SDD: System Design Document

ScheduFIRE

Riferimento	RAD: Requirement Analysis Document
Versione	0.5
Data	02/12/2019
Destinatario	Top Management
Presentato da	Annunziata Giusy, Bombardelli Emanuele, Bruno Biagio, Cipolletta Ciro, Giuliano Alfredo, Labanca Nicola, Perillo Francesca, Sottile Eugenio
Approvato da	

Revision History

Data	Versione	Cambiamenti	Autori
27/11/2019	0.1	Aggiunta e revisione: cap. 1.1 Obiettivi del sistema; cap. 1.2 Design Goals; cap. 1.3, 1.4, 1.5.	Bombardelli Emanuele Perillo Francesca Bruno Biagio Labanca Nicola Giuliano Alfredo Sottile Eugenio
28/11/2019	0.2	Aggiunta e revisione: cap. 2 Architettura del Sistema corrente; cap. 3.1 Panoramica; cap.3.2 Decomposizione in sottoinsiemi	Labanca Nicola Sottile Eugenio Giuliano Alfredo Bombardelli Emanuele Bruno Biagio Cipolletta Ciro
29/11/2019	0.3	Aggiunta e revisione: cap. 3.3 Mapping hardware/software; cap. 3.4 Gestione dei dati Persistenti; cap. 3.5 Controllo degli accessi e sicurezza	Cipolletta Ciro Sottile Eugenio Annunziata Giusy Perillo Francesca Giuliano Alfredo
30/11/2019	0.4	Aggiunta e revisione: cap. 3.6 Controllo flusso globale del sistema; cap. 3.7 Condizione limite; cap. 4 Servizi dei Sottosistemi.	Annunziata Giusy Bombardelli Emanuele Bruno Biagio Labanca Nicola Perillo Francesca
02/12/2019	0.5	Revisione qualità del documento	Sottile Eugenio
17/12/2019	0.6	Aggiunta elementi nel cap. 3.4 Gestione dei dati Persistenti	Sottile Eugenio

Sommario

1. Introduzione

1.1 Obiettivi del sistema

Il sistema che si vuole realizzare ha come scopo la schedulazione equa, trasparente e chiara delle squadre dei vigili del fuoco della provincia di Salerno da parte dei capi turno provinciali, e la visualizzazione da parte dei vigili.

Il nostro obiettivo è di realizzare un sistema che permetta di facilitare questo procedimento attraverso l'implementazione di un database per poter conservare gli estremi di identificazione dei vigili, i relativi giorni di ferie / malattie e le varie composizioni delle squadre per poter accedere in automatico ai dati utili, e l'implementazione di un algoritmo di schedulazione che garantisca un carico di lavoro equamente distribuito, conservando però una buona modificabilità da parte del capo turno in casi estremi.

Per raggiungere questo obiettivo è stato individuato l'attuale processo di schedulazione eseguito dai capo turni e la pubblicazione delle squadre non digitale.

Il miglioramento della procedura porterebbe altri vantaggi, tra cui la possibilità di ampliare la piattaforma all'utilizzo anche regionale/nazionale, o l'aggiunta di altre funzioni eseguite dai capo turni in Italia in maniera manuale.

Il sistema progettato è una web app alla quale potranno accedere capo turni (per la funzionalità di schedulazione, gestione delle squadre *et similia*) ed il personale della caserma (per la visualizzazione del calendario).

Possiamo dividere il sistema in 4 categorie:

- Gestione delle squadre: scheduler dei turni, modifica dei turni creati nel caso di imprevisti, visualizzazione di una specifica squadra;
- Gestione del personale: modifica del personale in caserma (aggiunta / rimozione / modifica di un membro);
- Gestione delle email: email generate automaticamente dal sistema per notificare chiunque entri a far parte di una squadra in un determinato turno;
- Calendar displayer: costruzione del calendario consultabile.

Il sistema dovrà consentire il login ad entrambe le figure individuate (CT e VF) a seconda dei dati immessi.

Poiché la funzionalità della gestione dei turni è estremamente importante per la sicurezza nel suolo nazionale, il sistema dovrà implementare sul fronte sicurezza dei robusti algoritmi di crittografia per i dati di login in modo tale da proteggere il sistema da accessi non desiderati e/o malevoli.

Inoltre, poiché si vuole rendere il passaggio al sistema agevole e facile, il sistema dovrà essere usabile, e non dovrà prevedere né corsi di aggiornamento, né una documentazione necessaria per l'utilizzo corretto.

Infine, dal punto di vista della manutenibilità, poiché è messo in conto di dover aggiungere funzionalità ed ampliare la portata del sistema, il codice dovrà essere facilmente iterabile ed incrementabile, così come la documentazione.

1.2 Design Goals

ScheduFIRE è un sistema che si impone di rispettare vari obiettivi di progettazione, che includono: performance, affidabilità, costi, mantenimento e criteri End user. Di seguito verranno analizzati nello specifico.

Criteri di performance:
Il tempo di risposta del prodotto software deve essere breve. Gli utilizzatori del sistema ScheduFIRE non devono incorrere in tempi di attesa lunghi, bensì minimi nello svolgimento delle funzionalità offerte dal sistema. La memoria necessaria al funzionamento del sistema dipende dalla tracciabilità offerta dal database, che verrà gestito in modo da evitare al minimo la ridondanza dei dati contenuti al suo interno, al fine da evitare l'utilizzo di memoria eccessiva.

