



# TEAM **COMMITTED**

UNIVERSITÀ DEGLI STUDI DI PADOVA

---

Specifica Tecnica V2.0

---

## Informazioni sul documento

<b>Nome documento</b>	Specifica Tecnica
<b>Data documento</b>	2012/01/30
<b>Redattori</b>	<ul style="list-style-type: none"><li>• Gabriele Facchin</li><li>• Lorenzo Braghetto</li><li>• Marco Begolo</li><li>• Giacomo Quadrio</li></ul>
<b>Verificatori</b>	<ul style="list-style-type: none"><li>• Massimo Dalla Pietà</li><li>• Alessandro Cornaglia</li><li>• Giorgio Maggiolo</li><li>• Lorenzo Braghetto</li></ul>
<b>Approvazione</b>	<ul style="list-style-type: none"><li>• Giacomo Quadrio</li><li>• Alessandro Cornaglia</li></ul>
<b>Uso documento</b>	Interno
<b>Lista distribuzione</b>	<ul style="list-style-type: none"><li>• <i>Team Committed</i></li><li>• <i>Prof. Tullio Vardanega</i></li></ul>

## Sommario

Questo documento vuol definire l'architettura generale che il prodotto dovrà avere.

## *Diario delle modifiche*

Modifica	Autore	Data	Versione
<i>Approvato.</i>	Alessandro Cornaglia	2012/06/03	V2.0
<i>Verificato. In attesa di approvazione</i>	Lorenzo Braghetto	2012/06/02	V1.2
<i>Apportate correzioni segnalate in RP. In attesa di verifica.</i>	Begolo Marco	2012/06/01	V1.1
<i>Approvato.</i>	Massimo Dalla Pietà	2012/01/30	V1.0
<i>Verificato. In attesa di approvazione</i>	Alessandro Cornaglia	2012/01/30	V0.16
<i>Completato capitolo 12 (Prototipi di interfaccia utente). In attesa di verifica</i>	Marco Begolo	2012/01/30	V0.15
<i>Realizzato Capitolo 11 : Tracciamento requisiti - componenti</i>	Gabriele Facchin	2012/01/29	V0.14
<i>Realizzato Capitolo 8 : Back-End</i>	Gabriele Facchin	2012/01/29	V0.13
<i>Realizzato Capitolo 10 : Tracciamento componenti - requisiti</i>	Marco Begolo	2012/01/28	V0.12
<i>Realizzato Capitolo 9 : Diagramma di attività</i>	Lorenzo Braghetto	2012/01/25	V0.11
<i>Realizzato Capitolo 13 : Introduzione</i>	Giacomo Quadrio	2012/01/24	V0.10
<i>Realizzato Capitolo 6 : Front-End Web</i>	Giacomo Quadrio	2012/01/23	V0.9
<i>Realizzato Capitolo 5 : Front-End Desktop</i>	Marco Begolo	2012/01/21	V0.8
<i>Realizzato Capitolo 7 : Front-End Mobile</i>	Lorenzo Braghetto	2012/01/18	V0.7
<i>Realizzato Capitolo 4 : Architettura Generale</i>	Giacomo Quadrio	2012/01/16	V0.6
<i>Realizzato Capitolo 3 : Strumenti</i>	Marco Begolo	2012/01/16	V0.5
<i>Iniziata realizzazione Capitolo 2: Design Pattern</i>	Giacomo Quadrio	2012/01/15	V0.4
<i>Realizzato Capitolo 1 : Introduzione</i>	Bragheto Lorenzo	2012/01/14	V0.3
<i>Sistemazione del file sorgente per fare in modo che includa tutti i capitoli come file a parte, così come la prima pagina. Iniziata la stesura del capitolo 3</i>	Gabriele Facchin	2012/01/13	V0.2
<i>Inizio creazione del documento. Creato lo schema base e iniziato a scrivere l'introduzione e il capitolo sulle comunicazioni. Impostato inoltre lo schema grafico iniziale del documento stesso.</i>	Gabriele Facchin	2012/01/13	V0.1

## Indice

<b>1 Introduzione</b>	<b>6</b>
1.1 Scopo del documento . . . . .	6
1.2 Scopo del prodotto . . . . .	6
1.3 Glossario . . . . .	6
1.4 Riferimenti . . . . .	6
1.4.1 Normativi . . . . .	6
1.4.2 Informativi . . . . .	6
<b>2 Architettura Generale</b>	<b>8</b>
<b>3 Design Pattern</b>	<b>12</b>
3.1 Multitier . . . . .	12
3.2 MVP . . . . .	13
3.3 Singleton . . . . .	14
3.4 Façade . . . . .	15
3.5 Data Access Object (DAO) . . . . .	16
3.6 Factory Method . . . . .	17
<b>4 Strumenti utilizzati</b>	<b>18</b>
4.1 Java . . . . .	18
4.2 XHTML . . . . .	18
4.3 MySql . . . . .	19
4.4 Android SDK . . . . .	19
4.5 JSP . . . . .	20
4.6 Tomcat . . . . .	20
<b>5 Front-end Desktop</b>	<b>21</b>
5.1 Presentation Tier . . . . .	21
5.1.1 com.safetyGame.desktop.view.Notifica . . . . .	21
5.1.2 com.safetyGame.desktop.view.Login . . . . .	22
5.1.3 com.safetyGame.desktop.view.Menu . . . . .	22
5.2 Control Tier . . . . .	23
5.2.1 com.safetyGame.desktop.logic.Timer . . . . .	23
5.2.2 com.safetyGame.desktop.logic.ControlNotifica . . . . .	23
5.2.3 com.safetyGame.desktop.logic.ControlLogin . . . . .	23
5.2.4 com.safetyGame.desktop.logic.ControlMenu . . . . .	24
5.2.5 com.safetyGame.desktop.logic.DatiLogin . . . . .	24
5.2.6 com.safetyGame.desktop.logic.ConnBack . . . . .	24
<b>6 Front-end Web</b>	<b>26</b>
6.1 View Tier . . . . .	26
6.1.1 com.safetyGame.frontWeb.pagine.Login . . . . .	26
6.1.2 com.safetyGame.frontWeb.pagine.Menu . . . . .	27
6.1.3 com.safetyGame.frontWeb.pagine.Notifica . . . . .	27
6.1.4 com.safetyGame.frontWeb.pagine.MostraDomanda . . . . .	27

6.1.5	com.safetyGame.frontWeb.pagine.MostraDatiD . . . . .	28
6.1.6	com.safetyGame.frontWeb.pagine.MostraDatiAA . . . . .	28
6.1.7	com.safetyGame.frontWeb.pagine.ModDatiD . . . . .	29
6.1.8	com.safetyGame.frontWeb.pagine.ModDatiAA . . . . .	29
6.1.9	com.safetyGame.frontWeb.pagine.ModDatiAS . . . . .	29
6.1.10	com.safetyGame.frontWeb.pagine.ElencoDomAS . . . . .	30
6.1.11	com.safetyGame.frontWeb.pagine.ModDomAS . . . . .	30
6.1.12	com.safetyGame.frontWeb.pagine.MostraStatD . . . . .	31
6.1.13	com.safetyGame.frontWeb.pagine.MostraStatAA . . . . .	31
6.1.14	com.safetyGame.frontWeb.pagine.RecuperoPass . . . . .	32
6.2	Logic Tier . . . . .	32
6.2.1	com.safetyGame.frontWeb.control.ControlLogin . . . . .	32
6.2.2	com.safetyGame.frontWeb.control.ControlMenu . . . . .	33
6.2.3	com.safetyGame.frontWeb.control.ControlNotifica . . . . .	33
6.2.4	com.safetyGame.frontWeb.control.ControlMostraDom . . . . .	33
6.2.5	com.safetyGame.frontWeb.control.ControlMostraDatiD . . . . .	34
6.2.6	com.safetyGame.frontWeb.control.ControlMostraDatiAA . . . . .	34
6.2.7	com.safetyGame.frontWeb.control.ControlModDatiD . . . . .	34
6.2.8	com.safetyGame.frontWeb.control.ControlModDatiAA . . . . .	35
6.2.9	com.safetyGame.frontWeb.control.ControlElencoDomAS . . . . .	35
6.2.10	com.safetyGame.frontWeb.control.ControlModDomAS . . . . .	35
6.2.11	com.safetyGame.frontWeb.control.ControlMostraStatD . . . . .	36
6.2.12	com.safetyGame.frontWeb.control.ControlMostraStatAA . . . . .	36
6.2.13	com.safetyGame.frontWeb.control.ControlModDatiAS . . . . .	36
6.2.14	com.safetyGame.frontWeb.control.ControlRecuperoPass . . . . .	37
6.2.15	com.safetyGame.frontWeb.control.Timer . . . . .	37
6.2.16	com.safetyGame.frontWeb.control.ConnHandler . . . . .	37
<b>7</b>	<b>Front-end Mobile</b>	<b>38</b>
7.1	View . . . . .	38
7.1.1	com.safetyGame.android.view.LoginActivity: . . . . .	38
7.1.2	com.safetyGame.android.view.DashboardActivity: . . . . .	39
7.1.3	com.safetyGame.android.view.PunteggiActivity: . . . . .	39
7.1.4	com.safetyGame.android.view.DatipersonaliActivity: . . . . .	39
7.1.5	com.safetyGame.android.view.DomandaActivity: . . . . .	39
7.1.6	com.safetyGame.android.view.RecuperoPasswordActivity: . . . . .	40
7.1.7	com.safetyGame.android.view.TimerNotifica: . . . . .	40
7.2	Logic . . . . .	40
7.2.1	com.safetyGame.android.utils.ConnectionUtils . . . . .	40
7.2.2	com.safetyGame.android.condivisi.Xxx . . . . .	41
<b>8</b>	<b>Back-end</b>	<b>42</b>
8.1	Presentation Tier . . . . .	42
8.1.1	com.safetyGame.back.connection.ApplicationConnection . . . . .	43
8.1.2	com.safetyGame.back.connection.WebConnection . . . . .	43
8.2	Logic Tier . . . . .	43

8.2.1	com.safetyGame.back.controller.GestioneDati . . . . .	44
8.2.2	com.safetyGame.back.controller.GestioneLogin . . . . .	44
8.2.3	com.safetyGame.back.controller.ControllerDomanda . . . . .	44
8.2.4	com.safetyGame.back.controller.ControllerCalderone . . . . .	45
8.2.5	com.safetyGame.back.controller.ControllerDipendenti . . . . .	45
8.2.6	com.safetyGame.back.controller.ControllerTrofei . . . . .	45
8.2.7	com.safetyGame.back.controller.ControllerDati . . . . .	46
8.2.8	com.safetyGame.back.controller.ControllerRecupero . . . . .	46
8.2.9	com.safetyGame.back.controller.ControllerPunteggi . . . . .	46
8.2.10	com.safetyGame.back.controller.ControllerLog . . . . .	47
8.3	Logic Tier . . . . .	47
8.3.1	com.safetyGame.back.checker.CheckXxx . . . . .	47
8.4	Data Access Tier . . . . .	48
8.4.1	com.safetyGame.back.access.AccessXxx . . . . .	48
8.4.2	com.safetyGame.back.condivisi.Xxx . . . . .	48
<b>9</b>	<b>Diagrammi di attività</b>	<b>49</b>
9.1	Ambito Dipendente . . . . .	49
9.2	Ambito Amministratore Azienda . . . . .	51
9.3	Ambito Amministratore Sicurezza . . . . .	52
9.4	Considerazioni finali . . . . .	53
<b>10</b>	<b>Tracciamento componenti - requisiti</b>	<b>54</b>
10.1	Desktop . . . . .	54
10.2	Web . . . . .	55
10.3	Mobile . . . . .	61
10.4	Back-end . . . . .	63
<b>11</b>	<b>Tracciamento requisiti - componenti</b>	<b>69</b>
11.1	Ambito Dipendente . . . . .	70
11.2	Ambito Amministratore Azienda . . . . .	73
11.3	Ambito Amministratore Sicurezza . . . . .	75
<b>12</b>	<b>Prototipi di interfaccia utente</b>	<b>78</b>
12.1	Interfaccia desktop . . . . .	78
12.2	Interfaccia web . . . . .	79
12.3	Interfaccia Mobile . . . . .	82
<b>13</b>	<b>Stime di fattibilità e di bisogno di risorse</b>	<b>85</b>

## 1 Introduzione

### 1.1 Scopo del documento

Il presente documento ha lo scopo di definire e mostrare le scelte progettuali, ad alto livello, che il gruppo **Team Committed** ha deciso di rispettare per la realizzazione del prodotto. Al suo interno verranno presentati i vari design pattern utilizzati nella creazione del prodotto, la struttura dei packages e le principali classi che li compongono.

### 1.2 Scopo del prodotto

Il prodotto denominato **SafetyGame** si propone di fornire uno strumento informatico per la gestione delle pratiche di sicurezza sul lavoro in modo dinamico, evitando corsi di formazione che spesso si dimostrano inutili per la poca attenzione prestata dai partecipanti.

Lo strumento si basa sul concetto di **gamification** che comporta competizione tra i dipendenti all'interno delle aziende creando un sano interesse per un argomento delicato come la sicurezza sul luogo di lavoro.

Il sistema è pensato sia per lavoratori che hanno una postazione fissa dotata di PC, sia per quelli che hanno la necessità di spostarsi e che quindi sono forniti di dispositivi mobili. Ad essi verranno poste periodicamente domande, di varia tipologia, le cui risposte comporteranno l'assegnazione di un punteggio generando una classifica aziendale.

### 1.3 Glossario

Al fine di evitare ogni ambiguità relativa al linguaggio e ai termini utilizzati nei documenti formali, il glossario viene incluso nel file **Glossario-V3.0.pdf**, dove vengono definiti e descritti i termini marcati da una sottolineatura.

### 1.4 Riferimenti

#### 1.4.1 Normativi

- **Norme di Progetto, v 3.0** (allegato **Norme\_di\_Progetto\_V3.0.pdf**)
- **Analisi dei Requisiti, v 3.0** (allegato **Analisi\_dei\_Requisiti\_V3.0.pdf**)

#### 1.4.2 Informativi

- **Java 7 JDK Documentation**  
<http://docs.oracle.com/javase/7/docs/api/>
- **W3School XHTML 1.1 Documentation**  
[http://www.w3school.com/html/html\\_xhtml.asp](http://www.w3school.com/html/html_xhtml.asp)
- **W3School CSS 3 Documentation**  
<http://www.w3school.com/css/default.asp>

- **MySql** Documentation  
<http://dev.mysql.com/doc/refman/5.5/en/index.html>
- **JSP** Documentation  
<http://java.sun.com/products/jsp/syntax/2.0/synstaxref20.html>
- **Tomcat** Documentation  
<http://tomcat.apache.org/tomcat-6.0-doc/index.html>
- *E. Gamma, R. Helm, R. Johnson, J. Vlissides (1995), Design Patterns, USA-Canada, Addison-Wesley.*

## 2 Architettura Generale

L'applicativo è stato progettato come unione di quattro sottoinsiemi, come mostrato in figura 2. Ogni sottosistema è stato inoltre suddiviso in componenti, i quali sono indipendenti tra di loro o, in ogni caso, con un basso indice di accoppiamento.

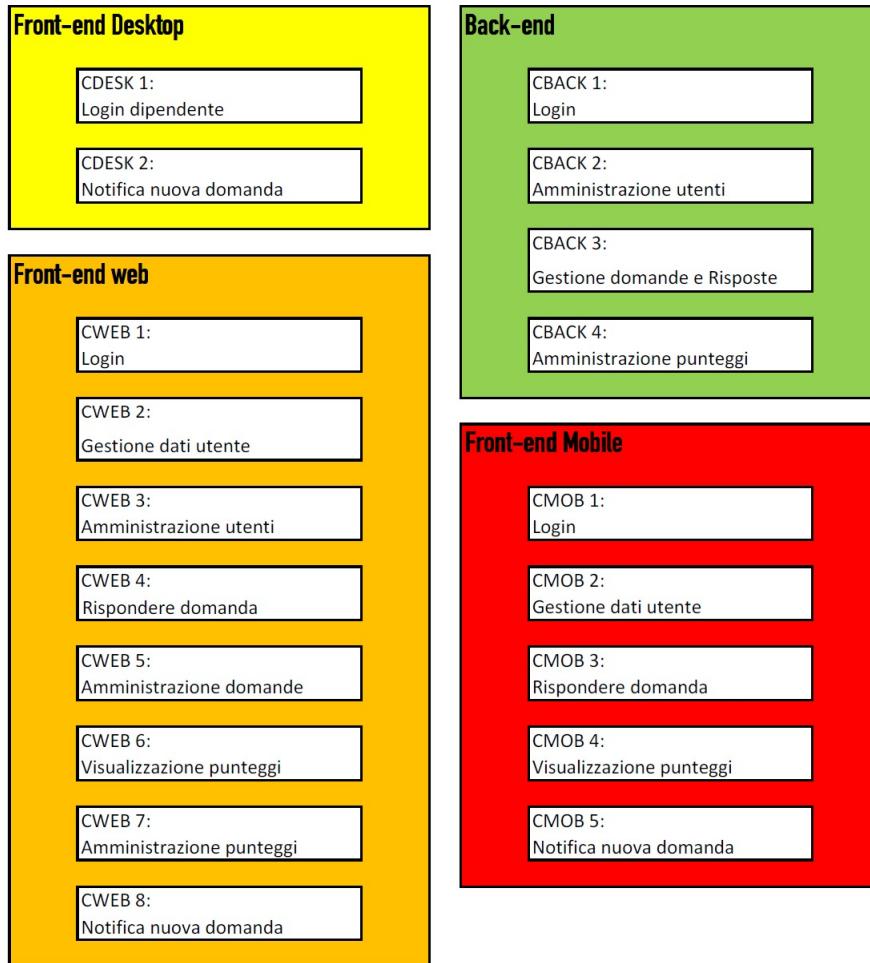


Figura 1: Architettura generale dell'intero sistema SafetyGame

Come specificato nell'*Analisi dei Requisiti*, il front-end è stato diviso in tre parti e principalmente si occuperanno di fornire l'interfaccia grafica agli utenti del sistema, quindi Dipendenti, Amministratori Installatori, Amministratori Sicurezza e Amministratori Azienda. In particolare abbiamo suddiviso il front-end:

- **Desktop:** il “front-end desktop” si occuperà di gestire la temporizzazione delle domande da sottoporre al Dipendente, mostrandogli un pop-up di notifica ogni qualvolta sia giunto il tempo di proporre una nuova domanda. Questo sistema è formato dai seguenti componenti:
  - **CDESK 1:** Login dipendente
  - **CDESK 2:** Notifica nuova domanda
- **Mobile:** il “front-end mobile” si occuperà dei dispositivi mobili Android. È composto dai seguenti componenti:
  - **CMOB 1:** Login
  - **CMOB 2:** Gestione dati utente
  - **CMOB 3:** Rispondere nuova domanda
  - **CMOB 4:** Visualizzazione punteggi
  - **CMOB 5:** Notifica nuova domanda
- **Web:** il “front-end web” gestirà la somministrazione delle domande sia al Dipendente connesso tramite applicazione desktop, sia a tutti gli utenti che sono connessi tramite browser. Esso è composto dai seguenti componenti:
  - **CWEB 1:** Login
  - **CWEB 2:** Gestione dati utente
  - **CWEB 3:** Amministrazione utenti
  - **CWEB 4:** Rispondere nuova domanda
  - **CWEB 5:** Amministrazione domande
  - **CWEB 6:** Visualizzazione punteggi
  - **CWEB 7:** Amministrazione punteggi
  - **CWEB 8:** Notifica nuova domanda

Il back-end si occuperà della persistenza delle informazioni e del loro recupero, in modo da renderle disponibili agli altri sottosistemi. È formato dai seguenti componenti:

- **CBACK 1:** Login
- **CBACK 2:** Amministrazione utenti
- **CBACK 3:** Gestione domande e Risposte
- **CBACK 4:** Amministrazione punteggi

Oltre a questa separazione in componenti, i sottosistemi sono stati divisi in livelli orizzontali, composti da elementi affini tra di loro, come mostrato in figura 2

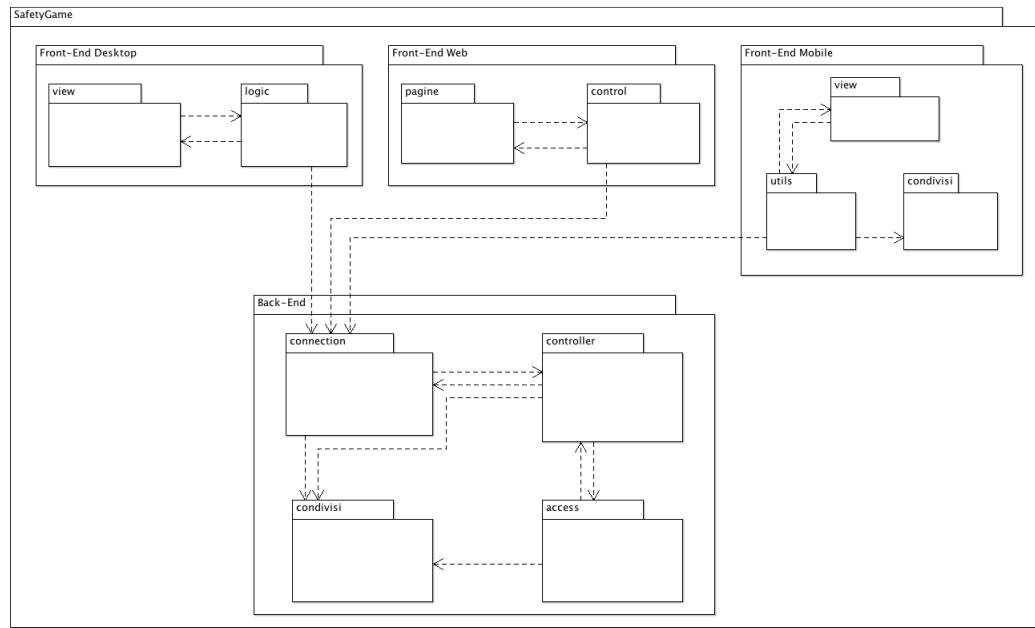


Figura 2: Architettura generale dell'intero sistema

- **com.safetyGame**  
Incorpora l'intero sistema SafetyGame
- **com.safetyGame.desktop**  
Racchiude al suo interno il Front-End Desktop
- **com.safetyGame.desktop.view**  
Definisce ed implementa l'interfaccia grafica usufruibile dagli utenti su di un dispositivo desktop
- **com.safetyGame.desktop.logic**  
Si occupa di controllare e gestire i dati che riceve in ingresso dall'interazione con l'utente tramite interfaccia grafica e di mantenere quest'ultima aggiornata
- **com.safetyGame.frontWeb**  
Racchiude al suo interno il Front-End Web
- **com.safetyGame.frontWeb.pagine**  
Costituisce la parte del sistema che definisce ed implementa l'interfaccia web usufruibile dagli utenti

- **com.safetyGame.frontWeb.control**

Costituisce la parte del sistema che si occupa di controllare e gestire i dati che riceve in ingresso l'applicazione dall'interazione con l'utente.

