Università degli studi di Salerno

Dipartimento di Informatica

***Corso di Laurea in Informatica***



***INGEGNERIA DEL SOFTWARE***

***Problem Statement***

***“VORWERK”***

*.*

**Docente:**

Andrea De Lucia

**Studenti:**

##### Nome Matricola

Vincenzo Marrazzo 0512105832

*Anno Accademico: 2020/21*

INDICE

Sommario

[**1.** **Introduzione** 2](#_Toc62744494)

[**2.** **Relazione con gli altri documenti** 3](#_Toc62744495)

[**3.** **Panoramica del sistema** 3](#_Toc62744496)

[**4.** **Funzionalità da testare** 4](#_Toc62744497)

[5. **Criteria Pass/Failed** 4](#_Toc62744498)

[**6.** **Approccio** 5](#_Toc62744499)

[**7.** **Sospensione e ripresa** 5](#_Toc62744500)

[7.1 Criteri di sospensione 5](#_Toc62744501)

[7.2 Criteri di ripresa 5](#_Toc62744502)

[**8.** **Materiale per il testing** 6](#_Toc62744503)

[**9.** **Test Cases** 6](#_Toc62744504)

[9.1 Gestione Robot 6](#_Toc62744505)

[9.1.1 Inserimento Robot/Modifica Robot/Rimuovi Robot 6](#_Toc62744506)

[9.1.2 Ricerca Robot 7](#_Toc62744507)

[9.2 Gestione Account 7](#_Toc62744508)

[9.2.1 Register Account/Login 7](#_Toc62744509)

[9.3 Gestione Premi 8](#_Toc62744510)

[9.3.1 Inserimento Premio/Modifica Premio/Rimuovi Premio 8](#_Toc62744511)

[9.4 Gestione Acquisto Robot 8](#_Toc62744512)

[9.4.1 Aggiungi Robot al Carrello/Rimuovi Robot dal Carrello 8](#_Toc62744513)

[9.4.2 Crea Ordine Robot 9](#_Toc62744514)

[9.5 Gestione Ricevimento Premi 10](#_Toc62744515)

[9.5.1 Crea Ordine Premio 10](#_Toc62744516)

# **Introduzione**

La stesura del documento di Test Plan come obiettivo quello di analizzare e gestire lo sviluppo e le attività di testing riguardanti il software VORWERK.  
Quest’attività sarà svolta in modo tale da testare il funzionamento di VORWERK in qualsiasi circostanza, al fine di mettere alla prova ogni singola funzionalità e caratteristica del sistema, in modo da verificare che non vi sia alcuna incongruenza tra il comportamento atteso e il comportamento ottenuto. I risultati di tale analisi saranno essenziali per identificare eventuali errori, in modo da poterli correggere o, più in generale, apportare modifiche per il miglioramento dei vari sottosistemi.

# **Relazione con gli altri documenti**

Il Test Plan presenta una stretta connessione con il resto dei documenti che fino ad oggi sono stati prodotti, in quanto descrivono le generali funzionalità che devono essere osservate all’interno del sistema. Per tale ragione permetterà di rilevare eventuali differenze tra quanto veniva descritto, ovvero ciò che si desiderava realizzare, e ciò che effettivamente il sistema esegue.

* **Requirements Analysis Document (RAD):** I requisiti funzionali e non funzionali del sistema sono ciò che collegano i due documenti, ciò perché i test verranno eseguiti su ogni funzionalità tenendo conto delle specifiche espresse all’interno del RAD
* **System Design Documenti (SDD):** rappresenta l’architettura del sistema suddiviso in tre layer: Presentation Layer, Business Layer e Storage Layer. Il test deve tenere conto di queste suddivisioni.
* **Object Design Document (ODD):** I packages e le varie classi individuate all’interno del documento saranno progetto principale del testing

