



System Design Document

Just Traditions



| | |
|---------------|---|
| Riferimento | 2022_SSD_C08 |
| Versione | 1.1 |
| Data | 07/02/2023 |
| Destinatario | Top managment |
| Presentato da | G. Sgambato, L. Sorrentino, M. Spagnuolo, D. Avella, G. Miele |
| Approvato da | F. Cirillo, G. Brescia |



Team Members

| Nome | Ruolo | Acronimo | Informazioni di contatto |
|--------------------|-----------------|----------|--|
| Gerardo Brescia | Project Manager | GB | g.brescia3@studenti.unisa.it |
| Franco Cirillo | Project Manager | FC | f.cirillo30@studenti.unisa.it |
| Giovanni Miele | Team Member | GM | g.miele18@studenti.unisa.it |
| Michele Spagnuolo | Team Member | MS | m.spagnuolo26@studenti.unisa.it |
| Lorenzo Sorrentino | Team Member | LS | l.sorrentino66@studenti.unisa.it |
| Domenico Avella | Team Member | DA | d.avella11@studenti.unisa.it |
| Giuseppe Sgambato | Team Member | GS | g.sgambato1@studenti.unisa.it |



Revision History

| Data | Versione | Descrizione | Autori |
|------------|----------|--|------------|
| 23/11/22 | 0.1 | Aggiunta "Obiettivo del Sistema" | GM, MS |
| 23/11/22 | 0.2 | Aggiunta "Design Goals & Trade Offs" | GM, MS |
| 23/11/22 | 0.3 | Aggiunta "Architettura di Sistemi simili" | DA, LS, GS |
| 23/11/22 | 0.4 | Aggiunta "Architettura del Sistema proposto" | DA, LS, GS |
| 24/11/22 | 0.5 | Aggiunta "Mapping Hardware/Software" | GM |
| 30/11/22 | 0.6 | Aggiunta "Gestione dati persistenti" | TUTTI |
| 01/12/22 | 0.7 | Aggiunta "Controllo degli accessi e sicurezza" | DA |
| 02/12/22 | 0.8 | Aggiunta "Controllo Flusso Globale" | GM |
| 03/12/22 | 0.9 | Aggiunta "Condizione limite" | MS |
| 05/12/22 | 0.10 | Aggiunta "Funzionalità dei Sottosistemi" | TUTTI |
| 10/12/2022 | 1.0 | Revisione | TUTTI |
| 07/02/2023 | 1.1 | Modifica condizioni limite | LS DA |



Sommario

| | | |
|---------|---------------------------------------|----|
| 1 | Introduzione | 5 |
| 1.1 | Obiettivo del Sistema | 5 |
| 1.2 | Design Goals & Trade-offs | 5 |
| 1.3 | Definizioni, Acronimi e Abbreviazioni | 8 |
| 1.4 | Riferimenti | 8 |
| 1.5 | Panoramica | 9 |
| 2 | Architettura di Sistemi simili | 9 |
| 3 | Architettura del Sistema proposto | 9 |
| 3.1 | Panoramica | 9 |
| 3.2 | Decomposizione in sottosistemi | 10 |
| 3.2.1 | Diagramma architetturale | 12 |
| 3.3 | Mapping hardware/software | 12 |
| 3.4 | Gestione dati persistenti | 14 |
| 3.4.1 | Progettazione concettuale | 14 |
| 3.4.1.1 | Schema ER | 14 |
| 3.4.2 | Progettazione logica | 14 |
| 3.4.2.1 | Ristrutturazione Schema ER | 15 |
| 3.4.3 | Dizionario dei dati | 15 |
| 3.5 | Controllo degli accessi e sicurezza | 18 |
| 3.6 | Controllo flusso globale del sistema | 20 |
| 3.7 | Condizione limite | 20 |
| 4 | Funzionalità dei sottosistemi | 25 |
| 4.1 | Sottosistema Gestione Profili | 25 |
| 4.2 | Sottosistema Autenticazione | 26 |
| 4.3 | Sottosistema Gestione Annunci | 26 |
| 4.4 | Sottosistema Visualizzazione Annunci | 27 |
| 4.5 | Sottosistema Prenotazioni | 27 |
| 5 | Glossario | 28 |



1 Introduzione

1.1 Obiettivo del Sistema

La piattaforma Just Traditions nasce dall'idea di valorizzare il turismo responsabile e sostenibile in Italia, favorendo la positiva interazione tra industria del turismo, comunità locali e viaggiatori.

Il sistema proposto parte dall'idea di poter fornire una piattaforma di facile utilizzo per l'utente e disegnare un'idea di turismo innovativa, che ponga al centro dell'attenzione la User-Experience, permettendo ad artigiani e clienti di interfacciarsi in maniera semplice, immediata ed efficace.

Just Traditions sarà reso accessibile ad ogni tipo di utente tramite un semplice web browser con estrema facilità, permettendo di prenotarsi ad una visita in pochi click.

A tale scopo, per agevolare le ricerche degli utenti, sarà possibile restringere la ricerca degli annunci a quelli di una determinata provincia; in modo tale che ogni utente possa conoscere immediatamente le attività della propria zona.

Just Traditions si occuperà di garantire la qualità degli annunci presenti sulla piattaforma, tramite l'approvazione di ciascun annuncio proposto, effettuata minuziosamente dagli amministratori del sistema.

Inoltre, l'applicazione web attingerà informazioni da un database relazionale con lo scopo di poter gestire e manipolare dati persistenti fondamentali e necessari per il funzionamento del sistema.

La piattaforma deve garantire un sistema di autenticazione efficace (username e password) con controlli di sicurezza al fine di preservare dati sensibili degli utenti da accessi non autorizzati.

1.2 Design Goals & Trade-offs

Nella presente sezione si andranno a presentare i Design Goals, ovvero le qualità sulle quali il sistema deve essere focalizzato, formalizzati esplicitamente così che qualsiasi importante decisione di design può essere fatta consistentemente seguendo lo stesso insieme di design goals.

