Università degli Studi di Napoli Federico II

**Corso di Laurea in Ingegneria Informatica**

**Corso di Ingegneria del Software**

**Prof. R. Pietrantuono- A.A. 2020 - 21**

***Progetto***

Sistema Scuola Guida

Studenti: Chiara Guglielmo N46004810

Salvatore Francesco Rossetta N46004881

Luca Tornincasa N46005066

Versione Definitiva del *05/06/2022*

**Indice**

1. Specifiche informali 1

2. Analisi e specifica dei requisiti 2

2.1ANALISI NOMI VERBI.……………………………………………………………………………….2

2.2 Revisione dei requisiti 2

2.3 Glossario dei termini 4

2.4 Classificazione dei requisiti 4

2.4.1 Requisiti funzionali 5

2.4.2 Requisiti sui dati 6

2.4.3 Vincoli / Altri requisiti 6

2.5 Modellazione dei casi d’uso 5

2.5.1 Attori e casi d’uso 6

2.5.2 Diagramma dei casi d’uso 7

2.5.3 Scenari 8

2.6 Diagramma delle classi 12

2.7 Diagrammi di sequenza 13

3. Stima dei costi 18

4. Piano di test funzionale 19

5. Progettazione 36

5.1 Diagramma delle classi 36

5.2 Diagrammi di sequenza 38

6. Implementazione GITHUB

7. Testing 43

7.2 Test funzionale 43

# Specifiche informali (Traccia)

*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* *email. 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

Per individuare le classi di analisi è possibile sfruttare svariate tecniche, tra cui la più utilizzata è l’analisi nome/verbo:

**•Nomi: candidati per classi o proprietà delle classi**

**•Verbi: candidati per le responsabilità delle classi**

Una buona classe di analisi è un’astrazione ben definita se i suoi elementi rispettano il principio di massima coesione e di minimo accoppiamento.

Nomi: Cliente, Istruttore, Prenotazione, Simulazione, Domanda, Segretario.

Verbi: Prenotare, Inserire, Esercitare, Inviare, Rispondere, Registrare.

## Revisione dei requisiti

1. La scuola guida offre corsi per patenti di guida tipo AM, A1, A2, A e B.

2. Il sistema deve consentire al segretario della scuola guida di registrare i clienti all’iscrizione.

3. Di 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, ed eventuali patenti già in possesso.

4. Il sistema deve generare credenziali (nome e password) per ogni cliente registrato.

5. Il sistema deve inviare una mail al cliente con le credenziali assegnate.

6. Il sistema deve consentire al segretario della scuola guida di registrare gli istruttori.

7. Di ogni istruttore si vuole memorizzare nome, cognome, telefono e indirizzo e-mail i giorni della settimana e le ore in cui è disponibile.

8. Il sistema deve offrire la possibilità agli utenti registrati di avviare la simulazione di un test scritto.

9. La simulazione del test scritto deve essere composta da 40 domande casuali.

10. Ogni domanda è composta da tre elementi distinti: la formulazione della domanda, una risposta esatta (V/F) e una eventuale immagine.

11. Le domande devono essere mantenute in un database interno all’applicazione.

12. Il sistema, per ogni domanda, deve ricevere in ingresso dall’utente la risposta che ritiene corretta (V/F).

13. Il sistema deve controllare la correttezza di ogni domanda.

14. Il sistema deve elaborare l’esito del test al termine delle 40 domande, con esito negativo se sono presenti almeno 5 errori.

15. Il sistema deve consentire al segretario e ai clienti di loggarsi all’interno dell’applicazione

16. Il sistema consente all’utente di prenotare una lezione di guida.

17. Per ogni prenotazione bisogna specificare istruttore, data e ora in cui si vuole effettuare la lezione di guida.

18. Una lezione ha la durata di un’ora e inizia allo scoccare dell’ora.

19 Il sistema verifica se la lezione specificata è disponibile per la prenotazione.

20. Il cliente può confermare la lezione in caso di riscontro positivo della disponibilità.

21. Il sistema ogni giorno alle 07:00 invia per posta elettronica un prospetto delle lezioni prenotate dai clienti agli istruttori e a ciascun cliente prenotato per quel giorno un promemoria.

22. Il promemoria riporta orario della lezione, nome, cognome e numero di telefono dell’istruttore.

## Glossario dei termini

|  |  |  |
| --- | --- | --- |
| Termine | Descrizione | Sinonimi |
| Patente | Facoltà di esercitare la guida di un veicolo. |  |
| Credenziali | Dati assegnati per l’accesso al software | ID-Password |
| Istruttore | Dipendente dell’autoscuola che insegna in ambiente teorico e pratico | Insegnante, dipendente |
| Simulazione | Riproduzione di una prova scritta per la patente di guida | Test |
| Prenotazione | Gestione di un appuntamento per una lezione di guida |  |
| Promemoria | Notifica per ricordare la presenza di lezioni di guida |  |

## Classificazione dei requisiti

Revisionati i requisiti forniti e estratti dalle specifiche informali, abbiamo suddiviso i requisiti nelle categorie:

* **Requisiti Funzionali.**
* **Requisiti sui Dati.**

I requisiti funzionali, estrapolati dal set specificato precedentemente (cfr. 2.1), specificano **“*Cosa”***il nostro Sistema/ la nostra Applicazione dovrà fare all’atto pratico e quali funzionalità dovrà garantire.

Al contempo i Requisiti sui Dati vengono forniti per esplicitare alcune **proprietà** fondamentali che i requisiti Funzionali devono rispettare per soddisfare le esigenze dei clienti.

## 2.4.1 Requisiti funzionali

|  |  |  |
| --- | --- | --- |
| ID | Requisito | Origine (n. frase dei requisiti revisionati) |
| RF01 | Il sistema deve consentire al segretario della scuola guida di registrare i clienti all’iscrizione. | 2 |
| RF02 | Il sistema deve generare credenziali (nome e password) per ogni cliente registrato. | 4 |
| RF03 | Il sistema deve inviare una mail al cliente con le credenziali assegnate. | 5 |
| RF04 | Il sistema deve consentire al segretario della scuola guida di registrare gli istruttori. | 6 |
| RF05 | Il sistema deve offrire la possibilità agli utenti registrati di avviare la simulazione di un test scritto. | 8 |
| RF06 | La simulazione del test scritto deve essere composta da 40 domande casuali. | 9 |
| RF07 | Il sistema, per ogni domanda, deve ricevere in ingresso dall’utente la risposta che ritiene corretta (V/F). | 12 |
| RF08 | Il sistema deve controllare la correttezza di ogni domanda. | 13 |
| RF09 | Il sistema deve elaborare l’esito del test al termine delle 40 domande, con esito negativo se sono presenti almeno 5 errori. | 14 |
| RF10 | Il sistema deve consentire al segretario e ai clienti di loggarsi all’interno dell’applicazione | 15 |
| RF11 | Il sistema consente all’utente di prenotare una lezione di guida. | 16 |
| RF12 | Il sistema verifica se la lezione specificata è disponibile per la prenotazione. | 19 |
| RF13 | Il sistema ogni giorno alle 07:00 invia per posta elettronica un prospetto delle lezioni prenotate dai clienti agli istruttori e a ciascun cliente prenotato per quel giorno un promemoria. | 21 |

## 

## Requisiti sui dati

|  |  |  |
| --- | --- | --- |
| ID | Requisito | Origine (n. frase dei requisiti revisionati) |
| RD01 | La scuola guida offre corsi per patenti di guida tipo AM, A1, A2, A e B. | 1 |
| RD02 | Di 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, ed eventuali patenti già in possesso*.* | 3 |
| RD03 | Di ogni istruttore si vuole memorizzare nome, cognome, telefono e indirizzo e-mail i giorni della settimana e le ore in cui è disponibile. | 7 |
| RD04 | Ogni domanda è composta da tre elementi distinti: la formulazione della domanda, una risposta esatta (V/F) e una eventuale immagine | 10 |
| RD05 | Per ogni prenotazione bisogna specificare istruttore, data e ora in cui si vuole effettuare la lezione di guida. | 17 |
| RD05 | Una lezione ha la durata di un’ora e inizia allo scoccare dell’ora. | 18 |
| RD06 | Il promemoria riporta orario della lezione, nome, cognome e numero di telefono dell’istruttore. | 22 |

## Vincoli / Altri requisiti

RNF01: Le domande di simulazione del test scritto devono essere mantenute in un database all’interno dell’applicazione.

RNF02: Per l’invio mensile della mail con le credenziali, deve essere disponibile un server di posta elettronica esterno al sistema.

RNF03: Per prenotare una lezione con un dato istruttore, ad una specifica ora di uno specifico giorno bisogna che tale lezione sia disponibile.

