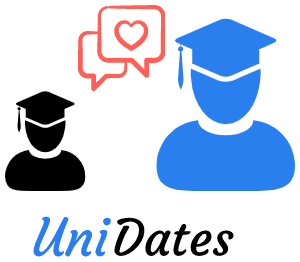
|  |
| --- |
| **Università degli Studi di Salerno Corso di Ingegneria del Software** |

**UniDates  
System Design Document  
Version 1.5**



Data: 30/01/2020

**Coordinatore del progetto:**

|  |  |
| --- | --- |
| **Nome** | **Matricola** |
| Andrea De Lucia |  |

**Partecipanti:**

|  |  |
| --- | --- |
| **Nome** | **Matricola** |
| Alessandro Pio D’Antuono [APD] | 0512105700 |
| Antonio Fasulo [AF] | 0512105904 |
| Raimondo Rapacciuolo [RR] | 0512105778 |
| Roberta Gambardella [RG] | 0512106012 |

|  |  |
| --- | --- |
| **Scritto da:** |  |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versione** | **Descrizione** | **Autore** |
| 10/11/2020 | 1.1 | Aggiunta di un template per il documento e introduzione | Tutti |
| 11/11/2020 | 1.2 | Aggiunta sottosistemi | Tutti |
| 14/11/2020 | 1.3 | Boundary Conditions | Tutti |
| 16/11/2020 | 1.4 | Gestione e controllo della sicurezza | Tutti |
| 17/11/2020 | 1.5 | Serivizi sottosistemi | Tutti |
| 30/01/2021 | 1.5 | Revisioni | Tutti |

1. **Introduzione**
2. **Obiettivi del sistema**

Il sistema da realizzare é un sito di dating: “UniDates.it”.

Sulla nostra piattaforma, ogni utente, puó registrarsi e creare un profilo personale caratterizzato da i propri dati anagrafici, alcuni dati personali, una o piú foto ed una lista di topic che esprimono alcune preferenze in diverse tematiche. Il profilo sará sempre modificabile successivamente dall’utente. Ogni utente avrá una propria sezione “Home” in cui sará possibile scorrere le foto di altri utenti che saranno selezionate dal nostro sistema di matchmaking che sceglierà quelle che piú corrispondono ai topic scelti.

Se due utenti si scambiano reciprocamente il “mi piace”, il sistema dará la possibilitá a questi ultimi di poter conversare in un’apposita finestra di chat. Un utente puó anche segnalare una foto altrui, se ritiene questa troppo offensiva, inserendo un motivo della segnalazione e specificandone i dettagli. Inoltre un utente potrá bloccare un altro utente se ritiene di essere infastidito da quest’ultimo.

La gestione del sito è affidata al Community Manager e ad i moderatori che si occuperanno di ricevere tutte le segnalazioni ricevute dagli utenti.

Il moderatore avrá il compito di inviare un ammonimento, con conseguente oscuramento della foto segnalata, allo studente che non ha rispettato il regolamento del sito (specificandone motivazione e dettagli) , o nel caso di ammonimenti multipli, segnalare l’utente al Community Manager (CM) o sospenderlo temporaneamente. Il CM avrá il compito di visualizzare le segnalazioni inviate dai moderatori e di sospendere permanentemente un utente della piattaforma in seguito ad ammonimenti multipli.

1. **Design Goals**

**2.1 Criteri di performance**

* Il tempo di pubblicazione di una foto è di un massimo di 5 secondi.

**2.2 Criteri di affidabilità**

* Eventuali inserimenti di input non validi sono opportunamente segnalati all’utente, il quale ha la possibilità di effettuare il re-inserimento.

**2.3 Criteri di usabilità**

* L’interfaccia si adatta a qualsiasi dispositivo.

**2.4 Criteri di sicurezza**

* La password di ogni account sará criptata per garantire la sicurezza degli utenti.
* Le informazioni personali tra due studenti saranno completamente visibili soltanto quando questi due avranno un match.

1. **Trade-offs**

**[Ottimizzazione vs Portabilitá]**

Al costo di presentare una minore ottimizzazione del sistema, quest’ultimo andrá a favorire una maggiore portabilitá, potendo essere eseguito sulla maggior parte dei dispositivi, con le medesime caratteristiche.

**[Manutenzione vs Disponibilitá]**

Al costo di avere una minore disponibilità dei servizi durante il periodo di manutenzione, sarà possibile effettuare manutenzioni più veloci e dirette mirata a risolvere i problemi presenti.

**[Scalabilitá vs Costi]**

Il sistema presenterà una scalabilità elevata anche se questo dovrebbe inficiare su maggiori costi di gestione.

**[Sviluppo rapido vs Funzionalitá]**

Al costo di velocizzare la fase di sviluppo, il sistema finale sarà ridotto di alcune funzionalità.

