



System Design Document Progetto Fund.It

Riferimento	Gruppo G13
Versione	0.9
Scritto da	Luigi Crescenzo, Francesco Esposito, Sabato Genovese, Angelo Meo

Revision History

Data	Versione	Descrizione	Autori
13/12/2021	0.1	Prima stesura del documento	Sabato Genovese
14/12/2021	0.2	Prima definizione Design Goals	Luigi Crescenzo
17/12/2021	0.3	Definizione Trade-Offs	Gruppo
18/12/2021	0.4	Definizione scomposizione in sottosistemi	Angelo Meo
21/12/2021	0.5	Definizione deployment diagram	Angelo Meo
22/12/2021	0.6	Ristrutturazione deployment diagram	Luigi Crescenzo
23/12/2021	0.6	Ristrutturazione entity class diagram	Gruppo
23/12/2021	0.7	Definizione schema ER	Sabato Genovese, Francesco Esposito
28/12/2021	0.7	Definizione dizionario dei dati	Gruppo
30/01/2022	0.8	Ristrutturazione della scomposizione in sottosistemi	Sabato Genovese
02/01/2022	0.8	Revisione della scomposizione in sottosistemi	Luigi Crescenzo, Sabato Genovese
03/01/2022	0.9	Definizione use cases Boundary Conditions	Gruppo



Sommario

Introduzione	5
Scopo del sistema	5
Obiettivi di Design (Design Goals)	5
Design Trade-off	7
Definizioni, acronimi e abbreviazioni	7
Riferimenti	7
Architettura del sistema corrente	8
Architettura del sistema proposto	8
Panoramica sulla sezione	8
Decomposizione in layer	9
Scomposizione in sottosistemi	10
Diagramma architetturale	11
Sottosistema Autenticazione	12
Sottosistema Gestione Donazioni	13
Mapping hardware/software	14
Class diagram ristrutturato	15
Gestione dei dati persistenti	16
Introduzione	16
Schema ER del database	17
Dizionario dei dati	18
Categoria	18
Donazione	18
Immagine	19
Campagna	19
FAQ	20
Utente	21
Segnalazione	22
Controllo degli accessi e sicurezza	23
Controllo globale del software	24
Condizioni limite	25
UCBC_1 – Spegnimento del sistema	25
UCBC_2 - Avvio del sistema	26
UCBC_3 - Fallimento del sistema	27
UCBC_4 - Errore di accesso ai dati persistenti	27



Servizi dei sottosistemi	28
Sottosistema Autenticazione	28
Sottosistema Gestione Utente	29
Sottosistema Gestione Categorie	30
Sottosistema Gestione Donazioni	30
Sottosistema Gestione Campagne	31
Sottosistema Gestione Segnalazioni	32
Sottosistema Gestione FAQ	32
Glossario	33

Introduzione

Scopo del sistema

Fund.It si propone come piattaforma online per la raccolta fondi, capace di semplificare la gestione e il monitoraggio delle campagne indette. Il sistema sarà gestito da uno o più amministratori, i quali hanno anche il compito di risolvere le segnalazioni effettuate dagli utenti sulle campagne. Inoltre, sarà possibile condividere le campagne presenti sulla piattaforma per aumentare la visibilità della stessa usufruendo dei profili social.

Il sistema può essere fruito da diversi dispositivi, sia mobili che PC Desktop, attraverso l'ausilio di un web browser.

Data la vasta disponibilità di web browser presenti sul mercato, il sistema dovrà essere fruibile attraverso i browser più comuni ed utilizzati, dando anche un grado di usabilità accettabile per i browser meno stabili e meno conosciuti.

La piattaforma sfrutterà un servizio esterno per la gestione delle transazioni bancarie, dato che si dà la possibilità di effettuare donazioni alle campagne pubblicate su di essa.

Obiettivi di Design (Design Goals)

Rank	ID Design Goal	Descrizione	Categoria
4	DG_1 Tempo di risposta	Il sistema dovrà ridurre al minimo i tempi di caricamento delle pagine richieste dall'utente. L'obiettivo preposto è quello di raggiungere un tempo di caricamento massimo di 4 secondi per pagina.	Performance
7	DG_2 Memoria	Il sistema dovrà avere a disposizione una quantità di memoria sufficiente per gestire tutti i dati della piattaforma quali utenti, campagne, donazioni, segnalazioni e in generale tutti i dati persistenti.	Performance
3	DG_3 Robustezza	Il sistema sarà in grado di reagire ad input errati da parte dell'utente, invalidando l'operazione interessata finché l'utente non inserisce dati che rispettino un determinato formato, descritto sotto forma di suggerimento. I suggerimenti saranno visualizzati con note ed effetti grafici che richiamino l'attenzione dell'utente.	Dependability
1	DG_4 Security	L'accesso al sistema sarà consentito per mezzo di e-mail e password. La password verrà crittografata utilizzando SHA-256. Inoltre saranno descritte delle politiche di	Dependability

		accesso ad aree del sistema destinate a tipologie differenti di utente.	
2	DG_5 Development Cost	Il costo di sviluppo per la piattaforma non dovrà superare le 200 ore. Il budget è stato calcolato moltiplicando il numero di ore (50 ore) che ogni membro del gruppo dovrà dedicare per il numero dei membri del team di sviluppo.	Cost Criteria
8	DG_6 Deployment Cost	Il team di sviluppo utilizzerà framework/strumenti open source e gratuiti per effettuare il deployment del sistema in modo da azzerare i costi legati all'approvvigionamento e l'uso.	Cost Criteria
6	DG_7 Portabilità	Il sistema potrà essere migrato su differenti macchine che supportano i requisiti minimi di sistema del Web server Apache Tomcat, in particolare la versione 9.0.24. Inoltre, la macchina che ospiterà il sistema dovrà supportare la JDK 15 di Java	Maintenance Criteria
5	DG_8 Usabilità	<p>Il sistema sarà disponibile su vari dispositivi mobili e non (Smartphone, Tablet, Notebook, PC Desktop e in generale di qualunque dispositivo che dispone di un web browser che supporti le tecnologie da noi adottate).</p> <p>Le interfacce del nostro sistema saranno progettate per migliorare l'aspetto delle funzionalità offerte dal sistema. L'utilizzatore dovrà avere familiarità con l'utilizzo di piattaforme web che supportano i pagamenti digitali. L'utente medio impiegherà un tempo stimato di 10 minuti per sfruttare a pieno le funzionalità base del sistema.</p>	End User Criteria

Design Trade-off

Trade-off	Descrizione
Coupling vs Complessità	Il sistema verrà progettato con lo scopo di ridurre l'accoppiamento tra i sottosistemi. Per raggiungere questo scopo, la complessità dei sottosistemi aumenterà con l'aumentare del numero di componenti utilizzate
Funzionalità vs Budget tempo	L'obiettivo preposto è quello di progettare ed implementare tutte le funzionalità richieste dal committente, anche se ciò comporterà l'andare oltre il budget di 200 ore complessivo
Affidabilità vs Performance	Il sistema implementa dei meccanismi che puntano all'incremento dell'affidabilità complessiva a discapito dei costi di performance. Inoltre verranno eseguiti dei backup periodici sulla base dati per poter attuare operazioni di recovery dei dati in situazioni critiche di errore.

