

“Software Engineering”

Course

a.a. 2019-2020

Lecturer: Prof. Henry Muccini (henry.muccini@univaq.it)

Progetto 6

Mettimi in contatto con...

Date	<22/12/2019>
Deliverable	Deliverable #1
Team (Name)	Flop Team

Team Members		
Nome e cognome	N# di matricola	E-mail
Alessandro D'Orazio	251811	alessandro.dorazio2@student.univaq.it
Lorenzo Cilli	253121	lorenzo.cilli@student.univaq.it

Table of Contents of this deliverable

1. Specifiche del progetto
2. Challenge/Risks Requirements or Tasks
3. Stato dell'arte
4. Raffinamento dei requisiti
5. Architettura software
6. Dati e loro modellazione
7. Design decisions
8. Design di basso livello
9. Come i NFR e FR sono soddisfatti dal design
10. Effort Recording
11. Appendix Prototype

1. Specifiche del progetto

UniChat è un servizio di messaggistica disponibile per contesti universitari. Permette infatti di effettuare lo scambio di messaggi tra studenti universitari, docenti e personale amministrativo. Ogni università ha un database differente, con lo stesso schema.

Le conversazioni sono raggruppate in stanze. Le stanze possono essere di tre tipi:

1. Chat diretta, in cui la stanza contiene solamente due partecipanti, ed entrambi i partecipanti possono inviare messaggi. La comunicazione è multidirezionale.
2. Gruppo, in cui la stanza contiene due o più partecipanti e tutti i partecipanti possono inviare messaggi. La comunicazione è multidirezionale.
3. Feed, in cui la stanza contiene due o più partecipanti, suddivisi in amministratori e lettori. Gli amministratori possono inviare e leggere messaggi, mentre i lettori possono esclusivamente leggerli. La comunicazione è monodirezionale.

2. Challenging/Risky Requirements or Tasks

Challenging Task	Date the task is identified	Date the challenge is resolved	Explanation on how the challenge has been managed
Gestione di chat diretta, gruppi e feed	09/12/2019	11/12/2019	Chat diretta, gruppi e feed vengono viste come stanze dal sistema. In questo modo, queste tre entità diventano una tipologia di stanza
Ottimizzazione invio messaggi	19/12/2019	21/12/2019	Uso di WebSocket

3. Stato dell'Arte

Schoology

Schoology è un social network e un ambiente di apprendimento virtuale.

Il software permette la condivisione di documenti in ambito universitario, infatti, accendendo tramite la propria università ed iscrivendosi ai corsi di interesse, è possibile comunicare con il docente, tramite un sistema di messaggistica, ed usufruire del materiale da lui caricato.

La condivisione dei documenti però, non è vincolata solamente ai docenti, anche uno studente può caricare i propri file, creare le proprie cartelle e farle vedere a qualunque utente sfogli il proprio profilo.

Il sistema, tuttavia, non permette la comunicazione tra studenti.

La sezione, nella homepage, delle attività recenti presenta un ottimo spunto per la gestione dei feed, infatti si presenta come una pagina nella quale un solo utente, gli/l'admin del corso, può effettuare comunicazioni che verranno inviate ad ogni utente iscritto.

Edmodo

Edmodo è un software che permette la comunicazione, collaborazione e coaching per scuole e insegnanti.

Il punto forte di Edmodo sono i differenti tipi di account, ne esistono tre: insegnante, studente e genitore.

Edmodo è molto concentrato sull'insegnante, infatti per accedere come studente o genitore bisogna avere il codice di accesso del corso da seguire/ seguito dal proprio figlio.

L'insegnante, come ogni piattaforma del genere, può condividere materiale, contattare uno o più studenti, assegnare compiti o quiz.

Lo studente, invece, per registrarsi ha bisogno del codice di accesso che solo un insegnante può fornire; una volta entrato nel corso può comunicare direttamente con il docente, usufruire dei materiali e svolgere i compiti da lui caricati.

L'utente genitore invece ha una funzione più da spettatore, questo infatti può vedere l'andamento del proprio figlio, sempre utilizzando il codice del corso fornito dall'insegnante, i suoi voti e persino comunicare con il docente per qualsiasi eventualità.

Telegram

Telegram è un servizio di messaggistica basato su cloud, che permette lo scambio di messaggi, documenti, immagini, ecc. in una conversazione o in gruppi; anch'esso nasce come servizio mobile per poi espandersi anche su PC con il software Telegram Web.

I messaggi inviati sono salvati sul cloud di Telegram, così da garantire la sincronizzazione istantanea, il risultato consente all'utente di poter accedere ai messaggi da diversi dispositivi contemporaneamente.

Una delle caratteristiche principali di Telegram che lo distingue dal resto dei suoi concorrenti sono le chat (classiche e segrete), i canali e i bot.

Le chat, come già detto, si dividono in due tipi, le chat classiche, che utilizzano una cifratura client-server ovvero è cifrata dal dispositivo fino al server quindi la conversazione rimane salvata in maniera cifrata sui server, e le chat segrete, che utilizza una cifratura *end-to-end* ossia è cifrata fra i due dispositivi coinvolti nella conversazione, questa chat ha un'ulteriore funzione: l'autodistruzione; questa funzione permette di impostare un timer nella chat e di eliminare tutti i contenuti alla fine del countdown aumentando la sicurezza per la privacy. I canali, chat in cui chiunque sia amministratore può inviare messaggi ai membri del canale, anche se questi ultimi non possono rispondere né commentare. I Bot sono degli account Telegram, gestiti da un programma, che offrono molteplici funzionalità con risposte immediate e completamente automatizzate.

4. Raffinamento dei Requisiti

4.1 Servizi (con prioritizzazione)

Importanza alta = 3 / Importanza media = 2 / Importanza bassa = 1

Complessità alta = 3 / Complessità media = 2 / Complessità bassa = 1

1. Stanza

1.1. Visualizzazione delle stanze. I=3 / C=1

L'utente può visualizzare le stanze in cui è partecipante o amministratore, in modo tale da poter inviare e visualizzare i messaggi relativi alla stanza, e nel caso sia amministratore, gestire la stanza.

1.2. Accesso ad una stanza. I=2 / C=2

Un utente può condividere un link ad altri utenti per l'accesso ad un gruppo o feed. Questo non accade nelle chat dirette poiché l'utente che crea la stanza seleziona all'atto della creazione l'altro utente.

1.3. Comunicazioni importanti. I=2 / C=2

L'utente amministratore di un gruppo o feed può mettere in evidenza dei messaggi. Evidenziando un messaggio, esso sarà sempre visibile per un periodo di tempo stabilito dall'amministratore, e sarà inviata una notifica ai partecipanti.

Nelle chat dirette entrambi i partecipanti possono evidenziare i messaggi, e gli effetti sono i medesimi dei messaggi evidenziati in gruppi e feed.

1.4. Informazioni della stanza. I=2 / C=1

Sarà possibile visionare i partecipanti della stanza e, se presenti, nome e amministratori di essa.

1.5. Archiviazione di una stanza. I=1/C=1

La stanza può essere archiviata dall'amministratore di un gruppo o di un feed, oppure da un partecipante della chat diretta, per nascondere la chat nella sezione "chat archiviate" e per non ricevere notifiche relative alla stanza.

1.6. Gestione dello SPAM. I=2 / C=2**1.6.1. Sistema antiflood. I=2 / C=3**

Non è possibile inviare lo stesso messaggio per tre volte di fila nell'arco di un minuto. In questo modo, la conversazione risulterà più pulita e funzionale.

