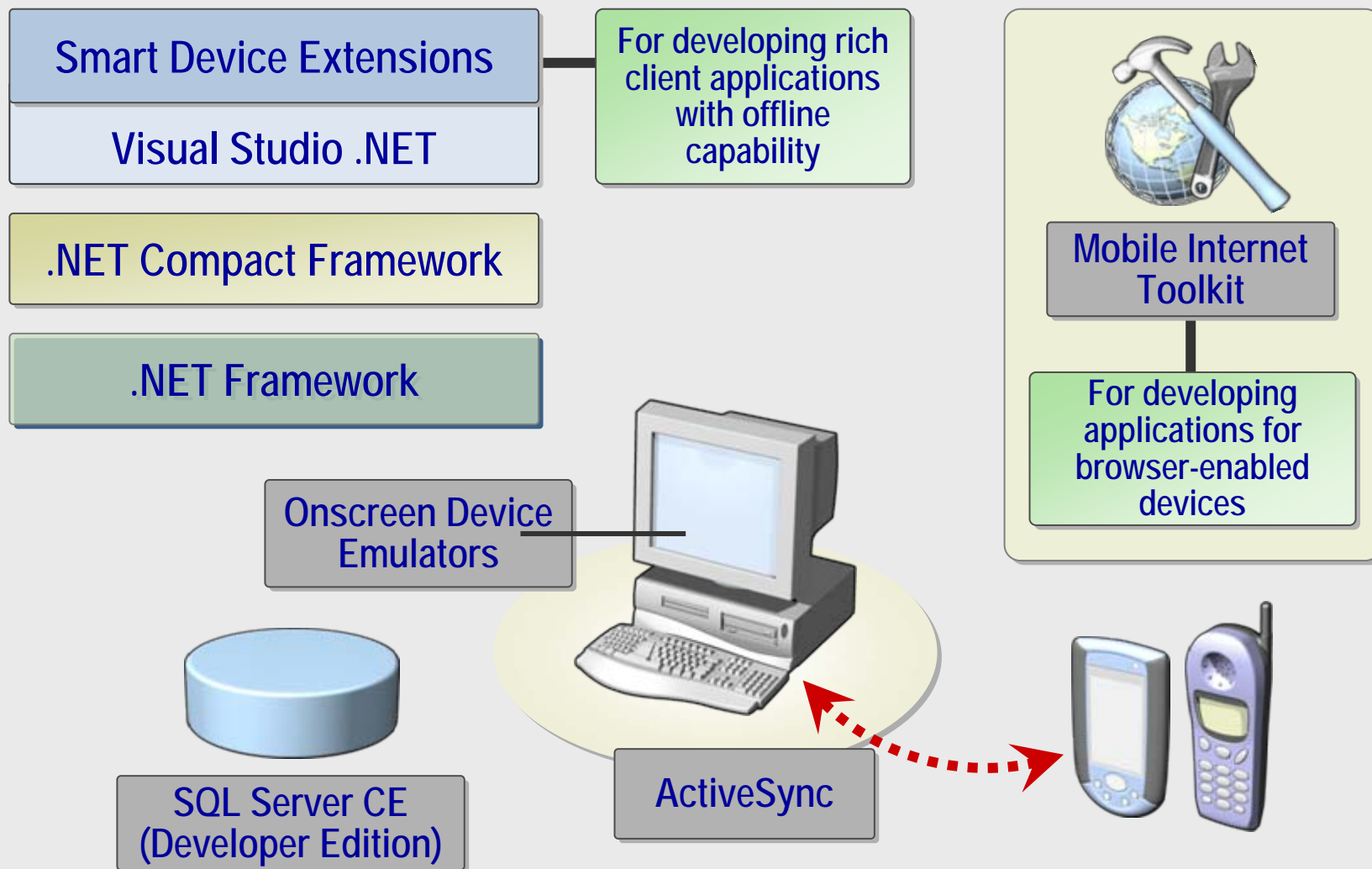


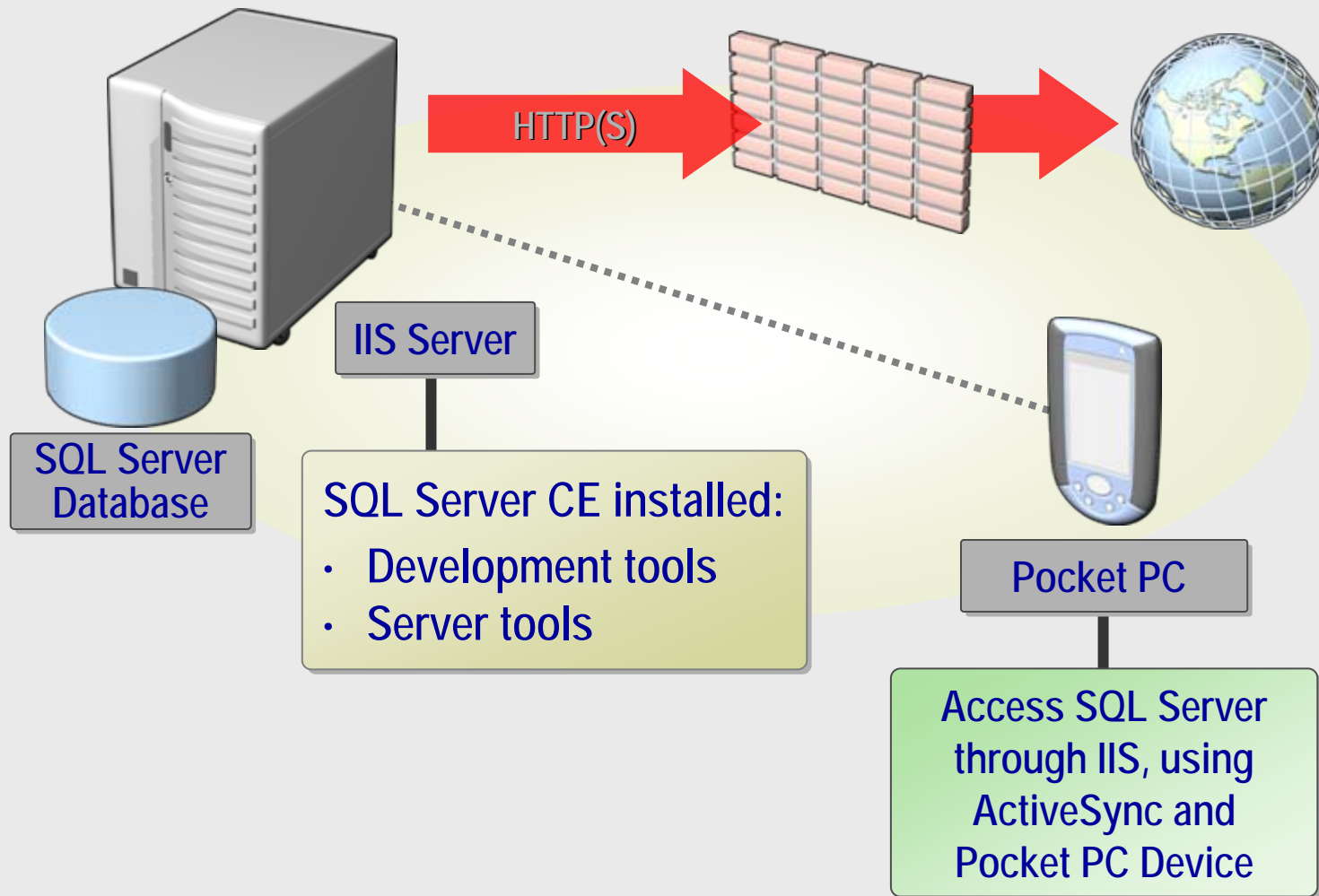
Mobilne aplikacije

12/13

Alati za razvoj mobilnih aplikacija



Kontekst rada mobilnih aplikacija



Ključne karakteristike mobilnih aplikacija

☐ Vrijeme pokretanja

- Mobilne aplikacije koriste se učestalo, te u kratkim vremenskim razmacima
- Važno da je aplikacija što brže raspoloživa za uporabu

☐ Vrijeme odziva

- Pri interakciji aplikacija treba brzo reagirati na korisnikov zahtjev.
- Važno je korisniku dati do znanja da se njegov zahtjev obrađuje (promjenom ekrana, ispisivanjem poruke, postavljanjem kursora u čekanje)

☐ Jasno određena usluga

- Usluga koju aplikacija pruža mora biti što jasnije definirana i lako pristupačna

☐ Podešena komunikacija s izvorom informacija

- Smanjena brzina prijenosa podataka na mobilnim uređajima
- Prijenos samo bitnih informacija, podešavanjem filtara i načina komunikacije

☐ Konzistentnost i prikladnost

- Uobičajeno zahtijevana konzistentnost korisničkog sučelja mora dodatno biti provedena sukladno mogućnostima uređaja.