La tabella sottostante riporta l'insieme di tutti gli obiettivi di design per il sistema proposto. Ciascun design goal è formato da:



- **ID:** identificatore univoco
- **Rank:** specifica un valore di priorità da 1 a 14 (1 massima, 14 minima)
- **Descrizione:** descrizione del design goal.
- **Categoria:** la categoria di appartenenza del design goal che possono essere:
 - **Performance:** includono i requisiti di spazio e velocità imposti sul sistema.
 - **Dependability:** determinano quanto sforzo deve essere speso per minimizzare i fallimenti del sistema (crash, falle di sicurezza) e le loro conseguenze.
 - **Maintenance:** determina quanto sforzo è necessario per modificare 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.
 - **Cost:** includono i requisiti di costo che il sistema dovrà avere in termini di sviluppo.
- **RNF di origine:** le sigle indicate corrispondono all'identificativo del requisito non funzionale dal quale l'obiettivo è stato ricavato.

| ID | Rank | Descrizione | Categoria | RNF di origine |
|---------------------------------|------|--|---------------|--------------------|
| DG_1 Sicurezza | 1 | Il sistema deve garantire un sistema di autenticazione efficiente, evitando l'accesso ai dati personali ad utenti non autorizzati, e garantire le funzionalità ad ogni utente consistentemente con il proprio ruolo. | Dependability | RNF_A_1 RNF_A_5 |
| DG_2 Sicurezza | 2 | Il sistema deve garantire sicurezza per le password degli utenti registrati che dovranno essere salvate in forma criptata. | Dependability | RNF_A_6 |



| | | | | |
|--|----|--|---------------|----------|
| DG_3 Costo di sviluppo | 3 | Lo sviluppo del sistema deve avere costi ridotti sia in termini di risorse umane (per cui è fissato un tetto di 50 ore-lavoro), sia in termini economici. | Cost | RNF_TP_1 |
| DG_4 Robustezza | 4 | Il sistema deve essere in grado di gestire eventuali errori di immissione da parte dell'utente preservando l'integrità dei dati ed invitandolo a eseguire nuovamente l'operazione. | Dependability | RNF_A_2 |
| DG_5 Tolleranza agli errori | 5 | Il sistema deve essere capace di lavorare sotto condizioni di errore ed in grado di effettuare recovery da eventuali fault, mostrando pagine di errore. | Dependability | RNF_A_4 |
| DG_6 Disponibilità | 6 | Il sistema deve essere accessibile 24/7 salvo interventi di manutenzione che possono incorrere in determinate fasce orarie. | Dependability | RNF_A_3 |
| DG_7 Estensibilità | 7 | Il sistema si presta facilmente all'aggiunta di nuove funzionalità con l'utilizzo del paradigma di programmazione Object Oriented | Maintenance | RNF_S_1 |
| DG_8 Leggibilità | 8 | Il codice relativo al prodotto deve essere commentato fornendo precise informazioni riguardanti ogni metodo. | Maintenance | RNF_S_2 |
| DG_9 Estensibilità | 9 | Il sistema deve risultare scalabile con il disaccoppiamento tra web server e dbms. | Maintenance | RNF_S_3 |
| DG_10 Capacità | 10 | Il sistema deve essere in grado di supportare l'utilizzo in contemporanea di almeno 100 utenti. | Performance | RNF_P_1 |
| DG_11 Tempo di risposta | 11 | Il sistema deve mediamente rispondere ad una richiesta da parte dell'utente in non più di 5 secondi | Performance | RNF_P_2 |



| | | | | |
|----------------------------------|----|---|----------|--------------------|
| DG_12 Usabilità | 12 | Il sistema deve risultare fruibile indipendentemente dalle dimensioni dello schermo e del dispositivo, utilizzando i più diffusi breakpoint: 576px, 768px, 992px, 1200px. | End User | RNF_U_1 |
| DG_13 Usabilità | 13 | Il sistema deve risultare facilmente comprensibile con l'uso delle "8 regole d'oro di Shneiderman" per il design delle interfacce grafiche | End User | RNF_U_2 RNF_U_3 |
| DG_14 Usabilità | 14 | Il sito deve risultare piacevole esteticamente all'utente. | End User | RNF_U_4 |

Trade-Off

A fronte del tempo ridotto da dedicare allo sviluppo del sistema previsto dal budget sono state assegnate delle priorità alle varie funzionalità del sistema.

Tempo di rilascio vs. Funzionalità

Si è preferito dare maggior importanza al buon funzionamento del sistema e ad un testing approfondito dello stesso, rispetto a un'attenta cura dell'aspetto estetico, allo scopo di rientrare nelle tempistiche definite per il rilascio.

Costi vs. Prestazioni

Considerato il budget economico ridotto che è stato allocato per la realizzazione del sistema, non è stato previsto l'utilizzo delle architetture più performanti sul mercato.

Sicurezza vs. Prestazioni

Vista la sensibilità dei dati trattati, per garantire la sicurezza della piattaforma si punta ad implementare sistemi che aumentino la stessa a discapito della velocità delle operazioni, le quali potrebbero impiegare fino a 5 secondi.

1.3 Definizioni, Acronimi e Abbreviazioni

- **Design Goal:** le qualità sulle quali il sistema deve essere focalizzato
- **Dati Persistenti:** dati che sopravvivono all'esecuzione del programma che li ha creati e che dunque vengono salvati



- **Mapping Hardware/Software:** studio della connessione tra parti fisiche e logiche di cui si compongono il sistema
- **SDD:** System Design Document
- **RAD:** Requirements Analysis Document

1.4 Riferimenti

- [MySQL](#)
- [Spring](#)
- [LaMiaCampania](#)
- [GetYourGuide](#)
- [Viator](#)

1.5 Panoramica

Nel documento si tratterà della fase di system design, in particolare si effettuerà l'analisi delle architetture simili, la decomposizione in sottosistemi del sistema proposto, descrivendo in seguito le funzionalità offerte da ognuno dei sottosistemi. Si tratterà della gestione dei dati persistenti, saranno inoltre descritte le condizioni limite e la strategia di deploy.

2 Architettura di Sistemi simili

Nel nostro caso, esistono architetture software simili, ma nessuna è specifica per tutto il territorio italiano, inoltre si è notato come queste siano mirate solo ad enti accreditati, trascurando quella che è la realtà di un piccolo artigiano che lavora nel privato.

I software simili, che trattano il settore del turismo, tendono sempre di più a pubblicizzare eventi come party, concerti e ristoranti, dimenticandosi sempre di più dalle piccole imprese di artigianato.

A fronte di queste mancanze, abbiamo preso in considerazione le architetture di sistemi esistenti che si occupano della prenotazione di tour in tutto il mondo o in specifiche regioni, in particolare:

- **LaMiaCampania**, piattaforma informativa per eventi e varie presenti nel territorio campano;
- **GetYourGuide**, piattaforma per la prenotazione di visite turistiche in tutto il mondo;



- **Viator**, come GetYourGuide è una piattaforma per la prenotazione di visite turistiche in tutto il mondo.

Le 3 piattaforme sono accessibili tramite sito web, raggiungibile da qualsiasi web browser di uso comune. LaMiaCampania, GetYourGuide e Viator sono inoltre dotate di un'applicazione scaricabile su qualsiasi dispositivo Android o iOS, che sostituisce l'accesso da sito web.

