**Università degli Studi di Salerno  
Corso di Ingegneria del Software**

****

**urNotes  
SDD**

**System Design Document**

****

Data: 5/12/2018

**Coordinatore del progetto:**

|  |  |
| --- | --- |
| Nome | Matricola |
| Francesco Parisi | 0512103582 |

**Partecipanti:**

|  |  |
| --- | --- |
| Nome | Matricola |
| Marco La Cortiglia | 0512103630 |
| Matteo Maiorano | 0512104578 |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Versione | Descrizione | Autore |
| 5/12/2018 | 0.1 | Sviluppo parte introduttiva e goal del sistema | Francesco Parisi |
| 11/12/2018 | 0.2 | Completamento e revisione documento | Tutti |
|  |  |  |  |

Indice

1. Introduzione 4

1.1 Obiettivi del Sistema 4

1.2 Design Goals 4

1.3 Definizioni, Acronimi e Abbreviazioni 5

1.4 Riferimenti 6

1.5 Panoramica ……………………………………………………………………………………6

2. Architettura Sistema Corrente 6

3. Architettura Sistema Proposto 7

3.1 Panoramica 7

3.2 Decomposizione Sottosistemi 4

**3.3** Mapping Hadware/Software..................................................................................................................10

**3.4** Gestione Dati Persistenti……...................................................................................................................10

**3.5** Controllo degli Accessi e Sicurezza……................................................................................................10

**3.6** Controllo Globale del Software…….......................................................................................................10

**3.7** Condizione Limite.......................................................................................................................................10

4.Servizi dei Sottosistemi………………………...................................................................................................10

  Glossario……………………………………………………………………………………………………………….…….…19

1. ***Introduzione***
   1. **Obiettivi del Sistema**

Gli Studenti iscritti all’Università necessitano l’uso di appunti, infatti molte volte permettono di affinare e approfondire concetti che risultano laboriosi e complessi.

Non è facile trovare l’appunto giusto e ciò comporta una perdita di tempo e denaro per lo studente.

Per far fronte a queste problematiche, si è pensato di realizzare una piattaforma web chiamata urNotes. L’Obiettivo è quello di offrire una piattaforma con interfaccia user-friendly e di facile utilizzo che permetta agli studenti in modo semplice ed efficiente l’acquisto di appunti validi e la possibilità di pubblicare appunti propri.

La piattaforma “urNotes” gestisce la ricerca, gli acquisti e la pubblicazione di appunti da parte degli studenti. Si tratta di un sito web, facile e intuitivo, in grado di agevolare i studenti. Dunque, l’obiettivo è quello di fornire un punto di incontro tra lo studente, permettendogli di risparmiare tempo nella ricerca, e la piattaforma, dove si potrà gestire in modo preciso e sicuro le varie richieste di pubblicazione inviate dagli studenti.

Per poter usufruire dei principali servizi offerti dalla piattaforma è necessaria la registrazione; gli studenti, potranno farlo autonomamente accedendo alla sezione dedicata.

L’Accesso consentito sarà diversificato tra i vari enti: studente e gestore del sito.

Una volta effettuato l'accesso, convalidato attraverso gli opportuni controlli effettuati dal sistema, lo studente sarà indirizzato alla sua pagina personale dove potrà visualizzare le operazioni che ha a disposizione. Lo studente ad esempio potrà aggiungere un appunto al carrello e completare l’acquisto. Potrà effettuare la propria richiesta di pubblicazione di un appunto inserendo le varie informazioni richieste, allegando il documento e confermarlo.

L’origine del nome del sistema serve a delineare il concetto di relazione tra lo studente e gli appunti, dove lo scopo non è solo quello di fornire materiale didattico allo studente per supportarlo nel percorso universitario, ma anche di dare la possibilità di condividere i propri appunti con altri studenti.