Criteri di affidabilità:
SchediFIRE garantisce una buona robustezza, informando i clienti di eventuali errori logici che potrebbero verificarsi durante l'utilizzo del software (come per esempio l'immissione di input non validi). Il tutto verrà gestito tramite appositi messaggi di errore, che cercheranno di migliorare l'esperienza di utilizzo del sistema. ScheduFIRE si impone inoltre di garantire un elevato livello di affidabilità dei servizi proposti, con lo scopo di evitare un non corretto funzionamento delle funzionalità. Una volta caricato, il sistema sarà disponibile a tutti gli utenti che ne richiederanno l'utilizzo. Il sistema sarà sicuro, in quanto ogni utente può svolgere solo le operazioni a lui consentite.

Criteri di costi:
Per la progettazione e lo sviluppo del sistema è stimato un costo complessivo di 400 ore, calcolate sulla base del singolo costo di ogni project member, che sarà pari a 50 ore di lavoro.

Criteri di manutenzione:
Uno degli scopi principali di ScheduFIRE è l'estendibilità. Il sistema nasce con lo scopo di poter aggiungere facilmente nuove funzionalità, dettate dalle esigenze del cliente, dall'avvento di nuove tecnologie o di nuovi algoritmi che rendono la schedulazione più efficacie. Questa prima versione del sistema si limiterà ad essere utilizzata esclusivamente dal Corpo dei Vigili del Fuoco di Salerno, e schedulerà i turni relativi esclusivamente al turno B. Ciò non esclude che il sistema non possa subire adattamenti, in quanto, modificando i relativi controlli, potrebbe essere utilizzato per la schedulazione di più turni e utilizzato in svariati Corpi della provincia di Salerno.

Criteri di usabilità:
ScheduFIRE sarà di facile comprensione e utilizzo, permetterà di effettuare in modo semplice e immediato le varie operazioni grazie a una interfaccia user-friendly. Il sistema risulterà migliore rispetto agli odierni metodi utilizzati, in quanto sarà veloce ed eviterà un sacco di operazioni che attualmente vengono svolti dalla mente umana.

1.2.1 Design Trade-off

Memoria vs Estendibilità:
ScheduFIRE deve permettere l'estensibilità a discapito della memoria utilizzata. Tale preferenza permette al cliente di richiedere agli sviluppatori nuove funzionalità, dando meno importanza alla memoria utilizzata.

Tempo di risposta vs Affidabilità:
ScheduFIRE sarà implementato in modo tale da preferire l'affidabilità al tempo di risposta, in modo tale da garantire un controllo più accurato dei dati in input a discapito del tempo di risposta del sistema. Il tempo di risposta del sistema però non subirà mai un ritardo critico, per evitare un esso che risulti poco prestante.

Disponibilità vs Tolleranza ai guasti:

ScheduFIRE deve sempre essere disponibile all'utente in caso di errore in una funzionalità, anche al costo di rendere non disponibile quest'ultima per un lasso di tempo. Il malfunzionamento di una data funzionalità inoltre, non dovrà in nessun modo danneggiare il funzionamento di altre attività.

Criteri di manutenzione vs Criteri di performance:

ScheduFIRE sarà implementato preferendo la manutenibilità alla performance in modo da facilitare gli sviluppatori nel processo di aggiornamento del software a discapito delle performance del sistema. Questo perché ScheduFIRE nasce con lo scopo principale di subire una seconda implementazione, per essere utilizzato anche in tutta la provincia di Salerno.

1.3 Definizioni, acronimi e abbreviazioni

RAD: Requirements Analysis Document;

SDD: System Design Document;

VF: Vigile del fuoco;

VVF: Vigili del fuoco;

CT: Capo turno provinciale;

Squadra: Insieme di Vigili del fuoco adempienti ad una data mansione comune.

DB: DataBase**;**

MySQL: DataBase Open Source basato sul linguaggio SQL;

DBMS: Database Management System;

SQL: Structured Query Language; linguaggio standardizzato per database basati sul modello relazionale;

1.4 Riferimenti

ScheduFIRE_RAD_v.1.7

Slide del corso, presenti sulla piattaforma e-learning,

Libro: Object-Oriented Software Engineering (Using UML, Patterns, and Java) Third Edition
Autori: Bernd Bruegge & Allen H. Dutoit

1.5 Panoramica

Capitolo 1: Evidenzia l'obiettivo del sistema, un elenco di definizioni, acronimi e abbreviazioni e i design goals. Questi risultano utili per comprendere la documentazione.

Capitolo 2: Descrive, le funzionalità offerte dal sistema corrente ed un breve accenno a ciò che si vuole sviluppare.

Capitolo 3: Si presenta l'architettura del sistema proposto, con una panoramica di ciò che fornirà il sistema software, si espliciterà la decomposizione in sottosistemi, il mapping hardware/software, la gestione dei dati persistenti, il controllo degli accessi e sicurezza, il controllo sul flusso globale del sistema ed in fine le condizioni limite.

Capitolo 4: Vengono presentati i servizi dei sottosistemi anche tramite diagrammi UML.

2. Architettura del Sistema Corrente

Attualmente, non esiste un prodotto software finalizzato alla schedulazione dei VVF nelle rispettive squadre, per un determinato giorno di lavoro, in base al carico lavorativo.

Questo lavoro viene effettuato manualmente dal CT, tramite la compilazione di un apposito foglio Excel.

