

Università degli Studi di Napoli Federico II

**Corso di Laurea in Ingegneria Informatica**

**Corso di Ingegneria del Software**

**Prof. S. Russo - A.A. 2021 - 22**

***Progetto***

GESTIONE DELLA SCUOLA GUIDA “NIRVANA”

A picture containing text, clipart

Description automatically generated

Gruppo: Nirvana

Studenti:

Francesco Pio Manna N46005124 francescopi.manna@studenti.unina.it

Claudio Pisa N46004717 cl.pisa@studenti.unina.it

Dario Pinto N46004545 dari.pinto@studenti.unina.it

Versione 1.2 del *29/03/22*

**Indice**

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 4

2.4.3 Vincoli / Altri requisiti 4

2.5 Modellazione dei casi d’uso 5

2.5.1 Attori e casi d’uso 5

2.5.2 Diagramma dei casi d’uso 5

2.5.3 Scenari 5

2.6 Modellazione dei dati 2

2.6.1Progettazione concettuale 2

2.7 Diagramma delle classi 2

2.8 Diagrammi di sequenza 2

2.9 Verifica della completezza dei requisiti 2

3. Stima dei costi 3

4. Piano di test funzionale 4

5. Progettazione 5

5.1 Progettazione della base di dati 5

5.1.1 Progettazione logica 5

5.2 Diagramma delle classi 5

5.3 Diagrammi di sequenza 5

6. Implementazione 6

7. Testing 7

7.1 Test strutturale 7

7.1.1 Complessità ciclomatica 7

7.1.2 Test di unità 8

7.2 Test funzionale 8

# Specifiche informali

**T3**: Gestione di una scuola guida

Si vuole realizzare un software per la gestione di una scuola guida. La scuola guida offre corsi per patenti di guida per motocicli (tipo AM, A1, A2, A) e per autoveicoli (tipo B). Il segretario della scuola guida registra i clienti nel sistema all’atto dell’iscrizione, inserendo il loro nome, cognome, data di nascita, indirizzo di posta elettronica, residenza, numero della carta di identità, tipo di patente che si vuole conseguire, ed eventuali patenti già in possesso. Una volta registrato, il cliente riceve dal sistema una mail con delle credenziali auto-generate (username e password) per accedere al sistema. Tramite il sistema, un cliente registrato può esercitarsi effettuando una simulazione della prova scritta d’esame. Il sistema seleziona a caso 40 domande a risposta VERO/FALSO tra quelle presenti in una apposita base dati e le presenta al cliente. Una domanda consiste in una formulazione (e.g., “Il segnale raffigurato indica che è vietata la svolta a destra”), una risposta corretta (e.g., “FALSO”), ed un’eventuale figura (e.g., un segnale stradale). Se il cliente commette più di cinque errori, il quiz termina con esito negativo. Altrimenti, l’esito sarà positivo. Il cliente può inoltre prenotare una lezione di guida con un istruttore. Una lezione di guida ha la durata di un’ora e inizia sempre allo scoccare dell’ora. Un istruttore è un dipendente della scuola guida cui sono associati anche i giorni della settimana e le ore in cui è disponibile (e.g., istruttore A: “LUN 15:00”, “LUN 16:00”, “VEN 09:00”). Gli istruttori sono inseriti nel sistema dal segretario, che per ciascun istruttore inserisce nome, cognome, telefono e indirizzo e-mail. All’atto della prenotazione, il cliente specifica istruttore, data e ora in cui desidera effettuare la lezione di guida (e.g., “ROSSI 07/03/2022 15:00”). Il sistema verifica se la lezione specificata è disponibile per la prenotazione. In caso positivo, il cliente può confermare la lezione e il sistema registra la prenotazione. Ogni giorno alle 07:00 il sistema invia per posta elettronica a ciascun istruttore un prospetto delle sue lezioni prenotate dai clienti per quel giorno, ed invia a ciascun cliente prenotato per quel giorno un promemoria con orario della lezione e nome, cognome e nr. di telefono dell’istruttore.