Il sistema può essere utilizzato da un qualsiasi utente dato che non vengono richieste competenze tecniche per l’uso. Gli unici presupposti sono il possesso di un dispositivo che abbia la possibilità di connettersi ad Internet.

* 1. **Design Goals**

I Design Goals sono organizzati in cinque categorie: Performance, Dependability, Cost, Maintenance, End User criteria. I Design Goals identificati per il nostro sistema sono i seguenti:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Priorità* | *Design Goal* | *Descrizione* | *Categoria* | *Origine* |
| *1* | *Memoria* | La memoria necessaria al funzionamento del sistema dipende dalla memoria utilizzata per il mantenimento del Database. | *Performance* | RNF\_3 Prestazioni |
| *2* | *Robustezza* | Il sistema informerà l’utente di eventuali errori nel caso di immissione di input non validi attraverso degli appositi messaggi. | *Dependability* | RNF\_2 Affidabilità |
| *3* | *Sicurezza* | L’accesso al sistema avviene mediante username e password. Inoltre, la sicurezza è garantita in quanto ogni utente può svolgere solo le operazioni a lui consentite. | *Dependability* | RNF\_2 Affidabilità |
| *4* | *Usabilità* | Il sistema sarà di facile comprensione e utilizzo, permettendo di effettuare in modo semplice e immediato le varie operazioni grazie a un’interfaccia user-friendly. | *End user* | RNF\_1 Usabilità |
| *5* | *Costo di sviluppo* | È stimato un costo complessivo di 180 ore per la progettazione e lo sviluppo del sistema (60 per ogni project member) | *Cost* | Business Case |
| *6* | *Estendibilità* | È possibile aggiungere nuove funzionalità al sistema, dettate dalle esigenze del cliente o dall’avvento di nuove tecnologie. | *Manteinance* | RNF\_4 Supportabilità |

* 1. **Definizioni, acronimi e abbreviazioni**

|  |  |
| --- | --- |
| ***urNotes*** | ***Nome del progetto*** |
| ***RAD*** | *Requirement Analysis Document* |
| ***SDD*** | System Design Document. |
| ***ODD*** | Object Design Document |
| **DB** | *Database* |
| **MySql** | Database Open Source basato sul linguaggio SQL, composto da un client a riga di comando e un server. |
| **DBMS** | Database Management System |
| ***RDBMS*** | Indica un database management system basato sul modello relazionale. |
| ***SQL*** | Linguaggio standardizzato per database basati sul modello relazionale |
| ***URL*** | Sequenza di caratteri che identifica univocamente l’indirizzo di una risorsa in Internet. |
| ***User Friendly*** | *Di facile utilizzo, anche per chi non è esperto.* |

* 1. **Riferimenti**
* Slide del corso, presenti sulla piattaforma e-learning,
* B. Bruegge, A.H. Dutoit, Object Oriented Software Engineering – Using UML, Patterns and Java, Prentice Hall
* [*https://www.studocu.com/it*](https://www.studocu.com/it) *Piattaforma dedicata alla condivisione di vari documenti (Appunti, Libri, Riassunti ecc.)*
* Documenti RAD\_urNotes

**1.5 Panoramica**

**Capitolo 1:** Contiene l’introduzione con l’obiettivo del sistema, i design goals e un elenco di definizioni, acronimi e abbreviazioni utili alla comprensione dell’intera documentazione.

**Capitolo 2**: Descrive, nel caso esista, le funzionalità offerte dal sistema corrente.

**Capitolo 3**: Viene presentata l’architettura del sistema proposto, in cui sarà gestita la decomposizione in sottosistemi, il mapping hardware/software, i dati persistenti, il controllo degli accessi e sicurezza, il controllo del flusso globale del sistema, le condizioni limite.

**Capitolo 4**: Vengono presentati i servizi dei sottosistemi.

È presente inoltre, un glossario, nel quale sono specificati e riassunti alcuni termini utilizzati nel documento al fine di rendere chiaro il significato di ogni capitolo e dei relativi paragrafi.