Il CT provvede a formare le squadre del turno a cui fa capo, bilanciando il carico di lavoro che grava sui vari VVF mediante un accurato ragionamento e confronto in base ai dati a sua disposizione, che possono risultare non esaustivi. Inoltre, il CT deve preoccuparsi che ad ogni squadra siano assegnati i giusti membri, abilitati a svolgere le giuste mansioni.

Le squadre sono diverse e sono: *Sala operativa, prima partenza, autoscala* e *autobotte*.

I VVF saranno poi informati riguardo la squadra di appartenenza, soltanto la mattina all'adunata.

Inoltre, il CT deve poter assegnare le ferie al personale a sua disposizione, provvedendo a sostituire i VVF che non risulteranno più disponibili per quel determinato periodo. Stesso discorso vale per i giorni di malattia, che richiedono un intervento tempestivo in quanto non riguardano periodi programmati.

Altri compiti del CT, non meno importanti, riguardano la possibilità, in casi di necessità, di aggiungere VVF e modificare i dati del personale.

Tutto ciò risulta molto sconveniente e gravoso per il CT, in quanto consiste di un compito laborioso e dispendioso.

Per agevolare le operazioni sopra citate, si è deciso di sviluppare una piattaforma online che permetta di svolgere le varie procedure in modo automatico, veloce e flessibile, in modo da ottimizzare i tempi e fornire un servizio affidabile per il contorto compito spettante al CT, comunicando anche, in maniera automatica ai vari VVF, la squadra di assegnazione.

Questa piattaforma concederà al CT la possibilità di generare le squadre per i vari giorni lavorativi (odierno e successivi), modificare la composizione delle squadre generate, inserire il periodo di ferie, tenendo traccia delle ferie residue, inserire il periodo di malattia, rimuovere i VVF dai giorni in cui risultano assenti, gestire il personale a disposizione e visualizzare il calendario comprendente l'effettiva generazione delle squadre. Inoltre, avviserà automaticamente i VVF facenti parte del turno a cui fa capo il CT, riguardo la composizione delle squadre generate, tramite una e-mail e darà la possibilità di accesso anche ai VVF, visualizzando così il calendario di lavoro.

La procedura risulterà quindi, molto rapida, ordinata ed esaustiva, ovviando inoltre a possibili errori che possono presentarsi mediante un dispendioso ed arduo lavoro umano.

3. Architettura del Sistema Proposto

3.1 Panoramica

Il sistema da noi proposto è un'applicazione web sviluppata tramite la Greenfield Engineering, in quanto non esiste nessun sistema che si occupi della problematica affrontata. L'applicazione sarà installata in locale, per motivi di sicurezza. Gli utenti saranno di due tipi: Capo Turno e Vigile del Fuoco che saranno già registrati al sistema, e potranno accedervi tramite login e disconnettersi tramite

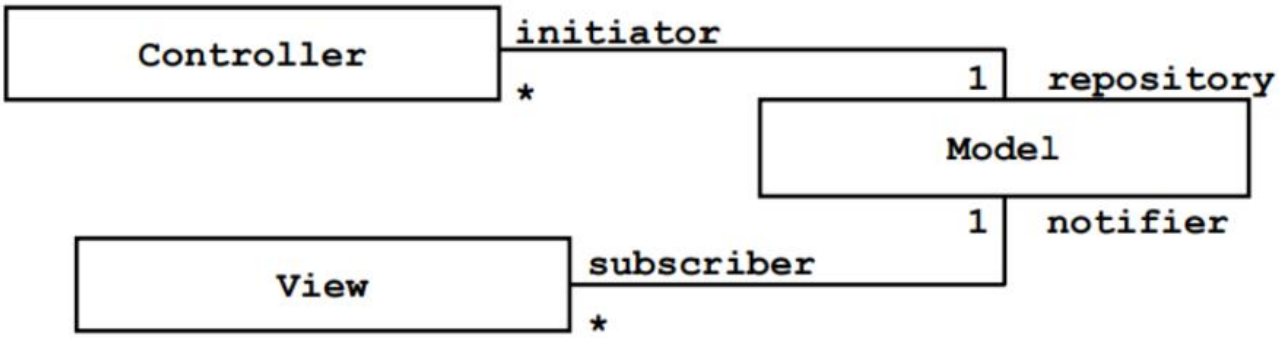
log-out. A seconda della tipologia di utente, la piattaforma offrirà diverse funzionalità. In particolare, il CT potrà generare le squadre di VVF in base ai giorni lavorativi del turno; visualizzare il calendario con i giorni lavorativi e le squadre generate; modificare le squadre generate; aggiungere, modificare ed eliminare un VF dal sistema; assegnare giorni di ferie; concedere giorni di malattia. Alla generazione delle squadre da parte del CT, verrà recapitata a tutti i VVF una mail informativa. I VVF, invece, tramite le credenziali del turno, potranno visualizzare il calendario e le squadre generate. Lo stile architetturale adottato è di tipo repository, più specificamente è un sistema basato sul modello MVC. Il modello MVC (Model View Controller) è un pattern architetturale molto diffuso nello sviluppo di software object-oriented, in grado di separare l'interfaccia grafica utente, dalla logica di business. È un'architettura multi-tier, in quanto le varie funzionalità del sito sono logicamente separate e suddivise su più livelli software differenti, in comunicazione tra loro.

