Barcoin

1 Introduzione 3

1.1 Informazioni sul progetto 3

1.2 Abstract 3

1.3 Scopo 3

Analisi 3

1.4 Analisi del dominio 3

1.5 Analisi dei costi e benefici 3

1.6 Analisi tecnologia blockchain 4

1.6.1 Abstract 4

1.6.2 Introduzione 4

1.6.3 Algoritmi di consenso, POW (Proof of Work) vs POS (Proof of Stake) 5

1.6.4 Struttura funzionale 6

1.6.5 Ulteriori informazioni 7

1.7 Analisi e specifica dei requisiti 8

1.8 Pianificazione 11

1.8.1 Software 12

1.8.2 Hardware 12

2 Progettazione 12

2.1 Design dei dati e database 12

2.2 Design delle interfacce 12

3 Implementazione 13

3.1 Grafica / XAML 13

3.2 Database / MySQL 16

3.3 Logica / C# 17

3.4 Algoritmo di credito 25

3.5 Dipendenze 25

4 Test 26

4.1 Risultati test 27

4.2 Mancanze/limitazioni conosciute 27

5 Consuntivo 28

6 Conclusioni 29

6.1 Sviluppi futuri 29

6.2 Considerazioni personali 29

7 Bibliografia 29

7.1 Sitografia 29

8 Allegati 29

9 Glossario 30

# Introduzione

## Informazioni sul progetto

Docente Responsabile: Ugo Bernasconi

Scuola: Scuola d’Arti Mestieri Trevano

Sezione: Informatica

Materia: Modulo 306

Data inizio progetto: 08.01.2019

Data consegna progetto: 10.05.2019

## Abstract

## Scopo

Lo scopo di questo progetto consiste nel rendere l’ultima release del prodotto Barcoin integrato con tecnologia blockchain e supporto multiutenza, così da rendere non solo le transazioni immutabili e più sicure ma fornendo anche il possibile utilizzo in un gruppo di lavoro.

Barcoin subirà inoltre delle modifiche di maggiore importanza per quanto riguarda UI e Database.

## Analisi

## Analisi del dominio

Ad oggi esistono diversi software che permettono il micro-management di finanze, vengono usati specialmente nei paesi più poveri, dove le persone “*non riescono ad ottenere credito e altri servizi dalle istituzioni finanziarie tradizionali per due ragioni: vengono reputati non solvibili (unbanked) e/o i costi legati all'offerta di questi servizi sono eccessivi e rendono l'operazione non conveniente economicamente*.” **Wikipedia**

Molti di questi applicativi risultano però proprietari e quindi non aperti al dominio pubblico, quelli che invece sono resi al pubblico risultano confusionari o poco ergonomici per l’utilizzo di tutti i giorni.

Barcoin è un software sviluppato completamente con tecnologie open-source e che offre le basi necessarie per portare al termine lo scopo definito dal progetto.

## Analisi dei costi e benefici

|  |  |
| --- | --- |
| **Categoria** | **Costo** |
| Personale | 174 ore \* 1 persona \* 60 CHF/h = 10440 CHF |
| **TOT** | **10440 CHF** |

## Analisi tecnologia blockchain

### Abstract

Una blockchain è essenzialmente un database decentralizzato, distribuito pubblicamente e contenente tutte le transazioni o eventi digitali che sono stati eseguiti e condivisi tra le sue parti partecipanti. Ogni pacchetto di dati al suo interno viene definito blocco.

La rete blockchain si occupa di certificare l’integrità e la sicurezza del sistema complessivo tramite differenti tecniche crittografiche ed algoritmiche, per esempio effettuare un hash sul blocco stesso ed inserire una referenza al suo precedente.

I vari blocchi possono essere salvati in differenti catene, oltre alla principale è infatti possibile creare dei rami esterni e parzialmente indipendenti; la catena più lunga è definita determinante in quanto possiede la maggior parte di nodi attivi. Finché quest’ultima non viene attaccata non esistono virtualmente metodi per prendere controllo della rete, inoltre, i nodi possono unirsi e abbandonare la rete a proprio piacimento, a patto che al loro rientro rispettino la catena più lunga creatasi durante la loro mancanza.

Ogni transazione nel registro pubblico è verificata dal consenso dei partecipanti al sistema, questo consenso viene raggiunto in modi diversi a dipendenza dell’algoritmo adottato; una volta confermata le informazioni al suo interno non possono mai essere cancellate o modificate.

Per usare un'analogia di base, è facile rubare una caramella all’interno di un contenitore tenuto in un armadio ma risulta molto difficile se esposto in un mercato osservato da migliaia di persone.

### Introduzione

L'attuale economia digitale si basa sulla dipendenza da una certa autorità attendibile.

Le nostre transazioni online si basano sulla fiducia di qualcuno/qualcosa, può essere la banca che ci offre il servizio di e-banking o il nostro ISP che ci offre un servizio internet per casa e/o telefono.  
In entrambi i casi ci affidiamo ad una terza entità per la sicurezza e la privacy delle nostre informazioni digitali, queste fonti di terze parti possono essere violate, manipolate o compromesse.

La tecnologia blockchain è nata per prevenire questi problemi legati alle terze persone.

### Algoritmi di consenso, POW (Proof of Work) vs POS (Proof of Stake)

Questi sono i due principali algoritmi tramite i quali blockchain opera e sono tra i fattori decisivi da considerare al momento di decidere se investire in un progetto di criptovaluta o meno, alcuni degli aspetti critici da considerare in questa scelta includono la velocità, le applicazioni, e l’efficienza di calcolo.

POW

Proof of Work è l’algoritmo più utilizzato e riconosciuto, molti dei capisaldi del settore lo adoperano come Bitcoin (BTC), Ethereum (ETH) e Dash (DASH).

Come minatore di criptovalute che utilizzano POW è necessario risolvere complessi problemi matematici alla ricerca della valuta virtuale, questo rende l’azione nel tempo esponenzialmente sempre più difficile. Dopo aver risolto con successo diversi calcoli per varie transazioni, le transazioni verificate vengono infine messe insieme e memorizzate in un nuovo blocco sul registro pubblico.

Punti a favore di POW

* Qui la maggior parte dei voti sono da sviluppatori, minatori e altri importanti membri della comunità, scelte congiunte.

Punti a sfavore di POW

* Spreco sia di potenza di calcolo che di elettricità nella generazione dei presupposti casuali.

POS

In seguito al fatto che l'estrazione in POW sia diventata lenta e costosa nel tempo, l’algoritmo POS è entrato in gioco. A differenza di POW che utilizza la potenza del computer, POS capitalizza sulla carenza della moneta. Si concentra di più sulla proprietà degli utenti, invece di una soluzione ad un calcolo.

L’utente deve dimostrare di essere il proprietario di un certo valore in criptovaluta, come tale ottiene dei benefici e crescite del suo bene sull’avanzare della linea temporale.