## Modellazione dei casi d’uso

## 2.5.1Attori e casi d’uso

***Attori Primari:***

* Cliente
* Istruttore
* Tempo
* Segretario

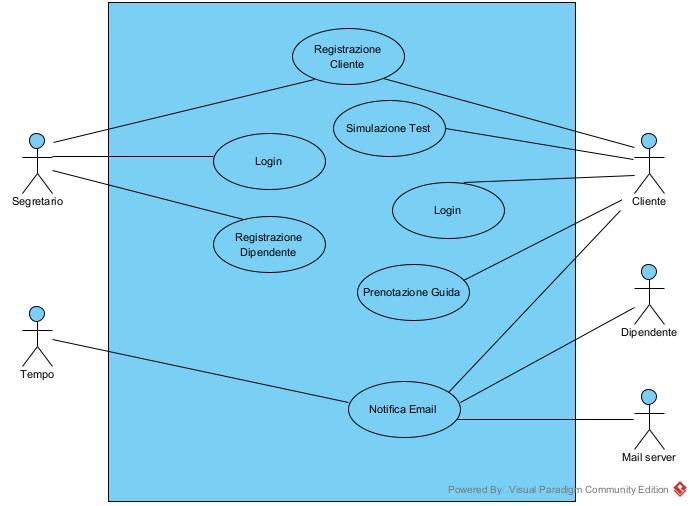
***Attori Secondari:***

* Cliente
* Istruttore

***Casi d’uso:***

* Registrazione utente
* Registrazione istruttore
* Simulazione test
* Invio notifica
* Prenotazione

## 2.5.2. Diagramma dei casi d’uso



|  |  |  |  |
| --- | --- | --- | --- |
| Caso d’uso | Attori Primari | Attori Secondari | Incl. / Ext. |
| Registrazione cliente | Segretario | Cliente |  |
| Simulazione test | Cliente |  |  |
| Notifica email | Tempo | Cliente/dipendente |  |
| Prenotazione guida | Cliente |  |  |
| Registrazione dipendente | Segretario | Dipendente |  |

## 

## 2.5.3 Scenari

|  |  |
| --- | --- |
| **Caso d’uso:** | **Registrazione utente** |
| **Attore primario** | Segretario |
| **Attore secondario** | Utente |
| **Descrizione** | Un utente viene registrato nel sistema dal segretario |
| **Pre-Condizioni** | L’utente non dev’essere già registrato nel sistema |
| **Sequenza di eventi**  **principale** | 1. Un nuovo utente chiede al segretario di registrarsi nel sistema 2. Il segretario inserisce le informazioni dell’utente nel sistema 3. Il sistema verifica la correttezza delle informazioni inserite 4. Il sistema registra il nuovo utente generando delle credenziali 5. Il sistema invia le credenziali via e-mail all’utente |
| **Post-Condizioni** | L’utente è stato registrato e ha ricevuto una mail con le credenziali |
| **Casi d’uso correlati** | *Nessuno* |
| **Sequenza di eventi alternativi** | 3.a Le informazioni inserite nel sistema non sono corrette  Il sistema segnala l’errore e chiede di modificare le informazioni non valide |

|  |  |
| --- | --- |
| **Caso d’uso:** | **Registrazione istruttore** |
| **Attore primario** | Segretario |
| **Attore secondario** | *Nessuno* |
| **Descrizione** | Un dipendente viene registrato nel sistema dal segretario |
| **Pre-Condizioni** | Il dipendente non dev’essere già registrato nel sistema |
| **Sequenza di eventi**  **principale** | 1. Un nuovo istruttore chiede al segretario di registrarsi nel sistema 2. Il segretario inserisce le informazioni dell’istruttore nel sistema 3. Il sistema verifica la correttezza delle informazioni 4. L’istruttore è stato correttamente registrato nel sistema |
| **Post-Condizioni** | Il dipendente è stato registrato correttamente nel sistema |
| **Casi d’uso correlati** | *Nessuno* |
| **Sequenza di eventi**  **alternativi** | 3.a Le informazioni inserite nel sistema non sono corrette  Il sistema segnala l’errore e chiede di modificare le informazioni non valide |

|  |  |
| --- | --- |
| **Caso d’uso:** | **Simulazione test scritto** |
| **Attore primario** | Utente |
| **Attore secondario** | *Nessuno* |
| **Descrizione** | Un utente registrato effettua una simulazione di un test scritto |
| **Pre-Condizioni** | L’utente è registrato e ha effettuato il login |
| **Sequenza di eventi**  **principale** | 1. L’utente avvia una simulazione del test scritto 2. Il sistema elabora 40 domande casuali da sottoporre all’utente 3. L’utente fornisce la propria risposta ad ogni domanda 4. Il sistema elabora il risultato della simulazione 5. Il sistema restituisce il risultato del test all’utente |
| **Post-Condizioni** | L’utente ha ricevuto l’esito per la simulazione |
| **Casi d’uso correlati** | *Nessuno* |
| **Sequenza di eventi**  **alternativi** | *Nessuna* |

|  |  |
| --- | --- |
| Caso d’uso: | Prenotazione |
| Attore primario | Cliente |
| Attore secondario | *nessuno* |
| Descrizione | Il cliente prenota una lezione di guida con l’istruttore. |
| Pre-Condizioni | Il cliente deve essere registrato.  L’istruttore deve essere registrato.  L’istruttore deve essere disponibile quel giorno a quell’orario.  La lezione inizia allo scoccare dell’ora e dura 1 ora. |
| Sequenza di eventi  principale | 1. Il caso d’uso inizia quando l’utente vuole prenotare la lezione. 2. Il sistema predispone la richiesta dell’identità dell’istruttore, l’orario e la data in cui ci si vuole prenotare. 3. Il cliente immette gli input per queste richieste. 4. Il sistema verifica la disponibilità dell’istruttore per quel giorno e quella data e chiede conferma al cliente. 5. Il cliente conferma la prenotazione. |
| Post-Condizioni | Il cliente prenota una lezione di guida. |
| Casi d’uso correlati | *Nessuno* |
| Sequenza di eventi  alternativi | Se l’istruttore non è disponibile:  al punto 5 la prenotazione viene annullata informando il cliente dell’indisponibilità. |

|  |  |
| --- | --- |
| Caso d’uso: | Invio notifica |
| Attore primario | Tempo |
| Attore secondario | *Nessuno* |
| Descrizione | Ogni mattina alle ore 7:00 viene inviata una mail ai clienti che hanno prenotato una guida per quel giorno e agli istruttori con gli appuntamenti odierni. |
| Pre-Condizioni | Devono essere le ore 7:00 di ogni mattina. |
| Sequenza di eventi  principale | 1. Il caso d’uso inizia alle ore 7 di ogni giorno. 2. Il sistema predispone un messaggio di posta elettronica per i clienti con un promemoria sull’appuntamento. 3. Il sistema predispone una lista degli appuntamenti giornalieri per tutti gli istruttori. 4. I clienti e gli istruttori ricevono i promemoria. |
| Post-Condizioni | I clienti registrati e gli istruttori ricevono il promemoria e il prospetto degli appuntamenti. |
| Casi d’uso correlati | *Nessuno* |
| Sequenza di eventi  alternativi | *Nessuno* |

|  |  |
| --- | --- |
| Caso d’uso: | Login Segretario/Cliente |
| Attore primario | Segretario/Cliente |
| Attore secondario | *Nessuno* |
| Descrizione | Il Segretario/Cliente inserisce nell’applicazione l’username e la password per accedere alle funzionalità relative. |
| Pre-Condizioni | Il Segretario/Cliente deve conoscere le credenziali. |
| Sequenza di eventi  principale | 1. L’attore inserisce l’username 2. L’attore inserisce la password 3. Il sistema verifica la correttezza delle credenziali inserite. 4. Se, le credenziali sono corrette l’attore può accedere alle funzionalità.   4.1 Altrimenti il sistema rifiuta l’accesso. |
| Post-Condizioni | I clienti registrati e il segretario possono accedere alle loro funzionalità |
| Casi d’uso correlati | *Nessuno* |
| Sequenza di eventi  alternativi | Altrimenti il sistema rifiuta l’accesso per quelle credenziali. |

## 2.6 Diagramma delle classi

## 

## Diagrammi di sequenza

Diagramma di sequenza per l’inserimento di un **nuovo dipendente:**

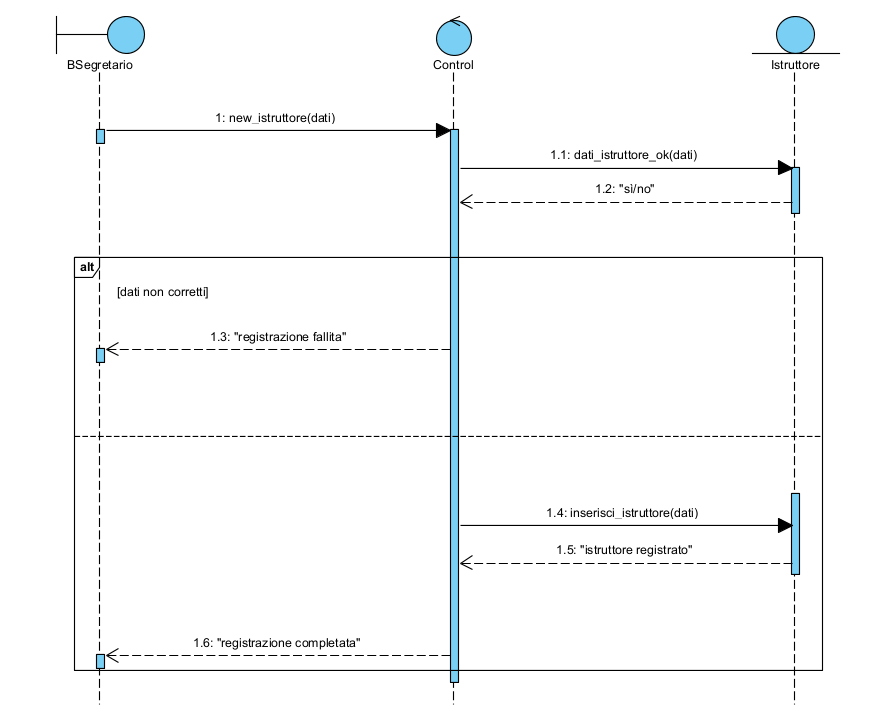


Diagramma per la **prenotazione** di una lezione pratica:

