



System Design Document

KeyMaster

Riferimento	
Versione	2.0
Data	27/11/2021
Destinatario	Prof.ssa F. Ferrucci
Presentato da	Andrea Santaniello, Mattia d'Argenio, Daniele Dello Russo, Michele Corcione
Approvato da	



Revision History

Data	Versione	Descrizione	Autori
20/11/2021	0.1	Prima stesura	Tutti
7/02/2022	0.2	Revisione	Tutti



Sommario

Revision History.....	2
1. Introduzione	4
1.1. Scopo del sistema	4
1.2. Design Goals & Trade-offs	4
1.2.1. Tempo di rilascio vs Funzionalità	5
1.2.2. Prestazioni vs Costi.....	6
1.2.3. Prestazioni vs Affidabilità.....	6
1.3. Definizioni, acronimi e abbreviazione.....	6
1.4. Riferimenti	6
1.5. Panoramica	6
2. Architettura di Sistemi simili	7
3. Architettura del Sistema proposto	7
3.1. Panoramica	7
3.2. Decomposizione in sottosistemi	8
3.2.1. Diagramma di deployment.....	10
3.3. Mapping hardware/software.....	10
3.4. Gestione dati persistenti	11
3.5. Controllo degli accessi e sicurezza.....	16
3.6. Controllo flusso globale sistema.....	17
3.7. Condizione limite.....	18
3.7.1. Start-up.....	18
3.7.2. Terminazione.....	19
3.7.3. Fallimento.....	20
4. Servizi dei Sottosistemi	20
5. Glossario	23



1. Introduzione

1.1. Scopo del sistema

L'obiettivo di KeyMaster è quello di fornire un sistema semplice di licence management (License-as-aService, LaaS) snello e facilmente implementabile in prodotti desktop e mobile.

Il sistema permette l'iscrizione di utenti che vogliono acquisire oppure creare una licenza per un software.

Le componenti principali del sistema si basano sulla creazione di applicazioni software e la gestione della licenza da parte degli utenti, che mediante un API autentifica il reale possesso di una licenza per un determinato software.

1.2. Design Goals & Trade-offs

Nelle seguenti tabelle vengono illustrati, divisi per categoria, gli obiettivi di design per il sistema.

Priorità	ID	Descrizione	Categoria	Origine
Media	DG_1	Il tempo di risposta della piattaforma dovrà essere al massimo 750 millisecondi in condizioni di rete ottimale.	Latenza	RNF_3.3.3
Alta	DG_2	Il Sistema dovrà gestire eventuali condizioni d'errore sia che esse siano state generate da input invalidi o da errori interni.	Robustezza	RNF_3.3.1
Alta	DG_3	Il sistema, in qualsiasi condizione, restituisce all'utente l'output da lui atteso.	Affidabilità	RNF_3.3.1
Alta	DG_4	Una volta realizzato, il Sistema sarà in funzione 24/24h su 7/7g	Disponibilità	RNF_3.3.3
Alta	DG_5	Al Sistema si accederà solo mediante credenziali	Sicurezza	RNF_3.3.2



		Il Database inoltre è accessibile ai soli gestori. Non sarà quindi possibile per utenti non autorizzati accedere a funzionalità protette.		RNF_3.3.8
Alta	DG_6	Lo sviluppo del prodotto richiederà costi ridotti sia in termini di risorse umane (per cui è fissato un tetto di 75 ore-lavoro), sia in termini economici (per cui si punta a ricorrere a soluzioni off-the-shelf open source)	Costi di Sviluppo	Top management
Media	DG_7	In quanto l'interazione con il Sistema avviene mediante Browser, c'è una totale indipendenza da Sistema Operativo.	Portabilità	RNF_3.3.6 RNF_3.3.7
Media	DG_8	Il Sistema sarà molto intuitivo, e le sue funzioni saranno semplici da apprendere senza la consultazione di alcuna documentazione.	Usabilità	RNF_3.3.1

1.2.1. Tempo di rilascio vs Funzionalità

Preferiamo consegnare un prodotto che permetta ai suoi utilizzatori di usufruire di tutti i suoi servizi, per tanto verranno implementate tutte le funzionalità del sistema ad alta priorità e potranno essere aggiunte altre funzionalità anche dopo il primo rilascio.