☐ Ograničeni kapacitet uređaja – prostorni, memorijski, procesni

Oblikovanje korisničkog sučelja mobilne aplikacije

❑ Ključne pretpostavke

- Kreativnost - potrebna zbog limitirane veličine zaslona ručnih računala.
- Disciplina - potrebna za održanje konzistentnosti odabranom modelu, odnosno dizajnu korisničkog sučelja.

❑ Ekran osjetljiv na dodir

- važno je minimizirati broj promjena rasporeda – ukoliko treba proći nekoliko koraka unosa, treba nastojati omogućiti da se unos obavi uzastopnim pritiskanjem tipke na istom mjestu, odgovarajući broj puta.

❑ Veličina zaslona i navigacija

- zbog veličine uređaja te načina držanja i unosa, dio informacija ostaje prekriven, pa treba posebno pripaziti na raspored kontrola i navigaciju

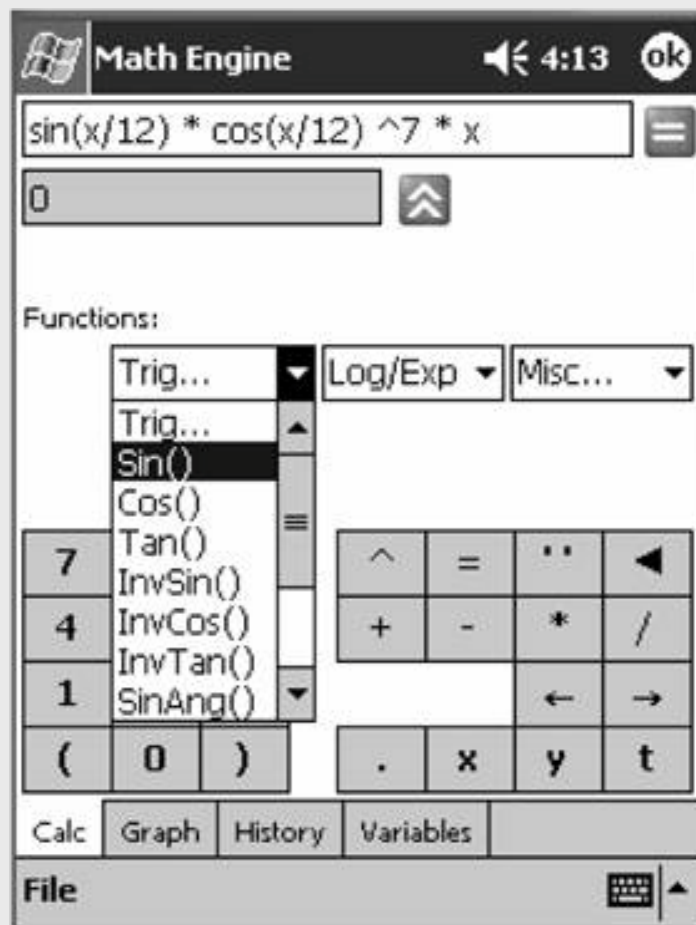
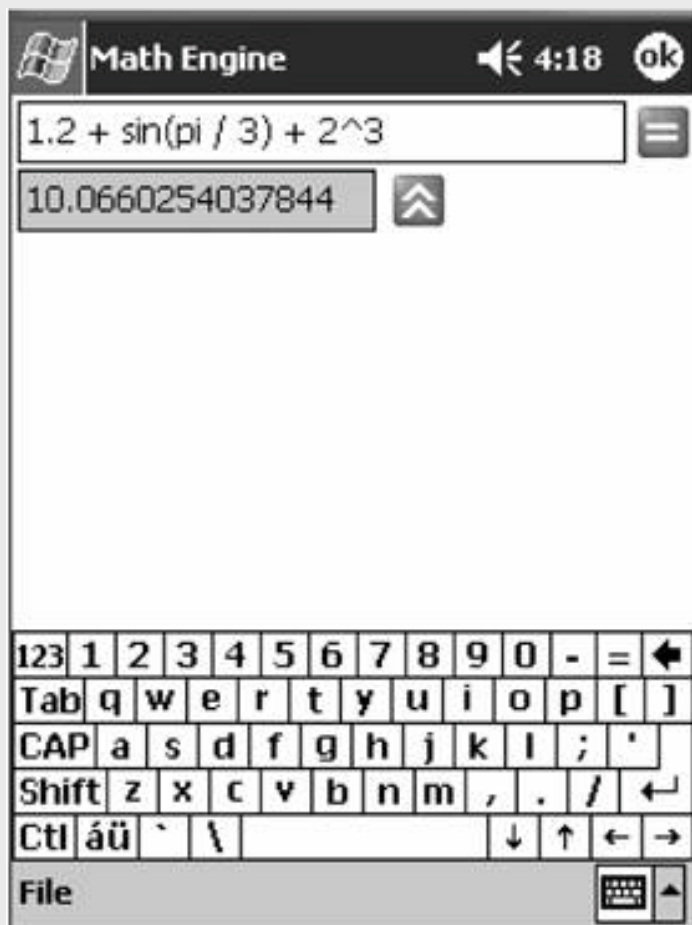
❑ Važnost velikih kontrola

- male i neprecizne kontrole frustriraju korisnika
- pri testiranju se koristi emulator pri čemu se programer smije služiti mišem, pa se stječe pogrešan dojam o lakoći korištenja
- realna interakcija mora biti moguća pod različitim kutovima pristupa uređaju

Oslonac na ugrađene kontrole

January 1900						
S	M	T	W	T	F	S
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10
Today: 1/18/07						

- ❑ Treba se osloniti na promišljene i dokazane kontrole
- ❑ Ugrađene kontrole su univerzalne - ne nužno i najbolje za sve primjene



Grafičko sučelje mobilnih uređaja

❑ Sustav direktorija na mobilnom uređaju

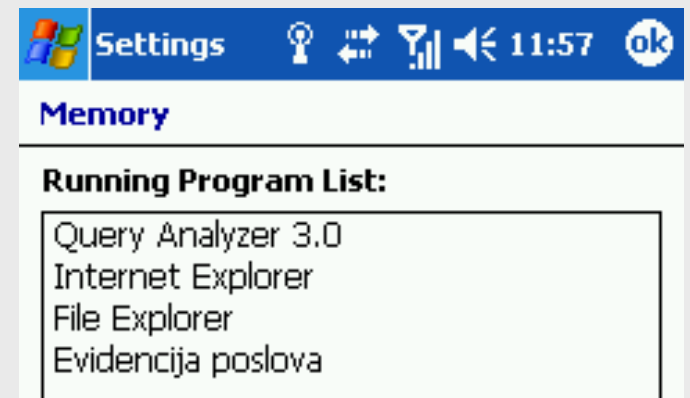
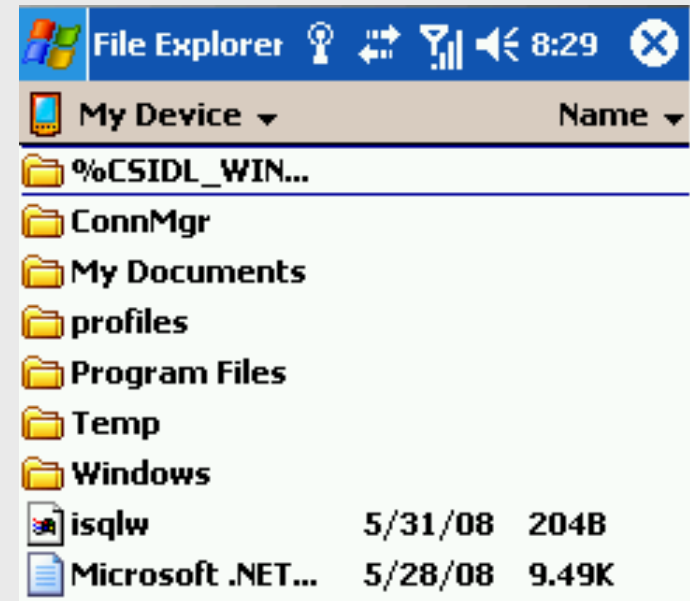
- *My Device* – korijen (kao *My Computer*)
- *Program Files*
 - Direktorij u koji se podrazumijevano kopira aplikacija prilikom objavljivanja