Dall'analisi effettuata su ciascuna delle piattaforme, si è concluso che queste utilizzano un database per la gestione dei dati persistenti, sul quale poi delle applicazioni web (scritte con tecnologie eterogenee come PHP, JSP/Java Servlets, ASP) effettuano delle interrogazioni.

GetYourGuide e Viator sono dotate di un sistema di autenticazione basato sull'inserimento di username e password da parte degli utenti, per garantire che le funzionalità e l'accesso ai dati sensibili solo agli utenti autorizzati. GetYourGuide e Viator prevedono inoltre un sistema di pagamenti basato sul framework di PayPal.

3 Architettura del Sistema proposto

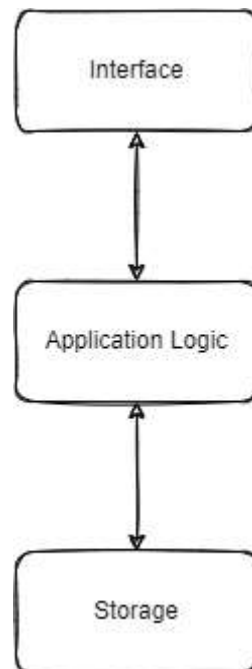
3.1 Panoramica

La piattaforma Just Traditions è un'applicazione che interagisce con gli utenti mediante un'interfaccia web e gestisce la persistenza dei dati mediante un database relazionale.

Il sistema prevede una suddivisione in 3 layer, utilizzando un'architettura Three-Tier: interface, application logic e storage.

- **interface:** si occupa della presentazione dei dati agli utenti e dell'interazione con questi ultimi;
- **application logic:** si occupa della logica applicativa, riceve i comandi da parte dell'interface layer e richiama lo storage layer per poter interagire con i dati persistenti, infine invoca l'interface layer per presentare i risultati all'utente;
- **storage:** si occupa di accedere ai dati persistenti.

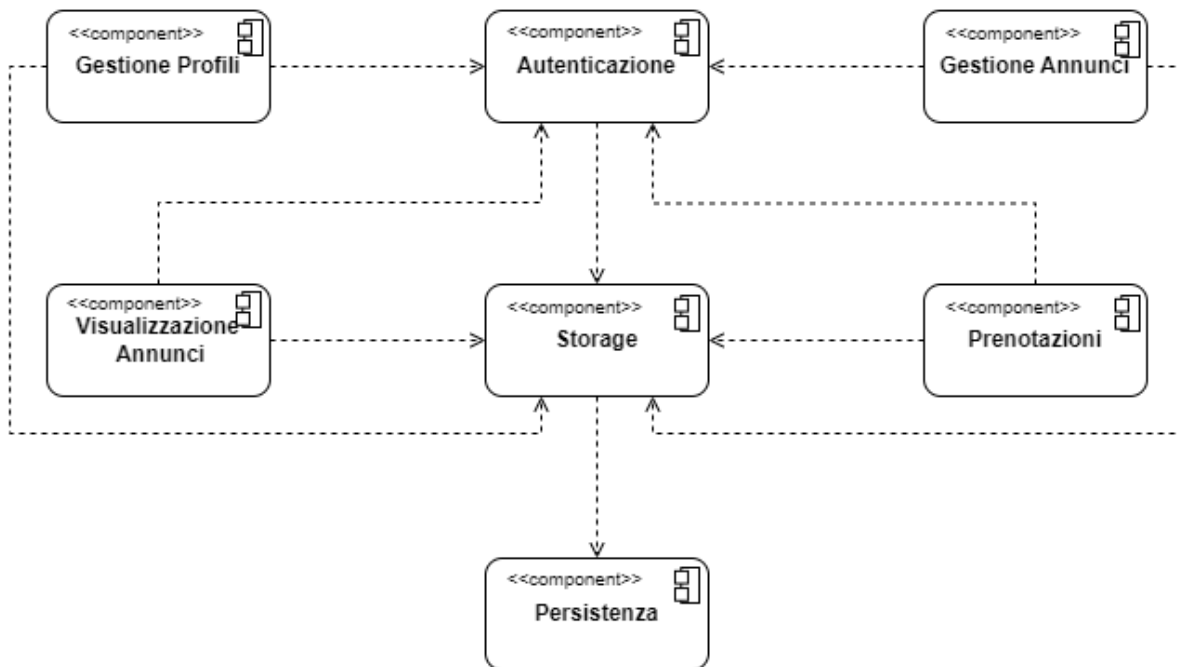
Si sottolinea come sia stato utilizzato questo tipo di architettura al fine di ridurre l'accoppiamento fra le componenti che rappresentano funzionalità differenti, separando la logica di business da quella di presentazione, facilitando anche l'aggiunta di funzionalità future.



3.2 Decomposizione in sottosistemi

I sottosistemi individuati sono:

- **Autenticazione**, si occupa di gestire: il login e il logout di clienti e amministratori, la registrazione di clienti;
- **Gestione Profili**, si occupa delle funzionalità di: visualizzazione dei profili di clienti e amministratori, modifica e cancellazione del profilo dei clienti, rimozione e aggiunta di amministratori;
- **Gestione Annunci**, si occupa delle funzionalità di sottomissione, modifica e rimozione di annunci;
- **Visualizzazione Annunci**, si occupa delle funzionalità di ricerca, visualizzazione e filtraggio di annunci;
- **Prenotazioni**, si occupa delle funzionalità che permettono di visualizzare ed effettuare prenotazioni di visite;
- **Storage**, si interpone fra i vari sottosistemi e il sottosistema di Persistenza;
- **Persistenza**, si occupa di gestire la persistenza dei dati con un database.



