



UNIVERSITÀ DEGLI STUDI
DI SALERNO

UNISA

System Design Document HAMPLANET.BLOG

Partecipanti:

Danese Vincenzo 0512107959

Nappi Nicola 0512109534

Russo Giovanni 0512118684

Vitiello Catello 0512110338

Presentato a:

Prof. Andrea De Lucia

Dipartimento di informatica

21 marzo 2024

Indice

| | | |
|----------|---|-----------|
| 1 | Introduction | 2 |
| 1.1 | Scopo del sistema | 2 |
| 1.2 | OBIETTIVI DI DESIGN | 2 |
| 1.3 | DEFINIZIONI, ACRONIMI E ABBREVIAZIONI | 3 |
| 1.4 | RIFERIMENTI | 3 |
| 1.5 | OVERVIEW | 3 |
| 2 | SISTEMA ATTUALE | 4 |
| 3 | SISTEMA PROPOSTO | 4 |
| 3.1 | PANORAMICA | 4 |
| 3.2 | SUBSYSTEM DECOMPOSITION | 4 |
| 3.3 | HARDWARE/SOFTWARE MAPPING | 5 |
| 3.4 | PERSISTENT DATA MANAGEMENT | 6 |
| 3.5 | ACCESS CONTROL AND SECURITY | 6 |
| 3.6 | GLOBAL SOFTWARE CONTROL | 7 |
| 3.7 | BOUNDARY CONDITION | 7 |
| 3.7.1 | USE CASES BOUNDARY CONDITIONS | 9 |
| 4 | SUBSYSTEM SERVICES | 14 |

1 Introduction

1.1 Scopo del sistema

1.2 OBIETTIVI DI DESIGN

| | | | | |
|-----|---------------|---------------|---|---------------------|
| DG1 | Performance | Memory | In base alle stime effettuate il server utilizzato deve avere una capacità di 250GB. | RNF4, 5, 6, 7 |
| DG2 | Dependability | Robustness | Il sistema deve notificare l'utente in caso di input errato e gestire propriamente le possibili eccezioni | Dominio Applicativo |
| DG3 | Dependability | Reliability | L'output delle operazioni deve coincidere con quanto specificato negli Use Cases e quindi con le aspettative degli utenti. | Dominio Applicativo |
| DG4 | Dependability | Security | Il sistema non deve permettere agli utenti di accedere a pagine per cui non possiedono l'autorizzazione tramite dei filtri. | RNF2 |
| DG5 | Maintenance | Extensibility | Il sistema deve facilitare la futura aggiunta di nuove funzionalità. | Dominio Applicativo |
| DG6 | Maintenance | Readability | Il codice deve essere modulare e commentato in modo da semplificarne la lettura e la comprensione. | RNF8 |
| DG7 | End User | Usability | Il sistema deve notificare l'utente circa l'esito delle operazioni in caso di errore con degli opportuni messaggi. | RNF1 |

| | |
|--------------------------------|---|
| Space vs Speed | Il sistema deve garantire l'accesso a una grande quantità di contenuti, questo potrebbe causare dei rallentamenti durante la ricerca o la fruizione degli stessi. |
| Delivery Time vs Functionality | È richiesto di sviluppare l'intero sistema proposto, implementando tutti gli Use Cases, entro i tempi previsti per le deadlines. |
| Delivery Time vs Usability | Per vincoli di tempo si è deciso di non rendere la grafica della piattaforma responsive. |

1.3 DEFINIZIONI, ACRONIMI E ABBREVIAZIONI

- Piattaforma: Applicazione web (HamPlanet.blog)
- RAD: Requirements Analysis Document
- UCBC: Use Case Boundary Condition
- RNF: Requisito Non Funzionale
- DG: Design Goal

1.4 RIFERIMENTI

Nel corso del documento facciamo riferimento ai Requisiti Non Funzionali, agli Use Cases e al Class Diagram presenti nel documento RAD.

1.5 OVERVIEW

Nei paragrafi successivi presentiamo:

- L'analisi dell'architettura di una piattaforma concorrente a Poorify.
- La decomposizione del sistema in sottosistemi.
- La strategia di deployment su hardware.
- La strategia adottata per la gestione dei dati persistenti.
- Le varie Boundary Condition del sistema.
- I servizi offerti dai vari sottosistemi.