# Analisi e specifica dei requisiti

## Analisi nomi-verbi

Si vuole realizzare un software per la gestione di una scuola guida. La scuola guida offre corsi per patenti di guida per motocicli (tipo AM, A1, A2, A) e per autoveicoli (tipo B). Il segretario della scuola guida registra i clienti nel sistema all’atto dell’iscrizione, inserendo il loro nome, cognome, data di nascita, indirizzo di posta elettronica, residenza, numero della carta di identità, tipo di patente che si vuole conseguire, ed eventuali patenti già in possesso. Una volta registrato, il cliente riceve dal sistema una mail con delle credenziali auto-generate (username e password) per accedere al sistema. Tramite il sistema, un cliente registrato può esercitarsi effettuando una simulazione della prova scritta d’esame. Il sistema seleziona a caso 40 domande a risposta VERO/FALSO tra quelle presenti in una apposita base dati e le presenta al cliente. Una domanda consiste in una formulazione (e.g., “Il segnale raffigurato indica che è vietata la svolta a destra”), una risposta corretta (e.g., “FALSO”), ed un’eventuale figura (e.g., un segnale stradale). Se il cliente commette più di cinque errori, il quiz termina con esito negativo. Altrimenti, l’esito sarà positivo. Il cliente può inoltre prenotare una lezione di guida con un istruttore. Una lezione di guida ha la durata di un’ora e inizia sempre allo scoccare dell’ora. Un istruttore è un dipendente della scuola guida cui sono associati anche i giorni della settimana e le ore in cui è disponibile (e.g., istruttore A: “LUN 15:00”, “LUN 16:00”, “VEN 09:00”). Gli istruttori sono inseriti nel sistema dal segretario, che per ciascun istruttore inserisce nome, cognome, telefono e indirizzo e-mail. All’atto della prenotazione, il cliente specifica istruttore, data e ora in cui desidera effettuare la lezione di guida (e.g., “ROSSI 07/03/2022 15:00”). Il sistema verifica se la lezione specificata è disponibile per la prenotazione. In caso positivo, il cliente può confermare la lezione e il sistema registra la prenotazione. Ogni giorno alle 07:00 il sistema invia per posta elettronica a ciascun istruttore un prospetto delle sue lezioni prenotate dai clienti per quel giorno, ed invia a ciascun cliente prenotato per quel giorno un promemoria con orario della lezione e nome, cognome e nr. di telefono dell’istruttore.

*LEGENDA:  
Classe  
Attributo  
Funzionalità  
Attore*

*Classe-Attore*

## Revisione dei requisiti

1. I corsi offerti dalla scuola guida per il conseguimento della patenta di guida sono per motocicli e per autoveicoli.
2. Il sistema deve consentire al segretario di registrare i clienti all’atto dell’iscrizione.
3. Per ogni cliente si vuole memorizzare nome, cognome, data di nascita, indirizzo di posta elettronica, residenza, numero della carta di identità, tipo di patente che si vuole conseguire e/o già possedute.
4. Il sistema deve inviare una mail ai clienti registrati contenente credenziali autogenerate (username e password).
5. Il sistema deve fornire la possibilità al cliente di esercitarsi effettuando una simulazione di prova scritta d’esame.
6. Il sistema deve selezionare a caso 40 domande a risposta VERO/FALSO e presentarle al cliente.
7. Una domanda consiste in una formulazione, una risposta corretta ed una eventuale figura.
8. Il sistema deve terminare il quiz con esito negativo nel caso in cui il cliente commetta più di cinque errori.
9. Il sistema deve garantire la possibilità al cliente di prenotare una lezione di guida indicando istruttore, data e ora della stessa.
10. Una lezione di guida ha la durata di un’ora e inizia sempre allo scoccare dell’ora.
11. Ad ogni istruttore sono associati i giorni della settimana e le ore in cui è disponibile.
12. Il sistema deve offrire al segretario una funzionalità per la registrazione degli istruttori nel sistema.
13. La registrazione degli istruttori deve prevedere l’inserimento di nome, cognome, telefono e indirizzo e-mail.
14. Il sistema deve verificare se la lezione specificata sia disponibile per la prenotazione.
15. Il sistema deve garantire al cliente la funzionalità per confermare una lezione nel caso in cui essa sia disponibile.
16. Il sistema deve registrare la prenotazione confermata dal cliente.
17. Il sistema deve inviare per posta elettronica (ogni giorno alle 07:00) a un istruttore un prospetto delle sue lezioni prenotate dai clienti per quel giorno.
18. Il sistema deve inviare a ciascun cliente prenotato per un determinato giorno un promemoria con orario della lezione e nome, cognome e numero di telefono dell’istruttore.
19. Le lezioni di guida sono erogate esclusivamente dal lunedì al venerdì.