❑ Aktivne aplikacije

- Ako se aplikacija nije zatvorila, ona ostaje aktivna
- Pregled aktivnih aplikacija: *Settings* → *System* → *Memory* → *Running Programs*

❑ Za unos se najčešće koristi posebna pisaljka (stylus) za pisanje po zaslonu osjetljivom na dodir

- klik = *tap*
- desni klik = *tap & hold* (zadržani klik)



Model pohrane podataka

☐ Datoteke

- Ako aplikacija koristi mali broj podatka vjerojatno je pametnije podatke spremati u datoteku.
- Ako su podaci namijenjeni za komunikaciju i razmjenu informacija s drugim uređajima, XML je pretpostavljeno najbolja odluka.

☐ Baze podataka

- Ako aplikacija koristi velik broj podataka nad kojima je potrebno postavljanje upita, obavljanje transakcija i sinkronizacija podataka, baza podataka je najbolje rješenje.
- Glavni nedostatak baza podataka je njihova dodatna veličina jer sadrže podatke o podacima i dodatne mehanizme implementacije.

☐ Lokalna ili udaljena baza podataka

- Ako aplikacija koristi veću količinu podataka bolje je bazu podataka imati na samom uređaju.
- Ukoliko baza podataka nije na uređaju treba ugraditi i komunikaciju s poslužiteljem za razmjenu podataka (web servis ili replikacija).

Compact Framework

❑ .NET Compact Framework (.NET CF)

- Verzija *.NET Frameworka* prilagođena za mobilne uređaje
- Uz knjižnice *.NET Frameworka*, definira i specifične za mobilne uređaje
- Mobilni uređaj koji želi pokrenuti aplikaciju napisanu u *Compact Frameworku* mora naravno imati instaliran *.NET Compact Framework*

❑ Posebnosti CF

- Prostor imena za interaktivne aplikacije je *Microsoft.WindowsCE.Forms*
- Alfanički unos obavlja se posebnom kontrolom - *InputPanel*

❑ Razlike između *.NET* i *.NET CF*

- CF podržava većinu kontrola osim *CheckListBox*, *PageSetupDialog*, *PrintDialog*, *PrintDocument*, *PrintPreviewControl*, *PrintPreviewDialog*, *RichTextBox*, ...
- CF ne podržava izmjene u *GridView* kontroli, MDI, Drag&Drop, GDI+
- CF ne podržava ASP.NET, OleDb

Ostale komponente

❑ Visual Studio i emulatori različitih mobilnih uređaja

- Tools \ Connect to Device
- Tools \ Device Emulator Manager – Connect, opcionalno Craddle

❑ Konfiguracija emulatora (PPC, telefon, tablet u raznim verzijama)

- Traka s alatima \ Device Options \ (emulator) \ Emulator Options \ Properties ... \ Network - bind to
 - Connected network Card – ukoliko je PC u domeni i ima statičku IP
 - **Loopback adapter – ukoliko je PC samostalan**

❑ Virtualni uređaj (po potrebi)

- **Settings \ Connections \ Network Card – Virtual Ethernet + The Internet**

❑ Prethodno instalirati

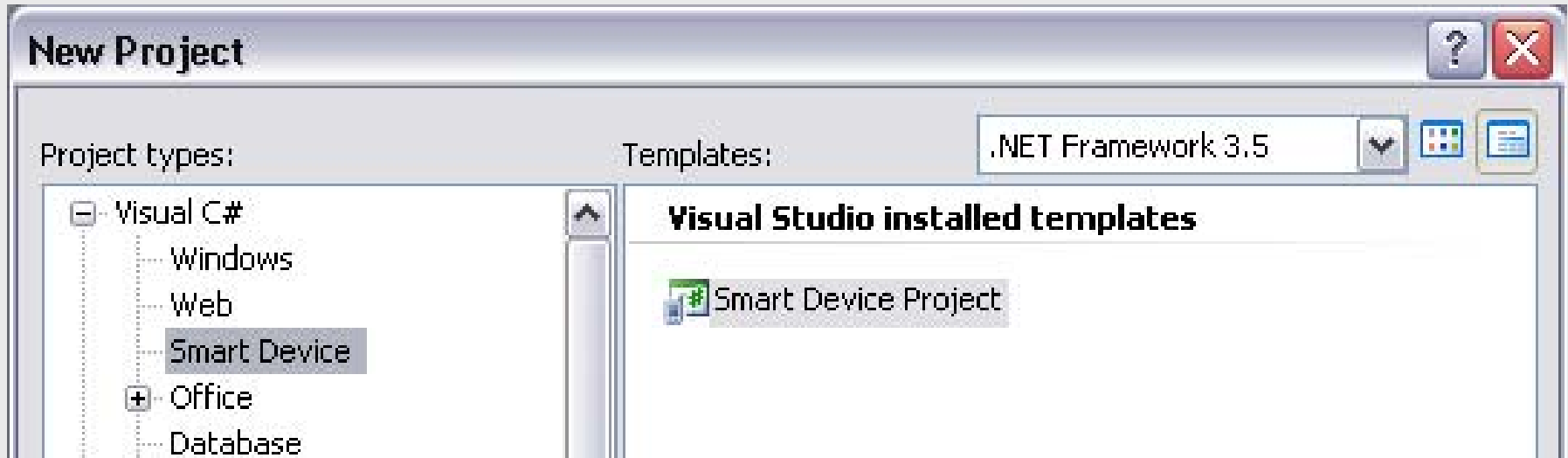
- **Loopback adapter – konekcija neumreženog računala**
 - linkovi na instalacije i upute na kraju folija (Reference)
- Virtual PC 2007 - Prilagodnik (zapravo *driver*) za virtualno mrežno okruženje

❑ MS ActiveSync (XP) ili Device Center (Vista, Win 7) - opcionalno

- omogućuje sinkronizaciju između stolnog i ručnog računala (Craddle)
- <http://www.microsoft.com/windowsmobile/activesync/activesync45.mspix>
- <http://www.microsoft.com/windowsmobile/devicecenter.mspix>

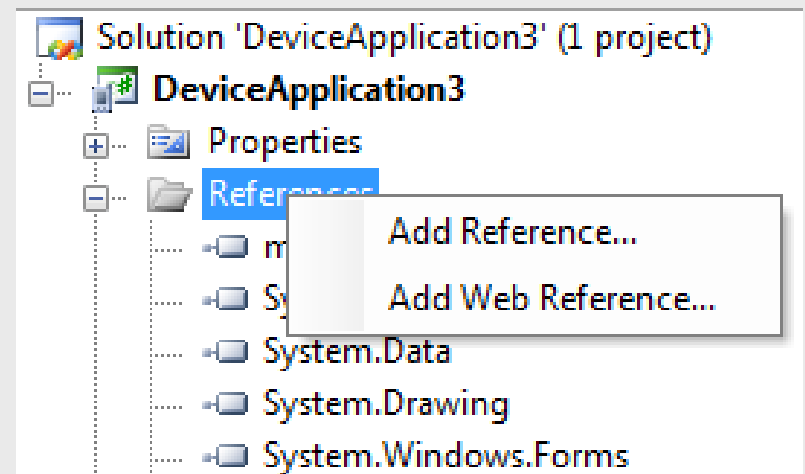
Stvaranje projekta za mobilne uređaje

❑ *File \ New ... \ Project – Smart Device*



❑ U reference stvorenog projekta potrebno je dodati

- `Microsoft.WindowsCE.Forms`
- `System.Data.SqlServerCe`
 - opcionalno, za rad s SQL bazom podataka na mobilnom uređaju