Definizioni, acronimi e abbreviazioni

Riferimenti

- Object Oriented Software Engineering – using UML, Patterns and Java 3 edition;
- Documentazione messa a disposizione dai tutor;
- Sample di documenti;
- Slide del corso;
- Riferimento alla notazione usata per il diagramma architetturale

<https://martinfowler.com/bliki/BallAndSocket.html>

Architettura del sistema corrente

Attualmente non esiste un sistema software che raggruppa tutte le funzionalità che offrirà il sistema. Il panorama di soluzioni alternative è costituito prevalentemente da soluzioni “fisiche”; pertanto, risulta difficile confrontare in maniera ragionevole il sistema rispetto i competitors;

Tramite il sistema da noi proposto, l’utente ha la possibilità di avere a disposizione una piattaforma digitale che semplifica alcuni aspetti di una raccolta “fisica”.

Architettura del sistema proposto

Panoramica sulla sezione

Il sistema da noi proposto sarà una web application basata su un’architettura Three Tier, in modo tale da partizionare i sottosistemi che hanno in comune le stesse operazioni in moduli. La separazione dell’interface layer dalla application logic consente di modificare e/o sviluppare nuove interfacce utente senza impattare gravemente sulla modifica dei sottosistemi del livello sottostante.

Per lo sviluppo del sistema verranno utilizzate le seguenti tecnologie per il front-end:

- HTML
- CSS
- JavaScript
- jQuery
- Bootstrap

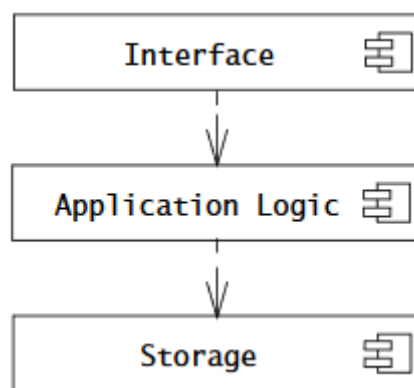
Mentre per il lato back-end troviamo Java come linguaggio Object Oriented e MySQL come DBMS per la base dati relazionale che sarà gestita dal servizio cloud di Microsoft Azure.

Si utilizzeranno le API esterne di servizi come Whatsapp, Facebook e Twitter per la condivisione delle campagne e il servizio di pagamenti digitali esterno FundPay.

Decomposizione in layer

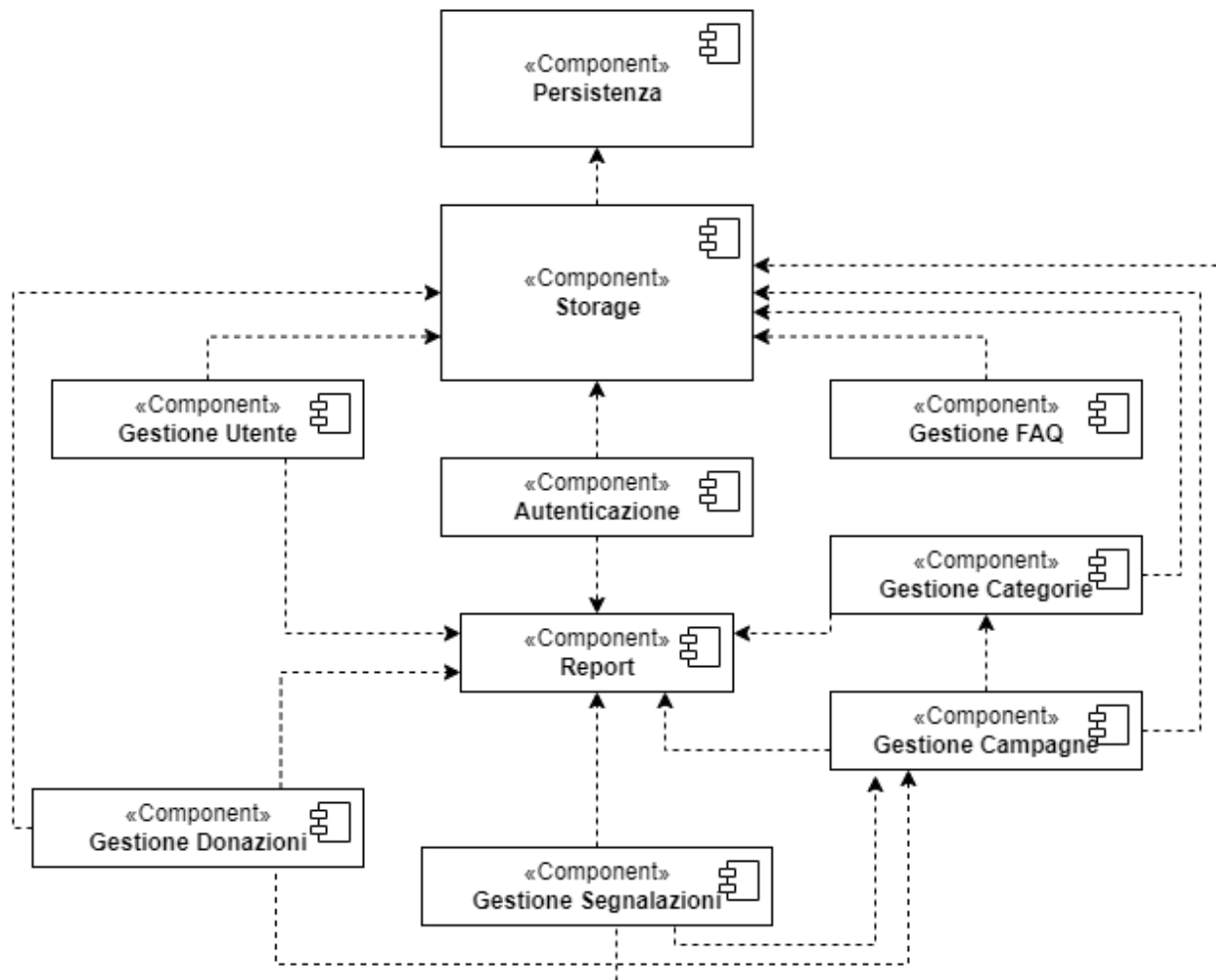
Il sistema software da noi progettato sarà suddiviso in tre layer che offrono rispettivamente le seguenti funzionalità:

- **Storage:** si occupa di memorizzare, recuperare e interrogare gli oggetti entity persistenti;
- **Interface:** include tutti i boundary object che si interfacciano con l'utente;
- **Application Logic:** include tutti gli oggetti control che fanno parte della logica di business. Inoltre, si interpone tra l'interface layer e storage layer, organizza il flusso di controllo delle operazioni.



