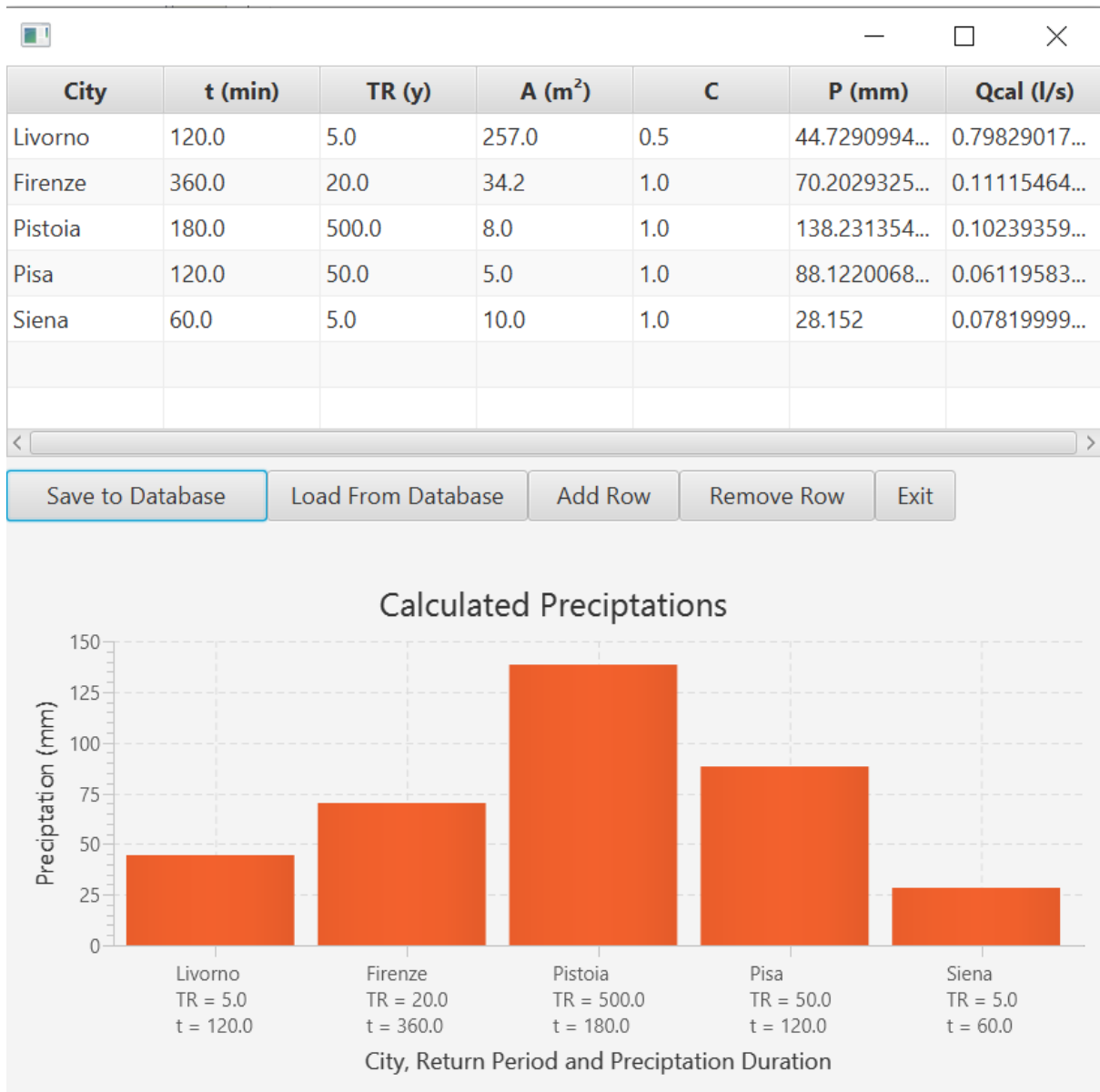


Runoff Calculator – Documento di Progetto

Fernando Ferreira Stroppa

Vista statica:



Vista Dinamica:

1. La tabella dei *Calculated precipitations* visualizza: *City, t, TR, A, Qmax, P* e *Q*, ordinati per ordini di inserimento.
2. L'utente può inserire una nuova riga, eliminare una riga esistente oppure modificare i valori di *City, t, TR, A, C* delle righe esistenti.
3. L'utente prime *Save to Database* per salvare la *tabella* in una base di dati, *Load to Database* per caricare tutti i valori salvati d'una base di dati, *Add row* per aggiungere una nuova riga, *Remove row* per eliminare la riga selezionata ed *Exit* per terminare l'applicazione.
4. Il calcolo della precipitazione è realizzato con i parametri *City, TR, t* e con i dati e formule ottenuti da <https://www.sir.toscana.it/lsp-2012>.
5. Il calcolo del Runoff è realizzato con il metodo razionale con i parametri *P, A* e *C*.