1. ***Architettura del Sistema corrente***

Al momento esistono varie piattaforme rivolte alla vendita di appunti universitari anche se, la difficoltà di ogni studente principalmente è sempre quella di non riuscire a trovare appunti validi per il proprio corso di studi.

Per agevolare gli studenti, abbiamo deciso di sviluppare una piattaforma online che permette di svolgere le varie procedure da casa, semplicemente registrandosi al sito. Questo, infatti, concederà allo studente la possibilità di ricercare rapidamente l’appunto o la dispensa al fine di trovare quella che più soddisfa le proprie esigenze e concludere l’acquisto.

Tramite un sistema di notifiche, inoltre, il gestore verrà informato nel caso in cui uno studente decide di richiedere la pubblicazione di un proprio appunto.

La procedura risulta più rapida e ordinata, evitando inoltre eventuali casi di errore dovuti alle numerose richieste.

1. ***Architettura del Sistema proposto***
   1. **Panoramica**

Il Sistema da noi proposto è un’applicazione web, in locale per motivi di sicurezza, che verrà sottoposto a reengineering allo scopo di aggiungere nuove funzionalità e di migliorare quelle già esistenti. Gli utenti saranno principalmente: studente e il gestore della piattaforma.

Tutti gli utenti potranno effettuare login e logout.

La Piattaforma metterà a disposizione diverse tipologie di funzionalità, a seconda del tipo di utente che effettua l'accesso. In particolare, lo studente potrà visualizzare e ricercare gli appunti di interesse, il carrello, inserire una recensione ad un appunto/dispensa, e ovviamente fare richiesta di pubblicazione di un appunto. Tale richiesta verrà inoltrata al gestore della piattaforma che, informato attraverso un apposito sistema di notifiche, provvederà a consultare e verificare i dati e il documento inviato.

Lo Stile architetturale scelto è di tipo Three-tier (“ a tre strati”). Questa, indica una particolare architettura software di tipo multi-tier per l’esecuzione di un'applicazione web che prevede la suddivisione dell'applicazione in tre strati dedicati rispettivamente alla interfaccia utente, alla logica funzionale e alla gestione dei dati persistenti.

* 1. **Decomposizione Sottosistemi**

La Decomposizione del sistema è composta da tre layer che si occupano di differenti funzionalità di seguito descritte:

* **Presentation Layer**: include tutte le interfacce grafiche e i boundary object che interfacciano con l’utente.
* **Application Logic Layer**: include tutti gli oggetti relativi al controllo e all’elaborazione dei dati.
* **Storage Layer**: effettua la memorizzazione, il recupero e l’interrogazione di oggetti persistenti.

I dati, i quali possono essere acceduti dall’application logic layer sono depositati su un database tramite DBMS.

Il Sistema è stato suddiviso in tre gestioni principali: Gestione Account,Gestione Documenti e Gestione Ordini.

Per ridurre la complessità della soluzione, decomponiamo il sistema in parti più piccole, chiamate sottosistemi. Le funzionalità sono state suddivise per area di gestione.

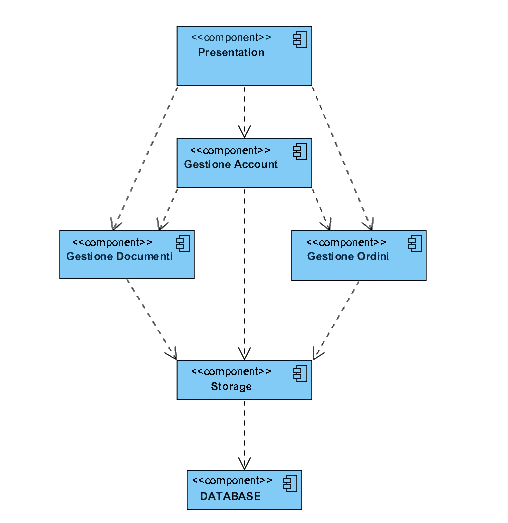
Riportiamo di seguito una breve descrizione delle funzionalità offerte da ciascun sottosistema:

**Sottosistema Gestione Account:** nel seguente sottosistema sono racchiuse tutte le funzionalità inerenti registrazione e autenticazione degli utenti sull'applicazione.