1.6.2. Segnalazioni. I=2 / C=1

L'utente di una stanza può segnalare un messaggio.

La segnalazione arriverà all'amministratore tramite notifica, che potrà decidere se silenziare o meno il mittente del messaggio.

2. Chat diretta**2.1. Creazione di una chat diretta. I=3 / C=1**

L'utente che vuole comunicare tramite chat diretta con un altro utente può creare una stanza apposita, nel caso in cui non esista già. Se la stanza esiste già, la comunicazione tra i due utenti mediante chat diretta viene effettuata nella stanza già esistente.

3. Gruppo**3.1. Creazione di un gruppo. I=3 / C=1**

All'atto di creazione di un gruppo, viene creata una stanza dove l'utente creatore viene nominato amministratore. Deve essere assegnato un nome al gruppo.

3.2. Gli amministratori di un gruppo possono modificare il nome del gruppo, aggiungere/rimuovere partecipanti, nominare altri amministratori ed evidenziare i messaggi importanti.**3.3. Gestione del ban I=2 / C=2****3.3.1. Gli amministratori di una stanza possono silenziare un utente all'interno di essa per un periodo di tempo scelto dall'amministratore. Prima del ban, l'amministratore può consultare uno storico relativo all'utente, che può utilizzare come parametro per stabilire la durata del ban.****3.3.2. Il ban può essere rimosso da un amministratore manualmente, oppure viene rimosso automaticamente alla scadenza del periodo di tempo****3.4. Ban automatico. I=2 / C=3**

È presente un dizionario di parole, stabilito dal personale amministrativo, in cui sono presenti tutte le parole vietate.

3.4.1. Nel caso in cui venga inviato un messaggio con una parola vietata, il messaggio non viene inviato e viene inviata una

notifica al mittente, contenente “il messaggio è stato eliminato per parole vietate”. La terza volta che uno stesso utente utilizza una parola vietata, esso viene silenziato per un giorno.

- 3.4.2. Gli sviluppatori del software forniscono una lista di parole vietate, per agevolare i compiti del personale amministrativo, che si occupa di inserire e rimuovere le parole nel dizionario.

- 3.5. Motivazioni. Un utente può essere bannato per linguaggio scorretto o per spam.

4. Feed

- 4.1. Creazione di un feed. I=3 / C=1

All'atto di creazione di un feed, viene creata una stanza dove l'utente creatore viene nominato amministratore. Deve inoltre essere assegnato un nome al feed.

- 4.2. Gli amministratori di un feed possono modificare il nome del feed, aggiungere/rimuovere partecipanti, nominare altri amministratori ed evidenziare i messaggi importanti.

5. Gestione degli utenti

- 5.1. Visualizzazione degli utenti presenti nella piattaforma. I=3 / C=1

UniChat prevede un sistema per la visualizzazione degli utenti registrati alla stessa università dell'utente autenticato. Qui è possibile vedere le informazioni di contatto degli utenti

- 5.2. Ban dalla piattaforma. I=2 / C=2

Qualora l'utente effettui molteplici volte delle azioni non consentite, come lo spam di messaggi e l'utilizzo di vocaboli inadeguati, è dovere del personale amministrativo di bannare l'utente dalla piattaforma per un periodo di tempo prefissato. In questo modo, l'utente non potrà temporaneamente inviare messaggi. Nel caso si tratti di un errore, l'utente può inviare una mail per effettuare un “ricorso”, in modo tale da rimuovere il ban prima che il tempo scada.

6. Filtraggio degli utenti. I=2 / C=2

Gli utenti possono essere filtrati per tipologia, facoltà e dipartimento. Tramite questi filtri è possibile inviare messaggi, creare gruppi e feed.

7. Invio e ricezione di messaggi. I=3 / C=2

Questo servizio consiste nello scambio di messaggi all'interno di una stanza. I messaggi devono essere scambiati real-time.

- 7.1. Invio di messaggi a più stanze. È possibile inviare lo stesso messaggio a più stanze tramite un sistema di filtraggio che mostra tutte le stanze

dell'utente.

- 7.2. Thread. È possibile commentare un messaggio, per facilitare l'esperienza d'uso nel caso ci siano molti messaggi.

8. **Registrazione.** I=3 / C=1

Gli utenti si possono registrare fornendo nome, cognome, email, password, tipo, università e facoltà. La registrazione viene effettuata autonomamente dall'utente.

- 8.1. Tipologie di utenti: studente, docente o personale dell'amministrazione.

8.2. Verifica degli utenti. I=3 / C=1

I docenti e il personale dell'amministrazione devono essere verificati per motivi di sicurezza, altrimenti qualsiasi utente potrebbe registrarsi come docente o personale amministrativo.

La verifica avviene mediante l'invio di un'email ad un utente del personale amministrativo dalla casella universitaria, contenente una scansione dei propri documenti. Il personale amministrativo dovrà poi confermarlo tramite un apposito pannello.

9. **Autenticazione.** I=3 / C=1

Gli utenti effettuano il login inserendo email e password

- 9.1. Reset della password. I=3 / C=1

Gli utenti possono autonomamente reimpostare la propria password nel caso in cui sia stata dimenticata.

10. **Comunicazioni con servizi esterni.**

- 10.1. Visualizzazione del numero telefonico. I=2 / C=1

Un utente può visualizzare il numero telefonico di un altro utente, per effettuare chiamate e inviare SMS.

- 10.2. Email. I=2 / C=1

L'utente può visualizzare l'email di un altro utente per avviare una conversazione. La conversazione avviene al di fuori di UniChat

- 10.3. App di messaggistica. I=1 / C=1

L'utente può impostare i propri contatti delle app di messaggistica, come Skype e Twitter.

11. **Ricerca dei messaggi.** I=2 / C=3

L'utente può effettuare una ricerca intelligente dei messaggi, che permette di visualizzare i messaggi contenenti una determinata frase (o parola), oppure i messaggi inviati in una determinata data.

12. Gestione delle informazioni dell'università. I=2 / C=2

Il personale amministrativo può inserire e rimuovere dipartimenti e corsi di laurea.

12.1. Inserimento di un'università. I=3 / C=1

È compito del team di sviluppo di inserire, modificare ed eliminare le università, tramite un apposito pannello.

4.2 Requisiti non Funzionali

1. UniChat deve garantire uno scambio di messaggi sempre disponibile, poiché la perdita di comunicazioni potrebbe causare problematiche di natura logistica e organizzativa. L'up-time deve essere del 99,8%.
2. La password viene codificata mediante un sistema "Bcrypt", per garantire la sicurezza della password in caso di Data Breach
3. Alla richiesta di reset della password, viene inviata un'email all'utente per notificare quest'operazione
4. I docenti e il personale amministrativo vengono verificati dagli utenti del personale amministrativo tramite un apposito pannello
5. L'università dovrà comunicare al team di UniChat, tramite un modello Excel, quali saranno gli utenti del personale amministrativo
6. Gli studenti non possono visualizzare il numero di telefono di docenti e personale amministrativo, poiché potrebbero utilizzarlo in modo inappropriato
7. I dati devono essere trattati seguendo le normative GDPR
8. Il sito Web di UniChat sarà dotato di un certificato SSL, per garantire la connessione sicura
9. Ogni operazione presente in UniChat deve essere possibile con il minor numero di click possibili e tramite un'interfaccia pulita
10. Gli studenti e i docenti saranno in grado di utilizzare al meglio la piattaforma dopo un training di 2 ore, mentre il personale amministrativo dopo un training di 3 ore, date le funzionalità aggiuntive a loro disposizione
11. Il sistema richiede due macchine per garantire l'operatività a seguito di guasti, che vengono risolti, in base al tipo di guasto, da ripresa a caldo o

ripresa a freddo

12. UniChat garantisce lo scambio di 1000 messaggi al secondo, mediante un sistema real-time sviluppato tramite WebSocket
13. Il sistema deve essere facile da usare e deve essere organizzato in modo tale che gli errori commessi dall'utente siano ridotti al minimo.
14. Inizialmente dovrà essere presente un utente master per ogni università, che si occuperà di verificare il personale amministrativo e i docenti. Le credenziali dell'utente master saranno relative ad un membro del personale amministrativo, scelto dall'università.