2 SISTEMA ATTUALE

Il principale concorrente di HamPlanet.blog è il noto blog di Geopop. La piattaforma di Geopop si basa su un'architettura Client-Server.

3 SISTEMA PROPOSTO

3.1 PANORAMICA

HamPlanet.blog è una webApplication.

Le prossime sezioni analizzeranno la struttura del sistema. Elenchiamo brevemente le componenti principali dell'applicazione:

- Un front-end web dinamico sviluppato con JSP, JS e CSS.
- Un back-end per la logica di business basato su Servlet Java.
- Un insieme di classi DAO che collegano la logica alla persistenza tramite JDBC.
- Database relazionale MYSQL su server per la persistenza dei dati.

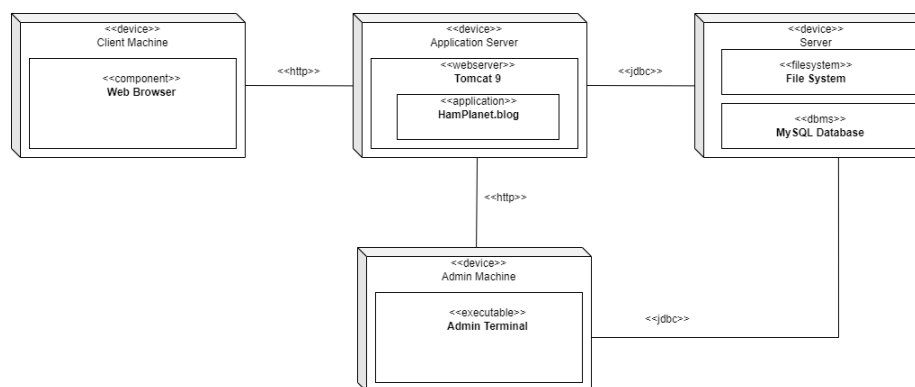
3.2 SUBSYSTEM DECOMPOSITION

La piattaforma HamPlanet.blog sfrutta una architettura web three-tier:

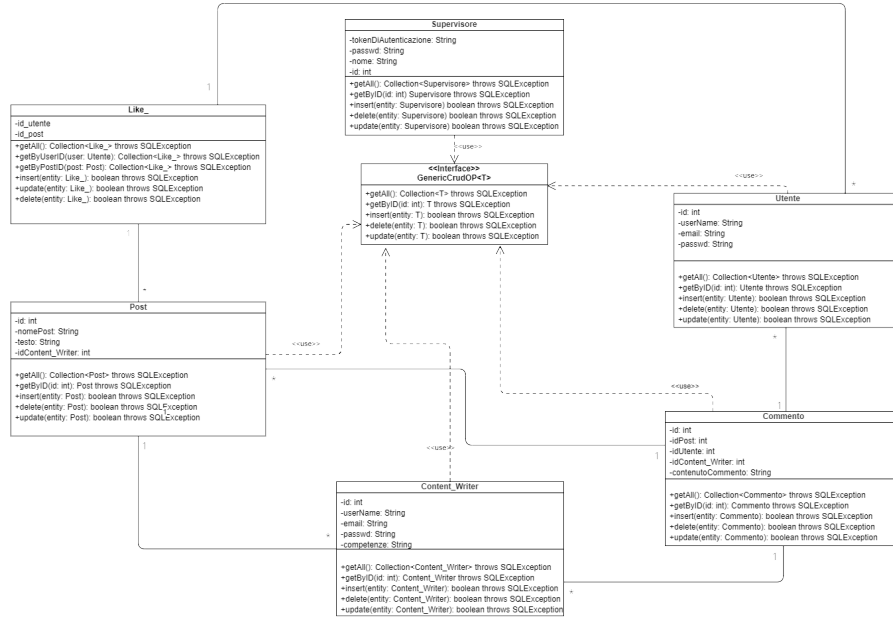
- **Presentation layer:** Interfaccia impiegata dagli utenti finali per interagire con la piattaforma tramite web browser dalle loro macchine client.
- **Logical layer:** La logica di business che gestisce le operazioni del sistema. Dividiamo tale strato in quattro sottosistemi/partizioni:
 - **Gestione Utente:** per la registrazione, l'autenticazione, la modifica e l'eliminazione del profilo utente.
 - **Gestione Navigazione:** per la ricerca dei contenuti e la navigazione tra le varie pagine della piattaforma.
 - **Gestione Post:** per la creazione, l'eliminazione, aggiunta o rimozione del mi piace ad un post.
 - **Gestione Commento:** per la creazione, l'eliminazione di un commento.
- **Data layer:** Lo strato software che permette di effettuare le operazioni su database necessarie al funzionamento della piattaforma.