**Sottosistema Gestione Documenti:** nel seguente sottosistema sono racchiuse le funzionalità per la gestione dei documenti e la gestione delle richieste.

**Sottosistema Gestione Ordini:** nel seguente sottosistema sono racchiuse le funzionalità necessarie all’acquisto dei documenti.

In generale, dunque, il sistema si compone delle seguenti componenti che si occupano di gestirne aspetti e funzionalità diverse:



Il Livello Presentation prevede un unico sottosistema:

**Presentation Layer**: Gestisce l’interfaccia grafica e gli eventi generati dall’interazione con il sistema.

Il Livello Application Logic prevede a sua volta una suddivisione in 3 sottosistemi:

* Gestione Account
* Gestione Documenti
* Gestione Ordini

Il Livello Storage prevede un unico sottosistema:

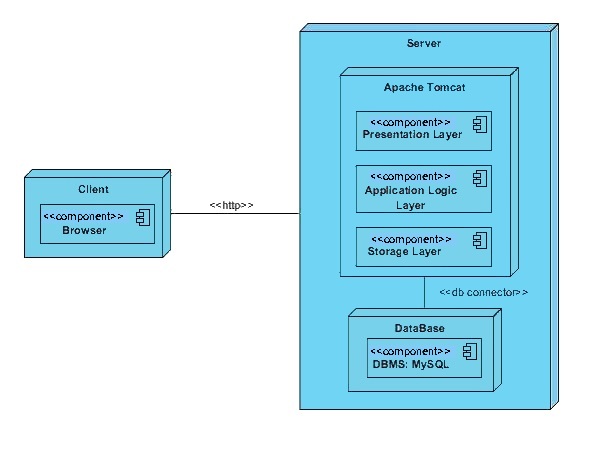
**Storage Layer**: Sistema che gestisce ed immagazzina i dati persistenti.

* 1. **Mapping hardware/software**

Il Sistema che si intende sviluppare utilizzerà una struttura hardware costituita da un Server che risponderà ai servizi richiesti dai client. Il client è una qualsiasi macchina attraverso il quale un utente può collegarsi, utilizzando una connessione internet, per accedere al sistema mentre la macchina server gestisce la logica e i dati persistenti contenuti nel database. Il client e il server saranno connessi tramite il protocollo HTTP, con il quale il client inoltra delle richieste al server e quest'ultimo provvederà a fornire i servizi richiesti.

Le componenti hardware e software necessarie per il client sono un computer dotato di connessione internet e di un web browser installato su di esso.

Per il server, invece, c'è necessità di una macchina con connessione ad Internet e con la capacità di immagazzinare una grande quantità di dati. La componente software necessaria è dunque un DBMS,nello specifico MySQL e Apache Tomcat, per consentire la comunicazione con più client.



**Web Server:** Il server utilizzato è Apache versione 8.5.

**Interface layer:** L’utente utilizza il sistema attraverso un Browser installato all’interno del suo calcolatore.

**Application Logic layer:** Il sistema è implementato in linguaggio HTML5 e java.

**Storage layer:** Rappresenta il collegamento con il server da parte del sistema e il suo compito è quello di gestire tutte le richiestedi accesso e modifiche sui dati permanenti presenti nel database.