File Locale in XML:

Il sistema, alla premuta del pulsante *Save to Database*, salva i seguenti dati in un file XML:

1. *City, t, TR, A, C*
2. Calculated Precipitation e calculated Runoff

La Figura 1 è un esempio di un file XML e la figura 2 mostra il file di validazione XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<allresults>
  <Toscana>
    <results>
      <city>Livorno</city>
      <precipitationTime>120.0</precipitationTime>
      <returnPeriod>5.0</returnPeriod>
      <area>257.0</area>
      <runoffCoefficientC>0.5</runoffCoefficientC>
      <precipitation>44.729099464210364</precipitation>
      <calculatedRunoff>0.7982901779376432</calculatedRunoff>
    </results>
    <results>
      <city>Firenze</city>
      <precipitationTime>360.0</precipitationTime>
      <returnPeriod>20.0</returnPeriod>
      <area>34.2</area>
      <runoffCoefficientC>1.0</runoffCoefficientC>
      <precipitation>70.20293255583893</precipitation>
      <calculatedRunoff>0.11115464321341163</calculatedRunoff>
    </results>
    <results>
      <city>Pistoia</city>
      <precipitationTime>180.0</precipitationTime>
```

Figura 1 - Esempio del file XML

```

<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001
  <xs:element name="allresults">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Toscana">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="results" maxOccurs="unbounded" minOccurs="0">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element type="xs:string" name="city"/>
                    <xs:element type="xs:float" name="precipitationTime"/>
                    <xs:element type="xs:float" name="returnPeriod"/>
                    <xs:element type="xs:float" name="area"/>
                    <xs:element type="xs:float" name="runoffCoefficientC"/>
                    <xs:element type="xs:float" name="precipitation"/>
                    <xs:element type="xs:float" name="calculatedRunoff"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

```

Figura 2 - File XSD

File del Log di Eventi in XML:

Il sistema, quando qualsiasi pulsante è premuto, invia al XML i seguenti dati:

1. Nome dell'applicazione (*Runoff Calculator*)
2. Event (*Pulsante Premuto*)
3. Data e ora (yyyy-MM-dd HH:mm:ss)

La Figura 3 è un esempio del file XML generato. La classe *XMLEvents* ha il metodo *sendToRemoteServer()*, il cui invia gli eventi a un server remoto con *socket* e può essere facilmente implementato se necessario.

