1. **Introduzione**

Lo scopo di questo documento è quello di analizzare, gestire lo sviluppo e le attività di testing riguardanti il software Gestione prenotazioni.

Tale lavoro consiste nel verificare il corretto funzionamento del Sistema analizzando i singoli casi, mettendoli alla prova tramite il testing. I risultati del testing verranno utilizzati per capire i problemi del nostro sistema cercando di correggere errori o apportare modifiche dove necessario.

Il processo di testing viene iterato fino a che non si soddisfino in maniera accettabile i requisiti funzionali e non funzionali del sistema.

1. **Relazione con gli altri documenti**

Al fine di verificare il corretto funzionamento di Gestione Prenotazioni abbiamo predisposto dei test case basati sulle funzionalità individuate nel RAD.

Tratteremo questi aspetti nella Sezione 8 di questo documento. Per ogni test case verrà specificato un **category partition** contenente gli input e gli output con il seguente risultato.

Per stilare questo documento sono state utilizzate:

* le slides del corso di Ingegneria del Software
* il libro B.Bruegge, A.H.Dutoit. Object-Oriented Software Engineering: Using UML, Patterns and Java. Third edition;

Per il testing di unità farà riferimento all’ SDD per la suddivisione in sottosistemi.

Per il testing d’integrazione per il riferimento all’interfaccia delle classi e i package definiti nell’ODD.

1. **Panoramica del sistema**

Il sistema sarà realizzato per l’Università degli Studi di Salerno.

Il sistema ha lo scopo di rendere più semplice e sistematica la prenotazione delle aule.

Il sistema è una piattaforma che permette ad uno studente o ad un docente la prenotazione delle aule.

La piattaforma mette a disposizione delle funzionalità che permette loro di organizzarsi in maniera autonoma senza contattare la segreteria e di velocizzare i tempi.

I gestori della piattaforma possono visualizzare le richieste di prenotazione ed hanno il completo controllo di esse.

1. **Funzionalità da testare e non**

**Gestione utente**

**TC\_GU1**:Registrazione

**TC\_GU2**:Autenticazione

**TC\_GU3**:Logout(NO)

**TC\_GU4**:Visualizzazione area personale(NO)

**TC\_GU5**:Modifica Password

**TC\_GU6:**Elimina prenotazione(NO)

**Gestione Eventi Amm.Dipartimento**

**TC\_GED1**:Accettazione richiesta di prenotazione (NO)

**TC\_GED2**:Rifiuto richiesta di prenotazione (NO)

**TC\_GED3**:Visualizzazione richieste prenotazione (NO)

**TC\_GED4**:Prenotazione aula

**TC\_GED5:**Aggiungi aule lista studenti

**TC\_GED6**:Rimuovi aule lista studenti

**Gestione vista degli utenti**

**TC\_GV1**:Visualizza storico prenotazioni (NO)

**TC\_GV2**:Visualizzazione area personale (NO)

**TC\_GV3**:Visualizzazione edificio (NO)

**TC\_GV4:**Visualizzazione aula (NO)

**TC\_GV5**:Visualizzazione calendario(NO)

**TC\_GV6:** Visualizzazione dipartimento(NO)

**Gestione eventi docente/studente**

**TC\_EDC1**: Richiesta di prenotazione aula

**Gestione Eventi Amm. Ateneo**

**TC\_AG1**:Aggiunta dipartimento

**TC\_AG2**:Rimozione dipartimento(NO)

**TC\_AG3**: Aggiunta edificio

**TC\_AG4**:Rimozione edificio(NO)

**TC\_AG5**:Aggiunta aula

**TC\_AG6**:Rimozione aula(NO)

**TC\_AG7**:Aggiunta aule a dipartimento

**TC\_AG8**:Rimozione aule a dipartimento(NO)

**TC\_AG9**:Visualizzazione elenco utenti (NO)

**TC\_AG10**:Assegnazione titolo amministratore

**TC\_AG11**:Rimozione titolo amministratore(NO)

**TC\_AG12**:Assegnazione amministrazione a dipartimento

**TC\_AG13**:Rimozione amministrazione a dipartimento (NO)

1. **Pass/Fail criteria**

Lo scopo del testing è quello di dimostrare la presenza di faults (errori) all’interno del sistema.

Le attività di testing, infatti, saranno mirate all’identificazione di questi faults e ad un successivo intervento per eliminarne la presenza.

Un test avrà successo se, dato un input al sistema, l’output osservato sarà diverso dall’output atteso dall’oracolo, quindi è stata identificata una failure. In tal caso questa verrà analizzata e si procederà alla correzione.