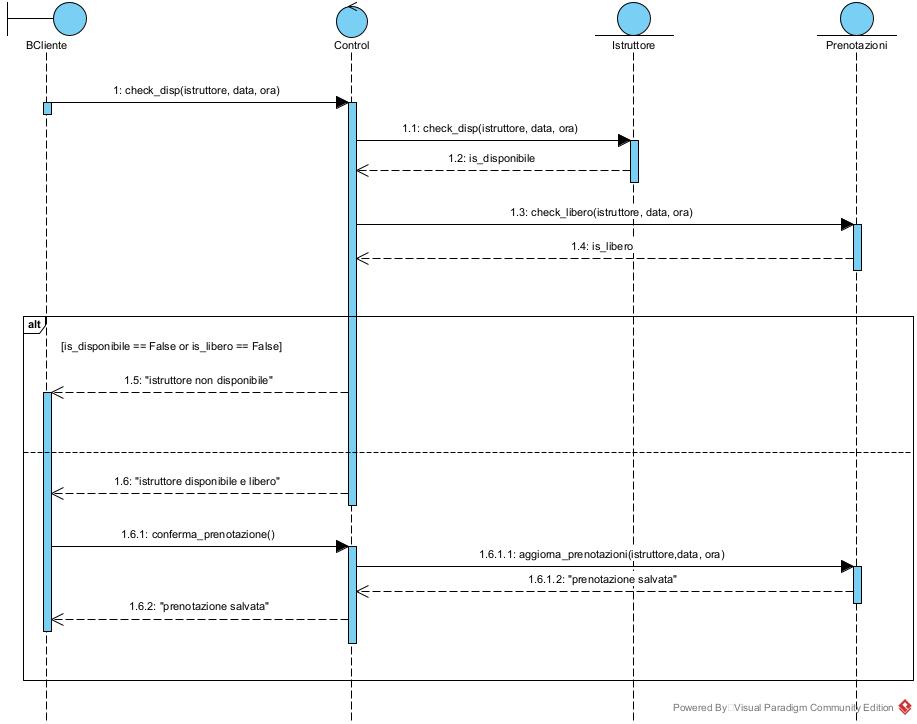


Diagramma di sequenza per un **nuovo cliente:**

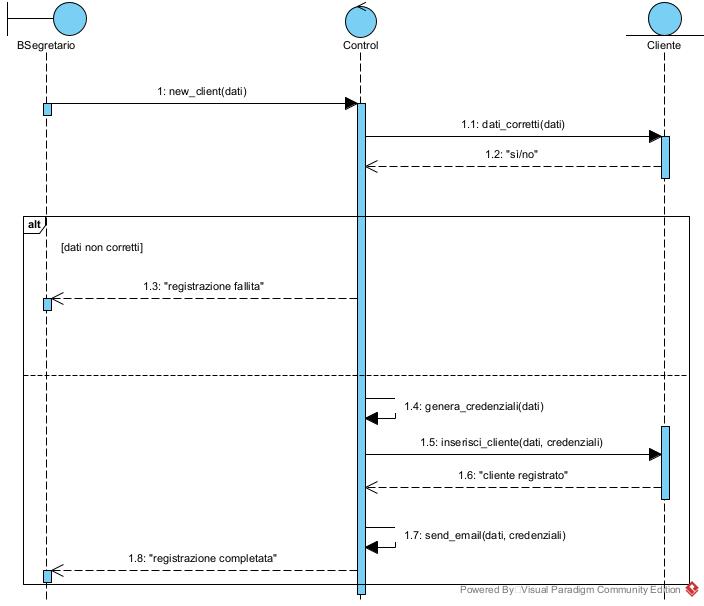


Diagramma di sequenza per la **simulazione**:

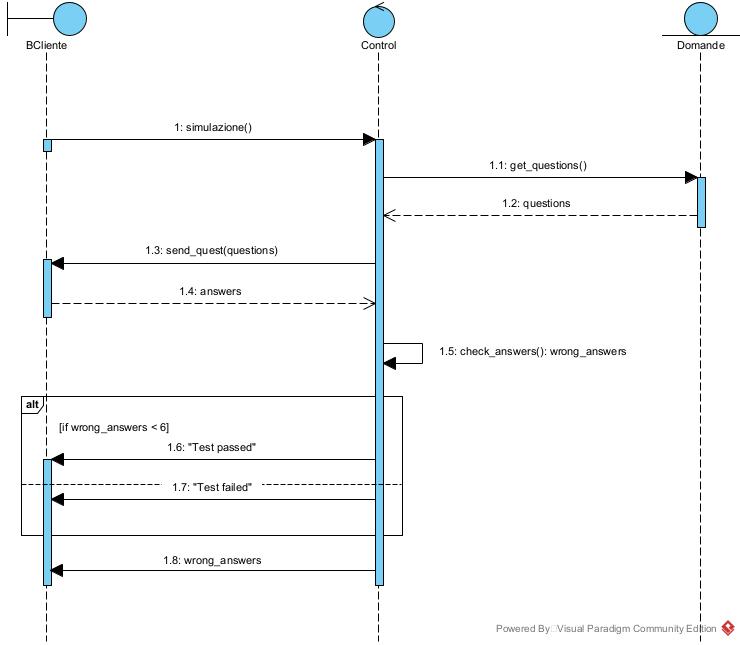
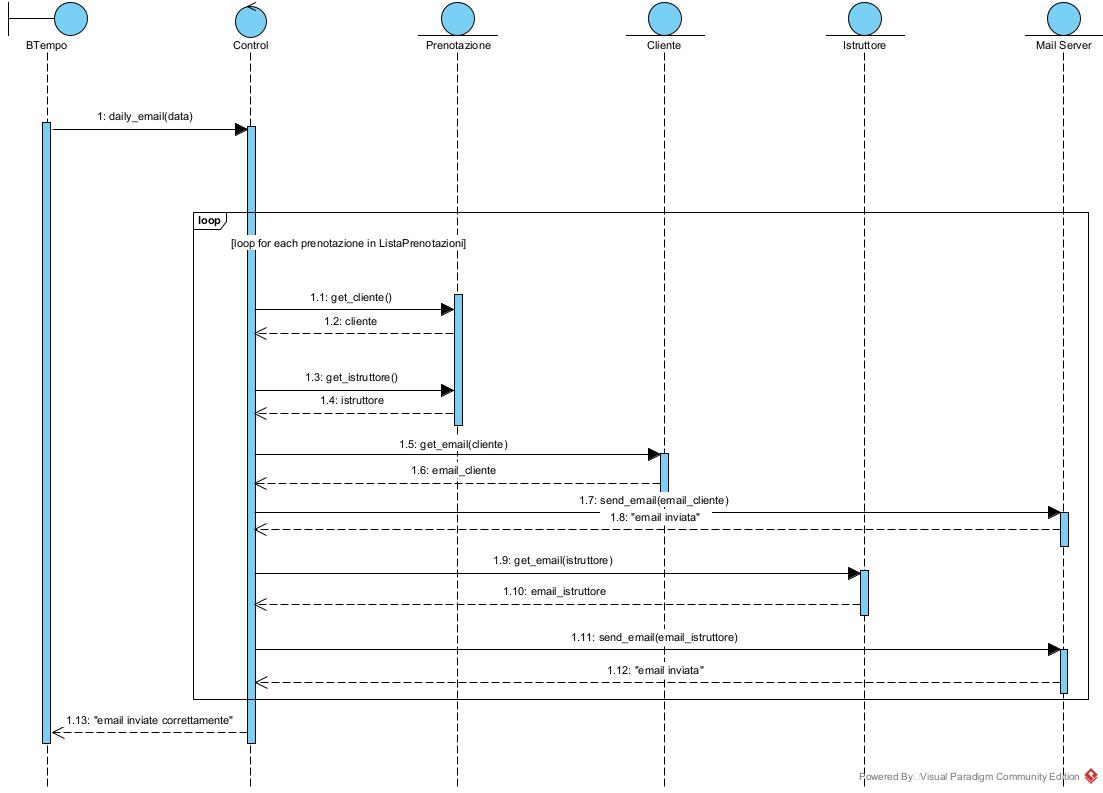
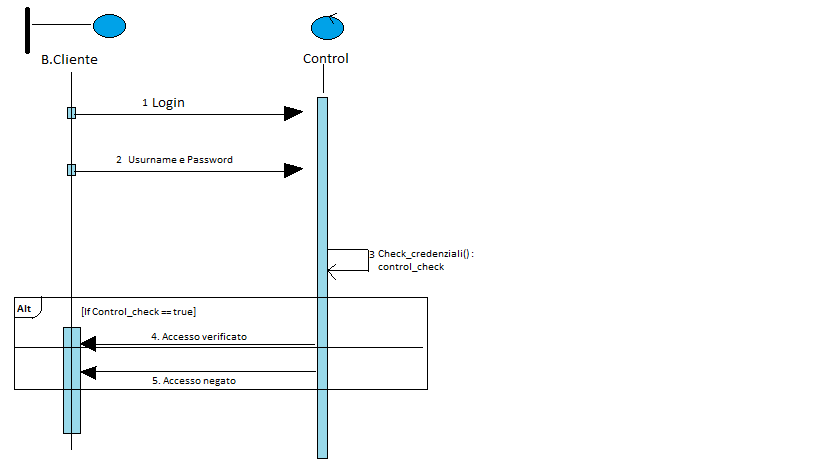


Diagramma di sequenza per **l’invio delle mail:**



**Diagramma di sequenza per il login:**



# Stima dei costi

La stima dei costi prende in esame le principali funzionalità del software da sviluppare, ossia:

* Registrazione di un nuovo cliente
* Registrazione di un nuovo dipendente
* Simulazione di un esame scritto
* Prenotazione per una lezione pratica

Di seguito proponiamo una trattazione di ogni funzionalità e di calcolo dei rispettivi parametri per la stima dei costi di ognuna. Infine, calcoleremo la stima dei costi totale sommando i valori ottenuti.

**Registrazione di un nuovo cliente**

Per la registrazione di un nuovo cliente le informazioni inserite in input sono le seguenti:

nome, cognome, email, data di nascita, ID, residenza, patente, altre patenti già in possesso.

Inoltre, verranno generate delle credenziali di username e password dal sistema che verranno poi inviate via mail al nuovo cliente. Pertanto, i valori di Data Element Type (DET), di External Input (EI) ed External Output (EO) sono i seguenti:

DET = 10, di cui EI = 8 e EO = 2

**Registrazione di un nuovo dipendente**

Per la registrazione di un nuovo cliente le informazioni inserite in input sono le seguenti: nome, cognome, email, telefono, ID, disponibilità.

