SYSTEM DESIGN DOCUMENT

*Repository GitHub:* [*PeppeVII/DocApp\_Classe03*](https://github.com/PeppeVII/DocApp_Classe03)

DOCAPP

|  |  |
| --- | --- |
| Versione | 1.1 |
| Data | 29/11/2022 |
| Destinatario | Prof. re Carmine Gravino |
| Presentato da | Bosso Francesco, D’Ambrosio Giuseppe, Napolitano Giuseppe, Russomando Antonio |
| Approvato da |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Ruolo** | **Nome** | **Matricola** | **Email** |
| Sviluppatore | G. D’Ambrosio | 0512110352 | g.dambrosio63@studenti.unisa.it |
| Sviluppatore | G. Napolitano | 0512111999 | g.napolitano80@studenti.unisa.it |
| Sviluppatore | A. Russomando | 0512110004 | a.russomando11@studenti.unisa.it |
| Sviluppatore | F. Bosso | 0512112335 | f.bosso4@studenti.unisa.it |

Revision History

| **Data** | **Versione** | **Cambiamenti** | **Autori** |
| --- | --- | --- | --- |
| **29/11/2022** | 1.0 | Stesura scheletro SSD | Russomando Antonio |
| **01/11/2022** | 1.1 | Aggiunta capitolo 1 | Russomando Antonio, Napolitano Giuseppe |
| **05/12/2022** | 1.2 | Aggiunta capitolo 2 e sezione 3.1 | Russomando Antonio, Napolitano Giuseppe,  D’Ambrosio Giuseppe |
| **08/12/2022** | 1.3 | Aggiunta sezione 3.2, 3.3 | Russomando Antonio, Napolitano Giuseppe, Bosso Francesco |

Sommario

1. Introduzione
   1. Scopo del sistema
   2. Design Goal
   3. Definizione, acronimi e abbreviazioni
   4. Riferimenti
   5. Panoramica del documento
2. Architettura del sistema corrente
3. Architettura del sistema proposto
   1. Panoramica
   2. Decomposizione in sottosistemi
   3. Mapping hardware/software
   4. Gestione dati persistenti
   5. Controllo degli accessi e sicurezza
   6. Controllo globale del software
   7. Condizione limite
4. Servizi dei sottosistemi
5. Introduzione
   1. Scopo del sistema

Lo scopo del sistema che si intende sviluppare è quello di facilitare e migliorare lo studio degli studenti universitari, in particolare degli studenti-lavoratori. Ad oggi i sistemi esistenti sul mercato offrono servizi simili ma limitati e molto spesso si possono accedere a questi servizi mediante l’acquisto di punti attraverso micro-transazioni.

L’obiettivo di DocApp è quello di ottimizzare il tempo di studio degli studenti universitari, permettendo loro di avere tutto a disposizione gratuitamente attraverso un sistema user-friendly.

DocApp è un’applicazione mobile alla quale avranno accesso l’amministratore di sistema, gli studenti e i creators. Possiamo dividere il sistema in due macro aree:

* **Area studente**: visualizzazione post, visualizzazione documenti, chat end-to-end, creazione post, scaricare documenti, caricare documenti;
* **Area amministratore**: visualizzazione post, azione sui documenti postati, visualizzazione documenti, visualizzazione studenti.
  1. Design Goals

I design goal identificano le qualità su cui deve essere focalizzato il sistema. È importante formalizzare esplicitamente poiché ogni importante decisione di design deve essere fatta seguendo lo stesso insieme di criteri.

I design goal sono suddivisi nelle seguenti categorie:

* **Performance**: includono i requisiti imposti sul sistema in termini di spazio e velocità;
* **Dependability**: determinano quanto sforzo deve essere speso per minimizzare i fallimenti del sistema e le loro conseguenze;
* **Maintenance**: determinano quanto lavoro serve per modificare ulteriormente il sistema dopo il suo rilascio;
* **End User**: includono qualità che sono desiderabili dal punto di vista dell’utente, ma che non sono state coperte dai criteri di Performance e Dependability.