Al contrario, un test fallirà quando l’output osservato sarà uguale a quello presente nell’oracolo.

Per oracolo si intende il risultato atteso, in base ai requisiti, dall’esecuzione di un caso di test.

La fase di testing non sarà considerata valida se una delle seguenti condizioni non verrà rispettata:

* Rieseguire tutti i test ogni volta che si introducono nuove caratteristiche al sistema o vengono modificate quelle presenti;
* Raggiungere un branch coverage non inferiore al 75%.

1. **Approccio**

Per il sistema Gestione Prenotazioni, il testing si compone tre fasi. Nella prima fase verranno eseguiti i test di unità dei singoli componenti, in modo da testare nello specifico la correttezza di ciascuna unità andando a constatare il corretto funzionamento. Questa fase verrà effettuata al completamento di ogni unità realizzata per poter individuare tempestivamente gli errori presenti nel codice.

Nella seconda fase verrà effettuato il testing di integrazione in cui si andrà a testare l’integrazione dei vari sottosistemi.

Infine, verrà eseguito il testing di sistema, che vedrà come oggetto di testing l’intero sistema assemblato nei suoi componenti.

Le attività di testing saranno eseguite parallelamente all’implementazione del sistema, dal momento che si è scelto di utilizzare un metodo di sviluppo incrementale in questa fase. Ogni volta che verrà introdotta una nuova funzionalità, verrà eseguito il testing di unità e successivamente il test di integrazione con quelle già presenti.

Nella sessione di testing del sistema verrà utilizzato un approccio “**BLACK BOX**” di tipo **SANDWICH MODIFICATO**.

Combina l’uso di strategie top-down e bottom-up. Il sistema è visto come se avesse 3 strati: un livello target nel mezzo, uno sopra il target, uno sotto il target.

Esso prevede che i test vengano effettuati sulle specifiche delle funzionalità da testare senza il bisogno di conoscerne la struttura implementativa.

Questo processo sarà dunque scomposto in 3 fasi:

● **Testing di unità**: Per effettuare il testing di ogni singola componente del sistema Gestione Prenotazioni, verrà utilizzata la tecnica “Black-Box testing” attraverso il framework JUnit. In questa fase saranno analizzate le funzionalità dell’applicazione ed il comportamento delle singole componenti senza tener conto della loro struttura

interna. Al fine di minimizzare il numero di test case, gli input saranno partizionati in classi di equivalenza. Tali classi serviranno a definire le categorie per poter utilizzare, come testing funzionale, il

Category Partition quindi per ogni classe di equivalenza verranno individuati dei vincoli a cui saranno

associate delle scelte specifiche. I risultati del testing verranno analizzati e usati per correggere gli errori che causano il fallimento del sistema. Se si verifica un errore con dei risultati inattesi si interviene in maniera tempestiva sulla componente in modo da renderla correttamente funzionante e procedere con le fasi di testing successive.

Per unità intendiamo un singolo servizio offerto dalle classi che compongono il back-end.

● **Testing di integrazione**: In seguito al test di unità di ogni componente, si procederà con il test di integrazione utilizzando un approccio bottom-up.

● **Testing di sistema**: Lo scopo principale di questa fase di testing è quello di dimostrare che l’intero sistema funzioni correttamente e che soddisfi effettivamente i requisiti funzionali e non funzionali descritti nel

documento di analisi dei requisiti (RAD). È da considerare l’attività più critica, in quanto può risultare molto complesso andare alla ricerca di eventuali errori, essendo impegnati tutti i sottosistemi. Inoltre abbiamo scelto di come Strategia di Testing il Category Partition per i form.

1. **Sospensione e ripristino**

La fase di testing occupa una parte importante nell’intera attività inerente il progetto, essa è anche molto delicata e può essere causa di imprevisti e slittamenti dei tempi a causa di errori e malfunzionamento del sistema.

**7.1 Criteri di sospensione**

Il testing sarà sospeso quando verrà raggiunta una branch coverage di almeno il 75%. Nel caso in cui la schedulazione del progetto fosse rispettata si cercherà di testare tutte le classi selezionate per ogni possibile input e il risultato dovrà essere quello atteso.

Nel momento in cui il testing rivelerà un errore si dovrà passare alla fase di correzione che dovrà essere eseguita da una ripetizione dell’intero processo di testing per rilevare la presenza di eventuali errori introdotti dalle correzioni stesse.

**7.2 Criteri di ripresa**

La ripresa del test avviene soltanto quando tutti i problemi relativi alla sospensione dello stesso sono stati risolti. L’attività di testing riprenderà a partire dal test case che ha causato la sospensione.