1.2.2. *Prestazioni vs Costi*

Tenuto conto del budget stanziato, e dalla necessità di avere funzioni chiave perfettamente operative nei tempi prestabiliti, si preferisce dedicare il monte ore a disposizione all'implementazione e revisione di quest'ultime.

1.2.3. *Prestazioni vs Affidabilità*

I dati gestiti dal sistema sono piuttosto sensibili, pertanto preferiamo garantire un maggior controllo di input e consistenza a scapito dei tempi di risposta.

1.3. Definizioni, acronimi e abbreviazione

Termine	Definizione
Sviluppatore	Colui che si occupa di gestire le licenze e i software
Utente	Colui che interagisce con il sistema

Acronimo	Definizione
OTS	Off The Shelf

1.4. Riferimenti

Di seguito una lista di riferimenti ad altri documenti utili durante la lettura:

- Statement Of Work;
- KM_RAD;
- Test Plan;
- Test Case Specification;

1.5. Panoramica

Al primo punto si descrive il sistema tramite i suoi obiettivi principali, gli obiettivi di design ed un elenco di definizioni che aiuteranno nella lettura di questo documento. Al secondo punto è presentato il sistema corrente. Al terzo punto è presentata l'architettura del sistema proposto. Qui gestiamo la decomposizione in sottosistemi, il mapping hardware/software, i dati persistenti, il controllo degli accessi, il controllo del



sistema globale e le condizioni limite. Al quarto punto sono presentati i servizi del sottosistema. Al quinto punto vengono raggruppati tutti i termini inerenti al sistema per una corretta comprensione del documento.

