

Università degli Studi di Napoli Federico II

**Corso di Laurea in Ingegneria Informatica**

**Corso di Ingegneria del Software**

**Prof. A.R. Fasolino - A.A. 2021 - 22**

***Progetto***

Sistema software per società di

forniture energetiche

Studente: Lorenzo Cappellieri N46005090 l.cappellieri@studenti.unina.it

Studente: Andrea Di Marco N46005154

and.dimarco@studenti.unina.it

Studente: Vincenzo Ciccarelli N46004893

vincenzo.ciccarelli@studenti.unina.it

Versione del *14/06/2022*

**INDICE**

Sistema di vendita (online e al botteghino) di biglietti per il cinema multisala “Rainbow MultiScreen”

1. Specifiche informali 1

2. Analisi e specifica dei requisiti 2

2.1 Analisi nomi-verbi 2

2.2 Revisione dei requisiti 2

2.3 Glossario dei termini 3

2.4 Classificazione dei requisiti 3

2.4.1 Requisiti funzionali 4

2.4.2 Requisiti sui dati 5

2.4.3 Vincoli / Altri requisiti 5

2.5 Modellazione dei casi d’uso 6

2.5.1 Attori e casi d’uso 6

2.5.2 Diagramma dei casi d’uso 2

2.5.3 Scenari 3

2.6 Diagramma delle classi 4

2.7 Diagrammi di sequenza 6

2.8 Verifica della completezza dei requisiti 7

3. Stima dei costi 8

4. Piano di test funzionale 10

5. Progettazione 13

5.1 Diagramma delle classi 13

5.2 Diagrammi di sequenza 13

6. Implementazione 15

7. Testing 16

7.1 Test strutturale 16

7.1.1 Complessità ciclomatica 16

7.2 Test funzionale 17

# Specifiche informali

*Si vuole realizzare un sistema software per consentire ad una società di forniture energetiche di gestire le forniture erogate ai clienti che hanno sottoscritto un contratto di fornitura.*

La società gestisce le forniture dei propri clienti ed ogni fornitura è assegnata ad una unità abitativa del cliente. Ogni cliente può possedere più unità abitative ed ogni unità abitativa può avere più forniture ma di diverso tipo. Ogni unità abitativa è caratterizzata da un indirizzo. L'indirizzo è formato da via, numero civico, numero interno, città e codice di avviamento postale. Ogni unità abitativa corrisponde ad un solo cliente. Il cliente è caratterizzato dal proprio nome, cognome, codice fiscale, e da un codice cliente univoco (numerico). Ogni fornitura è caratterizzata da un identificativo univoco, il tipo della fornitura, una tariffa unitaria (ossia il costo in euro per unità di fornitura), ed il tipo di Unità di fornitura. Possibili tipi di fornitura sono energia elettrica, gas, acqua e le rispettive unità di fornitura sono KiloWattOra, Metri Cubi, e Metri Cubi. Il sistema deve permettere ad un impiegato addetto alla registrazione delle forniture di assegnare una fornitura ad un cliente per una specifica abitazione. A tal fine il cliente deve fornire all’impiegato tutti i propri dati (nome, cognome, codice fiscale) e l’indirizzo dell’abitazione e scegliere il tipo di fornitura richiesta. L’impiegato inserisce i dati forniti dal cliente nel sistema, il sistema verificherà la correttezza e completezza di tali dati, e controllerà se il cliente è già registrato nel sistema, attraverso il suo codice fiscale. Se il cliente non è ancora registrato, il sistema provvederà a registrare le informazioni del cliente, ad assegnargli un codice cliente univoco, e a registrare i dati dell'unità abitativa fornita. Se il cliente e l’unità abitativa sono già registrati, il sistema controllerà che il cliente non possieda già una fornitura di quel tipo nell’unità abitativa specificata, ed in caso di esito positivo dei controlli creerà una nuova fornitura, assegnandole un identificativo univoco, il tipo e la tariffa unitaria. Per definire la tariffa, il sistema accede ad un proprio archivio interno che contiene tutte le tariffe attualmente vigenti. La fornitura verrà inoltre associata al cliente e alla unità abitativa specificata dal cliente. In caso di qualche controllo con esito negativo, il sistema dovrà restituire un apposito messaggio di errore e chiedere all’impiegato di fornire nuovi dati. In caso di esito positivo, il sistema restituisce l’Identificativo della fornitura creata. Al momento della creazione della nuova fornitura, il sistema creerà anche un contatore associato alla fornitura che riporterà le ultime due letture di consumi rilevati di fornitura (lettura1 e lettura2). Ogni lettura è costituita da un numero decimale (con due cifre decimali) e sono entrambe inizialmente poste a zero. Il cliente potrà successivamente utilizzare una apposita interfaccia per registrarsi al sistema, fornendo uno username (stringa di al più 20 caratteri) ed una password (stringa di esattamente 8 caratteri) univoci. Una volta autenticato, il cliente dovrà fornire mensilmente (entro l’ultimo giorno di ogni mese) la lettura dei consumi delle proprie forniture al sistema. A tal fine, il cliente dovrà specificare il proprio username e password, l’identificativo della fornitura, ed il valore della lettura. Il sistema controllerà la correttezza di username e password e che la fornitura esista e sia associata a quel cliente, dopo di che controllerà che la lettura fornita sia maggiore o uguale alla lettura2 già registrata. In caso di esito positivo dei controlli, il sistema aggiornerà lettura1 con il valore di lettura2 registrato nel contatore ed inoltre aggiornerà lettura2 con il valore fornito dal cliente. Una volta autenticato, il cliente potrà anche stampare l’ultima fattura per una data fornitura o per tutte le sue forniture.

Il sistema deve periodicamente (nel secondo giorno di ogni mese) generare le fatture mensili per ogni fornitura calcolando l’importo da pagare sulla base dei consumi registrati (e non ancora fatturati) e sulla base della tariffa stabilita. Il direttore commerciale può generare report sui consumi annuali di tutte le forniture di una data città, oppure generare statistiche sulle forniture erogate (consumi medi, minimi e massimi in un dato periodo e città). Per consentire un utilizzo agevole del sistema da parte dei suoi utenti, si richiede che il sistema sia dotato di interfacce grafiche user-friendly e che per ciascuna operazione di ricerca cliente o di ricerca fornitura il tempo di risposta sia non superiore a 5 secondi. Si richiede inoltre che sia assicurato l’accesso al sistema da parte esclusiva dei suoi utenti autorizzati. A tal fine, ogni impiegato autorizzato all’uso del sistema possiede le sue credenziali di accesso (username e password) e dovrà autenticarsi prima di eseguire qualunque operazione nel sistema.

Per tale sistema, il candidato realizzi un progetto secondo il facsimile inviato dal docente per e-mail, progetti la base di dati in maniera completa popolandola con dei dati d’esempio, ed implementi in linguaggio Java la funzionalità per l’impiegato di assegnazione di una fornitura ad un cliente.

# Analisi e specifica dei requisiti

## Analisi nomi-verbi

*Si vuole realizzare un sistema software per consentire ad una società di forniture energetiche di gestire le forniture erogate ai clienti che hanno sottoscritto un contratto di fornitura.*

La società gestisce le forniture dei propri clienti ed ogni fornitura è assegnata ad una unità abitativa del cliente. Ogni cliente può possedere più unità abitative ed ogni unità abitativa può avere più forniture ma di diverso tipo. Ogni unità abitativa è caratterizzata da un indirizzo. L'indirizzo è formato da via, numero civico, numero interno, città e codice di avviamento postale. Ogni unità abitativa corrisponde ad un solo cliente. Il cliente è caratterizzato dal proprio nome, cognome, codice fiscale, e da un codice cliente univoco (numerico).

Ogni fornitura è caratterizzata da un identificativo univoco, il tipo della fornitura, una tariffa unitaria (ossia il costo in euro per unità di fornitura), ed il tipo di Unità di fornitura. Possibili tipi di fornitura sono energia elettrica, gas, acqua e le rispettive unità di fornitura sono KiloWattOra, Metri Cubi, e Metri Cubi.

Il sistema deve permettere ad un impiegato addetto alla registrazione delle forniture di assegnare una fornitura ad un cliente per una specifica abitazione. A tal fine il cliente deve fornire all’impiegato tutti i propri dati (nome, cognome, codice fiscale) e l’indirizzo dell’abitazione e scegliere il tipo di fornitura richiesta. L’impiegato inserisce i dati forniti dal cliente nel sistema, il sistema verificherà la correttezza e completezza di tali dati, e controllerà se il cliente è già registrato nel sistema, attraverso il suo codice fiscale. Se il cliente non è ancora registrato, il sistema provvederà a registrare le informazioni del cliente, ad assegnargli un codice cliente univoco, e a registrare i dati dell'unità abitativa fornita. Se il cliente e l’unità abitativa sono già registrati, il sistema controllerà che il cliente non possieda già una fornitura di quel tipo nell’unità abitativa specificata, ed in caso di esito positivo dei controlli creerà una nuova fornitura, assegnandole un identificativo univoco, il tipo e la tariffa unitaria. Per definire la tariffa, il sistema accede ad un proprio archivio interno che contiene tutte le tariffe attualmente vigenti. La fornitura verrà inoltre associata al cliente e alla unità abitativa specificata dal cliente. In caso di qualche controllo con esito negativo, il sistema dovrà restituire un apposito messaggio di errore e chiedere all’impiegato di fornire nuovi dati. In caso di esito positivo, il sistema restituisce l’Identificativo della fornitura creata.

Al momento della creazione della nuova fornitura, il sistema creerà anche un contatore associato alla fornitura che riporterà le ultime due letture di consumi rilevati di fornitura (lettura1 e lettura2). Ogni lettura è costituita da un numero decimale (con due cifre decimali) e sono entrambe inizialmente poste a zero.

Il cliente (registrato) potrà successivamente utilizzare una apposita interfaccia per registrarsi al sistema, fornendo uno username (stringa di al più 20 caratteri) ed una password (stringa di esattamente 8 caratteri) univoci.