Inoltre, ciascun design goal viene descritto mediante le seguenti informazioni:

* **Rank**, specifica un valore di priorità compreso tra 1 (massima) e 16 (minima);
* **ID design goal,** specifica un identificatore univoco e un nome del design goal;
* **Descrizione,** specifica una descrizione del design goal;
* **Categoria,** specifica la categoria di appartenenza del design goal;
* **RNF di riferimento,** specifica il RNF a cui fa riferimento il design goal.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Rank** | **ID** | **Descrizione** | **Categoria** | **RNF di Origine** |
| **1** | DG\_01 Facilità d’uso | Il sistema deve risultare facilmente comprensibile e utilizzabile anche da chi è meno esperto. | End User | RNF\_1 |
| **2** | DG\_02 Tempi di Risposta | Il sistema deve garantire un tempo di risposta non superiore a 10 secondi. | Performance | RNF\_8 |
| **1** | DG\_03 Quantità di dati | Il sistema deve essere in grado di gestire tutti i documenti caricati dagli utenti, servirà quindi una grande complessità spaziale. | Performance | RNF\_7 |
| **7** | DG\_ 04 Fallimenti del sistema | Il sistema deve comportarsi in maniera adeguata in situazioni di fallimento avvisando l’utente tramite messaggi di avviso. | Dependability | RNF\_4 |
| **1** | DG\_05 Sicurezza del sistema | Il sistema deve mantenere un livello di sicurezza necessario affinché eventuali accessi minacciosi non comportino una perdita di dati sensibili o una modifica di essi. | Dependability | RNF\_3 |
| **3** | DG\_ 06 Efficienza del sistema | Il sistema deve essere in grado di offrire i propri servizi correttamente anche con un elevato numero di utenti connessi contemporaneamente. | Performance | RNF\_6 |
| **5** | DG\_ 07 Scalabilità | Il sistema deve essere scalabile, e facilmente adattabile a nuove funzionalità del dominio applicativo. | Maintenance | RNF\_9 |
| **2** | DG\_ 08 Installazione | Il sistema potrà essere installato tramite semplice pacchetto apk. | End User | RNF\_14 |

* + 1. Design trade-off

Efficienza vs Tempo di risposta

Il sistema deve essere in grado di offrire i propri servizi correttamente anche quando sovraccaricato da un alto numero di richieste a discapito del tempo di risposta che potrà variare conseguentemente.

Privacy e sicurezza vs Performance

Il sistema tramite una chat encryptata end-to-end garantirà il massimo livello di riservatezza nelle conversazioni tra utenti ma questo comporterà un carico maggiore sul sistema e conseguentemente anche nei tempi di invio e di ricezione dei messaggi stessi.

Performance e memoria client vs Tempo di risposta

Il sistema deve garantire risposte rapide all’utente evitando di sovraccaricare il dispositivo mobile. Ciò significa che il carico verrà applicato sul sistema distribuito che effettuerà il processing delle operazioni e fornirà risposte esaustive, rapide e leggere al client.

* 1. Definizione, acronimi e abbreviazioni

Nella sezione seguente vengono riportate alcune definizioni, acronimi e abbreviazioni presenti in tutto il documento:

* **SDD**: System Design Document
* **RAD**: Requirements Analysis Document
* **RNF**: Requisito non funzionale
* **UI**: Interfaccia Utente
  1. Riferimenti

Libro:

Object-Oriented Software Engineering (Using UML, Patterns, and Java) Third Edition

Autori: Bernd Bruegge, Allen H. Dutoit

Documenti:

RAD\_DocApp

* 1. Panoramica del documento

Il documento seguente è suddiviso in quattro capitoli:

***Capitolo 1***

Contiene una descrizione del sistema nella sua interezza, la descrizione dei design goal, un elenco di definizioni, acronimi e abbreviazioni utili alla comprensione della documentazione ed eventuali riferimenti ad altri documenti.