Calcoliamo quindi i valori di Internal Data Element Type (DET) ed External Input (EI) come segue:

DET = 6, di cui EI = 6

**Simulazione di un test scritto**

Per questa funzionalità il sistema dovrà estrarre dal database un certo numero di domande, definite da un testo, risposta e un eventuale immagine legata alla domanda.

Inoltre, l’utente dovrà inserire un input ad ogni domanda che gli viene proposta, ossia la sua risposta, che verrà poi controllata dal sistema. Pertanto, scrivo:

DET = 3, di cui EI = 1, EO = 2

**Prenotazione lezione pratica**

Per una prenotazione il cliente dovrà inserire ID dell’istruttore, giorno e orario in cui sostenere la lezione.

A questo punto il sistema dovrà effettuare delle query sul database per recuperare l’eventuale disponibilità dell’istruttore e, in caso affermativo, chiedere all’utente se desidera prenotare la lezione e l’utente potrà confermare.

DET = 4, di cui EI = 3, EO = 1, EQ = 1

In seguito sarà inviata una mail al cliente con informazioni sull’istruttore (nome, cognome e telefono) e sulla lezione (orario).

DET = 4, EQ = 1

Inoltre, verrà inviata una mail all’istruttore con informazioni su sul cliente (nome, cognome, patente).

DET = 3, EQ = 1

**Calcolo UFP**

Dopo aver calcolato i paraemtri di ogni funzionalità, posso calcolare il peso degli ILF.

Nel nostro specifico caso, per tutte le funzionalità ottengo un peso *basso* che mi permette di assegnare un valore di 7. Quindi, tutti e quattro gli ILF in totale avranno un peso di

ILF = 7 + 7 + 7 + 7 = 28.

Totale EI = 4

1. Per la registrazione cliente DET = 8, FTR = 1. Il peso sarà quindi *basso*.
2. Per la registrazione dipendente DET = 6, FTR = 1. Il peso sarà quindi *basso*.
3. Per la simulazione del test DET = 1, FTR = 1. Il peso sarà quindi *basso*.
4. Per la prenotazione della lezione DET = 4, FTR = 3. Il peso sarà quindi *medio.*
5. Per la conferma della prenotazione DET = 1, FTR = 1. Il peso sarà quindi *basso*.

Il peso totale sarà quindi pari a

Peso EI = 3 + 3 + 3 + 4 + 3 = 16.

Totale EO = 4

1. Per le credenziali utente ho DET = 2, FTR = 1
2. Calcolo correttezza domanda avrà DET = 1, FTR = 1
3. Per il calcolo correttezza del test avrò DET = 1, FTR = 0
4. Per la conferma della prenotazione ho DET = 1, FTR = 1

Tutti gli EO avranno peso basso, quindi posso calcolare

Peso EO = 4 + 4 + 4 + 4 = 16.

Totale EQ = 3

1. Per il riepilogo delle prenotazioni dell’istruttore ho DET = 3, FTR = 3.
2. Per il riepilogo delle prenotazioni del cliente ho DET = 4, FTR = 3.
3. Per la verifica della disponibilità della lezione ho DET = 1, FTR = 1.

Tutti gli EQ sono di livello *basso* e perciò avranno peso individuale pari a 3.

Peso EQ = 3 + 3 + 3 = 9.

Posso quindi calcolare la stima dei costi come somma di tutti i pesi trovati precedentemente.

UFP = 28 + 16 + 16 + 9 = 69.

* 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 | | | 0 |
| DISTRIBUZIONE ELABORAZIONE | | | 0 |
| PRESTAZIONI | | | 0 |
| UTILIZZO INTENSIVO CONFIGURAZIONE | | | 0 |
| FREQUENZA DELLE TRANSAZIONI | | | 0 |
| INSERIMENTO DATI INTERATTIVO | | | 0 |
| EFFICIENZA PER L’UTENTE FINALE | | | 0 |
| AGGIORNAMENTO INTERATTIVO | | | 0 |
| COMPLESSITA’ ELABORATIVA | | | 0 |
| RIUSABILITA’ | | | 0 |
| FACILITA’ INSTALLAZIONE | | | 0 |
| FACILITA’ GESTIONE OPERATIVA |  |  | 0 |
| MOLTEPLICITA’ DI SITI | | | 0 |
| FACILITA’ DI MODIFICA | | | 0 |

# Piano di test funzionale

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

La funzionalità “***Registrazione cliente***” è prevista per il segretario che, dopo essersi autenticato, inserisce alcuni dati nei rispettivi campi. Gli input sono tutti stringhe che devono essere necessariamente non nulle, ad eccezione della patente che si vuole conseguire, che è un insieme di valori, tra cui il cliente deve scegliere e delle altre patenti già conseguite, che può essere null. In questo caso il sistema registrerà che l’utente non ha patenti precedentemente già conseguite.

Il numero di test è dovuto ad un calcolo combinatorio fornito come segue:

**(N° Error + N° Single) + (N° Input Rimanenti per ogni Categoria moltiplicati fra loro)**

Nel nostro caso:

**(10 + 5) + (1\*1\*2\*1\*1\*1\*1\*1) = 16**

**CATEGORIE**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **NOME** | **COGNOME** | **EMAIL** | **ID** | **DATA DI NASCITA** | **RESIDENZA** | **PATENTE** | **ALTRE PATENTI** |
| * Null [error] * stringhe di soli caratteri | * Null [error] * stringhe di sole caratteri | * null [error] * email formato corretto * email supera lunghezza max [error] | * Null [error] * lunghezza non idonea [error] * caratteri alfanumerici | * null [error] * formato DD/MM/YYYY * caratteri [error] | * null [error] * stringa oltre lunghezza max [error] * stringa in lunghezza | * AM [single] * A1 [single] * A2 [single] * A [single] * B [single] | * Stringa, di default null |

**TEST SUITE**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Classi di equivalenza coperte** | **Descrizione** | **Pre-condizioni** | **Input** | **Output Attesi** | **Post-condizioni Attese** |
| 1 | Email troppo lunga | Email supera lunghezza mail | nessuna | Email = “provaprovaprovaprova@provaprovaprovaprova.it” | Messaggio di errore “Email non valida” | Il sistema torna sulla schermata di registrazione |
| 2 | Email null | La mail è una nulla | nessuna | Email = “ “ | Messaggio di errore “Email non valida” | Il sistema torna sulla schermata di registrazione |
| 3 | ID lunghezza non idonea | L’id non ha una lunghezza esatta di 7 caratteri | nessuna | Id = “aaaaa” | Messaggio di errore “ID non valido” | Il sistema torna sulla schermata di registrazione |
| 4 | Nome null | Il nome inserito è nullo | nessuna | Nome = “ “ | Messaggio di errore “nome non valido” | Il sistema torna sulla schermata di registrazione |
| 5 | Cognome null | Il cognome inserito è nullo | nessuna | Cognome = “ “ | Messaggio di errore “cognome non valido” | Il sistema torna sulla schermata di registrazione |
| 6 | ID null | L’id inserito è nullo | nessuna | Id = “ “ | Messaggio di errore “ID non valido” | Il sistema torna sulla schermata di registrazione |
| 7 | Data null | La data inserita è nulla | nessuna | Data = “ “ | Messaggio di errore “data non valida” | Il sistema torna sulla schermata di registrazione |
| 8 | Residenza null | La residenza inserita è nulla | nessuna | Residenza = “ “ | Messaggio di errore “residenza non valido” | Il sistema torna sulla schermata di registrazione |
| 9 | Le informazioni inserite sono corrette | Nome, cognome, id, email, residenza, patente, altre patenti = “…”  Data = “dd/mm/yyyy” | nessuna | Nome = “ab”  Cognome = “cd”  Email = [a@b.it](mailto:a@b.it)  Id = “1234567”  Residenza = “Via abc”  Data = “01/01/2000”  Patente = “A1”  Altre patenti = “AM” | Conferma di informazioni corrette | Il cliente è stato aggiunto al database |
| 10 | Patente non valida | Patente: AM, Altre patenti: AM | nessuna | Patente = “x”  Altre patenti = “x” |  | Il sistema torna sulla schermata di registrazione |
| 11 | Data di nascita non valida | Data di nascita con caratteri | nessuna | Data di nascita = “abc” | Messaggio di errore “data non valida” | Il sistema torna sulla schermata di registrazione |
| 12 | Patente A | Patente inserita A | Nessuna | Patente =a; | Conferma di informazioni corrette | Il cliente è stato aggiunto al database |
| 13 | Patente B | Patente inserita B | Nessuna | Patente =B | Conferma di informazioni corrette | Il cliente è stato aggiunto al database |
| 14 | Patente A1 | Patente inserita A1 | Nessuna | Patente =A1 | Conferma di informazioni corrette | Il cliente è stato aggiunto al database |
| 15 | Patente AM | Patente inserita AM | Nessuna | Patente =AM; | Conferma di informazioni corrette | Il cliente è stato aggiunto al database |
| 16 | Patente A2 | Patente inserita A2 | Nessuna | Patente =a2; | Conferma di informazioni corrette | Il cliente è stato aggiunto al database |

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

La funzionalità “***Registrazione dipendente***” è prevista per il segretario che, dopo essersi autenticato, inserisce alcuni dati nei rispettivi campi. Gli input sono tutti stringhe che devono essere necessariamente non nulle, inoltre le tre disponibilità del nuovo dipendente devono essere necessariamente diverse tra di loro. Non è possibile, infatti, registrare un dipendente con due disponibilità uguali.

Il numero di test è dovuto ad un calcolo combinatorio fornito come segue:

**(N° Error + N° Single) + (N° Input Rimanenti per ogni Categoria moltiplicati fra loro)**

Nel nostro caso:

**(18 + 0) + (1\*1\*1\*1\*1\*1\*1\*1) = 19**

Per semplicità didattica illustreremo solo parte dei test case che comporranno la test suite.