Standardna mobilna aplikacija

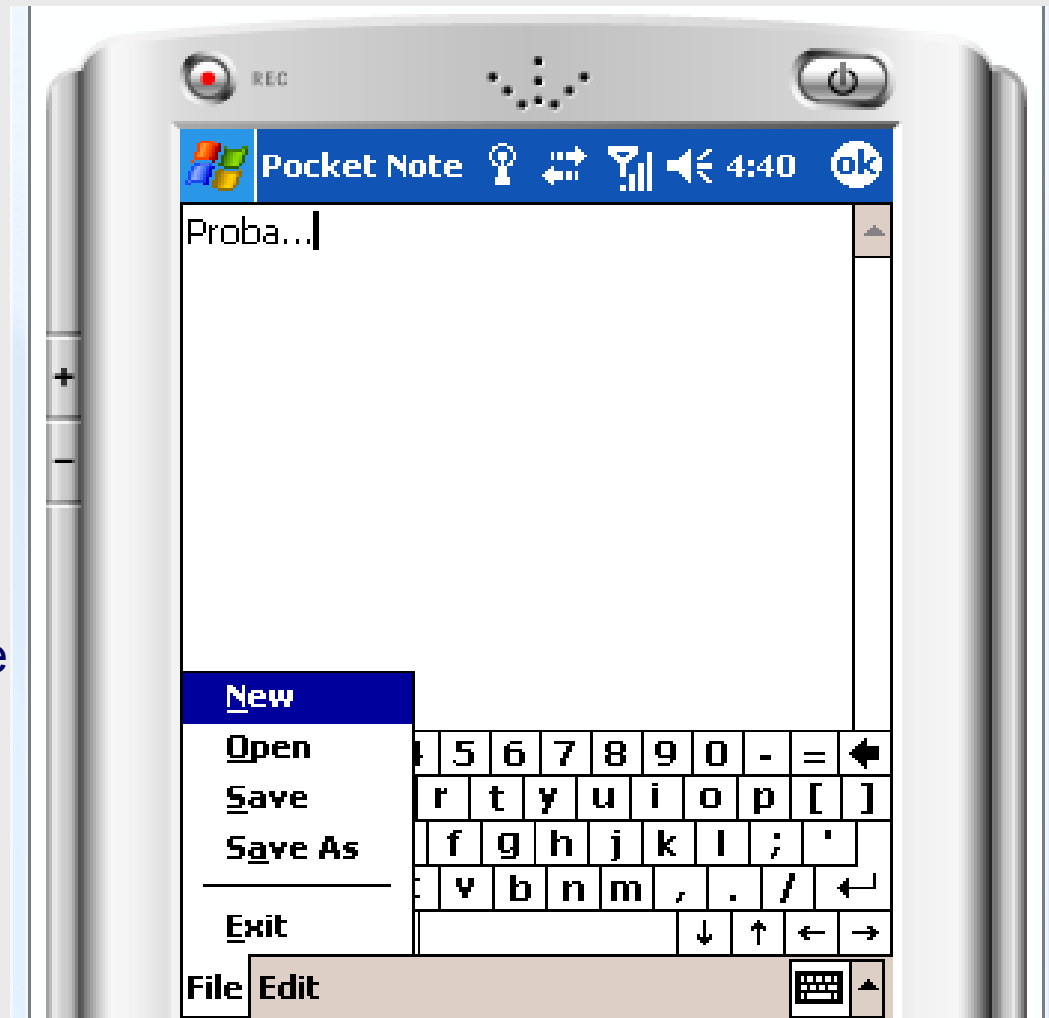
❑ Primjer: Mobilne \ PocketNotepad

❑ Funkcije uobičajenog programa

- Pregled, uređivanje i spremanje tekstovnog dokumenta

❑ Testiranje aplikacije

- radi bržeg izvođenja, testiranje stabilne aplikacije u emulatoru bolje je pokretati sa *Start without debugging*
- u slučaju više projekata potvrditi ugradnju (Deploy) samo željenog programa



Prostor imena Microsoft.WindowsCE.Forms

❑ Prostor imena Microsoft.WindowsCE.Forms

- Sadrži razrede za programiranje aplikacija na mobilnim uređajima (koristeći .NET Compact Framework)

❑ Kontrola InputPanel

- `Microsoft.WindowsCE.Forms.InputPanel`
- Sučelje između korisnika i SIP na mobilnom uređaju
- SIP (*Soft Input Panel*)
 - služi za jednostavniji unos podataka u mobilni uređaj
 - najčešće preko virtualne tipkovnice
- Neka svojstva i događaji:
 - `Enabled` – određuje vidljivost SIP
 - `CurrentInputMethod` – Keyboard, Letter Recognizer, Block Recognizer
 - Događaj `EnabledChanged` – okida se pri promjeni vidljivosti SIP

❑ Svojstvo forme MinimizeBox

- `True` – po zatvaranju forme program nastavlja raditi u pozadini
- `False` – klik na Ok (gornji lijevi kut) zatvara program

Kontrola InputPanel

❑ Prikaz kontrole inputPanel ovisno o fokusu

```
this.textBox.GotFocus +=  
    new System.EventHandler(this.textBox_GotFocus);  
this.textBox.LostFocus +=  
    new System.EventHandler(this.textBox_LostFocus);  
  
private void textBox_GotFocus(object sender, System.EventArgs e)  
{  
    inputPanel.Enabled = true;  
}  
  
private void textBox_LostFocus(object sender, System.EventArgs e)  
{  
    inputPanel.Enabled = false;  
}
```

