

# Diario di lavoro

Luogo	SAM Trevano
Data	06.12.2019

## Lavori svolti

Ho testato le nuove view create ieri. Le view si chiamano **VisualizzaView** e **SelezionaView**.

**VisualizzaView** contiene due bottoni, uno che permette di aprire il Qdc appena creato e l'altro che permette di tornare al Menu. Questa View viene generata dopo RequisitiView.

**SelezionaView** contiene un DataGrid e un bottone. Nel DataGrid vengono stampati tutti i Qdc che sono stati creati e si può selezionare quello desiderato. Il bottone riporta a MotivazioneView. Questa View viene generata dopo MenuView se si preme il pulsante Seleziona Qdc.

Dopo aver testato il funzionamento ho creato la funzione Get che permette di stampare i Qdc presenti nel Database nella DataGrid di SelezionaView. Per fare questo ho semplicemente inserito queste due righe nel costruttore di SelezionaViewModel:

```
QdcDbDataRepository repoQdc = new QdcDbDataRepository(new AppDbContext());
Qdcs = new ObservableCollection<Qdc>(repoQdc.Get());
```

Qdcs è riferito alla DataGrid nella View:

```
<DataGrid x:Name="QdcDataGrid" Grid.Column="0" Grid.Row="1" IsReadOnly="True"
          ItemsSource="{Binding Path=Qdcs}">
</DataGrid>
```

Dopo essermi assicurato che il Get funziona ho cambiato l'indirizzamento delle varie views per renderlo quello voluto e definitivo. Per fare questo ho semplicemente dovuto modificare qualche Binding nelle varie Views.

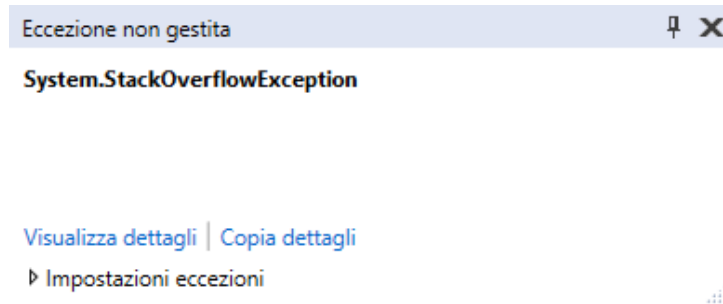
Dopodiché ho modificato il database, anche in questo caso la versione attuale sarà molto probabilmente quella definitiva. Nella tabella Qdc ho aggiunto i seguenti campi:

**PathTemplate:** Contiene il percorso del file di Template dei Qdc  
**PathRequisiti:** Contiene il percorso del file contenente tutti i requisiti per esteso  
**PathQdc:** Contiene il percorso nel quale salvare il Qdc.  
**PathVoti:** Contiene il percorso nel quale salvare i Voti.  
**NomePerito2:** Nome del secondo perito.  
**CogomePerito2:** Cognome del secondo perito.  
**MailPerito2:** Indirizzo email del secondo perito.

Ho aggiunto questi campi a Qdc (Model) e ho eseguito il comando Update-Database dalla shell di Nu-Get. Poi ho aggiunto staticamente tutti i requisiti specifici nella tabella Requisiti. Ora la tabella contiene tutti i requisiti esistenti, sia quelli standard per ogni quaderno dei compiti, sia quelli specifici da assegnare ogni volta. Alla fine della lezione ho aggiornato la documentazione e il gantt consuntivo su quanto svolto nelle ultime due/tre settimane.

### Problemi riscontrati e soluzioni adottate

Il collegamento da VisualizzaView a MenuView non funziona. Viene generato un errore al Runtime che impedisce di aprire l'applicazione:



Questa eccezione viene generata nel file BindableBase. Se cambio la destinazione da MenuView a QdcView o a MainView l'errore persiste ma viene generato in un altro file. Se la destinazione che inserisco è SelezionaView invece il tutto funziona senza errori. Non sono riuscito a risolvere questo errore, penso che sia dovuto al fatto che sto cercando di tornare a una view già istanziata, perché quando passo a SelezionaView (che non è mai stata istanziata fino a quel momento) l'errore non viene generato.

### Punto della situazione rispetto alla pianificazione

Sono abbastanza in ritardo. Dato le diverse ore di assenza a partire da ora dovrò cominciare a lavorare anche un po' a casa per riuscire a terminare il progetto. Questo weekend probabilmente recupererò un paio di ore.

### Programma di massima per la prossima giornata di lavoro

La prossima lezione continuerò l'implementazione della scelta di un Qdc già esistente e quindi dell'aggiunta dei voti e delle motivazioni. Nello specifico devo fare dei Get per ogni requisito di ogni Qdc e poi degli insert per salvare i voti e le motivazioni scritti per quel voto. Questa parte è su MotivazioneView e si rifà alla tabella Assegnazione del database.