**Database Server:** Il DBMS usato è MySQL.

* 1. **Gestione dati Persistenti**

**Utentitente**

|  |  |  |  |
| --- | --- | --- | --- |
| NOME | TIPO | NULL | KEY |
| username | Varchar(50) | NOT NULL | Chiave Primaria |
| nome | Varchar(20) | NOT NULL |  |
| cognome | Varchar(20) | NOT NULL |  |
| email | Varchar(50) | NOT NULL | Unique |
| password | Varchar(50) | NOT NULL |  |
| tipoUtente | Int | NOT NULL |  |
| Id\_utente | Integer(10) | NOT NULL | Unique |
| attivo | Int | NOT NULL |  |

**Tipo\_utenti**

|  |  |  |  |
| --- | --- | --- | --- |
| NOME | TIPO | NULL | KEY |
| tipoUtente | Int | NOT NULL | Chiave Primaria |
| nome | Varchar(20) | NOT NULL |  |

**Richieste**

|  |  |  |  |
| --- | --- | --- | --- |
| NOME | TIPO | NULL | KEY |
| id\_richiesta | Integer(10) | NOT NULL | Chiave Primaria |
| titolo | Varchar(50) | NOT NULL |  |
| pagine | Integer(10) | NOT NULL |  |
| nome\_materia | Varchar(50) | NOT NULL | Chiave Esterna |
| universita | Varchar(50) | NOT NULL |  |
| descrizione | Varchar(200) | NOT NULL |  |
| tipo | Varchar(50) | NOT NULL |  |
| data\_richiesta | Date | DEFAULT NULL |  |
| id\_utente | Integer(10) | NOT NULL | Chiave Esterna |
| attivo | Int | NOT NULL |  |

**Documenti**

|  |  |  |  |
| --- | --- | --- | --- |
| NOME | TIPO | NULL | KEY |
| codice | Integer(10) | AUTO INCREMENT NOT NULL | Chiave Primaria |
| titolo | Varchar(50) | NOT NULL |  |
| pagine | Integer(10) | NOT NULL |  |
| nome\_materia | Varchar(50) | NOT NULL | Chiave Esterna |
| università | Varchar(50) | NOT NULL |  |
| descrizione | Varchar(200) | NOT NULL |  |
| prezzo | Float | NOT NULL |  |
| tipo | Varchar(50) | NOT NULL |  |
| flag | Int | NOT NULL |  |

**Recensioni**

|  |  |  |  |
| --- | --- | --- | --- |
| NOME | TIPO | NULL | KEY |
| Id\_recensione | Integer(10) | AUTO INCREMENT NOT NULL | Chiave Primaria |
| descrizione | Varchar(50) | NOT NULL |  |
| username | Varchar(50) | NOT NULL | Chiave Esterna |
| codice | Integer(10) | DEFAULT NOT NULL | Chiave Esterna |
| flag | Int | NOT NULL |  |

**Materie**

|  |  |  |  |
| --- | --- | --- | --- |
| NOME | TIPO | NULL | KEY |
| nome | Varchar(50) | NOT NULL | Chiave Primaria |
| codice | Integer(10) | AUTO INCREMENT NOT NULL | Chiave Esterna |
| flag | Int | NOT NULL |  |

**Ordini**

|  |  |  |  |
| --- | --- | --- | --- |
| NOME | TIPO | NULL | KEY |
| serial\_Id | Integer(10) | AUTO INCREMENT NOT NULL | Chiave Primaria |
| Id\_vettore | Int(11) | DEFAULT NULL | Chiave Esterna |
| Id\_metodo | Int(11) | DEFAULT NULL | Chiave Esterna |
| Id\_utente | Int | DEFAULT NULL | Chiave Esterna |
| Id\_indirizzo | Int(11) | DEFAULT NULL | Chiave Esterna |
| totale\_documenti | Float | DEFAULT NULL |  |
| totale\_spedizione | Float | DEFAULT NULL | Chiave Esterna |
| totale\_ordine | Float | DEFAULT NULL |  |
| data\_ordine | datetime | DEFAULT NULL |  |
| attivo | Int | DEFAULT NULL |  |