Uređivanje teksta



❑ Postupci za obradu događaja *cut*, *copy*, *paste*

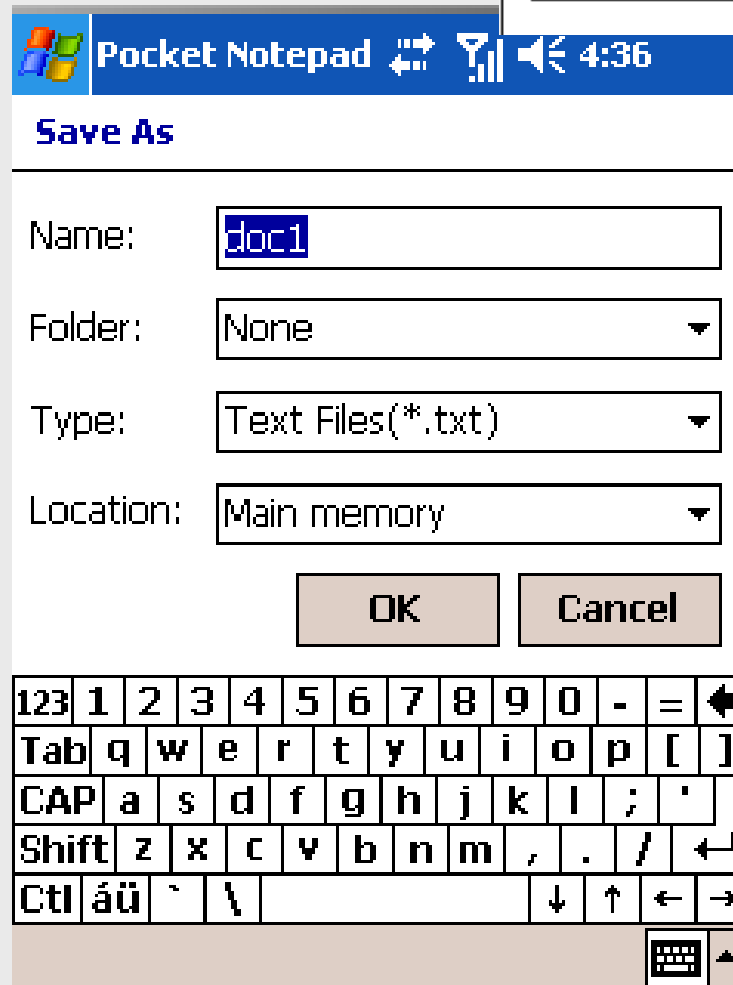
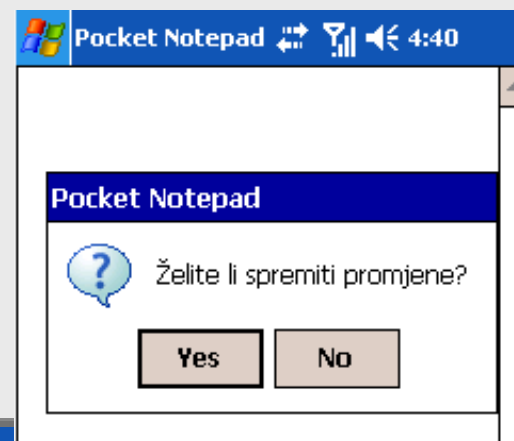
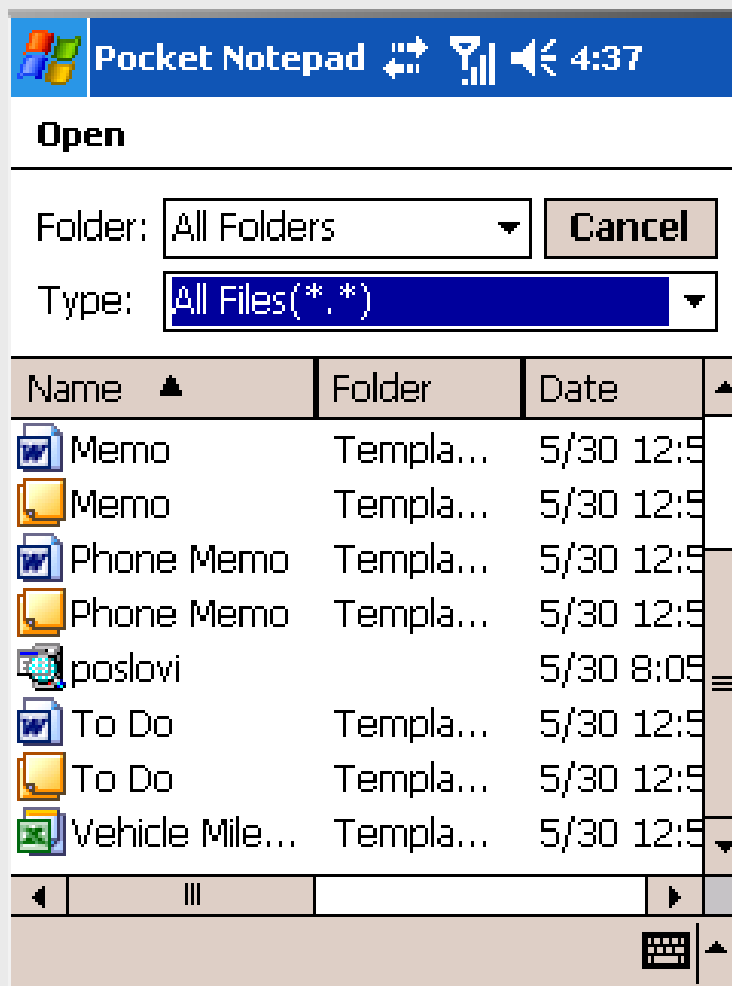
- Klik na element izbornika (*cut*, *copy*, *paste*)
- Kao privremeni spremnik označenog teksta koristi se član forme
 - `private string clipboard = "";`

```
private void mnuCut_Click(object sender, System.EventArgs e)
{
    clipboard = textBox.SelectedText;
    textBox.Text = textBox.Text.Remove(textBox.SelectionStart,
                                       textBox.SelectionLength);
}

private void mnuCopy_Click(object sender, System.EventArgs e)
{
    clipboard = textBox.SelectedText;
}

private void mnuPaste_Click(object sender, System.EventArgs e)
{
    textBox.SelectedText = clipboard;
    textBox.ScrollToCaret();
}
```

Dijalozi na mobilnom uređaju



Dijalozi na mobilnom uređaju (nastavak)

❑ *SaveFileDialog*

```
private void mnuSaveAs_Click(object sender, System.EventArgs e) {  
    if (saveFileDialog.ShowDialog() == DialogResult.OK) {  
        fileName = saveFileDialog.FileName;  
        Save();  
    }  
private bool Save() {  
    ...  
    using (TextWriter tw = File.CreateText(fileName)) {  
        tw.Write(textBox.Text);  
    }  
}
```

❑ *MessageBox*

```
private void mnuNew_Click(object sender, System.EventArgs e) {  
    if (hasChanges) {  
        DialogResult result = MessageBox.Show(...);  
        if (result == DialogResult.Yes) { ... }  
    }  
}
```

Lokalna evidencija na mobilnom uređaju

❑ **Primjer:**  Mobilne \ EvidencijaPoslova

❑ **Aplikacija za vođenje poslova**

- Mogućnost dodavanja i pregleda poslova za današnji dan
- Podaci se spremaju u XML datoteku ili u bazu podataka na mobilnom uređaju



Inicijalizacija zaslonske maske i povezivanje

❑ Inicijalizacija zaslonske maske (AppointmentList_Load)

```
//Hrvatska verzija dana u tjednu
System.Globalization.CultureInfo hrvatski =
    System.Globalization.CultureInfo.GetCultureInfo("hr-HR");

int day = (int)DateTime.Now.DayOfWeek;
DanLabel.Text = hrvatski.DateTimeFormat.DayNames[day];
DatumLabel.Text = DateTime.Now.ToString("dd.MM.yyyy");
```

❑ Inicijalno čitanje podataka (AppointmentList_Load)

❑ Postupak SetControls - povezivanje kontrola s podacima

```
vrijemeComboBox.DataSource = vrijemeStringArray;
rasporedListBox.DataSource = zadatakDataTable;
rasporedListBox.DisplayMember = "vrijeme";
nazivRasporedTextBox.DataBindings.Add("Text",
    zadatakDataTable, "naziv");
...
```

Lokalna baza podataka

❑ Prostor imena `System.Data.SqlServerCe`

- Razredi za rad s mobilnom bazom podataka
 - `SqlCeConnection`, `SqlCeCommand`, `SqlCeDataReader`, `SqlCeDataAdapter`, ...
- `SqlServerEngine` – razred za rad s objektima sustava za upravljanje mobilnom bazom podataka (kreiranje baze podataka)

❑ Baza podataka na mobilnom uređaju je datoteka s ekstenzijom `.sdf`

❑ Primjer Mobilne \ EvidencijaPoslova

- Stvaranje baze podataka i tablice

```
private void CreateDatabase() {  
    SqlCeEngine newSqlDB = new SqlCeEngine(sqlCEConnectionString);  
    newSqlDB.CreateDatabase();  
    SqlCeCommand makeTableCommand = new SqlCeCommand("CREATE TABLE...",  
        new SqlCeConnection(sqlCEConnectionString)  
    );  
    makeTableCommand.Connection.Open();  
    makeTableCommand.ExecuteNonQuery();  
}
```

Čitanje podataka iz baze podataka

❑ Čitanje zapisa iz baze podataka na mobilnom uređaju

```
private void GetDataFromSLQCE()  
{  
    Cursor.Current = Cursors.WaitCursor;  
    zadatakDataTable.Clear();  
    using (SqlCeConnection conn = new  
        SqlCeConnection(sqlCEConnectionString)) {  
        string sqlQuery =  
            "Select vrijeme, naziv, posao from raspored order by vrijeme";  
        using (SqlCeDataAdapter zadatakDataAdapter = new  
            SqlCeDataAdapter(sqlQuery, conn)) {  
            zadatakDataAdapter.Fill(zadatakDataTable);  
        }  
    }  
    Cursor.Current = Cursors.Default;  
}
```