Una volta autenticato, il cliente dovrà fornire mensilmente (entro l’ultimo giorno di ogni mese) la lettura dei consumi delle proprie forniture al sistema. A tal fine, il cliente dovrà specificare il proprio username e password, l’identificativo della fornitura, ed il valore della lettura. Il sistema controllerà la correttezza di username e password e che la fornitura esista e sia associata a quel cliente, dopo di che controllerà che la lettura fornita sia maggiore o uguale alla lettura2 già registrata. In caso di esito positivo dei controlli, il sistema aggiornerà lettura1 con il valore di lettura2 registrato nel contatore ed inoltre aggiornerà lettura2 con il valore fornito dal cliente.

Una volta autenticato, il cliente potrà anche stampare l’ultima fattura per una data fornitura o per tutte le sue forniture.

Il sistema deve periodicamente (nel secondo giorno di ogni mese) generare le fatture mensili per ogni fornitura calcolando l’importo da pagare sulla base dei consumi registrati (e non ancora fatturati) e sulla base della tariffa stabilita.

Il direttore commerciale può generare report sui consumi annuali di tutte le forniture di una data città, oppure generare statistiche sulle forniture erogate (consumi medi, minimi e massimi in un dato periodo e città).

Per consentire un utilizzo agevole del sistema da parte dei suoi utenti, si richiede che il sistema sia dotato di interfacce grafiche user-friendly e che per ciascuna operazione di ricerca cliente o di ricerca fornitura il tempo di risposta sia non superiore a 5 secondi. Si richiede inoltre che sia assicurato l’accesso al sistema da parte esclusiva dei suoi utenti autorizzati. A tal fine, ogni impiegato autorizzato all’uso del sistema possiede le sue credenziali di accesso (username e password) e dovrà autenticarsi prima di eseguire qualunque operazione nel sistema.

*LEGENDA:  
Classe  
Attributo  
Funzionalità  
Attore*

*Classe-Attore*

## Revisione dei requisiti

1. *Ogni cliente può possedere più unità abitative.*
2. *Ogni unità abitativa corrisponde un solo cliente.*
3. *Ogni unità abitativa può avere più forniture, ma di diverso tipo.*
4. *Ogni unità abitativa è caratterizzata da un indirizzo.*
5. *L’indirizzo è formato da: via, numero civico, numero interno, città e CAP.*
6. *Il cliente è caratterizzato da: nome, cognome, codice fiscale e da un codice cliente univoco.*
7. *Ogni fornitura è caratterizzata da un identificativo univoco, il tipo della fornitura, una tariffa unitaria ed il tipo di unità di fornitura.*
8. *Il sistema deve offrire all’impiegato una funzionalità per assegnare una fornitura ad un cliente per una specifica abitazione.*
9. *Il sistema deve verificare la correttezza e completezza dei dati inseriti dall’impiegato, e in caso di esito negativo dovrà restituire un apposito messaggio di errore.*
10. *Il sistema deve verificare la correttezza e completezza dei dati inseriti dall’impiegato, e in caso di esito positivo dovrà restituire l’identificativo della fornitura creata.*
11. *In fase di assegnazione della fornitura il sistema controllerà che il cliente sia registrato, nel caso in cui non lo fosse provvederà a registrare le informazioni del cliente e dell’unità abitativa.*
12. *In fase di assegnazione della fornitura il sistema controllerà che il cliente sia registrato, nel caso in cui lo fosse controllerà che il cliente non possieda già una fornitura di quel tipo nell’unità abitativa specificata.*
13. *In fase di assegnazione della fornitura il sistema controllerà che il cliente non possieda già una fornitura di quel tipo nell’unità abitativa specificata, creerà una nuova fornitura.*
14. *Il sistema assegna una tariffa rifacendosi ad un apposito archivio, in caso di esito positivo il sistema restituisce l’identificativo della fornitura creata.*
15. *Il sistema assegna la fornitura al cliente e all’unità abitativa associata.*
16. *Il sistema restituisce un messaggio di errore in caso di errori pregressi.*
17. *Il sistema crea un contatore associato alla fornitura che riporta le ultime due letture dei consumi.*
18. *Un contatore è formato da due letture costituita da un numero decimale.*
19. *Il sistema deve offrire al cliente una funzionalità per registrarsi.*
20. *In fase di registrazione il sistema deve verificare la validità di username e password.*
21. *L’ username deve essere una stringa di al più 20 caratteri e la password deve essere una stringa di esattamente 8 caratteri entrambi univoci.*
22. *Il sistema controlla la correttezza della lettura inserita.*
23. *Il sistema controlla che la lettura inserita sia maggiore dell’ultima lettura fornita.*
24. *Il sistema aggiorna l’ultima lettura in caso di esito positivo ai controlli.*
25. *Il sistema deve automaticamente generare le fatture mensili per ogni fornitura.*
26. *Il sistema deve offrire ai clienti registrati la possibilità di stampare l’ultima fattura per una data fornitura.*
27. *Il sistema deve offrire al direttore commerciale una funzionalità per generare report sui consumi annuali di tutte le forniture di una data città.*
28. *Il sistema deve offrire al direttore commerciale una funzionalità per generare statistiche sulle forniture erogate.*
29. *Il sistema deve assicurare che l’accesso sia effettuato dai soli utenti autorizzati.*

## Glossario dei termini

|  |  |  |
| --- | --- | --- |
| Termine | Descrizione | Sinonimi |
| Fornitura | Un contratto tramite il quale il servizio viene trasferito al cliente. |  |
| Cliente | Un utente registrato che usufruisce dei servizi forniti dalla società |  |
| Unità Abitativa | Uno spazio architettonico attribuito ad un solo cliente che può avere una fornitura di ogni tipo. |  |
| Impiegato | Un dipendente dedito alla registrazione di clienti e assegnazione di forniture. |  |
| Tariffa | Prezzo della fornita non suscettibile alle variazioni di mercato. |  |
| Contatore | Oggetto adibito a contenere le ultime due letture. |  |
| Lettura | Valore stimato del consumo mensile. |  |
| Fattura | Documento fiscale che comprova il servizio erogato e il diritto a riscuoterne il prezzo. |  |
| Direttore Commerciale | Responsabile dell’attuazione delle politiche commerciali dell’azienda. |  |
| Report | Resoconto annuale delle statistiche. |  |

## Classificazione dei requisiti

## Requisiti funzionali

|  |  |  |
| --- | --- | --- |
| ID | Requisito | Origine (n. frase dei requisiti revisionati) |
| RF01 | *Il sistema deve offrire all’impiegato una funzionalità per assegnare una fornitura ad un cliente per una specifica abitazione.* | 8 |
| RF02 | Il sistema deve verificare la correttezza e completezza dei dati inseriti dall’impiegato, e in caso di esito negativo dovrà restituire un apposito messaggio di errore. | 9 |
| RF03 | Il sistema deve verificare la correttezza e completezza dei dati inseriti dall’impiegato, e in caso di esito positivo dovrà restituire l’identificativo della fornitura creata. | 10 |
| RF04 | In fase di assegnazione della fornitura il sistema controllerà che il cliente sia registrato, nel caso in cui non lo fosse provvederà a registrare le informazioni del cliente e dell’unità abitativa. | 11 |
| RF05 | In fase di assegnazione della fornitura il sistema controllerà che il cliente sia registrato, nel caso in cui lo fosse controllerà che il cliente non possieda già una fornitura di quel tipo nell’unità abitativa specificata. | 12 |
| RF06 | In fase di assegnazione della fornitura il sistema controllerà che il cliente non possieda già una fornitura di quel tipo nell’unità abitativa specificata, creerà una nuova fornitura. | 13 |
| RF07 | Il sistema assegna una tariffa rifacendosi ad un apposito archivio, in caso di esito positivo il sistema restituisce l’identificativo della fornitura creata. | 14 |
| RF08 | Il sistema assegna la fornitura al cliente e all’unità abitativa associata. | 15 |
| RF09 | Il sistema restituisce un messaggio di errore in caso di errori pregressi. | 16 |
| RF10 | Il sistema crea un contatore associato alla fornitura che riporta le ultime due letture dei consumi. | 17 |
| RF11 | Il sistema deve offrire al cliente una funzionalità per registrarsi. | 19 |
| RF12 | In fase di registrazione il sistema deve verificare la validità di username e password. | 20 |
| RF13 | Il sistema controlla la correttezza della lettura inserita. | 22 |
| RF14 | Il sistema controlla che la lettura inserita sia maggiore dell’ultima lettura fornita. | 23 |
| RF15 | Il sistema aggiorna l’ultima lettura in caso di esito positivo ai controlli. | 24 |
| RF16 | Il sistema deve automaticamente generare le fatture mensili per ogni fornitura. | 25 |
| RF17 | Il sistema deve offrire ai clienti registrati la possibilità di stampare l’ultima fattura per una data fornitura. | 26 |
| RF18 | Il sistema deve offrire al direttore commerciale una funzionalità per generare report sui consumi annuali di tutte le forniture di una data città. | 27 |
| RF19 | Il sistema deve offrire al direttore commerciale una funzionalità per generare statistiche sulle forniture erogate. | 28 |

## Requisiti sui dati

|  |  |  |
| --- | --- | --- |
| ID | Requisito | Origine (n. frase dei requisiti revisionati) |
| RD01 | *Ogni cliente può possedere più unità abitative.* | 1 |
| RD02 | *Ogni unità abitativa corrisponde un solo cliente.* | 2 |
| RD03 | *Ogni unità abitativa può avere più forniture, ma di diverso tipo.* | 3 |
| RD04 | *Ogni unità abitativa è caratterizzata da un indirizzo.* | 4 |
| RD05 | *L’indirizzo è formato da: via, numero civico, numero interno, città e CAP.* | 5 |
| RD06 | *Il cliente è caratterizzato da: nome, cognome, codice fiscale e da un codice cliente univoco.* | 6 |
| RD07 | *Ogni fornitura è caratterizzata da un identificativo univoco, il tipo della fornitura, una tariffa unitaria ed il tipo di unità di fornitura.* | 7 |
| RD08 | *Un contatore è formato da due letture costituita da un numero decimale.* | 18 |
| RD09 | *L’ username deve essere una stringa di al più 20 caratteri e la password deve essere una stringa di esattamente 8 caratteri entrambi univoci* | 21 |

## Vincoli / Altri requisiti

|  |  |  |
| --- | --- | --- |
| ID | Requisito | Origine (n. frase dei requisiti revisionati) |
| RNF01 | Le richieste dei dati devono avere un tempo di risposta non superiore a 5 secondi per ogni operazione. |  |
| V01 | Il sistema deve assicurare che l’accesso sia effettuato dai soli utenti autorizzati. | 29 |
| V02 | Ogni impiegato dovrà autenticarsi prima di eseguire qualunque operazione di sistema. |  |

## Modellazione dei casi d’uso

## Attori e casi d’uso

***Attori Primari:***

* Impiegato
* Cliente
* Direttore Commerciale

***Casi d’uso:***

* UC1: Registrazione.
* UC2: Visualizza fattura.
* UC3: Inserimento letture mensile.
* UC4: Assegnazione fornitura.
* UC5: Genera fattura.
* UC6: Genera report consumi annuali.
* UC7: Genera statistiche sulle forniture annuali.