3.2 Decomposizione in sottosistemi

3.2.1 Decomposizione in Layer

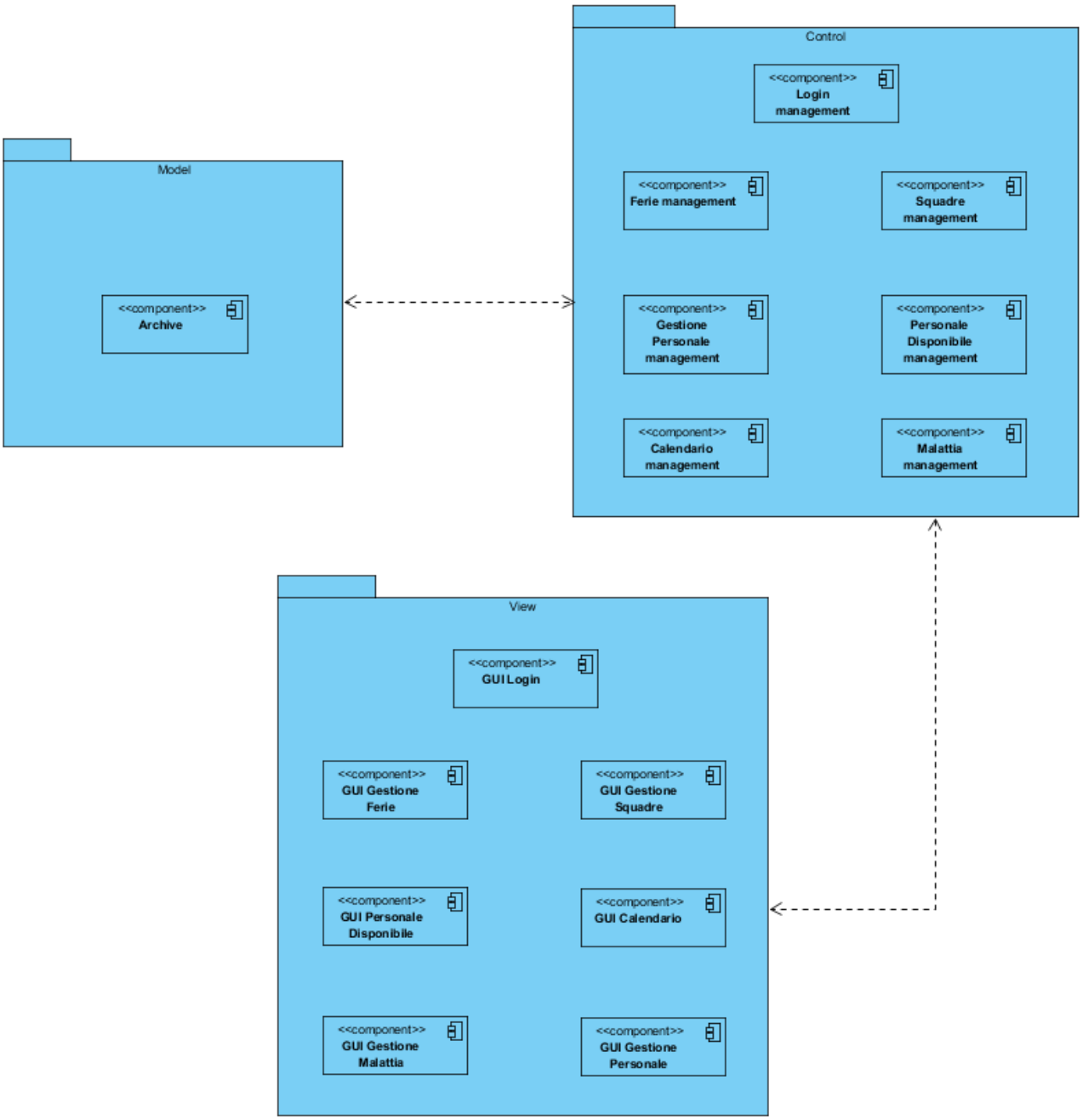
È stata prevista una decomposizione per il sistema basata sulla suddivisione in tre layer, ognuno dei quali gestisce diverse funzionalità:

- Model: si occupa della gestione dei dati persistenti implementando la struttura dati centrale;
- Control: integra la gestione della logica del sistema, tramite il controllo e l'elaborazione dei dati;
- View: si occupa di presentare e gestire un'interfaccia grafica per l'interazione tra l'utente e il sistema.



3.2.2 Decomposizione in sottosistemi

Abbiamo deciso di suddividere il sistema nei seguenti dodici sottosistemi, gestendo i singoli componenti con basso accoppiamento e alta coesione, in modo da garantire una elevata manutenibilità.



Il livello Model prevede la gestione del sottosistema:

- Archive

Il livello Control prevede la gestione dei sottosistemi:

- Ferie management
- Gestione Personale management
- Personale Disponibile management
- Malattia management
- Squadre management
- Calendario management
- Login management