## Glossario dei termini

|  |  |  |
| --- | --- | --- |
| Termine | Descrizione | Sinonimi |
| Corso | Uno dei corsi forniti dalla scuola guida per conseguire una patente |  |
| Segretario | Un dipendente che registra i clienti all’atto dell’iscrizione e gli istruttori |  |
| Cliente | Un cliente deve registrarsi al sistema | Cliente registrato |
| Prova scritta | Insieme di 40 domande per verificare le conoscenze acquisite dall’esaminando | Quiz |
| Istruttore | Un dipendente che assiste il cliente durante la lezione di guida |  |
| Cliente prenotato | Un cliente registrato che prenota una lezione di guida |  |

## Classificazione dei requisiti

### Requisiti funzionali

|  |  |  |
| --- | --- | --- |
| ID | Requisito | Origine (n. frase dei requisiti revisionati) |
| RF01 | Il sistema deve consentire al segretario di registrare i clienti all’atto dell’iscrizione. | 2 |
| RF02 | Il sistema deve inviare una mail ai clienti registrati contenente credenziali autogenerate (username e password). | 4 |
| RF03 | Il sistema deve fornire la possibilità al cliente di esercitarsi effettuando una simulazione di prova scritta d’esame. | 5 |
| RF04 | Il sistema deve selezionare a caso 40 domande a risposta VERO/FALSO e presentarle al cliente | 6 |
| RF05 | Il sistema deve terminare il quiz con esito negativo nel caso in cui il cliente commetta più di cinque errori | 8 |
| RF06 | Il sistema deve garantire la possibilità al cliente di prenotare una lezione di guida indicando istruttore, data e ora della stessa. | 9 |
| RF07 | Il sistema deve offrire al segretario una funzionalità per la registrazione degli istruttori nel sistema | 12 |
| RFo8 | Il sistema deve verificare se la lezione specificata sia disponibile per la prenotazione. | 14 |
| RFo9 | Il sistema deve garantire al cliente la funzionalità per confermare una lezione nel caso in cui essa sia disponibile | 15 |
| RF10 | Il sistema deve registrare la prenotazione confermata dal cliente | 16 |
| RF11 | Il sistema deve inviare per posta elettronica a un istruttore un prospetto delle sue lezioni prenotate dai clienti per quel giorno | 17 |
| RF12 | Il sistema deve inviare a ciascun cliente prenotato per un determinato giorno un promemoria con orario della lezione e nome, cognome e numero di telefono dell’istruttore | 18 |

### Requisiti sui dati

|  |  |  |
| --- | --- | --- |
| ID | Requisito | Origine (n. frase dei requisiti revisionati) |
| RD01 | I corsi offerti dalla scuola guida per il conseguimento della patenta di guida sono per motocicli e per autoveicoli. | 1 |
| RD02 | Per ogni cliente si vuole memorizzare nome, cognome, data di nascita, indirizzo di posta elettronica, residenza, numero della carta di identità, tipo di patente che si vuole conseguire e/o già possedute. | 3 |
| RD03 | Una domanda consiste in una formulazione, una risposta corretta ed una eventuale figura | 7 |
| RD04 | Ad ogni istruttore sono associati i giorni della settimana e le ore in cui è disponibile | 11 |
| RDo5 | La registrazione degli istruttori deve prevedere l’inserimento di nome, cognome, telefono e indirizzo e-mail | 13 |