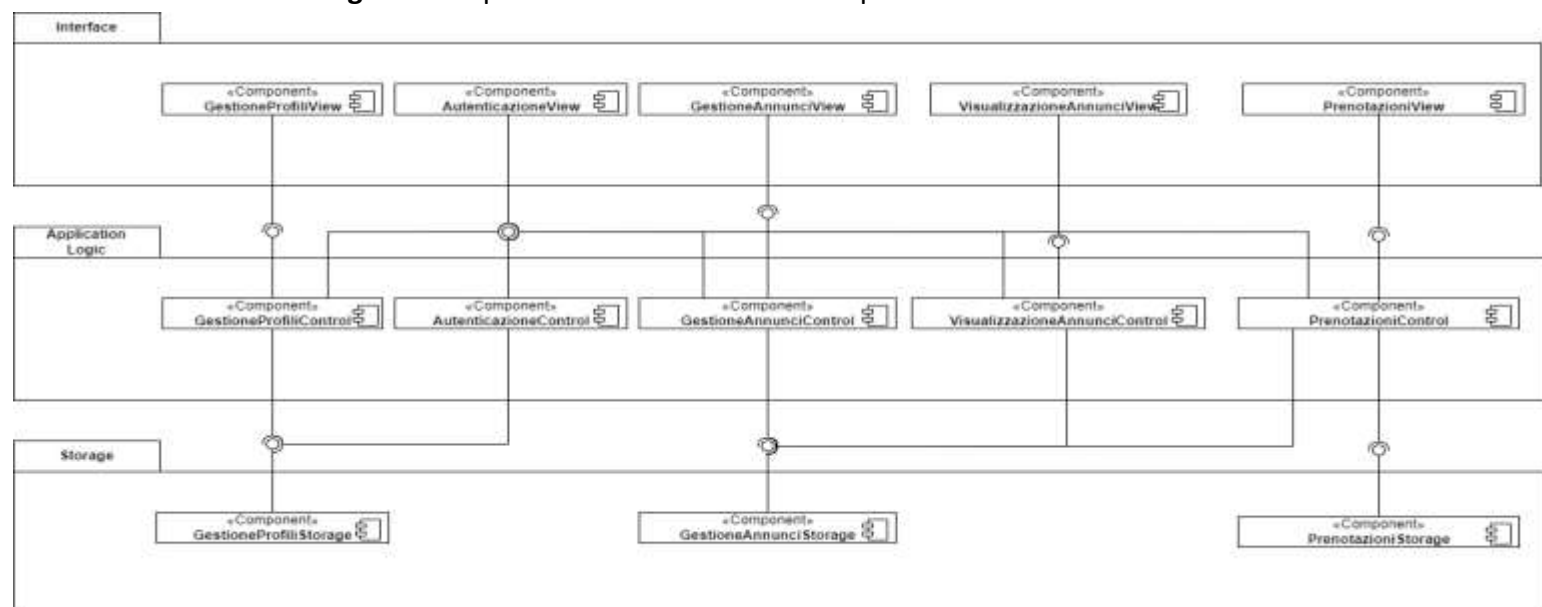
Alcuni sottosistemi saranno gestiti da componenti COTS (Commercial off the shelf), di seguito l'elenco:

- **Storage** verrà gestito da Spring Data JPA;
- **Persistenza** sarà gestita attraverso il DBMS relazionale MySQL.

3.2.1 Diagramma architetturale [\(link all'immagine\)](#)

Di seguito una vista dettagliata dell'architettura, evidenziando le componenti principali:

- **View:** contiene le varie view che saranno renderizzate per creare le pagine web da mostrare all'utente.
- **Control:** si occupa della logica per il controllo del sistema.
- **Storage:** si occupa di fornire l'accesso ai dati persistenti.



3.3 Mapping hardware/software

Just Traditions si compone di:

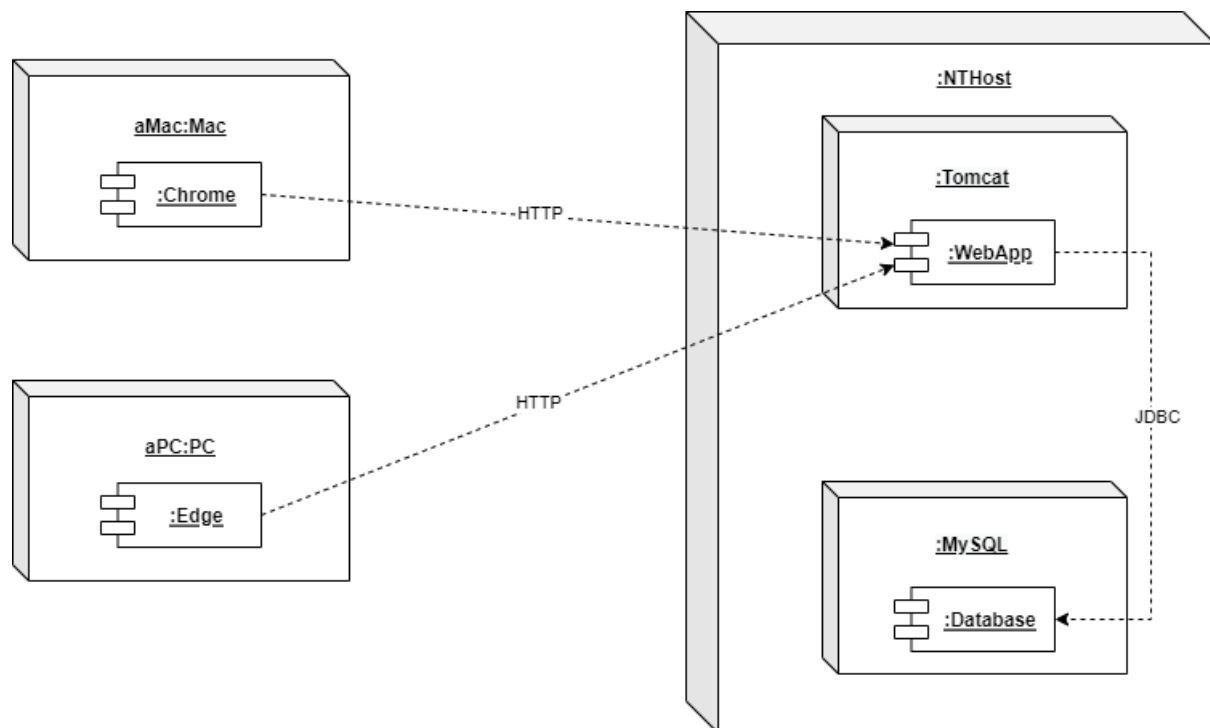
- **WebApp**, nella quale saranno allocati i layer tipici dell'architettura Three-Tier.
- **Database**, realizza il layer di persistenza.

Il sistema necessita di una macchina in grado di supportare:

- **Java SE 17**, al fine di garantire l'operabilità della WebApp.
- **MySQL**, per garantire invece l'operabilità del database con cui la WebApp si interfaccia. La scelta del DBMS suddetto è stata mutuata dalle numerose esperienze pregresse degli sviluppatori coinvolti con tale tecnologia, oltre che dalla sua semplicità d'integrazione.

Per rispettare il budget si è optato di far girare la WebApp e il Database in un unico server con kernel Windows NT, ma saranno separati in differenti ambienti di esecuzione per poter garantire alla piattaforma una maggiore scalabilità e estensibilità.

Di seguito il diagramma di deployment: ([link all'immagine](#))



Nel diagramma si fa menzione di dispositivi Mac e PC che, tramite browser Chrome ed Edge, comunicano con la WebApp tramite il protocollo HTTP, ma i potenziali dispositivi e i browser client non si limitano solo a questi nello schema, che sono stati scelti come esempio,

piuttosto qualsiasi dispositivo client in grado di eseguire un browser web sarà in grado di comunicare, tramite il protocollo HTTP, con la WebApp.