1. **Strumenti per il testing**

Il testing verrà effettuato nell’Ambiente di sviluppo(Eclipse) e sulle nostre macchine.

Per il testing useremo il tool **JUnit** per Eclipse.

Un tabella per annotare i test eseguiti:

|  |  |
| --- | --- |
| Parametro  Formato | |
| Categoria | Scelte |

|  |  |  |
| --- | --- | --- |
| Codice | Combinazione | Esito |
|  |  |  |
|  |  |  |

1. **Test Cases**

**Gestione Utente**

***TC\_GU1: Registrazione***

|  |  |
| --- | --- |
| Parametro: Nome  Formato: [a-zA-Z] | |
| Categoria | Scelte |
| Lunghezza [LN] | 1. <2 or > 25 [error]  2. >=3 and <=12[LN\_OK] |
| Formato[FN] | 1. Non rispetta il formato [error]  2. Rispetta il formato[FN\_OK] |

|  |  |
| --- | --- |
| Parametro: E-mail  Formato: ^\w+([\.-]?\w+)\*@]\* studenti.unisa.it OR ^\w+([\.-]?\w+)\*@]\*unisa.it | |
| Categoria | Scelte |
| Formato[FE] | 1. Non rispetta il formato [error]  2. Rispetta il formato[FE\_OK] |
| Registrato[RE] | 1. È già presente nel database[error]  2. Non è già registrata nel database[NRE\_ok] |

|  |  |
| --- | --- |
| Parametro: Password  Formato: ^\(?=.\*\d.\*\d)(?=.\*[A-Z])(?=.\*[a-z])[0-9°-Za-z]{8,32}$ | |
| Categoria | Scelte |
| Formato[FP] | 1. Non rispetta il formato [error]  2. Rispetta il formato[FP\_OK] |

|  |  |
| --- | --- |
| Parametro: Cognome  Formato: [a-zA-Z] | |
| Categoria | Scelte |
| Lunghezza [LC] | 1. Cognome >25 [error]  2. Cognome <0 [error] |
| Formato[FC] | 1. Non rispetta il formato [error]  2. Rispetta il formato[FC\_OK] |

***TC\_GU2: Autenticazione***

|  |  |
| --- | --- |
| Parametro: E-mail  Formato: ^\w+([\.-]?\w+)\*@]\* studenti.unisa.it \ OR ^\w+([\.-]?\w+)\*@]\*unisa.it\ | |
| Categoria | Scelte |
| Lunghezza [LE] | 1. >=3 and <=254 [FE\_OK]  2. email<=0 [error] |
| Formato[FE] | 1. Non è presente nel database[error]  2. Presente nel database[FE\_OK] |
|  |  |

|  |  |
| --- | --- |
| Parametro: Password  Formato: ^\(?=.\*\d.\*\d)(?=.\*[A-Z])(?=.\*[a-z])[0-9°-Za-z]{8,16}$\ | |
| Categoria | Scelte |
| Lunghezza [LP] | 1. < 8or >16 [error]  2. >=8 and <=16 [FP\_OK]  3. Password <=0 [error] |
| Formato[FP] | 1.Password inserita diversamente da quella presente nel database[error]  2. Password inserita uguale a quella presente ne database [FB\_OK] |

***TC\_GU3: Modifica Password***

|  |  |
| --- | --- |
| Parametro: Password  Formato: ^\(?=.\*\d.\*\d)(?=.\*[A-Z])(?=.\*[a-z])[0-9°-Za-z]{8,16}$\ | |
| Categoria | Scelte |
| Lunghezza [LP] | 1. < 8or >16 [error]  2. >=8 and <=16 [FP\_OK]  3. Password <=0 [error] |
| Formato[FP] | 1.Password inserita diversa da quella presente nel database[error]  2. Password inserita uguale a quella presente ne database [FB\_OK] |

|  |  |
| --- | --- |
| Parametro: Nuova Password  Formato: ^\(?=.\*\d.\*\d)(?=.\*[A-Z])(?=.\*[a-z])[0-9°-Za-z]{8,16}$ | |
| Categoria | Scelte |
| Lunghezza [LNP] | 1. < 8or >16 [error]  2. >=8 and <=16 [FP\_OK]  3. Password <=0 [error] |
| Formato[FNP] | 1.Non rispetta il formato[error]  2. Rispetta il formato[FNP\_OK] |

***Logout***

Category Partition non presente perché la funzionalità non prevede inserimento di dati da parte dell’utente.