Punti a favore di POS

* Non c'è bisogno di computer avanzati a livello hardware
* L'intero processo è a basso costo elettrico

Punti a sfavore di POS

* POS offre una piattaforma che contraddice uno dei principi della tecnologia blockchain, quello di non avere un'autorità centrale, consentendo a coloro che possiedono più monete di dettare le modifiche da attuare sulla piattaforma.
* I possessori di ingenti somme di criptovaluta subiscono un aumento maggiore rispetto a chi ne possiede in quantità minore.

POA (Extra)

In questo algoritmo, gli amministratori di una rete blockchain convalidano e approvano le transazioni.  
Si tratta di una soluzione altamente centralizzata, ma anche molto efficiente, potrebbe funzionare per reti private dove i membri conoscono e si fidano uno dell'altro.

### Struttura funzionale

In seguito viene dimostrato insieme ad alcune immagini significative il funzionamento di una rete blockchain, l’algoritmo di consenso applicato a questa specifica situazione è il POW.

Indirizzo1

Ogni utente all’interno della rete possiede un indirizzo univoco, questo viene usato per effettuare versamenti sul suo conto.

Si tratta di una stringa alfanumerica, solitamente lunga 32 caratteri.

Rif

Chiave privata2

Una chiava privata univoca viene accoppiata con ogni account utente, viene utilizzata per effettuare pagamenti verso altri indirizzi e **deve** rimanere confidenziale.

Rif



Chiave pubblica3

Una chiave pubblica viene generata applicando una funziona crittografica unilaterale sulla propria chiave privata, questa risultante può essere utilizzata per scoprire il proprio indirizzo su una rete specifica.



Chiave pubblica di script4

Questa chiave risulta essere una semplice conversione dell’indirizzo dalla sua forma “human friendly” ad una versione interpretabile dalla blockchain, la versione ottenuta dall’utente è solamente un’interfaccia di questo dato.

Rif





Rif

 Transazione5

Una transazione può non avere destinatario, o può averne diversi. Lo stesso si può dire per i mittenti. Su una blockchain, il mittente e il destinatario sono sempre astratti con una chiave pubblica di script, come dimostrato nei sotto capitoli precedenti.

Il complesso di informazioni finali in uscita, chiamato TxOut, è definito dalla quantità di cripto valuta inviata e la chiave pubblica di script del destinatario.

Punto d’uscita

Rif

Ogni TxOut è riferito in modo univoco a livello di blockchain dall'ID della transazione e dall’indice al suo interno. Chiamiamo tale riferimento un punto d’uscita.

Firma digitale

Tramite la propria chiave privata è possibile mascherare un messaggio da inviare all’interno della rete, quest’ultimo sarà riconoscibile e traducibile da tutte le persone

che conoscono la firma digitale dell’autore.

Mining

È detta mining l’azione che la comunità di una blockchain POW è sottoposta ad eseguire per ottenerne la sua cripto valuta.

I “miners” sono entità il cui unico obiettivo è quello di inserire una nuova transazione nella blockchain. Ognuno di loro tenta però di aggiungere un intero lotto di transazioni allo stesso tempo, andando così a formare un blocco invece che un record singolo. Altri nodi della rete (50%+1) confermano che il nuovo blocco obbedisca al protocollo di blockchain, se due minatori dovessero aggiungere un blocco allo stesso tempo viene data la priorità al ramo con l’indice di lavoro più alto, in altre parole, quello su cui sono stati eseguiti più cicli di calcolo. Se un minatore cercasse di includere una transazione non valida nel suo blocco, gli altri nodi non lo riconoscerebbero come valido e il minatore perderebbe l'investimento speso per la creazione del blocco, insieme alle possibili legittime transazioni.

Una volta che un minatore riesce a presentare un blocco valido, tutte le transazioni all'interno sono considerate confermate, quando questo accade tutti i minatori devono scartare il loro lavoro attuale e iniziare a lavorare su un nuovo blocco utilizzando nuove transazioni.

### Ulteriori informazioni

Blockchain è una tecnologia in continua crescita, le sue applicazioni diventano ogni giorno più varie ed accettate sia dalle comunità interne o esterne al suo sviluppo.

Migliaia di professionisti in crittografia, algoritmica e matematica si sfidano ogni giorno per trovare nuove soluzioni, cercando di combinare efficienza, sicurezza e scalabilità.

## Analisi e specifica dei requisiti

|  |  |
| --- | --- |
| **ID: REQ-01** | |
| **Nome** | Multiutenza |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | Deve esserci un form di login e registrazione per nuovi utenti Barcoin |
| **002** | Deve esserci un campo per il nome dell’utente di tipo testo |
| **003** | Deve esserci un’etichetta per il cognome del creditore di tipo testo |
| **004** | Deve esserci un’etichetta per la descrizione della linea di credito di tipo testo |

|  |  |
| --- | --- |
| **ID: REQ-02** | |
| **Nome** | Gestione in dettaglio delle linee di credito |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | Ogni linea di credito in anteprima deve avere una versione in dettaglio da visualizzare |
| **002** | Deve esserci un’etichetta per il tasso d’interesse di default di tipo testo |
| **003** | Deve esserci un’etichetta per il formato del tasso giornaliero di tipo testo |
| **004** | Deve esserci una tabella per la visualizzazione di ogni singola transazione di tipo **BarcoinTable** |
| **005** | Deve esserci un grafico per la visualizzazione del bilancio di tipo **BarcoinGraph** |