3.3 HARDWARE/SOFTWARE MAPPING

Gli utenti accedono alla piattaforma Poorify tramite web browser dalla propria macchina client stabilendo una connessione http con un server. L'applicazione viene eseguita nel web container Tomcat che interagisce con un database relazionale SQL, potenzialmente distribuito su molteplici macchine server. Gli amministratori del sistema possono interagire sia con Tomcat per effettuare startup e shutdown del sistema, sia operare direttamente su database tramite un terminale dedicato per aggiungere nuovi impiegati supervisori o lanciare delle query per l'aggiornamento o la creazione di contenuti.



3.4 PERSISTENT DATA MANAGEMENT



Facciamo riferimento al class diagram per decidere quali attributi vengo memorizzati sul Database relazionale o su filesystem. Il filesystem viene impiegato per la memorizzazione di dati molto pesanti, come le immagini.

I file multimediali sono acceduti tramite un path relativo a partire da una cartella di default all'interno del server in locale, così che venga facilitata l'eventuale migrazione del sistema su macchine diverse senza dover apportare modifiche alle procedure di Data Management.

La root del nostro filesystem presenta le seguenti sottocartelle:

- **/Profile:** per profile_image foto profilo della classe Content Writer
- **/Post:** per post_image della classe post

I file salvati sono denominati x.estensione dove x è l'id associato alla entry corrispondente nel database. Ad esempio profile_image del Content Writer con id 16 è salvata come 16.png nella sottocartella /Profile mentre la foto del post con id 5 è salvata come 5.png nella sottocartella /Post.

Gli attributi rimanenti sono dati primitivi e poco complessi quindi sono resi persistenti sul Database

3.5 ACCESS CONTROL AND SECURITY

Illustriamo nella seguente tabella le operazioni che ogni tipologia di utente può eseguire all'interno della piattaforma.

| | Utente non registrato | Utente | Content Writer | Supervisore |
|----------------------|---|--|--|--|
| Gestione Utente | <ul style="list-style-type: none"> • Registrazione | <ul style="list-style-type: none"> • Login • Logout • Modifica profilo • Visualizza profilo • Elimina profilo | <ul style="list-style-type: none"> • Login • Logout • Modifica profilo • Visualizza profilo • Elimina profilo | <ul style="list-style-type: none"> • Login • Logout • Eliminazione Profilo altrui |
| Gestione Post | | <ul style="list-style-type: none"> • Commentare • Mettere e togliere Like | <ul style="list-style-type: none"> • Creazione post • Eliminazione Post • Commentare | <ul style="list-style-type: none"> • Eliminazione Post |
| Gestione Navigazione | | <ul style="list-style-type: none"> • Navigazione Pagine • Visualizzare un post • Visualizzare proprio profilo | <ul style="list-style-type: none"> • Navigazione pagine • visualizza post • visualizza proprio profilo | <ul style="list-style-type: none"> • Navigazione pagine • Visualizza post • Visualizza pagina di attesa accettazione Content Writer |

La piattaforma sfrutta dei filtri per garantire la sicurezza impedendo agli utenti di eseguire operazioni e accedere a pagine per cui non possiedono l'autorizzazione. Inoltre i supervisori hanno il compito di eliminare i contenuti ritenuti offensivi o inopportuni e nel caso eliminare il profilo del responsabile.