Scomposizione in sottosistemi

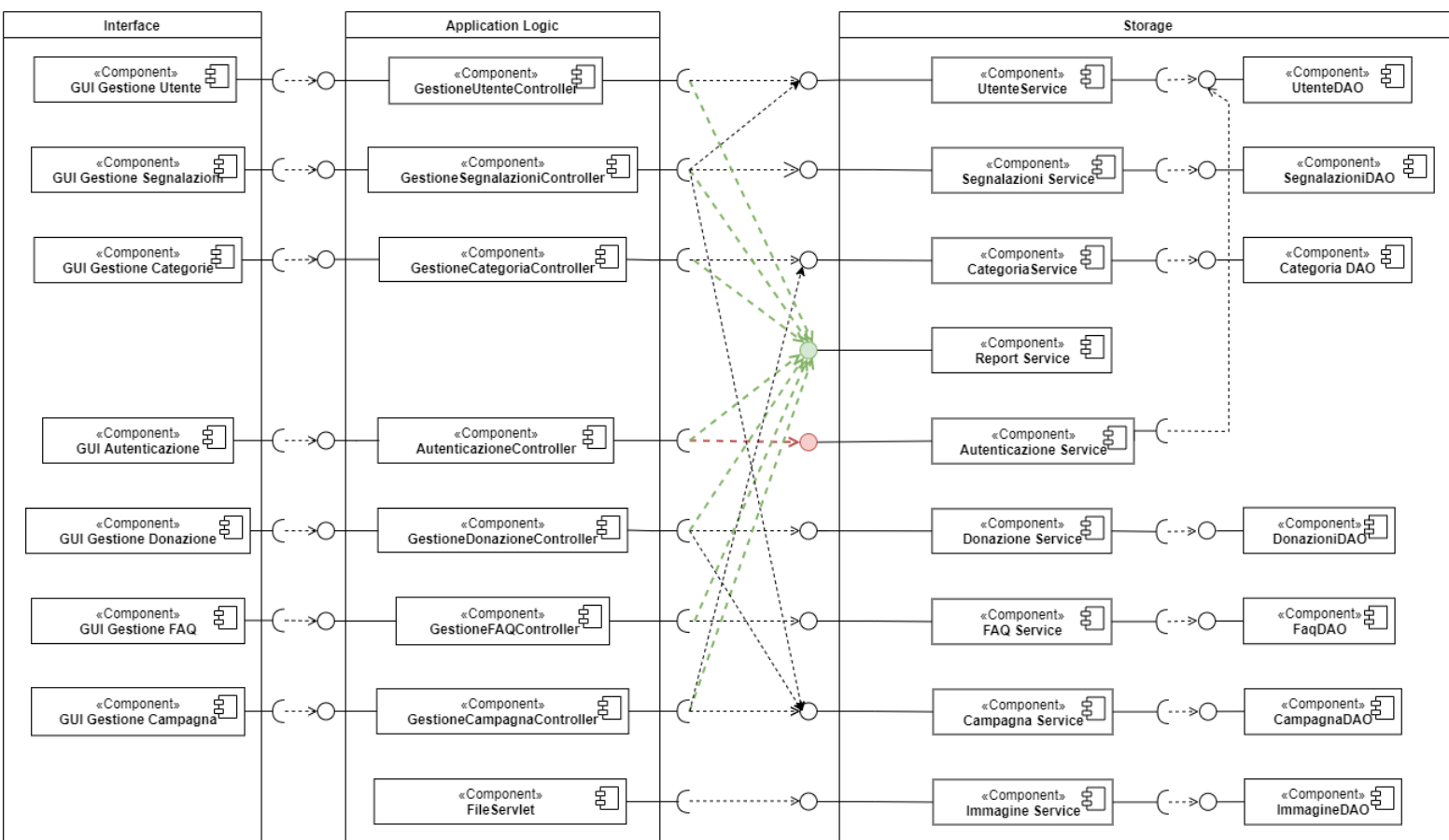
Il sistema da noi progettato è suddiviso nei seguenti sottosistemi, dove vengono esplicitate anche le loro dipendenze:



- Il sottosistema Persistenza sarà gestito da un DBMS relazionale basato sulla tecnologia cloud di Microsoft Azure
- Il sottosistema Storage è composto da tutti i DAO dei vari sottosistemi individuati.

Diagramma architetturale

- **Interface:** Il tier comprende le GUI che il sistema utilizzerà per interagire con l'utente;
- **Application Logic:** Comprende i sottosistemi che gestiscono la logica per il controllo del sistema;
- **Storage:** Comprende uno strato composto da Services che si occupano della logica di business, ed uno strato di DAO che si occupa di fornire l'accesso ai dati persistenti.