### Vincoli / Altri requisiti

|  |  |  |
| --- | --- | --- |
| ID | Requisito | Origine (n. frase dei requisiti revisionati) |
| V01 | Una lezione di guida ha la durata di un’ora e inizia sempre allo scoccare dell’ora. | 10 |
| V02 | Le lezioni di guida sono erogate esclusivamente dal lunedì al venerdì. | 19 |
| V03 | Il prospetto delle lezioni deve essere inviato ogni giorno alle 07:00. | 17 |

## Modellazione dei casi d’uso

### Attori e casi d’uso

***Attori Secondari:***

* Servizio e-mail

***Attori Primari:***

* Segretario
* Cliente
* Tempo
* Servizio e-mail

***Casi d’uso:***

* UC1: Registrazione clienti
* UC2: Invio credenziali al cliente
* UC3: Simulazione prova
* UC4: Prenotazione lezione
* UC5: Inserimento istruttori
* UC6: Verifica disponibilità lezione
* UC7: Conferma lezione
* UC8: Registrazione prenotazione
* UC9: Invio prospetto
* UC10: Invio promemoria

***Casi d’ uso di inclusione:***

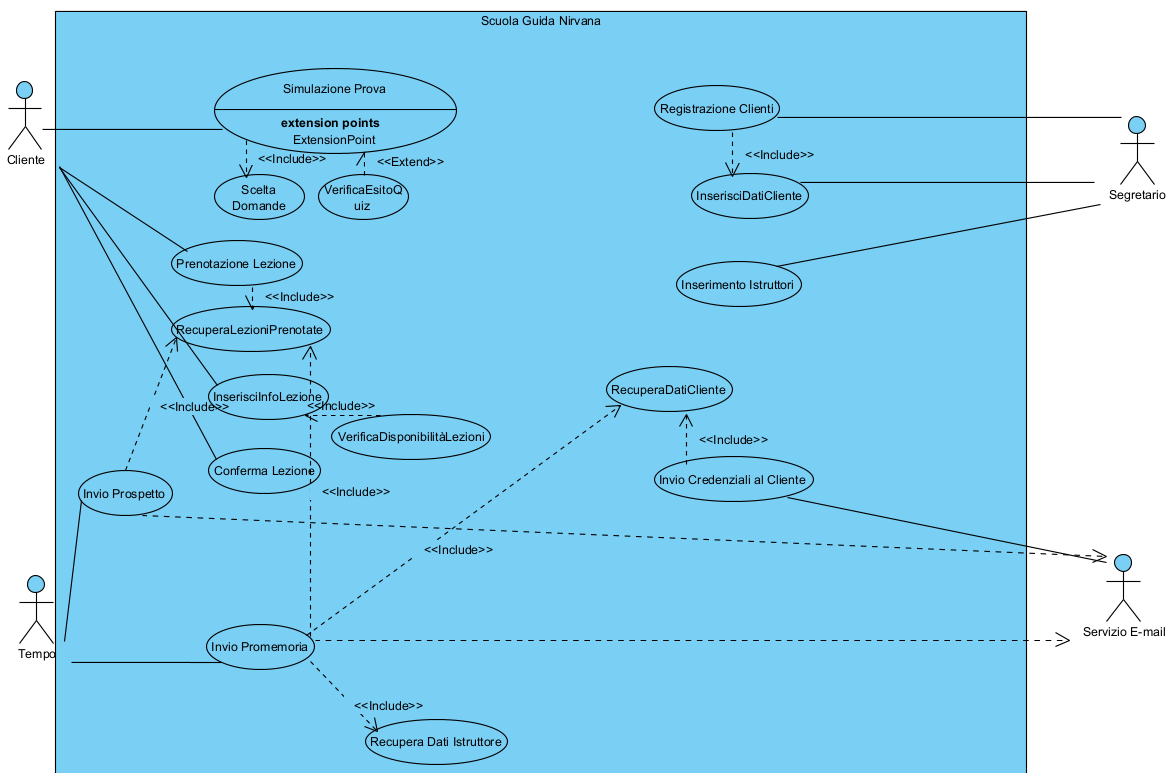
* UC11: Scelta domande
* UC12: InserisciDatiCliente
* UC13: RecuperaDatiCliente
* UC14: InserisciInfoLezione
* UC15: RecuperaLezioniPrenotate
* UC16: RecuperaDatiIstruttore