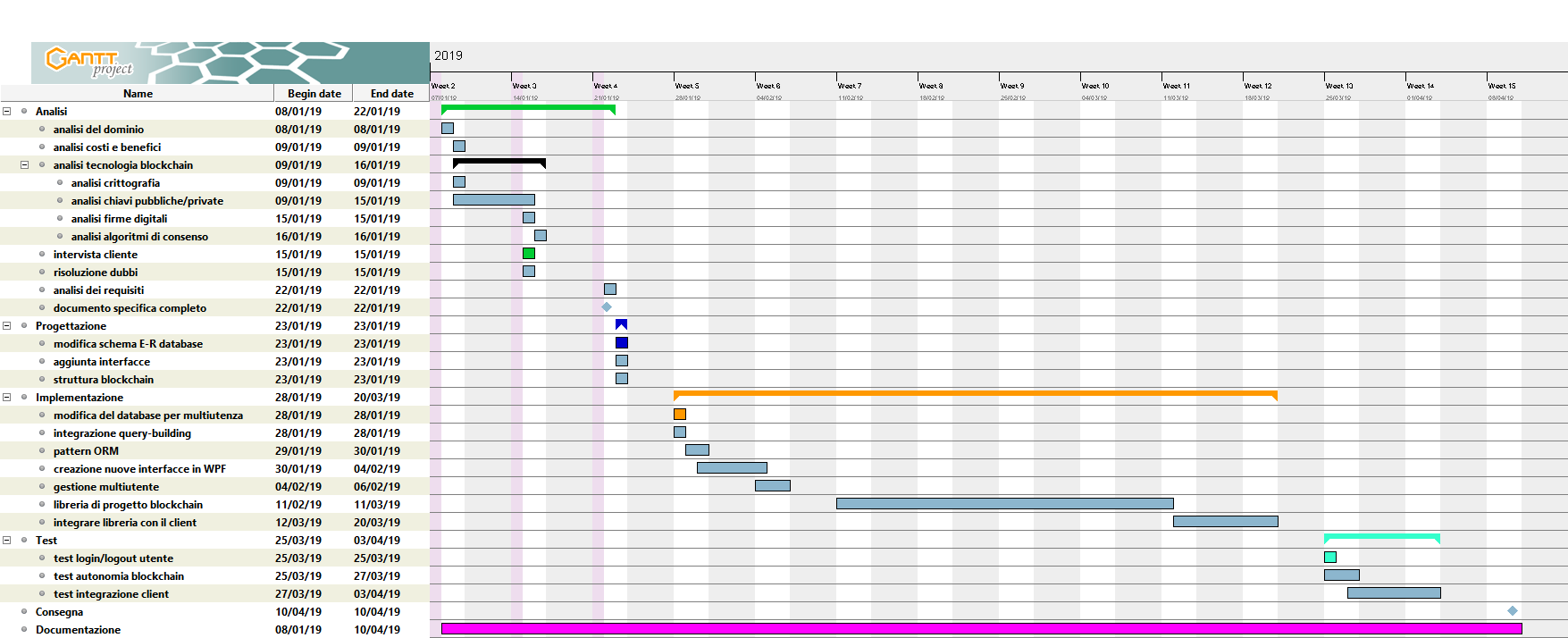
|  |  |
| --- | --- |
| **ID: REQ-03** | |
| **Nome** | Gestione BarcoinTable |
| **Priorità** | 1 |
| **Versione** | 1.2 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | Ogni transazione deve essere univoca e su righe diverse |
| **002** | Deve esserci un’etichetta per la data di prelievo/versamento di tipo testo |
| **003** | Deve esserci un’etichetta per il tasso di interesse effettivo di tipo testo |
| **004** | Deve esserci un’etichetta per l’importo di prelievo/versamento di tipo testo |
| **005** | Deve esserci un’etichetta per gli interessi maturati di tipo testo |
| **006** | Deve esserci un’etichetta per gli interessi pagati di tipo testo |
| **007** | Deve esserci un’etichetta per il bilancio degli interessi di tipo testo |
| **008** | Deve esserci un’etichetta per il principale pagato di tipo testo |
| **009** | Deve esserci un’etichetta per il bilancio del principale di tipo testo |
| **010** | Deve esserci un’etichetta per il totale in valuta dovuto di tipo testo |

|  |  |
| --- | --- |
| **ID: REQ-04** | |
| **Nome** | Gestione inserimento creditori |
| **Priorità** | 2 |
| **Versione** | 1.3 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | Tramite un bottone della pagina iniziale deve essere possibile raggiungere l’interfaccia per nuovi creditori |
| **002** | Deve esserci un campo per il nome del nuovo creditore di tipo testo |
| **003** | Deve esserci un campo per il cognome del nuovo creditore di tipo testo |
| **004** | Deve esserci un campo per la descrizione della nuova linea di credito di tipo testo |
| **005** | Deve esserci un campo per il tasso giornaliero di tipo numero intero |
| **006** | Deve esserci un campo per il tasso d’interesse di default di tipo numero decimale |

|  |  |
| --- | --- |
| **ID: REQ-05** | |
| **Nome** | Gestione inserimento transazioni |
| **Priorità** | 2 |
| **Versione** | 1.1 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | All’interno dell’interfaccia in dettaglio deve esserci un form per nuove transazioni |
| **002** | Deve esserci un campo per la data della nuova transazione di tipo G/M/A |
| **003** | Deve esserci un campo per il tasso d’interesse della nuova transazione di tipo numero decimale |
| **004** | Deve esserci un campo per il valore prelevato della nuova transazione di tipo numero interno. |

|  |  |
| --- | --- |
| **ID: REQ-06** | |
| **Nome** | Gestione esportazioni dati |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** |  |
| **Sotto requisiti** | |
| **001** | Devono esserci due key binding, ognuno dei quali per effettuare l’esportazione dei dati presenti all’interno della BarcoinTable in formato CSV o PDF. Questi devono chiaramente essere utilizzabili solamente quando ci si trova in un’interfaccia di dettaglio. |
| **002** | Premendo il tasto ‘e’ i dati devono essere esportati in formato CSV |
| **003** | Premendo il tasto ‘p’ i dati devono essere esportati in formato PDF |

## Pianificazione



### Software

Microsoft Word 2013

Gantt Project 2.8.5

Power Point 2013

Google Chrome 64.0

TortoiseGit 2.5.0

DB Schema 6.0.3

Visual Studio 2017 Community

Dev Express 18.1.5

DB Browser for SQLite 3.10.1

### Hardware

Laptop PC – Apple Macbook Air

# Progettazione

## Design dei dati e database

Il database è relativamente semplice, composto da 2 tabelle (**creditline** e **transaction**).

* La tabella **creditline** contiene alcuni dei dati anagrafici del creditore come nome e cognome (**creditorName** e **creditorSurname**), il motivo dell’apertura di tale credito (**description**) ed infine il valore percentuale del tasso d’interesse di base (**defaultInterestRate**) insieme al formato di tasso giornaliero (**diemRate**).
* La tabella **transaction** contiene i dati relativi alle transazioni dei creditori, la data di prelievo/versamento (**date**) con l’importo (**principal**), il tasso di interesse effettivo (**interestRate**) e la descrizione relativa (**description**).

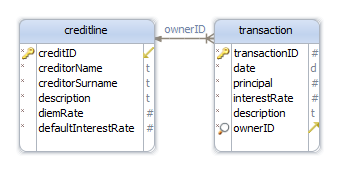


Figura Schema database creato con DbSchema

## Design delle interfacce

# Implementazione

## Grafica / XAML

L’implementazione grafica è una parte consistente del progetto, considerando la relazione al modello MVVM nel mio progetto con i relativi binding è stato necessario ragionare in modo accurato l’ordine di lavoro per ottenere delle interfacce leggere e funzionanti, seguendo comunque le linee guida tracciate nei mockup.

L’applicazione è composta da 3 interfacce principali ognuna delle quali serve un utilità diversa, andandole ad elencare:

* **MainView:** Interfaccia principale, contiene il titolo del progetto, versione corrente, un immagine con il comando per riportarsi allo stato inziale ed infine un controllore di contenuto per le altre due interfacce. All’avvio la DashboardView sarà selezionata come attiva.