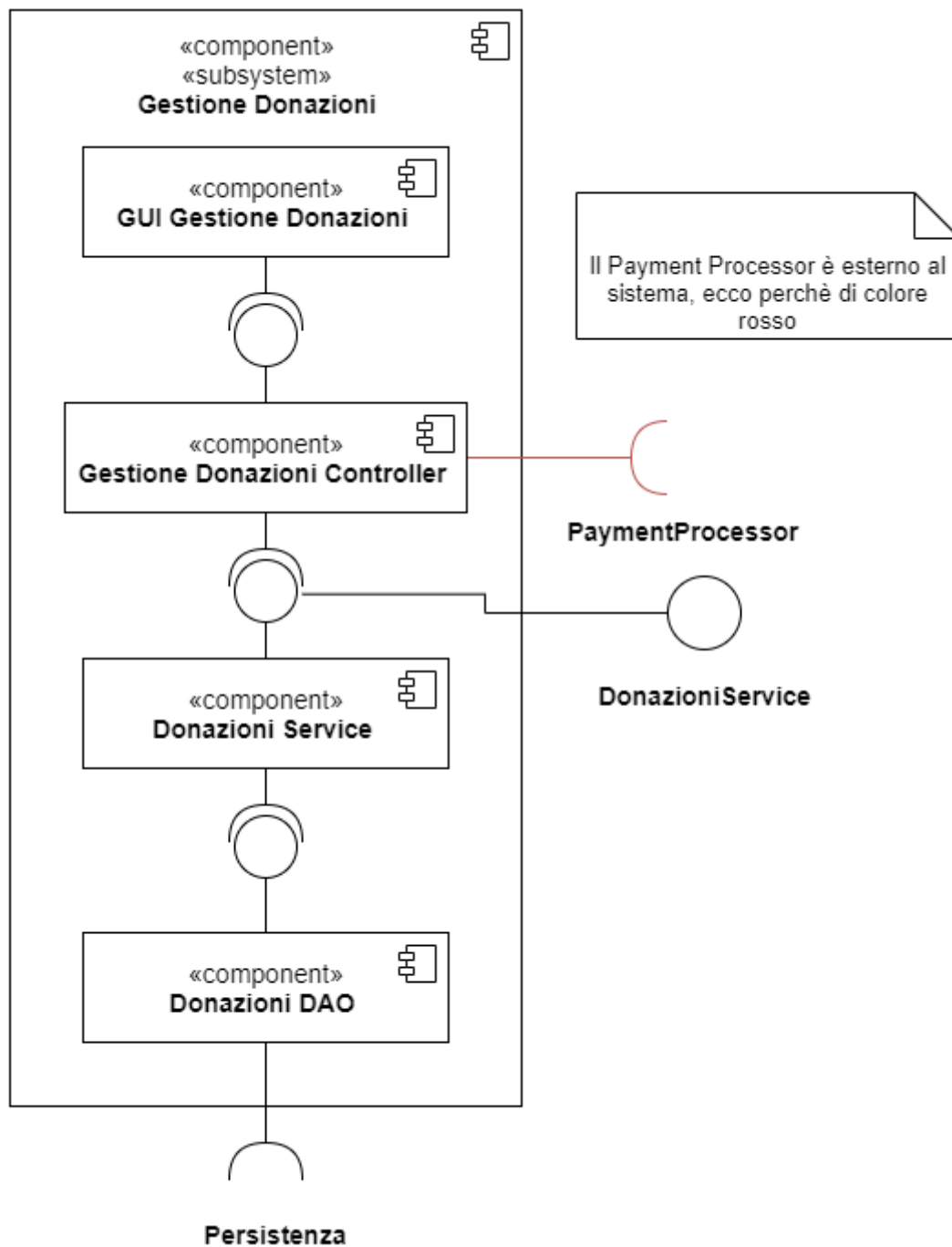
The diagram illustrates the component structure of the Authentication Module (Autenticazione). It is a UML Component Diagram showing the following components and their relationships:

- Autenticazione** (System Component): The root component, labeled «component» and «subsystem».
- GUI Autenticazione** (Component): A component that provides a provided interface (circle) to the AutenticazioneController.
- AutenticazioneController** (Component): A component that provides a provided interface (circle) to the AutenticazioneService and consumes a required interface (half-circle) from the AutenticazioneService.
- AutenticazioneService** (Component): A component that provides a provided interface (circle) to the AutenticazioneController and consumes a required interface (half-circle) from the UtenteDAO.
- AutenticazioneService** (Interface): A provided interface (circle) that the AutenticazioneController consumes.
- UtenteDAO** (Interface): A provided interface (half-circle) that the AutenticazioneService consumes.

The relationships are as follows:

- GUI Autenticazione provides a provided interface to AutenticazioneController.
- AutenticazioneController provides a provided interface to AutenticazioneService.
- AutenticazioneController consumes a required interface from AutenticazioneService.
- AutenticazioneService provides a provided interface to AutenticazioneController.
- AutenticazioneService consumes a required interface from UtenteDAO.