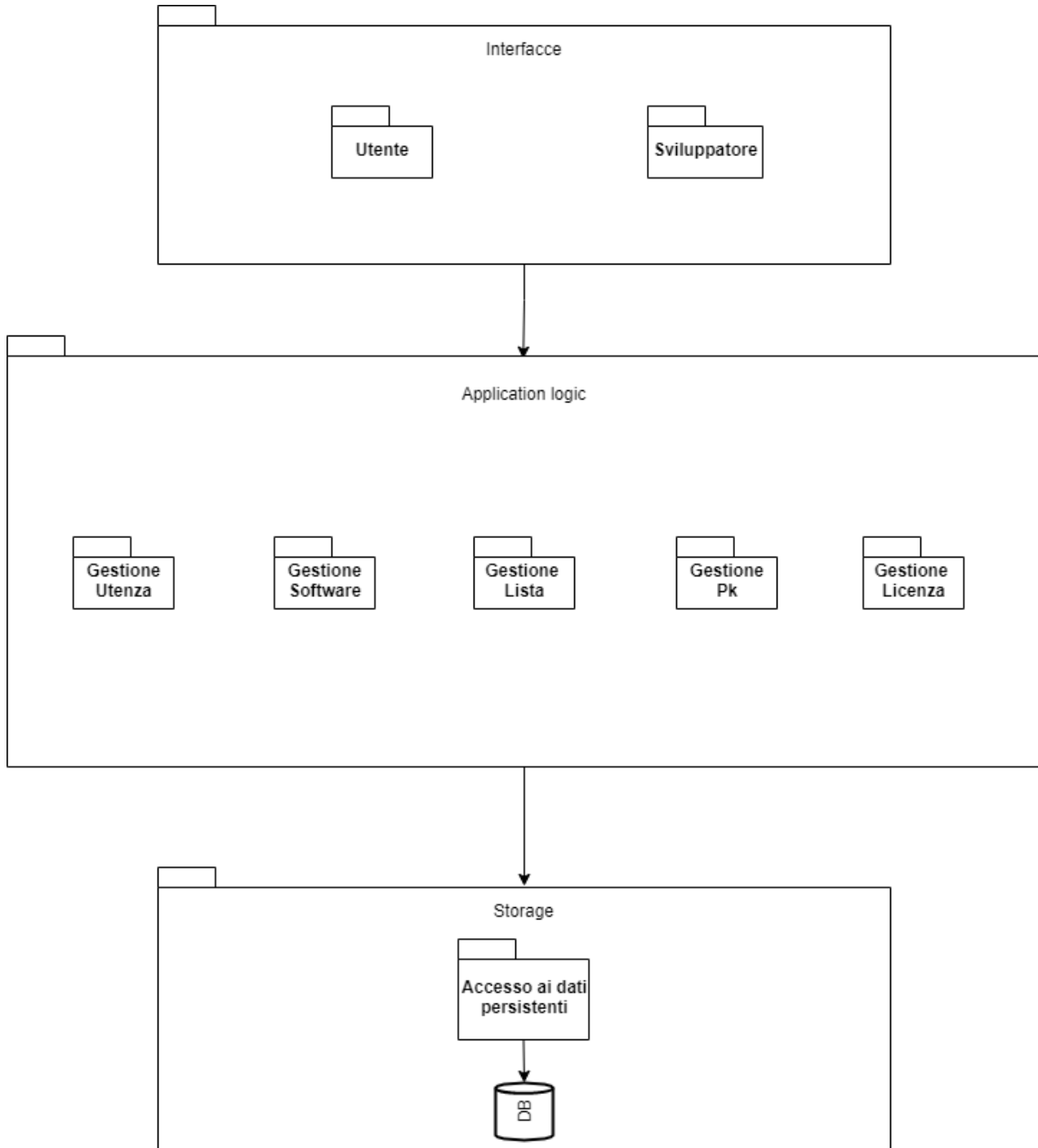
2. Architettura del sistema corrente

3. Architettura del Sistema proposto

3.1. Panoramica

KeyMaster sarà un'applicazione usufruibile tramite web. Il sistema sarà distribuito su due principali componenti hardware: un server per la gestione dei dati persistenti ed un web service che metterà a disposizione l'applicazione per gli utenti finali. Il sistema sarà diviso in vari sottosistemi: "Gui Utente", "Gui Sviluppatore" per quanto riguarda l'interfaccia utente, "Gestione Utenza", "Gestione Software", "Gestione Lista", "Gestione Pk" e "Gestione Licenza" per quanto riguarda il livello di Application Logic, "Accesso ai dati persistenti" per quanto riguarda il livello di storage.

3.2. Decomposizione in sottosistemi



Il sistema è basato su un'architettura three-tier. Questa decisione è dovuta al bisogno di mantenere un forte disaccoppiamento tra l'interfaccia grafica e la logica di business. Grazie a questa divisione è infatti possibile immaginare di sviluppare, in futuro, un'applicazione che permetterà un più semplice utilizzo del sistema il quale, per natura, necessita di essere accessibile dagli utenti in qualsiasi momento.



Il sistema si compone di tre layers:

- Il **layer di interfacce** che si occupa di presentare le informazioni ai diversi utenti e consente agli stessi di interagire con il sistema in modo semplice ed intuitivo.
- Il **layer di application-logic** che definisce la logica applicativa.
- Il **layer di storage** che si occupa della gestione dei dati persistenti.