- **com.safetyGame.mobile**

Racchiude al suo interno il Front-End Mobile

- **com.safetyGame.mobile.view**

Contiene tutte le classi che definiscono la grafica dell'applicazione Android

- **com.safetyGame.mobile.utils**

Contiene le classi che comunicano con le API del server

- **com.safetyGame.mobile.condivisi**

Package che contiene tutti i tipi di dato non nativi usati dall'applicazione

- **com.safetyGame.back**

Contiene gli oggetti deputati alla creazione dell'intero back-end

- **com.safetyGame.back.connection**

Contiene gli oggetti deputati all'interfacciamento dei vari front-end con il back-end

- **com.safetyGame.back.controller**

Contiene gli oggetti deputati al controllo dei dati immessi nei vari front-end e del loro eventuale indirizzamento verso l'acces-tier

- **com.safetyGame.back.access**

Contiene tutte le classi che si occupano di eseguire le operazioni che il controller-tier chiede che vengano eseguite sui database aziendali e centrali

- **com.safetyGame.back.condivisi**

Contiene le classi che conterranno le informazioni che dovranno essere condivise fra i vari strati del back-end

### 3 Design Pattern

Presenteremo qui di seguito i vari design pattern impiegati nella progettazione dell'architettura del progetto SafetyGame.

#### 3.1 Multitier

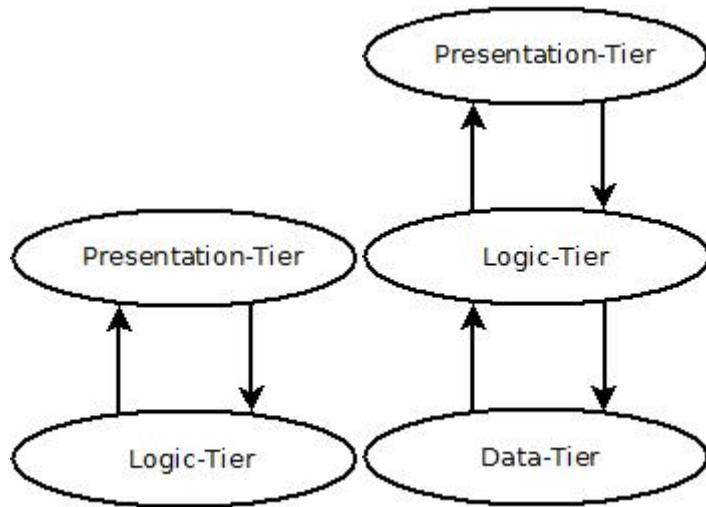


Figura 3: Diagramma dei design pattern TwoTier e ThreeTier

Il design pattern di tipo Multitier verrà implementato in forma *TwoTier* e *ThreeTier*.

- **Descrizione:** tale design pattern permette una disgiunzione fra i vari gruppi di entità che cooperano nell'erogazione del servizio. Esisterà un livello che si occuperà di interagire con il cliente offrendo l'interfaccia grafica e un secondo livello per l'esecuzione della parte algoritmica dell'applicazione.
- **Motivazione:** il beneficio principale apportato da questo paradigma risiede nel fatto che ogni livello può venir cambiato/aggiornato senza dover propagare gli effetti ai livelli adiacenti, essendo state definite delle precise interfacce fra i tier.  
Sebbene tale vantaggio derivi principalmente dal raggruppamento delle varie funzionalità in gruppi disgiunti, ma in stretta collaborazione, la struttura di tale design pattern risulta particolarmente calzante per dei servizi basati sull'architettura client-server, in quanto ogni livello non esiste semplicemente come raggruppamento logico a se stante, ma il suo ruolo viene adattato in relazione allo specifico ambiente di rete in cui esegue: nel caso

la morfologia della rete cambi, basterà aggiornare lo strato che lavorava nel determinato ambiente.

- **Contesto applicativo:**

- **TwoTier:** questo design pattern verrà utilizzato come struttura portante di tutti i Front-end in cui:
  1. *View-tier* offrirà l'interfaccia grafica agli utenti e si occuperà di reindirizzare le richieste dell'utente alle corrette operazioni
  2. *Logic-tier* si occuperà di gestire la temporizzazione delle domande, lo scambio di dati e la logica in generale
- **ThreeTier:** questo design pattern verrà utilizzato come struttura portante del Back-End in cui:
  1. *Presentation-tier* offrirà un'interfaccia di comunicazione fra i vari Front-End e i livelli sottostanti del Back-End
  2. *Logic-tier* si occuperà di gestire la parte algoritmica dell'applicazione, controllando la consistenza dei dati ricevuti dal Presentation-tier e dei dati richiesti al Data-tier
  3. *Data-tier* si occuperà dell'interazione con le basi di dati in cui saranno memorizzati i dati persistenti<sup>1</sup>

### 3.2 MVP

- **Descrizione:** Il design pattern MVP consente una completa disgiunzione fra le funzionalità che il prodotto deve offrire. È strutturato su tre componenti:

- **Model:** rappresenta la parte logica dell'applicazione, consente la memorizzazione ed il recupero dei dati presenti nel sistema
- **View:** rappresenta l'interfaccia grafica attraverso la quale gli utenti possono interagire con il sistema
- **Presenter:** rappresenta la componente che riceve richieste dalla componente View per la visualizzazione e le traduce in operazioni che agiranno sia sui dati che sulle viste

- **Motivazione:** Questo design pattern è stato scelto, e viene impiegato, per definire un'architettura del sistema che consenta di ottenere un buon livello di disaccoppiamento tra le viste e i modelli. L'architettura risulta inoltre essere solida e manutenibile, in quanto la modifica di una delle tre componenti non comporta alcun lavoro aggiuntivo sulle altre due componenti.

---

<sup>1</sup>Ad esempio le informazioni sugli utenti registrati nel sistema e le relative informazioni personali

- **Contesto applicativo:** il design pattern verrà utilizzato nello sviluppo dell'architettura generale per garantire al sistema una struttura ben organizzata e manutenibile. Sono state individuate tre macro componenti:

- *Model* ha il compito di astrarre i database dove sono presenti i dati del sistema e consentire il recupero di essi o l'inserimento di nuovi, senza che si debba intervenire sulle altre due componenti. Questo infatti perché le altre due componenti avranno solamente una visione di metodi esposti pubblicamente da parte di alcune classi, mentre l'implementazione effettiva dei metodi per interagire con le basi di dati sono nascosti ai vari componenti. Esso sarà implementato nel Data-tier del Back-end
- *View* ha il compito di rappresentare l'interfaccia grafica che consente l'interazione degli utenti con il sistema. Questo componente verrà implementato nei vari Front-end
- *Presenter* consente la comunicazione tra la componente View e quella Model. Presenter riceve le richieste da parte del View e le inoltra al Model sotto forma di chiamate ai metodi di questo, consentendo l'effettiva interazione tra utente e sistema. Inoltre Presenter indica alla View cosa deve essere mostrato agli utenti attraverso le opportune chiamate ad essa. Questo componente verrà implementato dai Presentation-tier e Logic-tier del Back-end

### 3.3 Singleton

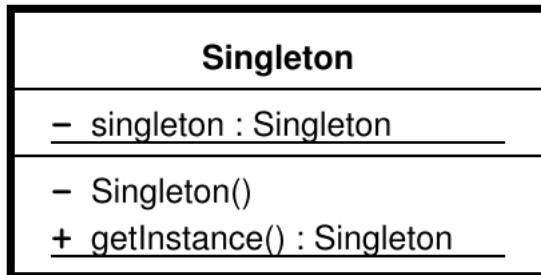


Figura 4: Esempio di Singleton

- **Descrizione:** Singleton rappresenta un design pattern creazionale che ha la funzione di garantire che di una data classe venga creata un'unica istanza, fornendo inoltre un punto di accesso globale ad essa.

- **Motivazione:** Si è scelto di utilizzare Singleton in quanto porta numerosi vantaggi tra i quali avere, come già detto, un accesso controllato all'unica istanza della classe, avere uno spazio di nomi ridotto ed evitare la dichiarazione di variabili globali. Un altro vantaggio rilevante è costituito dalla possibilità data dal principio della Lazy Initialization che permette la creazione dell'istanza della classe solo quando ciò diventa strettamente necessario.

Tutte le varie caratteristiche del design pattern Singleton comportano infine un ultimo vantaggio non da poco: il suo utilizzo comporta un notevole risparmio di risorse fisiche e questo lo rende particolarmente apprezzabile all'interno di applicativi web, migliorandone le prestazioni.

- **Contesto applicativo:** Singleton ha trovato applicazione nello strato Control Tier in ConnBack del front-end desktop, nello strato Logic Tier in ConnHandler del front-end web e nello strato Logic in ConnectionUtils del front-end mobile. Si è ritenuto utile utilizzare Singleton in queste circostanze per gestire in maniera unica le connessioni al back-end dei fari front-end del sistema.

### 3.4 Façade

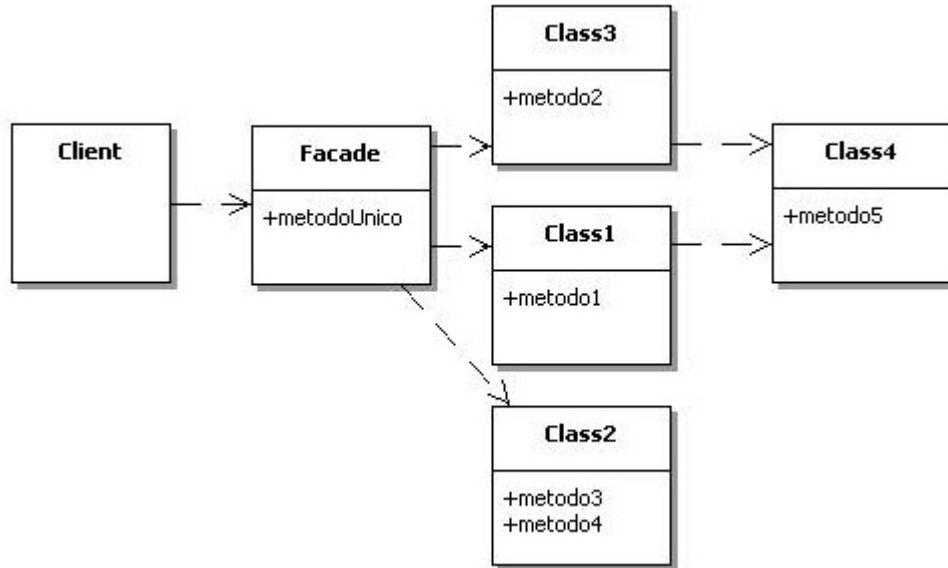


Figura 5: Esempio di façade

- **Descrizione:** Il design pattern façade permette, attraverso un'interfaccia più semplice, l'accesso a sottosistemi che espongono interfacce complesse e

molto diverse tra loro, nonché a blocchi di codice complessi ed è utilizzato, per esempio, nelle librerie di classi.

- **Motivazione:** L'implementazione di façade è molto semplice e comporta vantaggi molto interessanti. Nel dettaglio permette un migliore uso, comprensione e fase di testing delle classi che lo adottano, una maggiore leggibilità e riduce le dipendenze del codice esterno. In sostanza façade permette di nascondere la complessità di alcune operazioni e renderle più facilmente accessibili tramite l'unica interfaccia principale anziché andando a rivolgersi alle singole classi.
- **Contesto applicativo:** Façade ha trovato applicazione nella parte del Back-end, in dettaglio all'interno del Logic-Tier. Il Controller è infatti strutturato per fornire un'interfaccia unica dalla quale accedere per utilizzare tutt le varie classi necessarie come GestioneLogin, GestioneDomandeD, ecc.

### 3.5 Data Access Object (DAO)

- **Descrizione:** Un Data Access Object (DAO) è un oggetto che fornisce un'interfaccia astratta ad alcuni tipi di database, fornendo specifiche operazioni senza esporre dettagli del database.
- **Motivazione:** I motivi di implementazione si possono riscontrare nel fatto che DAO offre un'isolamento tale che separa i dati d'accesso di cui l'applicazione ha bisogno, in termini di oggetti e tipi di dato specifici del dominio (l'interfaccia pubblica di DAO), e come queste necessità possono essere soddisfatte da uno specifico DBMS (l'implementazione effettiva di DAO).  
Questo design pattern è stato scelto anche per la sua versatilità: esso si adatta a molti linguaggi di programmazione, tipologie di software con necessità di persistenza e a molti tipi di database. Il suo punto di forza è quindi permettere maggiore espandibilità al software che lo utilizza in quanto permette di, eventualmente, cambiare tipologia di database in maniera più semplice ed immediata.
- **Contesto applicativo:** DAO verrà utilizzato nel back-end del software SafetyGame, all'interno del Data Access Tier e avrà il compito di fornire i dovuti collegamenti ai vari database.

### 3.6 Factory Method

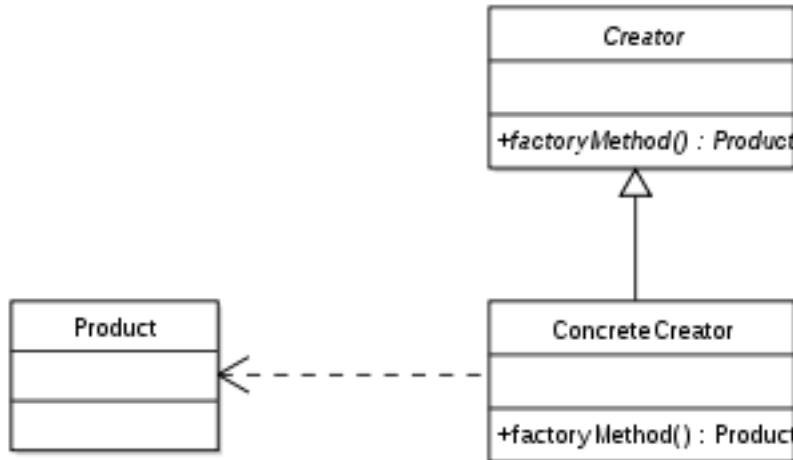


Figura 6: Schema del Factory Method

- **Descrizione:** Il Factory Method è un design pattern creazionale che fornisce un metodo per istanziare un oggetto senza sapere a priori la sua esatta classe.
- **Motivazione:** Il motivo per cui si è scelto questo design pattern va ricercato nel fatto che innanzitutto è uno dei design pattern fondamentali della programmazione ad oggetti e quindi, in quanto tale, risulta essere molto usato e supportato ed inoltre esso è particolarmente adatto allo sviluppo volto ai test. Nel dettaglio il pattern è molto interessante in quanto fornisce un'interfaccia per creare un oggetto ma lascia che le sottoclassi decidano quale oggetto istanziare.
- **Contesto applicativo:** Factory Method è utilizzato nel back-end, in particolare nello strato Data Access Tier - access. DAOFactory rappresenta il Creator cioè l'interfaccia che è fornita per creare un oggetto, SQLDAO-Factory il ConcreteCreator ed i vari SqlDAOLogic, Dipendenti ecc. i vari Product che permetteranno di istanziare l'oggetto necessario.

## 4 Strumenti utilizzati

### 4.1 Java

L'utilizzo di **Java 7** è legato all'approvazione ricevuta dal proponente durante l'incontro avuto con il *Proponente* in data 2011/12/05<sup>2</sup>

- **Vantaggi**

- **Multipiattaforma:** permettendo l'astrazione dalla macchina fisica tramite la JVM, assicura che l'applicativo risulti funzionante qualsiasi sia il sistema operativo installato sulla macchina che lo esegue<sup>3</sup>
- **Indipendenza dalle risorse fisiche:** per lo stesso motivo descritto sopra, l'esecuzione dell'applicativo risulterà uniforme indipendentemente dalle risorse fisiche utilizzate
- **Librerie:** alta disponibilità di librerie presenti in rete che semplificano l'interfacciamento con altri dispositivi e applicativi già esistenti
- **Compatibilità con piattaforme mobili:** considerata la richiesta di un'applicazione per dispositivi mobili<sup>4</sup>, l'utilizzo di tale linguaggio semplifica lo sviluppo dello stesso dell'applicazione mobile

- **Svantaggi**

- Non sono stati rilevati particolari svantaggi

### 4.2 XHTML

Si è deciso di utilizzare **XHTML 1.1** assieme a **CSS 2.1**<sup>5</sup> per la presentazione dell'interfaccia grafica dell'applicativo web sia agli utenti che per gli utenti autenticati. Si è scelto di utilizzare XHTML 1.1 anziché HTML5 in quanto quest'ultimo è uno standard ancora in fase di definizione e quindi soggetto ad eventuali modifiche. Inoltre l'applicazione sviluppata deve essere compatibile con il maggior numero di dispositivi e browser web possibili ed attualmente HTML5 è supportato unicamente dai browser web più moderni. Motivazioni simili ci hanno spinto ad utilizzare CSS 2.1 anziché CSS3 che, come HTML5, non è ancora supportato pienamente da tutti i browser web.

- **Vantaggi:**

- **Piattaforma mobile:** poichè la piattaforma deve essere fruibile anche da browser mobile, il semplice XHTML permette di definire agevolmente pagine web che si adattino al browser

<sup>2</sup>Rif: *verbale05122011.pdf*

<sup>3</sup>Requisito *RVOB 1.1*

<sup>4</sup>In particolare Android, come da requisito *RVOB 1.2*

<sup>5</sup>Requisito *RQD 6.2* e *RQD 6.1*

- **Uniformità di stile:** è possibile scrivere facilmente una struttura base dell'applicazione web dove poi andare ad inserire il contenuto tramite JSP

- **Svantaggi:**

- **Browser:** è possibile che alcuni browser (soprattutto quelli più datati) non riescano ad interpretare correttamente le informazioni contenute nelle pagine, rendendo difficoltosa, se non impossibile, la lettura delle pagine create con questo linguaggio

### 4.3 MySql

Per memorizzare tutti i dati che verranno immessi nel sistema SafetyGame, si è scelto di utilizzare la base di dati **MySql Community Server 5.5.20**

- **Vantaggi:**

- **Facilità d'uso:** poichè durante il corso di Basi di Dati abbiamo studiato i database relazioni (ed in particolare proprio MySql), l'uso di MySql ci è sembrata una scelta quasi obbligata
- **Relazioni con i vari componenti del sistema:** la documentazione riguardante l'interfacciamento con questa base di dati è di grande quantità, rispetto alle altre basi di dati

- **Svantaggi:**

- **Prestazioni:** poichè la versione Community Server non è stata pensata per supportare grandi carichi di lavoro<sup>6</sup>, quando<sup>7</sup> il database sarà oberato di lavoro e/o conterrà una quantità di dati abnorme le sue prestazioni potrebbero subire dei cali qualitativi. Per il momento abbiamo comunque deciso di tenere questa versione, a dispetto della *Enterprise* (ovvero quella a pagamento), perchè come base di dati per un prototipo è più che sufficiente

### 4.4 Android SDK

L'utilizzo dell'**Android SDK r16** è essenziale per lo sviluppo per l'applicazione mobile su Android

<sup>6</sup>Inteso, ad esempio, come migliaia di query nello stesso istante, piuttosto che contenere milioni di record in una singola tabella

<sup>7</sup>N.B. Non se, ma quando

## 4.5 JSP

**JSP 2.1** verrà utilizzato per generare dinamicamente le pagine che compongono l'applicativo web. A JSP si affiancherà l'uso dei JavaBeans, che permetteranno la possibilità di estendere agevolmente il codice senza dover creare alcuna nuova entità logica (secondo il modello MVC)

- **Vantaggi:**

- **Compatibilità con Java:** poichè gran parte del progetto sarà scritto in Java, l'utilizzo di JSP garantirà, con un certo grado di sicurezza, la compatibilità con i messaggi mandati dagli applicativi Java / Android
- **Sicurezza:** rispetto ad altri linguaggi di programmazione web (ex. php), JSP ha degli elevati standard di sicurezza interni che permettono di definire le funzionalità del programma senza troppi pensieri riguardanti la sicurezza interna dell'applicazione
- **Vasta libreria:** poichè basato su Java, JSP integra tutte le librerie native di Java. Questo ci permette, oltre che ad accedere a tutta una serie di codici già scritti, di riutilizzare parte della logica scritta per l'applicativo mobile, ovviamente modificando il codice da Java a JSP
- **Servlet e JavaBeans:** Oltre che sul server Tomcat, JSP si appoggia a due componenti vitali per il corretto funzionamento di un'applicazione web:
  - \* **Servlet:** applicazioni lato server che potenziano la capacità di fornire contenuti dinamici all'applicazione, alleggerendo il carico di lavoro al client e, soprattutto, riducendo il rischio di reverse-engineering sul codice creato per cercare eventuali falle di sicurezza
  - \* **JavaBeans:** piccole classi Java che permettono l'interazione fra differenti pagine JSP, quindi passibili di riutilizzo anche in componenti differenti dalla mera applicazione web

- **Svantaggi:**

- **Compilazione run-time:** ogni volta che si deve caricare una pagina JSP, se sono state fatte modifiche al contenuto della suddetta, il server dovrà compilare a run-time la pagina, provocando così un piccolo ritardo nel caricamento della stessa

## 4.6 Tomcat

**Tomcat 6.0.35** è un servlet reso necessario dall'uso di JSP.

## 5 Front-end Desktop

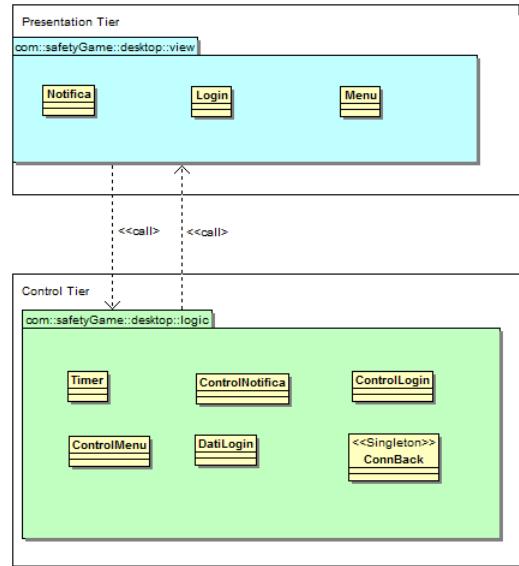


Figura 7: Front-end Desktop

### 5.1 Presentation Tier

**Tipo, obiettivo e funzione del componente:** questo livello costituisce la parte del sistema che definisce ed implementa l'interfaccia grafica usurfruibile dagli Utenti mediante l'uso di un dispositivo fisso.

**Relazioni d'uso di altre componenti:** il componente è costituito dal package view, e comunica con il Control Tier per rendere possibile la memorizzazione delle informazioni immesse nel sistema dagli utenti attraverso l'interfaccia web e per controllare i dati in transito per il sistema, dovuti all'interazione dell'utente con il sistema.

#### 5.1.1 com.safetyGame.desktop.view.Notifica

**Tipo, obiettivo e funzione del componente:** La classe Notifica definisce gli oggetti notifiche che compariranno a video al Dipendente Autenticato, le quali chiederanno la conferma per la visualizzazione di una domanda.

**Relazioni d'uso di altre componenti:** La classe Notifica utilizza i metodi messi a disposizione dalla classe ControlNotifica, contenuta nel package logic, per verificare le azioni compiute dagli utenti che interagiscono con la proposta di domanda.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene uti-

lizzata dalla classe ControlNotifica perchè viene aggiornata da questa a seconda delle azioni dell'utente e dalla classe Timer per la gestione delle temporizzazioni. **Attività svolte e dati trattati:** La classe definisce la struttura grafica della notifica che appare ad un Utente durante l'utilizzo del sistema. Essa resta in attesa che un utente fornisca una risposta alla notifica per poi delegare la classe ControlNotifica di effettuare i controlli del caso.

### 5.1.2 com.safetyGame.desktop.view.Login

**Tipo, obiettivo e funzione del componente:** La classe Login definisce la struttura, e la conseguente visualizzazione, dell'interfaccia grafica che consente ad un utente di effettuare il login al sistema tramite un dispositivo fisso.

**Relazioni d'uso di altre componenti:** La classe Login utilizza i metodi messi a disposizione della classe ControlLogin, contenuta nel package logic, per verificare i dati inseriti durante la fase di autenticazione degli utenti e la classe DatiLogin.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalla classe ControlLogin per visualizzare uno o più errori emersi nella fase di autenticazione da parte di un utente.

**Attività svolte e dati trattati:** La classe definisce la struttura dell'interfaccia grafica che consente agli utenti di autenticarsi al sistema. Essa resta in attesa che un utente inserisca i dati necessari per l'autenticazione al sistema per poi passarli alla classe ControlLogin, la quale svolgerà i vari controlli.

### 5.1.3 com.safetyGame.desktop.view.Menu

**Tipo, obiettivo e funzione del componente:** La classe Menu definisce la struttura, e la conseguente visualizzazione, dell'interfaccia grafica che mostra ad un utente, una volta autenticato, il menu contenente le varie azioni che questo può compiere nel sistema.

**Relazioni d'uso di altre componenti:** La classe Menu utilizza i metodi messi a disposizione della classe ControlMenu, contenuta nel package logic, per verificare le azioni compiute dagli utenti su di essa.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalla classe ControlMenu perchè viene aggiornata da questa a seconda delle azioni dell'utente. Anche la classe ControlLogin, del package logic, utilizza la classe Menu perchè richiama alcuni suoi metodi una volta che l'utente si è autenticato.

**Attività svolte e dati trattati:** La classe definisce la struttura grafica del menu che consente agli utenti loggati su un dispositivo fisso di navigare nel menu del sistema. Essa resta in attesa che un utente selezioni una delle voci presenti nel menu per poi delegare la classe ControlMenu di effettuare i controlli del caso.

## 5.2 Control Tier

**Tipo, obiettivo e funzione del componente:** questo livello costituisce la parte del sistema che si occupa sostanzialmente di controllare e gestire i dati che riceve in ingresso l'applicazione dall'interazione con l'utente, attraverso l'interfaccia grafica per dispositivi fissi, e di mantenere questa sempre aggiornata attuando le comunicazioni necessarie con il back-end.

**Relazioni d'uso di altre componenti:** il componente è costituito dal package logic, e comunica sia con il Presentation Tier per mantenere aggiornato lo stato dell'interfaccia per dispositivi fissi visibile agli utenti, che con il back-end per gestire le informazioni che l'utente scambia con l'applicazione.

### 5.2.1 com.safetyGame.desktop.logic.Timer

**Tipo, obiettivo e funzione del componente:** La classe Timer si occupa di gestire le varie temporizzazioni richieste dalle varie interfacce grafiche visibili agli utenti che sono autenticati su dispositivi fissi.

**Relazioni d'uso di altre componenti:** nessuna.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalla classe ControlNotifica per conoscere quando deve essere mostrata una notifica.

**Attività svolte e dati trattati:** La classe permette la gestione delle temporizzazioni richieste dal sistema.

### 5.2.2 com.safetyGame.desktop.logic.ControlNotifica

**Tipo, obiettivo e funzione del componente:** La classe ControlNotifica si occupa di gestire le notifiche che appaiono agli utenti una volta che sono autenticati nel sistema. Si occupa inoltre di catturare le risposte a queste, per poi passare tali informazioni al back-end.

**Relazioni d'uso di altre componenti:** La classe ControlNotifica utilizza i metodi messi a disposizione della classe Notifica contenuta nel package view. Usa i metodi di questa classe per lanciare e aggiornare le notifiche destinate agli utenti. La classe si occupa di informare il back-end sulle scelte fatte dagli utenti.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalla classe Notifica per tenere aggiornato il proprio stato e dal back-end che specifica il tipo di notifica da visualizzare.

**Attività svolte e dati trattati:** La classe ControlNotifica genera le notifiche volute dal back-end e resta in attesa delle azioni compite dagli utenti.

### 5.2.3 com.safetyGame.desktop.logic.ControlLogin

**Tipo, obiettivo e funzione del componente:** La classe ControlLogin si occupa di controllare le credenziali che un utente inserisce al momento dell'autenticazione, da dispositivo fisso, per poi trasmetterle al back-end. A seconda della risposta che riceve aggiorna la vista del sistema.

**Relazioni d'uso di altre componenti:** La classe ControlLogin utilizza i metodi messi a disposizione della classe Login, contenuta nel package view, per aggiornarne lo stato una volta che un utente ha provato ad autenticarsi inserendo credenziali errate. Se invece l'autenticazione va a buon fine la classe richiama metodi della classe Menu del package view.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalla classe Login per tenere aggiornato il proprio stato.

**Attività svolte e dati trattati:** La classe riceve in ingresso i dati necessari per l'autenticazione per poi passarli al back-end. In seguito alla risposta che ottiene aggiorna l'interfaccia grafica del sistema.

#### 5.2.4 com.safetyGame.desktop.logic.ControlMenu

**Tipo, obiettivo e funzione del componente:** La classe ControlMenu si occupa di controllare le azioni che un utente, una volta autenticato, compie nella schermata di menu. A seconda dell'azione che viene eseguita si occupa di aggiornare correttamente la schermata.

**Relazioni d'uso di altre componenti:** La classe ControlMenu utilizza i metodi messi a disposizione della classe Menu contenuta nel package view. Usa i metodi di questa classe per aggiornare l'interfaccia grafica a seconda delle azioni dell'utente, visualizzando nel caso l'interfaccia web.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalla classe Menu per tenere aggiornato il proprio stato.

**Attività svolte e dati trattati:** La classe riceve in ingresso le richieste di un utente autenticato e di conseguenza si occupa di gestire l'aggiornamento dell'interfaccia grafica.

#### 5.2.5 com.safetyGame.desktop.logic.DatiLogin

**Tipo, obiettivo e funzione del componente:** La classe DatiLogin si occupa di recuperare i dati forniti dall'utente durante l'autenticazione e di consegnarli successivamente al back-end che ne tiene traccia.

**Relazioni d'uso di altre componenti:** La classe DatiLogin comunica al back-end i dati di autenticazione immessi dagli utenti dopo che è stato controllato che questi sono stati inseriti correttamente.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalla classe Login per comunicare i dati inseriti dall'utente.

**Attività svolte e dati trattati:** La classe riceve in ingresso i dati di autenticazione di un utente e li passa al back-end per tenere traccia dello storico.

#### 5.2.6 com.safetyGame.desktop.logic.ConnBack

**Tipo, obiettivo e funzione del componente:** La classe ConnBack si occupa di gestire la connessione fra back-end e front-end desktop.

**Relazioni d'uso di altre componenti:** La classe ConnBack comunica con il back-end.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalle varie classi contenute nel package logic che hanno necessità di comunicare con il back-end.

**Attività svolte e dati trattati:** La classe permette la gestione della connessione fra front-end desktop e back-end.

## 6 Front-end Web

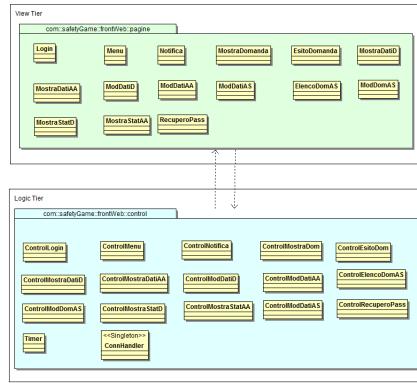


Figura 8: Front-end Web

### 6.1 View Tier

**Tipo, obiettivo e funzione del componente:** questo livello costituisce la parte del sistema che definisce ed implementa l’interfaccia web usurpuibile dagli Utenti mediante pagine web.

**Relazioni d’uso di altre componenti:** il componente è costituito dal package pagine, e comunica con il Logic Tier per rendere possibile la memorizzazione delle informazioni immesse nel sistema dagli utenti attraverso l’interfaccia web e per controllare i dati in transito per il sistema, dovuti all’interazione dell’utente con il sistema.

#### 6.1.1 com.safetyGame.frontWeb.pagine.Login

**Tipo, obiettivo e funzione del componente:** La classe Login definisce la struttura, e la conseguente visualizzazione, della pagina web che consente ad un utente di effettuare il login al sistema via web.

**Relazioni d’uso di altre componenti:** La classe Login utilizza i metodi messi a disposizione della classe ControlLogin, contenuta nel package control, per verificare i dati inseriti durante la fase di autenticazione degli utenti.

**Interfacce con e relazioni d’uso da altre componenti:** La classe viene utilizzata dalla classe ControlLogin per visualizzare uno o più errori emersi nella fase di autenticazione da parte di un utente.

**Attività svolte e dati trattati:** La classe definisce la struttura della pagina web che consente agli utenti di autenticarsi al sistema. Essa resta in attesa che un utente inserisca i dati necessari per l’autenticazione al sistema per poi passarli alla classe ControlLogin, la quale svolgerà i vari controlli.

### 6.1.2 com.safetyGame.frontWeb.pagine.Menu

**Tipo, obiettivo e funzione del componente:** La classe Menu definisce la struttura, e la conseguente visualizzazione, della pagina web che mostra ad un utente, una volta autenticato, il menu contenente le varie azioni che questo può compiere nel sistema.

**Relazioni d'uso di altre componenti:** La classe Menu utilizza i metodi messi a disposizione della classe ControlMenu, contenuta nel package control, per verificare le azioni compiute dagli utenti che interagiscono con la pagina web di menu.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalla classe ControlMenu perché viene aggiornata da questa a seconda delle azioni dell'utente. Anche la classe ControlLogin, del package control, utilizza la classe Menu perché richiama alcuni suoi metodi una volta che l'utente si è autenticato per aggiornare la vista web del sistema.

**Attività svolte e dati trattati:** La classe definisce la struttura della pagina web che consente agli utenti di navigare nel menu web del sistema. Essa resta in attesa che un utente selezioni una delle voci presenti nel menu per poi delegare la classe ControlMenu di effettuare i controlli del caso.

### 6.1.3 com.safetyGame.frontWeb.pagine.Notifica

**Tipo, obiettivo e funzione del componente:** La classe Notifica definisce la struttura, e la conseguente visualizzazione, della pagina web che segnala ad un utente autenticato, una notifica del sistema.

**Relazioni d'uso di altre componenti:** La classe Notifica utilizza i metodi messi a disposizione della classe ControlNotifica, contenuta nel package control, per verificare le azioni compiute dagli utenti che interagiscono con la pagina web che presenta una notifica segnalata dal sistema.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalla classe ControlNotifica perché viene aggiornata da questa a seconda delle azioni dell'utente.

**Attività svolte e dati trattati:** La classe definisce la struttura della pagina web che consente al sistema di mostrare notifiche agli utenti. Essa resta in attesa che un utente fornisca una risposta alla notifica per poi delegare la classe ControlNotifica di effettuare i controlli del caso.

### 6.1.4 com.safetyGame.frontWeb.pagine.MostraDomanda

**Tipo, obiettivo e funzione del componente:** La classe MostraDomanda definisce la struttura, e la conseguente visualizzazione, della pagina web che mostra ad un utente autenticato, una delle domande presenti nel sistema ed attende la sua risposta.

**Relazioni d'uso di altre componenti:** La classe MostraDomanda utilizza i metodi messi a disposizione della classe ControlMostraDom, contenuta nel package control, per verificare le azioni compiute dagli utenti che interagiscono con la pagina web che presenta una domanda posta dal sistema.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalla classe ControlMostraDom perchè viene aggiornata da questa a seconda delle azioni dell'utente e dalla classe ControlMenu.

**Attività svolte e dati trattati:** La classe definisce la struttura della pagina web che consente agli utenti di visualizzare, e successivamente rispondere, ad una domanda posta dal sistema. Essa resta in attesa che un utente fornisca la soluzione al quesito per poi delegare la classe ControlMostraDom di effettuare i controlli del caso.

#### 6.1.5 com.safetyGame.frontWeb.pagine.MostraDatiD

**Tipo, obiettivo e funzione del componente:** La classe MostraDatiD definisce la struttura, e la conseguente visualizzazione, della pagina web che mostra ad un Dipendente, i propri dati personali presenti nel sistema

**Relazioni d'uso di altre componenti:** La classe MostraDatiD utilizza i metodi messi a disposizione della classe ControlMostraDatiD, contenuta nel package control, per verificare le azioni compiute dagli utenti che interagiscono con la pagina web che presenta i dati personali di un Dipendente a seconda del tipo di utente che li richiede.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalla classe ControlMostraDatiD perchè viene aggiornata da questa a seconda delle azioni dell'utente e dalla classe ControlMenu.

**Attività svolte e dati trattati:** La classe definisce la struttura della pagina web che consente agli utenti di visualizzare i dati personali di un Dipendente registrato nel sistema.

#### 6.1.6 com.safetyGame.frontWeb.pagine.MostraDatiAA

**Tipo, obiettivo e funzione del componente:** La classe MostraDatiAA definisce la struttura, e la conseguente visualizzazione, della pagina web che mostra ad un Amministratore Azienda autenticato i dati personali di un Dipendente a scelta.

**Relazioni d'uso di altre componenti:** La classe MostraDatiAA utilizza i metodi messi a disposizione della classe ControlMostraDatiAA, contenuta nel package control, per verificare le azioni compiute dall'utente Amministratore Azienda che interagisce con la pagina web che presenta i dati personali dell'azienda.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalla classe ControlMostraDatiAA perchè viene aggiornata da questa a seconda delle azioni dell'Amministratore Azienda autenticato e dalla classe ControlMenu.

**Attività svolte e dati trattati:** La classe definisce la struttura della pagina web che consente all'Amministratore Azienda autenticato di visualizzare i dati personali di un dipendente registrato al sistema. Essa resta in attesa che un Amministratore Azienda autenticato compia un'azione sulla pagina, per poi delegare la classe ControlMostraDatiAA di effettuare i controlli del caso.

### 6.1.7 com.safetyGame.frontWeb.pagine.ModDatiD

**Tipo, obiettivo e funzione del componente:** La classe ModDatiD definisce la struttura, e la conseguente visualizzazione, della pagina web che consente ad un Dipendente autenticato di modificare alcuni dati personali dell'account, ed ad un Amministratore Azienda, quasi tutti i dati di un Dipendente a scelta.

**Relazioni d'uso di altre componenti:** La classe ModDatiD utilizza i metodi messi a disposizione della classe ControlModDatiD, contenuta nel package control, per verificare le azioni compiute da un Dipendente autenticato che modifica i propri dati personali.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalla classe ControlModDatiD perchè viene aggiornata da questa a seconda delle azioni del Dipendente e dalla classe ControlMenu.

**Attività svolte e dati trattati:** La classe definisce la struttura della pagina web che consente ad un Dipendente autenticato di modificare alcuni dati personali . Essa resta in attesa che un Dipendente autenticato o un Amministratore Azienda autenticato compia un'azione sulla pagina, per poi delegare la classe ControlModDatiD di effettuare i controlli del caso.

### 6.1.8 com.safetyGame.frontWeb.pagine.ModDatiAA

**Tipo, obiettivo e funzione del componente:** La classe ModDatiAA definisce la struttura, e la conseguente visualizzazione, della pagina web che consente ad un Amministratore Azienda autenticato di modificare alcuni dati personali dell'azienda.

**Relazioni d'uso di altre componenti:** La classe ModDatiAA utilizza i metodi messi a disposizione della classe ControlModDatiAA, contenuta nel package control, per verificare le azioni compiute da un Amministratore Azienda autenticato che modifica i dati personali dell'azienda.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalla classe ControlModDatiAA perchè viene aggiornata da questa a seconda delle azioni dell'Amministratore Azienda e dalla classe ControlMenu.

**Attività svolte e dati trattati:** La classe definisce la struttura della pagina web che consente ad un Amministratore Azienda autenticato di modificare alcuni dati personali . Essa resta in attesa che un Amministratore Azienda autenticato compia un'azione sulla pagina, per poi delegare la classe ControlModDatiAA di effettuare i controlli del caso.

### 6.1.9 com.safetyGame.frontWeb.pagine.ModDatiAS

**Tipo, obiettivo e funzione del componente:** La classe ModDatiAS definisce la struttura, e la conseguente visualizzazione, della pagina web che consente ad un Amministratore Sicurezza autenticato di modificare alcuni dati personali.

**Relazioni d'uso di altre componenti:** La classe ModDatiAS utilizza i metodi messi a disposizione della classe ControlModDatiAS, contenuta nel package

control, per verificare le azioni compiute da un Amministratore Sicurezza autenticato che modifica i dati personali.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalla classe ControlModDatiAS perchè viene aggiornata da questa a seconda delle azioni dell'Amministratore Sicurezza e dalla classe ControlMenu.

**Attività svolte e dati trattati:** La classe definisce la struttura della pagina web che consente ad un Amministratore Sicurezza autenticato di modificare alcuni dati personali . Essa resta in attesa che un Amministratore Sicurezza autenticato compia un'azione sulla pagina, per poi delegare la classe ControlModDatiAS di effettuare i controlli del caso.

#### 6.1.10 com.safetyGame.frontWeb.pagine.ElencoDomAS

**Tipo, obiettivo e funzione del componente:** La classe ElencoDomAS definisce la struttura, e la conseguente visualizzazione, della pagina web che mostra ad un Amministratore Sicurezza autenticato le domande presenti nel calderone centrale e quelle a cui possono essere sottoposti i dipendenti di una certa ditta.

**Relazioni d'uso di altre componenti:** La classe ElencoDomAS utilizza i metodi messi a disposizione della classe ControlElencoDomAS, contenuta nel package control, per verificare le azioni compiute dall'utente Amministratore Sicurezza che interagisce con la pagina web che contiene le varie domande presenti nel sistema.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalla classe ControlMostraDatiAA perchè viene aggiornata da questa a seconda delle azioni dell'Amministratore Azienda autenticato e dalla classe ControlMenu.

**Attività svolte e dati trattati:** La classe definisce la struttura della pagina web che consente all'Amministratore Azienda autenticato di visualizzare i dati personali dell'azienda o di un dipendente registrato al sistema. Essa resta in attesa che un Amministratore Azienda autenticato compia un'azione sulla pagina, per poi delegare la classe ControlMostraDatiAA di effettuare i controlli del caso.

#### 6.1.11 com.safetyGame.frontWeb.pagine.ModDomAS

**Tipo, obiettivo e funzione del componente:** La classe ModDomAS definisce la struttura, e la conseguente visualizzazione, della pagina web che consente ad un Amministratore Sicurezza autenticato di scegliere le domande a cui saranno sottoposti i dipendenti di una certa azienda fra quelle già presenti nel database generale.

**Relazioni d'uso di altre componenti:** La classe ModDomAS utilizza i metodi messi a disposizione della classe ControlModDomAS, contenuta nel package control, per verificare le azioni compiute dall'utente Amministratore Sicurezza che interagisce con la pagina web che consente di scegliere le domande specifiche per l'azienda.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene

utilizzata dalla classe ControlModDomAS perchè viene aggiornata da questa a seconda delle azioni dell'Amministratore Sicurezza autenticato e dalla classe ControlMenu.

**Attività svolte e dati trattati:** La classe definisce la struttura della pagina web che consente all'Amministratore Sicurezza autenticato di scegliere fra l'insieme generale delle domande quelle che si addicono all'azienda. Resta in attesa che un Amministratore Sicurezza autenticato compia un'azione sulla pagina, per poi delegare la classe ControlModDomAS di effettuare i controlli del caso.

#### 6.1.12 com.safetyGame.frontWeb.pagine.MostraStatD

**Tipo, obiettivo e funzione del componente:** La classe MostraStatD definisce la struttura, e la conseguente visualizzazione, della pagina web che consente ad un Dipendente autenticato di visualizzare i propri punteggi e statistiche.

**Relazioni d'uso di altre componenti:** La classe MostraStatD utilizza i metodi messi a disposizione della classe ControlMostraStatD, contenuta nel package control, per verificare le azioni compiute dai Dipendenti che interagiscono con la pagina web che mostra i punteggi e statistiche personali.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalla classe ControlMostraStatD perchè viene aggiornata da questa a seconda delle azioni dell'Dipendente autenticato e dalla classe ControlMenu.

**Attività svolte e dati trattati:** La classe definisce la struttura della pagina web che consente ad un Dipendente autenticato di visualizzare i propri punteggi e statistiche. Resta in attesa che un Dipendente autenticato compia un'azione sulla pagina, per poi delegare la classe ControlMostraStatD di effettuare i controlli del caso.

#### 6.1.13 com.safetyGame.frontWeb.pagine.MostraStatAA

**Tipo, obiettivo e funzione del componente:** La classe MostraStatAA definisce la struttura, e la conseguente visualizzazione, della pagina web che consente ad un Amministratore autenticato di visualizzare i punteggi dei dipendenti e le statistiche aziendali.

**Relazioni d'uso di altre componenti:** La classe MostraStatAA utilizza i metodi messi a disposizione della classe ControlMostraStatAA, contenuta nel package control, per verificare le azioni compiute dall'Amministratore Azienda che interagisce con la pagina web che mostra i punteggi e statistiche aziendali.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalla classe ControlMostraStatAA perchè viene aggiornata da questa a seconda delle azioni dell'Amministratore Azienda autenticato e dalla classe ControlMenu.

**Attività svolte e dati trattati:** La classe definisce la struttura della pagina web che consente ad un Amministratore Azienda autenticato di visualizzare i punteggi e statistiche aziendali. Resta in attesa che un Amministratore Azienda autenticato compia un'azione sulla pagina, per poi delegare la classe ControlMostraStatAA di effettuare i controlli del caso.

#### 6.1.14 com.safetyGame.frontWeb.pagine.RecuperoPass

**Tipo, obiettivo e funzione del componente:** La classe RecuperoPass definisce la struttura, e la conseguente visualizzazione, della pagina web che consente ad un Dipendente autenticato di visualizzare la schermata di recupero password quando smarrita.

**Relazioni d'uso di altre componenti:** La classe RecuperoPass utilizza i metodi messi a disposizione della classe ControlRecuperoPass, contenuta nel package control, per verificare le azioni compiute dal Dipendente che interagisce con la pagina web che consente di recuperare la propria password.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalla classe ControlRecuperoPass perché viene aggiornata da questa a seconda delle azioni che compie un Dipendente.

**Attività svolte e dati trattati:** La classe definisce la struttura della pagina web che consente ad un Dipendente di visualizzare la pagina di recupero password. Resta in attesa che un Dipendente compia un'azione sulla pagina, per poi delegare la classe ControlRecuperoPass di effettuare i controlli del caso.

## 6.2 Logic Tier

**Tipo, obiettivo e funzione del componente:** questo livello costituisce la parte del sistema che si occupa sostanzialmente di controllare e gestire i dati che riceve in ingresso l'applicazione dall'interazione con l'utente, attraverso le varie pagine web, e di mantenerle sempre aggiornate attuando le comunicazioni necessarie con il back-end.

**Relazioni d'uso di altre componenti:** il componente è costituito dal package control, e comunica sia con il View Tier per mantenere aggiornato lo stato dell'interfaccia web visibile agli utenti, che con il back-end per gestire le informazioni che l'utente scambia con l'applicazione.

#### 6.2.1 com.safetyGame.frontWeb.control.ControlLogin

**Tipo, obiettivo e funzione del componente:** La classe ControlLogin si occupa di controllare le credenziali che un utente inserisce al momento dell'autenticazione per poi trasmetterle al back-end. A seconda della risposta che riceve aggiorna la pagina web visibile agli utenti.

**Relazioni d'uso di altre componenti:** La classe ControlLogin utilizza i metodi messi a disposizione della classe Login, contenuta nel package pagine, per aggiornarne lo stato una volta che un utente ha provato ad autenticarsi inserendo credenziali errate. Se invece l'autenticazione va a buon fine la classe richiama metodi della classe Menu del package pagine.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalla classe Login per tenere aggiornato il proprio stato.

**Attività svolte e dati trattati:** La classe riceve in ingresso i dati necessari per l'autenticazione per poi passarli al back-end. In seguito alla risposta che ottiene aggiorna l'interfaccia web.

### 6.2.2 com.safetyGame.frontWeb.control.ControlMenu

**Tipo, obiettivo e funzione del componente:** La classe ControlMenu si occupa di controllare le azioni che un utente, una volta autenticato, compie nella schermata di menu. A seconda dell'azione che viene eseguita si occupa di aggiornare correttamente la schermata.

**Relazioni d'uso di altre componenti:** La classe ControlMenu utilizza i metodi messi a disposizione delle classi Menu, MostraDomanda, MostraDatiD, MostraDatiAA, ModDatiD, ModDatiAA, ModDatiAS, ElencoDomAS, ModDomAS, MostraStatD e MostraStatAA contenute nel package pagine. Usa i metodi di queste classi per aggiornare l'interfaccia web a seconda delle azioni dell'utente.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalla classe Menu per tenere aggiornato il proprio stato.

**Attività svolte e dati trattati:** La classe riceve in ingresso le richieste di un utente autenticato e di conseguenza si occupa di gestire l'aggiornamento dell'interfaccia web.

### 6.2.3 com.safetyGame.frontWeb.control.ControlNotifica

**Tipo, obiettivo e funzione del componente:** La classe ControlNotifica si occupa di gestire le pagine web che mostrano agli utenti le notifiche del sistema. Si occupa inoltre di catturare le risposte per poi passare queste informazioni al back-end.

**Relazioni d'uso di altre componenti:** La classe ControlNotifica utilizza i metodi messi a disposizione della classe Notifica contenuta nel package pagine. Usa i metodi di questa classe per aggiornare l'interfaccia web che mostra le notifiche agli utenti. La classe si occupa di informare il back-end sulle scelte fatte dagli utenti.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalla classe Notifica per tenere aggiornato il proprio stato e dal back-end che specifica il tipo di notifica da visualizzare.

**Attività svolte e dati trattati:** La classe ControlNotifica riceve richieste di segnalare una notifica dal back-end e gestisce successivamente l'interfaccia web che segnala una notifica restando in attesa delle azioni compite dagli utenti.

### 6.2.4 com.safetyGame.frontWeb.control.ControlMostraDom

**Tipo, obiettivo e funzione del componente:** La classe ControlMostraDom si occupa di gestire le pagine web che mostrano agli utenti le domande. Si occupa inoltre di catturare le risposte per poi passare queste informazioni al back-end.

**Relazioni d'uso di altre componenti:** La classe ControlMostraDom utilizza i metodi messi a disposizione della classe MostraDomanda contenuta nel package pagine. Usa i metodi di questa classe per aggiornare l'interfaccia web che mostra le domande agli utenti.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene

utilizzata dalla classe MostraDomanda per tenere aggiornato il proprio stato e dalla classe ControlMenu.

**Attività svolte e dati trattati:** La classe riceve in ingresso le soluzioni fornite dai vari utenti per poi passarle al back-end.

#### 6.2.5 com.safetyGame.frontWeb.control.ControlMostraDatiD

**Tipo, obiettivo e funzione del componente:** La classe ControlMostraDatiD si occupa di gestire le pagine web che mostrano ai Dipendenti autenticati i loro dati personali.

**Relazioni d'uso di altre componenti:** La classe ControlMostraDatiD utilizza i metodi messi a disposizione della classe MostraDatiD contenuta nel package pagine. Usa i metodi di questa classe per aggiornare l'interfaccia web che mostra ai Dipendenti i loro dati personali.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalla classe MostraDatiD per tenere aggiornato il proprio stato e dal back-end che informa la classe sui dati che sono presenti in memoria.

**Attività svolte e dati trattati:** La classe riceve in ingresso una richiesta di visualizzazione dati che passa al back-end, il quale fornisce i dati necessari ad aggiornare l'interfaccia web che si occupa di mostrare ai Dipendenti i loro dati personali.

#### 6.2.6 com.safetyGame.frontWeb.control.ControlMostraDatiAA

**Tipo, obiettivo e funzione del componente:** La classe ControlMostraDatiAA si occupa di gestire le pagine web che mostrano ad un Amministratore Azienda autenticato i dati personali dell'azienda.

**Relazioni d'uso di altre componenti:** La classe ControlMostraDatiAA utilizza i metodi messi a disposizione della classe MostraDatiAA contenuta nel package pagine. Usa i metodi di questa classe per aggiornare l'interfaccia web che mostra ad un Amministratore Azienda i dati personali dell'azienda.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalla classe MostraDatiAA per tenere aggiornato il proprio stato e dal back-end che informa la classe sui dati che sono presenti in memoria.

**Attività svolte e dati trattati:** La classe riceve in ingresso una richiesta di visualizzazione dati che passa al back-end, il quale fornisce i dati necessari ad aggiornare l'interfaccia web che si occupa di mostrare ad un Amministratore Azienda i dati personali dell'azienda.

#### 6.2.7 com.safetyGame.frontWeb.control.ControlModDatiD

**Tipo, obiettivo e funzione del componente:** La classe ControlModDatiD si occupa di gestire le pagine web che consentono ai Dipendenti autenticati di modificare alcuni loro dati personali.

**Relazioni d'uso di altre componenti:** La classe ControlModDatiD utilizza i metodi messi a disposizione della classe ModificaDatiD contenuta nel package

pagine. Usa i metodi di questa classe per aggiornare l'interfaccia web che consente ai Dipendenti di modificare alcuni loro dati personali. Inoltre controlla i dati che i Dipendenti modificano e li passa al back-end.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalla classe ModificaDatiD per tenere aggiornato il proprio stato.

**Attività svolte e dati trattati:** La classe riceve in ingresso una richiesta di cambio dati, e dopo averli controllati li passa al back-end, il quale ne assicura la permanenza in memoria.

#### 6.2.8 com.safetyGame.frontWeb.control.ControlModDatiAA

**Tipo, obiettivo e funzione del componente:** La classe ControlModDatiAA si occupa di gestire le pagine web che consentono ad un Amministratore Azienda autenticato di modificare alcuni dati personali dell'azienda.

**Relazioni d'uso di altre componenti:** La classe ControlModDatiAA utilizza i metodi messi a disposizione della classe ModificaDatiAA contenuta nel package pagine. Usa i metodi di questa classe per aggiornare l'interfaccia web che consente ad un Amministratore Azienda di modificare alcuni dati personali dell'azienda. Inoltre controlla i dati che un Amministratore Azienda modifica e li passa al back-end.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalla classe ModificaDatiAA per tenere aggiornato il proprio stato.

**Attività svolte e dati trattati:** La classe riceve in ingresso una richiesta di cambio dati, e dopo averli controllati li passa al back-end, il quale ne assicura la permanenza in memoria.

#### 6.2.9 com.safetyGame.frontWeb.control.ControlElencoDomAS

**Tipo, obiettivo e funzione del componente:** La classe ControlElencoDomAS si occupa di gestire le pagine web che consentono ad un Amministratore Sicurezza autenticato di visualizzare le domande presenti nel database generale.

**Relazioni d'uso di altre componenti:** La classe ControlElencoDomAS utilizza i metodi messi a disposizione della classe ElencoDomAS contenuta nel package pagine. Usa i metodi di questa classe per aggiornare l'interfaccia web che consente ad un Amministratore Sicurezza di visualizzare le domande presenti nel database generale.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalla classe ElencoDomAS per tenere aggiornato il proprio stato.

**Attività svolte e dati trattati:** La classe si occupa di mostrare ad un Amministratore Sicurezza le domande presenti nel database generale.

#### 6.2.10 com.safetyGame.frontWeb.control.ControlModDomAS

**Tipo, obiettivo e funzione del componente:** La classe ControlModDomAS si occupa di controllare e tenere aggiornate le pagine web che consentono ad un Amministratore Sicurezza autenticato di gestire l'insieme delle domande adeguate per una ben precisa azienda.

**Relazioni d'uso di altre componenti:** La classe ControlModDomAS utilizza i metodi messi a disposizione della classe ModDomAS contenuta nel package pagine. Usa i metodi di questa classe per aggiornare l'interfaccia web che consente ad un Amministratore Sicurezza di modificare l'insieme delle domande a cui possono essere sottoposti i dipendenti di una certa azienda. Inoltre controlla i dati che un'Amministratore Sicurezza inserisce prima di passarli al back-end.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalla classe ModDomAS per tenere aggiornato il proprio stato.

**Attività svolte e dati trattati:** La classe riceve in ingresso una richiesta di cambio dati, e dopo averli controllati li passa al back-end, il quale ne assicura la permanenza in memoria.

#### 6.2.11 com.safetyGame.frontWeb.control.ControlMostraStatD

**Tipo, obiettivo e funzione del componente:** La classe ControlMostraStatD si occupa di controllare e tenere aggiornate le pagine web che consentono ad un Dipendente autenticato di visualizzare i premi e le statistiche personali.

**Relazioni d'uso di altre componenti:** La classe ControlMostraStatD utilizza i metodi messi a disposizione della classe MostraStatD contenuta nel package pagine. Usa i metodi di questa classe per aggiornare l'interfaccia web che consente ad un Dipendente di visualizzare i premi e le statistiche personali .

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalla classe MostraStatD per tenere aggiornato il proprio stato.

**Attività svolte e dati trattati:** La classe consente ad un Dipendente di visualizzare i premi e le statistiche personali.

#### 6.2.12 com.safetyGame.frontWeb.control.ControlMostraStatAA

**Tipo, obiettivo e funzione del componente:** La classe ControlMostraStatAA si occupa di controllare e tenere aggiornate le pagine web che consentono ad un Amministratore Azienda autenticato di visualizzare i premi e le statistiche aziendali.

**Relazioni d'uso di altre componenti:** La classe ControlMostraStatAA utilizza i metodi messi a disposizione della classe MostraStatAA contenuta nel package pagine. Usa i metodi di questa classe per aggiornare l'interfaccia web che consente ad un Amministratore Azienda di visualizzare i premi e le statistiche aziendali .

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalla classe MostraStatAA per tenere aggiornato il proprio stato.

**Attività svolte e dati trattati:** La classe consente ad un Amministratore Azienda di visualizzare i premi e le statistiche aziendali.

#### 6.2.13 com.safetyGame.frontWeb.control.ControlModDatiAS

**Tipo, obiettivo e funzione del componente:** La classe ControlModDatiAS si occupa di gestire le pagine web che consentono ad un Amministratore Sicurezza autenticato di modificare alcuni dati personali.

**Relazioni d'uso di altre componenti:** La classe ControlModDatiAS utilizza i metodi messi a disposizione della classe ModificaDatiAS contenuta nel package pagine. Usa i metodi di questa classe per aggiornare l'interfaccia web che consente ad un Amministratore Sicurezza di modificare alcuni dati personali. Inoltre controlla i dati che un'Amministratore Sicurezza modifica e li passa al back-end.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalla classe ModificaDatiAS per tenere aggiornato il proprio stato.

**Attività svolte e dati trattati:** La classe riceve in ingresso una richiesta di cambio dati, e dopo averli controllati li passa al back-end, il quale ne assicura la permanenza in memoria.

#### 6.2.14 com.safetyGame.frontWeb.control.ControlRecuperoPass

**Tipo, obiettivo e funzione del componente:** La classe ControlRecuperoPass si occupa di gestire le pagine web che consentono ad un Dipendente autenticato, che ha smarrito la propria password, di recuperarla.

**Relazioni d'uso di altre componenti:** La classe ControlRecuperoPass utilizza i metodi messi a disposizione della classe ControlRecuperoPass contenuta nel package pagine. Usa i metodi di questa classe per aggiornare l'interfaccia web che consente ad un Dipendente di recuperare la password smarrita.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalla classe RecuperoPass per tenere aggiornato il proprio stato.

**Attività svolte e dati trattati:** La classe permette la gestione della pagina di recupero passowrd dei dipendenti.

#### 6.2.15 com.safetyGame.frontWeb.control.Timer

**Tipo, obiettivo e funzione del componente:** La classe Timer si occupa di gestire le varie temporizzazioni richieste dalle pagine web.

**Relazioni d'uso di altre componenti:** nessuna.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalla classe ControlNotifica e da ControlMostraDom.

**Attività svolte e dati trattati:** La classe permette la gestione delle temporizzazioni richieste dal sistema.

#### 6.2.16 com.safetyGame.frontWeb.control.ConnHandler

**Tipo, obiettivo e funzione del componente:** La classe ConnHandler si occupa di gestire la connessione fra back-end e front-end web.

**Relazioni d'uso di altre componenti:** La classe ConnHandler comunica con il back-end.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalle varie classi contenute nel package control che hanno necessità di comunicare con il back-end.

**Attività svolte e dati trattati:** La classe permette la gestione della connessione fra front-end web e back-end.

## 7 Front-end Mobile

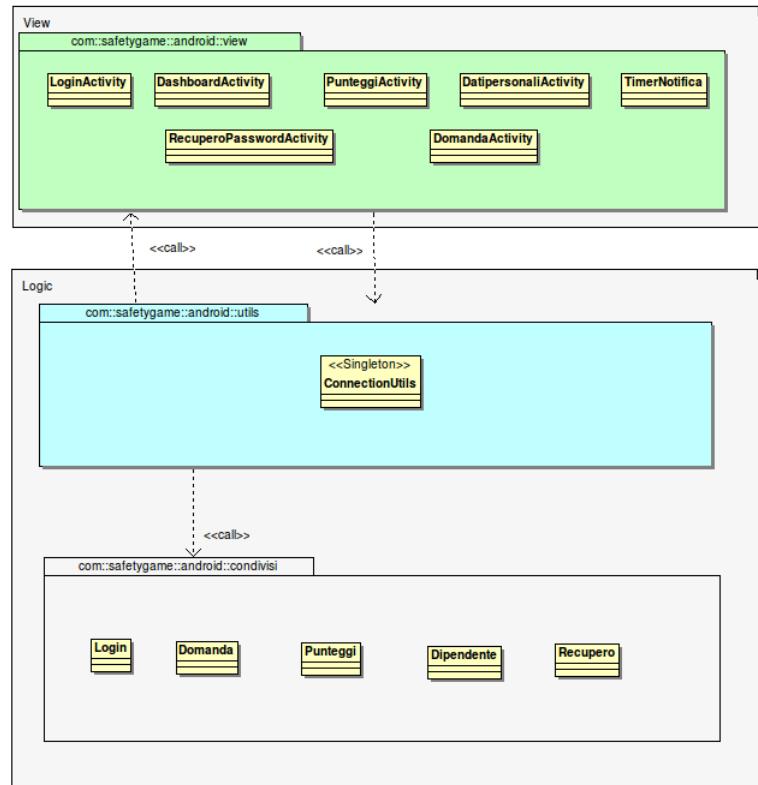


Figura 9: Front-end Mobile

### 7.1 View

**Tipo, obiettivo e funzione del componente:** Contiene tutte le classi Java che estendono la classe Java di Android “Activity” e che quindi riguardano la grafica dell’applicazione. Anche la classe TimerNotifica fa parte di questo componente anche se estende la classe Java Android “Service”.

**Relazioni d’uso di altre componenti:** Utilizza funzioni del package Utils.

**Interfacce con e relazioni d’uso da altre componenti:** View all’arrivo dei dati viene notificata da Utils.

#### 7.1.1 com.safetyGame.android.view.LoginActivity:

**Tipo, obiettivo e funzione del componente:** Activity che permette di effettuare il login ad un Dipendente.

**Relazioni d’uso di altre componenti:** La classe richiama ConnectionUtils

per inviare i dati al server, inoltre consente di arrivare alle classi DashboardActivity e RecuperoPasswordActivity; Utilizza oggetti di tipo Login per trasferire i dati.

**Interfacce con e relazioni d'uso da altre componenti:** Riceve i dati da ConnectionUtils.

**Attività svolte e dati trattati:** Raccoglie i dati inseriti dal Dipendente Autenticato e invia la richiesta di login al back-end.

#### 7.1.2 com.safetyGame.android.view.DashboardActivity:

**Tipo, obiettivo e funzione del componente:** Activity che permette di raggiungere le altre Activity del programma.

**Relazioni d'uso di altre componenti:** Può chiamare le classi PunteggiActivity, DomandaActivity e DatipersonaliActivity.

**Interfacce con e relazioni d'uso da altre componenti:** Viene chiamato da LoginActivity.

**Attività svolte e dati trattati:** Intercetta gli input e lancia altre Activity.

#### 7.1.3 com.safetyGame.android.view.PunteggiActivity:

**Tipo, obiettivo e funzione del componente:** Activity che permette ad un Dipendente Autenticato di visualizzare i vari punteggi e trofei.

**Relazioni d'uso di altre componenti:** Utilizza ConnectionUtils per richiedere dati al server, utilizza oggetti di tipo Punteggi per trasferire dati.

**Interfacce con e relazioni d'uso da altre componenti:** Viene chiamato da DashboardActivity e riceve i dati da ConnectionUtils.

**Attività svolte e dati trattati:** Visualizza i punteggi e trofei del Dipendente Autenticato.

#### 7.1.4 com.safetyGame.android.view.DatipersonaliActivity:

**Tipo, obiettivo e funzione del componente:** Activity che permette ad un Dipendente Autenticato di visualizzare e modificare i propri dati personali.

**Relazioni d'uso di altre componenti:** Utilizza ConnectionUtils per richiedere ed inviare modifiche riguardanti dati personali del Dipendente Autenticato, utilizza oggetti di tipo Dipendente per trasferire i dati.

**Interfacce con e relazioni d'uso da altre componenti:** È chiamato da DashboardActivity e riceve i dati richiesti da ConnectionUtils.

**Attività svolte e dati trattati:** Visualizza e permette di modificare i dati personali del Dipendente Autenticato.

#### 7.1.5 com.safetyGame.android.view.DomandaActivity:

**Tipo, obiettivo e funzione del componente:** Activity che permette ad un Dipendente Autenticato di visualizzare, richiedere e risponde alle domande.

**Relazioni d'uso di altre componenti:** utilizza ConnectionUtils per richiedere dati al server, utilizza oggetti di tipo Domanda per trasferire dati.

**Interfacce con e relazioni d'uso da altre componenti:** È chiamato da DashboardActivity o TimerNotifica e riceve i dati da ConnectionUtils.

**Attività svolte e dati trattati:** Permette di richiedere, visualizzare e rispondere a domande.

#### 7.1.6 com.safetyGame.android.view.RecuperoPasswordActivity:

**Tipo, obiettivo e funzione del componente:** Activity che permette di recuperare la password di un Dipendente.

**Relazioni d'uso di altre componenti:** utilizza ConnectionUtils per l'invio di dati al server.

**Interfacce con e relazioni d'uso da altre componenti:** È chiamato da LoginActivity e riceve dati da ConnectionUtils.

**Attività svolte e dati trattati:** Raccoglie i dati del Dipendente e richiede il recupero password.

#### 7.1.7 com.safetyGame.android.view.TimerNotifica:

**Tipo, obiettivo e funzione del componente:** Servizio timer che proporrà al Dipendente Autenticato una domanda secondo le specifiche impostate attraverso le notifiche standard di Android.

**Relazioni d'uso di altre componenti:** Chiama la classe DomandaActivity.

**Interfacce con e relazioni d'uso da altre componenti:** Nessuna.

**Attività svolte e dati trattati:** Periodicamente un timer a seconda delle impostazioni farà partire una notifica standard di Android che permette di visualizzare una nuova domanda.

## 7.2 Logic

**Tipo, obiettivo e funzione del componente:** Contiene le classi che comunicano con le API del server, ne ricevono i dati e notificano la View.

**Relazioni d'uso di altre componenti:** Comunica con il back-end e notifica al componente View quando i dati sono pronti.

**Interfacce con e relazioni d'uso da altre componenti:** È utilizzato dalla View.

**Attività svolte e dati trattati:** Invia delle richieste HTTP al server, il quale gli risponderà inviando i dati richiesti attraverso un XML, in seguito ne estrapolerà i dati e li renderà disponibili alla View.

#### 7.2.1 com.safetyGame.android.utils.ConnectionUtils

**Tipo, obiettivo e funzione del componente:** La classe ConnectionUtils viene utilizzata per effettuare la connessione fisica tra il front-end Mobile ed il back-end, per ricevere i dati, effettuarne il parser e creare gli opportuni oggetti da notificare alla View.

**Relazioni d'uso di altre componenti:** Richiama metodi del back-end in base alle richieste della View.

**Interfacce con e relazioni d'uso da altre componenti:** Viene richiamata dalle Activity per inviare o ricevere dati al back-end.

**Attività svolte e dati trattati:** La classe si occupa di effettuare le richieste HTTP al server e di riceverne le risposte tramite pagine XML.

#### 7.2.2 com.safetyGame.android.condivisi.Xxx

**Tipo, obiettivo e funzione del componente:** Le classi Xxx vengono utilizzate come contenitori per trasferire dati tra i vari componenti.

**Relazioni d'uso di altre componenti:** Nessuna.

**Interfacce con e relazioni d'uso da altre componenti:** Le classi vengono utilizzate da tutti gli altri package per trasferire informazioni tra i vari livelli del front-end.

**Attività svolte e dati trattati:** Ogni classe contiene metodi per leggere, inserire o modificare i dati contenuti al suo interno.

## 8 Back-end

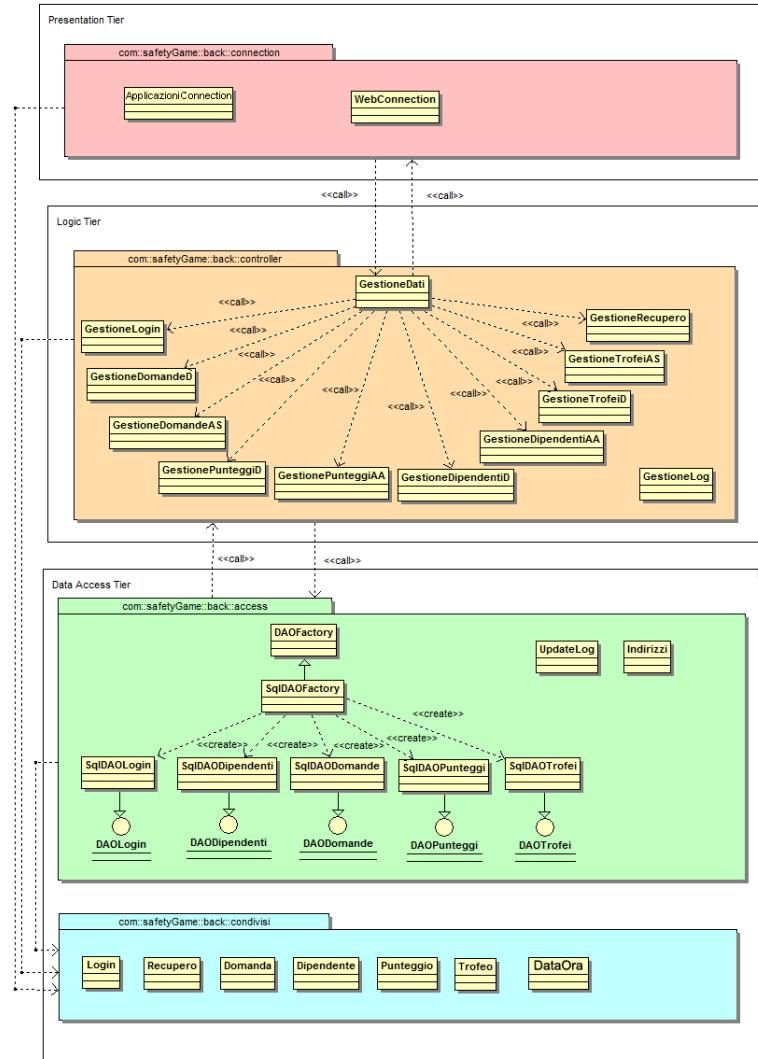


Figura 10: Back-end

### 8.1 Presentation Tier

**Tipo, obiettivo e funzione del componente:** Il Presentation Tier si occupa di gestire le connessioni con i vari Front-End, passando le richieste di lettura o inserimento dati al livello inferiore e traducendo eventualmente le risposte in xml per poi rimandarle ai relativi Front-End;

**Relazioni d'uso di altre componenti:** Il Presentation Tier richiama funzioni dal Logic Tier.

#### 8.1.1 com.safetyGame.back.connection.ApplicationConnection

**Tipo, obiettivo e funzione del componente:** La classe ApplicationConnection viene utilizzata per gestire le connessioni del Back-End con i vari Front-End Desktop e Mobile e trasferire richieste e dati tra loro ed i livelli inferiori del Back-End.

**Relazioni d'uso di altre componenti:** La classe utilizza metodi della classe Facade GestioneDati per inviare richieste di inserimento o lettura dati da database, inviandone le relative risposte alla classe richiedente del relativo Front-End Desktop e Mobile.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalle classi del livello Controller dei Front-End Desktop e Mobile.

**Attività svolte e dati trattati:** La classe riceve richieste di connessione dai vari Front-End Desktop e Mobile e si occupa di mantenerle, gestendo eventuali errori, inoltre riceve tutte le richieste di scambio dati tramite protocollo http e le invia alla classe GestioneDati del package controller, traducendo le risposte ottenute in xml e reinviandole al Front-End richiedente.

#### 8.1.2 com.safetyGame.back.connection.WebConnection

**Tipo, obiettivo e funzione del componente:** La classe WebConnection viene utilizzata per gestire le connessioni del Back-End con i vari Front-End Web e trasferire richieste e dati tra loro ed i livelli inferiori del Back-End.

**Relazioni d'uso di altre componenti:** La classe utilizza metodi della classe Facade GestioneDati per inviare richieste di inserimento o lettura dati da database, inviandone le relative risposte alla classe richiedente del relativo Front-End Web.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalle classi del livello Controller dei Front-End Web.

**Attività svolte e dati trattati:** La classe riceve tutte le richieste di scambio dati dai vari Front-End Web tramite protocollo http e le invia alla classe GestioneDati del package controller, traducendo le risposte ottenute in xml e reinviandole al Front-End richiedente, inoltre gestisce eventuali errori come l'improvvisa chiusura della pagina web durante la visione di una domanda da parte di un Dipendente.

### 8.2 Logic Tier

**Tipo, obiettivo e funzione del componente:** Il Logic Tier è deputato allo svolgimento di tutte le funzioni di sistema, fornendo metodi di richiesta ed elaborazione dati allo strato superiore. Comunica inoltre con il Data Access Tier, deputato al recupero dei dati.

**Relazioni d'uso di altre componenti:** Il Logic Tier richiama funzioni dal Data AccessTier, mentre espone funzioni al Presentation Tier.

### 8.2.1 com.safetyGame.back.controller.GestioneDati

**Tipo, obiettivo e funzione del componente:** Fa parte del Façade. Offre tutte le funzioni delle classi Gestione (tranne GestioneLog) al package connection del Presentation-tier. Quando una funzione viene chiamata, chiama la relativa funzione della classe che la contiene, non possiedendo altre istruzioni che “reindirizzamenti” delle chiamate di metodi.

**Relazioni d'uso di altre componenti:** I metodi interni alla classe chiamano i metodi delle altre classi facenti parte del package *back.controller* quando, a loro volta, verranno chiamati.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dai membri del package *back.connection* per svolgere tutte le operazioni deputate dal back-end.

**Attività svolte e dati trattati:** La sua funzione è quella di reindirizzare le chiamate dal Presentation-Tier alle classi deputate alla gestione delle richieste.

### 8.2.2 com.safetyGame.back.controller.GestioneLogin

**Tipo, obiettivo e funzione del componente:** La classe GestioneLogin viene utilizzata per fornire funzioni di login agli applicativi front-end.

**Relazioni d'uso di altre componenti:** La classe utilizza metodi della classe *SqIDAOLogin* del package *back.access* ed utilizza oggetti di tipo Login del package condivisi per scambiare dati.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalle classi ClientBackEnd dei vari front-end.

**Attività svolte e dati trattati:** La classe riceve richieste di autenticazione tramite la ricezione di oggetti Login e le invia a SqIDAOLogin per verificarne la consistenza, notificando alla classe GestioneDati il successo o fallimento dell'operazione.

### 8.2.3 com.safetyGame.back.controller.ControllerDomanda

**Tipo, obiettivo e funzione del componente:** La classe ControllerDomanda viene utilizzata per gestire richieste di visualizzazione di domande e di invio risposte da parte dei componenti front-end.

**Relazioni d'uso di altre componenti:** La classe utilizza metodi della classe CheckRispostaDomanda del package checker ed utilizza oggetti di tipo Domanda del package condivisi per scambiare dati.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalle classi ClientBackEnd dei vari front-end e provvede metodi a CheckRispostaDomanda per inviare i dati ottenuti ai relativi front-end.

**Attività svolte e dati trattati:** La classe riceve richieste di visualizzazione o risposta di una domanda da parte di un ClientBackEnd, richiamando i metodi di

CheckRispostaDomanda per ottenere una domanda da visualizzare o verificare la risposta data.

#### 8.2.4 com.safetyGame.back.controller.ControllerCalderone

**Tipo, obiettivo e funzione del componente:** La classe ControllerCalderone viene utilizzata per gestire richieste di visualizzazione, aggiunta ed eliminazione di domande nel database locale di domande, inoltre viene utilizzata per richieste di visualizzazione di domande presenti nel database centrale di domande.

**Relazioni d'uso di altre componenti:** La classe utilizza metodi della classe CheckModificaDomanda del package checker ed utilizza oggetti di tipo Domanda del package condivisi per scambiare dati.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalla classe ClientBackEnd del front-end web e provvede metodi a CheckModificaDomanda per inviare i dati ottenuti al front-end.

**Attività svolte e dati trattati:** La classe riceve richieste di visualizzazione, aggiunta od eliminazione di una o più domande dal database locale di domande o di visualizzazione di domande dal database centrale di domande e richiama i relativi metodi di CheckModificaDomanda per effettuare le operazioni richieste.

#### 8.2.5 com.safetyGame.back.controller.ControllerDipendenti

**Tipo, obiettivo e funzione del componente:** La classe ControllerDipendenti viene utilizzata per gestire richieste di visualizzazione, aggiunta, modifica ed eliminazione di account Dipendente dal database aziendale di dipendenti.

**Relazioni d'uso di altre componenti:** La classe utilizza metodi della classe CheckDipendenti del package checker ed utilizza oggetti di tipo Dipendente del package condivisi per scambiare dati.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalla classe ClientBackEnd del front-end web e provvede metodi a CheckDipendenti per inviare i dati ottenuti al front-end.

**Attività svolte e dati trattati:** La classe riceve richieste di visualizzazione, aggiunta, modifica od eliminazione di account Dipendente e richiama i relativi metodi di CheckDipendente per controllare la consistenza ed effettuare le operazioni richieste.

#### 8.2.6 com.safetyGame.back.controller.ControllerTrofei

**Tipo, obiettivo e funzione del componente:** La classe ControllerTrofei viene utilizzata per gestire richieste di visualizzazione, aggiunta, modifica ed eliminazione di Trofei dal database aziendale di trofei.

**Relazioni d'uso di altre componenti:** La classe utilizza metodi della classe CheckTrofei del package checker ed utilizza oggetti di tipo Trofeo del package condivisi per scambiare dati.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalla classe ClientBackEnd del front-end web per aggiunta, modifica

ed eliminazione di trofei, mentre viene utilizzata da tutti i front-end per la visualizzazione di trofei; Provvede metodi a CheckTrofei per inviare i dati ottenuti ai relativi front-end.

**Attività svolte e dati trattati:** La classe riceve richieste di visualizzazione, aggiunta, modifica od eliminazione di trofei e richiama i relativi metodi di CheckTrofei per controllare la consistenza ed effettuare le operazioni richieste.

#### 8.2.7 com.safetyGame.back.controller.ControllerDati

**Tipo, obiettivo e funzione del componente:** La classe ControllerDati viene utilizzata per gestire richieste di visualizzazione e modifica di dati account Dipendente dal database aziendale di dipendenti o per gestire richieste di modifica password di account Amministratore Azienda.

**Relazioni d'uso di altre componenti:** La classe utilizza metodi della classe CheckDati del package checker ed utilizza oggetti di tipo Dipendente del package condivisi per scambiare dati.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalle classi ClientBackEnd dei front-end per i dati account Dipendente e dal ClientBackEnd web per la password Amministratore Azienda; Provvede metodi a CheckDipendenti per inviare i dati ottenuti ai relativi front-end.

**Attività svolte e dati trattati:** La classe riceve richieste di visualizzazione o modifica di dati account Dipendente o di modifica password di account Amministratore Azienda e richiama i relativi metodi di CheckDati per controllare la consistenza ed effettuare le operazioni richieste.

#### 8.2.8 com.safetyGame.back.controller.ControllerRecupero

**Tipo, obiettivo e funzione del componente:** La classe ControllerRecupero viene utilizzata per gestire richieste di recupero password per account Dipendente, Amministratore Azienda ed Amministratore Sicurezza.

**Relazioni d'uso di altre componenti:** La classe utilizza metodi della classe CheckRecupero del package checker ed utilizza oggetti di tipo Recupero del package condivisi per scambiare dati.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalle classi ClientBackEnd dei front-end web e mobile, provvede metodi a CheckRecupero per inviare i dati ottenuti ai relativi front-end.

**Attività svolte e dati trattati:** La classe riceve richieste di recupero password di account Dipendente, Amministratore Azienda ed Amministratore Sicurezza e richiama i relativi metodi di CheckRecupero per controllare la consistenza ed effettuare le operazioni richieste.

#### 8.2.9 com.safetyGame.back.controller.ControllerPunteggi

**Tipo, obiettivo e funzione del componente:** La classe ControllerPunteggi viene utilizzata per gestire richieste di visualizzazione di punteggi relativi ad account Dipendente dal database aziendale di dipendenti.

**Relazioni d'uso di altre componenti:** La classe utilizza metodi della classe CheckPunteggi del package checker ed utilizza oggetti di tipo Punteggio del package condivisi per scambiare dati.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalle classi ClientBackEnd dei front-end e provvede metodi a CheckPunteggi per inviare i dati ottenuti ai front-end.

**Attività svolte e dati trattati:** La classe riceve richieste di visualizzazione di punteggi relativi ad account Dipendente e richiama i relativi metodi di CheckPunteggi per controllare la consistenza ed effettuare le operazioni richieste.

#### 8.2.10 com.safetyGame.back.controller.ControllerLog

**Tipo, obiettivo e funzione del componente:** La classe ControllerLog viene utilizzata per gestire richieste di aggiunta di Log per il tracciamento delle azioni dei Dipendenti all'interno del sistema.

**Relazioni d'uso di altre componenti:** La classe utilizza metodi della classe CheckLog del package checker ed utilizza oggetti di tipo Log del package condivisi per scambiare dati.

**Interfacce con e relazioni d'uso da altre componenti:** La classe viene utilizzata dalle classi ClientBackEnd dei front-end.

**Attività svolte e dati trattati:** La classe riceve richieste di aggiunta di Log relativi ad account Dipendente e richiama i relativi metodi di CheckLog per controllare la consistenza ed effettuare le operazioni richieste.

### 8.3 Logic Tier

**Tipo, obiettivo e funzione del componente:** Il Logic Tier si occupa di far scambiare gli oggetti contenenti dati tra il Presentation Tier ed il Data Access Tier, controllandone la conformità dei dati contenuti.

**Relazioni d'uso di altre componenti:** Il Logic Tier richiama funzioni sia dal Presentation Tier che dal Data Access Tier.

#### 8.3.1 com.safetyGame.back.checker.CheckXxx

**Tipo, obiettivo e funzione del componente:** Le classi CheckXxx vengono utilizzate per il controllo ed il trasferimento dei dati richiesti dal Presentation Tier e dal Data Access Tier.

**Relazioni d'uso di altre componenti:** Le classi CheckXxx utilizzano metodi delle classi del package access per ottenere ed inviare i dati richiesti dal package controller ed utilizzano metodi delle classi controller per inviare i dati ottenuti dal package access a chi li ha richiesti; utilizzano oggetti del package condivisi per scambiare dati con gli altri due package.

**Interfacce con e relazioni d'uso da altre componenti:** Le classi CheckXxx vengono richiamate dalle relative classi controller per ricevere le richieste degli utenti e dalle classi access per effettuare il passaggio di dati.

**Attività svolte e dati trattati:** Le classi CheckXxx forniscono metodi alle

relative classi dei package controller e access per far transitare oggetti contenenti dati dal Presentation Tier al Data Access Tier e viceversa, effettuando controlli di conformità dei dati contenuti in tali oggetti.

## 8.4 Data Access Tier

**Tipo, obiettivo e funzione del componente:** Il Data Access Tier si occupa di accedere ai vari database e di creare oggetti del package condivisi per trasferire i dati ottenuti ai livelli superiori.

**Relazioni d'uso di altre componenti:** Il Data Access Tier richiama funzioni dal Logic Tier.

### 8.4.1 com.safetyGame.back.access.AccessXxx

**Tipo, obiettivo e funzione del componente:** Le classi AccessXxx vengono utilizzate per interrogare le tabelle dei database secondo le richieste del Logic Tier.

**Relazioni d'uso di altre componenti:** Le classi utilizzano metodi del package checker per inviare dati al Presentation Tier ed usa query per interrogare i database.

**Interfacce con e relazioni d'uso da altre componenti:** Le classi vengono utilizzate dalle relative classi del package checker per cercare o modificare dati presenti nei database.

**Attività svolte e dati trattati:** Le classi ricevono richieste dalle classi del package checker riguardanti interrogazioni del database, una volta ottenuti i dati richiesti li inviano tramite metodi del package checker ai livelli più alti.

### 8.4.2 com.safetyGame.back.condivisi.Xxx

**Tipo, obiettivo e funzione del componente:** Le classi Xxx vengono utilizzate come contenitori per trasferire dati tra i vari Tier.

**Relazioni d'uso di altre componenti:** Nessuna.

**Interfacce con e relazioni d'uso da altre componenti:** Le classi vengono utilizzate da tutti gli altri package per trasferire informazioni tra i vari livelli del Back-end.

**Attività svolte e dati trattati:** Ogni classe contiene metodi per leggere, inserire o modificare i dati contenuti al suo interno.

## 9 Diagrammi di attività

In questa sezione si illustreranno diagrammi di attività relativi alle possibili interazioni di un utente con il front-end, suddividendo tali attività nei tre ambiti di Dipendente, Amministratore Sicurezza e Amministratore Azienda. Tutti e tre gli utenti possono chiedere la modifica da parte del sistema della propria password prima di accedere al sistema, il quale ne genererà una e la renderà nota solo a quell'utente; oppure possono effettuare il login all'interno del sistema e utilizzare quello che il sistema gli offre.

### 9.1 Ambito Dipendente

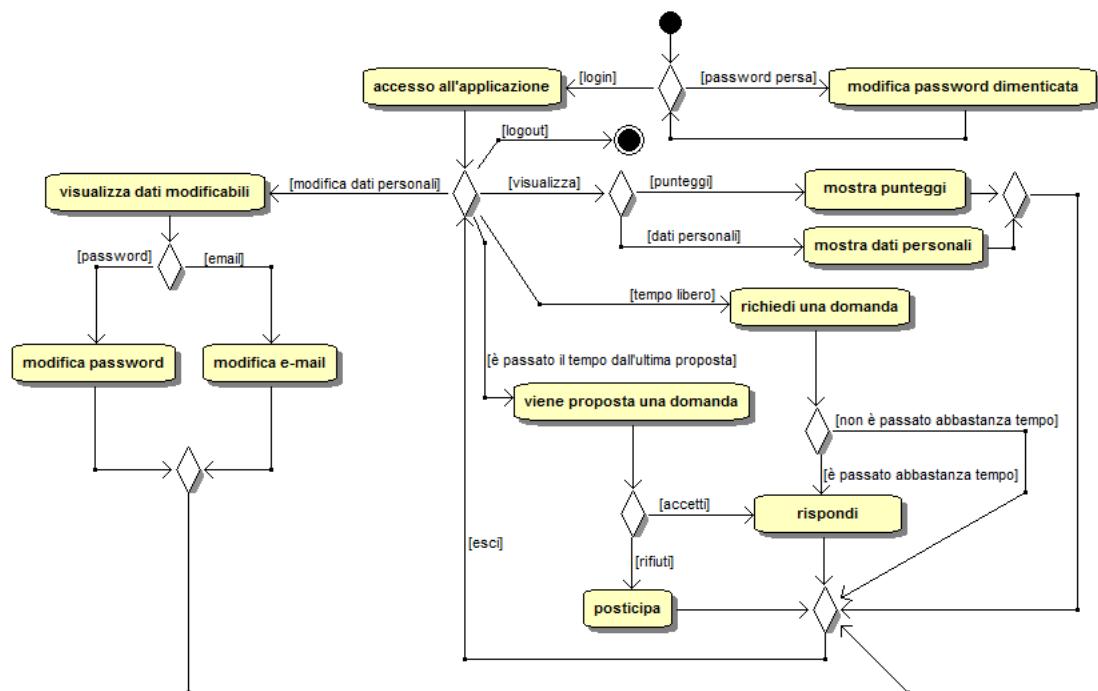


Figura 11: Ambito Dipendente

Il Dipendente, dopo che ha effettuato il login, si trova a disposizione diverse opportunità di azioni.

L'azione base che può fare è semplicemente aspettare che il sistema gli proponga di rispondere ad una domanda, alla quale può decidere se rifiutare (posticipandola) oppure accettare e quindi facendosela mostrare. L'azione di mostrare una domanda obbliga il Dipendente a rispondere alla domanda, nel caso che il Dipendente non sappia la risposta, può dare risposta nulla,

ma deve in ogni caso rispondere. L'applicazione che fa vedere questa domanda (browser o applicazione Android) impedirà la chiusura da parte dell'utente dell'applicazione.

È prevista una alternativa all'attesa di proposta domanda, ovvero la richiesta di una domanda, alla quale il sistema reagirà controllando il tempo trascorso dall'ultima domanda richiesta. Se il lasso di tempo trascorso si rivelerà essere sufficientemente grande, allora all'utente verrà mostrata la domanda alla quale si applicano le stesse regole riportate nel caso precedente. Nel caso in cui, invece, il tempo trascorso non fosse sufficiente, il sistema risponderà con un messaggio d'allerta e ritornerà alla pagina principale delle azioni.

Il Dipendente può inoltre modificare il proprio recapito e-mail e la propria password di autenticazione. Una volta che una di queste due azioni è stata effettuata, il sistema si riporterà sulla schermata principale.

L'ultimo insieme di azioni che può fare il Dipendente è la visualizzazione dei propri punteggi ottenuti durante l'utilizzo del sistema e i propri dati personali salvati in esso.

Infine può effettuare il logout, il quale consentirà al Dipendente di uscire dalla sessione autenticata, salvando ogni azione che è stata effettuata, senza però permettere di chiudere qualsiasi applicazione (nel caso si tratti di una applicazione per dispositivi fissi o dispositivi mobili).

## 9.2 Ambito Amministratore Azienda

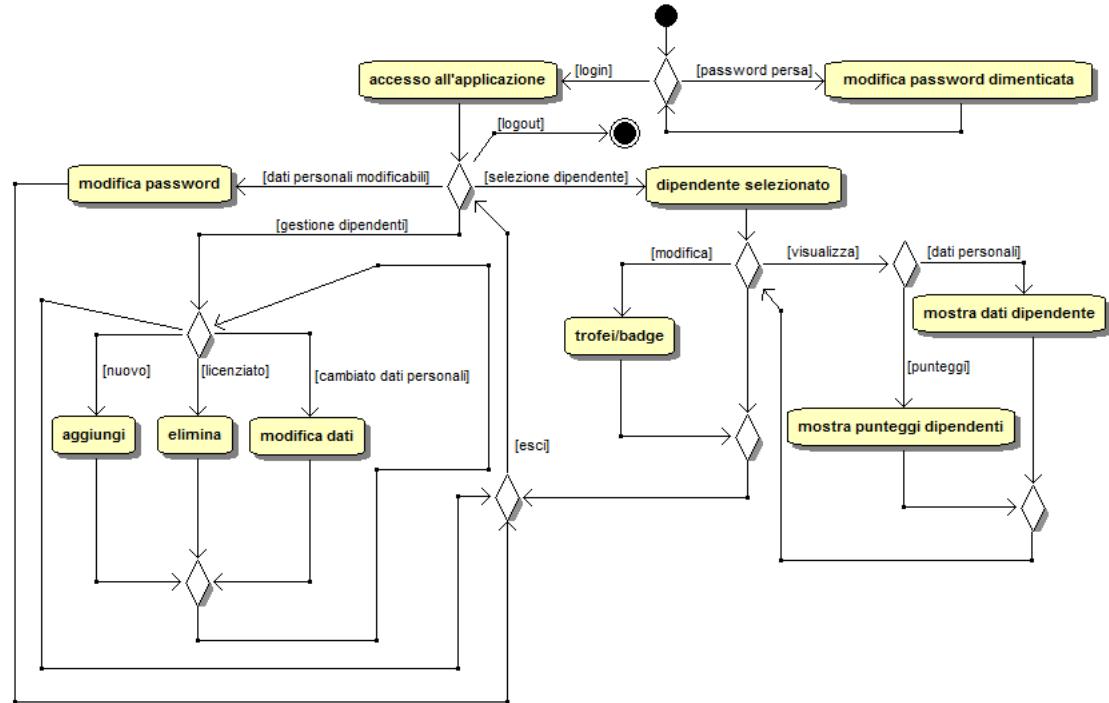


Figura 12: Ambito Amministratore Azienda

L'Amministratore Azienda, dopo che ha effettuato il login, si trova a disposizione diverse opportunità di azioni.

Come azione principale, l'Amministratore Azienda può gestire i dipendenti. La gestione dei Dipendenti si intende una scelta tra l'aggiunta di un account Dipendente al sistema, la cancellazione di un account Dipendente dal sistema (e di tutte le informazioni ad esso associate) e infine alla modifica dei dati personali del Dipendente.

- **L'aggiunta di un account Dipendente** richiede dei campi personali in modo da identificare univocamente una persona, in seguito username e password saranno associate automaticamente dal sistema e, sempre automaticamente, inviate alla e-mail specificata
- **La modifica dei dati** dell'account di un Dipendente, permette all'Amministratore Azienda di modificare, in casi eccezionali, i dati che sono stati messi nel sistema per identificare quella persona

Oltre alla modifica dei dati personali dei Dipendenti, è prevista la possibilità di modificare la propria password di autenticazione ricevendone conferma nell'indirizzo specificato.

L'ultimo insieme di azioni che può fare l'Amministratore Azienda, dopo aver selezionato un Dipendente, è la visualizzazione dei punteggi ottenuti durante l'utilizzo del sistema, i dati personali del Dipendente e infine la possibilità di aggiungere o rimuovere manualmente trofei/badge.

Infine può effettuare il logout, il quale consentirà al Amministratore Azienda di uscire dalla sessione autenticata, salvando ogni azione che è stata effettuata.

### 9.3 Ambito Amministratore Sicurezza

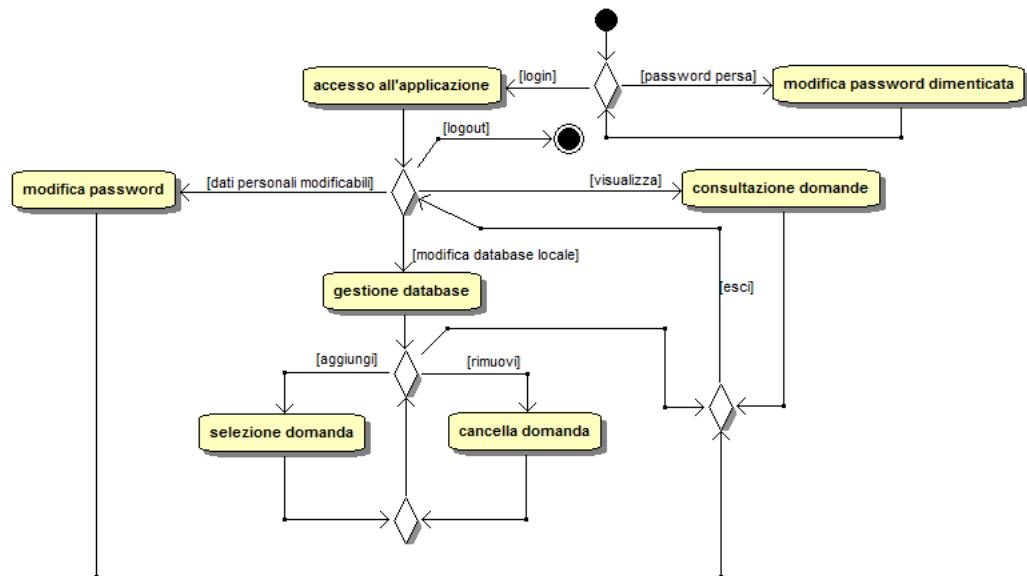


Figura 13: Ambito Amministratore Sicurezza

L'Amministratore Sicurezza, dopo che ha effettuato il login, si trova a disposizione diverse opportunità di azioni.

Come azione principale, l'Amministratore Sicurezza può gestire il database di domande interno dell'azienda. La gestione database di domande viene effettuata dal sito dell'azienda da parte di un unico Amministratore Sicurezza, il quale può:

- **aggiungere** una domanda all'insieme di domande interno (prese da un database generale esterno)
- **cancellare** una domanda, sempre dal database locale

Oltre alla modifica database di domande, è prevista la possibilità consultare l'elenco in dettaglio delle domande locali.

Può inoltre modificare la propria password personale per l'accesso a quell'azienda. Una volta modificata la password, viene inviata una notifica all'indirizzo mail dell'Amministratore Sicurezza.

Infine può effettuare il logout, il quale consentirà al Amministratore Sicurezza di uscire dalla sessione autenticata, salvando ogni azione che è stata effettuata.

#### 9.4 Considerazioni finali

Gli ambiti dell'Amministratore Azienda e Amministratore Sicurezza, verranno gestiti unicamente tramite web.

L'Ambito Dipendente invece, verrà gestito tramite web, applicazione Android e applicazione per dispositivi fissi. Tutte e tre le applicazioni permettono di eseguire ogni attività illustrata, tuttavia, l'applicazione per dispositivi fissi fornirà un comodo menu per ogni azione (una volta effettuato il login); il quale, una volta attivata la corrispondente voce nel menù, aprirà il browser e quindi sfrutterà la parte web del sistema. Invece l'applicazione Android sfrutterà le peculiarità del sistema stesso per gestire ogni singolo aspetto delle azioni intraprendibili dal Dipendente.

## 10 Tracciamento componenti - requisiti

### 10.1 Desktop

Package	Classe	Requisiti
view	Notifica	RFOB 6 RFOB 6.1 RQOB 1 RQOB 3 RQOP 3.1 RQOP 3.2
view	Login	RFOB 4 RFOB 5 RFOB 5.1 RQOB 1 RQOB 3 RQOP 3.1 RQOP 3.2
view	Menu	RFOB 7 RFOB 7.1 RFOB 10 RFOB 12 RFOB 15 RFOB 16 RQOB 1 RQOB 3 RQOP 3.1 RQOP 3.2
logic	Timer	RFOB 6 RFOB 7.1 RQOB 1 RQD 5 RVOB 1.1 RVOB 1.1.1
logic	ControlNotifica	RFOB 6 RFOB 6.1 RQOB 1 RQD 5 RVOB 1.1 RVOB 1.1.1
logic	ControlLogin	RFOB 5 RQOB 1 RQD 5 RVOB 1.1 RVOB 1.1.1

logic	ControlMenu	RFOB 7 RFOB 10 RFOB 12 RFOB 15 RFOB 16 RQOB 1 RQD 5 RVOB 1.1 RVOB 1.1.1
logic	DatiLogin	RFOB 5 RQOB 1 RQD 5 RVOB 1.1 RVOB 1.1.1
logic	ConnBack	RFOB 1 RPOB 1 RQOB 1 RQD 5 RVOB 1.1 RVOB 1.1.1

## 10.2 Web

Package	Classe	Requisiti
pagine	Login	RFOB 4 RFOB 5 RFOB 5.1 RFOB 17 RFOB 26 RQD 6 RQD 6.1 RQD 6.2 RQOB 1 RQOB 3 RQOP 3.1 RQOP 3.2
pagine	Menu	RFOB 7 RFOB 10 RFOB 12 RFOB 15 RFOB 16 RFOB 18

		RFOB 19 RFOB 20 RFOB 21 RFOB 22 RFOB 23 RFOB 24 RFOB 25 RFOB 27 RFOB 28 RFOP 29 RFOB 30 RQD 6 RQD 6.1 RQD 6.2 RQOB 1 RQOB 3 RQOP 3.1 RQOP 3.2
pagine	Notifica	RFOB 6 RFOB 6.1 RQD 6 RQD 6.1 RQD 6.2 RQOB 1 RQOB 3 RQOP 3.1 RQOP 3.2
pagine	MostraDomanda	RFOB 8 RFOB 9 RFOB 9.1 RFOB 9.1.1 RFD 9.1.2 RFOP 9.1.3 RFD 9.2 RFD 9.2.1 RQD 6 RQD 6.1 RQD 6.2 RQOB 1 RQOB 3 RQOP 3.1 RQOP 3.2
pagine	MostraDatiD	RFOB 12 RFOB 12.1 RQD 6

		RQD 6.1 RQD 6.2 RQOB 1 RQOB 3 RQOP 3.1 RQOP 3.2
pagine	MostraDatiAA	RFOB 22 RFOB 22.1 RQD 6 RQD 6.1 RQD 6.2 RQOB 1 RQOB 3 RQOP 3.1 RQOP 3.2
pagine	ModDatiID	RFOB 10 RFOB 21 RFOB 21.1 RQD 6 RQD 6.1 RQD 6.2 RQOB 1 RQOB 3 RQOP 3.1 RQOP 3.2
pagine	ModDatiAA	RFOB 11 RFOB 11.1 RQD 6 RQD 6.1 RQD 6.2 RQOB 1 RQOB 3 RQOP 3.1 RQOP 3.2
pagine	ModDatiAS	RFOB 11 RFOB 11.1 RQD 6 RQD 6.1 RQD 6.2 RQOB 1 RQOB 3 RQOP 3.1 RQOP 3.2
pagine	ElencoDomAS	RFOB 27 RFOB 27.1

		RQD 6 RQD 6.1 RQD 6.2 RQOB 1 RQOB 3 RQOP 3.1 RQOP 3.2
pagine	ModDomAS	RFOB 28 RFOB 28.1 RFOB 28.2 RQD 6 RQD 6.1 RQD 6.2 RQOB 1 RQOB 3 RQOP 3.1 RQOP 3.2
pagine	MostraStatD	RFOB 15 RFOB 15.1 RQD 6 RQD 6.1 RQD 6.2 RQOB 1 RQOB 3 RQOP 3.1 RQOP 3.2
pagine	MostraStatAA	RFOB 23 RFOB 23.1 RQD 6 RQD 6.1 RQD 6.2 RQOB 1 RQOB 3 RQOP 3.1 RQOP 3.2
pagine	RecuperoPass	RFOB 4 RQD 6 RQD 6.1 RQD 6.2 RQOB 1 RQOB 3 RQOP 3.1 RQOP 3.2
control	ControlLogin	RFOB 5 RFOB 17

		RFOB 26 RQD 5 RVOB 1 RVOB 2
control	ControlMenu	RFOB 7 RFOB 7.1 RFOB 10 RFOB 12 RFOB 12.1 RFOB 15 RFOB 15.1 RFOB 16 RFOB 18 RFOB 19 RFOB 19.1 RFOB 20 RFOB 21 RFOB 21.1 RFOB 22 RFOB 22.1 RFOB 23 RFOB 23.1 RFOB 24 RFOB 25 RFOB 27 RFOB 27.1 RFOB 28 RFOB 28.1 RFOB 28.2 RFOP 29 RFOP 29.1 RFOB 30 RQD 5 RVOB 1 RVOB 2
control	ControlNotifica	RFOB 6 RFOB 6.1 RFOB 13 RQD 5 RVOB 1 RVOB 2
control	ControlMostraDom	RFOB 8 RFOB 9 RFOB 9.1 RFOB 9.1.1

		RFD 9.1.2 RFOP 9.1.3 RFD 9.2 RFD 9.2.1 RQD 5 RVOB 1 RVOB 2
control	ControlMostraDatiD	RFOB 12 RFOB 12.1 RQD 5 RVOB 1 RVOB 2
control	ControlMostraDatiAA	RFOB 22 RFOB 22.1 RQD 5 RVOB 1 RVOB 2
control	ControlModDatiD	RFOB 10 RFOB 11 RFOB 11.1 RFOB 11.2 RFOB 21 RFOB 21.1 RQD 5 RVOB 1 RVOB 2
control	ControlModDatiAA	RFOB 4.1.1 RFOB 11 RQD 5 RVOB 1 RVOB 2
control	ControlElencoDomAS	RFOB 27 RFOB 27.1 RQD 5 RVOB 1 RVOB 2
control	ControlModDomAS	RFOB 28 RFOB 28.1 RFOB 28.2 RQOP 7 RQD 5 RVOB 1 RVOB 2
control	ControlMostraStatD	RFOB 15 RFOB 15.1

		RQD 5 RVOB 1 RVOB 2
control	ControlMostraStatAA	RFOB 23 RFOB 23.1 RQD 5 RVOB 1 RVOB 2
control	ControlModDatiAS	RFOB 4.1.1 RFOB 11 RFOB 11.1 RQD 5 RVOB 1 RVOB 2
control	ControlRecuperoPass	RFOB 4 RFOB 4.1 RFOB 4.1.1 RQD 5 RVOB 1 RVOB 2
control	Timer	RFOB 6 RFOB 7.1 RQD 5 RVOB 1 RVOB 2
control	ConnHandler	RFOB 1 RFOB 1.1 RPOB 1 RPOB 3 RQD 5 RVOB 1 RVOB 2

### 10.3 Mobile

Package	Classe	Requisiti
view	LoginActivity	RFOB 4 RFOB 5 RFOB 5.1 RQOB 1 RQOB 3 RQOP 3.1 RQOP 3.2

		RVOB 1.2 RPOB 1
view	DashBoardActivity	RFOB 7 RFOB 7.1 RFOB 10 RFOB 12 RFOB 15 RFOB 16 RQOB 1 RQOB 3 RQOP 3.1 RQOP 3.2 RVOB 1.2 RPOB 1
view	RecuperoPasswordActivity	RFOB 4 RFOB 4.1 RFOB 4.1.1 RQOB 1 RQOB 3 RQOP 3.1 RQOP 3.2 RVOB 1.2 RPOB 1
view	PunteggiActivity	RFOB 15 RFOB 15.1 RQOB 1 RQOB 3 RQOP 3.1 RQOP 3.2 RVOB 1.2 RPOB 1
view	DatiPersonaliActivity	RFOB 12 RFOB 12.1 RQOB 1 RQOB 3 RQOP 3.1 RQOP 3.2 RVOB 1.2 RPOB 1
view	DomandaActivity	RFOB 7 RFOB 7.1 RFOB 8 RFOB 9 RFOB 9.1.1 RFD 9.1.2

		RFOP 9.1.3 RFD 9.2 RFD 9.2.1 RQOB 1 RQOB 3 RQOP 3.1 RQOP 3.2 RVOB 1.2 RPOB 1
view	TimerNotifica	RFOB 6 RFOB 6.1 RQOB 1 RQOB 3 RQOP 3.1 RQOP 3.2 RVOB 1.2 RPOB 1
utils	ConnectionUtils	RFOB 1 RPOB 1 RPOB 3 RQD 5 RVOB 1
condivisi	Login	RFOB 5
condivisi	Dipendente	RFOB 10 RFOB 12 RFOB 12.1
condivisi	Domanda	RFOB 8 RFOB 9 RFOB 9.1 RFOB 9.1.1 RFD 9.1.2 RFOP 9.1.3 RFD 9.2 RFD 9.2.1 RFOB 14 RFOB 14.1
condivisi	Punteggi	RFOB 15.1
condivisi	Recupero	RFOB 4

#### 10.4 Back-end

Package	Classe	Requisiti

connection	DesktopConnection	RFOB 5 RFOB 5.1 RFOB 16 RPOB 1 RPOB 2 RQOB 1 RVOB 1 RVOB 2
connection	WebConnection	RFOB 5 RFOB 5.1 RFOB 16 RFOB 17 RFOB 25 RFOB 26 RFOB 30 RPOB 1 RPOB 2 RQOB 1 RVOB 1 RVOB 2
connection	MobileConnection	RFOB 5 RFOB 5.1 RFOB 16 RPOB 1 RPOB 2 RQOB 1 RVOB 1 RVOB 2
controller	GestioneLog	RFOB 3 RFOB 3.1
controller	GestioneLogin	RFOB 5 RFOB 5.1 RFOB 17 RFOB 26
controller	GestioneDemandeD	RFOB 6 RFOB 6.1 RFOB 7 RFOB 7.1 RFOB 8 RFOB 9 RFOB 9.1.1 RFD 9.1.2 RFOP 9.1.3 RFD 9.2

		RFD 9.2.1
controller	GestioneDomandeAS	RFOB 27 RFOB 27.1 RFOB 28 RFOB 28.1 RFOB 28.2
controller	GestionePunteggiD	RFOB 15 RFOB 15.1
controller	GestionePunteggiAA	RFOB 23 RFOB 23.1
controller	GestioneDipendentiID	RFOB 10 RFOB 11 RFOB 11.1 RFOB 11.2 RFOB 12 RFOB 12.1 RFOB 21 RFOB 21.1
controller	GestioneDipendentiAA	RFOB 11 RFOB 11.1 RFOB 22 RFOB 22.1
controller	GestioneBadgeD	RFOB 15 RFOB 15.1
controller	GestioneBadgeAA	RFOB 23 RFOB 23.1
controller	GestioneRecupero	RFOB 4 RFOB 4.1 RFOB 4.1.1
controller	GestioneDati	RFOB 4 RFOB 4.1 RFOB 4.1.1 RFOB 5 RFOB 5.1 RFOB 6 RFOB 6.1 RFOB 7 RFOB 7.1 RFOB 8 RFOB 9 RFOB 9.1.1 RFD 9.1.2 RFOP 9.1.3 RFD 9.2 RFD 9.2.1

		RFOB 10 RFOB 11 RFOB 11.1 RFOB 11.2 RFOB 12 RFOB 12.1 RFOB 15 RFOB 15.1 RFOB 17 RFOB 21 RFOB 21.1 RFOB 22 RFOB 22.1 RFOB 23 RFOB 23.1 RFOB 26 RFOB 27 RFOB 27.1 RFOB 28 RFOB 28.1 RFOB 28.2
access	DAOFactory	RPOB 2 RQOB 1 RVOB 1 RVOB 2
access	Sqldaofactory	RPOB 2 RQOB 1 RVOB 1 RVOB 2
access	SqldaoLogin	RFOB 5 RFOB 17 RFOB 26
access	SqldaoDipendenti	RFOB 11 RFOB 14 RFOB 14.1 RFOB 12 RFOB 12.1 RFOB 15.1 RFOB 21 RFOB 21.1 RFOB 22 RFOB 22.1
access	SqldaoDomande	RFOB 8 RFOB 9 RFOB 9.1.1

		RFD 9.1.2 RFOP 9.1.3 RFD 9.2 RFD 9.2.1 RFOB 14.1 RFOB 27.1 RFOB 28.1 RFOB 28.2
access	SqlDAO_Punteggi	RFOB 15.1 RFOB 23.1
access	SqlDAO_Badge	RFOB 15.1 RFOB 23.1
access	Indirizzi	RFOB 1 RFOB 1.1 RPOB 1 RPOB 2 RVOB 2
access	UpdateLog	RFOB 3 RFOB 3.1
condivisi	Login	RFOB 5 RFOB 17 RFOB 26
condivisi	Recupero	RFOB 4 RFOB 4.1 RFOB 4.1.1
condivisi	Dipendente	RFOB 11 RFOB 14 RFOB 14.1 RFOB 12 RFOB 12.1 RFOB 15.1 RFOB 21 RFOB 21.1 RFOB 22 RFOB 22.1
condivisi	Domanda	RFOB 8 RFOB 9 RFOB 9.1.1 RFD 9.1.2 RFOP 9.1.3 RFD 9.2 RFD 9.2.1 RFOB 14.1 RFOB 27.1 RFOB 28.1



		RFOB 28.2
condivisi	Punteggio	RFOB 15.1 RFOB 23.1
condivisi	Badge	RFOB 15.1 RFOB 23.1 RFOP 29 RFOP 29.1

## 11 Tracciamento requisiti - componenti

## 11.1 Ambito Dipendente

Codice	Desktop	Web	Mobile	Back-end
RFOB 1	logic.ConnBack	control.ConnHandler	utils.ConnectionUtils	access.indirizzo
RFD 2				controller.GestioneLogi access.UpdateLog
RFD 3				controller.GestioneLogi access.UpdateLog
RFOB 3.1				controller.GestioneRecuper access.UpdatePerco
RFOB 4	view.Login	page.Login	view.LoginActivity	controller.GestioneRecuper access.UpdatePerco
RFOB 4.1		page.RecuperPass	view.RecuperPassWordActivity	controller.GestioneRecuper access.UpdatePerco
RFOB 4.1.1		control.CtrRecuperPasswd	control.CtrRecuperPasswdActivity	controller.GestioneRecuper access.UpdatePerco
RFOB 5	view.Login	page.Login	view.LoginActivity	controller.GestioneRecuper access.UpdatePerco
RFOB 5.1	view.Login	page.Login	view.LoginActivity	connection.DesktopConnection connection.WebConnection connection.MobileConnection controller.GestioneData access.SqAO
RFOB 6	view.Notifica logic.TimerNotifica	page.Notifica control.ControlTimer	view.TimerNotifica	controller.GestioneData access.SqAO
RFOB 6.1	view.Notifica logic.ControlNotifica	page.Notifica control.ControlNotifica	view.TimerNotifica	controller.GestioneData access.SqAO
RFOB 7	view.Menu	page.Menu	view.Menu	controller.GestioneData access.SqAO
RFOB 7.1	view.Menu	control.ControlMenu	view.ControlMenu	controller.GestioneData access.SqAO
RFOB 8		control.ControlDomanda	view.DashboardActivity	controller.GestioneData access.SqAO
RFOB 9		control.ControlDomanda	view.DashboardActivity	controller.GestioneData access.SqAO
RFOB 9.1		control.ControlDomanda	view.DemandActivity	controller.GestioneData access.SqAO
RFOB 9.1.1		control.ControlDomanda	view.DemandActivity	controller.GestioneData access.SqAO
RFD 9.1.2		control.ControlDomanda	view.DemandActivity	controller.GestioneData access.SqAO
RFD 9.1.3		control.ControlDomanda	view.DemandActivity	controller.GestioneData access.SqAO
RFD 9.2		control.ControlDomanda	view.DemandActivity	controller.GestioneData access.SqAO
RFD 9.2.1		control.ControlDomanda	view.DemandActivity	controller.GestioneData access.SqAO
RFOB 10	view.Menu	page.Menu	view.Menu	controller.GestioneData access.SqAO
	logic.ControlMenu	page.ModDataID	control.Dipendente	controller.GestioneData access.SqAO





		view.Login view.Menu	view.DashBoardActivity view.RecuperopasswdActivity view.PanettigiaActivity view.DemandaaActivity view.TimerNotifica
RQOP 3.2	view.Login view.Menu	page.Menu page.Notifica page.ModDataID page.Recuperopass	view.DemandaaActivity view.DatIPersonalActivity view.DemandaaActivity view.TimerNotifica
RQOB 4	logic.Timer logic.ControlNotifica logic.ControlLogin logic.ControlMenu logic.DatiLogin logic.ConnBack	control.ControlLogin control.ControlNotifica control.ControlModDataID control.ControlNostrsS&ID control.ControlRecuperopass control.ConnHandler	utils.ConnectionUtils control.ControlMenu control.ControlNotifica control.ControlModDataID control.ControlNostrsS&ID control.ControlRecuperopass control.ConnHandler
RQD 5		page.Login page.Menu page.Notifica page.ModDataID page.Recuperopass	page.Menu page.Notifica page.ModDataID page.Recuperopass
RQD 6		page.Login page.Menu page.Notifica page.ModDataID page.Recuperopass	page.Menu page.Notifica page.ModDataID page.Recuperopass
RQD 6.1		page.Login page.Menu page.Notifica page.ModDataID page.Recuperopass	page.Menu page.Notifica page.ModDataID page.Recuperopass
RQD 6.2		page.Login page.Menu page.Notifica page.ModDataID page.Recuperopass	page.Menu page.Notifica page.ModDataID page.Recuperopass
RQOB 7		control.ControlLogin control.ControlNotifica control.ControlModDataID control.ControlNostrsS&ID control.ConnHandler	connection.DesktopConnection connection.WebConnection connection.MobileConnection access.SQlDAOFactory
RVOB 1.1	logic.Timer logic.ControlLogin logic.ControlMenu logic.ConnBack	logic.ControlNotifica logic.ControlLogin logic.ConnHandler logic.ConnBack	view.LoginActivity view.DashBoardActivity view.RecuperopasswdActivity view.PanettigiaActivity view.DemandaaActivity view.TimerNotifica
RVOB 1.1.1	logic.Timer logic.ConnBack	logic.ControlNotifica logic.ControlLogin logic.ConnHandler logic.ConnBack	
RVOB 1.2			view.LoginActivity view.DashBoardActivity view.RecuperopasswdActivity view.PanettigiaActivity view.DemandaaActivity view.TimerNotifica
RVOB 2	control.ConnHandler	control.ConnHandler control.ConnHandler	connection.DesktopConnection connection.WebConnection connection.MobileConnection access.SQlDAOFactory