***Casi d’ uso di estensione:***

* UC17: VerificaEsitoQuiz

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Caso d’uso | Attori Primari | Attori Secondari | Incl. / Ext. | Requisiti Corrispondenti |
| UC1: Registrazione clienti | Segretario | - | Include InserisciDatiClienti | RF01 |
| UC2: Invio credenziali al cliente | Servizio E-mail | - | Include RecuperaDatiCliente | RF02 |
| UC3: Simulazione prova | Cliente | - | Estende  EsitoQuiz | RF03 |
| UC4: Prenotazione lezione | Cliente | - | Include RecuperaLezioniPrenotate | RF06 |
| UC5: Inserimento istruttori | Segretario | - | - | RF07 |
| UC6: Verifica disponibilità lezione | - | - | Include InserisciInfoLezione | RF08 |
| UC7: Conferma lezione | Cliente | - | - | RF09 |
| UC8: Registrazione prenotazione | - | - | - | RF10 |
| UC9: Invio prospetto | Tempo | Servizio e-mail | Include RecuperaLezioniPrenotate | RF11 |
| UC10: Invio promemoria | Tempo | Servizio e-mail | Include RecuperaDatiCliente  RecuperaLezioniPrenotate  RecuperaDatiIstruttore | RF12 |
| UC11: Scelta domande | - | - | Incluso in Simulazione Prova | RF03, RF04 |
| UC12: InserisciDatiCliente | Segretario | - | Incluso in Registrazione Clienti | RF01 |
| UC13: RecuperaDatiCliente | - | - | Incluso in Invio credenziali al cliente e Invio promemoria | RF02, RF12 |
| UC14: InserisciInfoLezione | Cliente | - | Incluso in Verifica disponibilità lezione | RF08 |
| UC15: RecuperaLezioniPrenotate | - | - | Incluso in Prenotazione lezione e Invio Prospetto e Invio Promemoria | RF06, RF11, RF12 |
| UC16: RecuperaDatiIstruttore | - | - | Incluso in Invio promemoria | RF12 |
| UC17: VerificaEsitoQuiz | - | - | Estende Simulazione prova | RF03, RF05 |

### Diagramma dei casi d’uso



### Scenari

|  |  |
| --- | --- |
| Caso d’uso: | Registrazione clienti |
| Attore primario | Segretario |
| Attore secondario | - |
| Descrizione | Il segretario registra i clienti all’atto dell’iscrizione |
| Pre-Condizioni | Il cliente fornisce i suoi dati |
| Sequenza di eventi  principale | 1. I dati sono forniti al segretario dal cliente 2. Il segretario accede alla piattaforma 3. Il segretario inserisce i dati del cliente nel sistema 4. Il sistema controlla i dati inseriti   4.1 Se i dati inseriti non sono validi il sistema richiede il reinserimento degli stessi  5. Il sistema conferma la validità dei dati inseriti  6. Il sistema memorizza i dati nella piattaforma |
| Post-Condizioni | Il cliente è registrato nella piattaforma |
| Casi d’uso correlati | *InserisciDatiClienti* |
| Sequenza di eventi  alternativi | - |

|  |  |
| --- | --- |
| Caso d’uso: | Invio credenziali al cliente |
| Attore primario | Servizio e-mail |
| Attore secondario | - |
| Descrizione | Il sistema invia tramite mail le credenziali auto-generate |
| Pre-Condizioni | Il cliente è registrato nel sistema dal segretario |
| Sequenza di eventi  principale | 1. Il sistema recupera le informazioni del cliente 2. Il sistema genera automaticamente le credenziali 3. Il servizio e-mail invia le credenziali al cliente |
| Post-Condizioni | - |
| Casi d’uso correlati | *Registrazione clienti* |
| Sequenza di eventi  alternativi | - |

