Diario di lavoro

Luogo	SAM Trevano
Data	05.11.2019

Lavori svolti

La prima cosa che ho fatto è stata risolvere gli errori che mi impedivano di compilare, quindi di proseguire con lo svolgimento del progetto. Grazie all'aiuto del professore ho capito perché nonostante la classe AppDbContext fosse uguale a quella del progetto usato in precedenza (che riusciva a creare il database correttamente) le Migrations non funzionava. Quando nel progetto corrente infatti ho installato i pacchetti NuGet mi sono dimenticato di installare i pacchetti EntityFrameworkCore.Design e . Tools. Inoltre mi ero dimenticato di installare EntityFrameworkCore nel progetto WPF, questo era il motivo per cui veniva generato l'errore "Operazione o Metodo non implementato" che mi aveva tenuto occupato durante l'ultima lezione.
Ho deciso di eseguire il salvataggio sul database con Sqlite, in questo modo:

```
if (!optionsBuilder.IsConfigured)
    {
        optionsBuilder.UseSqlite("Data Source="
          + System.AppDomain.CurrentDomain.BaseDirectory
          + "QdcDb.sqlite");
    }
}
```

Dopo aver finalmente risolto questi errori ho preso da Gabriele alcune classi da inerire negli Helpers che permettono di creare un sistema di notifica tra i vari viewmodel. Le classi più importanti sono: Messenger.cs, MessageBase.cs, NotificationMessage.cs e Simpleloc.cs. (Ho anche cambiato il namespace da QdceValutazioni.Helpers a MVVM).
Dopodichè ho modificato leggermente MainViewModel per la gestione della navigazione tra le varie views. Ho aggiunto i riferimenti a tutti gli altri viewmodel:

```
/// <summary>
/// istanza dei ViewModel
/// </summary>
private QdcViewModel QdcVM;
private RequisitoViewModel RequisitoVM;
private DescrizioneViewModel DescrizioneVM;
private MotivazioneViewModel MotivazioneVM;
private MenuViewModel MenuVM;
```

Ho instanziato anche dei comandi IDelegateCommand per ogni viewmodel:

```
/// <summary>
/// istanza di IDelegateCommand per la delega delle operazioni
/// </summary>
public IDelegateCommand QdcListCommand { get; set; }
public IDelegateCommand RequisitoListCommand { get; set; }
public IDelegateCommand DescizioneListCommand { get; set; }
public IDelegateCommand MotivazioneListCommand { get; set; }
public IDelegateCommand MenuListCommand { get; set; }
```

Nel costruttore richiamo il metodo RegisterCommands() che viene sovrascritto da ViewBindableBase.cs. In questo metodo vengono creati i riferimenti ai viewmodel dichiarati in precedenza e vengono richiamati i rispettivi metodi OnList e CanList gestiti da DelegateCommand che si occupano di assegnare il viewmodel corrente:

```
private void OnQdcList(object obj)
{
    CurrentViewModel = QdcVM;
}
```

Poi ho aggiunto queste funzionalità anche a MenuViewModel, in modo che schiacciando il bottone Crea QDC si venga trasportati a quella view. Nelle view per permettere queste operazioni ho inserito le varie Binding, che funzionano in questo modo:

```
Command="{Binding Path=CreaQdcCommand}"
```

Infine ho ripreso la ricerca dei metodi per usare Office su VisualStudio, visto che dall'ultima volta sono passate più di due settimane.

Problemi riscontrati e soluzioni adottate

Oggi mi sono concentrato in parte sul risolvere degli errori, fortunatamente non ne sono sorti di nuovi.

Punto della situazione rispetto alla pianificazione

Purtroppo a causa di una leggera labirintite l'ultima settimana prima delle vacanze sono stato assente. Per questo motivo ho perso ben 12 ore di progetto. Al momento ho deciso di cercare di recuperare il meno possibile a casa perché non è questo lo scopo dle progetto, se a lungo andare mi accorgerò non riuscire a recuperare in tempo ovviamente impiegherò anche del tempo a casa per finire il progetto.

Programma di massima per la prossima giornata di lavoro

Finire l'integrazione di tutte le views, gestione degli insert di nuovi qdc e descrizioni (inizialmente senza gestione degli errori e dei campi).