- Postavljanje izgleda kursora na *WaitCursor* dok se podaci učitavaju, te vraćanje na podrazumijevani (*Default*) nakon učitavanja

Spremanje podataka u bazu podataka

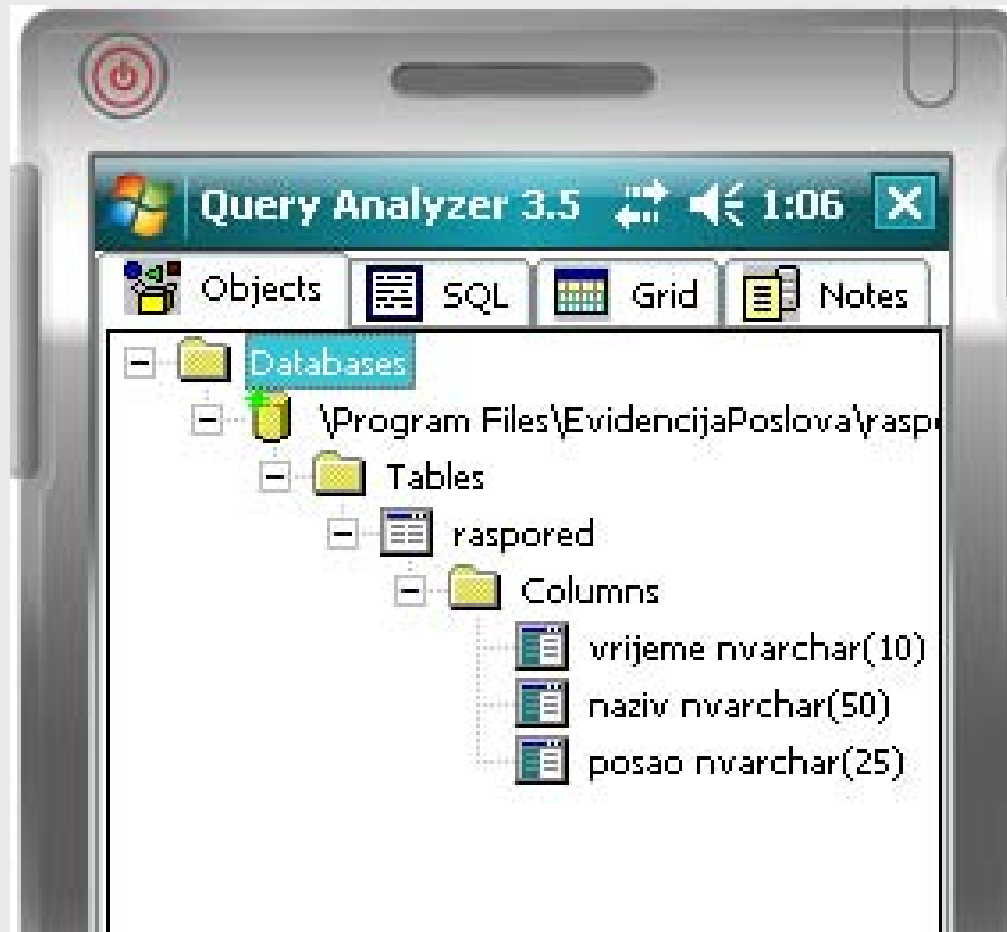
❑ Postupak SaveToSql

```
DataRow row = AddNewRow(); //postupak za dodavanje novog retka
using (SqlCeConnection conn = new SqlCeConnection(sqlCEConnectionString))
{
    conn.Open();
    using (SqlCeCommand newDataInsertCommand = new SqlCeCommand()) {
        newDataInsertCommand.Connection = conn;
        newDataInsertCommand.CommandText =
            "INSERT INTO raspored (vrijeme, naziv, posao) VALUES (?, ?, ?)";
        newDataInsertCommand.Parameters.Add("vrijeme", SqlDbType.NVarChar, 10);
        newDataInsertCommand.Parameters.Add("naziv", SqlDbType.NVarChar, 50);
        newDataInsertCommand.Parameters.Add("posao", SqlDbType.NVarChar, 25);
        newDataInsertCommand.Parameters["vrijeme"].Value = row["vrijeme"];
        newDataInsertCommand.Parameters["naziv"].Value = row["naziv"];
        newDataInsertCommand.Parameters["posao"].Value = row["posao"];
        newDataInsertCommand.ExecuteNonQuery();
    }
}
```

Upravljanje "mobilnom" bazom podataka

❑ Pregled baze podataka na mobilnom uređaju

- *Programs* → *Query Analyzer*
- Datoteka @*"Program Files\EvidencijaPoslova\raspored.sdf"*



Stvaranje i čitanje XML datoteke

☐ **Primjer**  **Mobilne \ EvidencijaPoslova – raspored.xml**

☐ **BuildXMLDoc - kreiranje datoteke (ako ne postoji na uređaju)**

```
string xmlDataFile = @"Program Files\EvidencijaPoslova\raspored.xml";  
...  
XmlDocument xmlDoc = new XmlDocument();  
XmlDeclaration dec = xmlDoc.CreateXmlDeclaration("1.0", "utf-8", null);  
xmlDoc.AppendChild(dec);  
XmlElement root = xmlDoc.CreateElement("raspored");  
xmlDoc.AppendChild(root);
```

☐ **Postupak GetDataFromXML - učitavanje iz datoteke u tablicu**

```
DataSet ds = new DataSet();  
ds.ReadXml(xmlDataFile);  
zadatakDataTable = ds.Tables[0];  
rasporedListBox.DataSource = zadatakDataTable;
```


Zapisivanje u XML datoteku

❑ Vlastiti postupak SaveToXML

- Dodavanje elementa “zadatak” s vrijednostima s korisničkog sučelja

```
XmlElement zadatakNode = xmlDoc.CreateElement("zadatak");  
xmlDoc.DocumentElement.AppendChild(zadatakNode);
```

```
XmlElement xmlElement = xmlDoc.CreateElement("vrijeme");  
xmlElement.InnerText = newRow["vrijeme"].ToString();  
zadatakNode.AppendChild(xmlElement);
```

```
XmlElement = xmlDoc.CreateElement("naziv");  
xmlElement.InnerText = newRow["naziv"].ToString();  
zadatakNode.AppendChild(xmlElement);
```