***Casi d’ uso di inclusione:***

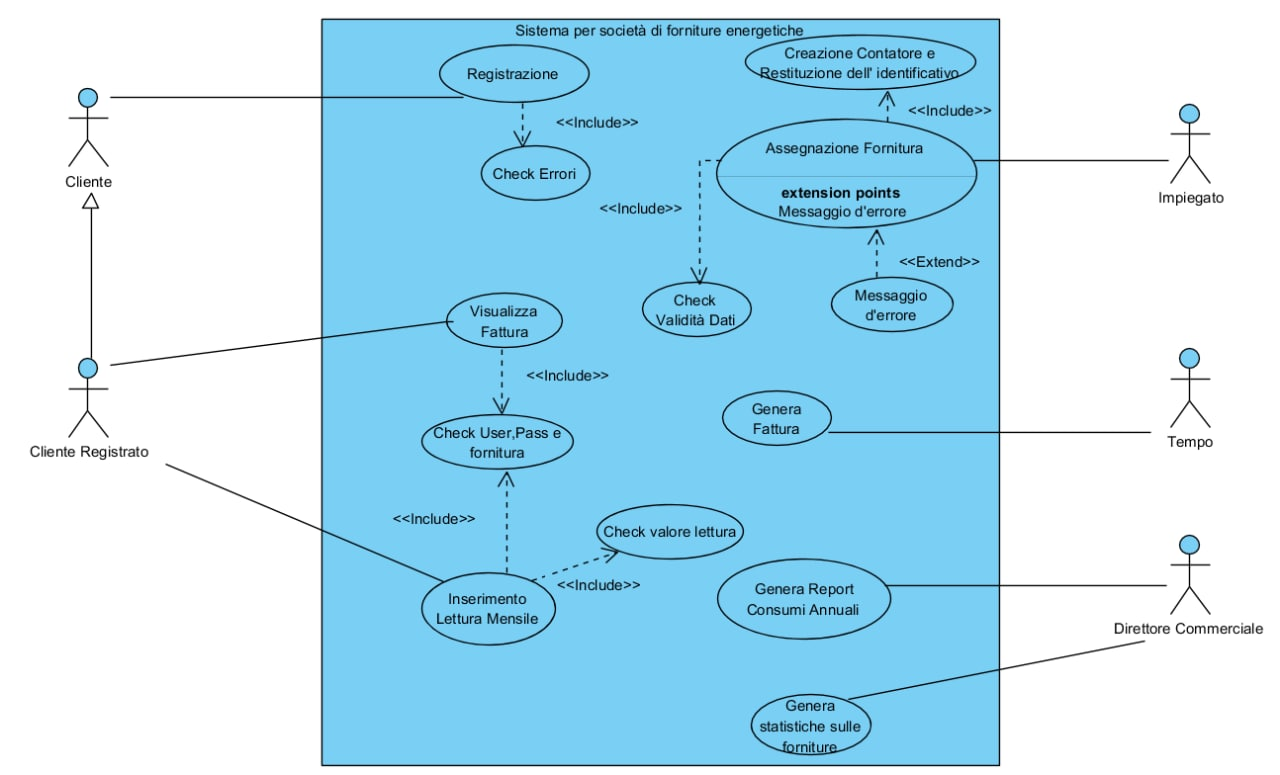
* UC8: Check errori.
* UC9: Check Username, password e fornitura
* UC10: Creazione contatore.
* UC11: Check validità dati.
* UC12: Check valore lettura

***Casi d’ uso di estensione:***

* UC13: Messaggio d’errore.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Caso d’uso | Attori Primari | Attori Secondari | Incl. / Ext. | Requisiti corrispondenti |
| UC1: Registrazione | Cliente | - | Check Errori | RF11, RF12 |
| UC2: Visualizza fattura | Cliente Registrato | - | Check User, password e fornitura | RF17 |
| UC3: Inserimento letture mensile | Cliente Registrato | - | Check User, password e fornitura | RF13, RF14, RF15 |
| UC4: Assegnazione fornitura | Impiegato | - | Messaggio d’errore  Check Validità dati  Creazione contatore e restituzione dell’identificativo | RF01, RF02, RF03, RF03, RF04, RF05, RF06, RF07, RF08, RF09, RF10 |
| UC5: Genera fattura | Tempo | - | - | RF16 |
| UC6: Genera report consumi annuali | Direttore commerciale | - | - | RF18 |
| UC7: Genera statistiche sulle forniture erogate | Direttore commerciale | - | - | RF19 |

## Diagramma dei casi d’uso



## Dsecrizioni casi d’uso

|  |  |
| --- | --- |
| Caso d’uso: | Registrazione |
| Attore primario | Cliente. |
| Attore secondario | - |
| Descrizione | Il cliente fornice i propri dati e verrà registrato. |
| Pre-Condizioni | Il cliente non è registrato. |
| Sequenza di eventi  principale | 1. Il caso d’uso inizia quando il sistema riscontra che il cliente non è registrato    1. Il Cliente fornisce username e password.    2. Il sistema controlla la validità di username e password. 2. Il sistema provvederà a registrare il cliente. |
| Post-Condizioni | Il cliente è registrato. |
| Casi d’uso correlati | *Assegnazione fornitura.* |
| Sequenza di eventi  alternativi | - |

|  |  |
| --- | --- |
| Caso d’uso: | Assegnazione Fornitura |
| Attore primario | Impiegato |
| Attore secondario | - |
| Descrizione | La fornitura viene associata ad un cliente |
| Pre-Condizioni | I clienti devono essere registrati |
| Sequenza di eventi  principale | 1. Il caso d’uso inizia quando l’impiegato fornisce i dati del cliente, l’indirizzo dell’unità abitativa e il tipo di fornitura al sistema 2. Se il cliente e l’unità abitativa sono già registrati, il sistema controlla che il cliente non possieda già una fornitura di quel tipo nell’unità specificata.    1. In caso di esito positivo il sistema crea una nuova fornitura.    2. In caso di esito negativo, il sistema restituisce un messaggio di errore e chiede all’ impiegato di fornire di nuovo i dati    3. Il sistema assegna un identificativo univoco, il tipo e la tariffa unitaria alla nuova fornitura.    4. Per definire la tariffa il sistema accede ad un archivio interno contenete la tariffe attuali.    5. La nuova fornitura viene associata al cliente e all’ unità abitativa. 3. Il sistema restituisce l’identificativo della fornitura creata. |
| Post-Condizioni | Viene creato un contatore associato alla fornitura. |
| Casi d’uso correlati | *Registrazione* |
| Sequenza di eventi  alternativi | - |

|  |  |
| --- | --- |
| Caso d’uso: | Visualizza fattura |
| Attore primario | Cliente registrato |
| Attore secondario | - |
| Descrizione | Il cliente stampa l’ultima fattura. |
| Pre-Condizioni | Il cliente deve essere registrato. |
| Sequenza di eventi  principale | 1. Il caso d’uso inizia quando il cliente fornisce i propri dati.  2. Il sistema controlla la validità dei dati.  2.1. Se il controllo è positivo.   * 1. Il cliente può stampare la fattura. |
| Post-Condizioni | - |
| Casi d’uso correlati |  |
| Sequenza di eventi  alternativi | - |

|  |  |
| --- | --- |
| Caso d’uso: | Inserimento lettura mensile |
| Attore primario | Cliente registrato |
| Attore secondario | - |
| Descrizione | Il Cliente registrato deve fornire mensilmente la lettura dei consumi delle proprie forniture al sistema. |
| Pre-Condizioni | Il cliente deve essere registrato. |
| Sequenza di eventi  principale | 1. Il caso d’ uso inizia quando il sistema controlla che il cliente è registrato. 2. Il cliente fornisce tutti i dati.    1. Il sistema controlla la correttezza dei dati e che la fornitura esiste e sia associata a quel cliente.    2. Il sistema controlla il valore della lettura.    3. Se i controlli hanno dato esito positivo il sistema aggiorna le letture. |
| Post-Condizioni | - |
| Casi d’uso correlati | *Visualizza fattura* |
| Sequenza di eventi  alternativi | - |

|  |  |
| --- | --- |
| Caso d’uso: | Genera fattura |
| Attore primario | Tempo |
| Attore secondario | - |
| Descrizione | Il sistema al secondo giorno di ogni mese genera le fatture mensili |
| Pre-Condizioni | - |
| Sequenza di eventi  principale | 1. Il caso d’uso inizia il secondo mese di ogni giorno 2. Il sistema calcola l’importo da pagare 3. Il sistema genera la fattura |
| Post-Condizioni | - |
| Casi d’uso correlati | *Visualizza fattura* |
| Sequenza di eventi  alternativi | - |

|  |  |
| --- | --- |
| Caso d’uso: | Genera report consumi annuali |
| Attore primario | Direttore commerciale |
| Attore secondario | - |
| Descrizione | Il sistema genera report sui consumi annuali di una data città |
| Pre-Condizioni | - |
| Sequenza di eventi  principale | 1. Il caso d’uso inizia quando il direttore inserisce la città 2. Il sistema calcola i consumi per quella città 3. Il sistema stampa il report annuale |
| Post-Condizioni | - |
| Casi d’uso correlati | *-* |
| Sequenza di eventi  alternativi | - |

|  |  |
| --- | --- |
| Caso d’uso: | Genera statistiche sulle forniture erogate |
| Attore primario | Direttore commerciale |
| Attore secondario | - |
| Descrizione | Il sistema genera statistiche sulle forniture erogate |
| Pre-Condizioni | - |
| Sequenza di eventi  principale | 1. Il caso d’uso inizia quando il direttore commerciale inserisce un periodo e una città 2. Il sistema genera le statistiche 3. Il sistema mostra a video le statistiche |
| Post-Condizioni | - |
| Casi d’uso correlati | *-* |
| Sequenza di eventi  alternativi | - |

## Diagramma delle classi

Diagramma delle classi di analisi.