# **Panoramica del sistema**

VORWERK è una piattaforma web con un’interfaccia user-friendly e di facile utilizzo. Così come è stato definito all’interno del documento SDD, la struttura del sistema è suddivisa secondo un’architettura “Three-Tier”, ovvero con tre livelli: Presentation Layer, Business Layer e Storage Layer. Il livello più alto avrà il compito di interagire con il livello applicativo che a sua volta andrà ad interagire con il database di VORWERK.  
Il sistema è stato diviso in diversi sottosistemi più piccoli, organizzati per gestioni:

* Gestione Robot
* Gestione Account
* Gestione Premi
* Gestione Acquisto Robot
* Gestione Ricevimento Premi

Ognuna di esse presenta delle operazioni che dovranno essere testate nel corso del processo di testing del sistema.

# **Funzionalità da testare**

Di seguito sono riportate le funzionalità, suddivise in base alla gestione, che saranno sottoposte o meno al testing: (per quanto riguarda “Crea Ordine” di Gestione Acquisto Robot e Gestione Ricevimento Premi viene effettuato senza la carta di credito e l’indirizzo del cliente per mancanza di tempo)

* **Gestione Robot:**
  + Insert Robot
  + Update Robot
  + Delete Robot
  + Ricerca Robot
* **Gestione Account:**
  + Register Account
  + Login
  + Visualizza Carrello
  + Visualizza Ordini
* **Gestione Premi:**
  + Insert Premio
  + Update Premio
  + Delete Premio
* **Gestione Acquisto Robot:**
  + Aggiungi Robot al Carrello
  + Rimuovi Robot dal Carrello
  + Crea Ordine
* **Gestione Ricevimento Premi:**
  + Crea Ordine

# **Criteria Pass/Failed**

Il testing darà esito positivo se l’output che verrà prodotto sarà diverso dall’output atteso, ciò sta a significare che la fase di testing avrà successo se verranno individuati degli errori. Una volta individuati sarà attuata una fase di revisione del codice sorgente ed effettuate le modifiche. A questo punto verrà iterata la fase di testing per verificare che le correzioni apportate hanno risolto gli errori e che ciò non abbia avuto alcun impatto su gli altri componenti del sistema.  
Nel caso in cui l’output osservato sarà uguale all’oracolo definito, allora il testing darà esito negativo.

# **Approccio**

Il testing si divide in 3 fasi, nella prima fase testeremo i componenti uno ad uno, per poi passare a testare le funzionalità delle integrazioni dei sottosistemi e poi si testerà il sistema assemblato per verificare che rispecchi la vista del cliente:

* **Testing di unità:** nella quale utilizziamo la tecnica “Black-Box testing” e andremo ad esaminare le funzionalità dell’applicazione ed il comportamento input/output senza tener conto della struttura interna. Verranno create classi di equivalenza scegliendo per ognuna un test case in modo tale da ridurre la ridondanza.
* **Testing di integrazione:** dopo aver sottoposto ciascun componente al testing di unità ed aver corretto gli eventuali errori, essi verranno integrati in sottosistemi più grandi per sottoporli ad un test di integrazione. Useremo la strategia “Sandwich Testing” che suddivide il sistema in tre target: sopra al target, target e sotto al target.
* **Testing di sistema:** scopo di questa parte di testing è testare le funzionalità principali del sistema e con più probabilità di fallimento. Trattandosi di un sistema web-based verrà utilizzato per il testing di sistema, il tool SELENIUM, il quale si occupa di simulare l’interazione con il sistema dal punto di vista dell’utente.

# **Sospensione e ripresa**

## Criteri di sospensione

Si potrà definire sospesa la fase di testing del sistema quando si sarà raggiunta una qualità del prodotto tale che non si ecceda con i costi previsti per essa. Questo processo sarà quindi portato avanti finché non rischi di ritardare la consegna finale del progetto.