***Visualizza area personale***

Category Partition non presente perché la funzionalità non prevede inserimento di dati da parte dell’utente.

* **9.2 Gestione Eventi Amm. di Dipartimento**
  + *9.2.1 Accettazione richiesta Prenotazione*

Category Partition non presente perché la funzionalità non prevede inserimento di dati da parte dell’utente.

* + *9.2.2 Rifiuto richiesta prenotazione*

Category Partition non presente perché la funzionalità non prevede inserimento di dati da parte dell’utente.

* + *9.2.3 Visualizzazione richieste prenotazione*

Category Partition non presente perché la funzionalità non prevede inserimento di dati da parte

dell’utente.

* + *9.2.4 prenotazione aula*

|  |  |
| --- | --- |
| Parametro: Edificio  Formato: [a-z A-Z 0-9] | |
| Categoria | Scelte |
| Esiste [EE] | 1. Non esiste l’edificio [error] 2. Esiste l’edificio [EE\_OK] |

|  |  |
| --- | --- |
| Parametro: Aula  Formato: [a-z A-Z 0-9] | |
| Categoria | Scelte |
| Esiste [EA] | 1. Non esiste l’aula [error] 2. Esiste l’aula [EA\_OK] |

|  |  |
| --- | --- |
| Parametro: Titolo  Formato: [a-z A-Z 0-9] | |
| Categoria | Scelte |
| Formato[FT] | 1. Non rispetta il formato [error] 2. Rispetta il formato [FT\_OK] |
| Lunghezza[LT] | 1. LT >=0 or LT >55 [LT\_OK] 2. LT <= 0 or 55 < LT [error] |

|  |  |
| --- | --- |
| Parametro: Descrizione  Formato: [a-z A-Z 0-9] | |
| Categoria | Scelte |
| Formato[FD] | 1. Non rispetta il formato [error] 2. Rispetta il formato [FD\_OK] |
| Lunghezza[LD] | 1. LT >=0 or LT >255 [LD\_OK] 2. LT <= 0 or 255 < LT [error] |

|  |  |
| --- | --- |
| Parametro: Data | |
| Categoria | Scelte |
| Selezione GG [GG] | 1. False [GG\_error] 2. True [GG\_OK] |
| Selezione MM [MM] | 1. IF (GG\_OK)False [MM\_error] 2. IF (GG\_OK)True [MM\_OK] |
| Selezione AAAA [AAAA] | 1. IF (GG\_OK AND MM\_OK)False [AAAA\_error] 2. IF (GG\_OK AND MM\_OK)True [AAAA\_OK] |
| Coerenza Data [CD] | 1:if([GG\_OK] && [MM\_OK] && [AAAA] ) Data > localDate() [ok]  2:if( [GG\_OK] && [MM\_OK] && [AAAA]) Data = localDate() [ok]  3:if([GG\_OK] && [MM\_OK] && [AAAA]) Data < localDate() [CD\_Error] |

|  |  |
| --- | --- |
| Parametro: Ora Inizio | |
| Categoria | Scelte |
| Selezione Ora Inizio [OI] | 1. False [OI\_error] 2. True [OI\_OK] |

|  |  |
| --- | --- |
| Parametro: Ora Fine | |
| Categoria | Scelte |
| Selezione Ora Inizio [OF] | 1. IF (OI\_OK)False [OF\_error] 2. IF(OI\_OK) True [OF\_OK] |
| Coerenza [OI && OF] | 1:if( [OI\_OK] &&[OF\_OK] )Orario inizio < Orario fine [ok]  2:if( [OI\_OK] &&[OF\_OK] )Orario inizio = Orario fine [OI && OF\_Error]  3:if( [OI\_OK] &&[OF\_OK] )Orario inizio > Orario fine [OI && OF\_Error] |

|  |  |
| --- | --- |
|  | |
|  |  |
|  |  |

Category Partition non presente perché la funzionalità non prevede inserimento di dati da parte dell’utente.