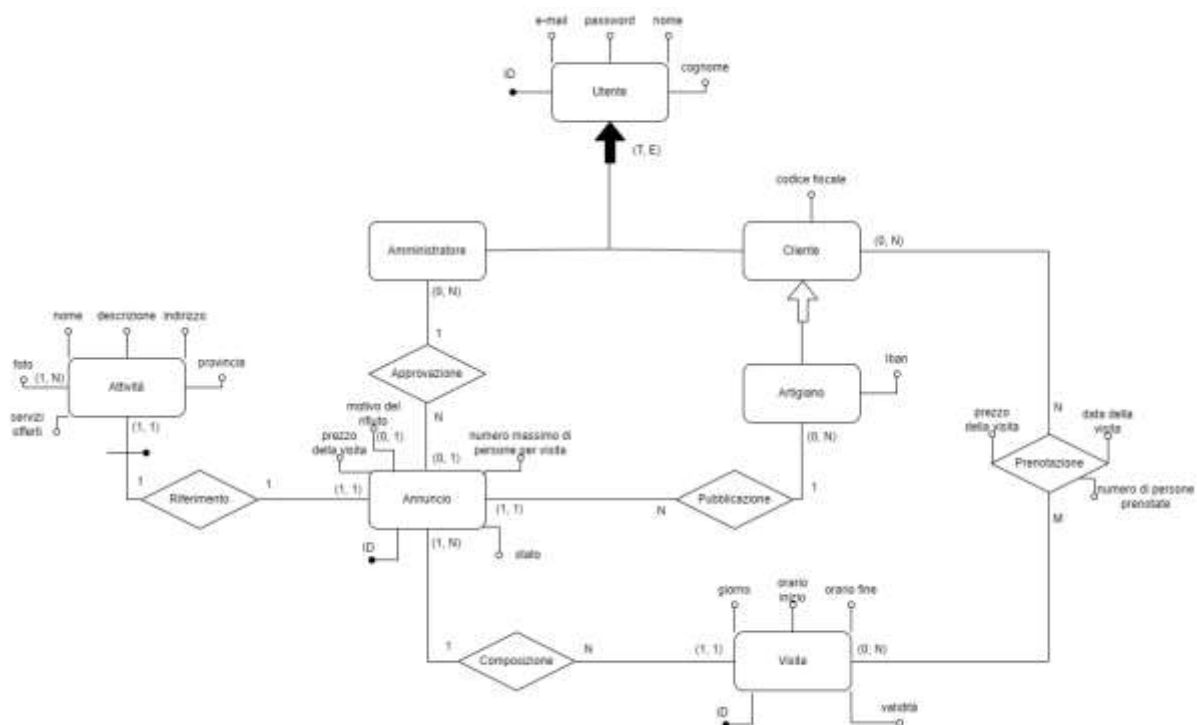
3.4 Gestione dati persistenti

3.4.1 Progettazione concettuale

Per la gestione dei dati persistenti si è optato per un database relazionale gestito tramite il DBMS MySQL.

Prima di arrivare all'implementazione, si è passati per una fase di progettazione concettuale, che ha prodotto, sulla base dei design goals e sulla base degli oggetti entity individuati nella fase di analisi il seguente schema E-R:

3.4.1.1 Schema ER [\(link all'immagine\)](#)



Chiavi candidate:

- (codice fiscale) di Cliente
- (email) di Utente
- (annuncio, giorno, orario inizio, orario fine) di Visita

Note:

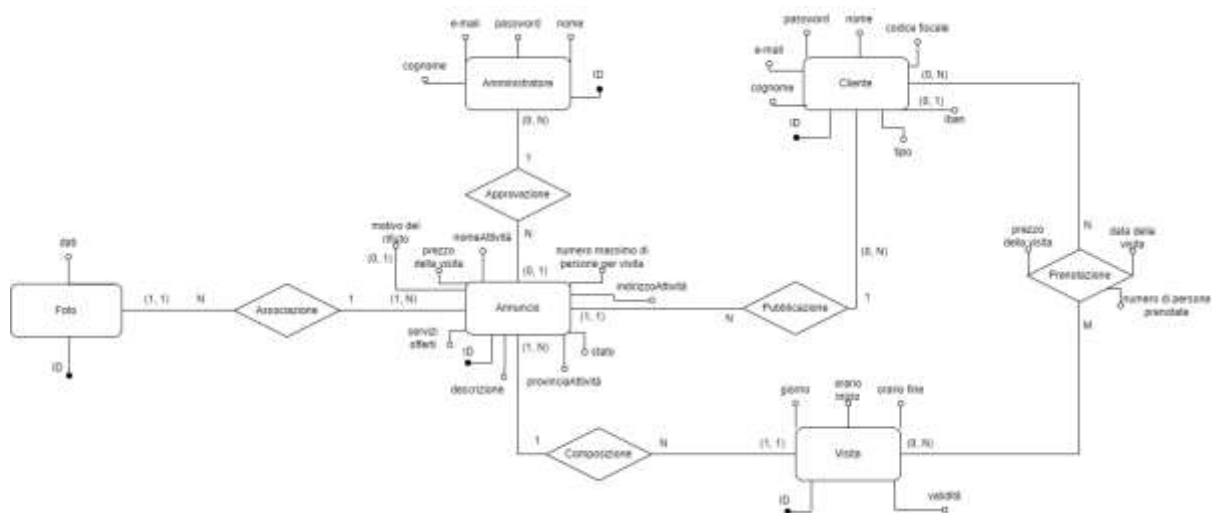
- “motivo del rifiuto” di Annuncio può e deve avere un valore solo se “stato” è “rifiutato”;

- l'attributo "stato" di Annuncio può assumere il valore "proposto", "in revisione", "approvato", "rifiutato";

3.4.2 Progettazione logica

Procediamo ora alla fase di progettazione logica che permetterà di colmare il gap tra il modello concettuale e l'implementazione.

3.4.2.1 Ristrutturazione Schema ER [\(link all'immagine\)](#)



Note:

- "motivo del rifiuto" di Annuncio può e deve avere un valore solo se "stato" è "rifiutato";
- l'attributo "stato" di Annuncio può assumere il valore "proposto", "in revisione", "approvato", "rifiutato";
- l'attributo "tipo" di Cliente può assumere il valore "artigiano" o "cliente";
- Cliente può partecipare all'associazione Pubblicazione solo se l'attributo "tipo" corrisponde ad "artigiano";
- Se l'attributo "tipo" di Cliente è "artigiano", l'attributo "iban" non deve essere "NULL".