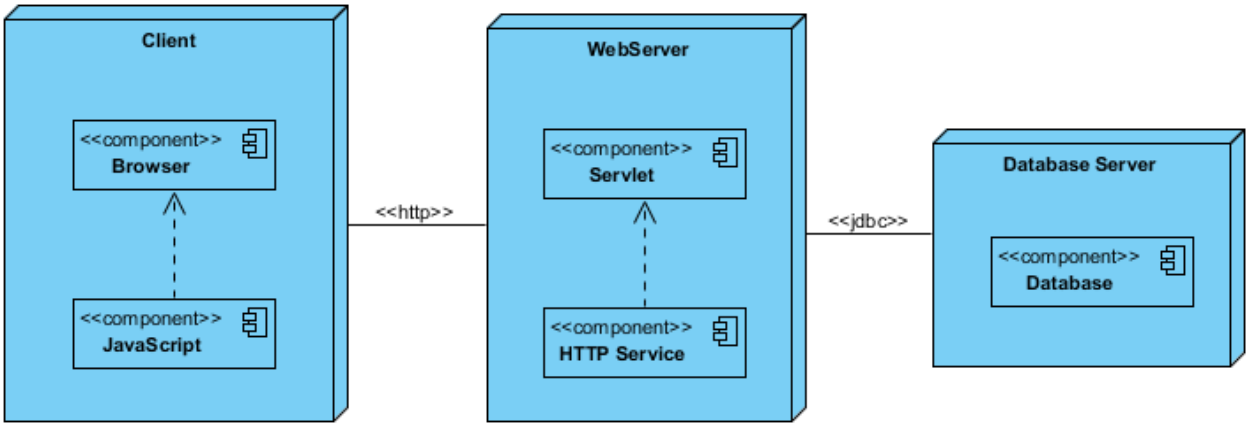
Il livello View prevede la gestione dei sottosistemi:

- GUI Gestione Ferie
- GUI Personale Disponibile
- GUI Gestione Malattia
- GUI Gestione Squadre
- GUI Calendario
- GUI Gestione Personale
- GUI Login

Deployment Diagram

L'utente usando il browser richiede la funzionalità tramite l'interfaccia messa a disposizione dal sistema, l'unico vincolo è la necessità di usare un browser che supporti la tecnologia JavaScript per poter usufruire di tutte le funzionalità del sistema.

Il browser si collega tramite protocollo HTTP al WebServer, nel quale vengono eseguite le funzioni per gestire la richiesta della funzionalità. Se opportuno, il WebServer si interfaccia tramite API JDBC ad un Database Server per gestire la persistenza dei dati.