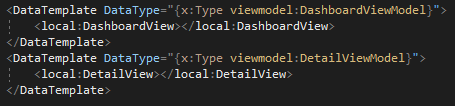
****

Figura Binding alle altre due interfacce

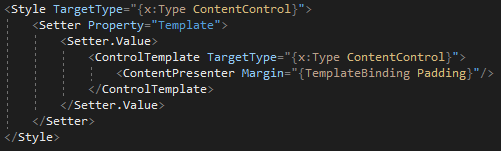
****

Figura Stile per ottenere un margine sul controllore di contenuto

**xaml**

Figura Binding per il logo di Barcoin

**xaml**

Figura Binding per visualizzare l'interfaccia attiva

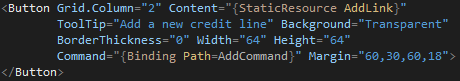


Figura Bottone con binding per raggiungere AddCreditorView

* **DashboardView:** Interfaccia per visualizzare le anteprime delle linee di credito contenute nel database, viene creato un bottone personalizzato per ognuna di queste e assegnato il comando per raggiungere la propria interfaccia DetailView.

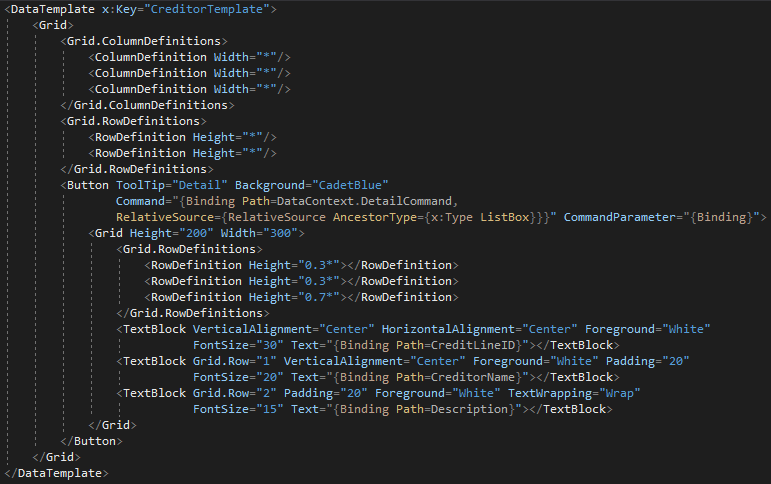
****

Figura Template applicato ad ogni bottone della Dashboard

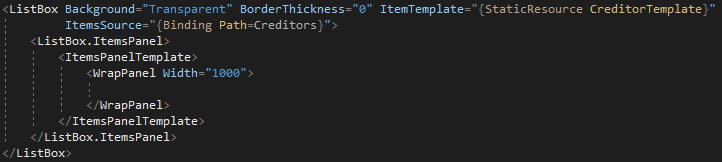
****

Figura Lista resa a ‘blocco’ dove vengono inseriti i bottoni

* **DetailView:** Interfaccia che implementa la maggior parte del progetto a livello grafico, oltre che alle informazioni secondarie, mostra infatti tutte le transazioni eseguite dal creditore scelto in una tabella filtrabile e ha al suo interno un grafico quale riassume il bilancio attuale.

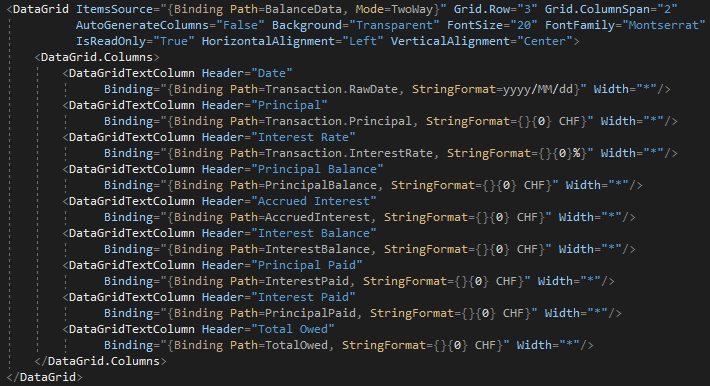


Figura Griglia contenente le transazioni del creditore

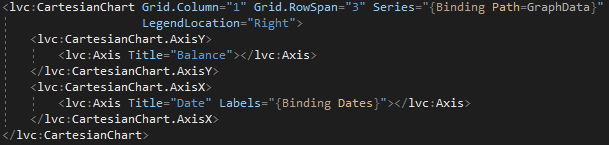


Figura Grafico del bilancio fatto con Live Charts

* **AddCreditorView:** Interfaccia per aggiungere nuovi creditori all’interno del sistema di Barcoin.

****

Figura I TextBox all'interno di questa interfaccia utilizzano tutti lo stile PlaceHolder, per rendere l'inserimento semplificato

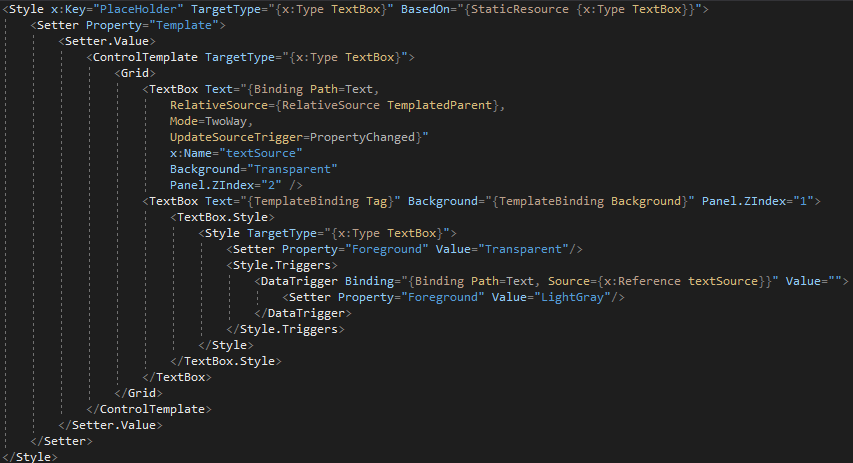


Figura Stile PlaceHolder applicato globalmente nel file App.xaml per ottenere l’effetto di testo fittizio

* **AboutView**

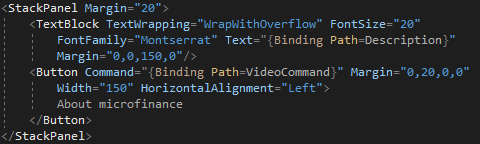
****

Figura La descrizione viene bindata dal ViewModel per lasciare lo XAML più pulito

## Database / MySQL

Impostate le interfacce dell’applicazione bisogna creare il database per il salvataggio dei dati relativi ai creditori.

Seguendo lo schema *barcoinschema.png* risulta molto facile la trascrittura in codice SQL per tabelle e relazioni, il file *barcoin.sql* se eseguito creerà una copia identica del database usato in sviluppo locale.