## Criteri di ripresa

Questa fase si potrà riprende nel caso in cui siano apportate modifiche o correzioni che generano errori o fallimenti, per cui i test case saranno nuovamente sottoposti al sistema in modo tale da assicurare una corretta risoluzione del problema.

# **Materiale per il testing**

Gli strumenti necessari per l’attività di test sono i computer, su cui è installato un browser, un server per lanciare il sistema e una database.

# **Test Cases**

## Gestione Robot

### Inserimento Robot/Modifica Robot

|  |  |
| --- | --- |
| **Parametro:** Nome  **Formato:** [A-Z a-z 0-9 ^\n]{5,30} (LNR= lunghezza nome robot) (FNR= formato nome robot) | |
| **Lunghezza[LNR]** | * <5 OR >30[error] * >=5 AND <=30 [LNR\_OK] | |
| **Formato[FNR]** | * Rispetta il formato [if LNR\_OK] [FNR\_OK] * Non rispetta il formato [if LNR\_OK] [error] | |

|  |  |
| --- | --- |
| **Parametro:** Descrizione  **Formato:** [A-Z a-z 0-9 . , ! ? % ^\n]{5,255} (LDR= lunghezza descrizione robot) (FDR= formato descrizione robot) | |
| **Lunghezza[LDR]** | * <5 OR >255[error] * >=5 AND <=255 [LDR\_OK] | |
| **Formato[FDR]** | * Rispetta il formato [if LDR\_OK] [FDR\_OK] * Non rispetta il formato [if LDR\_OK] [error] | |

|  |  |
| --- | --- |
| **Parametro:** Prezzo  **Formato:** [0-9]{1,3} (FPR= formato prezzo robot) (VPR= valore prezzo robot) | |
| **Formato[FPR]** | * Non rispetta il formato[error] * Rispetta il formato[FPR\_OK] | |
| **Valore[VPR]** | * Prezzo <=0 OR Prezzo >= 999 [error] * Prezzo >0 AND Prezzo <999 [VPR\_OK] | |

|  |  |
| --- | --- |
| **Parametro:** Quantità disponibile  **Formato:** [0-9]{1,3} (FQDR= formato quantità disponibile robot) (VQDR= valore quantità disponibile robot) | |
| **Formato[FQDR]** | * Non rispetta il formato[errore] * Rispetta il formato[FQDR\_OK] | |
| **Valore[VQDR]** | * Quantità disponibile <=0 OR Quantità disponibile >= 999 [error] * Quantità disponibile >0 AND Quantità disponibile <999 [VQDR\_OK] | |

|  |  |  |
| --- | --- | --- |
| Codice | Combinazione | Esito |
| TC\_2.0 - No Fields Form Insert Robot | LNR1-LDR1-FPR1-FQDR1 | Errore |
| TC\_2.1 - Only Name Robot | FNR1-LDR1-FPR1-FQDR1 | Errore |
| TC\_2.2 - Only Name and Price Robot | FNR1-LDR1-VPR2-FQDR1 | Errore |
| TC\_2.3 - Only Name, Price and Description Robot | FNR1-FDR1-VPR2-FQDR1 | Errore |
| TC\_2.4 - All Fields | FNR1-FDR1-VPR2-VQDR2 | Corretto |
| TC\_2.5.0 - NameFormatRobotException | LNR1-FDR1-VPR2-VQDR2 | Errore |
| TC\_2.5.1 - NameFormatRobotException | LNR1-FDR1-VPR2-VQDR2 | Errore |
| TC\_2.6.0 - PriceValueRobotException | FNR1-FDR1-VPR1-VQDR2 | Errore |
| TC\_2.6.1 - PriceValueRobotException | FNR1-FDR1-VPR1-VQDR2 | Errore |
| TC\_2.6.2 - PriceValueRobotException | FNR1-FDR1-FPR1-VQDR2 | Errore |
| TC\_2.7.0 - DescriptionFormatRobotException | FNR1-LDR1-VPR2-VQDR2 | Errore |
| TC\_2.7.1 - DescriptionFormatRobotException | FNR1-LDR1-VPR2-VQDR2 | Errore |
| TC\_2.8.0 - QuantityAvailableValueRobotException | FNR1-FDR1-VPR2-VQDR1 | Errore |
| TC\_2.8.1 - QuantityAvailableValueRobotException | FNR1-FDR1-VPR2-VQDR1 | Errore |
| TC\_2.8.2 - QuantityAvailableValueRobotException | FNR1-FDR1-VPR2-FQDR1 | Errore |
| TC\_5.0 - All Fields Update Robot | FNR1-FDR1-VPR2-VQDR2 | Corretto |