Il layer di presentazione presenta due sottosistemi:

- **Gui Utente:** rappresenta l'insieme delle pagine web con le quali l'utente interagisce per poter usufruire dei servizi del sistema.
- **Gui Sviluppatore:** rappresenta l'insieme delle pagine web con le quali un utente sviluppatore interagisce per poter effettuare operazioni di gestione dei software;

Il layer di business logic è diviso in cinque sottosistemi:

- **Gestione utenza:** fornisce le funzionalità per effettuare l'autenticazione, gestire la propria area utente e richiedere il controllo tramite Api;
- **Gestione software:** fornisce le funzionalità per gestire tutta la parte di personalizzazione software e modifica delle informazioni relative ad esso;
- **Gestione lista software:** fornisce le funzionalità per modificare una lista di software quindi l'aggiunta di un nuovo software o la rimozione di un software già presente nel database.
- **Gestione pk:** fornisce le funzionalità per generare un codice prodotto per l'attivazione di una nuova licenza;
- **Gestione licenza:** fornisce le funzionalità per l'attivazione di nuove licenze e la modifica delle stesse.

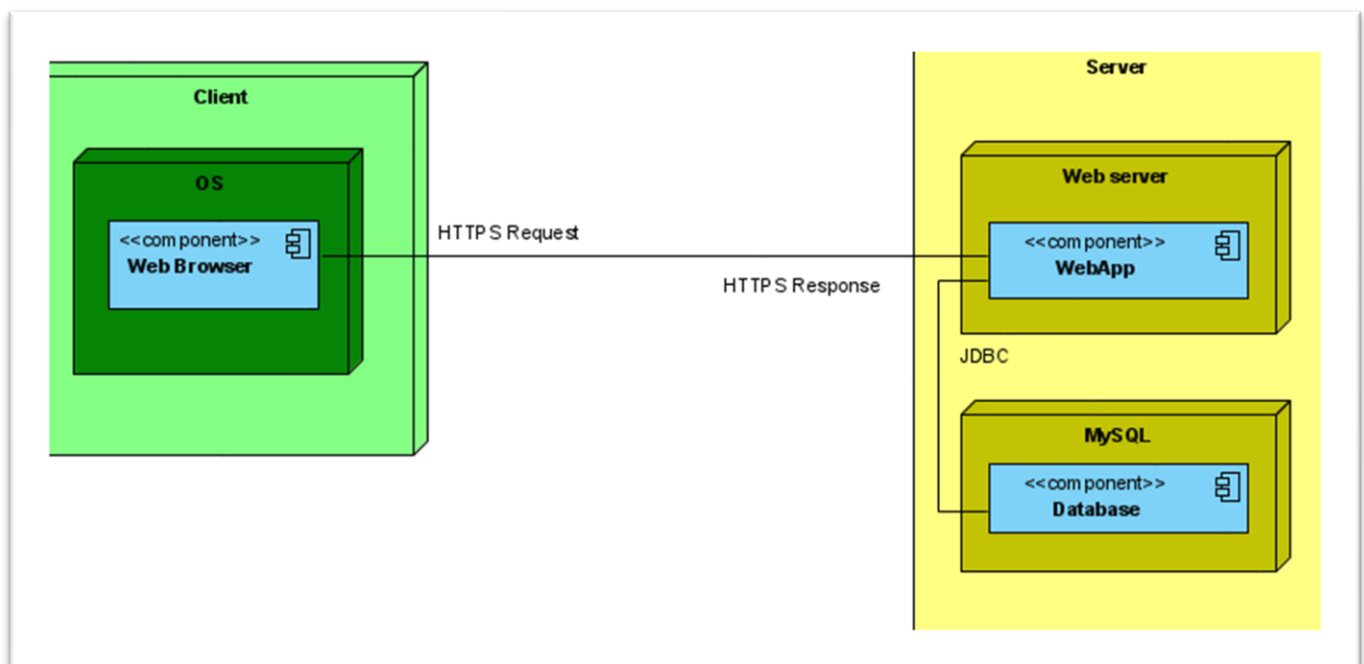
Il layer di persistenza è costituito da un unico sottosistema:

- **Accesso ai dati persistenti:** si occupa di leggere e modificare i dati memorizzati sul Database

3.2.1. Diagramma di deployment

KeyMaster è un sistema distribuito che prevede l'interazione di una web-application con un Database. Data la grande mole di dati contenuta nel Database si prevede che esso risiederà su un server pronto a ricevere e gestire richieste da diversi client, anche contemporaneamente.