4.3 Scenari d'uso dettagliati

1. Giorgio è uno studente di biologia. Utilizza ogni giorno UniChat tramite smartphone per scambiare messaggi relativi al progetto a cui sta lavorando, mediante il gruppo che ha creato lui stesso. Non ci ha messo molto a cercare gli altri studenti, poiché ha utilizzato il sistema dei filtri per facoltà. Fissa una scadenza per la consegna del progetto ed evidenzia il messaggio, in questo modo gli altri riceveranno una notifica per leggere il messaggio.
Giorgio inoltre deve sostenere un esame orale di chimica, ma non può presentarsi all'orario deciso dal docente, quindi tramite la chat diretta informa il professore di questa problematica.
Giorgio ha ricevuto qualche settimana fa un messaggio in cui era previsto un cambio aula per la lezione di fisica, ma non riesce a ricordarne i dettagli. Utilizza quindi la funzione di ricerca dei messaggi, filtrando tra i messaggi dell'ultimo mese, quelli in cui è presente il testo "cambia aula". In questo modo, Giorgio riesce a seguire la lezione di fisica.
2. Daniele è un docente di matematica, e ogni mattina, dopo aver bevuto il suo caffè, accende il computer ed apre UniChat. È un docente molto attivo, infatti possiede un feed dedicato all'insegnamento di "Analisi 1", da cui invia le comunicazioni agli studenti. Daniele sa che giovedì prossimo non sarà presente a lezione, quindi tramite i messaggi in evidenza, invia una notifica a tutti gli studenti per informarli. Non deve preoccuparsi di inserire i nuovi studenti nel proprio feed, poiché nella sua pagina personale ha inserito un link per partecipare al feed di "Analisi 1", di cui è docente insieme a due collaboratori, entrambi nominati da Daniele. I tre condividono un gruppo per

fissare gli appuntamenti e gli orari delle esercitazioni.

3. Stefano lavora nel personale amministrativo, e si occupa dei bandi pubblici. È appena uscito un bando pubblico per una borsa di studio riservata agli studenti di medicina. Stefano quindi, tramite il sistema di ricerca integrato in UniChat, invia la comunicazione a tutti gli studenti di medicina, senza doverli cercare manualmente.
Inoltre, ha un nuovo collega di nome Matteo nel personale amministrativo, da cui riceve una mail contenente una scansione dei documenti, per verificarlo nel sistema UniChat. La verifica avviene in meno di 30 secondi!
4. Mario è un ragazzo che si prende gioco degli altri. Prova a registrarsi all'interno di UniChat come docente, per inviare comunicazioni false, ma quando gli viene chiesto di inviare una mail dalla casella universitaria del docente contenente la scansione dei documenti, fallisce nel suo tentativo. Si registra quindi come studente, ed inizia a inviare parole offensive a tutti gli studenti, che però non ricevono nulla, poiché il sistema di SPAM non ha permesso l'invio dei messaggi. Arriva una comunicazione al personale amministrativo, che silenzia Mario dalla piattaforma.

4.4 Excluded Requirements

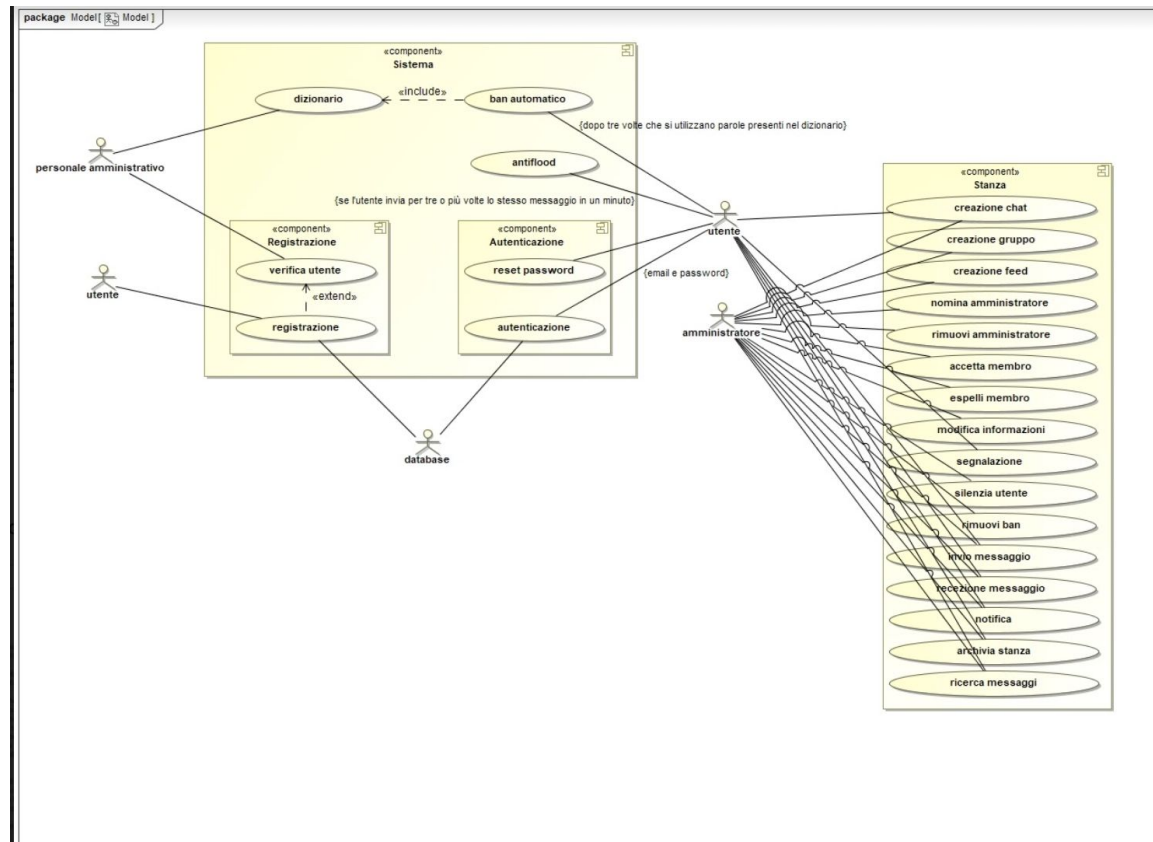
1. Accesso con SPID o credenziali universitarie. Non viene gestito poiché allungherebbe le tempistiche, e non è detto che gli utenti abbiano delle credenziali di questi due servizi. È possibile gestire questa tipologia di accesso come modalità secondaria, ma non è stato fatto per motivi di tempo
2. Utilizzo di allegati. Non possono essere inviati allegati, poiché, richiederebbe uno studio di fattibilità riguardo le performance e lo spazio. Nulla esclude che possano essere gestite in una futura release.
3. L'anno di corso non viene gestito dal software, poiché andrebbero analizzate situazioni come i fuori corso, l'avanzamento automatico e altre casistiche, che per motivi di tempo non sono state analizzate.

