanism built
- windows apps don't have validation controls, data needs to be validated through code

**When to use**
- want the client application to be responsible for much of the processing burden in an application
- Win32 desktop Visual Basic and Visual C++ like data entry systems, point of sale sys
- Games
- Quicker