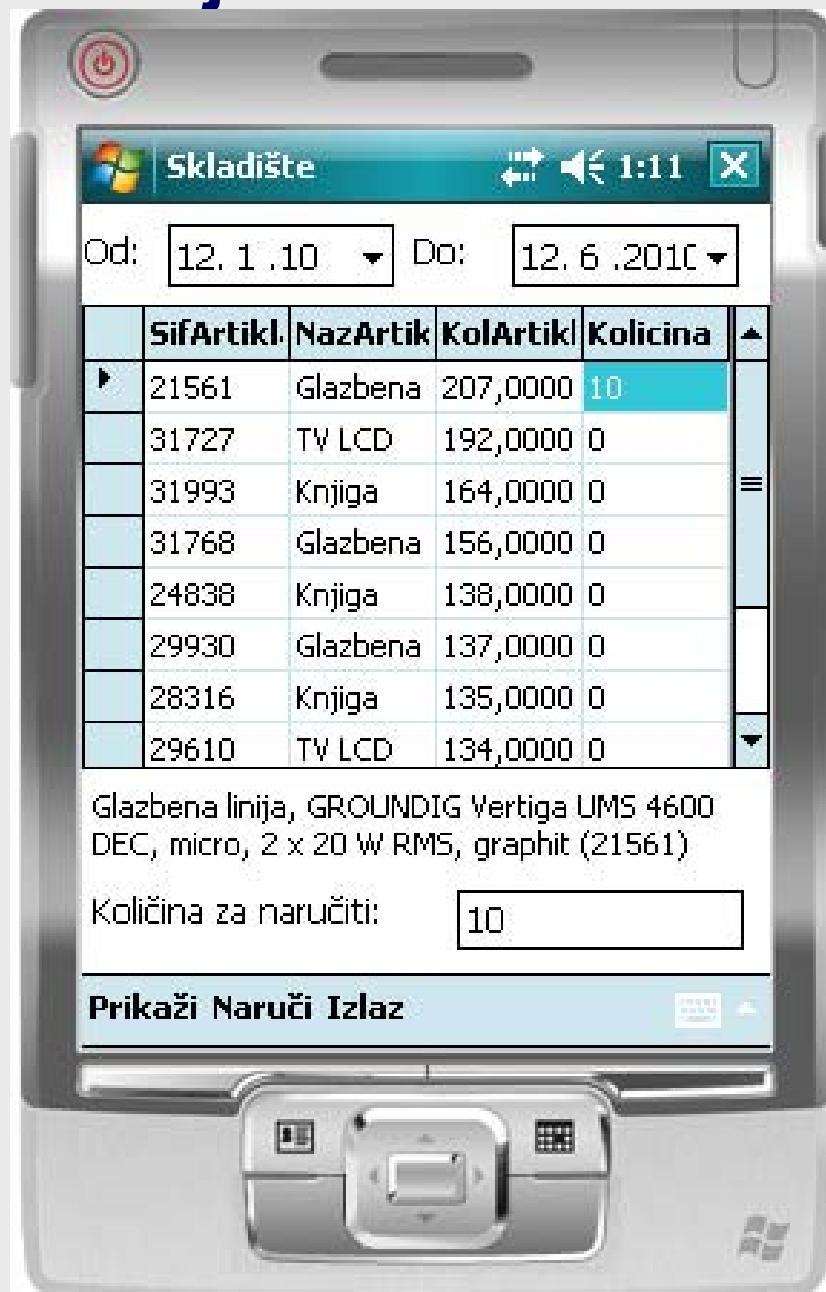
```
XmlElement = xmlDoc.CreateElement("posao");  
xmlElement.InnerText = newRow["posao"].ToString();  
zadatakNode.AppendChild(xmlElement);
```

```
xmlDoc.Save(xmlDataFile);
```

Primjer spajanja mobilnog uređaja na web servis

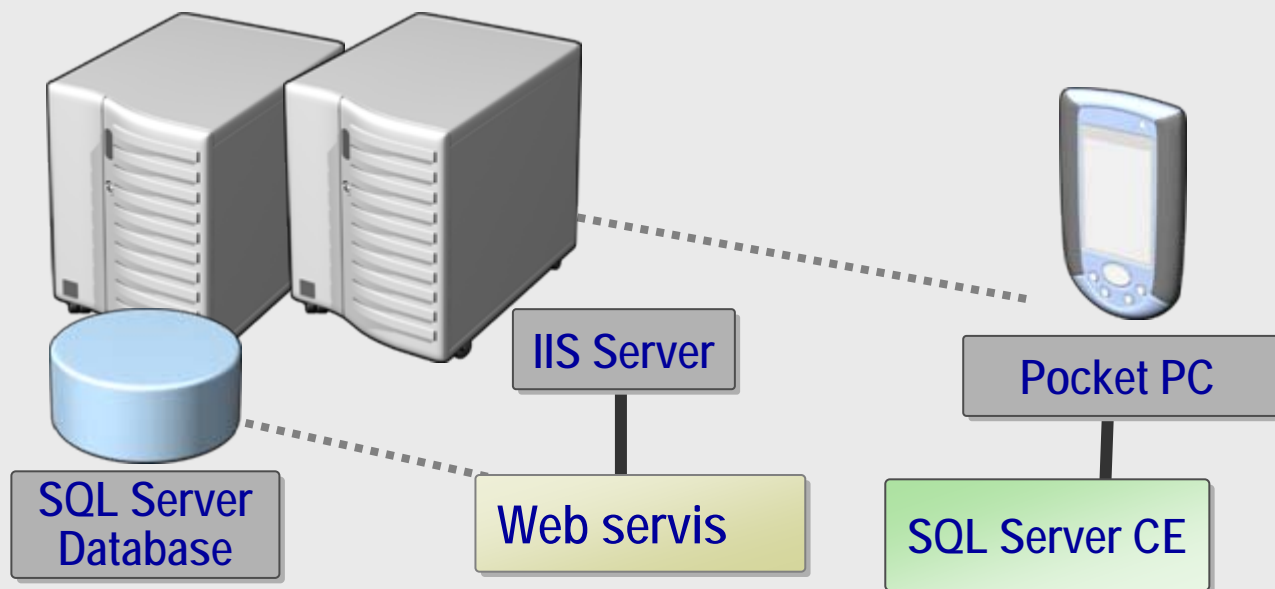
□ Primjer: Mobilne\Skladiste

- Skladištar unosi datume, te dohvaća popis artikala i prodane količine za odabrano razdoblje iz središnje baze podataka
- Skladištar pregledava popis te evidentira koliko pojedinih artikala treba naručiti
- Narudžba se zapisuje u središnju bazu podataka



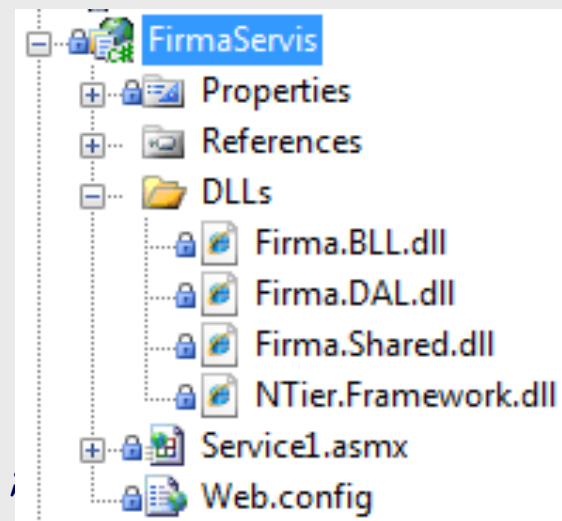
Udaljena baza podataka

❑ Pristup podacima putem Web servisa



❑ Primjer: Mobilne \ FirmaServis

- Dodavanje .dll datoteka
 - *Add → Existing Item* – odabrati datoteke
- Referenciranje odabranih *dll* s *Add Reference*
- Korištenje razreda definiranih u .dll datotekama
 - `using Firma;`
 - `StavkaList lista = new StavkaList();`



Dohvat podataka putem web servisa

❏ Primjer: Mobilne \ FirmaServis - DohvatiStavkeUkupno

- Dohvat liste ukupnog broja prodanih artikala u zadanom razdoblju
- Direktan upit na bazu, spremanje u `StavkaList` iz *Firma.Shared.dll*

```
[WebMethod]
```

```
public StavkaList DohvatiStavkeUkupno(  
    DateTime DatumOd, DateTime DatumDo){  
    upit = "...";  
    SqlConnection conn = new  
        SqlConnection(ConfigurationManager.ConnectionStrings  
            ["ConnectionString"].ToString());  
    ...  
    StavkaList lista = new StavkaList();  
    while (reader.Read()){  
        Stavka s = new Stavka();  
        ...  
        lista.Add(s);  
    }  
    return lista;}  

```

Pohrana podataka putem web servisa

❑ Stvaranje narudžbe od dobivenih stavki – NapraviNarudzbu

- Stvaranje novog dokumenta tipa 'N' (Naružba)
- Dodavanje stavki predanih kao parametar postupka
- Poziv SaveChanges na BLL sloju