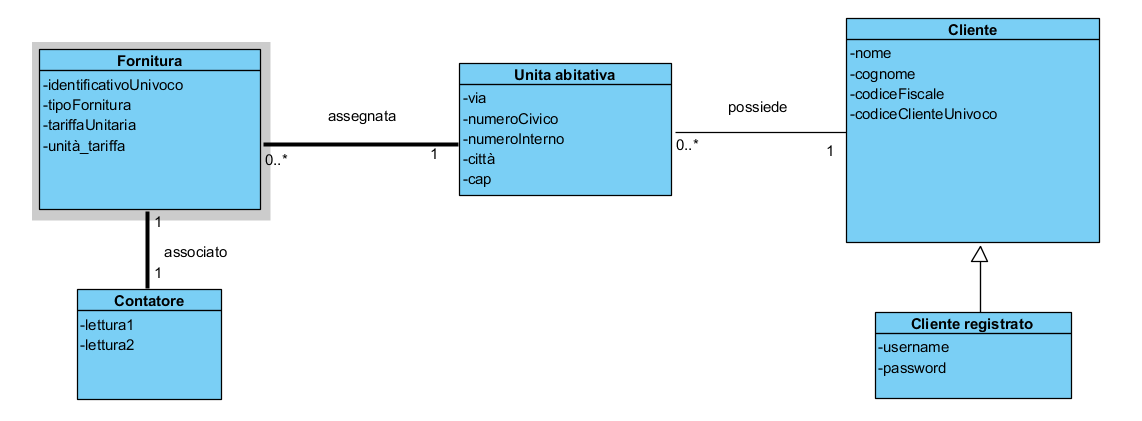
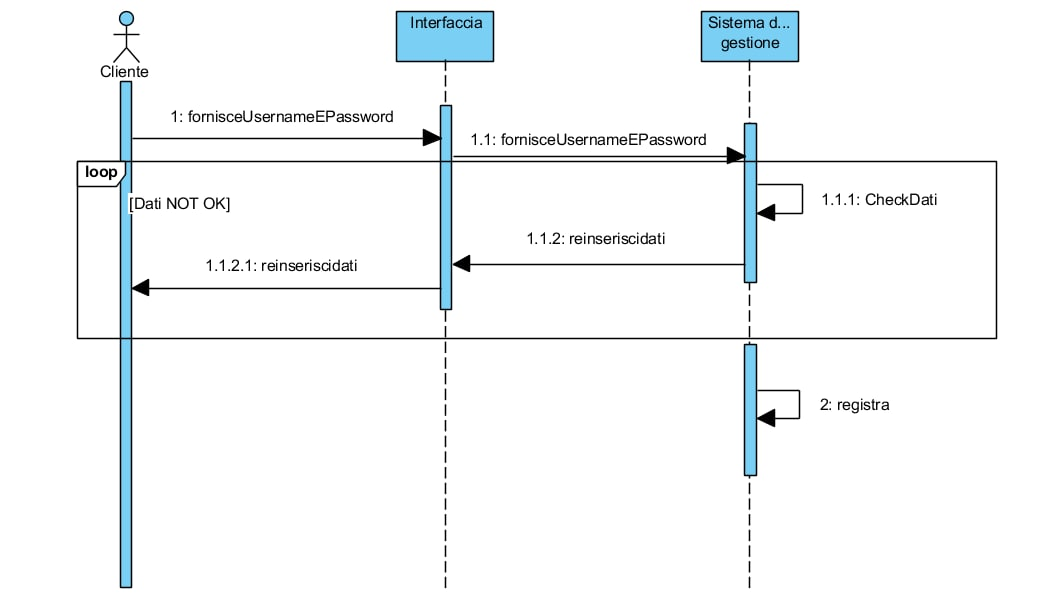


Diagramma delle classi raffinato (con classi Control e Boundary).