***Capitolo 2***

Contiene la descrizione di un sistema similare di riferimento.

**Capitolo 3**

Contiene la descrizione in modo dettagliato del sistema che verrà realizzato. In particolare verrà descritto come sarà suddiviso in sottosistemi, come avverrà il mapping hardware/software, come saranno gestiti i dati persistenti, il controllo degli accessi e sicurezza, il controllo globale del software e le condizioni limite.

**Capitolo 4**

Contiene la rappresentazione dei servizi dei sottosistemi del sistema generale.

1. Architettura del sistema corrente

Sistemi simili sono già esistenti sul mercato (Skuola.net e StuDocu), la differenza è che il sistema che si intende creare è totalmente gratuito, senza necessità di abbonarsi a servizi ed è utilizzabile da tutte le università. I costi di sviluppo del sistema si autososterranno tramite le ads.

1. Architettura del sistema proposto
   1. Panoramica

Il sistema (DocApp) rientra nel campo della Greenfield Engineering e verrà sottoposto a reengineering con lo scopo di aggiungere nuove funzionalità e migliorare quelle esistenti.

Il sistema è rivolto agli studenti universitari di tutte le facoltà: potranno effettuare login e logout; avranno la possibilità di registrarsi al sistema e potranno decidere di caricare documenti o chiederne ai colleghi attraverso la piattaforma e se vorranno potranno anche lasciare un feedback al documento visualizzato.

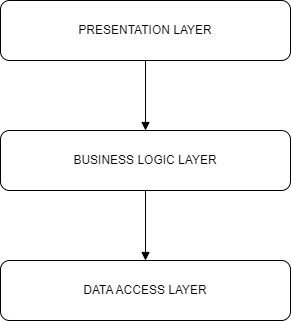
Il sistema verrà implementato seguendo un pattern architetturale di tipo three tier che prevede la suddivisione dell’applicazione in tre diversi moduli dedicati rispettivamente all’UI, alla logica di business e alla gestione dei dati persistenti.

Utilizzando quest’architettura, ciascuno dei tre moduli può essere modificato o sostituito indipendentemente dagli altri conferendo scalabilità e alta manutenibilità all’applicazione.