[WebMethod]

```
public void NapraviNarudzbu(Stavka[] listaStavki){
    DokumentList dokumenti = new DokumentList();
    Dokument dok = dokumenti.AddNew();
    dok.VrDokumenta = "N";
    dok.BrDokumenta = NaredniBrDokumenta();
    dok.IdPartnera = 0;
    ...
    foreach (Stavka s in listaStavki){
        Stavka nova = dok.Stavke.AddNew();
        nova.SifArtikla = s.SifArtikla;
        ...}
    DokumentBllProvider dokumentBll = new DokumentBllProvider();
    dokumentBll.SaveChanges(dokumenti.GetChanges());}
```

Pohrana podataka putem web servisa

- ❑ **Pri stvaranju novog dokumenta ne znamo od koga će se nabaviti pa uvodimo "nepoznatog partnera"**

- U tablicu Partner dodajemo (jednom) partnera s IdPartnera=0
- IdPartnera je Identity! – SET IDENTITY_INSERT Partner ON

- ❑ **BrDokumenta mora biti jedinstven**

- Dohvaćamo (u web servisu) naredni broj dokumenta kao najveći broj postojećih dokumenata u tablici *Dokument* uvećan za 1

```
SELECT MAX (BrDokumenta)+1 from Dokument
```

- ❑ **Upit nad bazom podataka**

```
SET IDENTITY_INSERT Partner ON
INSERT INTO Partner
        (IdPartnera
        ,TipPartnera
        ,IdMjestaPartnera
        ,AdrPartnera
        ,IdMjestaIsporuke
        ,AdrIsporuke)
VALUES
        (0
        , 'T'
        , null
        , null
        , null
        , null)
SET IDENTITY_INSERT Partner OFF
```

Spajanje na web servis

☐ Dodavanje reference na web servis

- Web servis treba biti objavljen (*publish*) na nekom poslužitelju, ne možemo ga pokrenuti iz *solutiona*, jer je mobilni uređaj drugo (udaljeno) računalo
- Potrebno je dodati referencu na web servis
- Mobilni uređaj mora biti spojen na Internet da bi mogao pokrenuti servis

☐ Kreiranje servisa - kopiranjem na poslužitelj

- kopiranje *.asmx* datoteke i *bin* foldera u *c:\inetpub\wwwroot\FirmaServis*
- stvaranje virtualnog kazala FirmaServis nad fizičkim folderom
- test - <http://localhost/firmaservis/skladiste.asmx>

☐ Na SQL Server potrebno je dodati korisnike baze podataka

- NT AUTHORITY\NETWORK SERVICE
- <POSLUZITELJ>\ASPNET
- SQL Server Manager → Security → Logins (npr. ASPNET, MojPC\ASPNET)
- SQL Server Manager → Databases → Firma → Security → Users (ASPNET)

Dohvat podataka na mobilni uređaj

❑ Prikaz dohvaćenih podataka u *DataGrid* kontroli

- Kontrola slična *DataGridView* kontroli windows aplikacija
- Referenca na `System.Windows.Forms.DataGrid`
- Elemente *DataGrid* za mobilni uređaj nije moguće uređivati!
- Nije moguće dodavati stupce koji nisu povezani na neki izvor podataka

❑ Uređivanje – mogući načini

- Korištenje nove forme za uređivanje podataka (npr. klik na grid -> dohvat elementa -> prikaz detalja elementa u kontrolama koje se mogu uređivati)
- Nevidljivi *TextBox* koji se prilikom promjene aktivne ćelije postavlja na njeno mjesto i omogućava unos – simulacija mreže koja se može uređivati

	SifArtikl.	NazArtik	KolArtikl	Kolicina	▲
	26035	DVD/DivX,	169.0000	0	
	21824	FireWire	143.0000	0	
	25018	DVD/DivX,	142.0000	0	≡
▶	24857	Knjiga	125.0000	10	
	30647	Notebook	124.0000	20	

Dohvat podataka na mobilni uređaj

❏ Primjer: Mobilne \ Skladiste – Form1 \ NapuniTablicu

- Spremanje podataka dobivenih od servisa u lokalnu tablicu
- proširenje tablice stupcem količine za narudžbu

```
private DataTable NapuniTablicu() {  
    ...  
    DataTable tablica = new DataTable();  
    tablica.Columns.Add("SifArtikla", typeof(int));  
    ...  
    tablica.Columns.Add("Kolicina", typeof(decimal));  
  
    Servis.Stavka[] lista = servis.DohvatiStavkeUkupno(datumOd,datumDo);  
    foreach (Servis.Stavka stavka in lista){  
        tablica.Rows.Add(stavka.SifArtikla, stavka.NazArtikla,  
            stavka.KolArtikla, 0);  
    }  
    return tablica;  
}
```

Uređivanje podataka

❑ Prikaz detalja za uređivanje

- Klikom mišem na redak mreže, prikazuju se detalji odabranog elementa i *TextBox* kontrola s mogućnošću unosa nove vrijednosti

25018	DVD/DivX	142.0000	0
24857	Knjiga	125.0000	10
30647	Notebook	124.0000	20
24830	Knjiga	122.0000	0
9399	GPS	121.0000	0
5382	Knjiga	120.0000	0

Knjiga "CCNA" + CD (Knjiga "CCNA" + CD)

Količina za naručiti:

```
private void dataGrid1_CurrentCellChanged(object sender, EventArgs e){
    if (indeksTrenutnogRetka >= 0){
        tablica.Rows[indeksTrenutnogRetka][(int)Redak.Kolicina] =
            Convert.ToDecimal(textBoxKolicina.Text.ToString());

        indeksTrenutnogRetka = dataGrid1.CurrentRowIndex;
        labelNazivArtikla.Text = tablica.Rows[indeksTrenutnogRetka]
            [(int)Redak.NazArtikla].ToString()
            + " (" + tablica.Rows[indeksTrenutnogRetka]
            [(int)Redak.SifArtikla].ToString() + ")";
        textBoxKolicina.Text = tablica.Rows[indeksTrenutnogRetka]
            [(int)Redak.Kolicina].ToString();
    }
}
```

Slanje podataka na web servis

- ❑ **Slanje stavki (artikala i količine) servisu koji će napraviti narudžbu**
 - Slanje samo onih stavki čija je količina >0

```
private void menuItemNaruci_Click(object sender, EventArgs e){
    List<Servis.Stavka> lista = new List<Servis.Stavka>();
    for (int i=0; i < tablica.Rows.Count; i++){
        DataRow redak = tablica.Rows[i];
        if ((decimal) redak[(int)Redak.Kolicina] > 0){
            Servis.Stavka stavka = new Servis.Stavka();
            stavka.SifArtikla = (int)redak[(int)Redak.SifArtikla];
            ...
            lista.Add(stavka);
        }
    }
    servis.NapraviNarudzbu(lista.ToArray());
}
```



Zadaci za vježbu

- ☐ Aplikaciju *EvidencijaPoslova* nadopuniti funkcionalnošću brisanja posla.
- ☐ Napisati aplikaciju za pregled, dodavanje i uređivanje podataka o poslovnim partnerima.
- ☐ Web servisu iz primjera dodati postupke za dohvat i spremanje podataka o artiklima (koristeći .dll datoteke iz WinFirma). Napraviti aplikaciju za pregled, izmjenu i pohranu promjena o artiklima.

Reference

❑ Resursi (TFS)

- Mobilne-Upute.PDF

❑ Instalacije loopback adaptera

- <http://www.windowsreference.com/windows-7/how-to-install-a-loopback-adapter-in-windows-7/>
- <http://support.microsoft.com/kb/839013>

❑ Instalacije novih emulatora

- <http://www.microsoft.com/downloads/details.aspx?familyid=06111a3a-a651-4745-88ef-3d48091a390b&displaylang=en>

❑ Ostali URL

- <http://msdn2.microsoft.com/en-us/windowsmobile/default.aspx>
- <http://www.microsoft.com/sql/editions/compact/default.mspx>
- <http://www.microsoft.com/windowsmobile/activesync/>
- <http://samples.gotdotnet.com/quickstart/CompactFramework/doc/sqlcemergereplication.aspx>
- [http://msdn.microsoft.com/en-us/library/ms229684\(VS.80\).aspx](http://msdn.microsoft.com/en-us/library/ms229684(VS.80).aspx)