**CATEGORIE**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **NOME** | **COGNOME** | **EMAIL** | **ID** | **TELEFONO** | **DISP1** | **DISP2** | **DISP3** |
| * Null [error] * stringhe di soli caratteri | * null [error] * stringhe di sole caratteri | * null [error] * email corretta * email supera lunghezza max [error] | * Null [error] * lunghezza non idonea [error] * caratteri alfanumerici | * null [error] * lunghezza non idonea [error] * caratteri [error] * 10 cifre | * Null [error] * DISP2 [error] * DISP3 [error] * Stringa diversa dalle altre due disponibilità | * Null [error] * DISP1 [error] * DISP3 [error] * Stringa diversa dalle altre due disponibilità | * Null [error] * DISP1 [error] * DISP2 [error] * Stringa diversa dalle altre due disponibilità |

**TEST SUITE**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Classi di equivalenza coperte** | **Descrizione** | **Pre-condizioni** | **Input** | **Output Attesi** | **Post-condizioni Attese** |
| 1 | Email troppo lunga | Email supera lunghezza mail | nessuna | Email = “provaprovaprovaprova@provaprovaprovaprova.it” | Messaggio di errore “Email non valida” | Il sistema torna sulla schermata di registrazione |
| 2 | Email null | La mail è una nulla | nessuna | Email = “ “ | Messaggio di errore “Email non valida” | Il sistema torna sulla schermata di registrazione |
| 3 | ID lunghezza non idonea | L’id non ha una lunghezza esatta di 7 caratteri | nessuna | Id = “aaaaa” | Messaggio di errore “ID non valido” | Il sistema torna sulla schermata di registrazione |
| 4 | Nome null | Il nome inserito è nullo | nessuna | Nome = “ “ | Messaggio di errore “nome non valido” | Il sistema torna sulla schermata di registrazione |
| 5 | Cognome null | Il cognome inserito è nullo | nessuna | Cognome = “ “ | Messaggio di errore “cognome non valido” | Il sistema torna sulla schermata di registrazione |
| 6 | ID null | L’id inserito è nullo | nessuna | Id = “ “ | Messaggio di errore “ID non valido” | Il sistema torna sulla schermata di registrazione |
| 7 | Telefono null | Il telefono inserito è nullo | nessuna | telefono = “ “ | Messaggio di errore “telefono non valido” | Il sistema torna sulla schermata di registrazione |
| 8 | Telefono lunghezza non idonea | Il telefono inserito è troppo lungo | nessuna | telefono = “ 12323231221“ | Messaggio di errore “telefono non valido” | Il sistema torna sulla schermata di registrazione |
| 8 | Le informazioni inserite sono corrette | Nome, cognome, id, telefono, disp1, disp2, disp3 = “…” | nessuna | Nome = “ab”  Cognome = “cd”  Email = [a@b.it](mailto:a@b.it)  Id = “1234567890”  Disp1 = “LUN 15”  Disp2 = “LUN 16”  Disp3 = “MER 9” | Conferma di informazioni corrette | Il cliente è stato aggiunto al database |
| 9 | Disponibilità non valide | DISP1 = DISP2  DISP1 = DISP3  DISP2 = DISP3 | nessuna | Disp1 = Disp2  Disp1 = Disp3  Disp2 = Disp3 | Messaggio di errore sulle disponibilità | Il sistema torna sulla schermata di registrazione |
| 11 | Disponibilità nulle | Disp null | nessuna | Disp1 = null  Disp2 = null  Disp3 = null | Messaggio di errore sulle disponibilità | Il sistema torna sulla schermata di registrazione |

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

La funzionalità “***Prenotazione***” è prevista per il cliente che, dopo essersi autenticato, inserisce alcuni dati nei rispettivi campi. L’input è una stringa che deve essere necessariamente non nulla, inoltre il giorno e l’orario della prenotazione devono essere necessariamente scelti fra le disponibilità del dipendente. Non è possibile, infatti, prenotarsi in un giorno in cui un dipendente non risulta disponibile.

Il numero di test è dovuto ad un calcolo combinatorio fornito come segue:

**(N° Error + N° Single) + (N° Input Rimanenti per ogni Categoria moltiplicati fra loro)**

Nel nostro caso:

**(6 + 0) + (1\*1\*1\*1) = 7**

**CATEGORIE**

|  |  |  |  |
| --- | --- | --- | --- |
| **ORARIO** | **ID ISTRUTTORE** | **DATA PRENOTAZIONE** | **USERNAME CLIENTE** |
| * Null [error] * Out of range (>24) [error] * Formato corretto (0/24) | * Null [error] * caratteri alfanumerici * Non esistente [error] | * null [error] * formato DD/MM/YYYY | * null [error] * stringa di caratteri |

**TEST SUITE**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Descrizione** | **Classi di equivalenza coperte** | **Pre-condizioni** | **Input** | **Output Attesi** | **Post-condizioni Attese** |
| 1 | L’id inserito è nullo | id null | nessuna | ID= “ “ | Messaggio di errore “ID non valido” | Il sistema torna sulla schermata di prenotazione |
| 2 | L’username inserito è nullo | Username null | nessuna | Username=” “ | Messaggio di errore “Username non valido” | Il sistema torna sulla schermata di prenotazione |
| 3 | Informazioni inserite correttamente | Id istruttore = caratteri alfanumerici  Username = stringa alfanumerica  Data = “DD/MM/YYYY” | nessuna | Id istruttore = “…”  Username = “…”  Data = “DD/MM/YYYY”  Giorno = “…” | La prenotazione è inserita correttamente | La prenotazione è inserita correttamente nel database |
| 4 | La data inserita è nulla | Data null | nessuna | Data = “ “ | Messaggio di errore “data non valida” | Il sistema torna sulla schermata di prenotazione |
| 5 | Orario non rientra nel range corretto | Orario out of range | Nessuna | Orario = “35” | Messaggio di errore “orario non valido” | Il sistema torna sulla schermata di prenotazione |
| 6 | Orario inserito è nullo | Orario null | Nessuna | Orario = “ ” | Messaggio di errore “orario non valido” | Il sistema torna sulla schermata di prenotazione |
| 7 | L’id inserito non è presente nel db. | id sbagliato | nessuna | ID= “sbagliato “ | // | Il sistema non procede alla ricerca dell’istruttore. |

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

La funzionalità “***Simulazione***” è prevista per il cliente che, dopo essersi autenticato, inserisce alcuni dati nei rispettivi campi. L’input è una risposta tra Vero o Falso e Prossima domanda. Non è possibile, infatti, rispondere in altro modo o il sistema riproporrà la stessa domanda.

Il numero di test è dovuto ad un calcolo combinatorio fornito come segue:

**(N° Error + N° Single) + (N° Input Rimanenti per ogni Categoria moltiplicati fra loro)**

Nel nostro caso:

**(1 + 3) + (0) = 4**

**CATEGORIA**

|  |
| --- |
| **RISPOSTA** |
| * Vera [single] * Falsa [single] * Stringa diversa da “Vero” o “falso” [error] * Non data [single] |

**TEST SUITE**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Descrizione** | **Classi di equivalenza coperte** | **Pre-condizioni** | **Input** | **Output Attesi** | **Post-condizioni Attese** |
| 1 | Risposta inserita | Risposta: Vera | nessuna | Risposta = “0” | Schermata prossima domanda | input inserito correttamente e prossima domanda mostrata |
| 2 | Risposta inserita | Risposta: Falsa | nessuna | Risposta = “1” | Schermata prossima domanda | input inserito correttamente e prossima domanda mostrata |
| 3 | Risposta non inserita | La risposta non è stata inserita: verrà assunta come non data [single] | nessuna | Risposta = “2” | Schermata prossima domanda | input inserito e prossima domanda mostrata |
| 4 | Input inserito non valido | Risposta: diversa da vero o falso | nessuna | Risposta=”abc” | Messaggio di errore | Viene mostrata di nuovo la stessa domanda |

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

Il cliente effettua il **“*Login”*** inserendo le sue credenziali. Se i due campi sono lasciati null si avrà un errore e si tornerà alla schermata precedente. Se le credenziali sono inserite, deve esserne controllata la correttezza. In caso di esito positivo, verrà mostrata la schermata di home page del cliente, altrimenti verrà mostrato un messaggio di errore che inviterà il cliente ad inserire nuovamente le credenziali.

Il numero di test è dovuto ad un calcolo combinatorio fornito come segue:

**(N° Error + N° Single) + (N° Input Rimanenti per ogni Categoria moltiplicati fra loro)**

Nel nostro caso:

**(2 + 0) + (2) = 4**

**CATEGORIE**

|  |  |
| --- | --- |
| **Username** | **Password** |
| * Null [error] * Stringa di caratteri | * Null [error] * Stringa di caratteri |

**TEST SUITE**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Descrizione** | **Classi di equivalenza coperte** | **Pre-condizioni** | **Input** | **Output Attesi** | **Post-condizioni Attese** |
| 1 | L’username inserito è nullo | Username null | Nessuna | Username = “ “ | Messaggio di errore “Username non valido” | Rimane la schermata del login |
| 2 | La password inserita è nullo | Password null | Nessuna | Password = “ “ | Messaggio di errore “Password non valida” | Rimane la schermata del login |
| 3 | Credenziali inserite correttamente | Username = stringa  Password = stringa | Nessuna | Username = “…”  Password = “…” | Messaggio di conferma | Si apre la schermata dell’home page |
| 4 | Credenziali inserite ma non corrette | Username = stringa  Password = stringa | Nessuna | Username = “…”  Password = “…” | Messaggio di errore “credenziali non corrette” | Rimane la schermata del login |

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

Il segretario effettua il “***Login”*** inserendo le sue credenziali. Se i due campi sono lasciati null si avrà un errore e si tornerà alla schermata precedente. Se le credenziali sono inserite, deve esserne controllata la correttezza. In caso di esito positivo, verrà mostrata la schermata di home page del segretario, altrimenti verrà mostrato un messaggio di errore che inviterà il segretario ad inserire nuovamente le credenziali.

Il numero di test è dovuto ad un calcolo combinatorio fornito come segue:

**(N° Error + N° Single) + (N° Input Rimanenti per ogni Categoria moltiplicati fra loro)**

Nel nostro caso:

**(2 + 0) + (2) = 4**

**Categorie**

|  |  |
| --- | --- |
| **Username** | **Password** |
| * Null [error] * Stringa di caratteri | * Null [error] * Stringa di caratteri |

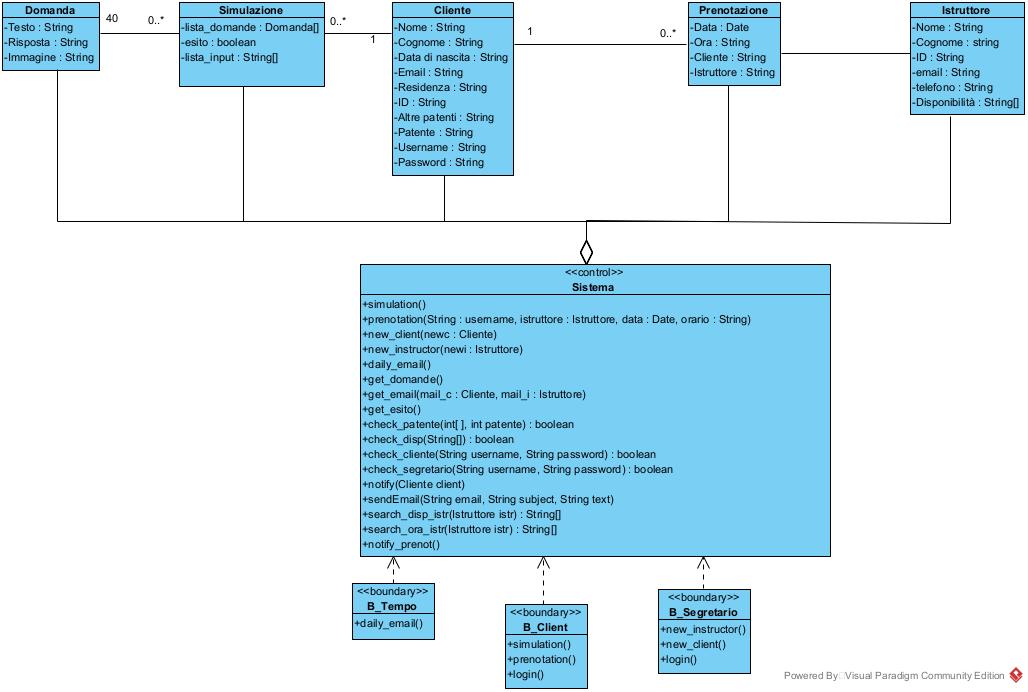
**TEST SUITE**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Descrizione** | **Classi di equivalenza coperte** | **Pre-condizioni** | **Input** | **Output Attesi** | **Post-condizioni Attese** |
| 1 | L’username inserito è nullo | Username null | Nessuna | Username = “ “ | Messaggio di errore “Username non valido” | Rimane la schermata del login |
| 2 | La password inserita è nullo | Password null | Nessuna | Password = “ “ | Messaggio di errore “Password non valida” | Rimane la schermata del login |
| 3 | Credenziali inserite correttamente | Username = stringa  Password = stringa | Nessuna | Username = “…”  Password = “…” | Messaggio di conferma | Si apre la schermata dell’home page |
| 4 | Credenziali inserite ma non corrette | Username = stringa  Password = stringa | Nessuna | Username = “…”  Password = “…” | Messaggio di errore “credenziali non corrette” | Rimane la schermata del login |

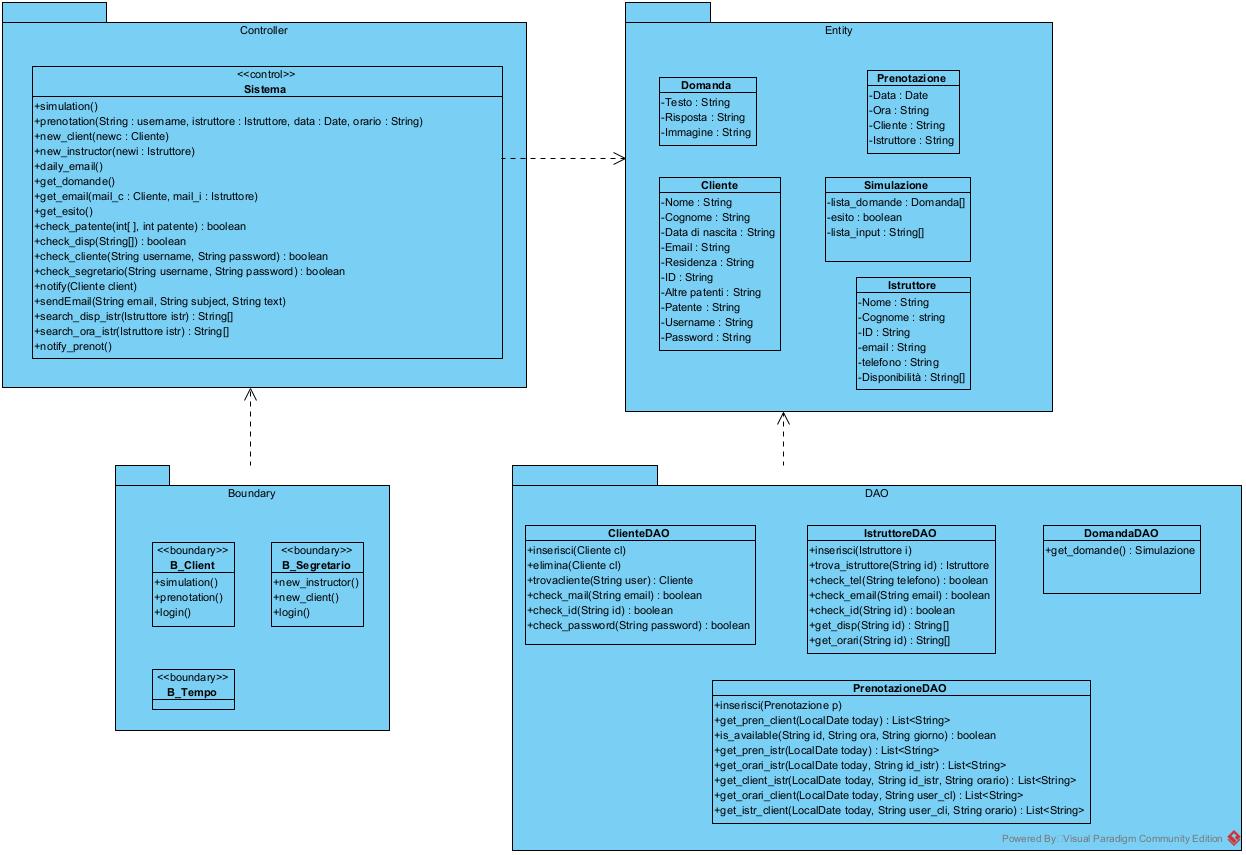
# Progettazione

## Diagramma delle classi

**Diagramma delle classi raffinato** e reificato con l’aggiunta di strutture dati e metodi per ogni entità e nel control del mio sistema.



**Diagramma dei package** con l’aggiunta della parte DAO per la gestione del DB.



## Diagrammi di sequenza

Diagramma di sequenza **Registrazione Cliente:**

Immagine che contiene tavolo

Descrizione generata automaticamente

Diagramma di sequenza **Registrazione Istruttore:**

Immagine che contiene tavolo

Descrizione generata automaticamente

Diagramma di sequenza **Prenotazione Lezione :**

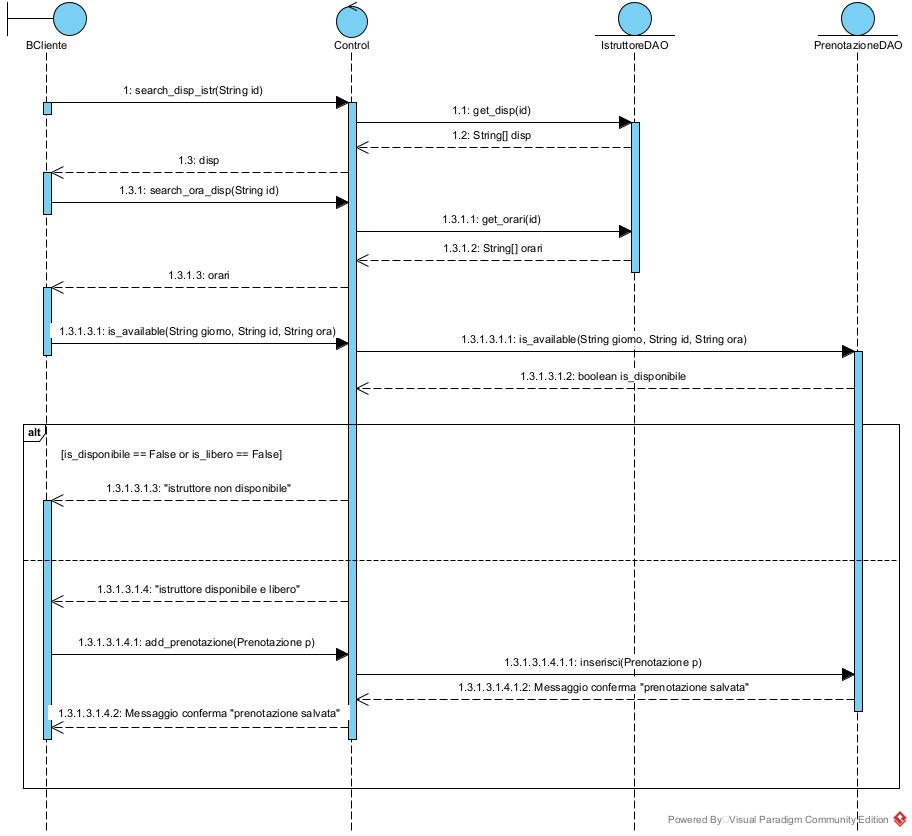


Diagramma di sequenza **Simulazione Esame:**

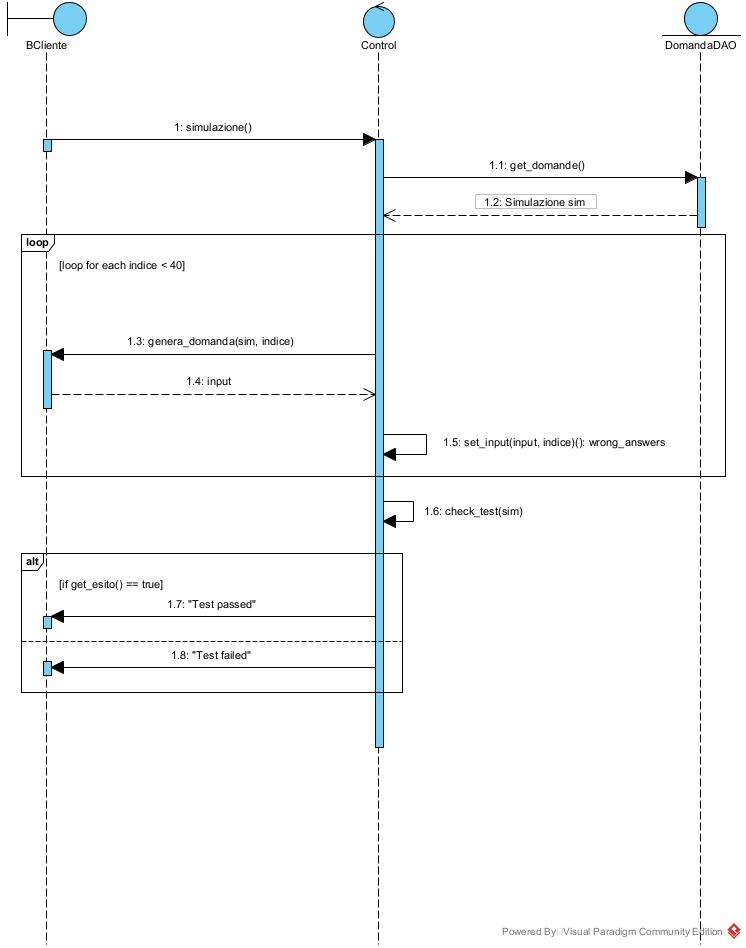
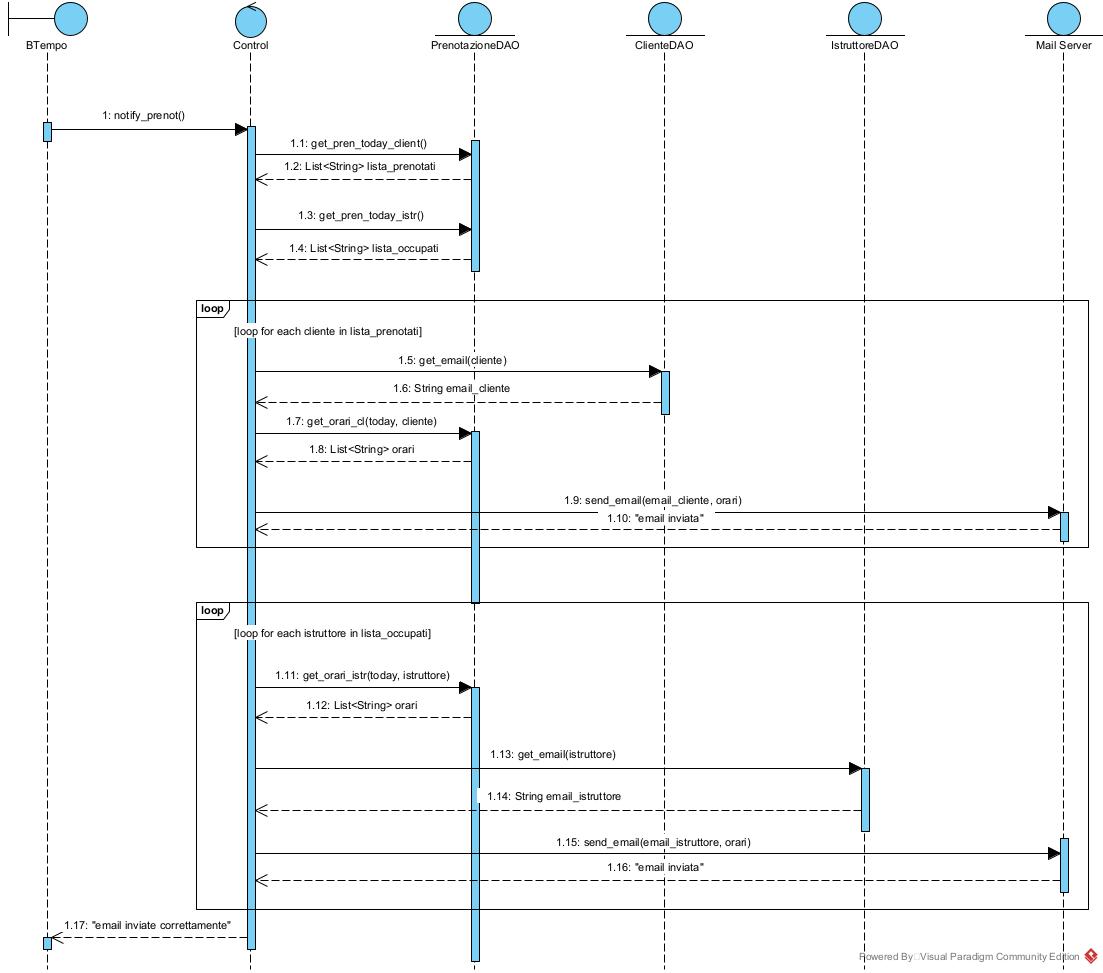


Diagramma di sequenza **Notifica Appuntamenti:**

****

# 7 Testing

## 7.1 Test funzionale

Una volta implementato il codice abbiamo messo in atto l’esecuzione dei **test case** esplicitati nella test suite all’interno del capitolo 4.

I risultati della test suite per “***Registrazione Cliente***” hanno avuto risultato *PASS* per la gran parte dei test case ad eccezione principalmente di quelli in cui la data di nascita veniva inserita come stringa di caratteri e non come number (id = 11).

I risultati della test suite per “***Registrazione Istruttore***” hanno avuto tutti risultato *PASS*.

I risultati della test suite per “***Prenotazione***” hanno avuto tutti risultato *PASS*.

I risultati della test suite per “***Simulazione***” hanno avuto tutti risultato *PASS*.

I risultati della test suite per “***Login Cliente***” hanno avuto tutti risultato *PASS*.

I risultati della test suite per “***Login Istruttore***” hanno avuto tutti risultato *PASS*.

Per la Registrazione Cliente, volendo **debuggare** il guasto presente all’interno del malfunzionamento con **id=11**, abbiamo provato diversi tipi di implementazioni.

Tuttavia, abbiamo scelto infine, di non effettuare cambiamenti per risolvere il guasto in quanto la data di nascita del cliente è un informazione puramente superficiale.

Inoltre, l’implementazione con Tipo **String** per il nostro codice serve per lavorare meglio con il DAO.

Infine, confrontando il lavoro del test funzionale con quello del test strutturale si può affermare che per esempio lo stesso malfunzionamento del caso id =11 in registrazione cliente non si sarebbe potuto riscontrare in quanto sarebbero stati inseriti solo valori String.

Ciò è dovuto al significato del test strutturale che prende come sorgente di informazioni per il test case il codice; dunque, questo genere di malfunzionamento con input di tipo sbagliato non viene verificato.

**TEST SUITE PER LA FUNZIONALITÀ *“REGISTRAZIONE CLIENTE*”.**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Classi di equivalenza coperte** | **Descrizione** | **Pre-condizioni** | **Input** | **Output Attesi** | **Post-condizioni Attese** | **Esito**  **(*FAIL*, *PASS*)** |
| 1 | Email troppo lunga | Email supera lunghezza mail | nessuna | Email = “provaprovaprovaprova@provaprovaprovaprova.it” | Messaggio di errore “Email non valida” | Il sistema torna sulla schermata di registrazione | ***PASS*** |
| 2 | Email null | La mail è una nulla | nessuna | Email = “ “ | Messaggio di errore “Email non valida” | Il sistema torna sulla schermata di registrazione | ***PASS*** |
| 3 | ID lunghezza non idonea | L’id non ha una lunghezza esatta di 7 caratteri | nessuna | Id = “aaaaa” | Messaggio di errore “ID non valido” | Il sistema torna sulla schermata di registrazione | ***PASS*** |
| 4 | Nome null | Il nome inserito è nullo | nessuna | Nome = “ “ | Messaggio di errore “nome non valido” | Il sistema torna sulla schermata di registrazione | ***PASS*** |
| 5 | Cognome null | Il cognome inserito è nullo | nessuna | Cognome = “ “ | Messaggio di errore “cognome non valido” | Il sistema torna sulla schermata di registrazione | ***PASS*** |
| 6 | ID null | L’id inserito è nullo | nessuna | Id = “ “ | Messaggio di errore “ID non valido” | Il sistema torna sulla schermata di registrazione | ***PASS*** |
| 7 | Data null | La data inserita è nulla | nessuna | Data = “ “ | Messaggio di errore “data non valida” | Il sistema torna sulla schermata di registrazione | ***PASS*** |
| 8 | Residenza null | La residenza inserita è nulla | nessuna | Residenza = “ “ | Messaggio di errore “residenza non valido” | Il sistema torna sulla schermata di registrazione | ***PASS*** |
| 9 | Le informazioni inserite sono corrette | Nome, cognome, id, email, residenza, patente, altre patenti = “…”  Data = “dd/mm/yyyy” | nessuna | Nome = “ab”  Cognome = “cd”  Email = [a@b.it](mailto:a@b.it)  Id = “1234567”  Residenza = “Via abc”  Data = “01/01/2000”  Patente = “A1”  Altre patenti = “AM” | Conferma di informazioni corrette | Il cliente è stato aggiunto al database | ***PASS*** |
| 10 | Patente non valida | Patente: AM, Altre patenti: AM | nessuna | Patente = “x”  Altre patenti = “x” |  | Il sistema torna sulla schermata di registrazione | ***PASS*** |
| 11 | Data di nascita non valida | Data di nascita con caratteri | nessuna | Data di nascita = “abc” | Messaggio di errore “data non valida” | Il sistema torna sulla schermata di registrazione | ***FAIL*** |
| 12 | Patente A | Patente inserita A | Nessuna | Patente =a; | Conferma di informazioni corrette | Il cliente è stato aggiunto al database | ***PASS*** |
| 13 | Patente B | Patente inserita B | Nessuna | Patente =B | Conferma di informazioni corrette | Il cliente è stato aggiunto al database | ***PASS*** |
| 14 | Patente A1 | Patente inserita A1 | Nessuna | Patente =A1 | Conferma di informazioni corrette | Il cliente è stato aggiunto al database | ***PASS*** |
| 15 | Patente AM | Patente inserita AM | Nessuna | Patente =AM; | Conferma di informazioni corrette | Il cliente è stato aggiunto al database | ***PASS*** |
| 16 | Patente A2 | Patente inserita A2 | Nessuna | Patente =a2; | Conferma di informazioni corrette | Il cliente è stato aggiunto al database | ***PASS*** |