**Ordini\_Vettori**

|  |  |  |  |
| --- | --- | --- | --- |
| NOME | TIPO | NULL | KEY |
| Id\_vettore | Int(11) | AUTO INCREMENT NOT NULL | Chiave Primaria |
| nome | Varchar(50) | DEFAULT NULL |  |
| descrizione | Varchar(50) | DEFAULT NULL |  |
| costo | Float | DEFAULT NULL |  |
| contrassegno | Int | DEFAULT NULL |  |
| attivo | Int | DEFAULT NULL |  |

**Ordini\_Metodi\_Pagamento**

|  |  |  |  |
| --- | --- | --- | --- |
| NOME | TIPO | NULL | KEY |
| Id\_metodo | Int(11) | AUTO INCREMENT NOT NULL | Chiave Primaria |
| nome | Varchar(50) | DEFAULT NULL |  |
| descrizione | Varchar(50) | DEFAULT NULL |  |
| In\_contanti | Int | DEFAULT NULL |  |
| attivo | Int | DEFAULT NULL |  |

**Ordini\_Documenti**

|  |  |  |  |
| --- | --- | --- | --- |
| NOME | TIPO | NULL | KEY |
| Id\_ordine | Int11) | AUTO INCREMENT NOT NULL | Chiave Primaria |
| serial\_Id | Int(11) | DEFAULT NULL | Chiave Esterna |
| codice | Int(11) | DEFAULT NULL | Chiave Esterna |
| titolo | Varchar(50) | DEFAULT NULL |  |
| quantita | Int(11) | DEFAULT NULL |  |
| prezzo\_documenti | Float | DEFAULT NULL |  |
| prezzo\_totale | Float | DEFAULT NULL |  |
| nome\_materia | Varchar(50) | DEFAULT NULL | Chiave Esterna |
| attivo | Int | DEFAULT NULL |  |

**Indirizzi**

|  |  |  |  |
| --- | --- | --- | --- |
| NOME | TIPO | NULL | KEY |
| id\_indirizzo | Int11) | AUTO INCREMENT NOT NULL | Chiave Primaria |
| nome | Varchar(50) | DEFAULT NULL |  |
| cognome | Varchar(50) | DEFAULT NULL |  |
| cap | Int | DEFAULT NULL |  |
| citta | Varchar(50) | DEFAULT NULL |  |
| provincia | Varchar(50) | DEFAULT NULL |  |
| telefono | Varchar(10) | DEFAULT NULL |  |
| cellulare | Varchar(10) | DEFAULT NULL |  |
| Id\_utente | Int(11) | DEFAULT NULL | Chiave Esterna |
| attivo | Int | DEFAULT 0 |  |

**Documenti\_Immagini**

|  |  |  |  |
| --- | --- | --- | --- |
| NOME | TIPO | NULL | KEY |
| Id\_immagine | Int11) | AUTO INCREMENT NOT NULL | Chiave Primaria |
| codice | Int(11) | DEFAULT NULL | Chiave Esterna |
| filename | Varchar(100) | DEFAULT NULL |  |
| Is\_default | Int | DEFAULT NULL |  |
| attivo | Int | DEFAULT NULL |  |

**Richieste\_File**

|  |  |  |  |
| --- | --- | --- | --- |
| NOME | TIPO | NULL | KEY |
| Id\_file | Int11) | AUTO INCREMENT NOT NULL | Chiave Primaria |
| Id\_richiesta | Int(11) | DEFAULT NULL | Chiave Esterna |
| filename | Varchar(100) | DEFAULT NULL |  |
| attivo | Int | DEFAULT NULL |  |

* 1. **Controllo degli accessi e sicurezza**

In urNotes ci sono attori che hanno il permesso di eseguire diverse operazioni. Per schematizzare al meglio il controllo degli accessi si è utilizzata una matrice degli accessi, dove le righe rappresentano gli attori e le colonne le classi. Ogni entry (attore, classe) contiene le operazioni consentite da quell’attore sulle istanze di quella classe.