1. **Definizioni, acronimi ed abbreviazioni**

* RAD: Requirements Analysis Document.
* SDD: System Design Document.
* ODD: Object Design Document.
* DB: Database.
* Greenfield engineering: Tipologia di sviluppo che comincia da zero, non esiste nessun sistema a priori e i requisiti sono ottenuti dall’utente finale e dal cliente. Nasce, perciò, a partire dai bisogni dell’utente.

1. **Riferimenti**

* Bernd Bruegge & Allen H. Dutoit, Object Oriented Software Engineering: Using UML, Patterns and Java, (2nd Prentice Hall, 2003).
* <http://www.tinder.com>

1. **Architettura del sistema e dei sottosistemi**

**6.1 Overview**

Il sistema UniDates è un’applicazione web, realizzata attraverso il modello client server.

L’obiettivo che si pone è di offrire un sito di incontri tra studenti universitari.

Il client gestisce la parte di presentazione, il server gestisce la logica applicativa e delega le funzioni di storage e gestione della persistenza ad un DBMS.

All’interno del sistema i manager sono individuati in base alle funzionalità inerenti al sottosistema, in modo tale da gestire alta coesione e basso accoppiamento.

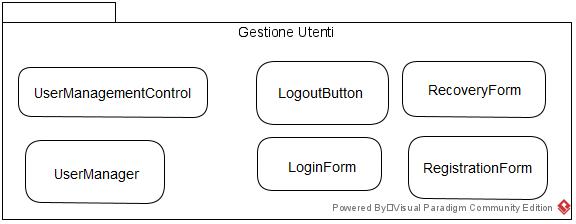
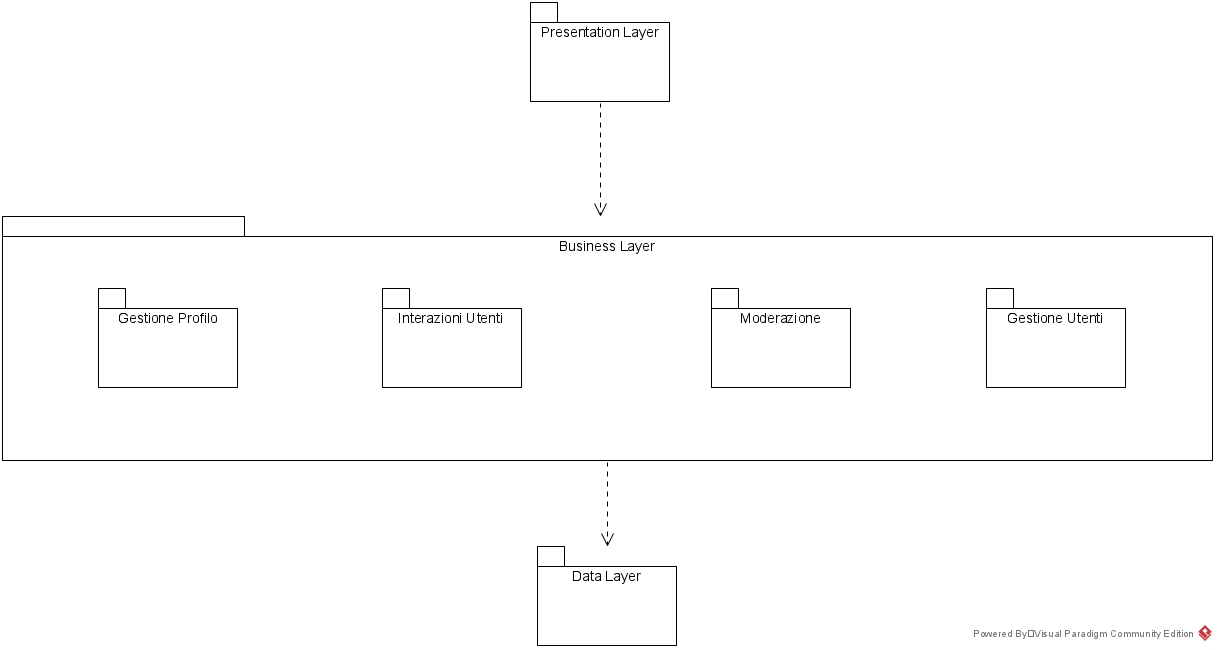
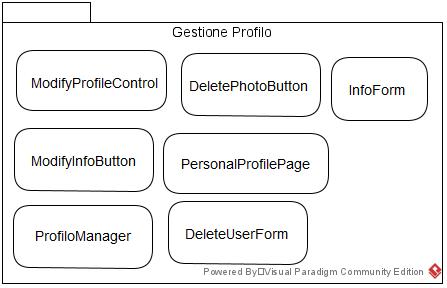
Questo aspetto rende il sistema molto più manutenibile.