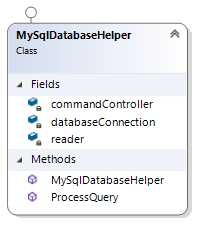
## Logica / C#

Preparata la banca dati MySQL ed avendo completato la prima fase manca ora la parte finale che metta in connessione tutti i moduli, adottando il modello MVVM (Model, View, ViewModel) ed applicando alcune altre convenzioni si ottengono i seguenti namespace:

**Helper:** Namespace contenente le risorse per i servizi.

**MVVM:** Namespace contenente tutti i file alla base della struttura MVVM.

**MySqlDatabaseHelper:** Classe per l’aiuto alla connessione della banca dati MySQL ed esecuzione delle relative query.



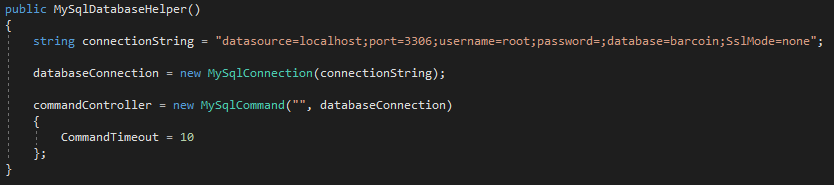


Figura Costruttore della classe per stabilire una connessione con il database.

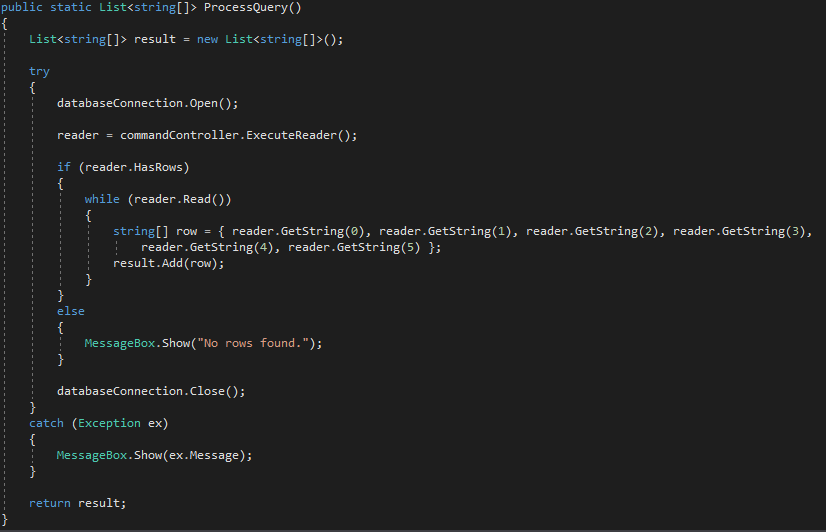
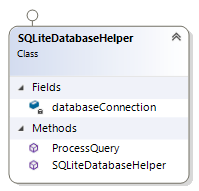
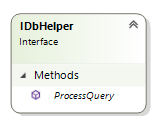
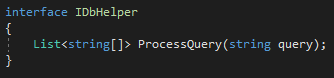


Figura Metodo per eseguire la query corrente e ritornarne il risultato

**SQLiteDatabaseHelper:** Copia logica di MySqlDatabaseHelper ma mirata alla connessione con una banca dati SQLite.

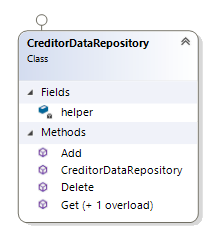


**IDbHelper:** Interfaccia implementata da entrambe le classi di aiuto.



**Service:** Namespace contenente i servizi dell’applicazione.

**CreditorDataRepository:** Classe per l’esecuzione di interrogazioni alla banca dati relative ai creditori, ritorna le informazioni in forma di oggetti.



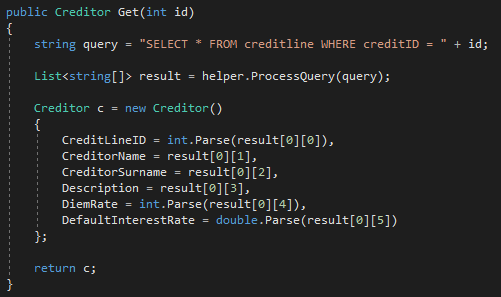


Figura Metodo per ritornare le informazioni di un singolo creditore

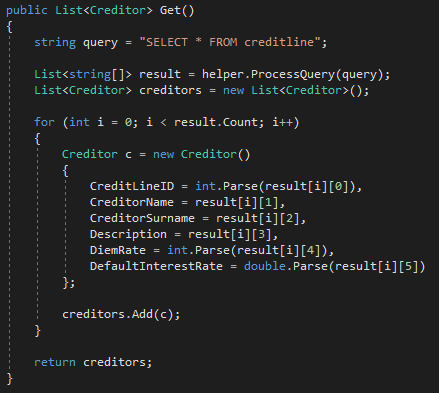


Figura Metodo per ritornare le informazioni di tutti i creditori

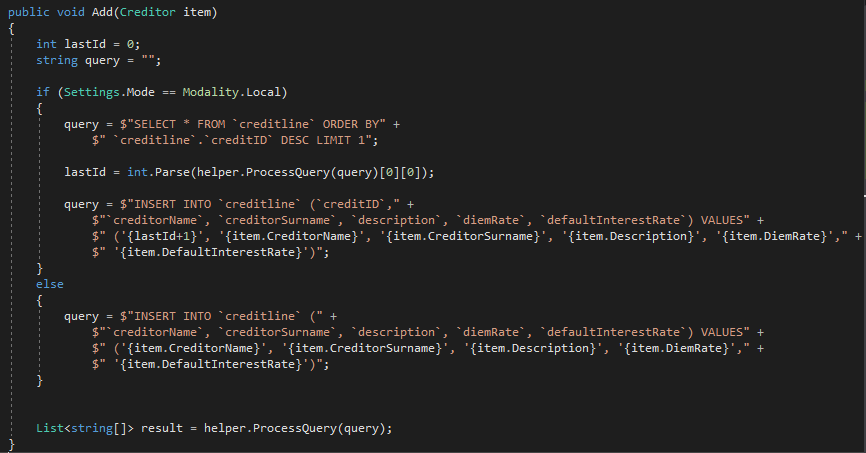
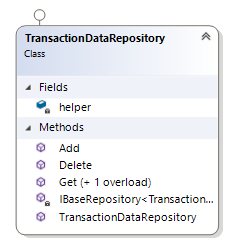


Figura Metodo per inserire un nuovo creditore, nel caso che l’utente stia utilizzando una banca dati SQLite andremo ad aggiungere l’id utente manualmente.

**TransactionDataRepository:** Classe per l’esecuzione di interrogazioni alla banca dati relative alle transazioni, ritorna le informazioni in forma di oggetti.