## 111.2 Ambito Amministratore Azienda



RFOB 23.1	pagina.MostraStatistica controllo.ControlMenu	controller.GestionePunteggia controller.GestioneBadgesA controllo.GestioneDati access.SuDAOBase condivisi.Punteggio condivisi.Badge
RFOB 24	pagina.Menu controllo.MenuControlMenu	connection.WebConnection
RFOB 25	pagina.MenuControlMenu controllo.ControlHandler	connection.WebConnection
RPOB 1		connection.WebConnection
RPOB 2		connection.WebConnection access.SqLDAOFactory access.accessIndirizzi
RQOB 3	controllo.ControlHandler pagina.Login pagina.Menu pagina.ModDataA pagina.MostraStatistica pagina.Recuperopass	connection.WebConnection access.SqLDAOFactory access.accessIndirizzi
RQOB 2.1		
RQOB 2.2		
RQOB 2.3		
RQOB 2.4		
RQOB 3	pagina.Login pagina.MostraStatistica pagina.ModDataID pagina.Recuperopass	
RQOP 3.1	pagina.Login pagina.MostraStatistica pagina.ModDataID pagina.Recuperopass	
RQOP 3.2	pagina.Login pagina.MostraStatistica pagina.ModDataID pagina.Recuperopass	
RQOB 4		
RQD 5	controllo.ControlLogin controllo.ControlMenu controllo.ControlModData controllo.ControlRecuperopass controllo.ControlRecuperopass controllo.ControlRecuperopass controllo.ControlRecuperopass	
RQD 6	pagina.Menu pagina.ModDataID pagina.ModDataAA pagina.Recuperopass	
RQD 6.1	pagina.Login pagina.Menu pagina.ModDataID pagina.ModDataAA pagina.ModDataSA pagina.Recuperopass	
RQD 6.2	pagina.Login pagina.Menu pagina.MostraStatistica	

RQOP 7		pagina.ModDataA pagina.ModDataAA pagina.MostraStatAA pagina.Recuperopass	control:ControlLogin control:ControlModDataIAA control:ControlModDataAA control:ControlModDataA control:ControlRecuperoPass control:ControlHandler	connection.WebConnection access.SqIDAOFactory access.SqIDAOFactory access.Indirizzi	
RVOB 1					
RVOB 2		pagina.ModDataIAA pagina.ModDataAA pagina.ModDataA pagina.RecuperoPass	control:ControlLogin control:ControlMenu control:ControlModDataIAA control:ControlModDataAA control:ControlModDataA control:ControlRecuperoPass control:ControlHandler	connection.WebConnection access.SqIDAOFactory access.SqIDAOFactory access.Indirizzi	
RFOB 1					
RFOB 2					
RFOB 3					
RFOB 3.1					
RFOB 4		pagina.Login pagina.RecuperoPass	control:ControlRecuperoPass control:ControlModDataIAS control:ControlRecuperoPass pagina.ModDataAS pagina.ModDataIAS pagina.ModDataAS	controller.GestioneRecupero controller.GestioneRecupero controller.GestioneRecupero controller.GestioneRecupero controller.GestioneRecupero controller.GestioneRecupero	
RFOB 4.1					
RFOB 4.1.1					
RFOB 11					
RFOB 11.1					
RFOB 16		pagina.ModDataIAS	pagina.ModDataIAS	controller.GestioneDipendenteA controller.GestioneDipendenteA	
RFOB 16.1					
RFOB 26					
RFOB 27					
RFOB 27.1		pagina.Menu pagina.ElencoDomAS	pagina.Menu pagina.ElencoDomAS control:ControlElencoDomAS	controller.GestioneDomandeAS controller.GestioneDomandeAS controller.GestioneDomandeAS	
RFOB 28					
RFOB 28.1					
RFOB 28.2					

### 11.3 Ambito Amministratore Sicurezza



-RFOP 29	pagina.Menu control.ControlMenu	conndivisi.Badge
-RFOP 29.1	pagina.Menu control.ControlMenu	connection.WebConnection
-RFOB 30	pagina.Menu control.ControlMenu	connection.WebConnection
-RPOB 1	pagina.Menu control.ControlMenu	connection.WebConnection
-RPOB 2	pagina.Menu control.ControlMenu	connection.WebConnection
-RQOB 3	control.CommandHandler pagina.Login pagina.ModDataAS pagina.BlancoDomAS pagina.ModDomAS pagina.RecuperoPass	connection.WebConnection access.SqIDAFactory access.SqIDAFactory
-RQOB 2	pagina.Login pagina.ModDataAS pagina.ModDomAS pagina.RecuperoPass	connection.WebConnection access.SqIDAFactory access.SqIDAFactory
-RQOB 2.1	pagina.Login pagina.ModDataAS pagina.ModDomAS pagina.RecuperoPass	connection.WebConnection access.SqIDAFactory access.SqIDAFactory
-RQOB 2.2	pagina.Login pagina.ModDataAS pagina.ModDomAS pagina.RecuperoPass	connection.WebConnection access.SqIDAFactory access.SqIDAFactory
-RQOB 2.3	pagina.Login pagina.ModDataAS pagina.ModDomAS pagina.RecuperoPass	connection.WebConnection access.SqIDAFactory access.SqIDAFactory
-RQOB 2.4	pagina.Login pagina.ModDataAS pagina.ModDomAS pagina.RecuperoPass	connection.WebConnection access.SqIDAFactory access.SqIDAFactory
-RQOB 3	pagina.Login pagina.ModDataAS pagina.ModDomAS pagina.RecuperoPass	connection.WebConnection access.SqIDAFactory access.SqIDAFactory
-RQOP 3.1	pagina.Login pagina.ModDataAS pagina.ModDomAS pagina.RecuperoPass	connection.WebConnection access.SqIDAFactory access.SqIDAFactory
-RQOP 3.2	pagina.Login pagina.ModDataAS pagina.ModDomAS pagina.RecuperoPass	connection.WebConnection access.SqIDAFactory access.SqIDAFactory
-RQOB 4	control.ControlLogin	
-RQD 5	control.ControlMenu control.ControlModDataAS control.ControlModDomAS control.ControlRecuperoPass control.CommandHandler	
-RQD 6	pagina.Login pagina.ModDataAS pagina.ModDomAS pagina.RecuperoPass	
-RQD 6.1	pagina.Login pagina.ModDataAS pagina.ModDomAS pagina.RecuperoPass	
-RQD 6.2	pagina.Login pagina.ModDataAS pagina.ModDomAS pagina.RecuperoPass	
-RQOP 7	control.ControlModDataAS control.ControlModDomAS control.ControlRecuperoPass control.CommandHandler	connection.WebConnection access.SqIDAFactory
-RVOB 1	control.ControlLogin control.ControlMenu control.ControlModDataAS control.ControlModDomAS control.ControlRecuperoPass	connection.WebConnection access.SqIDAFactory
-RVOB 2	control.ControlLogin control.ControlMenu control.ControlModDataAS control.ControlModDomAS control.ControlRecuperoPass	connection.WebConnection access.SqIDAFactory access.SqIDAFactory



control\_CommHandler

## 12 Prototipi di interfaccia utente

In questo capitolo saranno presentate alcune delle funzionalità del prodotto tramite prototipi di interfaccia. La realizzazione di questi prototipi ha seguito il più possibile i canoni di *interfaccia intuitiva*. Questo dovrebbe garantire una semplice comprensione delle funzionalità.

### 12.1 Interfaccia desktop

Questa parte dell'applicazione è riservata ai soli Dipendenti, i quali potranno autenticarsi tramite Login o accedere alla procedura di recupero password (Figura 14).



Figura 14: schermata di Login per l'applicazione desktop

Una volta effettuato il Login l'applicazione resterà in background per una certa quantità di tempo, al termine del quale apparirà nuovamente il pop-up che inviterà a rispondere ad una nuova domanda (Figura 15).



Figura 15: schermata di Notifica di una nuova domanda

Scegliendo di rispondere subito, l'applicazione aprirà il browser consentendo di ripondere ad una nuova domanda (tramite l'applicativo web). Diversamente, rimandando la risposta la domanda verrà messa in attesa, per essere riproposta successivamente.

Sia in caso di risposta che in caso di posticipo, l'applicazione resterà in background fino alla successiva scadenza del timer.

## 12.2 Interfaccia web

A differenza dell'applicazione desktop, l'applicazione web sarà accessibile sia ai Dipendenti sia agli Amministratori. Per questo motivo il Login (Figura 16) dovrà rendere disponibili diversi servizi a seconda della tipologia di Utente collegato.

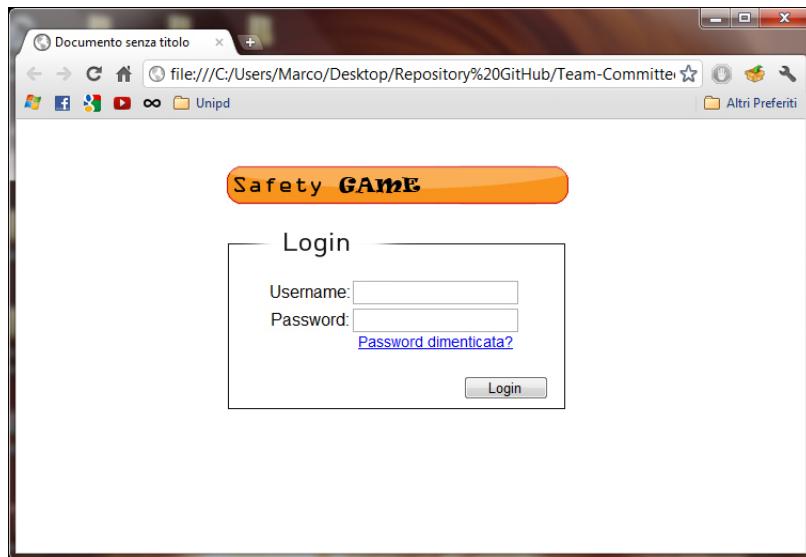
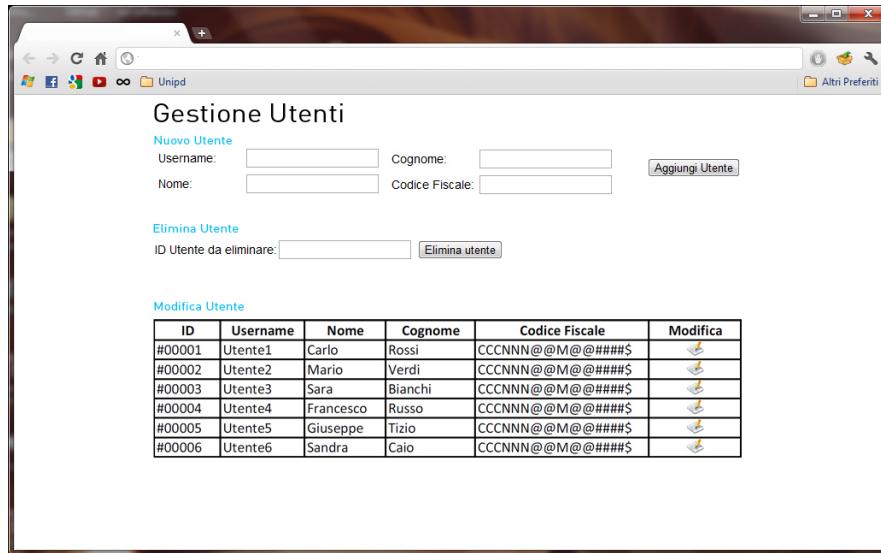


Figura 16: schermata di Login per l'applicativo web

Nel caso l'Utente sia un Dipendente, l'applicativo web permetterà di accedere ai relativi servizi, i medesimi disponibili nell'applicazione mobile.

Nel caso l'Utente sia un Amministratore, questi potrà accedere alle funzionalità di gestione.

In Figura 17 possiamo vedere come un Amministratore Azienda possa eliminare o aggiungere un Utente oppure modificarne i dati o visualizzarne le statistiche.

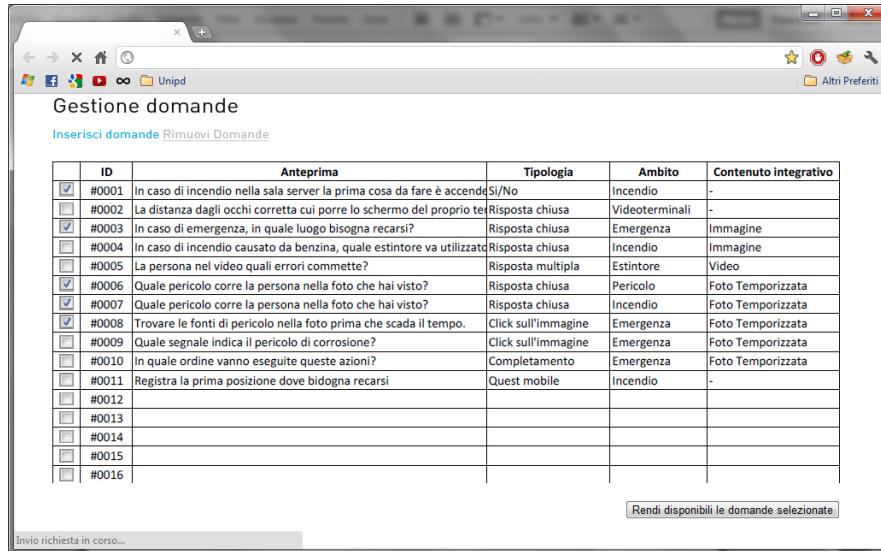


ID	Username	Nome	Cognome	Codice Fiscale	Modifica
#00001	Utente1	Carlo	Rossi	CCCNNN@#M@#####\$	
#00002	Utente2	Mario	Verdi	CCCNNN@#M@#####\$	
#00003	Utente3	Sara	Bianchi	CCCNNN@#M@#####\$	
#00004	Utente4	Francesco	Russo	CCCNNN@#M@#####\$	
#00005	Utente5	Giuseppe	Tizio	CCCNNN@#M@#####\$	
#00006	Utente6	Sandra	Caio	CCCNNN@#M@#####\$	

Figura 17: schermata per la gestione degli utenti

Inoltre, l'Amministratore Sicurezza potrà scegliere nuove domande da poter proporre, oppure rimuoverne altre già disponibili. Per fare questo vi saranno due schermate differenti.

La prima (Figura 18) permetterà di scegliere quali domande rendere disponibili tramite checkbox, per poi dare conferma tramite Button.



ID	Anteprima	Tipologia	Ambito	Contenuto integrativo
#0001	In caso di incendio nella sala server la prima cosa da fare è accende	Si/No	Incendio	-
#0002	La distanza dagli occhi corretta cui porre lo schermo del proprio te	Risposta chiusa	Videotermini	-
#0003	In caso di emergenza, in quale luogo bisogna recarsi?	Risposta chiusa	Emergenza	Immagine
#0004	In caso di incendio causato da benzina, quale estintore va utilizzat	Risposta chiusa	Incendio	Immagine
#0005	La persona nel video quali errori commette?	Risposta multipla	Estintore	Video
#0006	Quale pericolo corre la persona nella foto che hai visto?	Risposta chiusa	Pericolo	Foto Temporizzata
#0007	Quale pericolo corre la persona nella foto che hai visto?	Risposta chiusa	Incendio	Foto Temporizzata
#0008	Trovare le fonti di pericolo nella foto prima che scada il tempo.	Click sull'immagine	Emergenza	Foto Temporizzata
#0009	Quale segnale indica il pericolo di corrosione?	Click sull'immagine	Emergenza	Foto Temporizzata
#0010	In quale ordine vanno eseguite queste azioni?	Completamento	Emergenza	Foto Temporizzata
#0011	Registra la prima posizione dove bidogna recarsi	Quest mobile	Incendio	-
#0012				
#0013				
#0014				
#0015				
#0016				

Figura 18: schermata per l'inserimento di nuove domande

La seconda schermata (Figura 19) permetterà di scegliere quali domande, fra quelle già disponibili, rimuovere dal proprio set, sempre selezionando tramite Checkbox e dando conferma con Button.

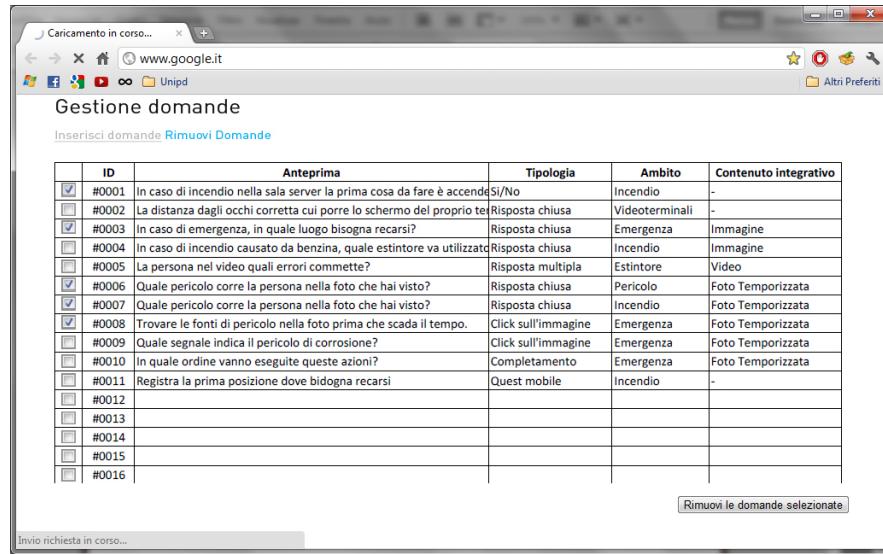


Figura 19: schermata per la rimozione di domande già presenti

### 12.3 Interfaccia Mobile

L’interfaccia dell’applicazione mobile sarà strutturata indicativamente come segue.

- Login e Notifica (Figura 20):

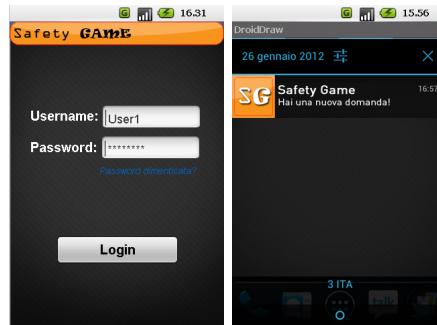


Figura 20: Schermate di Login e di Notifica per l’applicazione mobile

La schermata di Login permette ai Dipendenti di effettuare l’accesso. Un Amministratore non potrà effettuare l’accesso tramite l’applicazione mobile.

Come per gli altri Login, sarà possibile avviare la procedura per il recupero password.

Per le notifiche si utilizzerà il sistema di notifiche previsto dall'ambiente Android.

- **Rispondere ad una domanda** (Figura 21):

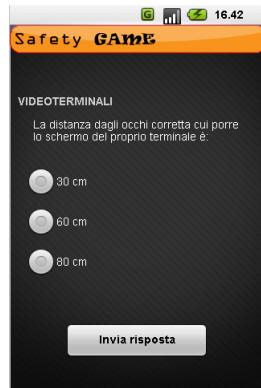


Figura 21: Risposta ad una domanda

La schermata per fornire una risposta non sarà standard, ma varierà a seconda della tipologia di domanda. Ad ogni modo, saranno disponibili degli elementi di scelta ed un bottone per l'invio della risposta.

- Visualizzazione dati Utente e modifica password (Figura 20):



Figura 22: schermata per visualizzare i propri dati e per cambiare la password.

Ogni utente avrà la possibilità di visualizzare i propri dati e di modificare la password tramite apposito form. Il tasto per confermare il cambio password sarà posizionato in modo da ridurre al minimo il rischio di invio involontario.

- Visualizzazione Punteggi e Badge (Figura 20):



Figura 23: schermata per la visualizzazione di punteggi e Badge

In questa schermata l'Utente avrà la possibilità di visualizzare i propri punteggi e i premi (o Badge) ricevuti.

## 13 Stime di fattibilità e di bisogno di risorse

L'analisi accurata dell'architettura progettata ha permesso di constatare che le tecnologie che si è scelto d'adottare risultano essere sufficientemente adeguate per la realizzazione del prodotto e riescono a ricoprire le esigenze progettuali. Nonostante non tutti gli strumenti scelti siano conosciuti dalla totalità del gruppo, la scelta di utilizzare linguaggi simili tra loro come JSP - Java - Android SDK ha permesso di facilitare l'apprendimento e migliorare l'integrità dei vari componenti di cui è costituito il prodotto.

Il reperimento delle risorse da impiegare nella realizzazione non si è rivelato problematico o complesso in quanto si è scelto di impiegare strumenti completamente gratuiti. Gli strumenti impiegati sono, come già citato, JSP per la generazione delle pagine dinamiche dell'applicazione web, Java per la realizzazione dell'applicazione Desktop ed Android SDK per la realizzazione dell'applicazione mobile; o ancora l'IDE Eclipse o BOUML.