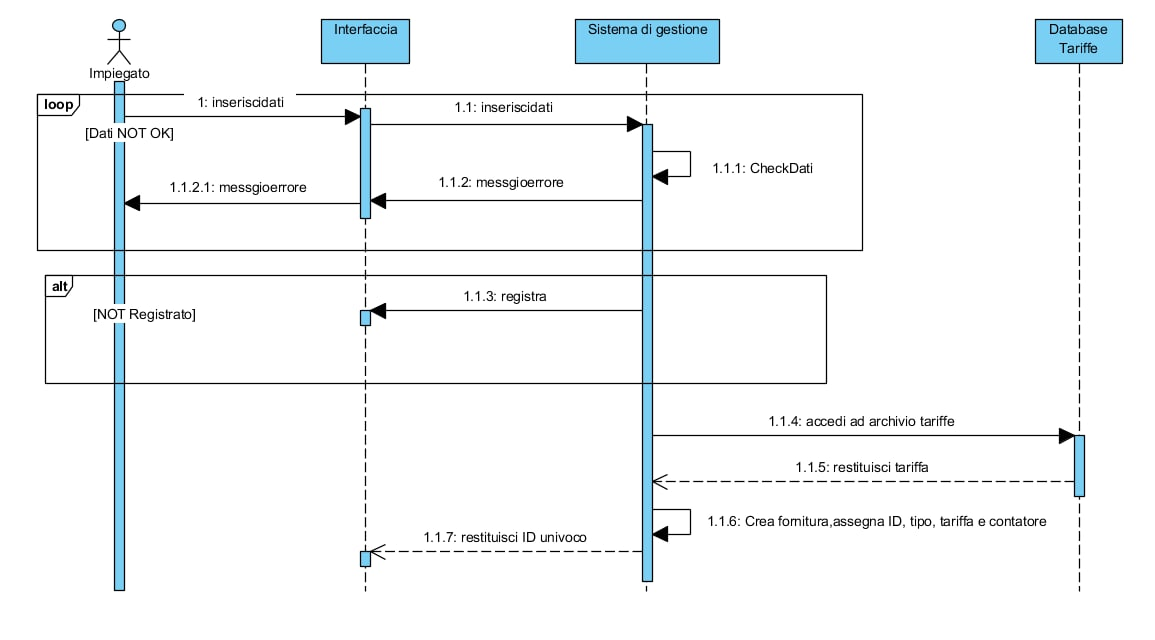


## Diagrammi di sequenza

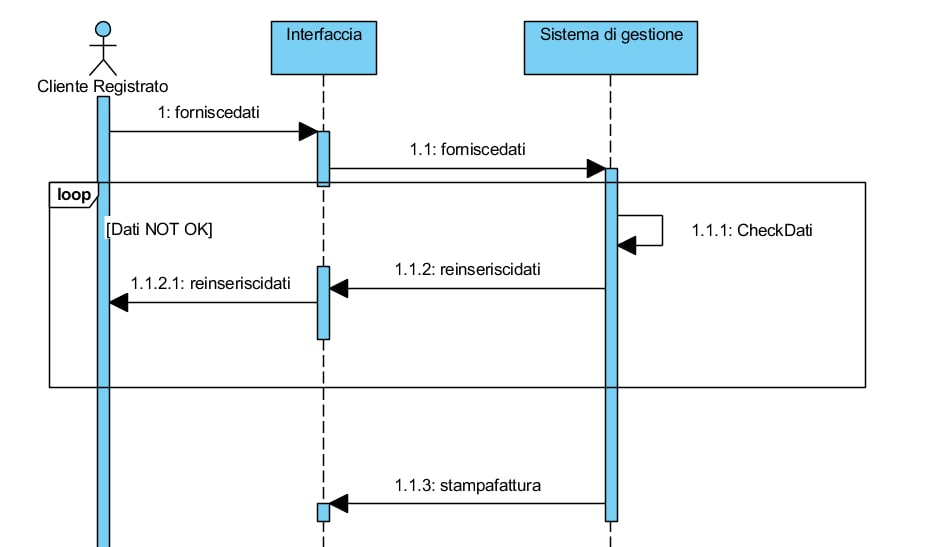
## Registrazione



## Assegnazione Fornitura

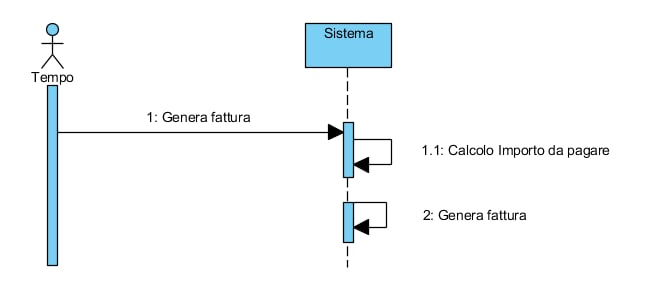


## Visualizza Fattura

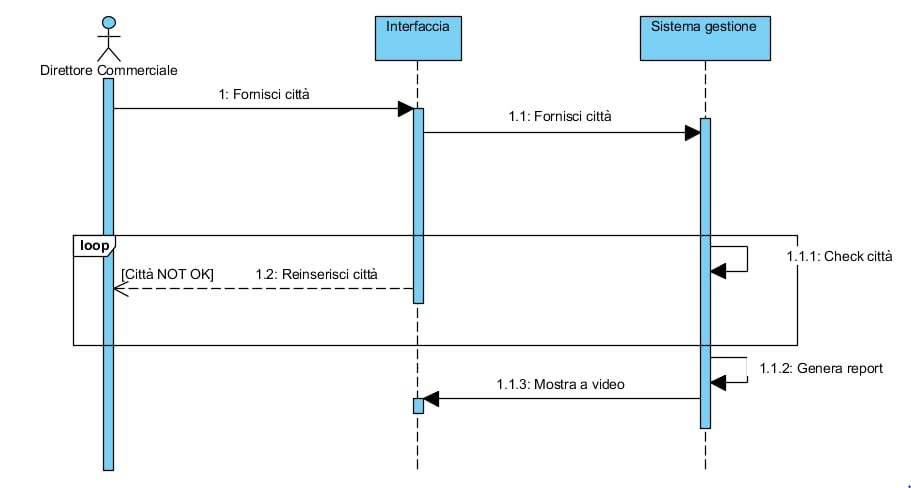


## Inserimento lettura mensile

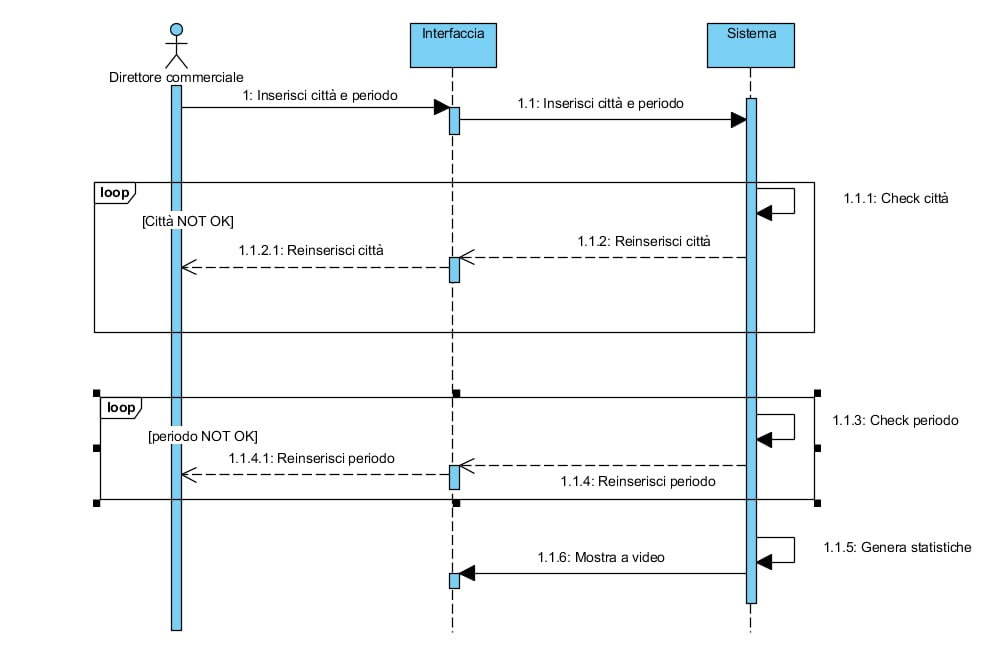
## Genera Fattura



## Genera report consumi annuali



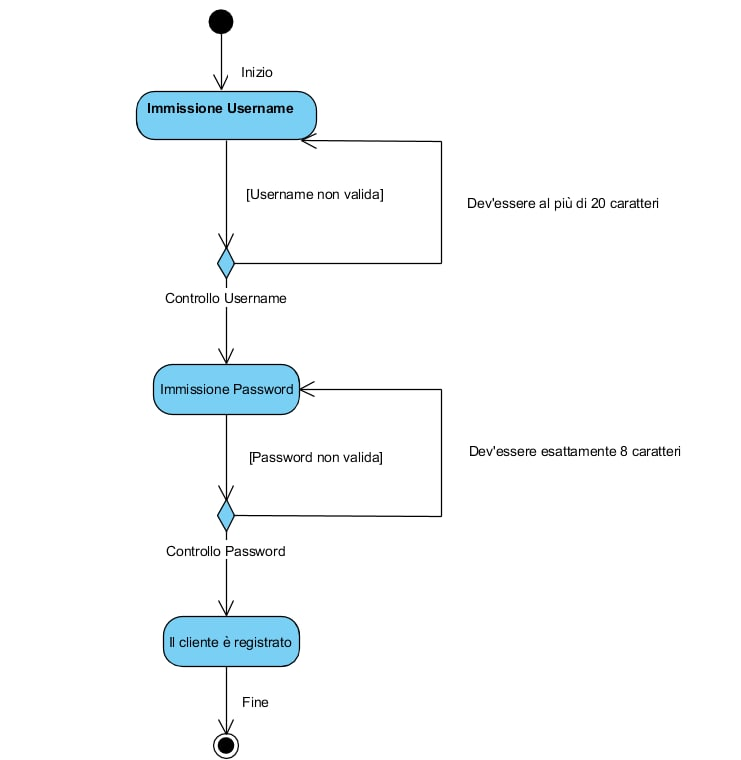
## Genera statistiche sulle forniture erogate



## Diagramma di Attività

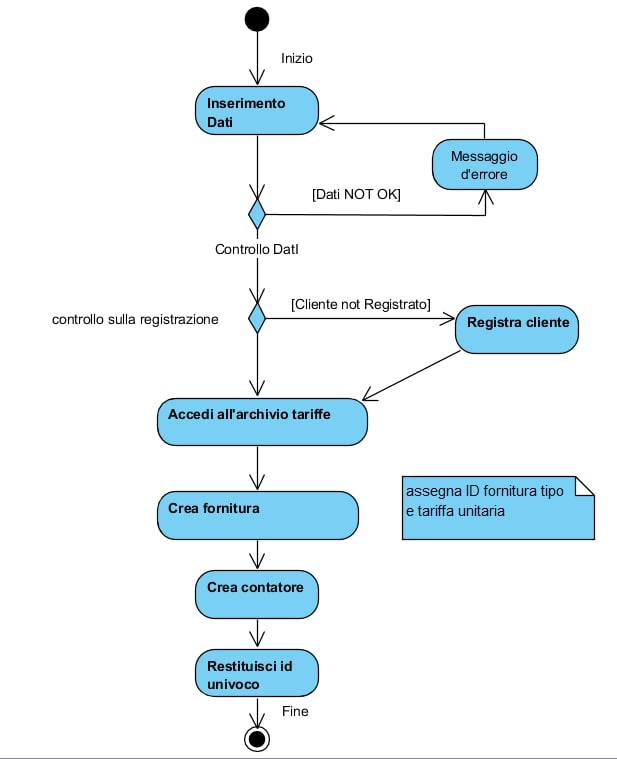
## Registrazione

Diagramma di flusso per la modellazione del processo di registrazione.



## Assegnazione Fattura

Diagramma di flusso per la modellazione del processo di Assegnazione Fattura.



## Verifica della completezza dei requisiti

Si suggerisce allo studente di verificare che tutti i requisiti informali siano rappresentati nel modello UML e/o negli scenari. A tale scopo, lo studente può elencare i requisiti (funzionali, requisiti sui dati, altri requisiti) riportando per ciascun requisito gli elementi dei diagrammi UML con i quali è rappresentato nel modello di analisi. Dopo l’elencazione, lo studente verifichi che tutti i requisiti siano stati modellati nei diagrammi UML.

Legenda: UCD = Use Case Diagram, CD = Class Diagram

* **RF01** è modellato nell’UCD con l’attore “Impiegato” e con il caso d’uso UC4.
* **RF02** è modellato nell’UCD con l’attore “Impiegato” e con il caso d’uso UC13.
* **RF03** è modellato nell’UCD con il caso d’uso UC11.
* **RF04** è modellato nell’UCD con il caso d’uso UC4.
* **RF05** è modellato nell’UCD con il caso d’uso UC11
* **RF06** è modellato nell’UCD con il caso d’uso UC4.
* **RF07** è modellato nell’UCD con il caso d’uso UC4.
* **RF08** è modellato nell’UCD con il caso d’uso UC4.
* **RF09** è modellato nell’UCD con il caso d’uso UC13.
* **RF10** è modellato nell’UCD con il caso d’uso UC10.
* **RF11** è modellato nell’UCD con l’attore “Cliente” e il caso d’uso UC1.
* **RF12** è modellato nell’UCD con il caso d’uso UC8.
* **RF13** è modellato nell’UCD con il caso d’uso UC9.
* **RF14** è modellato nell’UCD con il caso d’uso UC9.
* **RF15** è modellato nell’UCD con il caso d’uso UC3.
* **RF16** è modellato nell’UCD con l’attore “Tempo” e il caso d’uso UC5.
* **RF17** è modellato nell’UCD con l’attore “Cliente Registrato” e il caso d’uso UC2.
* **RF18** è modellato nell’UCD con l’attore “Direttore Commerciale” e il caso d’uso UC6.
* **RF19** è modellato nell’UCD con l’attore “Direttore Commerciale” e il caso d’uso UC7.
* **RD01** è modellato nel CD con l’associazione tra le classi “Cliente” e “Unità abitativa”.
* **RD02** è modellato nel CD con l’associazione tra le classi “Cliente” e “Unità abitativa”.
* **RD03** è modellato nel CD con l’associazione tra le classi “Unità abitativa” e “Fornitura” tramite l’attributo “Tipo Fornitura”.
* **RD04** è modellato nel CD tramite l’attributo “Tipo Fornitura” indirizzo della classe “Unità abitativa”.
* **RD05** è modellato nel CD con gli attributi “Via”, “Numero Civico”, “Numero Interno”,” Città” e “Cap” della classe “Indirizzo”.
* **RD06** è modellato nel CD con gli attributi “Nome”,” Cognome”, “Codice Fiscale” e “Codice Cliente Univoco” della classe “Cliente”.
* **RD07** è modellato nel CD con gli attribuiti “Identificativo Univoco”, “Tipo Fornitura”,” Tariffa Unitaria” e “Unità Tariffa” della classe “Fornitura”.
* **RD08** è modellato nel CD con gli attribuiti “Lettura1” e “Lettura2” della classe “Contatore”.
* **RD09** è modellato nel CD con gli attributi “Username” e “Password” della classe “Cliente Registrato” .
* **RD10**

# Stima dei costi

Riportare la stima dei costi secondo il metodo dei Punti Funzione.

* Tabella di riferimento per le complessità di dati e transazioni

|  |  |  |  |
| --- | --- | --- | --- |
|  | SEMPLICE | MEDIO | COMPLESSO |
| NILF | **7** | **10** | **15** |
| NEIF | **5** | **7** | **10** |
| NEI | **3** | **4** | **6** |
| NEO | **4** | **5** | **7** |
| NEQ | **3** | **4** | **6** |

* Tabella elenco dei fattori correttivi (il cui valore è compreso tra 0 e 5)

|  |  |
| --- | --- |
| Comunicazione Dati | 2 |
| Distribuzione Elaborazione | 3 |
| Prestazione | 2 |
| Utilizzo Intensivo Configurazione | 1 |
| Frequenza delle transazioni | 5 |
| Inserimento Dati Interattivo | 3 |
| Efficienza Per l'Utente Finale | 2 |
| Aggiornamento Interattivo | 0 |
| Complessità Elaborativa | 2 |
| Riusabilità | 3 |
| Facilità Installazione | 0 |
| Facilità Gestione Operativa | 3 |
| Molteplicità dei Siti | 0 |
| Facilità di Modifica | 3 |

FORMULA

**Registrazione**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | VALORE | SEMPLICE | MEDIO | COMPLESSO | TOT |
| NILF | 1 | **7** | **10** | **15** | 10 |
| NEIF | 0 | **5** | **7** | **10** |  |
| NEI | 2 | **3** | **4** | **6** | 6 |
| NEO | 0 | **4** | **5** | **7** |  |
| NEQ | 0 | **3** | **4** | **6** |  |

|  |  |
| --- | --- |
| UFP= | 16 |
| LLOC/FP = | **848** |

NILF: Il sistema controlla che l’username sia univoco [1 medio]

NEI: Username, Password [2 semplici]

|  |  |
| --- | --- |
| FP= | 14.88 |
| JAVA = | **789** |

**Assegnazione Fornitura**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | VALORE | SEMPLICE | MEDIO | COMPLESSO | TOT |
| NILF | 2 | **7** | **10** | **15** | 20 |
| NEIF | 1 | **5** | **7** | **10** | 5 |
| NEI | 2 | **3** | **4** | **6** | 27 |
| NEO | 0 | **4** | **5** | **7** |  |
| NEQ | 0 | **3** | **4** | **6** |  |

|  |  |
| --- | --- |
| FP= | 48.36 |
| JAVA = | **2563** |

|  |  |
| --- | --- |
| UFP= | 33 |
| LLOC/FP = | **2756** |

NILF: Il sistema controlla che il cliente non possieda già una fornitura di quel tipo nell’unità specificata, Il sistema controlla che il cliente sia registrato. [2 medi]

NEIF: Il sistema accede ad un archivio contenente le tariffe annuale [1 semplice]

NEI: Dati Cliente, Dati Unità abitativa [2 medi]

**Visualizza Fattura**

|  |  |
| --- | --- |
| FP= | 18.6 |
| JAVA= | **986** |

|  |  |
| --- | --- |
| UFP= | 20 |
| LLOC/FP = | **1060** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | VALORE | SEMPLICE | MEDIO | COMPLESSO | TOT |
| NILF | 1 | **7** | **10** | **15** | 10 |
| NEIF | 0 | **5** | **7** | **10** |  |
| NEI | 2 | **3** | **4** | **6** | 6 |
| NEO | 1 | **4** | **5** | **7** | 4 |
| NEQ | 0 | **3** | **4** | **6** |  |

NILF: Il sistema controlla la validità dei dati. [1 medio]

NEI: Username e Password [2 semplici]

NEO: Il sistema mostra a video la fattura [1 semplice]

**Inserimento lettura mensile**

|  |  |
| --- | --- |
| FP= | 43 |
| JAVA= | **2267** |

|  |  |
| --- | --- |
| UFP= | 46 |
| LLOC/FP = | **2438** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | VALORE | SEMPLICE | MEDIO | COMPLESSO | TOT |
| NILF | 4 | **7** | **10** | **15** | 34 |
| NEIF | 0 | **5** | **7** | **10** |  |
| NEI | 4 | **3** | **4** | **6** | 12 |
| NEO | 0 | **4** | **5** | **7** |  |
| NEQ | 0 | **3** | **4** | **6** |  |

NILF: Il sistema controlla la validità di Username e Password, il sistema che la fornitura esista, il sistema controlli che l’associazione fornitura-cliente esita, il sistema controlla il valore della lettura. [2 semplice,2 medi].

NEI: Username, Password, Id Fornitura e lettura [4 semplici]

**Genera Fattura**

|  |  |
| --- | --- |
| FP= | 6.5 |
| JAVA= | **345** |

|  |  |
| --- | --- |
| UFP= | 7 |
| LLOC/FP = | **371** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | VALORE | SEMPLICE | MEDIO | COMPLESSO | TOT |
| NILF | 0 | **7** | **10** | **15** |  |
| NEIF | 0 | **5** | **7** | **10** |  |
| NEI | 0 | **3** | **4** | **6** |  |
| NEO | 1 | **4** | **5** | **7** | 4 |
| NEQ | 1 | **3** | **4** | **6** | 3 |

NEQ: Il sistema calcola l’importa della fattura [1 semplice]

NEO: Il sistema genera la fattura [1 semplice]

|  |  |
| --- | --- |
| UFP= | 26 |
| LLOC/FP= | **1378** |

**Genera report consumi annuali**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | VALORE | SEMPLICE | MEDIO | COMPLESSO | TOT |
| NILF | 1 | **7** | **10** | **15** | 10 |
| NEIF | 0 | **5** | **7** | **10** |  |
| NEI | 3 | **3** | **4** | **6** | 9 |
| NEO | 1 | **4** | **5** | **7** | 4 |
| NEQ | 1 | **3** | **4** | **6** | 3 |

|  |  |
| --- | --- |
| FP= | 24.18 |
| JAVA= | **1282** |

NILF: Il sistema controlla la validità di Username e Password. [1 medio]

NEI: Username, Password, città [3 semplici]

NEO: Il sistema mostra il report [1 semplice]

NEQ: Il sistema calcola i consumi per quella città [1 semplice]

**Genera statistiche sulle forniture erogate**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | VALORE | SEMPLICE | MEDIO | COMPLESSO | TOT |
| NILF | 1 | **7** | **10** | **15** | 10 |
| NEIF | 0 | **5** | **7** | **10** |  |
| NEI | 4 | **3** | **4** | **6** | 12 |
| NEO | 1 | **4** | **5** | **7** | 4 |
| NEQ | 1 | **3** | **4** | **6** | 3 |

|  |  |
| --- | --- |
| FP= | 26.97 |
| JAVA= | **1429** |

|  |  |
| --- | --- |
| UFP= | 29 |
| LLOC/FP = | **1537** |

NILF: Il sistema controlla la validità di username e password. [1 medio]

NEI: Username, Password, città e periodo [4 semplici]

NEO: Il sistema elabora le statistiche richieste [1 semplice]

NEQ: Il sistema mostra a video le statistiche [1 medio]

# Piano di test funzionale

Progettare i casi di test funzionale con la tecnica del *Category Partition Testing*. Descrivere il procedimento di calcolo.

**PIANO DI TEST UTILIZZANDO IL METODO DEL *CATEGORY-PARTITION TESTING* PER LA FUNZIONALITÀ “*Registrazione*”.**

|  |  |
| --- | --- |
| **USERNAME** | **PASSWORD** |
| * Stringa di lunghezza <= 20. * Stringa di lunghezza > 20 [ERROR]. | * Stringa di lunghezza pari a 8. * Stringa di lunghezza > 8. [ERROR]. * Stringa di lunghezza < 8. [ERROR]. |

Il numero di test da effettuarsi senza particolari vincoli è:

Introduciamo i vincoli [ERROR] .

Il numero di test da eseguire per testare singolarmente i vincoli è 3 (1 per Username, 2 per Password).

Il numero di test risultante è: (1\*1) + 3 = 4.

**TEST SUITE**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Descrizione** | **Classi di equivalenza coperte** | **Pre-condizioni** | **Input** | **Output Attesi** | **Post-condizioni Attese** |
| 1 | Tutti input validi. | Username valido  Password valida | Il cliente non è registrato. | {Username: “Ivano”,  Password: “Andrea01”} | Il cliente è registrato. | Cliente è inserito nel database. |
| 2 | Username > 20 caratteri. | Username stringa > 20 caratteri [ERROR],  Password valida |  | {Username: “Ilbuonivanobuono2022”, Password:”Andrea01”} | Username troppo lungo! |  |
| 3 | Password < 8. | Username valido, Password stringa <8[ERROR] |  | {Username: “Ivano”,  Password: “Andrea”} | Password troppo corta! |  |
| 4 | Password > 8. | Username valido, Password stringa >8[ERROR] |  | {Username: “Ivano”,  Password: “Andrea2001”} | Password troppo lungo! |  |

**PIANO DI TEST UTILIZZANDO IL METODO DEL *CATEGORY-PARTITION TESTING* PER LA FUNZIONALITÀ “*Assegnazione fornitura*”.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **NOME** | **COGNOME** | **CODICE FISCALE** | **VIA** | **NUMERO CIVICO** | **CAP** | **NUMERO INTERNO** | **CITTÀ** | **TIPO FORNITURA** |
| * Stringa di caratteri. * Formato non concesso. [ERROR] | * Stringa di caratteri. * Formato non concesso. [ERROR] | * Stringa alfa-numerica di lunghezza =16. * Stringa alfa-numerica di lunghezza <16. [ERROR] * Stringa alfa-numerica di lunghezza >16. [ERROR] * Formato non concesso. [ERROR] | * Stringa di caratteri. * Formato non concesso. [ERROR] | * Stringa alfa-numerica. * Formato non concesso. [ERROR] | * Stringa di numeri di lunghezza = 5. * Stringa di numeri di lunghezza > 5. [ERROR] * Stringa di numeri di lunghezza < 5. [ERROR] * Formato non concesso. [ERROR] | * Stringa alfa-numerica. * Formato non concesso. [ERROR] | * Stringa di caratteri. * Formato non concesso. [ERROR] | * Elettricità. * Gas. * Acqua. * Benzina. [ERROR] |

Il numero di test da effettuarsi senza particolari vincoli è:

Introduciamo i vincoli [ERROR] .

Il numero di test da eseguire per testare singolarmente i vincoli è 13 (1 per Nome,1 per Cognome,3 per Codice Fiscale,1 per Via,1 per Numero Civico,3 per CAP,1 per Numero Interno, 1 per Città e 1 per Tipo Fornitura).

Il numero di test risultante è: (1\*1\*1\*1\*1\*1\*1\*1\*1) + 13 = 14.