Sottosistema Gestione Donazioni

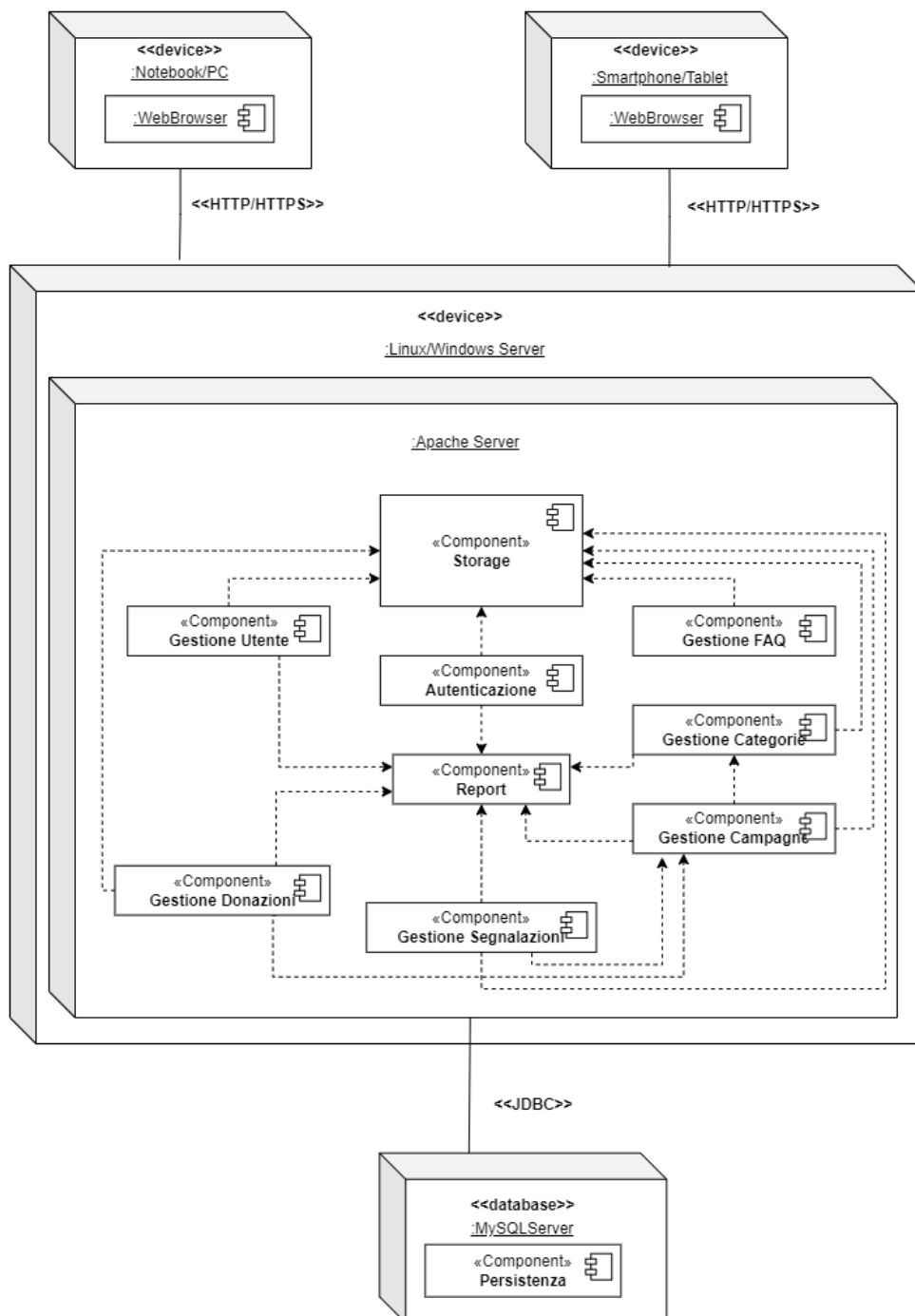


Mapping hardware/software

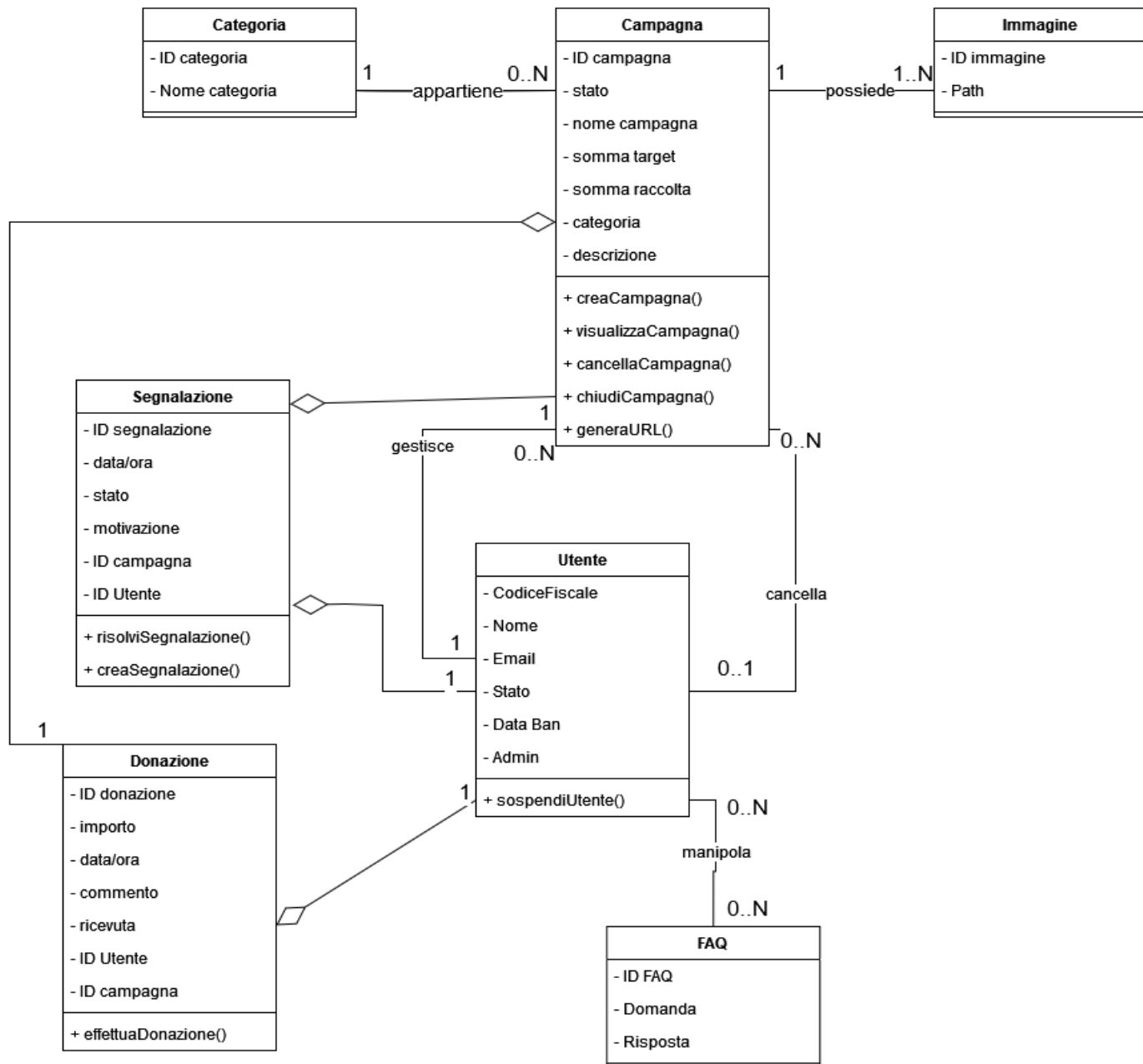
Il sistema da sviluppare sarà ospitato su una macchina in grado di supportare i requisiti minimi per supportare il web server Apache.

Il deploy del sistema sarà eseguito su un singolo nodo, il database sarà disponibile tramite il servizio cloud per i database relazionali di Microsoft Azure come tutti i servizi esterni per la condivisione delle campagne e la gestione dei pagamenti.