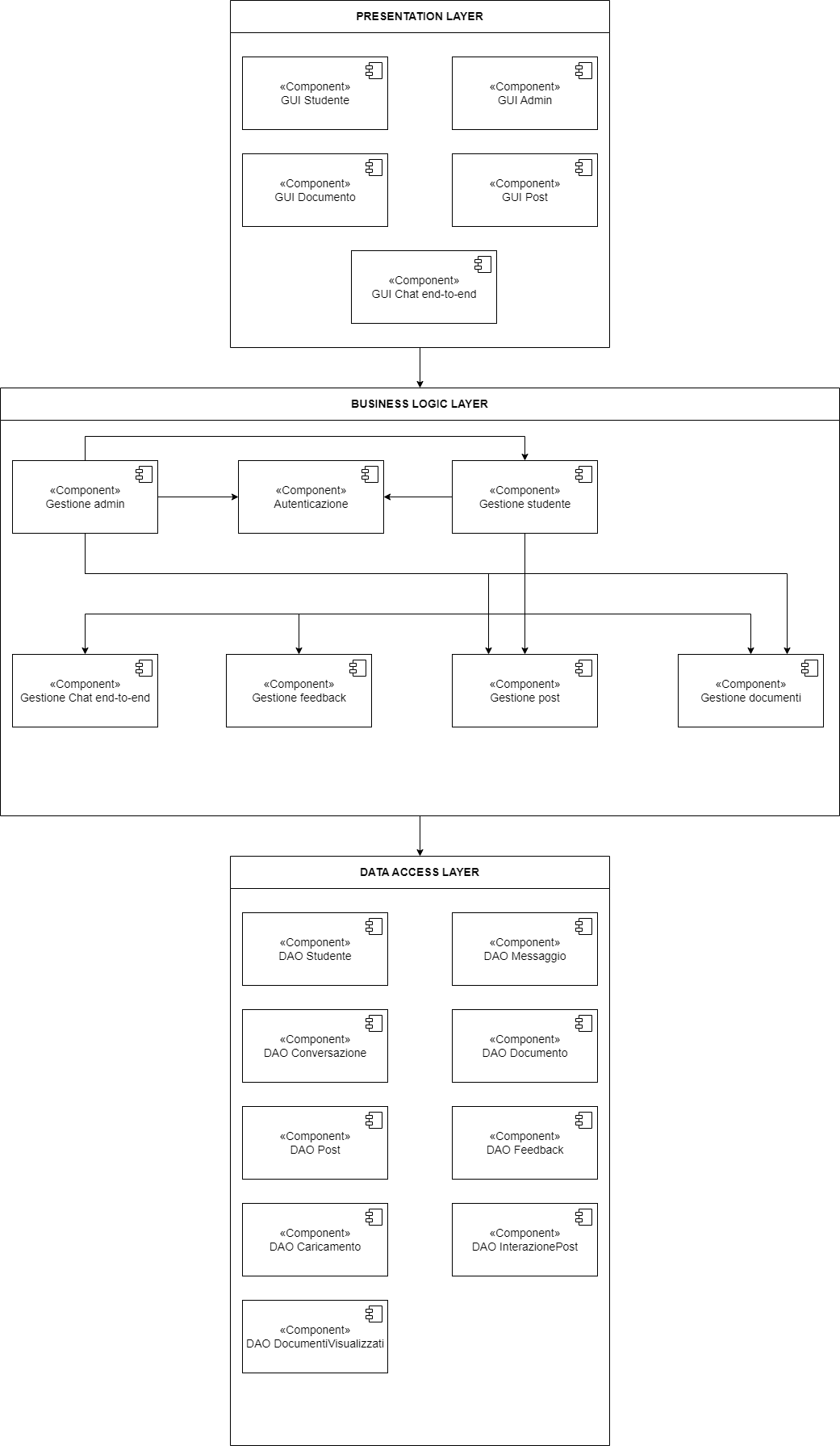
3.2 Decomposizione in sottosistemi

Il sistema sarà scomposto in tre livelli differenti, seguendo la logica architetturale del pattern three tier. Si avranno i seguenti livelli con annessi sottosistemi:

* **Presentation layer**: raggruppa gli elementi di interfaccia grafica;
* **Business Logic layer**: elabora le informazioni raccolte nel Presentation layer a volte rispetto alle informazioni presenti nel Data Access layer utilizzando le logiche progettuali. Esso può anche modificare le informazioni contenute nel Data Access layer;
* **Data Access layer**: è il layer in cui vengono archiviate e gestite le informazioni elaborate dall’applicazione.



I sottosistemi raggruppati per ogni layer sono rappresentati nel component diagram seguente:



Il Presentation layer è suddiviso in cinque sottoinsiemi:

* GUI Admin
* GUI Studente
* GUI Documento
* GUI Post
* GUI Chat end-to-end

Il Business Logic layer è suddiviso in sette sottosistemi:

* Gestione admin
* Gestione studente
* Gestione chat end-to-end
* Gestione feedback
* Gestione post
* Gestione documenti
* Gestione di autenticazione

Il Data Access layer è suddiviso in otto sottosistemi:

* DAO Studente
* DAO Post
* DAO Messaggio
* DAO Conversazione
* DAO Feedback
* DAO DocumentiVisualizzati
* DAO InterazionePost
* DAO Documento
* DAO Caricamento

**Deployment Diagram**

L’utente (Client) richiede le funzionalità tramite l’interfaccia che il sistema mette a

disposizione a patto che si possieda un dispositivo con versione Android uguale o superiore alla Android Oreo (API 26-27), in modo che le funzioni definite dal sistema possano eseguire in maniera corretta. Il Client si connette al Sistema sul quale vengono eseguite le funzioni apposite

richieste dal client. La parte DBMS racchiude e gestisce la persistenza dei dati. Il Sistema viene eseguito sul web server Tomcat.