In questa versione si prevede che gli utenti del sistema accedano ad esso tramite browser installati sui loro dispositivi personali.



3.3. Mapping hardware/software

KeyMaster ha due componenti hardware principali:

- Server per l'applicazione web a cui l'utente si collegherà.



- Server per il database utilizzato dal sistema per la persistenza dei dati

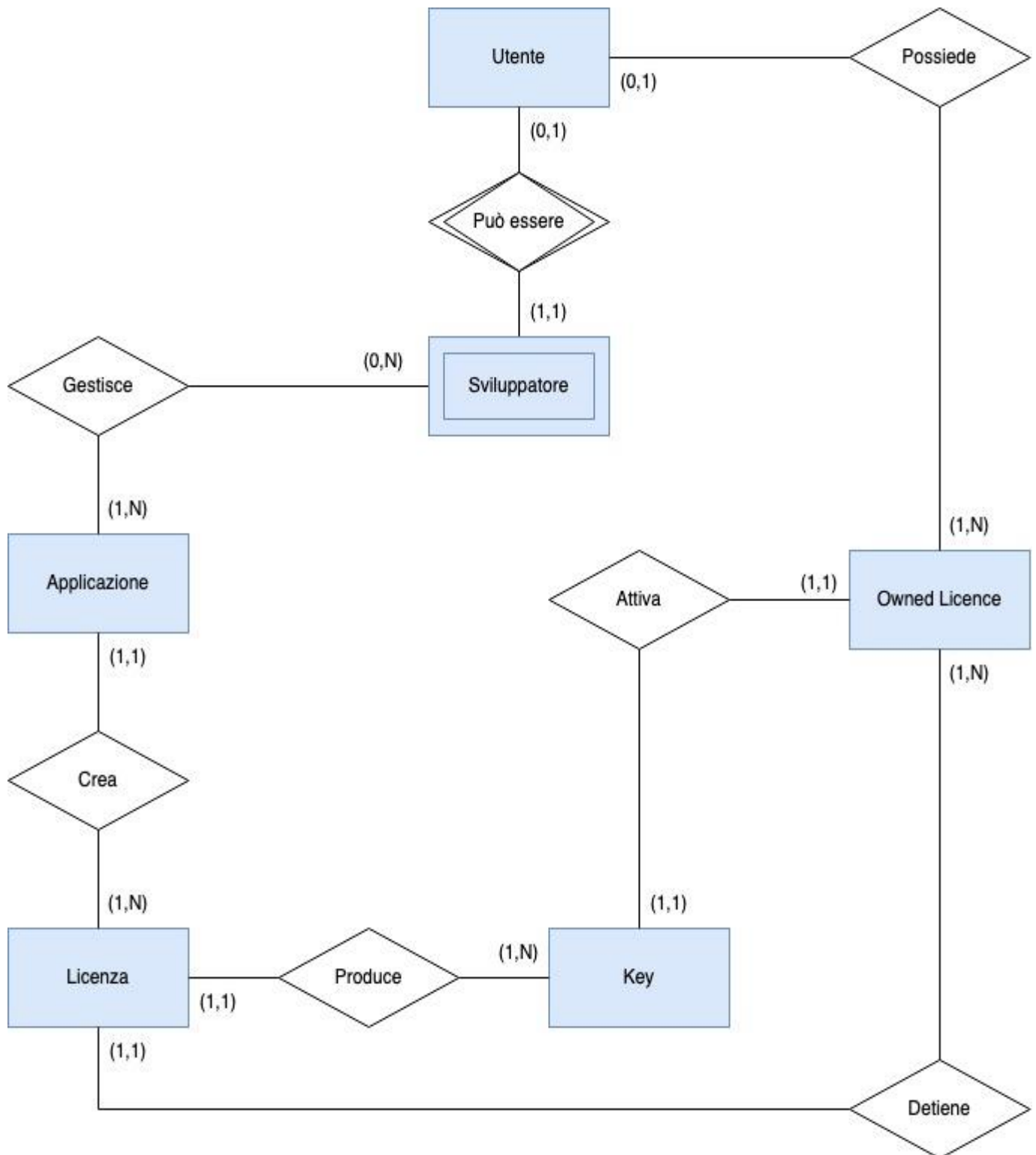
Il sistema necessita di una macchina in grado di supportare Apache Tomcat, al fine di garantire l'operabilità della WebApp, e MySQL, per garantire invece l'operabilità del database con cui la WebApp si interfaccia.