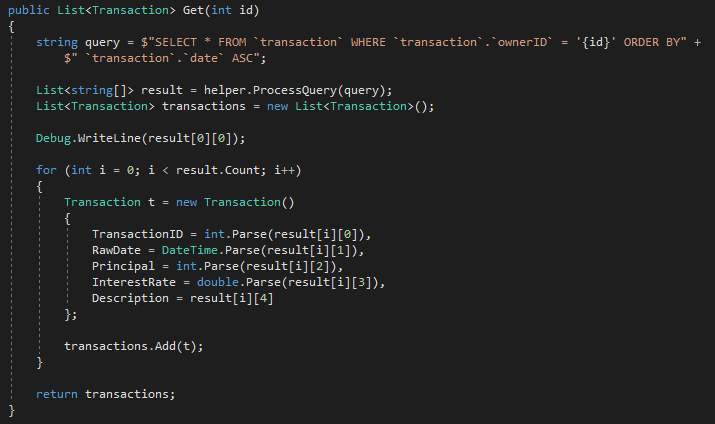


Figura Metodo per ottenere tutte le transazioni di un creditore

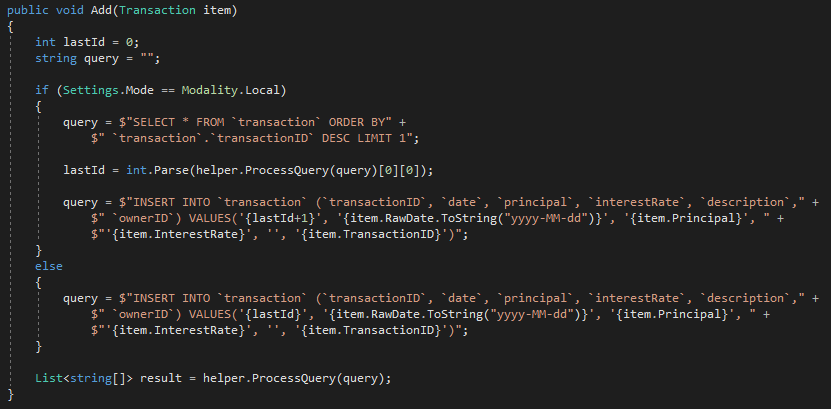
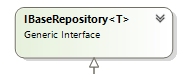
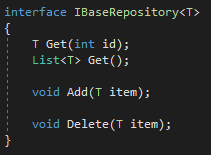


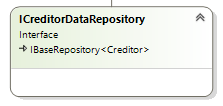
Figura Metodo per aggiungere una transazione ad un creditore, nel caso che l’utente stia utilizzando una banca dati SQLite andremo ad aggiungere l’id della transazione manualmente.

**IBaseRepository:** Interfaccia base per definire una repository di dati qualunque.

****

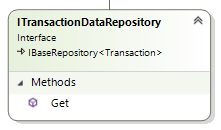
**ICreditorDataRepository:** Interfaccia specifica da implementare per dati dei

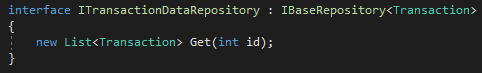
creditori.

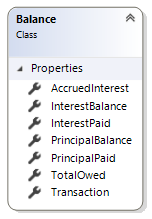




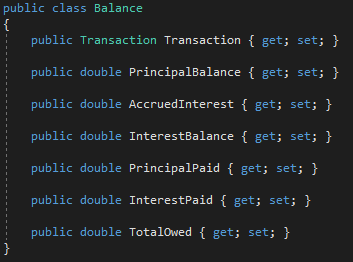
**ITransactionDataRepository:** Interfaccia specifica da implementare per dati

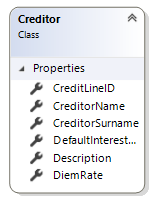
delle transazioni.



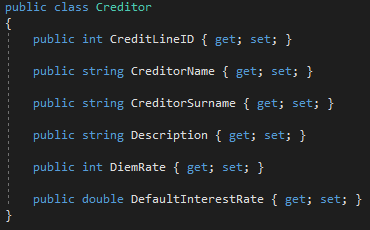
**Model:** Namespace dei modelli di dati.

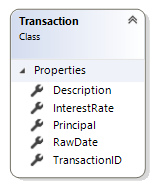
**Balance:** Classe per il modello dati di un bilancio (creditore + transazione).



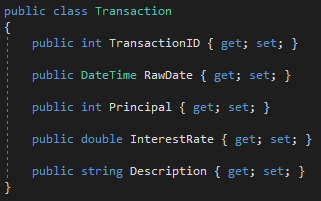


**Creditor:** Classe per il modello dati di un creditore.



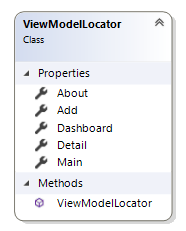


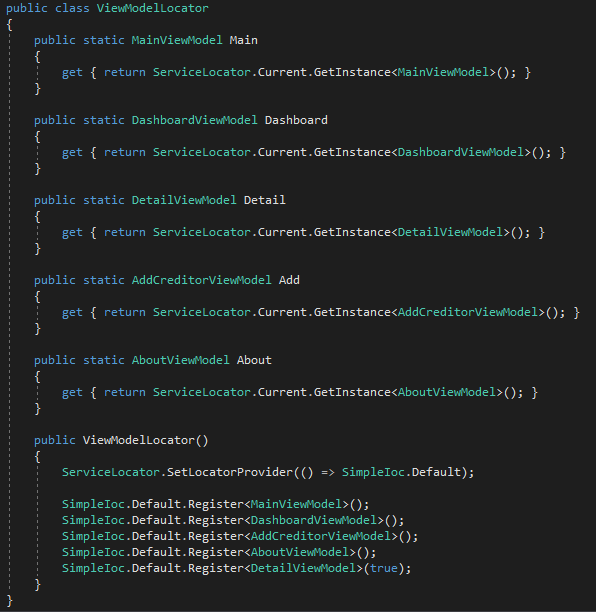
**Transaction:** Classe per il modello dati di una transazione.

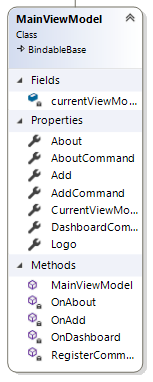


**ViewModel:** Namespace per i controllori grafico/logici (View <> Model).

**ViewModelLocator:** Controllore per la gestione del ViewModel corrente.







**MainViewModel:** Controllore per l’interfaccia MainView.

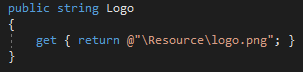


Figura Proprietà per ottenere il link al logo di Barcoin

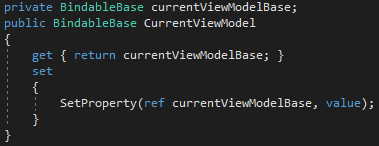


Figura Proprietà per assegnare il controllore attuale a livello grafico



Figura Comando per impostare il controllore Dashboard

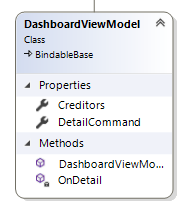
**DashboardViewModel:** Controllore per l’interfaccia DashboardView.