**TEST SUITE**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Descrizione** | **Classi di equivalenza coperte** | **Pre-condizioni** | **Input** | **Output Attesi** | **Post-condizioni Attese** |
| 1 | Tutti input validi. | Nome valido  Cognome valido  Codice Fiscale valido  Via valido  Numero Civico valido  CAP valido  Numero Interno valido  Città valido  Tipo Fornitura valido | Il Cliente è registrato. | {Nome: Lorenzo  Cognome: Cappellieri  Codice Fiscale: CCPLNZ99M16F839J  Via: Domenico Fontana  Numero Civico: 41  CAP: 80128  Numero Interno: 1B  Città: Napoli  Tipo Fornitura: Elettricità} | Al Cliente è stata associata la fornitua | Viene Creato un Contatore. |
| 2 | Nome: Formato non valido. | Nome formato non valido [ERROR]  Cognome valido  Codice Fiscale valido  Via valido  Numero Civico valido  CAP valido  Numero Interno valido  Città valido  Tipo Fornitura valido |  | {Nome: Lorenz0  Cognome: Cappellieri  Codice Fiscale: CCPLNZ99M16F839J  Via: Domenico Fontana  Numero Civico: 41  CAP: 80128  Numero Interno: 1B  Città: Napoli  Tipo Fornitura: Elettricità} | Il formato del nome inserito non è valido. |  |
| 3 | Cognome: Formato non valido. | Nome valido  Cognome formato non valido [ERROR]  Codice Fiscale valido  Via valido  Numero Civico valido  CAP valido  Numero Interno valido  Città valido  Tipo Fornitura valido |  | {Nome: Lorenzo  Cognome: Cappellier!  Codice Fiscale: CCPLNZ99M16F839J  Via: Domenico Fontana  Numero Civico: 41  CAP: 80128  Numero Interno: 1B  Città: Napoli  Tipo Fornitura: Elettricità} | Il formato del cognome inserito non è valido. |  |
| 4 | Codice Fiscale : Formato non valido. | Nome valido  Cognome valido  Codice Fiscale formato non valido  [ERROR]  Via valido  Numero Civico valido  CAP valido  Numero Interno valido  Città valido  Tipo Fornitura valido |  | {Nome: Lorenzo  Cognome: Cappellieri  Codice Fiscale: CCPLNZ99M16F839£  Via: Domenico Fontana  Numero Civico: 41  CAP: 80128  Numero Interno: 1B  Città: Napoli  Tipo Fornitura: Elettricità} | Il formato del Codice Fiscale inserito non è valido. |  |
| 5 | Codice Fiscale < 16. | Nome valido  Cognome valido  Codice Fiscale < 16  [ERROR]  Via valido  Numero Civico valido  CAP valido  Numero Interno valido  Città valido  Tipo Fornitura valido |  | {Nome: Lorenzo  Cognome: Cappellieri  Codice Fiscale: CCPLNZ99M16F839  Via: Domenico Fontana  Numero Civico: 41  CAP: 80128  Numero Interno: 1B  Città: Napoli  Tipo Fornitura: Elettricità} | Il formato del Codice Fiscale inserito deve essere di 16 caratteri. |  |
| 6 | Codice Fiscale > 16. | Nome valido  Cognome valido  Codice Fiscale > 16  [ERROR]  Via valido  Numero Civico valido  CAP valido  Numero Interno valido  Città valido  Tipo Fornitura valido |  | {Nome: Lorenzo  Cognome: Cappellieri  Codice Fiscale: CCPLNZ99M16F839J9  Via: Domenico Fontana  Numero Civico: 41  CAP: 80128  Numero Interno: 1B  Città: Napoli  Tipo Fornitura: Elettricità} | Il formato del Codice Fiscale inserito deve essere di 16 caratteri. |  |
| 7 | Via: Formato non valido. | Nome valido  Cognome valido  Codice Fiscale valido  Via formato non valido [ERROR]  Numero Civico valido  CAP valido  Numero Interno valido  Città valido  Tipo Fornitura valido |  | {Nome: Lorenzo  Cognome: Cappellieri  Codice Fiscale: CCPLNZ99M16F839J  Via: Domenico Fontan@  Numero Civico: 41  CAP: 80128  Numero Interno: 1B  Città: Napoli  Tipo Fornitura: Elettricità} | Il formato della via inserito non è valido. |  |
| 8 | Numero civico: Formato non valido. | Nome valido  Cognome valido  Codice Fiscale valido  Via valido  Numero Civico formato non valido [ERROR]  CAP valido  Numero Interno valido  Città valido  Tipo Fornitura valido |  | {Nome: Lorenzo  Cognome: Cappellier1  Codice Fiscale: CCPLNZ99M16F839J  Via: Domenico Fontana  Numero Civico: 41?  CAP: 80128  Numero Interno: 1B  Città: Napoli  Tipo Fornitura: Elettricità} | Il formato del Numero civico inserito non è valido. |  |
| 9 | CAP: Formato non valido. | Nome valido  Cognome valido  Codice Fiscale valido  Via valido  Numero Civico valido  CAP formato non valido [ERROR]  Numero Interno valido  Città valido  Tipo Fornitura valido |  | {Nome: Lorenzo  Cognome: Cappellieri  Codice Fiscale: CCPLNZ99M16F839J  Via: Domenico Fontana  Numero Civico: 41  CAP: 8012L  Numero Interno: 1B  Città: Napoli  Tipo Fornitura: Elettricità} | Il formato del CAP inserito non è valido. |  |
| 10 | CAP < 5. | Nome valido  Cognome valido  Codice Fiscale valido  Via valido  Numero Civico valido  CAP < 5[ERROR]  Numero Interno valido  Città valido  Tipo Fornitura valido |  | {Nome: Lorenzo  Cognome: Cappellieri  Codice Fiscale: CCPLNZ99M16F839J  Via: Domenico Fontana  Numero Civico: 41  CAP: 8012  Numero Interno: 1B  Città: Napoli  Tipo Fornitura: Elettricità} | Il CAP deve essere di 5 numeri. |  |
| 11 | CAP > 5 | Nome valido  Cognome valido  Codice Fiscale valido  Via valido  Numero Civico valido  CAP > 5[ERROR]  Numero Interno valido  Città valido  Tipo Fornitura valido |  | {Nome: Lorenzo  Cognome: Cappellieri  Codice Fiscale: CCPLNZ99M16F839J  Via: Domenico Fontana  Numero Civico: 41  CAP: 801280  Numero Interno: 1B  Città: Napoli  Tipo Fornitura: Elettricità} | Il CAP deve essere di 5 numeri. |  |
| 12 | Numero Interno: Formato non valido. | Nome valido  Cognome valido  Codice Fiscale valido  Via valido  Numero Civico valido  CAP valido  Numero Interno formato non valido [ERROR]  Città valido  Tipo Fornitura valido |  | {Nome: Lorenzo  Cognome: Cappellieri  Codice Fiscale: CCPLNZ99M16F839J  Via: Domenico Fontana  Numero Civico: 41  CAP: 80128  Numero Interno: 1B;  Città: Napoli  Tipo Fornitura: Elettricità} | Il formato del numero interno inserito non è valido.! |  |
| 13 | Città: Formato non valido. | Nome valido  Cognome valido  Codice Fiscale valido  Via valido  Numero Civico valido  CAP valido  Numero Interno valido  Città formato non valido [ERROR]  Tipo Fornitura valido |  | {Nome: Lorenzo  Cognome: Cappellieri  Codice Fiscale: CCPLNZ99M16F839J  Via: Domenico Fontana  Numero Civico: 41  CAP: 80128  Numero Interno: 1B  Città: Napol1  Tipo Fornitura: Elettricità} | Il formato della Città inserito non è valido! |  |
| 14 | Tipo Fornitura: Non presente. | Nome valido  Cognome valido  Codice Fiscale valido  Via valido  Numero Civico valido  CAP valido  Numero Interno valido  Città valido  Tipo Fornitura non presente[ERROR] |  | {Nome: Lorenzo  Cognome: Cappellieri  Codice Fiscale: CCPLNZ99M16F839J  Via: Domenico Fontana  Numero Civico: 41  CAP: 80128  Numero Interno: 1B  Città: Napoli  Tipo Fornitura: Benzina} | La Fornitura inserita non è disponibile. |  |

**PIANO DI TEST UTILIZZANDO IL METODO DEL *CATEGORY-PARTITION TESTING* PER LA FUNZIONALITÀ “*Visualizza Fattura*”.**

|  |  |
| --- | --- |
| **USERNAME** | **PASSWORD** |
| * Stringa di lunghezza <= 20. * Stringa di lunghezza > 20 [ERROR]. | * Stringa di lunghezza pari a 8. * Stringa di lunghezza > 8. [ERROR]. * Stringa di lunghezza < 8. [ERROR]. |

Il numero di test da effettuarsi senza particolari vincoli è:

Introduciamo i vincoli [ERROR] .

Il numero di test da eseguire per testare singolarmente i vincoli è 3 (1 per Username, 2 per Password).

Il numero di test risultante è: (1\*1) + 3 = 4.

**TEST SUITE**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Descrizione** | **Classi di equivalenza coperte** | **Pre-condizioni** | **Input** | **Output Attesi** | **Post-condizioni Attese** |
| 1 | Tutti input validi. | Username valido  Password valida | Il cliente è registrato. | {Username: “Ivano”,  Password: “Andrea01”} | Visualizzazione Fattura. |  |
| 2 | Username > 20 caratteri. | Username stringa > 20 caratteri [ERROR],  Password valida |  | {Username: “Ilbuonivanobuono2022”, Password:”Andrea01”} | Username troppo lungo! |  |
| 3 | Password < 8. | Username valido, Password stringa <8[ERROR] |  | {Username: “Ivano”,  Password: “Andrea”} | Password troppo corta! |  |
| 4 | Password > 8. | Username valido, Password stringa >8[ERROR] |  | {Username: “Ivano”,  Password: “Andrea2001”} | Password troppo lungo! |  |

**PIANO DI TEST UTILIZZANDO IL METODO DEL *CATEGORY-PARTITION TESTING* PER LA FUNZIONALITÀ “*Inserimento Lettura Mensile*”.**

|  |  |  |  |
| --- | --- | --- | --- |
| **USERNAME** | **PASSWORD** | **ID FORNITURA** | **LETTURA** |
| * Stringa di lunghezza <= 20. * Stringa di lunghezza > 20 [ERROR]. | * Stringa di lunghezza pari a 8. * Stringa di lunghezza > 8. [ERROR]. * Stringa di lunghezza < 8. [ERROR]. | * Stringa numerica. * Formato non valido.   [ERROR] | * Numero decimale. * Numero Intero. [ERROR] * Formato non valido.   [ERROR] |

Il numero di test da effettuarsi senza particolari vincoli è:

Introduciamo i vincoli [ERROR] .