### Ricerca Robot

|  |  |
| --- | --- |
| **Parametro:** Nome  **Formato:** [A-Z a-z 0-9 ^\n] {1,} (LNR= lunghezza nome robot) (FNR= formato nome robot) | |
| **Lunghezza[LNR]** | * <1[error] * >=1 [LNR\_OK] | |
| **Formato[FNR]** | * Rispetta il formato [if LNR\_OK] [FNR\_OK] * Non rispetta il formato [if LNR\_OK] [error] | |

|  |  |  |
| --- | --- | --- |
| Codice | Combinazione | Esito |
| TC\_4.0 - No Field Ricerca Robot | LNR1 | Errato |
| TC\_4.1 - Insert Name and receive at least One Suggestion | FNR1 | Corretto |
| TC\_4.2 - Insert Name and receive No Suggestions | FNR1 | Corretto |

## Gestione Account

### Register Account

|  |  |
| --- | --- |
| **Parametro:** E-mail  **Formato:** [A-z 0-9\.\+\_-]+@[A-z 0-9\.\_-]+\.[A-z]{2,3} (FE= formato email) (EE= esiste email) | |
| **Formato[FE]** | * Non rispetta il formato [error] * Rispetta il formato [FE\_OK] | |
| **Esistente[EE]** | * Esiste nel database [if FE\_OK] [error] * Non esiste nel database [if FE\_OK] [EE\_OK] | |

|  |  |
| --- | --- |
| **Parametro:** password  **Formato:** ((?=.\*\d)(?=.\*[a-z])(?=.\*[A-Z]).{8,20}) (LP= lunghezza password) (FP= formato password) | |
| **Lunghezza[LP]** | * <8 OR >20[error] * >=8 AND <=20 [LP\_OK] | |
| **Formato[FP]** | * Non rispetta il formato [if LP\_OK] [error] * Rispetta il formato [if LP\_OK] [FP\_OK] | |

|  |  |
| --- | --- |
| **Parametro:** Data di Nascita  **Formato:** ([12]\\d{3}(\/)(0[1-9]|1[0-2])(\/)(0[1-9]|[12]\\d|3[01])) (LDN= lunghezza data nascita) (FDN= formato data nascita) | |
| **Lunghezza[LDN]** | * dd<1 and dd>31, mm<1 and mm>12, yyyy<1900 and yyyy>9999 [error] * 1<=dd<=31, 1<=mm<=12, 1<=yyyy<=9999 [LDN\_OK] | |
| **Formato[FDN]** | * Rispetta il formato [if LDN\_OK] * Non rispetta il formato [if LDN\_OK] [error] | |