|  |  |  |  |
| --- | --- | --- | --- |
| Sottosistema  Attore | Sottosistema  Account | Sottosistema  Documenti | Sottosistema  Ordini |
| Studente | -Registrazione;  -Login;  -Logout; | -Visualizza Documenti;  -Invia Richiesta Pubblicazione;  -Inserisci Recensione;  -Visualizza Recensioni;  -Aggiungi al Carrello; | -Completa Pagamento;  -Visualizza Ordine; |
| Gestore | -Login;  -Logout; | -Accetta/Rifiuta Richiesta;  -Visualizza Richiesta;  -Aggiungi Documento;  -Aggiungi Materia;  -Rimuovi Documento;  -Rimuovi Materia;  -Visualizza Documenti;  -Modifica info Documento;  -Scarica Documento Richiesta;  -Carica Documento; | -Visualizza Ordini |

**3.6 Controllo flusso globale del sistema**

La sequenza di azioni nel sistema urNotes non sono prestabilite: le funzionalità vengono scelte di volta in volta dall’utente, ragione per cui, il controllo del flusso globale del sistema è di tipo event-driven, ovvero guidato dagli eventi. Il ciclo principale attende un evento esterno e quando questo si verifica, viene spedito all’oggetto appropriato (sulla base dell’informazione associata all’evento). Non appena riceve una richiesta, il sistema invoca gli opportuni metodi sui sottosistemi coinvolti nell’operazione e invia l’output che verrà mostrato graficamente all’utente finale.

**3.7 Condizione Limite**

**3.7.1 Start-Up**

Per il primo start-up del sistema “urNotes” è necessario l'avvio di un web server che fornisca il servizio di un Database MySQL per la gestione dei dati persistenti. In seguito, tramite l'interfaccia di Login, sarà possibile autenticarsi tramite opportune credenziali (username e password).

Una volta effettuato l'accesso, “urNotes” presenterà all'utente la propria HomePage, dalla quale sarà possibile usufruire di tutte le operazioni che la piattaforma offre.

**3.7.2 Terminazione**

Al momento della corretta chiusura dell’applicazione, si ha la terminazione del sistema con un regolare Logout. Per consentire la corretta terminazione del server, il gestore del sistema dovrà effettuare la procedura di terminazione, dopo la quale nessun client potrà connettersi al sistema.

**3.7.3 Fallimento**

Possono verificarsi diversi casi di fallimento del sistema:

1. Nel caso di guasti dovuti al sovraccarico del database con successivo fallimento dello stesso è prevista come procedura preventiva il salvataggio periodico dei dati sotto forma di codice SQL per la successiva rigenerazione del DB.

2. Nel caso in cui si verifichi un'interruzione inaspettata dell'alimentazione non sono previsti metodi che ripristinino lo stato del Sistema precedente allo spegnimento non voluto.

3. Un altro caso di fallimento potrebbe derivare dal software stesso che causa una chiusura inaspettata dovuta ad errori commessi durante la fase di implementazione. Non essendo previste politiche correttive, l’unica o perazione consentita in questa particolare situazione è la chiusura del sistema e il suo successivo riavvio.

4. Un altro caso di fallimento potrebbe essere dovuto ad un errore critico nell'hardware, contro il quale non è prevista alcuna contromisura.