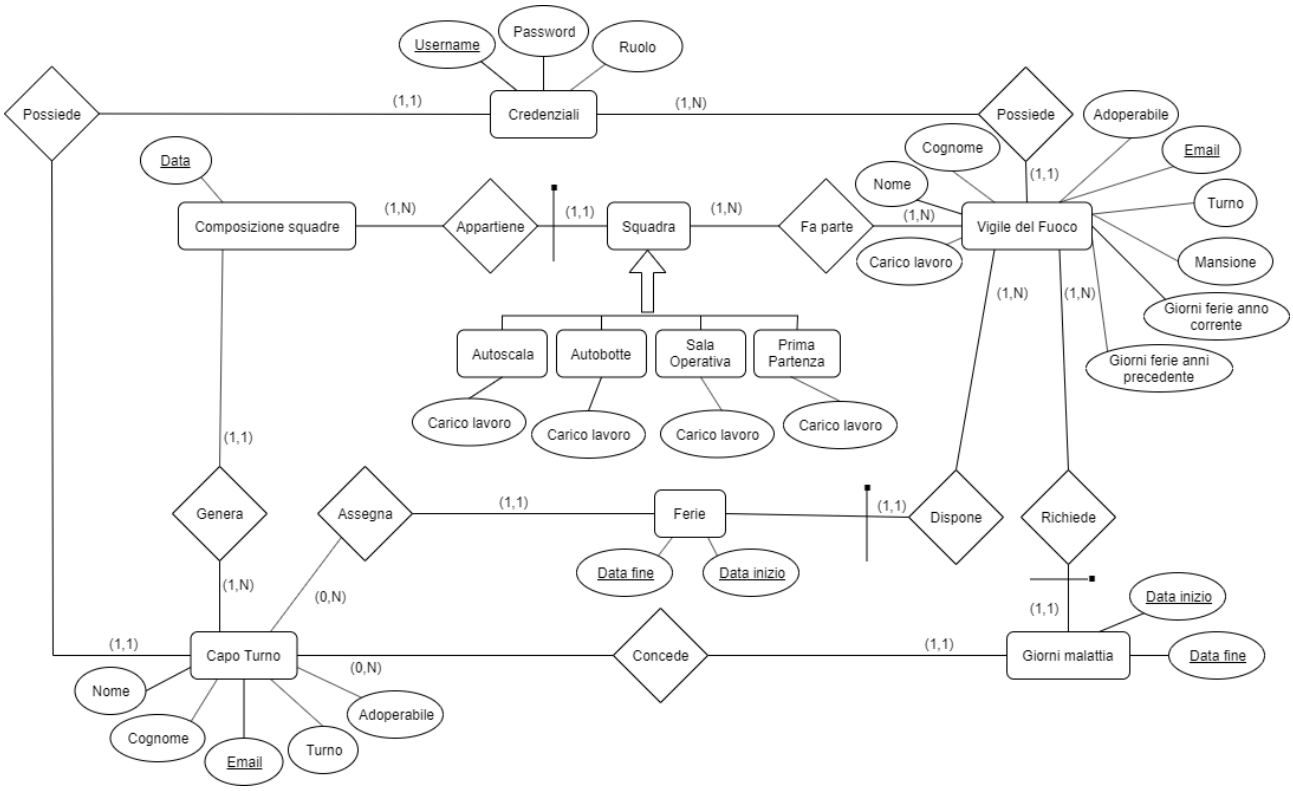


3.3 Mapping Hardware/Software

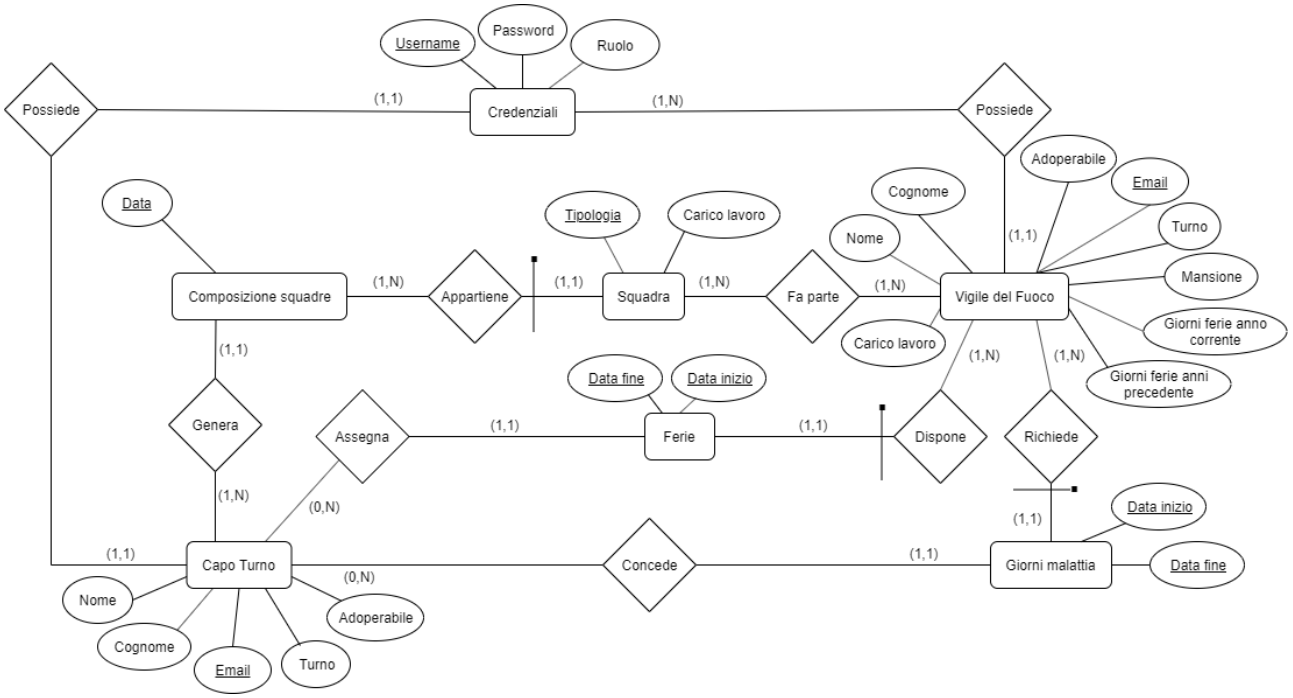
Il sistema che si desidera sviluppare utilizzerà una struttura hardware costituita da un Server che risponderà ai servizi richiesti dai client. Il client è un qualsiasi apparecchio attraverso il quale un utente può collegarsi per accedere al sistema mentre la macchina server gestisce la logica e i dati contenuti nel database. L'accesso al database avviene tramite JDBC, mentre la connessione tra client e il server avverrà tramite il protocollo http, con il quale il client inoltra delle richieste al server e quest'ultimo provvederà a fornire i servizi richiesti. Le componenti hardware e software per il client sono un computer, con installato su di esso un web browser, mentre il server necessita di immagazzinare una discreta quantità di dati. La componente software necessaria è dunque un DBMS, per consentire la comunicazione con più client.

3.4 Gestione dati persistenti

Per la gestione dei dati persistenti si è adottato un Database relazionale che permette accessi efficienti ai dati e un ampio spazio per la memorizzazione. Il diagramma E/R è una trasposizione del Class Diagram, sviluppato nel RAD; questa scelta deriva da una maggiore dimestichezza nel trattamento dei modelli relazionali. Questa scelta garantisce l'accesso concorrente ai dati in maniera affidabile, in quanto viene salvata una copia di backup dei dati ed è quindi possibile ripristinarli in caso di danni software o hardware. Inoltre, il DBMS protegge l'accesso ai dati, quindi diversi utenti aventi funzionalità diverse possono accedere ad aree distinte del database.ù



Nella fase di ristrutturazione del diagramma, si è deciso per l'eliminazione della generalizzazione dell'entità "Squadra", in quanto l'attributo "Carico lavoro" crea ridondanza. Nel seguente diagramma, quindi, si aggiungerà l'attributo "Tipologia" all'entità "Squadra".



CapoTurno

Email	Nome	Cognome	Turno	Adoperabile

ComposizioneSquadre

Data

Squadra

Tipologia **CaricoLavoro**

VigileDelFuoco

Email **Nome** **Cognome** **Turno** **Mansione** **CaricoLavoro** **GiorniFerieAnnoCorrente** **GiorniFerieAnnoPrecedente** **Adoperabile**

Credenziali

Username **Password** **Ruolo**

Ferie

DataInizio **DataFine**

GiorniMalattia

DataInizio **DataFine**

CapoTurno

NAME	TYPE	NULL	KEY
Email	VARCHAR (50)	NOT NULL	PRIMARY KEY
Nome	VARCHAR (50)	NOT NULL	
Cognome	VARCHAR (50)	NOT NULL	
Turno	CHAR	NOT NULL	
Adoperabile	BOOLEAN	NOT NULL	
Username	VARCHAR (50)	NOT NULL	FOREIGN KEY