* 1. Mapping hardware/software

Il sistema che si desidera sviluppare utilizzerà una struttura hardware composta da un Server che risponderà ai servizi richiesti dal client. Il client è una macchina attraverso la quale un utente può collegarsi, usando una connessione a internet, per accedere al sistema mentre la macchina server gestisce la logica e i dati persistenti inseriti nel database. Client e server saranno connessi tramite il protocollo HTTP, con cui il client inoltra richieste al server e provvederà a fornire i servizi richiesti. Le componenti hardware e software di cui ha bisogno il client sono un qualsiasi dispositivo mobile dotato di connessione internet. Per quanto riguarda il server, vi è la necessità di avere a disposizione una macchina avente connessione a internet e con capacità di immagazzinare grandi quantità di dati. La componente di cui ci si ha bisogno è un DBMS, nel nostro caso MySQL, per consentire il salvataggio dei dati in maniera persistente. (eventualmente da modificare)

* 1. Gestione dati persistenti

Per la memorizzazione dei dati si utilizza un database relazionale per consentire brevi tempi di risposta e ridurre i limiti di spazio di archiviazione. È possibile ripristinare lo stato del database in caso di danni software o hardware attraverso una copia dei dati fatta periodicamente. I dati presenti nel database sono privatizzati, vale a dire che il DBMS consente un accesso protetto, di conseguenza con operazioni diverse, l’utente, può accedere a diverse sezioni del database. Di seguito riportato il class diagram solo delle informazioni che restano persistenti nel database, non sono stati inseriti i documenti scaricati in quanto verranno salvati sul dispositivo.

Lo schema seguente evidenzia le chiavi e in che modo le istanze sono collegate tra di loro.

**Studente**

|  |  |  |  |
| --- | --- | --- | --- |
| NOME | TIPO | CONSTRAINTS | KEY |
| nickname | varchar(16) | not null, unique |  |
| email | varchar(40) | not null | primary key |
| password | varchar(64) | not null, unique |  |
| salt | varchar(25) | not null, unique |  |
| is\_admin | bool | not null |  |

**Documento**

|  |  |  |  |
| --- | --- | --- | --- |
| NOME | TIPO | CONSTRAINTS | KEY |
| id\_documento | bigint | not null | primary key |
| nome | varchar(30) | not null |  |
| descrizione | text | not null |  |
| universita | varchar(40) | not null |  |
| facolta | varchar(40) | not null |  |
| corso\_di\_studio | varchar(30) | not null |  |
| dimensione | bigint | not null |  |

**Feedback**

|  |  |  |  |
| --- | --- | --- | --- |
| NOME | TIPO | CONSTRAINTS | KEY |
| documento | bigint | not null | foreign key, primary key |
| studente | varchar(30) | not null | foreign key, primary key |

**Caricamento**

|  |  |  |  |
| --- | --- | --- | --- |
| NOME | TIPO | CONSTRAINTS | KEY |
| documento | bigint | not null | foreign key, primary key |
| studente | varchar(30) | not null | foreign key ,primary key |

**Post**

|  |  |  |  |
| --- | --- | --- | --- |
| NOME | TIPO | CONSTRAINTS | KEY |
| id\_post | bigint | not null | primary key |
| titolo | varchar(30) |  |  |
| testo | text | not null |  |
| is\_domanda | bool | not null |  |

**InterazionePost**

|  |  |  |  |
| --- | --- | --- | --- |
| NOME | TIPO | CONSTRAINTS | KEY |
| post | bigint | not null | foreign key, primary key |
| studente | varchar(30) | not null | foreign key, primary key |

**Messaggio**

|  |  |  |  |
| --- | --- | --- | --- |
| NOME | TIPO | CONSTRAINTS | KEY |
| id\_messaggio | bigint | not null | primary key |
| conversazione | bigint | not null | foreign key, primary key |
| studente | varchar(40) | not null | foreign key |
| testo | text | not null |  |
| timestamp | timestamp | not null |  |