WebApp e DBMS saranno quindi installati sullo stesso nodo in modo da ridurre i possibili fallimenti o ritardi di propagazione delle informazioni dovuti a problemi di connettività.

3.4. Gestione dati persistenti

Per la gestione dei dati persistenti, KeyMaster si affida ad un database relazionale gestito tramite MySQL. La struttura dei dati memorizzati segue il seguente schema :

Schema Entità/Relazione del database





Dizionario dei dati.

Di seguito si mostrano gli attributi per ogni entità individuata

Nome Entità	Utente		
Descrizione	Contiene i dati relativi ad un utente registrato		
Nome Campo	Tipo	Vincolo di chiave	Altri vincoli
ID	INT	PRIMARY KEY	NOT NULL
Username	VARCHAR(32)		NOT NULL
Password	VARCHAR(64)		NOT NULL
email	VARCHAR(64)		NOT NULL
Firstname	VARCHAR(64)		
Lasname	VARCHAR(64)		
Hardwareid	VARCHAR(128)		
isDeveloper	TINYINT(1)		



Nome Entità	Applicazione		
Descrizione	I dati contenuti nelle applicazioni		
Nome Campo	Tipo	Vincolo di Chiave	Altri vincoli
ID	INT (11)	PRIMARY KEY	NOT NULL
Owner ID	INT (11)	FOREIGN KEY	NOT NULL
Nome	VARCHAR(32)		NOT NULL
Descrizione	TEXT		NOT NULL
Website	VARCHAR(100)		NOT NULL
Versione	INT (11)		NOT NULL
Apikey	VARCHAR(128)		NOT NULL

Nome Entità	Licenza		
Descrizione	I dati relative alle licenze		
Nome Campo	Tipo	Vincoli di Chiave	Altri vincoli
ID	INT(11)	PRIMARY KEY	NOT NULL
App Id	INT(11)	FOREIGN KEY	NOT NULL
Nome	VARCHAR(64)		NOT NULL
Descrizione	TEXT		NOT NULL
Durata	INT(11)		NOT NULL
Tipo	INT(11)		NOT NULL



Nome Entità	Key		
Descrizione	Contiene le informazioni delle chiavi generate		
Nome Campo	Tipo	Vincoli di Chiave	Altri vincoli
ID	INT(11)	PRIMARY KEY	NOT NULL
Licenza ID	INT(11)	FOREIGN KEY	NOT NULL
Licenza Key	VARCHAR(128)		NOT NULL
Riscattata	TINYINT(1)		

Nome Entità	Licenza Owned		
Descrizione	I dati relativi alle licenze possedute dall'utente		
Nome Campo	Tipo	Vincoli di Chiave	Altri vincoli
ID	INT(11)	PRIMARY KEY	NOT NULL
User ID	INT(11)	FOREIGN KEY	NOT NULL
Licenza ID	INT(11)	FOREIGN KEY	NOT NULL
Key ID	INT(11)	FOREIGN KEY	NOT NULL
Epoch Attivazione	GIBINT(32)		NOT NULL



3.5. Controllo degli accessi e sicurezza

Il controllo degli accessi è garantito tramite l'autenticazione e l'utilizzo di attributi per specificare il livello di permessi dell'attore.

Nella prima versione si utilizzerà SSL su connessione HTTP, tuttavia con autocertificato locale.