|  |  |  |
| --- | --- | --- |
| Codice | Combinazione | Esito |
| TC\_1.0 - No Fields Registration | FE1 | Errato |
| TC\_1.1 Only Email | FE2-LP1 | Errato |
| TC\_1.2 Only Email and Password | FE2-FP2-FDN2 | Errato |
| TC\_1.3 All Fields | FE2-FP2-FDN1 | Corretto |
| TC\_1.4.0 FormatEmailException | FE1-FP2-FDN1 | Errato |
| TC\_1.4.1 FormatEmailException | FE1-FP2-FDN1 | Errato |
| TC\_1.4.2 EmailExistsException | EE1-FP2-FDN1 | Errato |
| TC\_1.5.0 FormatPasswordException | FE2-FP1-FDN1 | Errato |
| TC\_1.5.1 FormatPasswordException | FE2-FP1-FDN1 | Errato |
| TC\_1.5.2 FormatPasswordException | FE2-LP1-FDN1 | Errato |
| TC\_1.5.3 FormatPasswordException | FE2-LP1-FDN1 | Errato |
| TC\_1.5.4 FormatPasswordException | FE2-FP1-FDN1 | Errato |
| TC\_1.6.0 FormatDataDiNascitaException | FE2-FP1-non è maggiorenne | Errato |
| TC\_1.6.1 FormatDataDiNascitaException | FE2-FP1-FDN2 | Errato |

### Login

|  |  |
| --- | --- |
| **Parametro:** E-mail  **Formato:** [A-z 0-9\.\+\_-]+@[A-z 0-9\.\_-]+\.[A-z]{2,3} (EE= esiste email) (lE= lunghezza email) | |
| **Esistente[EE]** | * Esiste nel database [EE\_KO] * Non esiste nel database [EE\_OK] | |
| **Lunghezza[LE]** | * <1[error] * >=1 [LE\_OK] | |

|  |  |
| --- | --- |
| **Parametro:** password  **Formato:** ((?=.\*\d)(?=.\*[a-z])(?=.\*[A-Z])(?=.\*[@#$%]).{8,20}) (CP=combacia la password) (LP= lunghezza password | |
| **Combacia[CP]** | * Combacia la password [CP\_OK] * Non combacia la password [CP\_KO] | |
| **Lunghezza[LP]** | * <1[error] * >=1 [LP\_OK] | |

|  |  |  |
| --- | --- | --- |
| Codice | Combinazione | Esito |
| TC\_0.0 - No Fields Login | LE1 | Errore |
| TC\_0.1 - Only Email | LE2-LP1 | Errore |
| TC\_0.2 - All Field | EE2-CP1 | Corretto |
| TC\_0.3 - EmailNotExistsException | EE2 | Errore |
| TC\_0.4 - PasswordDoesNotMatchException | EE1-CP2 | Errore |

## Gestione Premi

### Inserimento Premio/Modifica Premio

|  |  |
| --- | --- |
| **Parametro:** Nome  **Formato:** [A-Z a-z 0-9 ^\n] {5, 30} (LNP= lunghezza nome premio) (FNP= formato nome premio) (ENP= esiste nome premio) | |
| **Lunghezza[LNP]** | * <5 OR >30[error] * >=5 AND <=30 [LNP\_OK] | |
| **Formato[FNP]** | * Rispetta il formato [if LNP\_OK] [FNP\_OK] * Non rispetta il formato [if LNP\_OK] [error] | |
| **Esistente[ENP]** | * Esiste nel database [if FNP\_OK] [error] * Non esiste nel database [if FNP\_OK] [EE\_OK] | |

|  |  |
| --- | --- |
| **Parametro:** Descrizione  **Formato:** [A-Z a-z 0-9 . , ! ? % ^\n {5, 255} (LDP= lunghezza descrizione premio) (FDP= formato descrizione premio) | |
| **Lunghezza[LDP]** | * <5 OR >255[error] * >=5 AND <=255 [LDP\_OK] | |
| **Formato[FDP]** | * Rispetta il formato [if LDP\_OK] [FDP\_OK] * Non rispetta il formato [if LDP\_OK] [error] | |

|  |  |
| --- | --- |
| **Parametro:** Punti  **Formato:** [0-9] {1,3} (FP= formato punti) (VP= valore punti) | |
| **Formato[FP]** | * Non rispetta il formato[error] * Rispetta il formato[FP\_OK] | |
| **Valore[VP]** | * Prezzo <=0 OR Prezzo >= 999 [error] * Prezzo >0 AND Prezzo <999 [VP\_OK] | |

|  |  |
| --- | --- |
| **Parametro:** Quantità disponibile  **Formato:** [0-9] {1,3} (FQDP= formato quantità disponibile premio) (VQDP= valore quantità disponibile premio) | |
| **Formato[FQDP]** | * Non rispetta il formato[errore] * Rispetta il formato[FQDP\_OK] | |
| **Valore[VQDP]** | * Quantità disponibile <=0 OR Quantità disponibile >= 999 [error] * Quantità disponibile >0 AND Quantità disponibile <999 [VQDP\_OK] | |

|  |  |  |
| --- | --- | --- |
| Codice | Combinazione | Esito |
| TC\_3.0 - No Fields Form Insert Premio | LNP1-LDP1-FP1-FQDP1 | Errato |
| TC\_3.1 - Only Name Premio | FNP1-LDP1-FP1-FQDP1 | Errato |
| TC\_3.2 - Only Name and Points Premio | FNP1-LDP1-VP2-FQDP1 | Errato |
| TC\_3.3 - Only Name, Points and Description Premio | FNP1-FDP1-VP2-FQDP1 | Errato |
| TC\_3.4 - All fields | FNP1-FDP1-VP2-VQDP2 | Corretto |
| TC\_3.5 - NameExistsPremioException | ENP1-FDP1-VP2-VQDP2 | Errato |
| TC\_3.6.0 - NameFormatPremioException | LNP1-FDP1-VP2-VQDP2 | Errato |
| TC\_3.6.1 - NameFormatPremioException | LNP1-FDP1-VP2-VQDP2 | Errato |
| TC\_3.7.0 - PointsValuePremioException | FNP1-FDP1-VP1-VQDP2 | Errato |
| TC\_3.7.1 - PointsValuePremioException | FNP1-FDP1-VP1-VQDP2 | Errato |
| TC\_3.7.2 - PointsValuePremioException | FNP1-FDP1-FP1-VQDP2 | Errato |
| TC\_3.8.0 - DescriptionFormatPremioException | FNP1-LDP1-VP2-VQDP2 | Errato |
| TC\_3.8.1 - DescriptionFormatPremioException | FNP1-LDP1-VP2-VQDP2 | Errato |
| TC\_3.9.0 - QuantityAvailableValuePremioException | FNP1-FDP1-VP2-VQDP1 | Errato |
| TC\_3.9.1 - QuantityAvailableValuePremioException | FNP1-FDP1-VP2-VQDP1 | Errato |
| TC\_3.9.2 - QuantityAvailableValuePremioException | FNP1-FDP1-VP2-FQDP1 | Errato |
| TC\_6.0 - All Fields Update Premio | FNP1-FDP1-VP2-VQDP2 | Corretto |

# **Glossario**

* **LNR:** lunghezza nome robot
* **FNR:** formato nome robot
* **LDR:** lunghezza descrizione robot
* **FDR:** formato descrizione robot
* **FPR:** formato prezzo robot
* **VPR:** valore prezzo robot
* **FQDR:** formato quantità disponibile robot
* **VQDR:** valore quantità disponibile robot
* **FE:** formato email
* **EE:** esiste email
* **LP:** lunghezza password
* **FP:** formato password
* **CP:** combacia password
* **LDN:** lunghezza data nascita
* **FDN:** formato data nascita
* **LNP:** lunghezza nome premio
* **FNP:** formato nome premio
* **ENP:** esiste nome premio
* **LDP:** lunghezza descrizione premio
* **FDP:** formato descrizione premio
* **FP:** formato punti
* **VP:** valore punti
* **FQDP:** formato quantità disponibile premio
* **VQDP:** valore quantità disponibile premio
* **FIR:** formato id robot
* **VIR:** valore id robot
* **FQS:** formato quantità scelta
* **VQS:** valore quantità scelta