**Scenari**

|  |
| --- |
| **Nome Scenario:** Start-Up |
| **Partecipanti:** Marco, Gestore della Piattaforma |
| **Flusso di Eventi:**  1. Marco, gestore dela piattaforma urNotes, decide di avviare il sistema e quindi clicca sul pulsante “Avvia”.  2. Il sistema, con le opportune procedure di avvio, attiva il server e si mette in ascolto di eventuali richieste.  3. Il sistema notifica il successo della procedura. |
| **Nome** **Scenario:** Terminazione |
| **Partecipanti**: Marco, Gestore della Piattaforma |
| **Flusso di Eventi:**  1. Marco, gestore dela piattaforma urNotes, decide di arrestare il sistema e quindi clicca sul pulsante “Stop”.  2. Il sistema effettua le procedure necessarie per la terminazione del server. Da questo momento nessun client sarà in grado di connettersi al server  3. Il sistema notifica il successo della procedura. |

***4. Servizi dei Sottosistemi***

**Sottosistema Account**

|  |  |
| --- | --- |
| Servizi | Descerizione |
| Registrazione Studente | Permette allo studente di registrarsi al sistema e di utilizzarne i servizi. |
| Login | Permette allo studente di entrare nel sistema. |
| Logout | Permette allo studente di uscire dal sistema. |

**Sottosistema Documenti**

|  |  |
| --- | --- |
| Servizi | Descerizione |
| Visualizza Documento | Permette allo studente di visualizzare la pagina descrittiva del documento scelto. |
| Invia Richiesta Pubblicazione | Permette allo studente di inviare una richiesta di pubblicazione al gestore. |
| Inserisci Recensione | Permette allo studente di inserire una recensione. |
| Visualizza Recensioni | Permette allo studente di visualizzare le recensioni inserite da altri studenti. |
| Aggiungi al Carrello | Permette allo studente di aggiungere un appunto al carrello. |
| Rifiuta Richiesta | Permette al gestore di accettare o rifiutare una richiesta di pubblicazione inviata da uno studente. |
| Visualizza Richiesta | Permette al gestore di visualizzare una richiesta di pubblicazione inviata da uno studente. |
| Aggiungi Documento | Permette al gestore di aggiungere un nuovo documento al catalogo. |
| Aggiungi Materia | Permette al gestore di aggiungere un nuova materia al catalogo. |
| Rimuovi Documento | Permette al gestore di rimuovere un documento dal catalogo. |
| Rimuovi Materia | Permette al gestore di rimuovere un materia dal catalogo. |
| Modifica info Documento | Permette al gestore di modificare le informazioni relative a un documento. |
| Scarica Documento Richiesta | Permette al gestore di scaricare il documento per verificarlo. |
| Carica Documento | Permette al gestore di caricare nel catalogo il nuovo documento da lui inserito. |

**Sottosistema Ordini**

|  |  |
| --- | --- |
| Servizi | Descerizione |
| Completa Pagamento | Permette allo studente di completare il pagamento inserendo i propri dati. |
| Visualizza Ordine | Permette allo studente di visualizzare l’ordine da lui effettuato. |
| Visualizza Ordini | Permette al gestore di visualizzare gli ordini effettuati dagli studenti. |

***Glossario***

|  |  |
| --- | --- |
| **Sigla/Termine** | **Definizione** |
| urNotes | Nome del sistema che verrà sviluppato. |
| Piattaforma | Base software e/o hardware su cui sono sviluppare e/o eseguite applicazioni. |
| Login | La procedura di accesso a un sistema informatico con le credenziali registrate al sito web. |
| Logout | La procedura per scollegarsi da un sistema informatico a cui si era avuto accesso tramite un’operazione di login. |
| Sito web | Un insieme di pagine web correlate. |
| Notifica | Messaggistica istantanea grazie alla quale il messaggio perviene al destinatario. |
| Attore | Tipi di utenti del sistema; modella un’entità esterna che comunica col sistema. |
| UML | Insieme di linguaggi che, utilizzati congiuntamente, consentono di descrivere/modellare aspetti diversi di un sistema con un approccio Object Oriented. |
| Client | Ogni computer collegato al server e in grado di scambiare dati con esso. |
| Server | Computer di elevate prestazioni che in una rete fornisce un servizio agli altri elaboratori collegati, detti client. |
| Web-based | Termine utilizzato per indicare un’applicazione accessibile/fruibile via web. |