Il numero di test da eseguire per testare singolarmente i vincoli è 6 (1 per Username, 2 per Password,1 per Id Fornitura,2 per Lettura).

Il numero di test risultante è: (1\*1\*1\*1) + 6 = 7.

**TEST SUITE**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Descrizione** | **Classi di equivalenza coperte** | **Pre-condizioni** | **Input** | **Output Attesi** | **Post-condizioni Attese** |
| 1 | Tutti input validi. | Username valido  Password valida  Id Fornitura valido  Lettura valida | Il cliente è registrato. | {Username: “Ivano”,  Password: “Andrea01”  Id Fornitura: “1”  Lettura: “200.11”} | Lettura inserita. | Inserisci Lettura. |
| 2 | Username > 20 caratteri. | Username stringa > 20 caratteri [ERROR],  Password valida  Id Fornitura valido  Lettura valida |  | {Username: “Ilbuonivanobuono2022”, Password:”Andrea01”  Id Fornitura: “1”  Lettura: “200.11”} | Username troppo lungo! |  |
| 3 | Password < 8. | Username valido, Password stringa <8[ERROR]  Id Fornitura valido  Lettura valida |  | {Username: “Ivano”,  Password: “Andrea”  Id Fornitura: “1”  Lettura: “200.11”} | Password troppo corta! |  |
| 4 | Password > 8. | Username valido, Password stringa >8[ERROR]  Id Fornitura valido  Lettura valida |  | {Username: “Ivano”,  Password: “Andrea2001”  Id Fornitura: “1”  Lettura: “200.11”} | Password troppo lungo! |  |
| 5 | Id Fornitura Formato non valido | Username valido  Password valida  Id Fornitura formato non valido[ERROR],  Lettura valida |  | {Username: “Ivano”,  Password: “Andrea01”  Id Fornitura: “C”  Lettura: “200.11”} | Formato Id Fornitura non valido. |  |
| 6 | Lettura formato non valido | Username valido  Password valida  Id Fornitura valido  Lettura formato non valido[ERROR], |  | {Username: “Ivano”,  Password: “Andrea01”  Id Fornitura: “1”  Lettura: “200.C”} | Formato Lettura non valido. |  |
| 7 | Lettura: Numero Intero | Username valido  Password valida  Id Fornitura valido  Lettura numero intero[ERROR], |  | {Username: “Ivano”,  Password: “Andrea01”  Id Fornitura: “1”  Lettura: “200”} | Inserire un numero decimale come Lettura. |  |

**PIANO DI TEST UTILIZZANDO IL METODO DEL *CATEGORY-PARTITION TESTING* PER LA FUNZIONALITÀ “Genera report consumi annuali”.**

|  |
| --- |
| **CITTÀ** |
| * Stringa di caratteri * Formato non concesso [ERROR]. |

Il numero di test da effettuarsi senza particolari vincoli è: 2

Introduciamo i vincoli [ERROR] .

Il numero di test da eseguire per testare singolarmente i vincoli è 1 (1 per Città).

Il numero di test risultante è: 1 + 1 = 2.

**TEST SUITE**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Descrizione** | **Classi di equivalenza coperte** | **Pre-condizioni** | **Input** | **Output Attesi** | **Post-condizioni Attese** |
| 1 | Tutti input validi. | Città valida | La città è presente nel database | {Città: “Formia”} | Report generato |  |
| 2 | Città formato non valido | Città formato non valido [ERROR] |  | {Città: “Adone2”} | Il formato della Città inserito non è valido! |  |

**PIANO DI TEST UTILIZZANDO IL METODO DEL *CATEGORY-PARTITION TESTING* PER LA FUNZIONALITÀ “Genera statistiche sulle forniture erogate”.**

|  |  |
| --- | --- |
| **CITTÀ** | **PERIODO** |
| * Stringa di caratteri * Formato non concesso [ERROR]. | * Formato {mm/aa – mm/aa} con 0<mm<13 e data<data odierna * Data > Data odierna [ERROR] * Mese > 12 [ERROR] * Mese < 1 [ERROR] |

Il numero di test da effettuarsi senza particolari vincoli è:

Introduciamo i vincoli [ERROR] .

Il numero di test da eseguire per testare singolarmente i vincoli è 4 (1 per Città, 3 per Periodo).

Il numero di test risultante è: (1\*1) + 4= 5.

**TEST SUITE**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Descrizione** | **Classi di equivalenza coperte** | **Pre-condizioni** | **Input** | **Output Attesi** | **Post-condizioni Attese** |
| 1 | Tutti input validi. | Città valida  Periodo valido | La città è presente nel database | {Città: “Formia”,  Periodo: “02/2020-03/2021”} | Statistiche generate |  |
| 2 | Città formato non valido | Città formato non valido [ERROR] |  | {Città: “Adone2”,  Periodo: “02/2020- 03/2021} | Il formato della Città inserito non è valido! |  |
| 3 | Periodo non valido | Città valida  Periodo con data > data odierna [ERROR] |  | {Città: “Formia”,  Periodo “02/2020-03/3000”} | La data inserita è successiva alla data odierna |  |
| 4 | Mese < 1 | Città valida  Periodo con mese < 1 [ERROR] |  | {Città: “Formia”,  Periodo “02/2020-00/2021”} | Il formato del mese inserito non è valido! |  |
| 5 | Mese > 12 | Città valida  Periodo con mese > 12 [ERROR] |  | {Città: “Formia”,  Periodo “15/2020-03/2021”} | Il formato del mese inserito non è valido! |  |

# Progettazione

## Diagramma delle classi

Riportare il diagramma delle classi di progettazione. Reificare eventuali classi associative del diagramma delle classi di analisi. Specificare argomenti e tipo di ritorno delle operazioni (per quelle più significative, coinvolte nei casi d’uso sviluppati fino alla implementazione). Includere classi del dominio della soluzione, come strutture dati e classi DAO. Raggruppare le classi in package.



## Diagrammi di sequenza

Riportare i diagrammi di sequenza di progetto per il/i casi d’uso sviluppati fino alla codifica in Java.



# Implementazione

Non includere il codice sorgente, ma descrivere l’implementazione in Java, descrivendo gli artefatti di codifica:

* Elencare:
  + package, classi, tipi di eccezione definiti
* Elencare gli artefatti necessari per l’installazione ed esecuzione del programma, senza ovviamente l’ambiente di sviluppo come Eclipse (DB h2, eventuali librerie e versioni di Java che l’utilizzatore deve avere installati, file .class, .jar, …)
* Produrre un eventuale diagramma di deployment
* Eventualmente inserire la documentazione del codice prodotta con Javadoc (relativamente alle funzionalità implementate)
* Riportare il numero di LOC e di LLOC scritte in Java
* Confrontare con la stima dei costi effettuata e commentare eventuali scostamenti

# Testing

## Test strutturale

## Complessità ciclomatica

Costruire il Control Flow Graph per uno o due dei metodi delle classi implementate (si scelgano metodi non proprio banali), e:

- si mostri il calcolo del numero ciclomatico;

- si indichino i percorsi linearmente indipendenti;

Prima o a fianco del CFG riportare il codice Java del metodo.

Es.:

public static boolean CalcolaStatistica(String stringaData, int CAP, boolean scelta) {

if(scelta)

stringaData = "01/"+stringaData;

LocalDate data = null;

DateTimeFormatter dateTimeFormatter = DateTimeFormatter.ofPattern("dd/MM/yyyy");

try {

if(!(MonitoraggioAmbiente.getInstance().VerificaCAP(CAP))) { //Verifica CAP

System.err.println("CAP non valido");

return false;

}

}

catch(IOException e) {

System.err.println("Errore di I/O durante il controllo del CAP" + e.getMessage());

return false;

}

try {

data = LocalDate.parse(stringaData, dateTimeFormatter); //Dalla sequenza data.parse() != null

if(data==null) { //Verifica DATA

System.err.println("Data non valida");

return false;

}

}catch(DateTimeParseException e) {

System.err.println("Data non valida");

return false;

}

java.util.Map<String, Float> Dati = null;

try {

if(scelta) {

Dati = MonitoraggioAmbiente.getInstance().CalcolaStatisticaMese(CAP, data);

System.out.println("Statistica del mese inserito:\n");

}

else {

Dati = MonitoraggioAmbiente.getInstance().CalcolaStatisticaGiorno(CAP, data);

System.out.println("Statistica del giorno inserito:\n");

}

for(String key:Dati.keySet()) {

System.out.println(key +":"+(key.length()<14?"\t\t":"\t") + Dati.get(key)); //Output

}

} catch (StatisticaException e) {

System.err.println(e.getMessage()); return false;

} catch (MisuraException e) {

System.err.println(e.getMessage()); return false;

}

return true;

}

}

Control Flow Graph



NUMERO CICLOMATICO:

numero di regioni chiuse del grafo = 6

numero di nodi predicati (0,2,4,6,10) +1 = 6

# archi – # nodi + 2 = (15 - 11) + 2 = 6

CAMMINI:

1. 0-1-2-3
2. 0-1-2-4-5
3. 0-1-2-4-6-7-9
4. 0-1-2-4-6-8-9
5. 0-1-2-4-6-7-9-10-9
6. 0-2-4-6-7-9

## Test funzionale

Descrivere i risultati dell’esecuzione dei test funzionali precedentemente pianificati adoperando lo schema di tabella seguente.

Descrivere le eventuali azioni di *debugging* a seguito di casi di test con esito *FAIL*.

Commentare se gli eventuali difetti rilevati dal test funzionale potevano essere rilevati anche da un test strutturale.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Descrizione** | **Classi di equivalenza coperte** | **Pre-condizioni** | **Input** | **Output Attesi** | **Post-condizioni Attese** | **Output Ottenuti** | **Post-condizioni Ottenute** | **Esito**  **(*FAIL*, *PASS*)** |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |