



UNIVERSITÀ DEGLI STUDI DI PALERMO  
DIPARTIMENTO DI INGEGNERIA



# Analisi dei requisiti della suite vivere

Periti informatici  
Antonio Morabito  
Daniele Orazio Susino

Data ultimo aggiornamento  
12 gennaio 2026

# Indice

<b>1</b>	<b>Preambolo</b>	<b>3</b>
1.1	Introduzione . . . . .	3
1.2	Requisiti speciali . . . . .	3
<b>2</b>	<b>Risorse Umane (HR)</b>	<b>4</b>
2.1	Idea . . . . .	4
2.2	Analisi dei requisiti . . . . .	4
2.2.1	La piattaforma . . . . .	4
2.2.2	Gli utenti . . . . .	6
2.2.3	Gli staffer . . . . .	6
2.2.4	Gli amministratori . . . . .	6
2.2.5	I super admin . . . . .	6
2.2.6	Extra . . . . .	7
2.3	Stato . . . . .	7
<b>3</b>	<b>Problemi Tecnici</b>	<b>8</b>
3.1	Idea . . . . .	8
3.2	Analisi dei requisiti . . . . .	8
3.3	Stato . . . . .	8
<b>4</b>	<b>Kaffettino</b>	<b>9</b>
4.1	Idea . . . . .	9
4.2	Analisi dei requisiti . . . . .	9
4.2.1	La piattaforma . . . . .	9
4.2.2	Utenti . . . . .	9
4.2.3	Amministratori . . . . .	10
4.2.4	La piattaforma embedded . . . . .	11
4.2.5	Auditing . . . . .	12
4.2.6	Extra . . . . .	12
4.3	Stato . . . . .	12
<b>5</b>	<b>Magazzino</b>	<b>13</b>
5.1	Idea . . . . .	13
5.2	Analisi dei requisiti . . . . .	13
5.2.1	La piattaforma . . . . .	13
5.2.2	Gli amministratori . . . . .	13
5.2.3	Gli utenti . . . . .	14
5.2.4	Extra . . . . .	14
5.3	Stato . . . . .	14
<b>6</b>	<b>Assistest</b>	<b>15</b>
6.1	Idea . . . . .	15
6.2	Analisi dei requisiti . . . . .	15
6.3	Stato . . . . .	16
<b>7</b>	<b>Drive</b>	<b>17</b>
7.1	Idea . . . . .	17
7.2	Analisi dei requisiti . . . . .	17
7.3	Stato . . . . .	18
<b>8</b>	<b>Eventi</b>	<b>19</b>
8.1	Idea . . . . .	19
8.2	Analisi dei requisiti . . . . .	19
8.3	Stato . . . . .	20
<b>9</b>	<b>Orientamento</b>	<b>21</b>
9.1	Idea . . . . .	21
9.2	Analisi dei requisiti . . . . .	21

9.3	Stato	21
<b>10</b>	<b>Orari</b>	<b>22</b>
10.1	Idea	22
10.2	Analisi dei requisiti	22
10.3	Stato	22
<b>11</b>	<b>Aule Libere</b>	<b>23</b>
11.1	Idea	23
11.2	Analisi dei requisiti	23
11.3	Stato	24
<b>12</b>	<b>QR</b>	<b>25</b>
12.1	Idea	25
12.2	Analisi dei requisiti	25
12.3	Stato	25
<b>13</b>	<b>Calendario</b>	<b>26</b>
13.1	Idea	26
13.2	Analisi dei requisiti	26
13.3	Stato	26
<b>14</b>	<b>Oggetti Smarriti</b>	<b>27</b>
14.1	Idea	27
14.2	Analisi dei requisiti	27
14.3	Stato	28
<b>15</b>	<b>Elezioni</b>	<b>29</b>
15.1	Idea	29
15.2	Analisi dei requisiti	29
15.3	Stato	29
<b>16</b>	<b>Filigrana</b>	<b>30</b>
16.1	Idea	30
16.2	Analisi dei requisiti	30
16.3	Stato	30
<b>17</b>	<b>Segnalazioni</b>	<b>31</b>
17.1	Idea	31
17.2	Analisi dei requisiti	31
17.3	Stato	32

# 1 Preambolo

## 1.1 Introduzione

Questo documento è stato redatto in modo tale da riuscire ad avere una **chiara visione del lavoro che verrà svolto**, in quanto con si è deciso di creare una **suite di applicativi** utili per la vita universitaria sia degli studenti che degli associati.

Ogni **progetto verrà descritto tramite un'idea, i requisiti e lo stato** nella quale vergerà.

Questo **documento potrebbe essere aggiornato in futuro**, in quanto i requisiti dei software potrebbero cambiare inavvertitamente.

## 1.2 Requisiti speciali

Esistono dei **requisiti** che sono **da applicare a tutti i servizi** della suite, verranno elencati di seguito.

Ogni servizio offerto dalla suite Vivere sarà **accessibile esclusivamente tramite un account registrato**. Non sarà possibile accedere da guest.

Per creare gli account **saranno accettate solamente mail UNIPA**, in quanto sono mail che associano univocamente la persona all'account e pertanto permettono di assicurarsi che la persona esista.

È assolutamente obbligatorio che i **servizi siano pensati in favore delle normative europee vigenti in ambito di protezione dei dati e della privacy delle persone (GDPR)**. Questo implica il fatto che **non si possono memorizzare opinioni politiche delle persone** (anche se si tratta di elezioni universitarie). Inoltre, i dati personali memorizzati dovranno essere minimali, ovvero essere memorizzati per uno scopo specifico, i dati che non sono strettamente utili a ciò che viene offerto dovranno essere inseriti in modo volontario e non obbligatorio (ergo non devono essere barriere per quanto riguarda l'accesso ai servizi offerti).

Dovrà essere implementato per ogni servizio un **log-auditing in append e non modificabile, accessibile solo ai "Super Admin"**.

**I log saranno memorizzati per un tempo massimo di x Anni (?), da capire con Antonio questa cosa.**

Non si potranno cancellare e verranno memorizzati direttamente in OS. Saranno di tipo append-only e non modificabili, al fine di garantire l'integrità storica delle operazioni.

Ogni **software** dovrà essere **adeguatamente documentato e commentato**, questo in modo tale da permettere a futuri developer di poter prendere in mano lo sviluppo senza sforzi.

Ogni **software** dovrà **reindirizzare, in caso di mancata autenticazione**, alla piattaforma Risorse Umane.

Bisogna prevedere un modo per fare **visualizzare i changelog al primo login utile** in una qualsiasi piattaforma.

Ogni piattaforma deve prevedere un link per una **pagina di suggerimenti su cosa poter migliorare o su idee che vengono** agli studenti.

## 2 Risorse Umane (HR)

### 2.1 Idea

La piattaforma risorse umane è la **piattaforma di gestione dei profili utente degli associati e degli studenti**. **Tutti** gli studenti e gli associati **dovranno registrarsi tramite questa piattaforma** per poter usufruire della suite Vivere.

### 2.2 Analisi dei requisiti

#### 2.2.1 La piattaforma

La piattaforma dovrà permettere di registrarsi tramite form, specificando:

- **Nome e Cognome**
- **Anno di nascita** (dato necessario per kaffettino)
- **Corso** (dato necessario per il drive)
- **Anno di corso**: da prevedere anche FC, Laureando, Part Time, 4° superiore e 5° superiore, ma deve essere bloccato l'accesso ai minori di 16 anni. (Dato necessario per oggetti smarriti)
- **Numero di telefono**
- **Email**: rigorosamente formato UNIPA, tranne se risulta come studente delle superiori (Dato necessario per autenticazione)
- **Tag Telegram** (opzionale)
- **Tag Instagram** (opzionale)
- **Scuola di provenienza** (opzionale, ma obbligatoria per gli staffer per la piattaforma orientamento)
- **Password**

Le **password** dovranno essere forti e non comuni, per evitare accessi indesiderati alla piattaforma:

- Dovranno essere di **minimo 12 caratteri**
- Deve essere presente **almeno un carattere speciale**
- Deve essere presente **almeno un numero**
- Deve essere presente **almeno una lettera maiuscola**
- Deve essere presente **almeno una lettera minuscola**

La piattaforma e, per estensione tutte le altre, devono prevedere **4 tipologie di utente**:

- **Studente**: Utente semplice senza particolari permessi.
- **Staff**: Utente che è stato promosso da un amministratore a membro di staff (associato), in determinate applicazioni avrà più poteri di uno studente normale, ma non sempre.
- **Amministratore**: Utente che si occupa delle configurazioni dei software della suite, spesso aggiungendo o rimuovendo dati organizzativi o interagendo con i profili degli utenti.

- **Super Admin:** Utente che è già presente nelle migrazioni originali, ha gli stessi permessi dell'amministratore, ma i suoi permessi non possono essere revocati. Saranno gli unici a poter accedere ai log.

Inoltre, esclusivamente agli staffer ed agli amministratori, sarà richiesto inserire:

- Codice Fiscale
- Città, Provincia, stato di nascita

Verrà usata la **mail come autenticazione a due fattori**, obbligatoriamente. Questo perché serve a verificare che l'utente è veramente chi si dichiara di essere (grazie al formato UNIPA).

Alla login implementare rate limiting e invalidazione token, con hasing bcrypt.

Sempre alla login bisogna effettuare il controllo per password comuni e possibilmente attaccabili tramite rainbow table.

Dopo la registrazione di un utente, **l'utente deve essere messo "in attesa"** tramite apposita schermata, visualizzabile anche quando l'utente prova a fare il login durante questa fase.

Dopo che un utente si è registrato, **la piattaforma deve effettuare uno scrape dei professori presenti nell'ateneo ed eventualmente mostrare agli staffer, accanto a quel nome, un avviso se un match è stato trovato.** Questo per evitare che i professori possano accedere ai servizi della piattaforma se non autorizzati da uno staffer.

Quando verrà effettuata una registrazione, verrà mandata una mail agli amministratori dello stesso corso dello studente che ha fatto la registrazione, esortandoli ad accettarlo.

A seguito di un ban, la piattaforma dovrà mandare una mail con la motivazione del ban (banalmente bisogna inserire un form dove si specifica la motivazione).

All'inizio dell'anno accademico gli studenti frequentanti le superiori verranno messi in una sezione "da confermare" dove bisognerà manualmente confermarli per farli passare di anno.

Ogni inizio di anno accademico la piattaforma in automatico incrementerà l'anno accademico delle persone.

In fase di registrazione, la piattaforma verificherà che l'età dello studente sia superiore o uguale a 16 anni. In caso contrario, la registrazione non sarà consentita.

In fase di registrazione, si potrà scegliere se ricevere le email e per quali servizi. Ovviamente con opzione di cambiare preferenza in qualsiasi momento dal proprio profilo utente.

Gli studenti, **appena risulteranno agibili per avere una mail unipa, saranno obbligati a cambiarla** (questo per evitare persone che usano all'infinito il proprio indirizzo mail).

Il sistema dovrà verificare il corretto inserimento di nome e cognome facendo il confronto con la mail istituzionale.

Il sistema dovrà prevedere un **sistema di notifiche**, arriveranno sia in piattaforma che tramite mail.

Ogni notifica può avere dei **documenti associati in allegato.**

Ogni ottobre la piattaforma chiederà agli utenti se le informazioni sul loro anno frequentato sono corrette, in modo tale da mantenere veritiero il DB.

### 2.2.2 Gli utenti

Ogni utente può **resettare password** tramite una mail che gli arriva direttamente in casella, previa conferma dell'indirizzo email oscurato.

Gli utenti che si registreranno visualizzeranno una **schermata di attesa nel frattempo che il loro account verrà accettato**, siccome gli amministratori dovranno verificare l'iscrizione per renderla valida.

Il nome visualizzato e l'**username** per fare l'accesso sarà in **formato unipa** (es: MarioLuigi.Rossi03)

In qualsiasi momento un utente può decidere di **disiscriversi da eventuali mailing list** oppure anche dalla piattaforma.

Gli utenti potranno ricevere svariate notifiche dai sistemi Vivere, consultabili dall'apposita **sezione notifiche sul proprio profilo**.

Quando un utente verrà promosso a gestore auletta, gli dovrà arrivare una mail dove lo si esorta a venire nella sua auletta di riferimento a prendere le chiavi. Inoltre, quando verrà rimosso come gestore, verrà esortato a venire in auletta per restituire le chiavi.

Nel caso una **persona bannata** provi a **rifarsi l'account tramite la seconda mail istituzionale**, il sistema deve **bloccare in automatico la richiesta**, riconoscendo giustamente che è la stessa persona (grazie al formato UNIPA).

### 2.2.3 Gli staffer

Un qualsiasi staffer può **accettare una qualsiasi richiesta di registrazione**.

Ogni utente non studente, **può essere associato ad una o più alette come gestore auletta**. Egli riceverà diverse comunicazioni in merito alla gestione delle alette a lui assegnate.

### 2.2.4 Gli amministratori

Gli amministratori potranno **cambiare a piacimento il ruolo degli utenti**.

Deve essere previsto **almeno un amministratore per ogni corso**, questo implica che una persona potrà essere **amministratore anche di più corsi**, siccome potrebbe capitare che prenda in gestione un corso simile dove ancora non ci sono associati.

Dunque, **una persona può far parte di più staff**, questo per evitare che ci siano corsi senza staff e che quindi si blocchi l'accesso ai servizi vivere per quel determinato corso.

Gli amministratori potranno modificare e, in caso di comportamenti scorretti, bloccare gli utenti.

Gli amministratori potranno **aggiungere e rimuovere gestori auletta a piacimento**.

Gli amministratori potranno **specificare se un determinato utente possiede un ruolo istituzionale** (ad esempio CDD). Ovviamente una persona può ricoprire **più ruoli istituzionali**.

Gli amministratori potranno **mandare notifiche** in broadcast.

### 2.2.5 I super admin

Il super admin è un utente che non fa manutenzione regolarmente delle piattaforme, ma viene **chiamato solamente per questioni di GDPR compliance, log auditing, ecc...**

### 2.2.6 Extra

Magari si potrebbe fare in modo che il **sito web prenda i ccs da questo DB**? Ad ogni modo, per quanto riguarda i ruoli istituzionali, il sistema deve **prevedere la data nella quale i rappresentanti decadono e di conseguenza archiviare il periodo di attività di rappresentanza se non rinnovato**.

## 2.3 Stato

In implementazione

## 3 Problemi Tecnici

### 3.1 Idea

La piattaforma Problemi Tecnici deve **permettere agli utenti di segnalare un problema tecnico** ai manutentori delle applicazioni.

### 3.2 Analisi dei requisiti

#### Gli utenti

Potranno segnalare dei problemi tecnici in un form, specificando:

- Titolo problema
- Descrizione problema
- Foto (anche multiple) o video del problema
- Applicazione del problema (Kaffettino, Eventi, ecc...)

Chiaramente la piattaforma dovrà inviare anche l'email dell'utente che è stato a segnalare in modo che sia poi ricontattabile. Da lì in poi si procederà per mail.

#### Extra

Sarebbe carino e più professionale, **trasformarla in una piattaforma di ticketing vera e propria**, magari integrando nel profilo utente tra le notifiche gli sviluppi. Sarebbe anche bello poter avere una chat asincrona direttamente in app e senza scambio di email se non per la notifica iniziale al gestore. Inoltre una sezione per vedere le problematiche segnalate ed il loro stato sarebbe interessante da prevedere.

### 3.3 Stato

In ideazione

## 4 Kaffettino

### 4.1 Idea

Piattaforma che permette di **organizzare l’inventario e la vendita di prodotti nelle alette** agli associati.

### 4.2 Analisi dei requisiti

#### 4.2.1 La piattaforma

Nella pagina principale della piattaforma bisognerà **allegare una foto ”buongiorno kaffè!” diversa ad ogni caricamento**.

Le immagini devono essere, in ordine di priorità:

- **Giorno del compleanno**
- **Debito**
- **Festività** (Natale, Santa Lucia, Pasqua, Pasquetta, Capodanno, San Silvestro, Halloween, Morti, Epifania, Santo Stefano, Festa della donna, Festa della repubblica, Pride, Anniversario Ingegneria ed altre associazioni, Capodanno, Black Friday, Domenica delle palme, festa della liberazione, Festa della mamma, Festa del papà, Festa dei nonni, Compleanno di rosone, San francesco, San Valentino, Ferragosto, Giorno della memoria, ecc ...)
- **Giorno della settimana** (Lunedì, martedì, mercoledì, ...)

La piattaforma dovrà **segnalare** agli amministratori **quando le unità del magazzino scendono sotto la soglia**, consigliando di fare rifornimento.

All’inizio di ogni settimana, la piattaforma **invierà una mail** agli utenti indebitati, **esortando di saldare i propri debiti** a prescindere dalla quantità (anche se sei di un centesimo in debito in sostanza).

Il **debito vale per singolo conto**, e non si potrà comprare nessun prodotto se pagando si sfora il debito massimo.

Ogni **prodotto** dovrà avere i **dati del fornitore**: Nome, Indirizzo, N Telefono (opzionale), Email (opzionale).

I **coupon** (codici sconto) **non saranno cumulabili** e saranno validi solamente per i prodotti scelti dagli amministratori, avranno inoltre una chiara data di scadenza.

È previsto che la **card dell’auletta possa avere un PIN**, a discrezione di ogni auletta, in modo tale da evitare che le persone la usino per badgare ingiustamente.

#### 4.2.2 Utenti

Ogni **utente** (studente, staff, admin) potrà **effettuare la registrazione** al servizio ma per avere il **conto abilitato dovrà avere una carta associata**.

Qualora un **utente prova a fare l’accesso al servizio senza una carta associata**, allora deve **apparire il messaggio** ”Impossibile proseguire siccome non hai ancora una carta associata! Per favore vieni in auletta per avere la card”.

Gli utenti potranno **richiedere una card sostitutiva in caso di smarrimento**.

Quando la **card verrà assegnata** riceveranno una **mail** che conferma che l'operazione è andata a **buon fine**.

Ogni utente potrà **visualizzare il proprio saldo**.

Ogni utente potrà **visualizzare il proprio storico** (data ed ora, importo, auletta di consumazione), spesa totale (oggi, settimana, mese, anno), spesa media (giornaliera, settimanale).

Inoltre implementare il **confronto con lo storico (questo mese +12% rispetto alla media) e previsioni**.

Implementare un **istogramma con tipologie di prodotti acquistati e quantità**.

Includere anche **statistiche anonime** ("bevi più caffè del 60% degli studenti).

Ovviamente per partecipare alle statistiche bisogna dare il **consenso**.

**Caffè gratis offerto al compleanno**, con tanto di **musichetta e messaggino**.

Il giorno del **compleanno** dovrà essere notificata la persona via **mail** facendogli gli auguri e **spronandola a venire in auletta** e riscattare il caffè gratuito.

Un utente potrà avere **più conti**, il saldo **fa riferimento al magazzino e non alle alette** perché così manteniamo le alette del secondo, terzo piano e del DEIM come afferenti ad ingegneria. Il **saldo** dell'utente è quindi **spendibile** solamente nelle **alette con magazzino in comune** afferente all'auletta dove è stata ricaricata la card. Quindi per esempio, il saldo ricaricato ad ingegneria sarà spendibile al terzo piano, secondo piano e DEIM, mentre il saldo ad economia sarà spendibile solo ad economia.

Qualora **durante l'acquisto si supera il debito massimo**, l'acquisto non va a buon fine **segnalando il saldo insufficiente**.

Gli utenti potranno **abilitare i coupon sconto** sul prossimo acquisto se lo abiliteranno dal portale. Gli utenti potranno pagare anche tramite la piattaforma, selezionando il prodotto che intendono acquistare e l'auletta (magari l'auletta selezionata automaticamente in base alla posizione).

#### 4.2.3 Amministratori

Gli amministratori potranno **accedere alla dashboard dei resoconti e visualizzare i resoconti delle vendite e delle ricariche** filtrando per giorno, settimana, mese, anno.

I resoconti potranno essere **filtrati** come globali, per singole alette, per utente.

I **resoconti di una determinata auletta** dovranno essere **accessibili** solo da **amministratori** di un **corso afferente a quell'auletta**.

Tutti i resoconti saranno **scaricabili come Excel**.

Gli amministratori, tramite il portale, potranno **ricaricare i conti degli utenti**, ma **non potranno ricaricare da soli il proprio conto**, ci dovrà essere un **secondo amministratore che lo carichi**.

Potranno anche **assegnare e modificare la tessera** di un utente.

Potranno **aggiungere prodotti nel magazzino** (con l'opzione di aggiungere prodotti non presenti nel magazzino ma presenti in altri magazzini, o crearne di nuovi).

Una volta che i **prodotti vengono creati non possono essere cancellati**, siccome per storicità bisogna mantenere le compravendite degli utenti.

Per rimuovere un prodotto dalla vendita, infatti, **si potrà semplicemente delistare**, ma continuerà ad esistere nella base di dati.

Gli amministratori potranno anche **modificare i nomi ed i costi dei prodotti**, ma non si potrà cancellare.

Potranno anche **aggiungere, modificare e delistare alette**.

Potranno anche **modificare il debito massimo di ogni auletta** (default  $3 \times$  valore del caffè).

Gli amministratori potranno **modificare la soglia di avviso per ogni prodotto**, in modo tale che il sistema una volta raggiunta quella soglia mandi una mail ai gestori auletta.

Gli amministratori potranno **gestire dei coupon per dare una percentuale di sconto agli acquisti**.

Gli amministratori potranno visualizzare le **statistiche del proprio magazzino**, con trend di consumo, previsioni di guadagni e di consumi ecc...

Gli amministratori potranno anche **impostare e recuperare il pin della card dell'auletta** nell'eventualità non vogliano che si usi per badgare .

Gli amministratori, dall'app web, potranno **offrire degli item**, in modo tale che contino come item scalati dal magazzino ma non come soldi entrati dall'auletta.

#### 4.2.4 La piattaforma embedded

È necessaria la **creazione e l'utilizzo di una piattaforma embedded** in quanto essa verrà predisposta in ogni locale dell'associazione per effettuare il badging dei prodotti consumati.

La piattaforma embedded avrà come componenti minimi:

- Microcontrollore: Idealmente un ESP32
- Sensore NFC
- Keypad
- Schermo Oled
- 3 LED (Rosso, Giallo, Verde)
- Cassa
- Lettore MP3
- Buzzer

Dovrà essere creata una PCB ad hoc, per permettere ai collegamenti elettronici di operare senza interferenze e nelle condizioni ideali.

**Fare BOM (Bill Of Materials) di tutto quanto quello che sarà servito per creare la scheda PCB**

L'embedded dovrà prevedere, inserendo un apposito codice per sbloccare la modalità amministratore, un modo per **permettere di cambiare account del wifi per collegarsi al WIFI dell'ateneo**. Questo per evitare che debba venire in loco il tecnico per resettare la connessione.

Dovrà prevedere un modo per **cambiare l'auletta di appartenenza** (questo per fare in modo di poter spostare eventualmente l'embedded).

Dovrà **scaricare i prodotti del magazzino alla quale fa afferenza**, e dovrà prevedere un modo per **ricaricare la lista di prodotti senza accendere e spegnere**.

Dovrà **riprodurre** la canzone di **buon compleanno** al primo badge del compleanno di un utente, concedendogli un caffè gratis.

In caso di acquisto, dovrebbe mostrare: "Grazie NomeUtente!".

Deve prevedere la **mascotte Coffy** che cambia espressione in base a quello che succede a schermo.

Dovrà **visualizzare i messaggi di errori specifici** del backend (Nessuna connessione, prodotto non esistente, saldo insufficiente)

In caso di **errore critico**, visualizzare una schermata di fallback con Coffy triste e con scritta di **chiamare un amministratore** (pensare a sensori non disponibili, errori critici di memoria).

#### 4.2.5 Auditing

**Tutte le azioni devono essere loggate**, in particolare le azioni amministrative:

- Ricariche
- Modifiche prezzi
- Assegnazioni card
- Sconti
- ETC ...

#### 4.2.6 Extra

L'embedded dovrà **prevedere un modo per sbloccare un dispensatore di cialde**.

Dovrebbe prevedere delle **animazioni durante il caricamento**.

È complicato ma sarebbe interessante prevedere un **comportamento offline in modo da gestire gli acquisti in assenza di connessione**, mettendo tutto in una coda di transazioni. Questo sarebbe utile per quei giorni dove la connessione UNIPA è congestionata, si potrebbe memorizzare tutto in memoria flash?

Sarebbe ideale **predisporre la piattaforma per accettare in futuro anche pagamenti tramite carta di credito** per ricaricare il saldo.

### 4.3 Stato

In sviluppo, i componenti sono stati acquistati e si sta procedendo a creare la PCB ed il design finale.

## 5 Magazzino

### 5.1 Idea

La piattaforma magazzino deve permettere di rendicontare tutti gli item comprati, in uso ed utilizzati.

### 5.2 Analisi dei requisiti

#### 5.2.1 La piattaforma

La piattaforma dovrà differenziare tra gli item in magazzino e quelli in auletta. Questo sarà possibile tramite la scansione del codice a barre degli item.

In origine, tutti gli item saranno inseriti come in magazzino e ogni volta che verranno impegnati in auletta verranno rimossi dai beni in magazzino e verranno inseriti nei beni in auletta.

Si dovranno poter allegare dei documenti (scontrini, fatture) per ogni ordine effettuato (da capire bene, come ordiniamo? Questa parte è molto pratica, da parlarne con Depa)

#### 5.2.2 Gli amministratori

Gli amministratori potranno creare, modificare e delistare degli item (Giornalini, Magliette, Caffè, ...) in magazzino, specificando se esistono diverse versioni per lo stesso item.

Gli amministratori potranno creare e modificare i tag, ad esclusione delle magliette che saranno embedded nell'app.

Quindi durante la creazione dovranno inserire:

- Nome item
- Costo unitario
- Versioni (Pensare a branding ingegneria, ateneo, economia...)
- Foto (con possibilità di inserirla per ogni versione diversa)
- Quantità in magazzino
- Quantità riservate
- Punto di ritiro (con possibilità di inserirla per ogni versione diversa)
- Se l'item è in preorder (questo implica il fatto che la quantità in magazzino possa essere uguale a 0)
- Quantità massima per persona di un item per tipo (per evitare che una persona prenoti 27 magliette uguali)
- Tag di tipologia per filtro (vestiario, cibo) e per opzioni uniche (taglia per le magliette)
- Codice a barre per ogni variante

Gli amministratori possono decidere di dare in comodato d'uso un item bloccato ad un determinato utente, specificando anche una data ultima di riconsegna se necessario, il sistema manderà in automatico notifica e mail all'utente.

Gli amministratori potranno visualizzare tutti i beni dati in comodato d'uso, a chi sono stati dati e per quanto tempo.

Gli amministratori potranno mandare notifiche ed email automatiche alle persone che non avranno ancora dato indietro l'item prestato.

Gli amministratori possono modificare il numero di item bloccati, nel caso vengano smarriti o semplicemente si rompono.

Gli amministratori potranno impostare una prenotazione come pagata quando riceveranno i soldi fisicamente.

Gli amministratori potranno ovviamente guardare le statistiche di magazzino, quanti item vengono consumati giornalmente, settimanalmente, mensilmente.

Potranno anche vedere i trend e le previsioni.

### **5.2.3 Gli utenti**

Gli utenti potranno prenotare un qualsiasi item ed anche diverse versioni dello stesso item, rispettando la quantità massima per persona.

Ogni item verrà inserito nel carrello e quando l'utente sarà pronto potrà prenotare tutti gli item.

L'utente potrà vedere una sezione apposita con tutte le sue prenotazioni e quindi sapere dove andare a ritirare i suoi oggetti.

L'utente potrà vedere se ha beni in comodato d'uso ed il relativo storico di tali beni.

### **5.2.4 Extra**

Sarebbe interessante implementare una piattaforma embedded. Magari un semplice raspberry pi, uno schermo ed una pistola scansora codice a barre per semplificare la vita nel uscire molti item dal magazzino.

## **5.3 Stato**

In ideazione

## 6 Assistest

### 6.1 Idea

Piattaforma che permette alle aspiranti matricole di partecipare a simulazioni di prove TOLC e TOL. Look molto stile gamification

### 6.2 Analisi dei requisiti

#### Gli studenti

**Importante** Come già detto in vivere HR, gli studenti non potranno registrarsi se non hanno almeno 16 anni.

La matricola dovrà avere un'account valido (ruolo studente).

Potranno partecipare a due tipi di simulazioni:

- In tempo reale con l'istruttore (Stile Kahoot)
- Nel tempo libero (stile quiz patente)

Potranno Partecipare alle simulazioni in tempo reale tramite codice alfanumerico di 6 caratteri a gruppi da 3. Potranno effettuare le simulazioni private scegliendo i panetti o i pacchetti di panetti alla quale sono interessati tutte le volte che vogliono, ricevendo alla fine il summary delle risposte date.

Qualora si disconnettano durante una simulazione in tempo reale, allora verrà mantenuto lo score e si potranno riconnettere. Avranno la possibilità inoltre di scegliere un nickname per la singola partita.

Potranno vedere i propri risultati per ogni simulazione fatta, con relative statistiche (media dei risultati, picco minimo e massimo) relative a tutte le simulazioni fatte e trend giornalieri, settimanali, mensili. Potranno visualizzare anche l'analisi delle aree di miglioramento (es. "devi migliorare in fisica")

Potranno segnalare la risposta ad una domanda in modo tale che poi venga sistemata.

#### L'istruttore

Qualsiasi membro di staff potrà fare l'istruttore.

Potrà creare, modificare e cancellare panetti. Potrà creare, modificare e cancellare pacchetti di panetti.

Un panetto è una raccolta di domande relative a un singolo argomento (ad esempio, 'Fisica', 'Matematica', ecc.), mentre un pacchetto di panetti consiste in più panetti raggruppati per argomenti o macrotemi (ad esempio, 'Fisica - Meccanica', 'Fisica - Termodinamica', ecc.). Ogni domanda dovrà prevedere anche una relativa risposta.

Potrà creare, modificare e cancellare le simulazioni. Visualizzare un riepilogo dei risultati degli studenti per ogni simulazione. Visualizzare i risultati dei singoli studenti per ogni simulazione. Visualizzare i risultati dei singoli studenti per ogni simulazione priva degli studenti.

Durante la creazione delle simulazioni, si dovranno scegliere i panetti di domande da inserire (o pacchetti di panetti). Ad esempio si potrà scegliere il panetto di panetti completo di una materia (ES. "Fisica"), oppure panetti su macro argomenti divisi per materia (Es. "Fluidodinamica", "Dinamica", "Termodinamica", "Cinematica", ecc...)

Qualora lo studente volesse contattare l'istruttore, la piattaforma dovrà prevedere un modo per mostrare il tag telegram/instagram o il contatto whatsapp dell'istruttore.

Una volta finito l'assistest, lo studente verrà caricato in automatico sulla piattaforma HR sulla sezione matricole.

Ogni panetto è composto da domande a risposta multipla, vero falso ed aperta. Le domande aperte saranno usate esclusivamente nel caso di conti numerici da fare (attenzione a non essere case sensitive ed a specificare bene il formato), ovviamente con un certo grado di tolleranza ai calcoli effettuati.

Gli istruttori potranno visualizzare sia i risultati generali di un'esercitazione, sia filtrandoli per utente, dando feedback direttamente tramite la piattaforma ad utenti che ne hanno bisogno, specificando la singola domanda.

Gli istruttori potranno visualizzare le domande che sono state segnalate e potranno modificarle.

Potranno dare il via alla simulazione stile kahoot manualmente, in modo tale da assicurarsi che tutti gli studenti siano entrati.

### **La piattaforma**

Imporrà un limite di tempo per le simulazioni real time, ma non imporrà limiti per quelle private.

Il codice alfanumerico per accedere alle lobby verrà generato sul momento, ovvero quando l'istruttore creerà la lobby.

### **Extra**

Magari aggiungere delle streak stile duolingo che permettono di ottenere qualche bonus? Implementare sistema di punti in modo tale da poter fare una gara ed elargire premi al migliore? Si potrebbe implementare una leaderboard dove gli studenti possono vedere i progressi rispetto agli altri. La piattaforma sbloccherà contenuti multimediali in base ad una roadmap specificata dagli istruttori.

## **6.3 Stato**

In ideazione

## 7 Drive

### 7.1 Idea

Utility che permette agli studenti di condividere appunti e materiale di studio.

### 7.2 Analisi dei requisiti

Tutti gli studenti potranno accedere alla piattaforma se sono stati verificati nel db.

Gli studenti potranno visualizzare, scaricare e chiedere l'approvazione degli appunti.

Questo vuol dire che i membri di staff dovranno controllare il materiale mandato dai ragazzi, potranno accettarlo o eliminarlo sempre con un commento costruttivo per fare capire agli studenti come contribuire alla comunità.

I membri di staff potranno visualizzare, scaricare, aggiungere e cancellare gli appunti condivisi.

Ogni file sarà identificato con:

- Nome del file
- Nome dell'autore (possibilità di lasciarlo anonimo)
- Data di caricamento

La piattaforma dovrà mantenere un log di elementi caricati, modificati e rimossi.

La pagina iniziale dovrà permettere di selezionare qualsiasi macroarea (Ingegneria, Medicina, ecc...), dopo di che si potrà selezionare il corso, successivamente le materie (mettendo in una cartella a parte le materie non più erogate), subito dopo l'anno di erogazione ed infine la docenza.

Ogni elemento dovrà avere dei tag che permetteranno di filtrare.

I tag in questione indicheranno la tipologia: Appunti, Esercitazioni, Esercizi, Video (possibilmente anche link esterni tipo youtube), slide, documentazioni, etc ...

Oltre ad un tag di tipologia ci saranno anche i tag di "ufficialità": Materiale del docente, materiale approvato dal docente, materiale non approvato dal docente.

Inoltre ogni elemento avrà un ranking in termini di punteggio (stelle) dato dagli studenti, in questo modo gli studenti potranno sapere se in un determinato blocco di appunti manca qualcosa o ci sono degli errori, oppure se è ancora pertinente per poter preparare la materia.

Insieme al punteggio, gli utenti potranno lasciare dei commenti e questi verranno moderati dagli amministratori, cancellandoli o approvandoli.

Se dalla pubblicazione del commento passerà più di una settimana, il commento verrà approvato automaticamente.

I file si potranno anche sostituire, in modo tale da aggiornare le criticità.

Per questo bisogna anche tenere traccia dei cambiamenti (stile git con i commenti dei commit).

Gli utenti potranno anche flaggare dei commenti come offensivi, ed in quel caso potrebbero essere prese azioni disciplinari contro l'offendente.

## REGOLE

Alcune regole per tutelare la piattaforma devono essere messe in atto, infatti chi trasgredirà questo regolamento potrebbe venire bannato da altri amministratori.

- Non sarà permesso caricare video-lezioni dei docenti e registrazioni senza consenso esplicito con email formale inoltrata ai gestori.
- Gli appunti non si potranno caricare come file word ma solo come PDF filigranato (vedesi utility di filigrana)
- Le dispense dei professori non si dovranno filigranare.
- Si dovrà promuovere un contesto di collaborazione ed ogni forma di hate-speech sarà punita con il ban.

## EXTRA

Sarebbe interessante implementare un sistema di moderazione AI dei commenti. Sarebbe interessante prevedere una barra di ricerca che permetta di navigare il file system più velocemente. Potrebbe essere carino incentivare il submitting di materiale magari dando accesso a sconti tramite kaffettino? O comunque dei premi / badge. In questo caso ogni studente potrebbe avere il proprio profilo e quindi li potremmo fare visualizzare i suoi contributi.

## 7.3 Stato

In ideazione

## 8 Eventi

### 8.1 Idea

La piattaforma eventi permetterà di comunicare la propria partecipazione a degli eventi specifici interni dell'associazione e di creare dei sondaggi.

### 8.2 Analisi dei requisiti

Ci saranno due tipi di sondaggi:

- Sondaggi "When To Meet": La piattaforma dovrà prevedere dei sondaggi per scegliere la data e l'orario di un evento. Pensare a <https://timeful.app>.
- Sondaggi "What About": Sondaggi dove si chiederanno una serie di domande, pensare a <https://luma.com/?locale=it>

I sondaggi potranno essere marchiati come "solo per staff" ed in quel caso potranno essere visualizzabili solo a membri di staff.

Ogni evento può avere dei **documenti associati in allegato**.

#### **Gli amministratori**

Potranno creare modificare o annullare eventi.

Potranno visualizzare le risposte ai sondaggi.

Potranno decidere se in automatico aggiungere al calendario dell'intervistato l'evento tramite la piattaforma calendario.

Potranno convocare ufficialmente alle assemblee ed ai vari eventi gli utenti, mandando comunicazione per notifica ed email. Sarebbe interessante anche per riunioni dei corsi ecc

#### **Gli utenti**

Potranno rispondere ai sondaggi, vedere i sondaggi che hanno compilato e vedere le loro risposte. Inoltre potranno visualizzare i sondaggi alla quale possono rispondere.

Gli utenti potranno inserire una nota opzionale quando confermano o meno la presenza ("Porto i panettoni", "Vengo 10 min in ritardo")

Potranno vedere anche le statistiche su eventi alla quale si è partecipato (invitato in x eventi, partecipato in y eventi)

#### **Da vedere con di maio**

Statistiche. Recensioni / Feedback. Collaboratori. Invitare ex partecipanti altri eventi Reminder mail. Posizioni. Esportare in csv/excel.

Confermare le entrate ed i pagamenti. Poter allegare file all'evento (resoconti) o mandare file a Tutti poter decidere quando e se re inviare notifiche a chi non ha dato conferma di partecipazione ad una cosa "tener segnato sia per l'associato che per chi gestisce chi deve pagare cose come magliette ed ecc"

### 8.3 Stato

In ideazione

## 9 Orientamento

### 9.1 Idea

Piattaforma che dovrà mantenere i contatti dei responsabili orientamenti delle scuole ed aiutare nell'organizzare gli orientamenti.

### 9.2 Analisi dei requisiti

#### **Gli amministratori**

Potranno aggiungere scuole al database. Potranno aggiungere i responsabili degli orientamenti delle scuole.

#### **Gli staffer**

Potranno farsi generare una mail da mandare ai responsabili delle scuole.

Potranno creare, modificare o annullare gli orientamenti.

Potranno dare disponibilità verso gli orientamenti.

Gli potrà arrivare una mail in caso si crei un orientamento alla propria scuola di provenienza.

#### **Altro**

La piattaforma dovrà integrarsi con il calendario per mettere tutti gli orientamenti in calendario.

### 9.3 Stato

In ideazione

## 10 Orari

### 10.1 Idea

La piattaforma degli orari permetterà di scaricare l'orario di un corso.

### 10.2 Analisi dei requisiti

#### **Gli utenti**

Potranno utilizzare questa utility purché sia ovviamente registrato e verificato.

Dovranno semplicemente selezionare un corso di studi e l'anno di frequenza, dopodiché il download del file immagine verrà avviato in automatico.

Potranno anche selezionare delle materie a scelta di altri corsi di laurea.

Potranno esportare l'orario in modo tale da poterlo inserire nel proprio calendario.

#### **Gli amministratori**

Potranno caricare dei layout e delle palette da utilizzare per ogni macroarea ("Ingegneria", "Medicina", "Economia", ecc...).

#### **La piattaforma**

Dovrà prendere il layout specificato dagli amministratori ed inserire al suo interno le informazioni del corso facendo uno scraping dal sito di UNIPA con gli orari effettivi. L'orario dovrà prevedere anche le date di lezione, le date di sessione e le sospensioni.

### 10.3 Stato

In ideazione

## 11 Aule Libere

### 11.1 Idea

La piattaforma delle aule libere servirà per visualizzare le aule libere dell'ateneo in una determinata data ed ora, oppure anche in un range di date ed ore.

### 11.2 Analisi dei requisiti

#### La piattaforma

Dovrà essere un wrapper per <https://offertaformativa.unipa.it/offweb/public/aula/aulaCalendar.seam>, ma offrendo qualche funzionalità in più. La piattaforma dovrà permettere di selezionare:

- Data (o range di date) nella quale si vuole prenotare l'aula
- Fascia oraria nella quale si vuole prenotare l'aula
- Quantità di posti necessari
- Tipologia di aula (Didattica, Didattica innovativa, Informatica, Impianto sportivo, Laboratorio, Sala Interna)
- Edificio
- Dipartimento
- Scuole (Formazione insegnanti, Scuola di medicina e chirurgia)
- Attributi dell'aula (Provvista di LIM, Provvista di lavagna, provvista di amplificazione, provvista di postazioni computer, in ristrutturazione)
- Se provvista di postazioni computer

Se verrà selezionata l'opzione "provvista di postazioni computer", allora si dovrà poter selezionare anche la quantità di postazioni necessaria

Lo scaricamento iniziale delle aule e dei loro attributi avverrà dal sito dell'università.