4.5 Assunzioni

1. I permessi di ogni tipologia di utente sono gli stessi

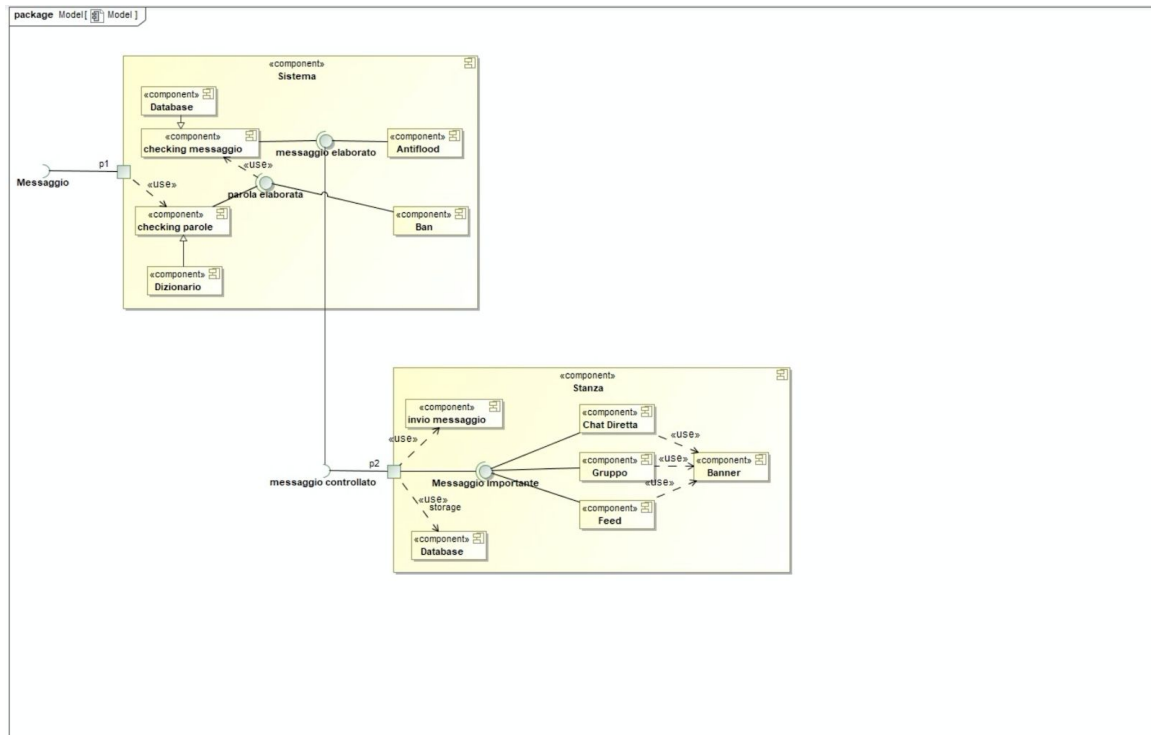
2. Ogni docente o membro del personale amministrativo ha una mail universitaria

4.6 Use Case Diagrams



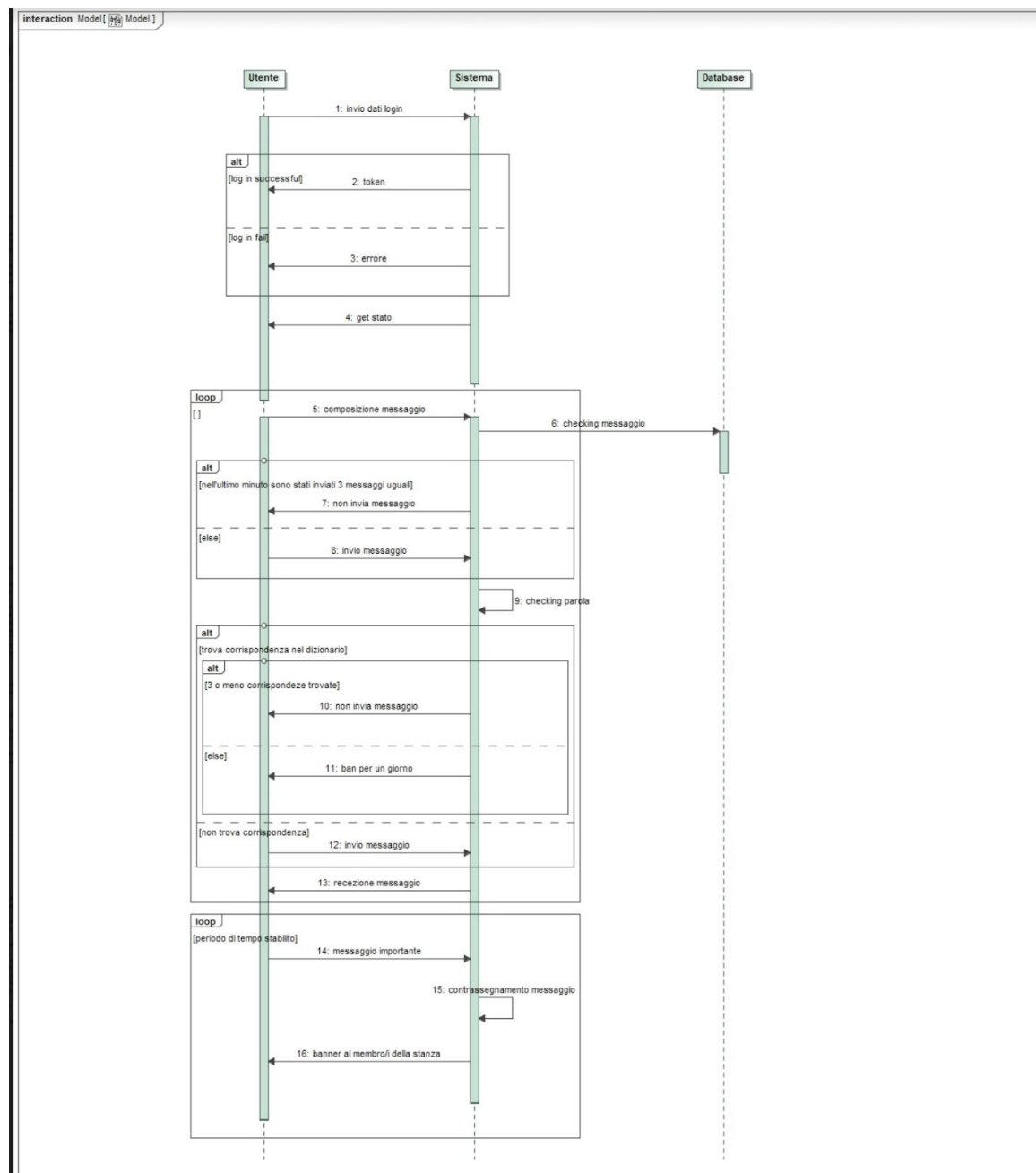
5. Architettura Software

5.1 Vista statica del sistema: Component Diagram



Il component diagram si suddivide in 2 grandi componenti : Sistema e Stanza. Nel Sistema prendiamo in input un messaggio scritto dall'utente. Questo viene controllato parola per parola grazie all'utilizzo di un dizionario delle parole proibite. Se il sistema trova una corrispondenza con una parola scritta dall'utente e con una parola all'interno del dizionario il sistema avvia un protocollo di ban che prevede per le prime due volte la semplice eliminazione del messaggio alla terza si viene bannati (silenziosi per un giorno). Una volta passato il controllo delle parole si passa al controllo del messaggio stesso; se il messaggio, nell'arco dell'ultimo minuto, è presente già tre volte all'interno del database del software questo viene eliminato perché etichettato come messaggio di spam da un sistema di antiflood. Una volta aver superato tutti i controlli il messaggio ormai controllato passa per l'ultimo componente, la Stanza. All'interno della stanza il messaggio viene memorizzato all'interno del database e inviato al destinatario/i; è possibile anche contrassegnarlo come importante, se l'amministratore del gruppo/feed contrassegna il messaggio come importante questo viene inviato sottoforma di banner a tutti gli utenti presenti nella stanza.

5.2 Vista dinamica del sistema: Sequence Diagram



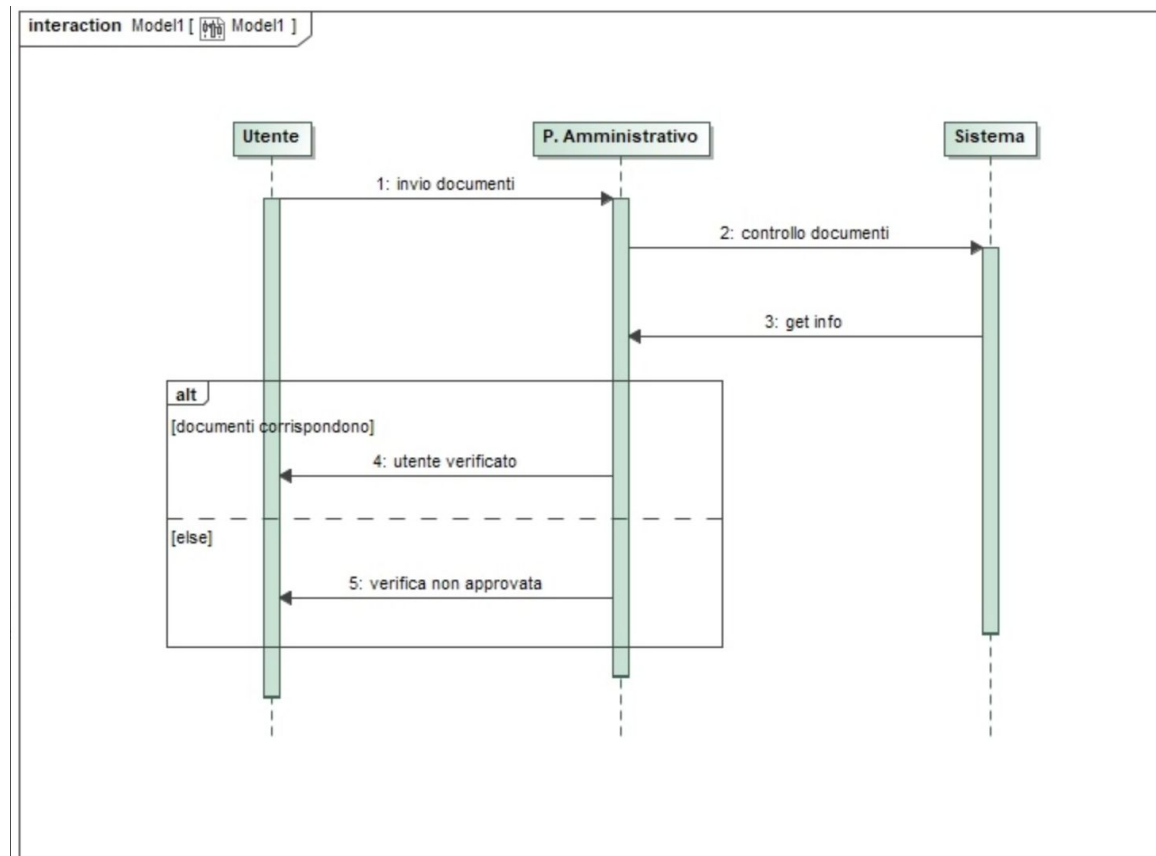
Abbiamo la parte di login dove la lifeline Utente tramite il messaggio "invio dati login" invia i dati al sistema.

A questo punto il sistema json web token fornisce un token o un errore in base all'esito della verifica tramite messaggio verso l'utente; a questo punto il sistema fornisce lo stato all'utente e si può iniziare.

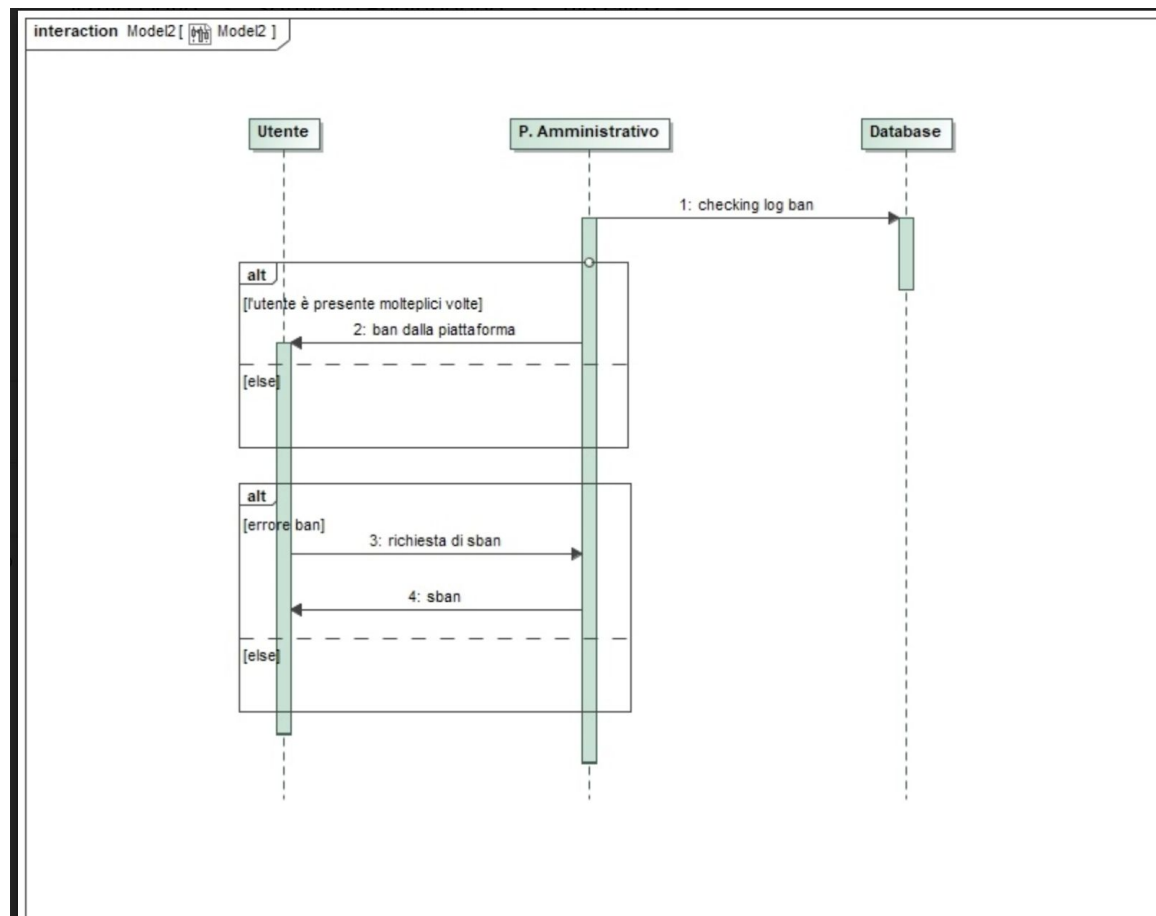
L'utente compone il messaggio che verrà controllato nel database, se nell'ultimo minuto sono presenti 3 messaggi uguali il messaggio non viene inviato altrimenti si;

avviene un altro controllo svolto sul dizionario e se nel messaggio ci sono parole presenti nel dizionario il messaggio viene eliminato, alla terza volta si viene silenziati per un giorno, altrimenti si invia il messaggio.

Infine abbiamo i messaggi importanti, tramite il messaggio "messaggio importante" il sistema contrassegna il messaggio e invia tramite "banner al membro/i della stanza" un banner a tutti gli utenti che si trovano all'interno della stanza dell'utente che ha contrassegnato il messaggio come importante



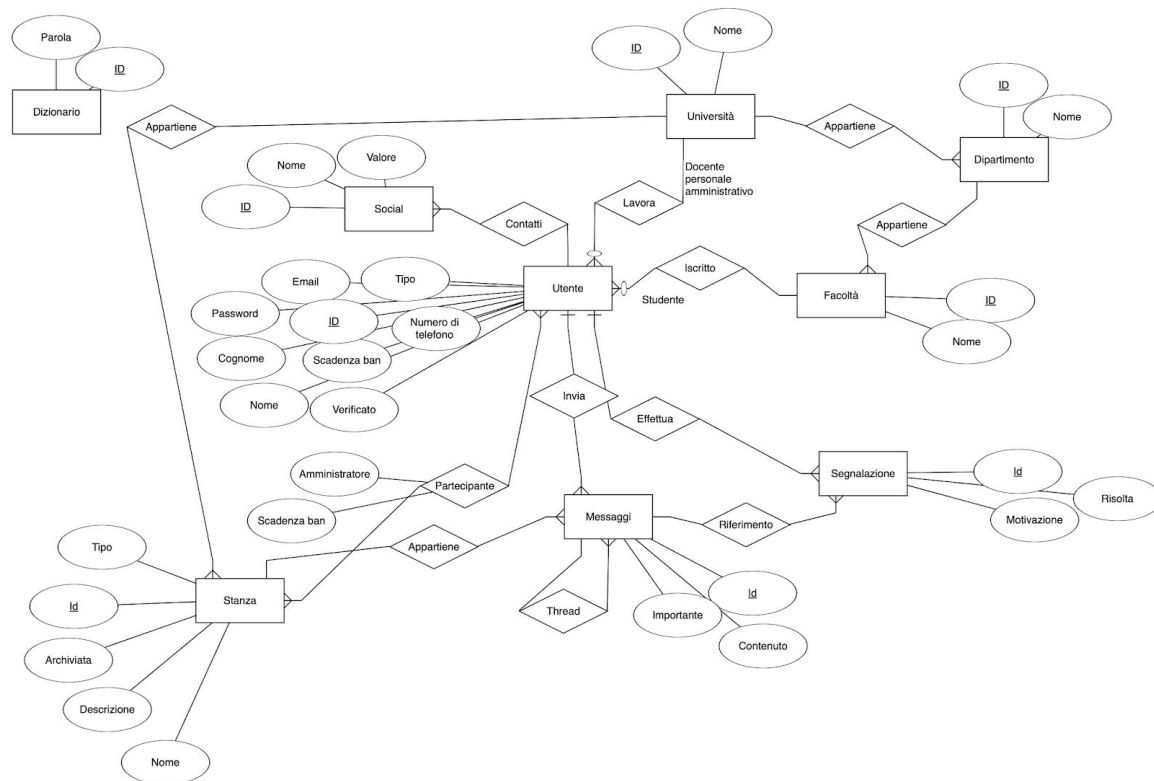
L'utente che vuole verificarsi invia i documenti al personale amministrativo che a sua volta li controlla nel sistema che fornisce le info richieste. A questo punto se le info corrispondono ai documenti inviati l'utente viene riconosciuto altrimenti la verifica viene annullata.



Qui il personale amministrativo controlla il log dei ban dei vari utenti; se un utente risulta presente per tre o più volte il personale amministrativo banna definitivamente l'utente dalla piattaforma.

Può comunque succedere di un errore da parte del personale di fatti è presente, tramite messaggio "richiesta di sban", l'opzione per contattare il personale amministrativo per farsi sbannare

6. Dati e loro modellazione



Chiave primaria *Chiave esterna*

Dizionario (ID, parola)

Università (ID, nome)

Dipartimento (ID, nome, *università_id*)

Facoltà (ID, nome, *dipartimento_id*)

Utente (ID, nome, cognome, email, password, tipo, verificato, numero_di_telefono, scadenza_ban, *facoltà_id*, *università_id*)

L'utente, nel caso sia uno studente, è associato al suo corso di laurea, mentre se è un docente o fa parte del personale amministrativo, è associato all'università.

Tipo: studente=1 docente=2 personale amministrativo=3

Social (ID, nome, valore, *utente_id*)

Viene data totale autonomia all'utente dei canali da mostrare all'interno della piattaforma

Stanza (ID, tipo, nome, descrizione, archiviata, *università_id*)

Tipo: chat diretta/gruppo/feed

Messaggi (ID, contenuto, importante, *stanza_id*, *utente_id*)

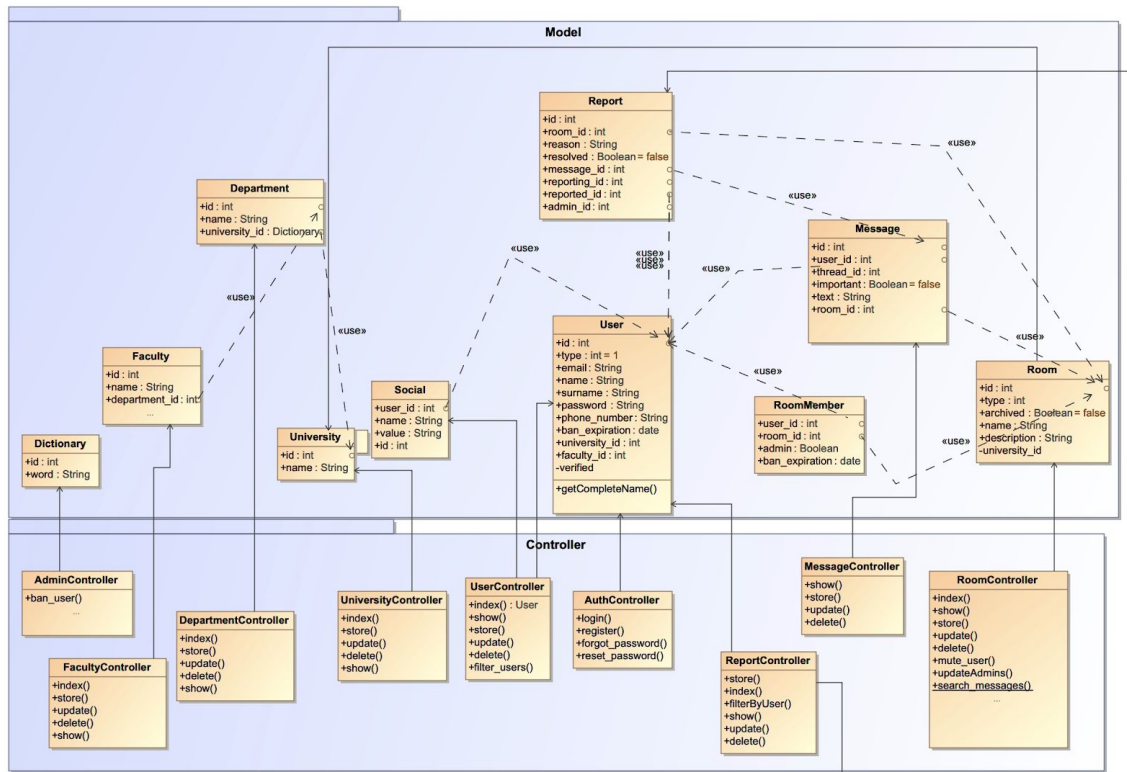
Partecipante(*utente_id*, *stanza_id*)

Segnalazione (ID, motivazione, risolta, *utente_id*, *messaggio_id*)

7. Design Decisions

1. Gestione di chat diretta, gruppi e feed tramite un sistema di gestione a stanze. Questo permette di generalizzare i tre concetti in un'unica entità, riducendo le tempistiche
2. Autenticazione nel sistema tramite credenziali salvate nel database di UniChat, anziché l'utilizzo di servizi esterni come SPID o credenziali universitarie
3. Il ban automatico si basa sul dizionario, poiché è una soluzione efficace e non richiede molto tempo per la sua implementazione
4. La verifica degli utenti è manuale. In questo modo la verifica degli utenti è responsabilità del personale amministrativo, che si presume conosca la struttura universitaria, eliminando le problematiche legate ad una verifica di tipo automatica, che potrebbe verificare utenti malintenzionati.

8. Design di Basso Livello



Il Class Diagram mostra la strutturazione di UniChat, evidenziando un pattern basato su modelli e controller. I modelli si occupano di definire le entità presenti nel sistema, mentre i controller gestiscono le operazioni effettuabili.

Le operazioni principali sono di tipo CRUD: create, read, update, delete, poiché sono le operazioni basilari per il funzionamento del software. Le altre operazioni degne di nota sono quelle presenti nella parte di autenticazione (AuthController) e di gestione del gruppo (mute_user e updateAdmins)

9. Explain how the FRs and the NFRs are satisfied by design

Le chat dirette, i gruppi e i feed sono gestiti tutti e tre come stanze dato che, in fondo, le operazioni che differiscono tra le tre tipologie di conversazione sono minime.

Nella chat diretta i messaggi vengono scambiati direttamente tra due persone; il gruppo invece è una conversazione con più utenti, la differenza sta nel numero di invii e ricezioni; infine il feed è una via di mezzo tra le due dove c'è un unico mittente e molti destinatari.

Affrontiamo adesso un argomento a tratti cruciale per il nostro software, il ban. Abbiamo due tipi di ban, il primo automatico che si limita a silenziare l'utente per un periodo limitato; il secondo invece manuale che prevede un ban a tempo indeterminato.

Il primo ban, da non confondere con il sistema di Antiflood (un sistema per limitare lo spam che annulla l'invio dello stesso messaggio per più di tre volte nell'arco di un minuto), è appunto un ban automatico gestito direttamente dal sistema, questo infatti, mediante l'uso di un dizionario delle parole proibite (scritto da un personale amministrativo o addirittura da uno sviluppatore) scannerizza ogni parola del messaggio da inviare e controlla se è presente uno o più termini presenti nel dizionario.

Appena ne trova uno il sistema non invia il messaggio e lo elimina contrassegnando l'utente; se l'utente viene contrassegnato tre volte viene silenziato per un giorno.

Il secondo ban è manuale, infatti direttamente il personale amministrativo controlla tramite un log quante volte un utente è stato silenziato; al terzo silenziamento si viene bannati dalla piattaforma.

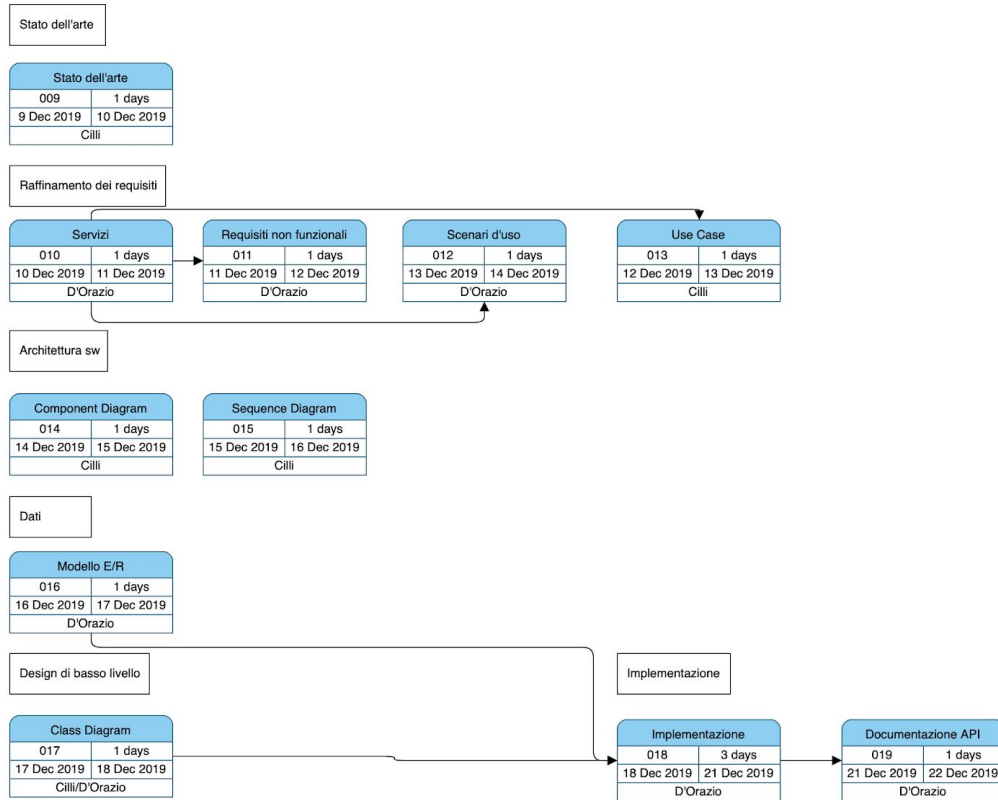
È tuttavia possibile insorgere in qualche errore da parte del personale amministrativo; infatti l'utente può inviare una email così da permettere la correzione da parte del personale amministrativo e venire sbannati.

Gli utenti si possono registrare fornendo nome, cognome, email, password, tipo, università e facoltà. La registrazione viene effettuata autonomamente dall'utente. Tuttavia per registrarsi come docente o personale amministrativo bisogna essere verificati da un altro personale amministrativo fornendo i documenti alla registrazione.

Il personale competente della verifica degli utenti controllerà nell'università in questione e se i documenti risultano l'utente verrà verificato.

10. Effort Recording

10.1 PERT



Nonostante le tempistiche non fossero a nostro favore, ci siamo applicati al massimo per ottenere un software ingegnerizzato nel modo migliore possibile. Siamo soddisfatti di aver rispettato le tempistiche.

10.2 Logging

Personal Journal

Team (number and name): Flop team

Student name: Alessandro D'Orazio, Lorenzo Cilli

Student number: 251811, 253121

Email: alessandro.dorazio2@student.univaq.it, lorenzo.cilli@student.univaq.it

When	Time spent (min)	Partners	Brief Description of the performed task	Category	Sub-Category
11	28	90	A Analisi e ristrutturazione delle specifiche del progetto	Learning	Studio del progetto
11	28	30	AL Analisi progetto	Doing	Studio del progetto
11	29	50	AL Stato dell'arte	Doing	Stato dell'arte
11	30	150	A Raffinamento dei requisiti	Learning	Raffinamento dei requisiti
12	1	30	A Use Case Diagrams	Doing	Raffinamento dei requisiti
12	2	45	L Use Case Diagram	Doing	Raffinamento dei requisiti
12	2	180	L Raffinamento dei requisiti	Learning	Raffinamento dei requisiti
12	3	45	A Studio Sequence Diagram	Learning	Studio individuale
12	3	30	A Studio tool MagicDraw	Learning	Studio software
12	3	150	L Component Diagram	Doing	Architettura software
12	3	45	A Sequence Diagram	Doing	Architettura software
12	4	90	L Studio Component Diagram	Learning	Studio individuale
12	4	45	L Studio Class Diagram	Learning	Studio individuale
12	4	45	L Studio tool MagicDraw	Learning	Studio software
12	4	150	A Modello E/R	Doing	Dati e loro modellazione
12	4	60	L Class Diagram	Doing	Design di basso livello
12	4	90	A Scrittura documento	Doing	Rifiniture finali
12	5	30	A Effort Recording	Doing	Effort Recording
12	5	45	A Rifinitura requisiti	Doing	Raffinamento dei requisiti
12	5		Milestone1		
12	10	90	L Stato dell'arte	Doing	Stato dell'arte
12	11	120	A Servizi	Doing	Raffinamento dei requisiti
12	12	180	A Requisiti non funzionali	Doing	Raffinamento dei requisiti
12	13	180	L Use Case Diagrams	Doing	Raffinamento dei requisiti
12	14	150	A Scenari d'uso dettagliati	Doing	Raffinamento dei requisiti
12	15	270	L Component Diagram	Doing	Architettura software
12	16	120	L Studio Sequence Diagram	Learning	Studio individuale
12	17	300	L Sequence Diagram	Doing	Architettura software
12	17	150	A Modello E/R	Doing	Dati e loro modellazione
12	17	90	AL Class Diagram	Doing	Design di basso livello
12	18	120	A Configurazione progetto	Doing	Implementazione
12	19	105	A Progettazione API e routes	Doing	Implementazione
12	20	180	A Modelli, migrations e factory	Doing	Implementazione
12	21	240	A Controller	Doing	Implementazione
12	21	60	A Studio Postman	Learning	Studio individuale
12	21	120	L Explain how the FRs and the NFRs are satisfied by design	Doing	Raffinamento dei requisiti
12	22	90	A Documentazione API con Postman	Doing	Implementazione
12	22	30	A Appendix Prototype	Doing	Implementazione
12	22	60	A Pert Chart	Doing	Effort Recording
12	22	60	A Studio Pert Chart	Learning	Effort Recording

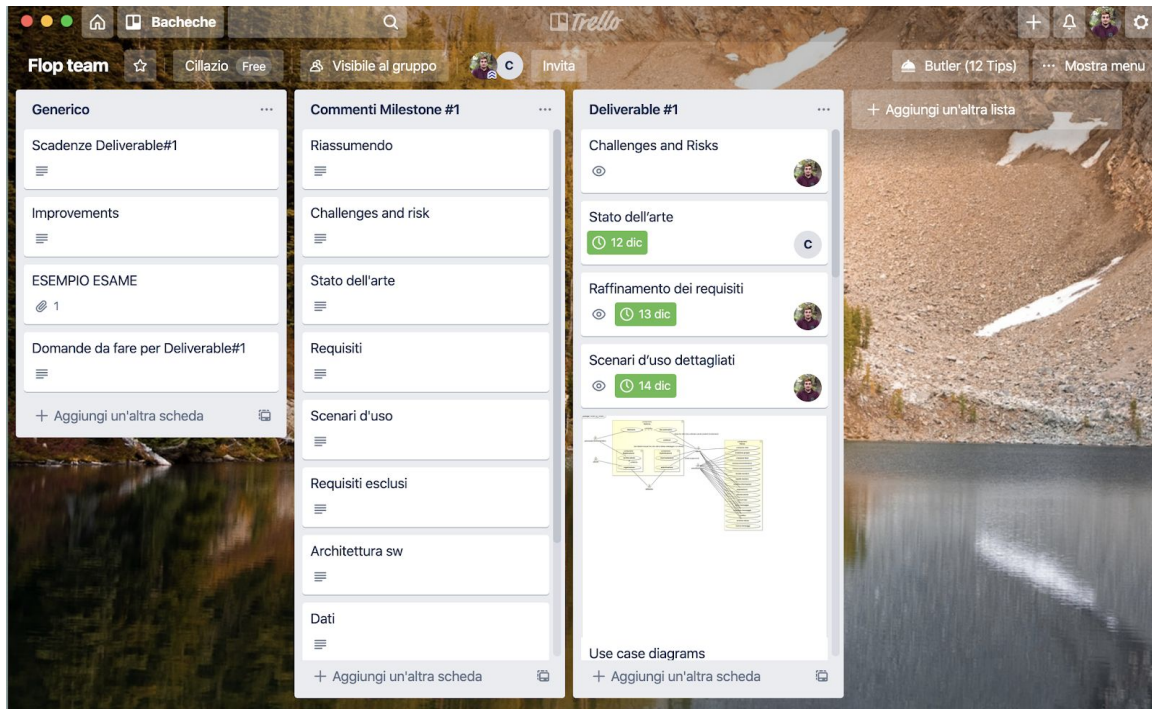
Ore Alessandro	37,5
Ore Lorenzo	28,3
Ore in comune	2,8
Ore totali	71,5

Ore Learning	15
Ore Doing	53

Ore Alessandro: 37.5

Ore Lorenzo: 28

L'organizzazione è avvenuta tramite chiamate Skype, incontri e la gestione di una bacheca su Trello.



11. Appendix. Prototype

Il prototipo, sviluppato con il framework Laravel, si basa su un pattern MVC-like. È presente sia la componente dei modelli che quella dei controller, ma non è stata effettuata un'implementazione delle viste.

Il prototipo segue la definizione dello schema del Class Diagram, includendo una classe PHP (sviluppata da Alessandro D'Orazio in un contesto lavorativo) che gestisce il ritorno dei dati in JSON, ottimizzando la scrittura del codice e le tempistiche grazie al riuso del codice.

Le operazioni permesse sono le CRUD (create, read, update, delete) per le seguenti entità: utenti, università, dipartimenti, facoltà, stanze e messaggi. I social vengono inseriti tramite l'update dell'utente.

La sicurezza viene garantita tramite JSON Web Token con scadenza di 1 ora e possibilità di refresh del token avendo il token precedente. Questo consente di non utilizzare le sessioni, ed offre un'elasticità maggiore rispetto ai metodi tradizionali per la gestione dell'autenticazione.

Quando si invia un messaggio, avviene un controllo dal server che, prima di inserire il messaggio nel db, verifica se l'utente è silenziato, oppure se ha utilizzato una parola vietata.

Le API sono strutturate seguendo le linee guida di Laravel, utilizzando quindi un pattern {url}/api/{entity}/{action}

La fase di testing viene effettuata tramite le factory di Laravel, e una breve sessione di testing con casi limite. Abbiamo provato a caricare centinaia di migliaia di messaggi, contenuti in migliaia di stanze, senza ottenere problematiche di performance.

Nota per la documentazione mediante Postman

Poiché la preferenza del team è di sviluppare un prodotto creando i componenti, testandoli ed estendendoli, alcune funzionalità, come l'implementazione del Web Sockets non sono state introdotte. Riteniamo che la soluzione migliore sia di fornire dei componenti funzionanti al 100%, piuttosto che un componente sviluppato in poco tempo ma non funzionante.

Documentazione mediante Postman

<https://documenter.getpostman.com/view/8518915/SWLYAW1g>