|  |  |
| --- | --- |
| Caso d’uso: | Invio newsletter |
| Attore primario | Tempo |
| Attore secondario | Server di posta elettronica |
| Descrizione | Il primo giorno di ogni mese viene inviata la newsletter delle offerte a tutti i clienti registrati |
| Pre-Condizioni | E’ il primo giorno di un mese. |
| Sequenza di eventi  principale | 1. Il caso d’uso inizia alle ore … del primo giorno del mese. 2. Il sistema predispone un messaggio di posta elettronica con le offerte del mese 3. Il sistema predispone una lista di destinatari della newsletter, prelevando i loro indirizzi email dall’anagrafica dei clienti registrati 4. Il sistema invia il messaggio predisposto e la lista di destinatari al server di posta elettronica. 5. Il server di posta elettronica invia il messaggio ai clienti registrati destinatari. |
| Post-Condizioni | I clienti registrati ricevono la newsletter. |
| Casi d’uso correlati | *nessuno* |
| Sequenza di eventi  alternativi | Se il server di posta elettronica non è disponibile:  Punto di estensione: MailServerIndisponibile  Il sistema notifica all’amministratore un warning con il mancato invio e riprova dopo 15 minuti. |

## Modellazione dei dati

### 2.6.1 Progettazione concettuale

Riportare il diagramma E/R (Entità/Relazioni) che modella a livello concettuale i dati di interesse.

## Diagramma delle classi

Riportare il diagramma delle classi di analisi.

## Diagrammi di sequenza

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

## 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.

# 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 | **3** | **4** | **6** |
| NEIF | **4** | **5** | **7** |
| NEI | **3** | **4** | **6** |
| NEO | **7** | **10** | **15** |
| NEQ | **5** | **7** | **10** |

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

|  |  |  |  |
| --- | --- | --- | --- |
| FATTORI CORRETTIVI | | | |
| COMUNICAZIONE DATI | | |  |
| DISTRIBUZIONE ELABORAZIONE | | |  |
| PRESTAZIONI | | |  |
| UTILIZZO INTENSIVO CONFIGURAZIONE | | |  |
| FREQUENZA DELLE TRANSAZIONI | | |  |
| INSERIMENTO DATI INTERATTIVO | | |  |
| EFFICIENZA PER L’UTENTE FINALE | | |  |
| AGGIORNAMENTO INTERATTIVO | | |  |
| COMPLESSITA’ ELABORATIVA | | |  |
| RIUSABILITA’ | | |  |
| FACILITA’ INSTALLAZIONE | | |  |
| FACILITA’ GESTIONE OPERATIVA |  |  |  |
| MOLTEPLICITA’ DI SITI | | |  |
| FACILITA’ DI MODIFICA | | |  |

# Piano di test funzionale

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

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Categoria...** | **Categoria...** | **Categoria...** | **Categoria...** | **Categoria...** |
| * ... * ... * ... | * ... * ... * ... | * ... * ... * ... | * ... * ... * ... | * ... * ... * ... |

**TEST SUITE**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Descrizione** | **Classi di equivalenza coperte** | **Pre-condizioni** | **Input** | **Output Attesi** | **Post-condizioni Attese** |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

# Progettazione

## Progettazione della base di dati

### Progettazione logica

Riportare (con un diagramma) lo schema logico della base dati (modello relazionale). Indicare le chiavi primarie per ogni tabella.

Riportare anche i comandi SQL di creazione dello schema della base dati.

## 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 i casi d’uso sviluppati fino alla codifica in Java (uno per membro del gruppo).

# 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 di unità

Definire una test suite secondo il criterio di adeguatezza della copertura dei cammini di base per ciascuno dei metodi della sottosezione precedente.

## 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*)** |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |