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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Numero Studenti** | **Corsi** | **Appelli per Corso** | **Data** | **Stato Appello** | **Esito Ricerca** |
| * 0 [SINGLE] * 1 * >1 | * 0 [ERR] * 1 * >1 | * 0 [ERR] * 1 * >1 | * 0 [SINGLE] * 1 * >1 | * Aperto * Chiuso | * 0 * 1 * >1 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Descrizione** | **Classi di equivalenza coperte** | **Pre-condizioni** | **Input** | **Output Attesi** | **Post-condizioni Attese** | **Output Ottenuti** | **Post-condizioni Ottenute** | **Esito**  **(*FAIL*, *PASS*)** |
| 1 | 1 studente, prenota per 1 corso, 1 appello contenente 1 data | STUDENTE: **1**  CORSO: **1**  APPELLO: **1**  DATA: **1** | Docente già esistente Stato appello: Aperto | Studente()  creaCorso()  Data() x1  creaAppello(data)  prenotazioneAppello() | 1 | N/A | 1 | N/A | PASS |
| 2 | 0 studenti, prenotano per 1 corso, 1 appello con 1 data | STUDENTE: 0  CORSO: **1**  APPELLO: **1**  DATA: **1** | Docente già esistente  Stato appello: Aperto | creaCorso()  Data() x1  creaAppello(data) | 0 | N/A | 0 | N/A | PASS |
| 3 | 1 studente, prenota per 2 corsi, ogni corso contiene 1 appello con una data | STUDENTE: **1**  CORSO: 2  APPELLO: 2  DATA: **1** | Docente già esistente  Stato appello: Aperto | Studente()  creaCorso()x2  Data()  creaAppello(data) x2  prenotazioneAppello() | Corso 1 = 1;  Corso 2 =  1; | N/A | 1/1 | N/A | PASS |
| 4 | Più studenti, prenotano 1 appelli di un corso, contenente 1 data | STUDENTE: 2  CORSO: 1  APPELLO: 1  DATA: **1** | Docente già esistente  Stato appello: Aperto | Studente()x2  creaCorso()  Data()  creaAppello(data)  prenotazioneAppello()x2 | 2 | N/A | 2 | N/A | PASS |
| 5 | 1 studente tenta di prenotarsi 2 volte allo stesso appello di 1 corso contenente 1 data | STUDENTE: **1**  CORSO: 1  APPELLO: 1  DATA: 1 | Docente già esistente  Stato appello: Aperto | Studente()  creaCorso()  Data()  creaAppello(data)  prenotazioneAppello()x2 | 1 | N/A | 1 | N/A | PASS |
| 6 | più studenti si prenotano a 2 appelli di 1 corso con 1 data ciascuno. 3 studenti al primo appello, 2 al secomdo | STUDENTE: 5  CORSO: 1  APPELLO: 2  DATA: 2 | Docente già esistente  Stato appello: Aperto | Studente()x5  creaCorso()x1  Data() x2  creaAppello(data) x2  prenotazioneAppello()x5 | Appello1 = 3;  Appello 2= 2; | N/A | 3/2 | N/A | PASS |
| 7 | 4 studenti si prenotano a 2 appelli di un corso. Ciascun appello 2 date.  2 studenti prenotano al primo, 2 studenti si prenotano al secondo. | STUDENTE: 4  CORSO: 1  APPELLO: 2  DATA: 4 | Docente già esistente  Stato appello: Aperto | Studente()x4  Data()  creaCorso()  creaAppello(data)x4  prenotaAppello()x4  chiudiAppello()  prenotaAppello()x4 | Appello1 = 4;  Appello 2= 4; | N/A | 4/4 | N/A | PASS |
| 8 | 3 studenti si prenotano a 1 appello di 2 corsi contenenti una data ciascuno. 2 si prenotano all’appello del primo corso e 3 all’appello del  secondo. | STUDENTE: 3  CORSO: 2  APPELLO: 2  DATA: 2 | Docente già esistente  Stato appello: Aperto | Studente()x3  creaCorso()x2  Data()x2  creaAppello(data) x2  prenotazioneAppello()x6 | Corso1= 2  Corso2=  3 | N/A | 3/3 | N/A | PASS |
| 9 | 1 studente si prenota ad un appello di un corso, contenente una data, CHIUSO. | STUDENTE: 1  CORSO: 1  APPELLO: 1  DATA: 1 | Docente già esistente | Studente()  creaCorso()  Data()  creaAppello(data)  chiudiAppello()  prenotazioneAppello() | 0 | N/A | 0 | N/A | PASS |
| 10 | 2 studenti si prenotano a 1 appello di 1 corso contenente 1 data. Il primo si prenota quando l’appello è aperto, il secondo quando è chiuso. | STUDENTE: 2  CORSO: 1  APPELLO: 1  DATA: 1 | Docente già esistente | Studente()x2  creaCorso()  Data()  creaAppello(data)  prenotazioneAppello()  chiudiAppello()  prenotazioneAppello() | 1 | N/A | 1 | N/A | PASS |
| 11 | 1 studente si prenota a 1 appello di 1 corso contenente 1 data. Tenta poi di prenotarsi ad un altro appello dello stesso corso con il precedente ancora aperto. | STUDENTE: 1  CORSO: 1  APPELLO: 2  DATA: 2 | Docente già esistente | Studente()  creaCorso()  Data()x2  creaAppello()x2  prenotaAppello(data1)  prenotaAppello(data2) | APPELLO 1 = 1;  APPELLO 2 = 0; | N/A | 1/0 | N/A | PASS |
| 12 | 2 studenti tentano di prenotarsi a 1 appello di 2 corsi contenenti 1 data quando sono già prenotati in precedenza a 1 appello aperto di quegli stessi corsi | STUDENTE: 2  CORSO: 2  APPELLO: 4  DATA: 2 | Docente già esistente | Studente()x2  creaCorso()x2  Data()x2  creaAppello()x4  prenotaAppello(data1)x2  prenotaAppello(data2)x2 | Corso1->  Appello 1 = 2;  Appello 2 = 0;  Corso2->  Appello 1 = 2;  Appello 2 = 0; | N/A | 2/0; 2/0 | N/A | PASS |
| 13 | 1 studente si prenota all’appello 2 (contenente1 data) di un corso, quando l’appello 1(contenente1 data) è chiuso. Quando il suo appello è concluso si prenota al numero 3 (contenente1 data) | STUDENTE: 1  CORSO: 1  APPELLO: 3  DATA: 3 | Docente già esistente | Studente()  creaCorso()  Data()x3  creaAppello()x3  chiudiAppello  prenotaAppello(data2)  chiudiAppello  prenotaAppello(data3) | Appello 1 = 0;  Appello 2 = 1;  Appello 3 = 1; | N/A | 0/1/1 | N/A | PASS |
| 14 | 1 studente si prenota ad 1 appello di 1 corso che non ha data. | STUDENTE: 1  CORSO: 1  APPELLO: 1  DATA: 0 | Docente già esistente  Stato appello: Aperto | Studente()  creaCorso()  creaAppello()  prenotaAppello(null) | 0 | N/A | 0 | N/A | PASS |