ComposizioneSquadre

NAME	TYPE	NULL	KEY
Data	DATETIME	NOT NULL	PRIMARY KEY
EmailCT	VARCHAR (50)	NOT NULL	FOREIGN KEY

Squadra

NAME	TYPE	NULL	KEY
Tipologia	VARCHAR (50)	NOT NULL	PRIMARY KEY
CaricoLavoro	INT	NOT NULL	

DataComposizioneSquadra	DATETIME	NOT NULL	PRIMARY KEY FOREIGN KEY

ComponenteDellaSquadra

NAME	TYPE	NULL	KEY
Tipologia	VARCHAR (50)	NOT NULL	PRIMARY KEY
EMAIL_VF	VARCHAR (50)	NOT NULL	PRIMARY KEY FOREIGN KEY
DataComposizioneSquadra	DATETIME	NOT NULL	PRIMARY KEY FOREIGN KEY

VigileDelFuoco

NAME	TYPE	NULL	KEY
Email	VARCHAR (50)	NOT NULL	PRIMARY KEY
Nome	VARCHAR (50)	NOT NULL	
Cognome	VARCHAR (50)	NOT NULL	
Turno	CHAR	NOT NULL	
Mansione	VARCHAR (50)	NOT NULL	
GiorniFerieAnnoCorrente	INT	NOT NULL	
GiorniFerieAnnoPrecedente	INT	NOT NULL	
CaricoLavoro	INT	NOT NULL	
Adoperabile	BOOLEAN	NOT NULL	
Username	VARCHAR (50)	NOT NULL	FOREIGN KEY

Credenziali

NAME	TYPE	NULL	KEY
Username	VARCHAR (50)	NOT NULL	PRIMARY KEY
Password	CHAR(64)	NOT NULL	
Ruolo	VARCHAR (50)	NOT NULL	

Ferie

NAME	TYPE	NULL	KEY
DataInizio	DATE	NOT NULL	PRIMARY KEY
DataFine	DATE	NOT NULL	PRIMARY KEY
EmailCT	VARCHAR (50)	NOT NULL	FOREIGN KEY
EmailVF	VARCHAR (50)	NOT NULL	FOREIGN KEY

GiorniMalattia

NAME	TYPE	NULL	KEY

DataInizio	DATE	NOT NULL	PRIMARY KEY
DataFine	DATE	NOT NULL	PRIMARY KEY
EmailCT	VARCHAR (50)	NOT NULL	FOREIGN KEY
EmailVF	VARCHAR (50)	NOT NULL	FOREIGN KEY

3.5 Controllo degli accessi di sicurezza

ScheduFIRE consente l'accesso alla piattaforma a due diverse tipologie di attore. Il CT che ha il permesso di eseguire tutte le operazioni disponibili, e il VF che risulta molto limitato nell'utilizzo della piattaforma. Si è utilizzata una matrice degli accessi per schematizzare al meglio il controllo degli accessi. Le righe rappresentano gli attori, mentre le colonne le classi e per ogni attore, sono segnate le azioni consentite da esso sulle istanze delle varie classi.

Attori	Sistema
VF	• Login • Visualizza Calendario • Visualizza Squadre • Logout
CT	• Login • Visualizza Calendario • Gestione Squadre • Visualizza VF Disponibile • Aggiungi VF • Rimuovi VF • Modifica VF • Aggiungi Malattie • Aggiungi Ferie • Rimuovere Ferie • Rimuovere malattie • Logout

3.6 Controllo flusso globale del sistema

Il sistema ScheduFIRE è interattivo, pensato per facilitare la visualizzazione dei turni e la gestione delle squadre, dunque ogni funzionalità del sistema viene avviata grazie ad un comando impartito da CT, tramite un'interfaccia grafica. Dunque, basta selezionare sull'interfaccia il controllo corrispondente alla funzionalità del sistema che si desidera svolgere. Il sistema resta in attesa di un'azione da parte di un attore, dunque ecco perché richiede una continua interazione da parte dell'utente, ragione per cui, il controllo del flusso globale del sistema è di tipo event-driven, ovvero guidato dagli eventi.

3.7 Condizioni limite

Startup

Per il primo startup del sistema è necessario l'avvio di un Web Server che fornisca il deployement e l'utilizzo di un database MySQL per la gestione dei dati persistenti. La popolazione di questo avverrà automaticamente tramite script al primo utilizzo.

Quando un utente si reca sulla piattaforma, gli verrà mostrata una pagina di login dove poter inserire le proprie credenziali (email "*@vigilfuoco.it" e password) e, nel caso di credenziali valide, mostrerà all'utente la relativa Home Page dalla quale potrà accedere a tutte le funzionalità offerte dal sistema.