**TEST SUITE PER LA FUNZIONALITÀ *“REGISTRAZIONE DIPENDENTE*”.**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Classi di equivalenza coperte** | **Descrizione** | **Pre-condizioni** | **Input** | **Output Attesi** | **Post-condizioni Attese** | **Esito**  **(*FAIL*, *PASS*)** |
| 1 | Email troppo lunga | Email supera lunghezza mail | nessuna | Email = “provaprovaprovaprova@provaprovaprovaprova.it” | Messaggio di errore “Email non valida” | Il sistema torna sulla schermata di registrazione | ***PASS*** |
| 2 | Email null | La mail è una nulla | nessuna | Email = “ “ | Messaggio di errore “Email non valida” | Il sistema torna sulla schermata di registrazione | ***PASS*** |
| 3 | ID lunghezza non idonea | L’id non ha una lunghezza esatta di 7 caratteri | nessuna | Id = “aaaaa” | Messaggio di errore “ID non valido” | Il sistema torna sulla schermata di registrazione | ***PASS*** |
| 4 | Nome null | Il nome inserito è nullo | nessuna | Nome = “ “ | Messaggio di errore “nome non valido” | Il sistema torna sulla schermata di registrazione | ***PASS*** |
| 5 | Cognome null | Il cognome inserito è nullo | nessuna | Cognome = “ “ | Messaggio di errore “cognome non valido” | Il sistema torna sulla schermata di registrazione | ***PASS*** |
| 6 | ID null | L’id inserito è nullo | nessuna | Id = “ “ | Messaggio di errore “ID non valido” | Il sistema torna sulla schermata di registrazione | ***PASS*** |
| 7 | Telefono null | Il telefono inserito è nullo | nessuna | telefono = “ “ | Messaggio di errore “telefono non valido” | Il sistema torna sulla schermata di registrazione | ***PASS*** |
| 8 | Telefono lunghezza non idonea | Il telefono inserito è troppo lungo | nessuna | telefono = “ 12323231221“ | Messaggio di errore “telefono non valido” | Il sistema torna sulla schermata di registrazione | ***PASS*** |
| 8 | Le informazioni inserite sono corrette | Nome, cognome, id, telefono, disp1, disp2, disp3 = “…” | nessuna | Nome = “ab”  Cognome = “cd”  Email = [a@b.it](mailto:a@b.it)  Id = “1234567890”  Disp1 = “LUN 15”  Disp2 = “LUN 16”  Disp3 = “MER 9” | Conferma di informazioni corrette | Il cliente è stato aggiunto al database | ***PASS*** |
| 9 | Disponibilità non valide | DISP1 = DISP2  DISP1 = DISP3  DISP2 = DISP3 | nessuna | Disp1 = Disp2  Disp1 = Disp3  Disp2 = Disp3 | Messaggio di errore sulle disponibilità | Il sistema torna sulla schermata di registrazione | ***PASS*** |
| 11 | Disponibilità nulle | Disp null | nessuna | Disp1 = null  Disp2 = null  Disp3 = null | Messaggio di errore sulle disponibilità | Il sistema torna sulla schermata di registrazione | ***PASS*** |

**TEST SUITE PER LA FUNZIONALITÀ *“PRENOTAZIONE*”.**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Descrizione** | **Classi di equivalenza coperte** | **Pre-condizioni** | **Input** | **Output Attesi** | **Post-condizioni Attese** | **Esito**  **(*FAIL*, *PASS*)** |
| 1 | L’id inserito è nullo | id null | nessuna | ID= “ “ | Messaggio di errore “ID non valido” | Il sistema torna sulla schermata di prenotazione | ***PASS*** |
| 2 | L’username inserito è nullo | Username null | nessuna | Username=” “ | Messaggio di errore “Username non valido” | Il sistema torna sulla schermata di prenotazione | ***PASS*** |
| 3 | Informazioni inserite correttamente | Id istruttore = caratteri alfanumerici  Username = stringa alfanumerica  Data = “DD/MM/YYYY” | nessuna | Id istruttore = “…”  Username = “…”  Data = “DD/MM/YYYY”  Giorno = “…” | La prenotazione è inserita correttamente | La prenotazione è inserita correttamente nel database | ***PASS*** |
| 4 | La data inserita è nulla | Data null | nessuna | Data = “ “ | Messaggio di errore “data non valida” | Il sistema torna sulla schermata di prenotazione | ***PASS*** |
| 5 | Orario non rientra nel range corretto | Orario out of range | Nessuna | Orario = “35” | Messaggio di errore “orario non valido” | Il sistema torna sulla schermata di prenotazione | ***PASS*** |
| 6 | Orario inserito è nullo | Orario null | Nessuna | Orario = “ ” | Messaggio di errore “orario non valido” | Il sistema torna sulla schermata di prenotazione | ***PASS*** |
| 7 | L’id inserito non è presente nel db. | id sbagliato | nessuna | ID= “sbagliato “ | // | Il sistema non procede alla ricerca dell’istruttore. | ***PASS*** |

**TEST SUITE PER LA FUNZIONALITÀ *“SIMULAZIONE*”.**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Descrizione** | **Classi di equivalenza coperte** | **Pre-condizioni** | **Input** | **Output Attesi** | **Post-condizioni Attese** | **Esito**  **(*FAIL*, *PASS*)** |
| 1 | Risposta inserita | Risposta: Vera | nessuna | Risposta = “0” | Schermata prossima domanda | input inserito correttamente e prossima domanda mostrata | ***PASS*** |
| 2 | Risposta inserita | Risposta: Falsa | nessuna | Risposta = “1” | Schermata prossima domanda | input inserito correttamente e prossima domanda mostrata | ***PASS*** |
| 3 | Risposta non inserita | La risposta non è stata inserita: verrà assunta come non data [single] | nessuna | Risposta = “2” | Schermata prossima domanda | input inserito e prossima domanda mostrata | ***PASS*** |
| 4 | Input inserito non valido | Risposta: diversa da vero o falso | nessuna | Risposta=”abc” | Messaggio di errore | Viene mostrata di nuovo la stessa domanda | ***PASS*** |

**TEST SUITE PER LA FUNZIONALITÀ *“LOGIN CLIENTE*”.**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Descrizione** | **Classi di equivalenza coperte** | **Pre-condizioni** | **Input** | **Output Attesi** | **Post-condizioni Attese** | **Esito**  **(*FAIL*, *PASS*)** |
| 1 | L’username inserito è nullo | Username null | Nessuna | Username = “ “ | Messaggio di errore “Username non valido” | Rimane la schermata del login | ***PASS*** |
| 2 | La password inserita è nullo | Password null | Nessuna | Password = “ “ | Messaggio di errore “Password non valida” | Rimane la schermata del login | ***PASS*** |
| 3 | Credenziali inserite correttamente | Username = stringa  Password = stringa | Nessuna | Username = “…”  Password = “…” | Messaggio di conferma | Si apre la schermata dell’home page | ***PASS*** |
| 4 | Credenziali inserite ma non corrette | Username = stringa  Password = stringa | Nessuna | Username = “…”  Password = “…” | Messaggio di errore “credenziali non corrette” | Rimane la schermata del login | ***PASS*** |

**TEST SUITE PER LA FUNZIONALITÀ *“LOGIN ISTRUTTORE*”.**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Descrizione** | **Classi di equivalenza coperte** | **Pre-condizioni** | **Input** | **Output Attesi** | **Post-condizioni Attese** | **Esito**  **(*FAIL*, *PASS*)** |
| 1 | L’username inserito è nullo | Username null | Nessuna | Username = “ “ | Messaggio di errore “Username non valido” | Rimane la schermata del login | ***PASS*** |
| 2 | La password inserita è nullo | Password null | Nessuna | Password = “ “ | Messaggio di errore “Password non valida” | Rimane la schermata del login | ***PASS*** |
| 3 | Credenziali inserite correttamente | Username = stringa  Password = stringa | Nessuna | Username = “…”  Password = “…” | Messaggio di conferma | Si apre la schermata dell’home page | ***PASS*** |
| 4 | Credenziali inserite ma non corrette | Username = stringa  Password = stringa | Nessuna | Username = “…”  Password = “…” | Messaggio di errore “credenziali non corrette” | Rimane la schermata del login | ***PASS*** |