Segue un UML deployment diagram che descrive il mapping hardware/software



Class diagram ristrutturato



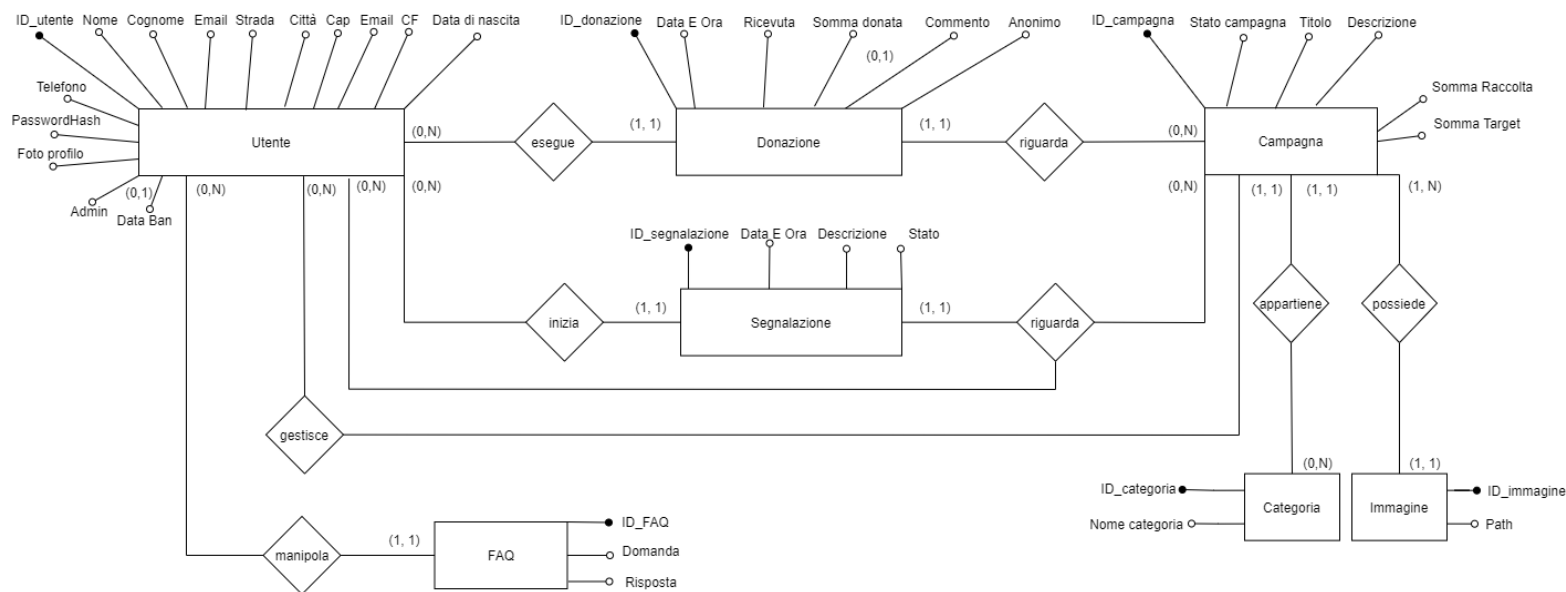
Gestione dei dati persistenti

Introduzione

La gestione dei dati persistenti verrà eseguita per mezzo di un database relazionale insieme ad un DMBS che si occuperà di gestire la consistenza, gli accessi e l'accesso concorrente. Tale scelta è giustificata dai servizi offerti che in questo caso ci aiutano nel soddisfare i system goal presentati, in particolare:

- Supporto alle operazioni atomiche (Ciò significa che se tutte le operazioni vengono eseguite correttamente si procede al commit, mentre si procede al rollback e quindi nessuna modifica viene eseguita alla base dati se si verifica un errore)
- Possibilità di specificare vincoli intrarelazionali come Not Null nel caso in cui non sia permesso il valore null per un campo, Unique che impone che l'attributo o insieme di attributi siano una superchiave oltre al vincolo di Primary Key che permette di identificare una tupla in modo univoco all'interno di una relazione. Ai vincoli già esplicitati si aggiungono quelli interrelazionali che definiscono le Foreign Key tra relazioni.
- Possibilità di creare viste per differenti utenti, queste ultime possono essere materializzate se avviene la memorizzazione come relazione nella base dati, oppure virtuali se vengono memorizzate. La scelta rispetto ai tipi presentati è sintetizzabile nei trade-off spazio vs tempo e performance vs tempo.
- Supporto alle routine, stored function e stored procedure
- Possibilità di specificare privilegi e aree per ogni utente abilitato all'accesso e manipolazione della base dati
- Supporto a meccanismi di backup e restore al fine di far fronte ad eventuali guasti

Schema ER del database



N.B: L'attributo "sommaRaccolta" dell'entità Campagna è un attributo derivabile che si può ottenere calcolando la somma totale degli importi delle donazioni effettuate su di una campagna. Per questo motivo è stato deciso di accorpare l'attributo "sommaRaccolta" all'entità Campagna, in modo tale da evitare ripetute e costose operazioni di calcolo della somma raccolta.

Schema Logico

Utente (idUtente, dataBan*, admin, fotoProfilo, passwordHash, telefono, nome, cognome, email, strada, città, cap, cf, dataDiNascita);

Campagna (idCampagna, stato, titolo, descrizione, sommaRaccolta, sommaTarget, idCategoria, idUtente);

Donazione (idDonazione, dataOra, ricevuta, sommaDonata, commento, anonimo, idUtente, idCampagna);

Segnalazione (idSegnalazione, dataOra, descrizione, idUtenteSegnalatore, idUtenteSegnalato, stato, idCampagna);

Categoria (idCategoria, nomeCategoria);

Immagine (idImmagine, path, idCampagna);

FAQ (idFaq, domanda, risposta, idUtente);

Dizionario dei dati

Categoria

Nome Entità	Categoria		
Descrizione	Entità che accomuna campagne con uno scopo simile		
Nome Campo	Tipo	Vincolo di chiave	Altri vincoli
NomeCategoria	varchar(100)		NOT NULL
ID_Categoria	Int	PRIMARY KEY	NOT NULL, AUTO_INCREMENT

Donazione

Nome Entità	Donazione		
Descrizione	Entità che rappresenta le donazioni effettuate da un utente verso una campagna		
Nome Campo	Tipo	Vincolo di chiave	Altri vincoli
IDdonazione	Int	PRIMARY KEY	NOT NULL, AUTO_INCREMENT
DataOra	Datetime		NOT NULL
Ricevuta	Varchar(200)		NOT NULL
SommaDonata	Double		NOT NULL
Commento	Varchar(150)		
Anonimo	Tinyint(1)		NOT NULL, Default = 0
IDutente	Int	FOREIGN KEY(Utente)	NOT NULL
IDcampagna	Int	FOREIGN KEY(Campagna)	NOT NULL

Immagine

Nome Entità	Immagine		
Descrizione	Entità che rappresenta le immagini descrittive relative ad una campagna		
Nome Campo	Tipo	Vincolo di chiave	Altri vincoli
IDcampagna	Int	FOREIGN KEY(Campagna)	NOT NULL
Path	Varchar(300)		NOT NULL
ID_Immagine	Int	PRIMARY KEY	NOT NULL

Campagna

Nome Entità	Campagna		
Descrizione	Entità che rappresenta le campagne create da un utente		
Nome Campo	Tipo	Vincolo di chiave	Altri vincoli
IDcampagna	Int	PRIMARY KEY	NOT NULL, AUTO_INCREMENT
StatoCampagna	Enum(Attiva, Chiusa, Cancellata)		NOT NULL, Default = Attiva
Titolo	Varchar(150)		NOT NULL
Descrizione	Text(3000)		NOT NULL
SommaRaccolta	Double		NOT NULL
SommaTarget	Double		NOT NULL
IdCategoria	Int	FOREIGN KEY (Categoria)	NOT NULL
IDutente	Int	FOREIGN KEY (Utente)	NOT NULL

FAQ

Nome Entità	FAQ		
Descrizione	Entità che racchiude le domande degli utenti registrati alla piattaforma e le risposte degli admin.		
Nome Campo	Tipo	Vincolo di chiave	Altri vincoli
IDfaq	Int	PRIMARY KEY	NOT NULL, AUTO_INCREMENT
Domanda	Varchar(200)		NOT NULL
Risposta	Varchar(200)		NOT NULL
IDutente	Int	FOREIGN KEY (Utente)	NOT NULL