```

<date>09-01-2021 15:36:06</date>
</Events>
<Events>
  <nameOfApplication>Runoff Calculator</nameOfApplication>
  <eventTag>SaveToDataBase</eventTag>
  <date>09-01-2021 15:36:48</date>
</Events>
<Events>
  <nameOfApplication>Runoff Calculator</nameOfApplication>
  <eventTag>exit</eventTag>
  <date>09-01-2021 15:36:50</date>
</Events>
<Events>
  <nameOfApplication>Runoff Calculator</nameOfApplication>
  <eventTag>removeRow</eventTag>
  <date>09-01-2021 15:36:55</date>
</Events>
<Events>
  <nameOfApplication>Runoff Calculator</nameOfApplication>
  <eventTag>removeRow</eventTag>
  <date>09-01-2021 15:36:57</date>
</Events>
<Events>
  <nameOfApplication>Runoff Calculator</nameOfApplication>
  <eventTag>LoadFromDataBase</eventTag>
  <date>09-01-2021 15:36:58</date>
</Events>
<Events>
  <nameOfApplication>Runoff Calculator</nameOfApplication>
  <eventTag>addRow</eventTag>

```

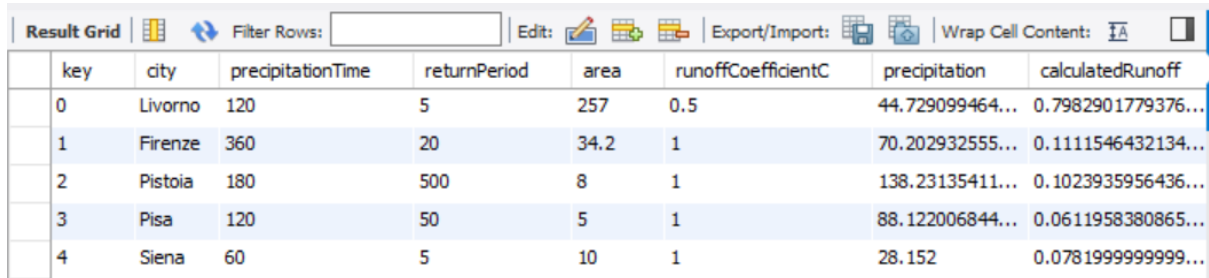
Figura 3 - Esempio del contenuto del XML di Eventi

Archivio in SQL:

Il sistema archivia i seguenti dati:

1. I dati inseriti nella interfaccia (*City, Tr, t, Area, C*)
2. I dati calcolati dell'applicazione (*Runoff, Preciptation*)

La Figura 4 è un esempio della basi di dati in SQL.



key	city	precipitationTime	returnPeriod	area	runoffCoefficientC	precipitation	calculatedRunoff
0	Livorno	120	5	257	0.5	44.729099464...	0.7982901779376...
1	Firenze	360	20	34.2	1	70.202932555...	0.1111546432134...
2	Pistoia	180	500	8	1	138.23135411...	0.1023935956436...
3	Pisa	120	50	5	1	88.122006844...	0.0611958380865...
4	Siena	60	5	10	1	28.152	0.0781999999999...

Figura 4 - Esempio dei valori in un file SQL

Cache Locale Degli Input:

Il sistema salva i seguenti dati su un file binario alla chiusura (sia chiusura normale che da un'eccezione):

1. I dati inseriti nella interfaccia (*City, Tr, t, Area, C*)
2. La riga selezionata, i campi calcolati e non ancora inseriti nel database e tutte le modificazioni prima del salvataggio.

Quando si apre l'applicazione, il file binario deve essere caricato.

Manual Utente

The workspace:

Editable					Not Editable	
City	t (min)	TR (y)	A (m ²)	C	P (mm)	Qcal (l/s)
Livorno	122.0	5.0	257.0	0.5	44.8856546...	0.78795172...
Firenze	360.0	20.0	34.2	1.0	70.2029325...	0.11115464...
Pistoia	180.0	500.0	8.0	1.0	138.231354...	0.10239359...
Pisa	120.0	50.0	5.0	1.0	88.1220068...	0.06119583...
Siena	60.0	5.0	10.0	1.0	28.152	0.07819999...