Le operazioni che gli utenti possono effettuare sono riportate nella matrice degli accessi.



Oggetto Attore	Gestione software	Gestione lista	Gestione pk	Gestione utenza	Gestione licenza
Ospite				<ul style="list-style-type: none"> Login 	
Utente	<ul style="list-style-type: none"> Modificare info software 	<ul style="list-style-type: none"> visualizza lista 	<ul style="list-style-type: none"> Visualizza menu Aggiunta, modifica ed eliminazione valutazione 	<ul style="list-style-type: none"> Logout Modifica area utente Controllo via API 	<ul style="list-style-type: none"> Attivazione software tramite pk
Sviluppatore	<ul style="list-style-type: none"> Modificare info software 	<ul style="list-style-type: none"> aggiunta ed eliminazione software 	<ul style="list-style-type: none"> Generazione pk 	<ul style="list-style-type: none"> Logout Modifica area utente 	<ul style="list-style-type: none"> Modifica limitazioni Attivazione licenza assegnandolo a un utente

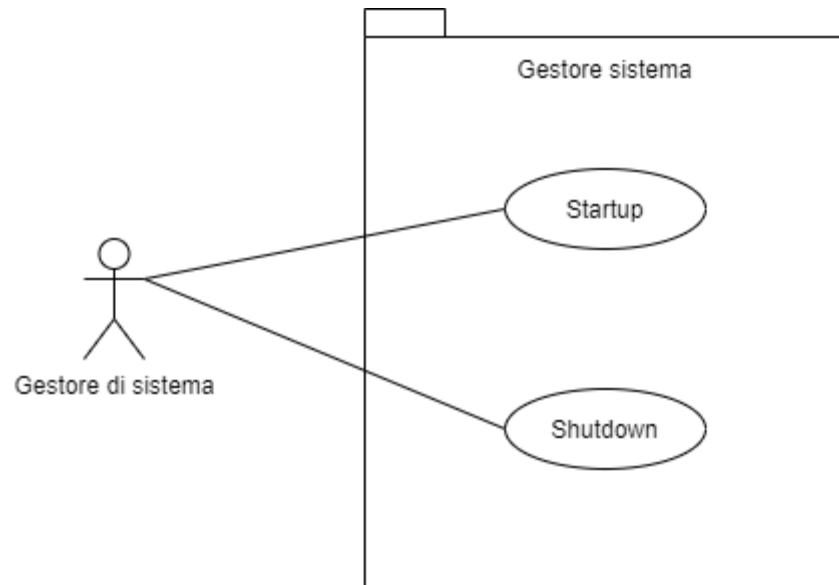
3.6. Controllo flusso globale sistema

Il sistema adotta un controllo di flusso globale di tipo thread-driven, permette l'interazione concorrente tra la WebApp e più client gestiti dal server Apache Tomcat.

Questa soluzione permette al sistema di poter rispondere a più utenti contemporaneamente, ogni richiesta da parte di un utente verrà eseguita in un thread dedicato garantendo la mutua esclusione nelle sezioni critiche.

3.7. Condizione limite

Le boundary conditions sono relative all'avvio, allo spegnimento del sistema e fallimento del sistema.



3.7.1. Start-up

Identificativo UC_CL_01	Start-up	Data	27/11/20
		Vers.	1.0
		Autore	Tutti
Descrizione	Lo UC fornisce la funzionalità dell'avvio del sistema.		
Attore Principale	Gestore di sistema Inizializza il caso d'uso.		
Attori secondari	NA		
Entry Condition	Il gestore ha l'accesso alla macchina del sistema.		
Exit condition Onsuccess	Il sistema è avviato correttamente.		
Exit condition On failure	Il sistema non è avviato.		
Rilevanza/User Priority	Alta		
Frequenza stimata	5/anno		
Extension point	NA		
Generalization of	NA		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			