3.4.3 Dizionario dei dati

Di seguito si mostrano gli attributi per ogni tabella individuata.

| Nome Tabella | amministratore | | |
|----------------|---|-------------------|----------------------------|
| Descrizione | Contiene i dati relativi ad un amministratore | | |
| Nome Attributo | Tipo | Vincolo di chiave | Altri vincoli |
| id | BIGINT | PRIMARY KEY | NOT NULL AUTO-INCREMENT |
| email | VARCHAR(319) | UNIQUE | NOT NULL |
| password | CHAR(32) | | NOT NULL |
| nome | VARCHAR(30) | | NOT NULL |
| cognome | VARCHAR(30) | | NOT NULL |

| Nome Tabella | cliente | | |
|----------------|--|-------------------|----------------------------|
| Descrizione | Contiene i dati relativi a un cliente o un artigiano | | |
| Nome Attributo | Tipo | Vincolo di chiave | Altri vincoli |
| id | BIGINT | PRIMARY KEY | NOT NULL AUTO-INCREMENT |
| email | VARCHAR(319) | UNIQUE | NOT NULL |
| password | CHAR(32) | | NOT NULL |
| nome | VARCHAR(30) | | NOT NULL |
| cognome | VARCHAR(30) | | NOT NULL |
| codiceFiscale | CHAR(16) | UNIQUE | NOT NULL |
| iban | CHAR(27) | | |
| tipo | VARCHAR(9) | | NOT NULL |



| Nome Tabella | visita | | |
|-------------------------------|--|------------------------|----------------------------|
| Descrizione | Contiene i dati relativi ad una visita | | |
| Nome Attributo | Tipo | Vincolo di chiave | Altri vincoli |
| id | BIGINT | PRIMARY KEY | NOT NULL AUTO-INCREMENT |
| giorno | VARCHAR(9) | | NOT NULL |
| orarioInizio | TIME | | NOT NULL |
| orarioFine | TIME | | NOT NULL |
| validità | BOOLEAN | | NOT NULL |
| annuncio | BIGINT | FOREIGN KEY (annuncio) | NOT NULL |
| Vincoli su attributi multipli | UNIQUE(annuncio, giorno, orarioInizio, orarioFine) | | |

| Nome Tabella | annuncio | | |
|-------------------------|---|-------------------|-------------------------|
| Descrizione | Contiene i dati relativi ad un annuncio | | |
| Nome Attributo | Tipo | Vincolo di chiave | Altri vincoli |
| id | BIGINT | PRIMARY KEY | NOT NULL AUTO-INCREMENT |
| nomeAttività | VARCHAR(40) | | NOT NULL |
| provinciaAttività | VARCHAR(21) | | NOT NULL |
| indirizzoAttività | VARCHAR(128) | | NOT NULL |
| descrizione | VARCHAR(1024) | | NOT NULL |
| serviziOfferti | VARCHAR(255) | | NOT NULL |
| numMaxPersonePer Visita | INT | | NOT NULL |
| prezzoVisita | DECIMAL(5, 2) | | NOT NULL |



| Nome Tabella | annuncio | | |
|------------------|--------------|------------------------------|----------|
| stato | VARCHAR(12) | | NOT NULL |
| motivoDelRifiuto | VARCHAR(255) | | |
| amministratore | BIGINT | FOREIGN KEY (amministratore) | |
| artigiano | BIGINT | FOREIGN KEY (cliente) | NOT NULL |

| Nome Tabella | foto | | |
|----------------|--------------------------------------|------------------------|-------------------------|
| Descrizione | Contiene i dati relativi ad una foto | | |
| Nome Attributo | Tipo | Vincolo di chiave | Altri vincoli |
| id | BIGINT | PRIMARY KEY | NOT NULL AUTO-INCREMENT |
| dati | BLOB | | NOT NULL |
| annuncio | BIGINT | FOREIGN KEY (annuncio) | NOT NULL |

| Nome Tabella | prenotazione | | |
|---------------------|--|-----------------------|-------------------------|
| Descrizione | Contiene i dati relativi ad una prenotazione | | |
| Nome Attributo | Tipo | Vincolo di chiave | Altri vincoli |
| prezzoVisita | DECIMAL(5, 2) | | NOT NULL |
| dataVisita | DATE | | NOT NULL |
| numPersonePrenotate | INT | | NOT NULL |
| cliente | BIGINT | FOREIGN KEY (cliente) | NOT NULL |
| visita | BIGINT | FOREIGN KEY (visita) | |
| id | BIGINT | PRIMARY KEY | NOT NULL AUTO-INCREMENT |



3.5 Controllo degli accessi e sicurezza

Per prevenire accessi non autorizzati ad oggetti modificabili appartenenti all'entità del dominio, il sistema garantisce un controllo degli accessi basato su autenticazione tramite l'immissione di username e password.

Al fine di garantire un alto livello di sicurezza, i dati sensibili degli utenti saranno memorizzati nel database utilizzando sistemi di cifratura. Il sistema fornirà metodi di modifica su input dell'utente, dei propri dati sensibili. In generale, il protocollo di accesso al database sarà basato su SSL.

Per tener traccia dell'utente, verrà utilizzata la sessione del server che avrà durata di 30 minuti dopo l'ultima interazione dell'utente con il sistema.

La tabella sottostante riporta le funzionalità a cui gli attori hanno accesso, attraverso i diversi sottosistemi. Nella lettura della tabella sottostante, si tenga presente il diagramma degli attori riportato nel RAD, per comprendere le gerarchie, quindi le funzionalità che i diversi attori ereditano da altri.

| Sottosistema | Gestione Profili | Autenticazione | Gestione Annunci | Visualizzazione Annunci | Prenotazioni |
|----------------|---|---|--|--|---|
| Attore | | | | | |
| Cliente | Visualizzazione Profilo Artigiano; Visualizzazione Profilo Personale; Modifica Profilo; Cancellazione Account; | Login; Registrazione; Logout; | | Ricerca Annunci; Visualizzazione Lista Annunci Approvati; Visualizzazione Dettagli Annuncio; | Visualizzazione Prenotazioni Personali; Effettua Prenotazione; |
| Artigiano | | | Sottomissione Annuncio; Modifica Annuncio; Rimozione Annuncio; | Visualizzazione Lista Annunci Sottomessi; | Visualizzazione Prenotazioni Annuncio; |
| Amministratore | Visualizzazione Amministratori; Rimozione Amministratore; Aggiunta Amministratore; | Login Amministratore; Logout Amministratore; | Modifica Stato Annuncio; | Visualizzazione Lista Annunci; Filtra Lista Annunci; Visualizzazione Scheda Annuncio; | |



3.6 Controllo flusso globale del sistema

Il sistema Just Traditions è un sistema interattivo, per cui ogni funzionalità viene avviata in seguito ad un comando impartito dall'utente, tramite l'utilizzo di un'interfaccia grafica.