La tabella è divisa in campi editabili e non editabili. Per editare un campo, viene fatto un doppio click in qualsiasi cellula editabile, viene inserito il valore (che dovrebbe essere limitato ai valori indicati nella tabella 1) e viene premuto Enter nella tastiera. I campi *P* e *Qcal* saranno calcolati automaticamente dopo la premuta del Enter. Il grafico sarà aggiornato automaticamente dopo la premuta del Enter.

Tabella 1 - Valori ammissibile

Cellula	Valori Ammissibile
City	Arezzo, Firenze, Grosseto, Livorno, Lucca, Massa-Carrara, Pisa, Pistoia, Prato, Siena
t (min)	$15 < t \leq 720$
Tr (y)	2; 5; 10; 20; 30; 100; 150; 200; 500
A (m ²)	$0 < A \leq 5000$
C	$0 < C \leq 1.0$

Pulsanti:

Save to Database: Salva i valori della tabella nel file SQL e XML.

Load from Database: Carica i valori del file SQL nella tabella.

Add row: Viene inserita una riga nella tabella con i valori default (Arezzo, 60, 2, 10, 1).

Delete row: Elimina la riga selezionata.

Exit: Esci della applicazione.

Diagramma UML

La Figura 5 è un diagramma UML con tutte le classi dell'applicativo. Il diagramma UML della Figura 5 mostra solamente la struttura di alto livello. I metodi, variabili ed imports non rilevanti alla struttura di alto livello sono stati omise.

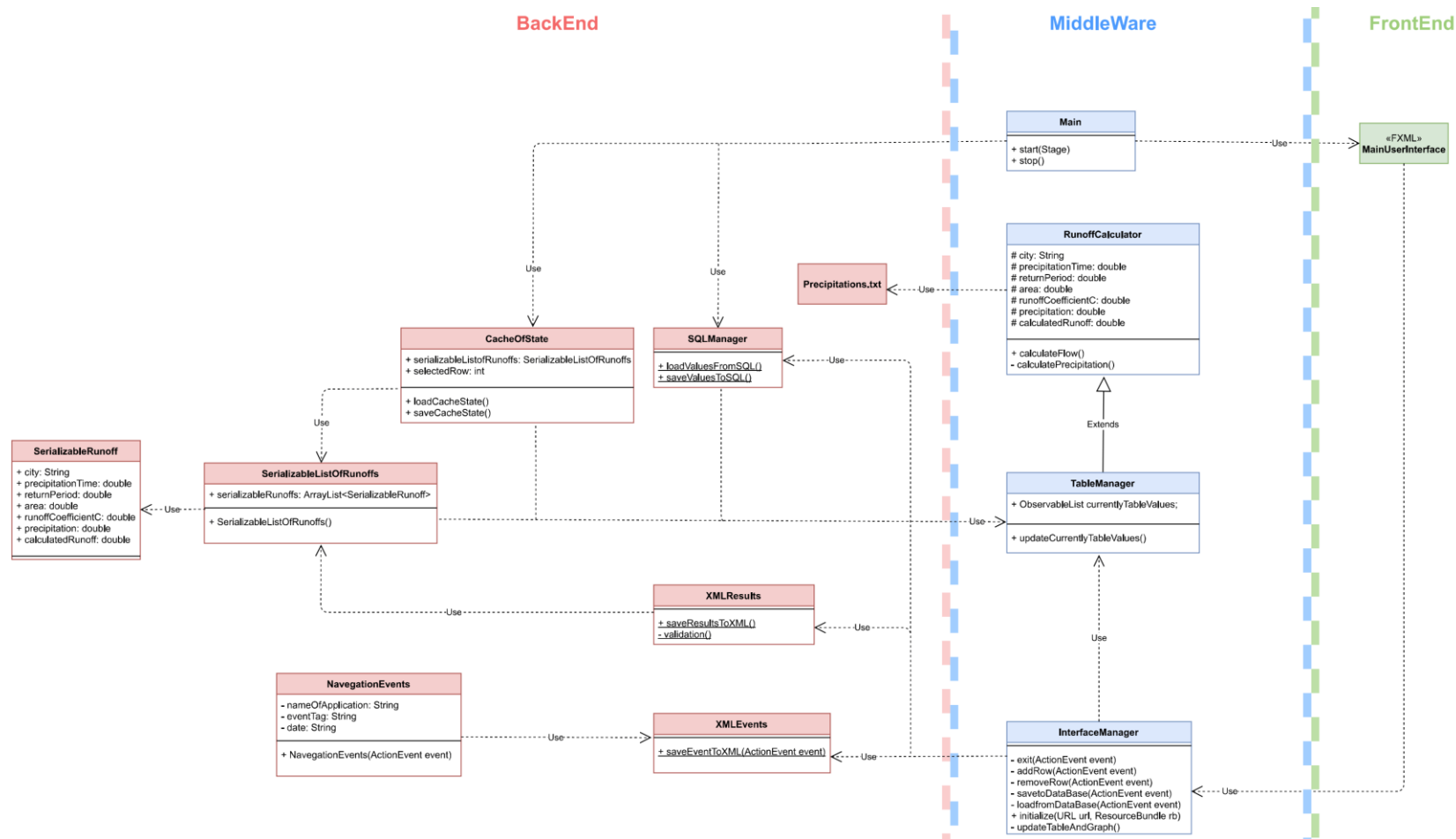


Figura 5 - High Level UML

Responsabilità delle Classi

CacheOfState: Salva lo stato dell'applicativo nel momento della chiusura in un file binario e lo carica all'inizio dell'applicazione.

NavegationEvents: contiene le variabili serializzabili per la classe *XMLEvents*.

XMLEvents: salva gli eventi in un file XML. Può essere modificata per inviare gli eventi a un server remoto.

SQLManager: Salva e carica i valori inseriti e calcolati su SQL.

SerializableListOfRunoffs: Lista serializzabile delle istanze della classe dei *SerializableRunoff*.

SerializableRunoff: Lista serializzabile delle variabili della classe dei *RunoffCalculator*

XMLResults: Salva e valida i valori inseriti e calcolati su XML.

MainUserInterface.fxml: FXML con l'interfaccia grafica dell'applicativo.

InterfaceManager: Fa la relazione tra pulsanti e FXML. Esecuta i metodi delle classi *TableManager*, *XMLEvents*, *XMLResults* ed *SQLManager*. Attualizza il grafico.

Main: Inizia l'applicativo. Chiama il metodo di caricamento dei valori del SQL e Cache. Chiama il metodo di salvataggio della Cache nella chiusura dell'applicativo.

RunoffCalculator: Contiene le *business rules* di calcolo di Runoff e precipitazioni.

TableManager: Responsabile per la relazioni dei dati della tabella con il *Backend* e con la classe *InterfaceManager* (per attualizzare la tabella nella interfaccia grafica).