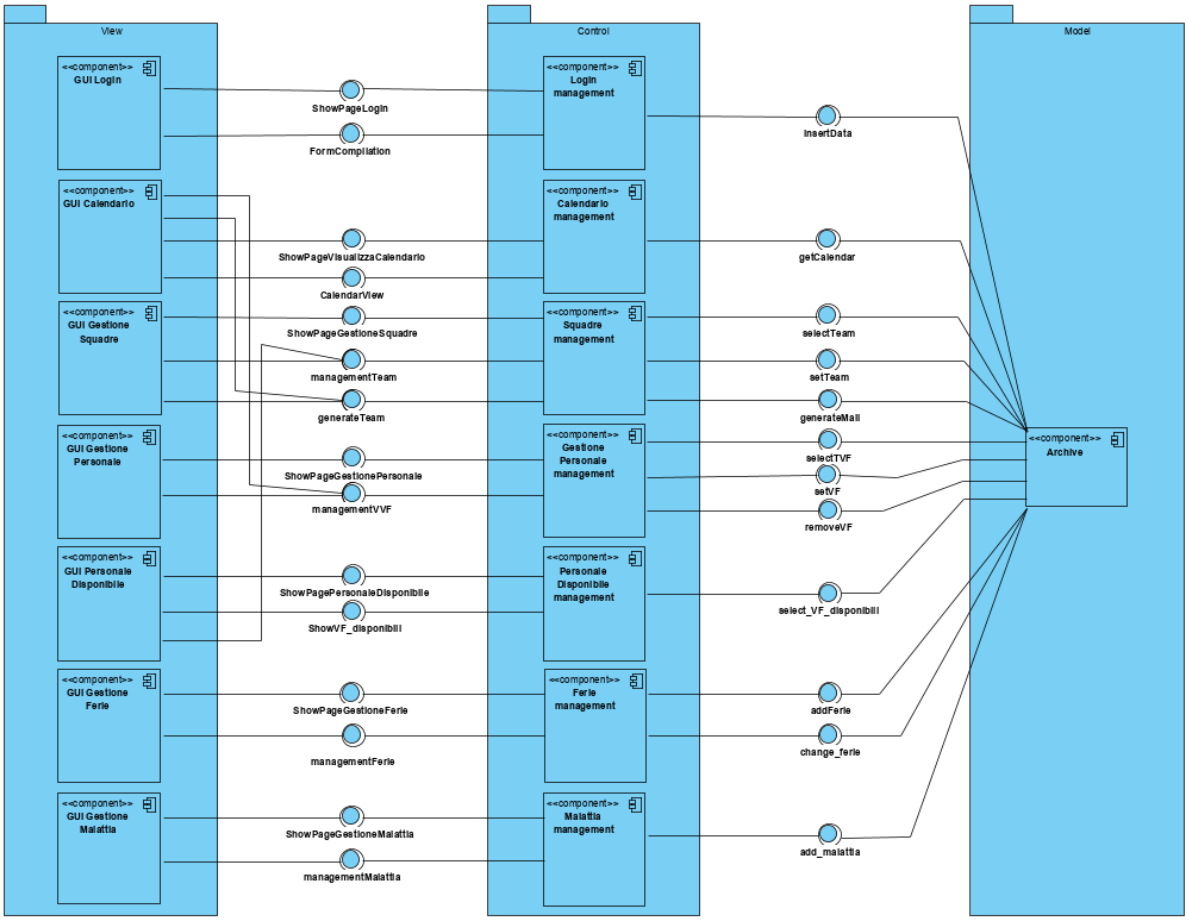
Terminazione

La chiusura della sessione dell'utente per una corretta terminazione della piattaforma avverrà tramite la pressione del pulsante di logout. Non è prevista una funzionalità di terminazione del server (poiché questo dovrà essere utilizzabile costantemente da più sedi separate), se non per brevi periodi di manutenzione a carico del team.

Fallimento

1. Nel caso di sovraccarico del database che porta a conseguenti guasti, è prevista una funzionalità di salvataggio preventiva periodica dei dati presenti sotto forma di codice SQL per permettere il roll-back del DB al momento più vicino al guasto;
2. Nel caso di un'interruzione inaspettata della corrente non è prevista funzionalità di ripristino al momento prima del danno, poiché è improbabile che ciò comprometta la consistenza del database o il normale funzionamento della piattaforma;
3. Se la terminazione inaspettata dovesse accadere per colpa di un errore dell'implementazione (consultabile tramite apposito file di log), come per il caso di fallimento 2, non è prevista funzionalità di ripristino per gli stessi motivi;
4. Se il fallimento dovesse essere causato dalla rottura dell'hardware dell'utente, non è prevista funzionalità di ripristino per i motivi sopra citati.

4. Servizi dei Sottosistemi



GUI Login offre 2 servizi all'interfaccia Control:

- ShowPageLogin
- FormCompilation

GUI Calendario offre 4 servizi all'interfaccia Control:

- ShowPageVisualizzaCalendario
- CalendarView
- generateTeam
- managementVVF

GUI Gestione Squadre offre 3 servizi all'interfaccia Control:

- ShowPageGestioneSquadre
- managementTeam
- generateTeam

GUI Gestione Personale offre 2 servizi all'interfaccia Control:

- ShowPageGestionePersonale
- managementVVF

GUI Gestione Personale offre 2 servizi all'interfaccia Control:

- ShowPageGestionePersonale
- managementVVF

GUI Personale Disponibile offre 3 servizi all'interfaccia Control:

- ShowPagePersonaleDisponibile
- ShowVF_disponibili
- managementTeam

GUI Gestione Ferie offre 2 servizi all'interfaccia Control:

- ShowPageGestioneFerie
- managementFerie

GUI Gestione Malattia offre 2 servizi all'interfaccia Control:

- ShowPageGestioneMalattia
- managementMalattia

Login management offre 1 servizo all'interfaccia Model:

- insertData

Calendario management offre 1 servizo all'interfaccia Model:

- getCalendar

Squadre management offre 3 servizi all'interfaccia Model:

- selectTeam
- setTeam
- generateMail

Gestione Personale management offre 3 servizi all'interfaccia Model:
- selectTVF
- setVF
- removeVF

Personale Disponibile management offre 1 servizo all'interfaccia Model:
- select_VF_disponibili

Ferie management offre 2 servizi all'interfaccia Model:
- addFerie
- change_ferie

Malattia management offre 1 servizo all'interfaccia Model:
- add_malattia