Quando un utente intende utilizzare una funzionalità del sistema attraverso l'interfaccia, quest'ultima selezionerà il controllore del sottosistema corrispondente.

Il controllore gestirà il flusso di eventi e si interfacerà direttamente con i sottosistemi dedicati alla logica applicativa.

Essendo una web-application, il sistema utilizzerà un meccanismo di controllo del flusso di tipo event-driven.

Il sistema prevede anche un controllo globale di tipo thread-driven concorrente poiché per ogni evento generato dai client verrà creato un thread ad esso dedicato, per svolgere le operazioni richieste da ogni utente in modo concorrente.

Ogni accesso in scrittura sui dati verrà effettuato in modo sequenziale, impedendo che due thread accedano contemporaneamente a una variabile persistente.

A fronte dell'elevato numero di utenti che potrebbe utilizzare la piattaforma in contemporanea e dell'elevata interattività della stessa si è optato per l'utilizzo di Spring, che si serve del web server Tomcat integrato, che supporta l'esecuzione di applicativi web basati su Java Servlets e pagine JSP e gestisce autonomamente ciascuna richiesta da un client creando un thread dedicato.

3.7 Condizione limite

Start-up

Per il primo start-up del sistema "Just Traditions" è necessario l'avvio dell'eseguibile .jar che permetta l'esecuzione della web app e di un Database MySQL per la gestione dei dati persistenti.

| | | | |
|-------------------------------------|---|-----------------|-------------------|
| Identificativo UC_BC_1 | Avvio del Sistema | Data | 4/12/2022 |
| | | Versione | 1.0 |
| | | Autori | Michele Spagnuolo |
| Descrizione | Lo UC permette l'avvio del sistema | | |
| Attore principale | Amministratore | | |
| Attori secondari | NA | | |
| Entry condition | Deve essere scaricato l'eseguibile .jar all'interno del server L'Amministratore accede al Server | | |
| Exit condition On success | Il sistema viene avviato correttamente | | |
| Exit condition On failure | Il sistema non viene avviato | | |



| Flusso di eventi principale | | |
|--|----------------|--|
| 1 | Amministratore | Lancia il servizio DBMS MySql (su Server Windows), accedendo all'interfaccia grafica MySQL workbench e selezionando la voce server, a quel punto sceglie l'opzione StartUp/Shutdown e con l'apposito comando lancia il MySQL Server; |
| 2 | Sistema | Notifica che il database è stato avviato correttamente. |
| 3 | Amministratore | Lancia il .jar eseguibile, tramite prompt dei comandi inserendo username e password del Database. |
| 4 | Sistema | Notifica che il web server è stato avviato correttamente. |
| I Flusso di Eventi Alternativo: non è possibile avviare correttamente il database MySQL | | |
| 2.a1 | Sistema | Notifica l'Amministratore l'impossibilità di avviare il database, specificando il motivo di tale errore. |
| 2.a2 | Amministratore | Corregge gli errori all'avvio del database. |
| 2.a3 | Amministratore | Esegue il Passaggio 1. |
| II Flusso di Eventi Alternativo: non è possibile avviare correttamente l'eseguibile .jar | | |
| 4.a1 | Sistema | Notifica l'Amministratore l'impossibilità di avviare il web server, specificando il motivo di tale errore. |
| 4.a1 | Amministratore | Corregge gli errori. |
| 4.a1 | Amministratore | Esegue il Passaggio 3. |

Terminazione

Al momento della corretta chiusura dell'applicazione, si ha la terminazione del sistema con un segnale di terminazione, dopo la quale nessun client potrà connettersi al sistema.

| Identificativo UC_BC_2 | Terminazione del Sistema | Data | 4/12/2022 |
|---------------------------|--|----------|-------------------|
| | | Versione | 1.0 |
| | | Autori | Michele Spagnuolo |
| Descrizione | Lo UC permette la terminazione del sistema | | |
| Attore principale | Amministratore | | |
| Attori secondari | NA | | |



| | | |
|---|----------------|--|
| Entry condition | | L'Amministratore accede al Server AND Il Sistema è stato precedentemente avviato AND Il Sistema non è stato ancora spento |
| Exit condition On success | | Il sistema viene spento correttamente |
| Exit condition On failure | | Il sistema non viene spento |
| Flusso di eventi principale | | |
| 1 | Amministratore | Esegue sulla macchina il comando che termina l'esecuzione dell'eseguibile .jar. |
| 2 | Sistema | Notifica che l'esecuzione del server web è terminata correttamente. |
| 3 | Amministratore | Esegue sulla macchina il comando che termina l'esecuzione del database MySQL. |
| 4 | Sistema | Notifica che l'esecuzione del database è terminata correttamente. |
| I Flusso di eventi alternativo: non è stato possibile terminare correttamente il server web | | |
| 2.a1 | Sistema | Notifica all'Amministratore che non è stato possibile terminare correttamente l'esecuzione del server web, specificando i motivi di tale errore. |
| 2.a2 | Amministratore | Risolve gli errori nella terminazione del server web. |
| 2.a3 | Amministratore | Esegue il Passaggio 1. |
| II Flusso di eventi alternativo: non è stato possibile terminare correttamente il database MySQL | | |
| 4.a1 | Sistema | Notifica all'Amministratore che non è stato possibile terminare correttamente l'esecuzione del database MySQL, specificando i motivi di tale errore. |
| 4.a2 | Amministratore | Risolve gli errori nella terminazione del database. |
| 4.a3 | Amministratore | Esegue il Passaggio 3. |



Fallimento

Durante l'esecuzione del sistema "Just Traditions" potrebbero verificarsi diversi errori risultanti nella terminazione inaspettata del sistema.

| | | | | |
|------------------------------|----------------|--|----------|-------------------|
| Identificativo UC_BC_3 | | Fallimento del Sistema | Data | 4/12/2022 |
| | | | Versione | 1.0 |
| | | | Autori | Michele Spagnuolo |
| Descrizione | | L'UC definisce il comportamento del Sistema in caso di fallimento. | | |
| Attore principale | | Amministratore | | |
| Attori secondari | | NA | | |
| Entry condition | | Il Sistema viene terminato inaspettatamente | | |
| Exit condition On success | | Il Sistema viene riavviato correttamente | | |
| Exit condition On failure | | Il Sistema non viene riavviato | | |
| Flusso di eventi principale | | | | |
| 1 | Sistema | Mostra una pagina di errore, specificando le informazioni relative al problema riscontrato | | |
| 2 | Amministratore | Risolve il problema riscontrato | | |
| 3 | Amministratore | Include UC_BC_1 | | |