**Conversazione**

|  |  |  |  |
| --- | --- | --- | --- |
| NOME | TIPO | CONSTRAINTS | KEY |
| id\_conversazione | bigint | not null | primary key |
| nome\_conversazione | varchar(25) | not null |  |
| studente | varchar(30) | not null | foreign key |

**DocumentiVisualizzati**

|  |  |  |  |
| --- | --- | --- | --- |
| NOME | TIPO | CONSTRAINTS | KEY |
| documento | bigint | not null | foreign key, primary key |
| studente | varchar(30) | not null | foreign key, primary key |

* 1. Controllo degli accessi e sicurezza

Con il sistema “DocApp” possono interagire diversi attori che possono eseguire diverse funzioni. Utilizziamo quindi una matrice degli accessi per capire quali attori possono eseguire quali funzioni. Nella seguente tabella verranno riportate:

* in alto, un’astrazione delle istanze delle classi del nostro sistema;
* sul lato sinistro, gli attori;
* la cella che incrocia attore e istanza rappresenta il permesso che quell’attore ha su quella istanza

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Studente** | **Admin** | **Documento** | **Post** | **Conversazione** |
| **Utente non registrato** | 1)registra |  |  |  |  |
| **Studente** | 1) modifica dati personali |  | 1)visualizza  2)Inserisce feedback | 1)Visualizza post  2)Commenta post | 1) Partecipa  2)Chatta |
| **Admin** | 1)Bannare |  | 1)Elimina | 1)Elimina |  |

* 1. Controllo globale del software

Il flusso del sistema di “DocApp” richiede una continua interazione dell’utente, per cui il controllo del flusso globale del sistema è di tipo event-driven, vale a dire che le azioni del sistema sono guidate dall’input dell’utente. Per quanto riguarda la concorrenza, le funzionalità offerte dal Server, garantiscono un’interazione concorrente con tutti gli utenti connessi al sistema. Infatti, ogni utente connesso al sistema, tramite il suo

dispositivo, avrà un thread dedicato tramite il quale il server interagirà con lui.

* 1. Condizione limite

**Start-up**

Lo start-up del sistema prevede l’avvio del server nel quale il sistema è installato e l’avvio del DBMS per accedere ai dati persistenti memorizzati nel database. Quando sia il Server che DBMS sono in esecuzione, il sistema carica in memoria centrale le principali applet attraverso le quali gli utenti possono effettuare le operazioni. Dopo l’avvio del sistema l’utente può interagire con esso.

**Shut-down**

Quando il sistema deve essere arrestato, il gestore del sistema termina l’esecutivo del server.

Quando ciò avviene tutte le risorse che il sistema utilizza (connessione al database e connessione alla rete) vengono rilasciate, e nessun utente potrà più connettersi al sistema.

**Fallimento**

1. Nel caso in cui si presentasse un’ interruzione inaspettata dell’alimentazione, non vi sono metodi che possano ripristinare lo stato del sistema precedente allo spegnimento non voluto. Qualsiasi transazione con il database viene annullata e viene ripristinato lo stato consistente più recente delle informazioni persistenti.

2. In caso di guasti dovuti al sovraccarico di informazioni al database, la rete viene congestionata. Il Server percepirà questo stato e inviterà tutti i client connessi di riprovare ad effettuare le operazioni in un secondo momento.

3. Nel caso in cui si presentasse una chiusura inaspettata del software, dovuta a errori avvenuti durante la fase di implementazione, il server risponderà con una pagina di errore.

4. Nel caso in cui si verificasse un sovraccarico di richieste al server, questo rimarrà congestionato.

5. Se un utente invia al server informazioni errate (volutamente o meno), o che non permettono la corretta esecuzione di un’operazione, il server risponderà con una pagina di errore.

6. Nel caso di errore critico dell’hardware, non è prevista una soluzione.

1. Servizi dei sottosistemi