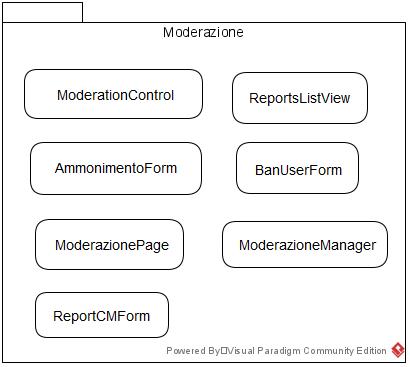
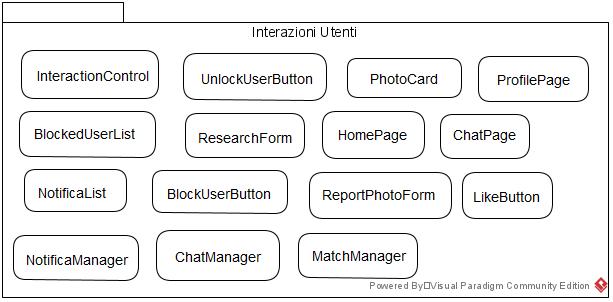
**6.2 Decomposizione in sottosistemi**

Il sistema è stato scomposto in quattro sottosistemi al fine di raggiungere l’obiettivo di minimizzare l’accoppiamento e garantire una forte coesione.

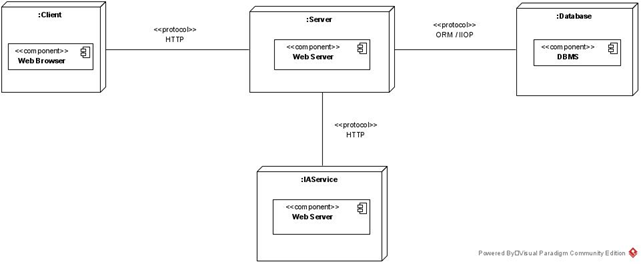
I sottosistemi sono divisi per tipo di funzionalità (dove facciamo riferimento nel documento RAD, nella sezione 1 di “Requisiti”):

* **Gestione Utenti**: comprende tutte le funzionalità riguardanti gli utenti, e corrispondono ai requisiti funzionali: RF1, RF2, RF3, RF4.
* **Interazioni Utenti**: comprende tutte le funzionalitá riguardanti le interazioni tra gli utenti (scambio di messaggi, mettere like alle foto, visualizzare altri profili etc.): RF8, RF9, RF10, RF11, RF12, RF13, RF14, RF15, RF16, RF17.
* **Gestione Profilo**: comprende tutte le funzionalità che rendono possibile la cancellazione e la modifica del profilo. Le funzionalità corrispondono ai requisiti funzionali: RF5, RF6, RF7.
* **Moderazione**: comprende le funzionalità di segnalazioni, ammonimento e sospensioni utente e/o foto utente. Le funzionalità corrispondono ai requisiti: RF18, RF19, RF20, RF21, RF22, RF23, RF24, RF25.





**6.3 Mapping Hardware-Software**



Il sistema sarà installato su una macchina che fungerà da server (sarà quindi provvista di un WebServer, ossia quello fornito da Spring Boot).

Il server interagirà con un database (come DBMS verrà utilizzato PostgreSQL).

Il client sarà un qualsiasi computer provvisto di browser ed una connessione ad internet. Il modulo di intelligenza artificiale si troverà su una macchina separata rispetto al server. Quest’ultimo vi accederà come un servizio esterno (API REST).

Il sistema finale sarà composto da 4 macchine diverse:

* Una che fungerà da Server su cui sarà installato un Web Server (fornito da Spring Boot)
* Una conterrà il database che interagirà con il Server (come DBMS viene usato PostgreSQL)
* Il client potrà essere qualsiasi macchina provvista di un browser ed una connessione ad Internet
* Un Web Service che integrerà il modulo di Intelligenza Artificiale che fornirà delle API Rest al server, per l’accesso.

1. **Gestione persistente dei dati**

Per la persistenza dei dati, si è pensato di utilizzare un database piuttosto che un file, la scelta è basata sul fatto si necessita di un’organizzazione logica dei dati, di sicurezza per l’utilizzo e gestione dei dati sensibili.Inoltre la scelta è influenzata dal fatto che i dati sono accessibili in concorrenza da più utenti che li richiedono.