1	Gestore di sistema:	Avvia MySQL.
2	Gestore di sistema:	Avvia Tomcat.
3	Sistema:	Visualizza messaggio di avvenuto avviamento con successo.
Note		
Special Requirements		
3.7.2. Terminazione		
Identificativo UC_CL_02	Shutdown	Data 27/11/20
		Vers. 1.0
		Autore Tutti
Descrizione	Lo UC fornisce la funzionalità della terminazione del sistema.	
Attore Principale	Gestore di sistema Inizializza il caso d'uso.	
Attori secondari	NA	
Entry Condition	Il gestore ha l'accesso alla macchina del sistema.	
Exit condition Onsuccess	Il sistema è terminato correttamente.	
Exit condition On failure	Il sistema non è terminato.	
Rilevanza/User Priority	Alta	
Frequenza stimata	5/anno	
Extension point	NA	
Generalization of	NA	
1	Gestore di sistema:	Termina MySQL.
2	Gestore di sistema:	Termina Tomcat.
3	Sistema:	Visualizza messaggio di avvenuta terminazione con successo.
Note		
Special Requirements		



3.7.3. Fallimento

Si potrebbero verificare fallimenti sia hardware che software:

- Fallimento hardware
 - Corruzione dei dati e perdita parziale o totale dei dati persistenti: il sistema non prevede una strategia di backup a causa del budget.
- Fallimento software
 - Impossibilità di connessione al database: il sistema visualizza un messaggio di errore e invita l'utente a riprovare più tardi.

4. Servizi dei Sottosistemi

In questa sezione vengono descritti i servizi che ogni sottosistema implementa.

Gestione Licenza	
Servizio	Descrizione
Visualizzazione licenza	Il sottosistema permette di visualizzare tutte le licenze possedute
Visualizzazione pk	il sottosistema permette di visualizzare tutti i pk associati a quella licenza, in particolare permette di controllare se un pk è stato già riscosso oppure no
Attivazione licenza utente	Il sottosistema permette all'utente normale di attivare una licenza tramite pk
Attivazione licenza sviluppatore	Il sottosistema permette all'utente sviluppatore di attivare una licenza assegnandola ad un determinato utente
Modifica limitazioni	Il sottosistema permette all'utente sviluppatore di modificare la durata e le limitazioni di una determinata licenza
Aggiunta nuova licenza	Il sottosistema permette di aggiungere una nuova licenza



Gestione lista software	
Servizio	Descrizione
Visualizza lista	Il sottosistema permette di visualizzare tutti i software disponibili
Aggiunta software	Il sottosistema permette di aggiungere un nuovo software alla lista
Rimozione software	Il sottosistema permette di rimuovere un software presente

Gestione software	
Servizio	Descrizione
Visualizza software	Il sottosistema permette di visualizzare le informazioni del software nel dettaglio
Modificare info	Il sottosistema permette di modificare le informazioni di ogni software disponibile



Gestione utenza	
Servizio	Descrizione
Login	Il sottosistema permette di effettuare l'accesso inserendo le proprie credenziali
Logout	Il sottosistema permette di disconnettere la sessione dell'utente loggato.
Modifica area utente	Il sottosistema permette di modificare tutte le informazioni relative all'utente con la possibilità di diventare utente sviluppatore anche dopo la registrazione
Controllo via API	Il sottosistema permette di effettuare un controllo via API

Gestione pk	
Servizio	Descrizione
Generazione pk	Il sottosistema permette di generare un product key relativo ad un software specifico



5. Glossario

Termine	Definizione
Utente	Utente finale che usufruisce dei prodotti software
Sviluppatore	Utente che mette a disposizione i propri prodotti software per gli utenti finali
API	Application Program Interface. Esegue l'autenticazione per determinare l'autenticità di una licenza posseduta.
PK	E un codice alfanumerico. Utilizzato dall'utente per attivare una licenza.
Licenza	È il contratto definito dall'utente sviluppatore che permette l'utilizzo del prodotto software per gli utenti finali
Area Utente	È anche vista come l'area personale dell'utente, contenente informazioni personali