Utente

Nome Entità	Utente		
Descrizione	Entità che descrive una persona registrata sulla piattaforma		
Nome Campo	Tipo	Vincolo di chiave	Altri vincoli
IDutente	Int	PRIMARY KEY	NOT NULL, AUTO_INCREMENT
DataBan	DateTime		Default = NULL
Admin	Tinyint(1)		NOT NULL, Default = 0
FotoProfilo	Varchar(200)		NOT NULL, Default = placeholder
Passwordhash	Varchar(65)		NOT NULL
Telefono	Varchar(15)		NOT NULL
Nome	Varchar(50)		NOT NULL
Cognome	Varchar(50)		NOT NULL
Email	Varchar(70)		NOT NULL
Strada	Varchar(70)		NOT NULL
Città	Varchar(30)		NOT NULL
Cap	Varchar(5)		NOT NULL
CF	Varchar(20)		NOT NULL
dataDiNascita	Date		NOT NULL

Segnalazione

Nome Entità	Segnalazione		
Descrizione	Entità che rappresenta le segnalazioni effettuate da un utente verso una campagna		
Nome Campo	Tipo	Vincolo di chiave	Altri vincoli
IDsegnalazione	Int	PRIMARY KEY	NOT NULL, AUTO_INCREMENT
DataOra	Datetime		NOT NULL
Descrizione	Varchar(300)		NOT NULL
Stato	Enum(Attiva, Risolta, Archiviata)		NOT NULL, Default = Attiva
IDutenteSegnalatore	Int	FOREIGN KEY (Utente)	NOT NULL
IDutenteSegnalato	Int	FOREIGN KEY(Utente)	NOT NULL
IDcampagna	Int	FOREIGN KEY(Campagna)	NOT NULL

Controllo degli accessi e sicurezza

Attori	Guest	Utente Semplice	Utente Admin
Oggetti			
Utente	Registrazione Login	Login Logout VisualizzaDashboard ModificaProfilo	Login Logout VisualizzaDashboard ModificaProfilo VisualizzaUtenti SospensioneUtente
FAQ	VisualizzaPaginaFAQ	VisualizzaPaginaFAQ	VisualizzaPaginaFAQ CancellaPaginaFAQ ModificaPaginaFAQ InserisciPaginaFAQ
Donazione		VisualizzaDonazioni EffettuaDonazione	VisualizzaDonazioni EffettuaDonazione
Segnalazione		SegnalaCampagna	SegnalaCampagna VisualizzaSegnalazioni RisolviSegnalazione
Campagna	RicercaCampagna CondividiCampagna VisualizzaCampagna	CreaCampagna RicercaCampagna CancellaCampagna ModificaCampagna ScriviCommento CondividiCampagna ChiudiCampagna VisualizzaCampagna	CreaCampagna RicercaCampagna CancellaCampagna ModificaCampagna ScriviCommento CondividiCampagna ChiudiCampagna VisualizzaCampagna
Categoria	VisualizzaCategorie	VisualizzaCategorie	VisualizzaCategorie ModificaCategoria InserisciCategoria



Controllo globale del software

Il sistema software Fund.It é una web application di tipo event-driven. L'utente farà uso di un'interfaccia grafica per creare un evento che sarà gestito da un apposito handler. Lato backend tutte le richieste HTTP/S verranno intercettate al fine di eseguire un dispatch verso il sottosistema appropriato ed in base al tipo di evento verranno chiamati servizi dell'interface layer che si occuperà di costruire una risposta appropriata per l'utente.

Condizioni limite

Boundary conditions per l'avvio del sistema, spegnimento del sistema e fallimento del sistema.

UCBC_1 – Spegnimento del sistema

Identificativo	UCBC_1 - Spegnimento del sistema	Data	03/01/2022
		Versione	0.2
		Autore	Crescenzo Luigi
Descrizione	Questo use case descrive i passi da eseguire per effettuare lo spegnimento del sistema		
Attore Principale	Amministratore del sistema		
Attori secondari	MySQL; Tomcat;		
Entry Condition	Il sistema è in funzione; L'amministratore ha eseguito l'accesso sulla piattaforma server su cui è installata la piattaforma;		
Exit condition On success	Tutte le risorse usate dal sistema vengono rilasciate; Il sistema va nello stato "offline";		
Exit condition On failure	Le risorse usate dal sistema vengono parzialmente rilasciate; Viene mostrato un log sulla console con relativi errori; I servizi del sistema sono parzialmente funzionanti;		
FLUSSO DI EVENTI PRINCIPALE			
1	Amministratore:	Accede alla console di log dell'application container e spegne il server.	
2	Sistema:	In caso di errore, stamperà il corrispettivo log nella console del server sulla quale è deployato. In caso affermativo, verrà undeployato dal server.	
3	Amministratore:	Entra nel portale di Microsoft Azure ed arresta la risorsa database	
4	Sistema MySQL:	Esegue il comando e inizia l'arresto della risorsa database	
I Scenario/Flusso di eventi Alternativo: La risorsa database non viene arrestata correttamente			
4.1	Sistema MySQL:	Genera un log con relativo errore sul portale Azure	
4.2	Amministratore:	Ritenta l'arresto della risorsa database	
II Scenario/Flusso di eventi Alternativo: Il server Tomcat non riesce ad arrestarsi correttamente			
2.1	Sistema Tomcat:	Genera un log con relativo errore sulla console	
2.2	Amministratore:	Ritenta l'arresto del server Tomcat	

UCBC_2 - Avvio del sistema

Identificativo	UCBC_2 - Avvio del sistema	Data	03/01/2022
		Versione	0.2
		Autore	Genovese Sabato
Descrizione	Lo use case descrive l'avvio del sistema		
Attore Principale	Amministratore		
Attori secondari	NA		
Entry Condition	L'amministratore effettua con successo il login al server; Il sistema si trova nello stato spento		
Exit condition On success	Il sistema viene avviato e tutti servizi sono disponibili		
Exit condition On failure	Il sistema non cambia stato.		
FLUSSO DI EVENTI PRINCIPALE			
1	Amministratore:	Esegue l'accesso alla piattaforma Azure e avvia il server MySQL dalla sezione apposita	
2	Sistema (MySQL):	Procede ad avviare il servizio richiesto, esegue alcuni check sui dati persistenti per verificare la consistenza dei dati, rende disponibili tutti i servizi per gli utenti.	
3	Amministratore:	Esegue sul server i comandi per avviare il web server Tomcat	
4	Sistema:	Procede ad avviare il servizio richiesto, esegue alcuni check sulla configurazione, rende disponibile la web application al pubblico.	
I Scenario/Flusso di eventi Alternativo: Il server MySQL non riesce ad avviarsi correttamente			
2.1	Sistema (MySQL):	Mostra un log con l'errore	
2.2	Amministratore:	Corregge l'errore ed esegue il passo 1 del flusso principale	
II Scenario/Flusso di eventi Alternativo: Il web server Tomcat non riesce ad avviarsi correttamente			
4.1	Sistema:	Mostra un log con l'errore	
4.2	Attore:	Corregge l'errore ed esegue il passo 3 del flusso principale	