I dati che saranno resi persistenti sono (Riferimenti RAD, sezione: “Class Diagram”):

* Studente, Moderatore, Community Manager
* Notifica
* Match
* Chat
* Messaggio
* Profilo
* Foto
* Segnalazione
* Ammonimento
* Sospensione

1. **Gestione e controllo della sicurezza**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attori/Sottosistemi** | **Gestione Utenti** | **Interazioni Utenti** | **Gestione Profilo** | **Moderazione** |
| **Utente non registrato** | * Registrazione |  |  |  |
| **Studente** | * Login * Logout * Recupero password | * Mettere “Mi piace” * Iniziare una chat * Avviare una chat * Visualizzare profilo altrui * Visualizzare le notifiche * Visualizzazione della home * Ricerca tramite email * Inviare segnalazioni. * Bloccare/Sbloccare un utente. | * Modificare le informazioni del profilo * Cancellazione profilo * Visualizzare profilo personale |  |
| **Moderatore** | * Login * Logout * Recupero password |  |  | * Inviare un ammonimento * Sospendere temporaneamente uno studente * Visualizzare segnalazioni utente * Accedere al pannello moderazione * Oscurare una foto * Inviare una segnalazione al CM |
| **Community Manager** | * Login * Logout * Recupero password |  |  | * Sospendere uno studente * Visualizzare segnalazioni moderatore |

1. **Boundary Condition**

|  |  |
| --- | --- |
| ***Nome*** | **Startup del sistema** |
| ***Attori partecipanti*** | Amministratore del sistema. |
| ***Entry condition*** | L’amministratore del sistema si trova sulla macchina che ospita il web server. |
| ***Flusso di eventi*** | 1. L’amministratore effettua il deploy  2. Il sistema inizia l’esecuzione e  inizializza i manager |
| ***Exit condition*** | Il sito diventa raggiungibile. |

|  |  |
| --- | --- |
| ***Nome*** | **Shutdown del sistema** |
| ***Attori partecipanti*** | Amministratore del sistema. |
| ***Entry condition*** | L’amministratore del sistema si trova sulla macchina che ospita il web server che è attualmente in esecuzione. |
| ***Flusso di eventi*** | 1. L’amministratore effettua l’undeploy del sistema. 2. Il sistema notifica gli utenti dello shutdown. 3. Il sistema termina l’esecuzione. |
| ***Exit condition*** | Il sito non è più raggiungibile. |

1. **Servizi dei sottosistemi**
2. **Gestione utenti**

|  |  |
| --- | --- |
| **Servizi** | **Descrizione** |
| registrazione | Servizio che si occupa della registrazione di un utente non registrato al sistema. |
| login | Servizio che permette ad un utente registrato di effettuare il login al sistema. |
| isActive | Servizio che permette di controllare se un utente è stato attivato. |
| isBanned | Servizio che permette di controllare se un utente è attualmente sospeso. |
| isPresent | Servizio che permette di controllare se un utente è presente nel sistema. |

1. **Gestione profilo**

|  |  |
| --- | --- |
| **Servizi** | **Descrizione** |
| cancellaProfilo | Servizio che permette allo studente di cancellare il proprio profilo. |
| modificaProfilo | Servizio che permette allo studente di modificare le informazioni e le foto personali. |
| visualizzaProfiloPersonale | Servizio che permette ad uno studente di visualizzare il proprio profilo. |

1. **Interazioni Utenti**

|  |  |
| --- | --- |
| **Servizi** | **Descrizione** |
| avviareChat | Servizio che permette ad uno studente di avviare una chat. |
| visualizzaStudentiSuggeriti | Servizio che permette ad uno studente di visualizzare le informazioni di altri studenti suggeriti. |
| visualizzaNotifiche | Servizio che permette ad uno studente di visualizzare tutte le notifiche. |
| ricercaUtente | Servizio che permette ad uno studente di ricercare gli altri utenti tramite indirizzo email |
| aggiungiMatch | Servizio di sistema che aggiunge il match tra due studenti. |
| invioMessaggio | Servizio che consente ad uno studente di inviare un messaggio ad un altro utente. |
| visualizzaAltriProfili | Servizio che consente ad uno studente di visualizzare i profili di altri studenti. |
| bloccaStudente | Servizio che permette ad uno studente di bloccare uno studente. |
| sbloccaStudente | Servizio che permette ad uno studente di sbloccare uno studente bloccato |

1. **Moderazione**

|  |  |
| --- | --- |
| **Servizi** | **Descrizione** |
| inviaSegnalazione | Servizio che permette ad uno studente di segnalare una foto |
| inviaAmmonimento | Servizio che permette ad un moderatore/CM di inviare un ammonimento |
| inviaSospensione | Servizio che permette ad un CM di sospendere uno studente |
| segnalazioniUtente | Servizio che permette ad un moderatore di visualizzare le segnalazioni ricevute dagli utenti. |
| ammonimentiRicevuti | Servizio che permette ad uno studente di visualizzare gli ammonimenti ricevuti. |
| segnalazioniModeratori | Servizio che permette ad un CM di visualizzare le segnalazioni ricevute dai modaratori. |