Errore di Accesso ai Dati Persistenti

Durante l'esecuzione del sistema "Just Traditions" potrebbero verificarsi diversi errori risultanti nell'impossibilità di accesso ai dati persistenti.

| | | | |
|---------------------------|--|----------|-------------------|
| Identificativo UC_BC_4 | Errore Di Accesso ai Dati Persistenti | Data | 4/12/2022 |
| | | Versione | 1.0 |
| | | Autori | Michele Spagnuolo |
| Descrizione | L'UC descrive il comportamento del sistema qualora fosse impossibile accedere ai dati persistenti. | | |
| Attore principale | Amministratore | | |
| Attori secondari | NA | | |
| Entry condition | Il Sistema non può accedere ai dati persistenti | | |



| | | |
|--|----------------|--|
| Exit condition On success | | Il Sistema riprende il normale funzionamento |
| Exit condition On failure | | Il Sistema non riprende il normale funzionamento |
| Flusso di eventi principale | | |
| 1 | Sistema | Mostra una pagina di errore, specificando le informazioni relative al problema riscontrato |
| 2 | Amministratore | Accede all'interfaccia grafica MySQL workbench, tramite la quale risolve il problema riscontrato |

4 Funzionalità dei sottosistemi

In questa sezione vengono descritte le funzionalità di ogni sottosistema precedentemente elencato.

4.1 Sottosistema Gestione Profili

| Funzionalità | Descrizione | Interfaccia |
|--|---|------------------------|
| Visualizzazione Profilo Artigiano | Questa funzionalità permette di visualizzare il profilo di uno specifico artigiano. | GestioneProfiliControl |
| Visualizzazione Profilo Personale | Questa funzionalità permette al cliente di visualizzare il proprio profilo. | GestioneProfiliControl |
| Visualizzazione Amministratori | Questa funzionalità permette di visualizzare la lista di tutti gli amministratori. | GestioneProfiliControl |
| Modifica Profilo | Questa funzionalità permette di modificare il proprio profilo. | GestioneProfiliControl |
| Cancellazione Account | Questa funzionalità permette di cancellare il proprio account dalla piattaforma. | GestioneProfiliControl |
| Rimozione Amministratore | Questa funzionalità permette di rimuovere un amministratore dalla piattaforma | GestioneProfiliControl |
| Aggiunta Amministratore | Questa funzionalità permette di aggiungere un amministratore alla piattaforma. | GestioneProfiliControl |



4.2 Sottosistema Autenticazione

| Funzionalità | Descrizione | Interfaccia |
|-----------------------------|--|-----------------------|
| Login | Questa funzionalità permette di effettuare l'accesso alla piattaforma tramite le proprie credenziali. | AutenticazioneControl |
| Logout | Questa funzionalità permette di disconnettersi dalla piattaforma. | AutenticazioneControl |
| LoginAmministratore | Questa funzionalità permette all'amministratore di effettuare l'accesso alla piattaforma tramite le proprie credenziali. | AutenticazioneControl |
| LogoutAmministratore | Questa funzionalità permette all'amministratore di disconnettersi dalla piattaforma. | AutenticazioneControl |
| Registrazione | Questa funzionalità permette di registrarsi sulla piattaforma. | AutenticazioneControl |

4.3 Sottosistema Gestione Annunci

| Funzionalità | Descrizione | Interfaccia |
|--------------------------------|--|------------------------|
| Sottomissione Annuncio | Questa funzionalità permette di sottomettere un nuovo annuncio alla piattaforma. | GestioneAnnunciControl |
| Rimozione Annuncio | Questa funzionalità permette di rimuovere un annuncio dalla piattaforma. | GestioneAnnunciControl |
| Modifica Stato Annuncio | Questa funzionalità permette all'amministratore di cambiare lo stato di un annuncio. | GestioneAnnunciControl |
| Modifica Annuncio | Questa funzionalità permette di modificare un annuncio presente sulla piattaforma. | GestioneAnnunciControl |

4.4 Sottosistema Visualizzazione Annunci

| Funzionalità | Descrizione | Interfaccia |
|--|---|-------------------------------|
| Ricerca Annunci | Questa funzionalità permette di cercare un determinato annuncio presente sulla piattaforma. | VisualizzazioneAnnunciControl |
| Visualizzazione Lista Annunci Approvati | Questa funzionalità permette al cliente di visualizzare tutti gli annunci approvati dagli | VisualizzazioneAnnunciControl |



| | | |
|---|--|-------------------------------|
| | amministratori presenti sulla piattaforma. | |
| Visualizzazione Dettagli Annuncio | Questa funzionalità permette al cliente di visualizzare i dettagli di un annuncio. | VisualizzazioneAnnunciControl |
| Visualizzazione Lista Annunci Sottomessi | Questa funzionalità permette all'artigiano di visualizzare la lista degli annunci che ha sottomesso alla piattaforma | VisualizzazioneAnnunciControl |
| Visualizzazione Lista Annunci | Questa funzionalità permette all'amministratore di visualizzare tutti gli annunci presenti sulla piattaforma. | VisualizzazioneAnnunciControl |
| Filtra Lista Annunci | Questa funzionalità permette all'amministratore di filtrare la lista di tutti gli annunci presenti sulla piattaforma | VisualizzazioneAnnunciControl |
| Visualizzazione Scheda Annuncio | Questa funzionalità permette all'amministratore di visualizzare la scheda di un annuncio presente sulla piattaforma. | VisualizzazioneAnnunciControl |

4.5 Sottosistema Prenotazioni

| Funzionalità | Descrizione | Interfaccia |
|---|--|---------------------|
| Visualizzazione Prenotazioni Personali | Questa funzionalità permette di visualizzare le prenotazioni effettuate. | PrenotazioniControl |
| Effettua Prenotazione | Questa funzionalità permette di effettuare una prenotazione ad un annuncio. | PrenotazioniControl |
| Visualizzazione Prenotazioni Annuncio | Questa funzionalità permette all'artigiano di visualizzare tutte le prenotazioni effettuate per un suo annuncio. | PrenotazioniControl |



5 Glossario

- **Just Traditions:** Nome della Web Application che si andrà a realizzare.
- **Piattaforma:** Base software o hardware su cui sono sviluppate o eseguite applicazioni.
- **SSL:** Secure Sockets Layer, protocollo di sicurezza Internet basato sulla crittografia, per il trasporto dei dati.
- **Windows NT:** kernel del sistema operativo Windows.
- **COTS:** Commercial Off The Shelf, si riferisce a componenti hardware e software disponibili sul mercato per l'acquisto da parte di aziende di sviluppo interessate a utilizzarli nei loro progetti.
- **ASP:** Active Server Page (ASP) è una tecnologia Microsoft che permette di realizzare siti Web dinamici in grado di interagire con utenti.