La piattaforma dovrà poter lasciare modificare agli amministratori le aule.

Questo vorrà dire:

- Creare, modificare e cancellare aule esistenti
- Aggiungere, modificare e rimuovere gli attributi dell'aula

Una volta effettuata la ricerca, la piattaforma dovrà mostrare la lista di aule da poter selezionare. Le aule in ristrutturazione si dovranno visualizzare ma dovranno spuntare come non prenotabili.

Una volta selezionata un aula, verrà visualizzata nella propria pagina:

- La posizione su una mappa leaflet.
- Gli orari nella quale l'aula è impegnata

Inoltre, la piattaforma, dovrà tenere traccia delle personalità dell'ateneo che si occupano di prenotare le aule nei vari dipartimenti. Gli amministratori, dovranno poter modificare queste personalità ed il loro dipartimenti di riferimento. Più personalità possono avere a che fare con un singolo dipartimento. La piattaforma quindi, dovrà preparare una mail da mandare alla personalità di riferimento afferente a quell'aula. La funzionalità della preparazione delle mail dovrà essere utilizzabile solo da membri di staff o amministratori.

### **11.3 Stato**

In ideazione

## 12 QR

### 12.1 Idea

La piattaforma dei QR sarà esclusiva agli staffer e dovrà permettere di creare QR in diversi modi.

### 12.2 Analisi dei requisiti

#### **Gli staffer**

Potranno inserire un link o un testo generico che verrà trasformato in un QR code.

Il QR potrà essere modificato in base a:

- Colore
- Forma dei punti
- Formato (SVG, PNG, JPEG)
- Dimensione (Se il formato non è SVG)
- Logo interno (Ci saranno tutti quelli delle sotto-associazioni)

### 12.3 Stato

In ideazione

## 13 Calendario

### 13.1 Idea

La piattaforma calendario è pensata per un uso principalmente interno, a differenza della piattaforma eventi. Permetterà di sapere quali sono gli eventi in associazione e di mostrarli in calendario.

### 13.2 Analisi dei requisiti

#### Gli amministratori

Potranno aggiungere, modificare, cancellare gli inviti agli eventi. Un evento sarà caratterizzato da:

- Nome
- Giorno ed orario
- Tag ("CDD", "CCS", "Tutti", "Riunione di staff", ETC...).
- Staff di appartenenza ("Ingegneria", "Economia", "Ateneo"...)
- Priorità (?) probabilmente verrà abusata dagli amministratori

#### Gli utenti

Potranno registrarsi o cancellarsi dalle varie categorie di eventi. Quindi gli utenti potranno gestire gli eventi come meglio credono e potranno filtrare le tipologie di eventi che gli arrivano, ad esclusione di quelli dettati dalla propria carica istituzionale.

#### La Piattaforma

Dovrà spedire gli inviti via mail ed essi dovranno essere compatibili con calendari google, outlook.

#### La piattaforma embedded

Sarà una semplice dashboard di visualizzazione calendario da mettere in auletta, si potrebbe tranquillamente fare con una vecchia raspberry pi ed uno schermo. Ovviamente questa sarebbe aggiornata in tempo reale con sync dalla piattaforma web

#### EXTRA

Sarebbe interessante cercare di capire se si possa automatizzare l'invio in base a ciò che già ci viene spedito (vedesi CDD).

### 13.3 Stato

In ideazione

## 14 Oggetti Smarriti

### 14.1 Idea

La piattaforma degli oggetti smarriti permetterà agli utenti di reclamare degli oggetti smarriti come propri.

### 14.2 Analisi dei requisiti

#### La piattaforma

La piattaforma si dovrà interfacciare con quella degli orari, in modo tale da dare un consiglio per quanto riguarda i corsi e la relativa annata nella quale fare girare l'annuncio.

A proposito di annuncio, deve preparare un copy per quanto riguarda il mandare la comunicazione nei gruppi dei singoli anni.

Dovrà arrivare una mail ai ragazzi se l'oggetto smarrito è possibile che sia stato perso esclusivamente nel loro corso di appartenenza, questo per evitare email spam.

#### Gli amministratori

Gli amministratori potranno inserire, modificare e cestinare un oggetto smarrito.

Per ogni oggetto smarrito bisognerà specificare:

- Una foto dell'oggetto
- Una descrizione, idealmente con una caratteristica unica dell'oggetto
- Edificio ed aula in cui è stato trovato
- Un orario indicativo nella quale è stato perso.
- Dei tag (elettronica, documenti, materiale di cartoleria, ecc. . . )

Potranno contrassegnare un oggetto come consegnato solo dopo che almeno una persona si è prenotata.

In caso di cestinazione, non si cancellerà la entry nel db, ma verrà contrassegnata come cestinata (ovvero che l'oggetto è stato gettato).

Un amministratore può anche annullare un reclamo e rendere di nuovo l'oggetto reclamabile se la persona che ha effettuato il primo reclamo non si presenta oppure ha perso interesse nel reclamare l'oggetto.

Ovviamente tutte le modifiche effettuate su un item dovranno essere loggate.

La piattaforma si dovrà interfacciare con quella degli orari, in modo tale da dare un consiglio per quanto riguarda i corsi e la relativa annata nella quale fare girare l'annuncio.

A proposito di annuncio, deve preparare un copy per quanto riguarda il mandare la comunicazione nei gruppi dei singoli anni.

Dovrà arrivare una mail ai ragazzi se l'oggetto smarrito è possibile che sia stato perso esclusivamente nel loro corso di appartenenza, questo per evitare email spam.

Tutti gli oggetti saranno sempre listati nello storico anche se consegnati o cestinati, in modo tale da essere sempre rintracciabili.

Dopo un mese da quando l'oggetto è stato trovato, la piattaforma manderà una mail agli amministratori dell'auletta afferente a quell'oggetto per spronarli a cestinare l'oggetto smarrito.

Questo perché ovviamente se non verrà effettuato nessun reclamo l'oggetto andrà cestinato e quindi distrutto.

### **Gli studenti**

Gli studenti potranno visionare l'archivio degli oggetti smarriti in base all'auletta di appartenenza del corso ed in base all'aula nella quale è stato ritrovato l'oggetto smarrito.

Si potrà filtrare anche per parole chiave ("Astuccio", "Orologio", "Laptop").

Potranno fare un reclamo e proclamare come proprio l'oggetto smarrito, verranno quindi esortati sia dalla piattaforma che da una mail automatica a presentarsi giorno dopo per un ritiro.

### **EXTRA**

Potrebbe essere interessante integrare il tutto con una mappa dell'università e le relative coordinate.

## **14.3 Stato**

In ideazione

## 15 Elezioni

### 15.1 Idea

Piattaforma che permetterà ai membri di staff di calcolare, coefficientare e catalogare i voti.

### 15.2 Analisi dei requisiti

Si devono poter creare nuove elezioni e selezionare i corsi ai quali esse fanno riferimento, si possono inoltre selezionare interi dipartimenti o l'intera università. Deve essere possibile candidare un membro di staff ad una qualsiasi carica che però nel caso dei CCS possono essere solo relativi al proprio corso di laurea e dei CDD del proprio dipartimento. Deve essere possibile creare, cancellare e modificare delle nuove cariche (metti caso nasce ne nasce una nuova) e si deve poter decidere se possono partecipare studenti solo del proprio corso o del proprio dipartimento. Deve essere possibile creare, cancellare e modificare i poli universitari ed i corsi a loro afferenti. La piattaforma deve offrire una schermata di resoconto dei risultati adempiuti per ogni corso, dipartimento e polo universitario. (Chiaramente in base al fatto che le elezioni siano ). La visualizzazione deve essere suddivisa in corsi. Ad ogni votazione si dovranno scegliere le metodologie di votazione (D'Hondt, maggioritario, proporzionale, ecc), i candidati e le posizioni di questi ultimi. Si deve prevedere un menù "candidati" dove poter inserire le candidature degli studenti, prevedendo tutte le cariche possibili (CCS, CDD, CSU, CNSU, CDA, CDA, ERSU) e le eventuali custom che si metteranno. I voti di staff dovranno essere categorizzati come tali. Sarà prevista anche una utility per organizzare giri aule.

Sezione dove possiamo verificare se una lista di persone è inserita nel sistema.

### 15.3 Stato

In ideazione

## 16 Filigrana

### 16.1 Idea

Utility che permette di convertire file .docx in file .pdf e di filigranarli con tutti i loghi dell'associazione.

### 16.2 Analisi dei requisiti

La utility di filigrana è esclusiva ai membri di staff ed agli amministratori.

Deve permettere di convertire file .docx in file .pdf e di filigranarli con tutti i loghi dell'associazione.

Si potranno scegliere il logo (in base ai macrostaff) e la percentuale di opacità di quest'ultimo.

Si potrà scegliere tra la versione della filigrana con il logo oppure solo con il testo.

Si potranno caricare uno o più documenti contemporaneamente sia tramite selezione che come drag and drop.

Qualora sia stato caricato più di un file, allora i file verranno zippati.

L'opacità non dovrà mai essere inferiore del 5%.

### 16.3 Stato

Implementazione completata, aspetto che mi vengano date le filigrane solo con il testo per pushare e mettere in up il sito definitivamente.

## 17 Segnalazioni

### 17.1 Idea

La piattaforma delle segnalazioni è una utility che permette di effettuare delle segnalazioni riguardanti le aule, i dipartimenti, ed in generale gli spazi universitari.

### 17.2 Analisi dei requisiti

#### **Gli utenti**

Potranno effettuare delle segnalazioni che saranno identificate da:

- Foto o video breve della problematica
- Descrizione breve (Es. "LIM non funzionante").
- Descrizione lunga (Es. "La LIM in aula non funziona correttamente, spesso sfarfallando rende le lezioni impossibili").
- Aula o luogo (da capire se separare le due cose)
- Priorità: Bassa, Media, Alta, Pericolo
- Stato: Rifiutata, non segnalata, segnalata, sistemata
- Tipologia: Manutenzione ordinaria, manutenzione straordinaria, ecc ...
- Data segnalazione

Gli utenti potranno inserire solamente la foto o il video della segnalazione, le descrizioni e l'aula o il luogo.

Queste segnalazioni dovranno poi essere revisionate e confermate da membri di staff.

Ogni utente può vedere la lista di segnalazioni fatte nel proprio dipartimento, potendo inoltre filtrare per tutti i parametri della segnalazione detti prima.

Chiaramente si dovrà poter vedere anche lo storico delle segnalazioni risolte

#### **Ogni staffer**

Ogni staffer potrà aggiungere, modificare e rifiutare una segnalazione.

Durante la fase di accettazione di una segnalazione, lo staffer dovrà assegnare una priorità ed una tipologia alla segnalazione.

Dovrà essere mantenuta un changelog per ogni segnalazione, ovviamente append-only.

Per rifiutare una segnalazione, bisogna però inserire un commento obbligatorio in modo tale da giustificare il rifiuto.

Potrà farsi generare una mail per poter segnalare le problematiche in base alle categorie, infatti, in genere è comodo raggruppare le segnalazioni e mandare una singola mail.

Anche loro potranno vedere solo la lista di segnalazioni fatte nel proprio dipartimento, filtrando anche loro allo stesso modo degli utenti normali.

## **Gli amministratori**

Potranno vedere tutte le segnalazioni di tutti i dipartimenti.

Potranno filtrare in base ai dipartimenti

Potranno aggiungere o modificare le personalità che si dovranno occupare di sistemare le segnalazioni ed a quale dipartimento fanno parte. Dovranno specificare di che tipologie di intervento si occupano le personalità.

## **La piattaforma**

La piattaforma, durante la fase di creazione di una segnalazione dovrà avvertire se esiste già una segnalazione simile. La piattaforma dovrà mantenere una lista di personalità che si occupano della manutenzione ed in generale di persone alla quale devono essere mandate le segnalazioni, in modo tale che in fase di segnalazione ufficiale il sistema indichi in base alla tipologia della segnalazione, la personalità alla quale mandare la mail.

La piattaforma deve prevedere che in base alle tipologie di intervento ed all'edificio ci siano diverse personalità dell'ateneo alla quale fare riferimento per le segnalazioni (Signor Vincenzo, Fanale, ...).

Come già detto, la piattaforma dovrà provvedere a generare/precompilare una mail per segnalare le problematiche, direttamente aprendo il client ed inserendo le email delle personalità interessate.

Idealmente la piattaforma aggregherà quindi ogni problematica relativa ad una tipologia specifica in un'unica mail, che poi il membro di staff potrà modificare e mandare.

La piattaforma deve mandare delle mail ad ogni cambio di stato della segnalazione da loro fatta, sia per quanto riguarda la persona che ha sottomesso originariamente la segnalazione che per quella che ha mandato la mail. Quindi ad ogni cambio di stato (Rifiutata, Non segnalata, segnalata, sistemata) ci sarà una rispettiva mail

## **Regole**

Alcune regole da rispettare saranno:

- Non si devono creare segnalazioni duplicate di altre

## **Extra**

Sarebbe carino implementare un sistema di analisi per quanto riguarda le statistiche di problematiche risolte, medie di problematiche del mese, aule più problematiche ecc. ecc.

## **17.3 Stato**

In ideazione