3.6 GLOBAL SOFTWARE CONTROL

L'architettura della piattaforma HamPlanet.blog è intrinsecamente Event-driven in quanto le servlet reagiscono agli eventi generati dall'input utente, ciò permette di aggiungere nuove funzionalità in maniera semplice senza dover modificare il codice già scritto. L'applicazione è anche distribuita e quindi alcune operazioni su database possono causare delle indesiderate race conditions. Risulta pertanto necessario gestire la concorrenza tramite thread e applicare dei meccanismi di sincronizzazione per garantire la coerenza delle risorse della piattaforma.

3.7 BOUNDARY CONDITION

Startup e shutdown del sistema sono gestiti tramite due eseguibili Python.

- **startup.py** effettua la configurazione del sistema. In questa fase viene creato e popolato, previo inserimento delle credenziali dell'amministratore del sistema, il database MySQL impiegato dalla piattaforma, e viene eseguito lo startup del web container Tomcat.
- **supervisore.py** permette di gestire (aggiungere/cambiare/eliminare) i supervisori della piattaforma tramite alcune query predefinite.
- **shutdown.py** effettua lo shutdown del web container Tomcat, la piattaforma risulterà quindi offline.

Il sistema deve risultare robusto nella gestione dell'input utente, ad esempio interrompendo l'operazione sul nascere. In caso di eccezioni o fallimenti non previsti però non si potrà garantire il corretto funzionamento della piattaforma e potrebbe risultare necessario un riavvio.

3.7.1 USE CASES BOUNDARY CONDITIONS

UCBC-1: STARTUP

| | |
|----------------------|--|
| USE CASE NAME | Startup |
| PARTICIPATING ACTORS | Amministratore del sistema |
| ENTRY CONDITION | <ul style="list-style-type: none">• L'amministratore esegue <i>startup.py</i> |
| FLOW OF EVENTS | <ol style="list-style-type: none">1. L'amministratore fornisce username e password del database2. Il programma provvede all'inizializzazione del sistema utilizzando i file <i>HPBlog_DB.sql</i> e <i>PopulateDB.sql</i>3. Il programma provvede ad effettuare il deploy del file <i>NomeFile.war</i> sul web container Tomcat4. Il programma effettua lo startup di Tomcat |
| EXIT CONDITION | <ul style="list-style-type: none">• La piattaforma HamPlanet.blog è online |
| EXCEPTIONS | <ul style="list-style-type: none">• Se al punto 1 le credenziali fornite sono errate, l'amministratore viene notificato con un messaggio di errore e può ritentare l'autenticazione• Se al punto 2 <i>startup.py</i> non trova <i>HPBlog_DB.sql</i> oppure <i>PopulateDB.sql</i> il programma termina con un messaggio di errore• Se al punto 3 <i>startup.py</i> non trova il file <i>NomeFile.war</i> il programma termina con un messaggio di errore• Se al punto 4 <i>startup.py</i> non riesce ad avviare Tomcat il programma termina con un messaggio di errore |

UCBC-2: ADD SUPERVISOR

| | |
|----------------------|---|
| USE CASE NAME | Add Supervisor |
| PARTICIPATING ACTORS | Amministratore del sistema |
| ENTRY CONDITION | <ul style="list-style-type: none">• L'amministratore esegue <i>overseer.py</i> selezionando l'opzione aggiungi |
| FLOW OF EVENTS | <ol style="list-style-type: none">1. L'amministratore fornisce username e password del database2. L'amministratore inserisce <i>email</i> e <i>password</i> del nuovo supervisore3. Il programma verifica che i campi rispettino il formato richiesto4. Il programma verifica che l'email non sia già utilizzata5. Il programma salva i dati sul database |
| EXIT CONDITION | <ul style="list-style-type: none">• Il supervisore è stato aggiunto con successo |
| EXCEPTIONS | <ul style="list-style-type: none">• Se al punto 1 le credenziali fornite sono errate, l'amministratore viene notificato con un messaggio di errore e può ritentare l'autenticazione• Se al punto 3 uno dei campi non rispetta il formato richiesto, il programma mostra un messaggio di errore e l'amministratore può ritentare l'inserimento• Se al punto 4 l'email inserita risulta già utilizzata, il programma mostra un messaggio di errore e l'amministratore può ritentare l'inserimento |

UCBC-3: UPDATE SUPERVISOR

| | |
|----------------------|--|
| USE CASE NAME | Update Supervisor |
| PARTICIPATING ACTORS | Amministratore del sistema |
| ENTRY CONDITION | <ul style="list-style-type: none">• L'amministratore esegue <i>overseer.py</i> selezionando l'opzione cambia |
| FLOW OF EVENTS | <ol style="list-style-type: none">1. L'amministratore fornisce username e password del database2. L'amministratore inserisce <i>email</i> del supervisore da modificare3. Il programma verifica l'esistenza del supervisore4. L'amministratore inserisce la nuova password5. Il programma verifica che il campo email rispetti il formato richiesto6. Il programma aggiorna i dati del supervisore sul database |
| EXIT CONDITION | <ul style="list-style-type: none">• Il supervisore è stato aggiornato con successo |
| EXCEPTIONS | <ul style="list-style-type: none">• Se al punto 1 le credenziali fornite sono errate, l'amministratore viene notificato con un messaggio di errore e può ritentare l'autenticazione• Se al punto 3 non viene trovata l'email fornita, il programma mostra un messaggio di notifica "<i>Email Non Trovata</i>" e l'amministratore può ritentare l'inserimento• Se al punto 5 il campo email non rispetta il formato corretto, il programma mostra un messaggio di errore e l'amministratore può ritentare l'inserimento |

UCBC-4: REMOVE SUPERVISOR

| | |
|----------------------|--|
| USE CASE NAME | Remove Supervisor |
| PARTICIPATING ACTORS | Amministratore del sistema |
| ENTRY CONDITION | <ul style="list-style-type: none">• L'amministratore esegue <i>overseer.py</i> selezionando l'opzione rimuovi |
| FLOW OF EVENTS | <ol style="list-style-type: none">1. L'amministratore fornisce username e password del database2. L'amministratore inserisce <i>email</i> del supervisore da rimuovere3. Il programma verifica l'esistenza del supervisore4. Il programma chiede conferma all'amministratore per procedere5. L'amministratore conferma la propria decisione6. Il programma elimina i dati del supervisore dal database |
| EXIT CONDITION | <ul style="list-style-type: none">• Il supervisore è stato rimosso con successo |
| EXCEPTIONS | <ul style="list-style-type: none">• Se al punto 1 le credenziali fornite sono errate, l'amministratore viene notificato con un messaggio di errore e può ritentare l'autenticazione• Se al punto 3 non viene trovata l'email fornita, il programma mostra un messaggio di notifica "<i>Email Non Trovata</i>" e l'amministratore può ritentare l'inserimento• Se al punto 5 l'amministratore non effettua la conferma il programma termina normalmente |

UCBC-5: SHUTDOWN

| | |
|----------------------|--|
| USE CASE NAME | Shutdown |
| PARTICIPATING ACTORS | Amministratore del sistema |
| ENTRY CONDITION | <ul style="list-style-type: none">• L'amministratore esegue <i>shutdown.py</i> |
| FLOW OF EVENTS | <ol style="list-style-type: none">1. L'amministratore conferma di voler eseguire lo shutdown del sistema2. Il programma effettua lo shutdown del web container Tomcat |
| EXIT CONDITION | <ul style="list-style-type: none">• La piattaforma HamPlanet.blog è offline |
| EXCEPTIONS | <ul style="list-style-type: none">• Se al punto 1 l'amministratore non conferma lo shutdown, il programma termina normalmente e la piattaforma è ancora online |

4 SUBSYSTEM SERVICES

| Sottosistemi | Servizi | Operazioni |
|----------------------|---|--|
| Gestione Utente | Registrazione Autenticazione Modifica Profilo Eliminazione Profilo | Registrazione Login, Logout Modifica profilo Eliminazione profilo |
| Gestione Post | Creazione Post funzioni social Commentare | Creazione post like/unlike post Commentare, togliere commento |
| Gestione Navigazione | Navigazione pagine Visualizzazione pagine | Navigazione pagine Visualizza pagina utente/ Content writer e post |