UCBC_3 - Fallimento del sistema

Identificativo	UCBC_3 - Fallimento del sistema	Data	03/01/2022
		Versione	0.1
		Autore	Meo Angelo
Descrizione	Questo caso d’uso descrive il comportamento del sistema in caso di fallimento.		
Attore Principale	Amministratore		
Attori secondari	NA		
Entry Condition	Il sistema viene terminato in modo anomalo		
Exit condition On success	Il sistema viene riavviato correttamente		
Exit condition On failure	Il sistema continua a rispondere in modo anomalo e non viene riavviato		
FLUSSO DI EVENTI PRINCIPALE			
1	Amministratore:	Include UCBC_2	

UCBC_4 - Errore di accesso ai dati persistenti

Identificativo	UCBC_4 - Errore di accesso ai dati persistenti	Data	23/12/2021
		Versione	0.1
		Autori	Francesco Esposito
Descrizione	Questo caso d’uso descrive il comportamento del sistema in caso di errori nell’accesso ai dati persistenti o nel caso che tali dati siano corrotti.		
Attore Principale	Amministratore		
Attori secondari	NA		
Entry Condition	Il sistema non riesce ad accedere ai dati persistenti o i dati persistenti sono corrotti.		
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:	Avvisa l’amministratore dell’impossibilità di accedere ai dati nel database e cessa di processare eventuali richieste esterne.	
2	Amministratore:	L’amministratore spegne il sistema, ripristina il database utilizzando l’ultimo backup disponibile e riavvia il sistema.	

Servizi dei sottosistemi

In questa sezione vengono descritti i servizi di ogni sottosistema precedentemente elencati.

Sottosistema Autenticazione

Servizio	Descrizione	Interfaccia
Login	Questo servizio permette di effettuare l'accesso al sistema per permettere all'utente registrato di usufruire di tutte le funzionalità offerte dalla piattaforma	AutenticazioneService
Logout	Questo servizio permette di disconnettersi dal sistema	AutenticazioneService
Registrazione	Questo servizio permette di effettuare la registrazione di un account	AutenticazioneService

Sottosistema Gestione Utente

Servizio	Descrizione	Interfaccia
Visualizza Dashboard Utente	Questo servizio permette all'utente di visualizzare la propria area utente	UtenteService
Modifica Profilo	Questo servizio permette la modifica dei dati dell'utente	UtenteService
Visualizza Utenti	Questo servizio permette all'utente amministratore di visualizzare la lista degli utenti registrati alla piattaforma	UtenteService
Promuovi/Declassa Utente	Questo servizio permette all'utente amministratore di promuovere un utente semplice allo status di utente admin o di declassare un utente admin allo stato di utente semplice	UtenteService
Sospensione utente	Questo servizio permette all'utente amministratore di sospendere un utente per un periodo di tempo	UtenteService

Sottosistema Gestione Categorie

Servizio	Descrizione	Interfaccia
Inserisci Categoria	Questo servizio permette all'utente amministratore di inserire una nuova categoria	CategoriaService
Modifica Categoria	Questo servizio permette all'utente amministratore la modifica di una categoria presente sulla piattaforma	CategoriaService
Visualizza Categorie	Questo servizio permette agli utenti la visualizzazione di tutte le categorie presenti sulla piattaforma	CategoriaService

Sottosistema Gestione Donazioni

Servizio	Descrizione	Interfaccia
Effettua Donazione	Questo servizio permette agli utenti di donare una somma di denaro ad una raccolta fondi	DonazioniService
Visualizza Donazioni	Questo servizio permette all'utente di visualizzare le donazioni da lui effettuate o all'admin di visualizzare tutte le donazioni effettuate sul sito.	DonazioniService
Commenta	Questo servizio permette al donatore di scrivere un commento sulla campagna	DonazioniService

Sottosistema Gestione Campagne

Servizio	Descrizione	Interfaccia
Ricerca Campagna	Questo servizio permette la visualizzazione delle campagne che corrispondono all'input della barra di ricerca.	CampagnaService
Trova Campagna	Questo servizio permette la visualizzazione di una specifica Campagna	CampagnaService
Ricerca Campagna Per Categoria	Questo servizio permette la ricerca di una campagna per categoria	CampagnaService
Creazione Campagna	Questo servizio permette all'utente la creazione di una propria campagna di raccolta fondi	CampagnaService
Modifica Campagna	Questo servizio permette agli utenti la modifica di una propria campagna	CampagnaService
Condividi Campagna	Questo servizio permette la condivisione di una campagna	CampagnaService
Chiudi Campagna	Questo servizio permette all'utente la chiusura di una propria campagna di raccolta fondi	CampagnaService
Visualizza Campagne	Questo servizio permette la visualizzazione delle campagne presenti sulla piattaforma	CampagnaService
Cancellazione di una Campagna	Questo servizio permette la cancellazione da parte dell'utente.	CampagnaService
Visualizza Campagne Attive	Questo servizio ottiene tutte le campagne Attive	CampagnaService

Sottosistema Gestione Segnalazioni

Servizio	Descrizione	Interfaccia
Segnala Campagna	Questo servizio permette agli utenti di segnalare una campagna sospetta	SegnalazioniService
Visualizza Segnalazioni	Questo servizio permette all'utente amministratore di visualizzare le segnalazioni effettuate sulla piattaforma ed il loro stato	SegnalazioniService
Risolvi Segnalazione	Questo servizio permette all'utente amministratore di risolvere una segnalazione verso una campagna	SegnalazioniService

Sottosistema Gestione FAQ

Servizio	Descrizione	Interfaccia
Inserisci FAQ	Questo servizio permette la creazione di una pagina FAQ	FAQService
Cancella FAQ	Questo servizio permette la cancellazione di una pagina FAQ	FAQService
Modifica FAQ	Questo servizio permette la modifica di una pagina FAQ esistente	FAQService
Visualizza FAQ	Questo servizio permette la visualizzazione delle pagine FAQ presenti sulla piattaforma	FAQService



Glossario

- Fundraiser = soggetto che crea la raccolta fondi
- DBMS = Database Management System
- Microsoft Azure = Piattaforma di proprietà della Microsoft che offre servizi cloud