* **9.3 Gestione vista degli utenti**
  + *9.3.1 Visualizza storico prenotazioni*
    - Category Partition non presente perché la funzionalità non prevede inserimento di dati da parte dell’utente.
  + *9.3.2 visualizza edificio*
    - Category Partition non presente perché la funzionalità non prevede inserimento di dati da parte dell’utente.
  + *9.3.3 visualizza aula*
    - Category Partition non presente perché la funzionalità non prevede inserimento di dati da parte dell’utente.
  + *9.3.4 visualizza calendario*
    - Category Partition non presente perché la funzionalità non prevede inserimento di dati da parte dell’utente.
  + *9.3.5 visualizza dipartimento*
    - Category Partition non presente perché la funzionalità non prevede inserimento di dati da parte dell’utente.
* **9.4 Gestione eventi Docente/Studente**
  + *9.4.1 Richiesta di prenotazione aula*

*Vedi Sezione 9.4.2*

* **9.5 Gestione eventi Amm. Generale**
  + *9.5.1 aggiunta dipartimento*

|  |  |
| --- | --- |
| Parametro: Nome  Formato: [a-z A-Z 0-9] | |
| Categoria | Scelte |
| Formato[FN] | 1. Non rispetta il formato [error] 2. Rispetta il formato [FN\_OK] |
| Lunghezza[LN] | 1. LN >=0 or LT >55 [LN\_OK] 2. LN <= 0 or 55 < LN [error] |

* + *9.5.2 rimozione dipartimento*
    - Category Partition non presente perché la funzionalità non prevede inserimento di dati da parte dell’utente.
  + *9.5.3 aggiunta edificio*

|  |  |
| --- | --- |
| Parametro: Nome  Formato: [a-z A-Z 0-9] | |
| Categoria | Scelte |
| Formato[FN] | 1. Non rispetta il formato [error] 2. Rispetta il formato [FN\_OK] |
| Lunghezza[LN] | 1. LN >=0 or LT >55 [LN\_OK] 2. LN <= 0 or 55 < LN [error] |

* + *9.5.4 rimozione edificio*
    - Category Partition non presente perché la funzionalità non prevede inserimento di dati da parte dell’utente.
  + *9.5.5 aggiunta aula*

|  |  |
| --- | --- |
| Parametro: Edificio  Formato: [a-z A-Z 0-9] | |
| Categoria | Scelte |
| Esiste[EE] | 1. Non esiste l’edificio [error] 2. Esiste l’edificio [EE\_OK] |

|  |  |
| --- | --- |
| Parametro: Nome  Formato: [a-z A-Z 0-9] | |
| Categoria | Scelte |
| Formato[FN] | 1. Non rispetta il formato [error] 2. Rispetta il formato [FN\_OK] |
| Lunghezza[LN] | 1. LN >=0 or LT >55 [LN\_OK] 2. LN <= 0 or 55 < LN [error] |

* + *9.5.6 rimozione aula*
    - Category Partition non presente perché la funzionalità non prevede inserimento di dati da parte dell’utente.
  + *9.5.7 aggiunta aule a dipartimento*
  + *9.5.8 rimozione aule a dipartimento*
    - Category Partition non presente perché la funzionalità non prevede inserimento di dati da parte dell’utente.
  + *9.5.9 visualizzazione elenco utenti*
    - Category Partition non presente perché la funzionalità non prevede inserimento di dati da parte dell’utente.
  + *9.5.10 assegnazione titolo amministratore*

|  |  |
| --- | --- |
| Parametro: Nome  Formato: [a-z A-Z 0-9] | |
| Categoria | Scelte |
| Formato[FN] | 1. Non rispetta il formato [error] 2. Rispetta il formato [FN\_OK] |
| Lunghezza[LN] | 1. LN >=0 or LT >55 [LN\_OK] 2. LN <= 0 or 55 < LN [error] |

* + *9.5.11 rimozione titolo amministratore*

|  |  |
| --- | --- |
| Parametro: Nome  Formato: [a-z A-Z 0-9] | |
| Categoria | Scelte |
| Formato[FN] | 1. Non rispetta il formato [error] 2. Rispetta il formato [FN\_OK] |
| Lunghezza[LN] | 1. LN >=0 or LT >55 [LN\_OK] 2. LN <= 0 or 55 < LN [error] |

* + - Category Partition non presente perché la funzionalità non prevede inserimento di dati da parte dell’utente.
  + *9.5.12 assegnazione amministratore a dipartimento*

|  |  |
| --- | --- |
| Parametro: Nome  Formato: [a-z A-Z 0-9] | |
| Categoria | Scelte |
| Formato[FN] | 1. Non rispetta il formato [error] 2. Rispetta il formato [FN\_OK] |
| Lunghezza[LN] | 1. LN >=0 or LT >55 [LN\_OK] 2. LN <= 0 or 55 < LN [error] |

* + *9.5.13 rimozione amministratore a dipartimento*
    - Category Partition non presente perché la funzionalità non prevede inserimento di dati da parte dell’utente.

1. **Testing Schedule**

Tutte le specifiche dei Test Case, inclusi quelli Funzionali ed’ unità verranno descritte nel documento di specifica dei Test Case.