Figura Comando per impostare il controllore Detail



Figura Utilizzo del servizio CreditorDataRepository per ottenere tutti i creditori

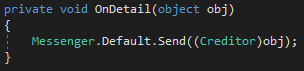
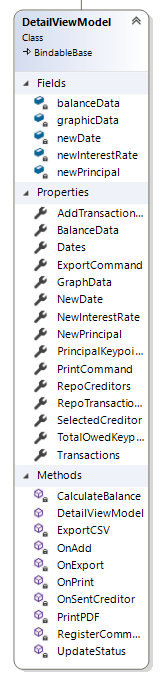


Figura Metodo per inviare il creditore scelto dall'utente nell'interfaccia grafica al controllore Detail

****

**DetailViewModel:** Controllore per l’interfaccia DetailView



Figura Liste contenenti i dati per il grafico del bilancio



Figura Proprietà per il creditore selezionato



Figura Metodo per ricevere l'oggetto Creditor inviato dal controllore Dashboard



Figura Utilizzo del servizio CreditorDataRepository per ottenere tutte le transazioni del creditore attuale

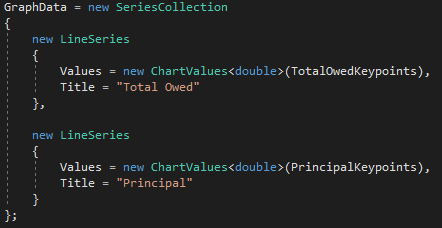


Figura Definizione delle linee con i loro valori e titoli, da inserire nel Live Chart



Figura Metodo per i calcoli da eseguire secondo l’algoritmo di credito

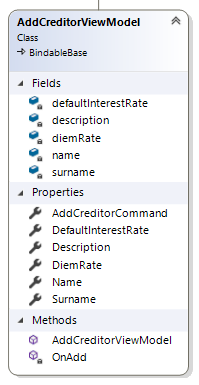
 **AddCreditorViewModel:** Controllore per l’aggiunta di nuovi creditori.



Figura Comando per richiamare il metodo di aggiunta

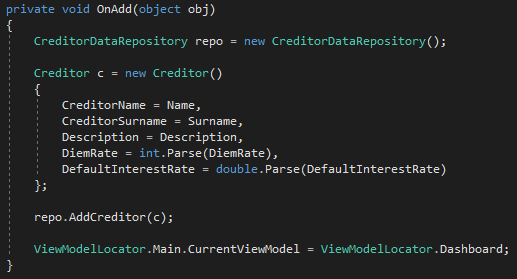


Figura Metodo eseguito quando il comando AddCreditor viene effettuato

## Algoritmo di credito

Fondamento logico principale su cui si basa Barcoin, il calcolo degli interessi e bilanci alla fine di ogni transazione per poter ottenere un sistema corretto e trasparente. Ha necessitato diversi ragionamenti ed analisi di alcuni fogli di calcolo Excel già esistenti, il risultato è il seguente:

B = Bilancio totale

C = Capitale accreditato totale

IA = Bilancio interesse

I = Interesse percentuale di una singola transazione

= Interessi accumulati delle transazioni precedenti

T = Tasso giornaliero stabilito (360/365)

= Data accredito relativa

= Data corrente

*Esempio:*

Per ottenere questo bilancio prendo il capitale totale di 120CHF, gli aggiungo il 6.5% di interesse diviso per il tasso giornaliero stabilito, infine sommo gli interessi calcolati nei bilanci precedenti moltiplicati per l’intervallo di giorni tra la data attuale e la data di accredito relativa.

## Dipendenze

CommonServiceLocation v1.0.0 – Fornisce un livello astratto per l’inserimento di dipendenze logiche.

LiveCharts v0.9.7 – Visualizzazione dei dati semplice, flessibile ed interattiva per .NET.

LiveCharts.WPF v0.9.7 – Estensione di LiveCharts per WPF.

Interactivity.WPF v2.0.20525 – Pacchetto di Microsoft per favorire l’interattività WPF.

Microsoft.Data.SQLite v2.1.0 – Fornisce un DBMS e DB completamente offline in formato di file unico.

PDF Sharp v1.32.3057 – Fornisce una collezione di metodi e funzionalità per creare file in formato PDF.

# Test

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-01  REQ-01 | **Nome:** | Controllo anteprima |
| **Descrizione:** | Visualizzare i creditori registrati in versione anteprima. | | |
| **Prerequisiti:** | La banca dati deve contenere almeno un creditore. | | |
| **Procedura:** | 1. Aprire l’applicazione ed aspettare per lo splash screen di finire | | |
| **Risultati attesi:** | Una serie di bottoni verranno visualizzati quanti sono i creditori registrati nella banca dati. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-02  REQ-02 | **Nome:** | Controllo dettaglio |
| **Descrizione:** | Visualizzare i creditori registrati in versione dettaglio. | | |
| **Prerequisiti:** | TC-01. | | |
| **Procedura:** | 1. Premere su uno dei bottoni presenti nell’interfaccia di anteprima | | |
| **Risultati attesi:** | L’utente viene mandato ad una nuova schermata con due dati testuali, un form per inserire nuove transazioni, una tabella che elenca tutte le transazioni effettuate ed un grafico del bilancio monetario.  Ogni sessanta secondi deve essere effettuato un aggiornamento automatico dei dati mostrati. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-03  REQ-04 | **Nome:** | Controllo inserimento creditore |
| **Descrizione:** | Inserimento nella banca dati di un nuovo creditore. | | |
| **Prerequisiti:** | Nessuno. | | |
| **Procedura:** | 1. Premere il bottone in alto a destra con l’immagine di un “+” 2. Compilare tutti i campi presenti nel form con i dati appartenenti al nuovo creditore. 3. Premere il bottone “Create” | | |
| **Risultati attesi:** | L’utente viene riportato indietro alla schermata in anteprima con un nuovo bottone per l’utente creato. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-04  REQ-05 | **Nome:** | Controllo inserimento transazione |
| **Descrizione:** | Inserimento nella banca dati di una nuova transazione. | | |
| **Prerequisiti:** | TC-02. | | |
| **Procedura:** | 1. Compilare il form nell’interfaccia di dettaglio con i dati della nuova transazione. 2. Premere su “Add Transaction” | | |
| **Risultati attesi:** | La tabella delle transazioni e il grafico dei bilanci vengono aggiornati per comprendere il nuovo record, l’utente avrà una nuova riga di tabella ed una coppia di colonne in più. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-05  REQ-06 | **Nome:** | Controllo esportazioni PDF e CSV |
| **Descrizione:** | Esportare i vari dati in formato PDF e CSV. | | |
| **Prerequisiti:** | TC-02. | | |
| **Procedura:** | 1. Premere il tasto “e”. 2. Premere il tasto “p”. | | |
| **Risultati attesi:** | Andando nella cartella di compilazione del progetto l’utente troverà le cartelle “ExportedTransactions” e “PDFTransactions”, al loro interno ci saranno gli omonimi file esportati con l’identificativo di *NomeCognome*. | | |

## Risultati test

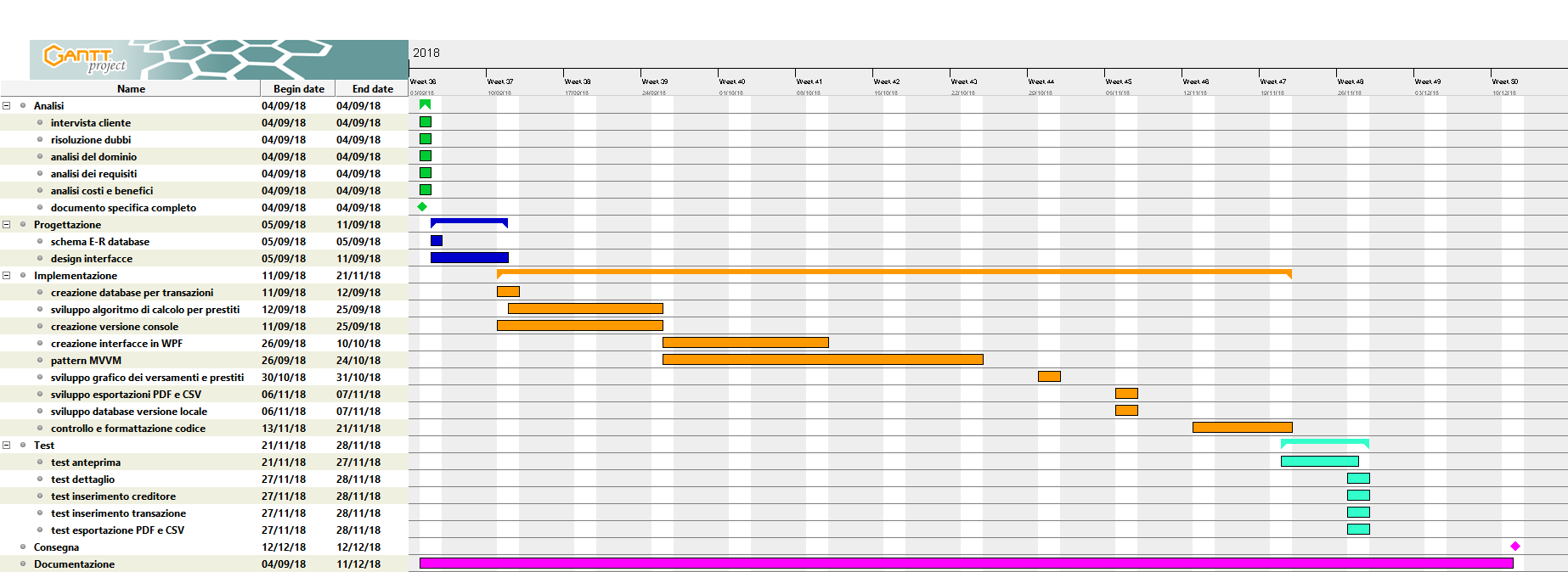
* TC-01 ✓
* TC-02 ✓
* TC-03 ✓
* TC-04 ✓
* TC-05 ✓

## Mancanze/limitazioni conosciute

Come spiegato in modo abbastanza dettagliato all’inizio del documento, Barcoin non si presenta neanche lontanamente come un software per livelli aziendali o in grande volume, presenta infatti gli strumenti e funzionalità limitate per un campione di persone ristretto.

L’utilizzo del linguaggio C# porta inoltre delle limitazioni a livello di software e hardware, a differenza dei suoi simili C e C++ è ad alto livello di astrazione, con una compilazione più pesante e meno performante.

# Consuntivo



# Conclusioni

## Sviluppi futuri

Come sviluppi possibili da implementare ci sarebbe in primo punto la versione web, disponibile a tutti aventi una connessione internet e in secondo punto la creazione di una valuta proprietaria per sorpassare i problemi di conversione, inflazione e truffa.

## Considerazioni personali

Ho trovato personalmente il progetto divertente e particolarmente interessante verso le mie conoscenze economiche, con il professore è già stata discussa la possibilità di legare questo progetto con uno degli sviluppi futuri e renderlo basato su una valuta virtuale.

L’obiettivo sarebbe quello di creare una struttura basata su tecnologia blockchain per mantenere e controllare l’autenticità di questa valuta usata in Barcoin.

# Bibliografia

Le risorse utilizzate come supporto e fondamenta per lo sviluppo di Barcoin, alcune di queste sono propri componenti software mentre altre sono testi, documentazioni o media.

## Sitografia

<https://lvcharts.net/>, Sito di documentazione LiveCharts WPF

<https://docs.microsoft.com/en-us/dotnet/framework/wpf/>, Sito di documentazione Microsoft WPF

<https://en.wikipedia.org/wiki/Microfinance>, Wikipedia sulla micro finanza

<https://www.vertex42.com/Calcators/line-of-credit-tracker.html>, Base algoritmica di Barcoin

<https://dev.mysql.com/doc/>, Sito di documentazione MySQL

<https://www.freepik.com>, Creatore dell’immagine utilizzata come logo

<https://www.flaticon.com/authors/eleonor-wang>, Creatore dell’immagine rappresentante il punto di domanda

<https://www.flaticon.com/authors/dave-gandy>, Creatore dell’immagine rappresentante il segno di addizione.

<http://www3.weforum.org/docs/WEF_Realizing_Potential_Blockchain.pdf>, Blockchain documento 1

<https://en.wikipedia.org/wiki/Blockchain>, Blockchain documento 2

<https://programmingblockchain.gitbook.io/programmingblockchain/>, Blockchain documento 3

<https://blockgeeks.com/guides/what-is-blockchain-technology/>, Blockchain documento 4

[https://cryptodigestnews.com/](https://cryptodigestnews.com/blockchain-consensus-what-you-should-know-95e2d04ed79a) , Blockchain forum

# Allegati

Allegato A: *I4\_Diari\_Barcoin.docx*

# Glossario

|  |  |
| --- | --- |
| **Parola** | **Significato** |
| WPF | Windows Presentation Foundation, modello di Microsoft per la costruzione di applicativi con interfaccia utente basata sul linguaggio XAML. |
| MVVM | Model View ViewModel, gruppo di regole e standard per definire una struttura dati da seguire negli sviluppi di applicativi. |
| Micro-Management | Stile di gestione in cui un manager osserva e / o controlla da vicino il lavoro dei suoi dipendenti o clienti. |
| Diem Rate | Tasso giornaliero prestabilito all’avvio di una linea di credito che indica quanti giorni effettivi vengono considerati dalla banca all’interno di un anno. |
| Unbanked | Che non ha la possibilità di pagare, non solvibile. |
| CSV | Comma-separated values, formato di dati separati da un carattere, solitamente dalla virgola. |
| PDF | Portable Document Format, formato di dati molto simile ad un documento Word ma senza la possibilità di modifica. |