# Ricevimento 04/09/2020

## Descrizione del sistema

Si vuole realizzare un’applicazione web in grado di fornire un registro elettronico ad un istituto scolastico e a tutte le entità ad esso correlate (segreteria, docenti e studenti).

## Processo di sviluppo

Si è scelto di adottare il framework di sviluppo agile SCRUM, cercando di restare quanto più fedeli alla SCRUM guide. Per supportare lo sviluppo è stato utilizzato il software Jira.

### Dubbi

* Documentazione da fornire
  + *Documenti e modelli da produrre in fase di avvio:* è sufficiente il product backlog? Se non lo è, quali documenti si devono produrre?
  + *Documenti per la specifica dei requisiti:* sono sufficienti storie utente ed epic inserite nel product backlog?
  + *Documenti per l’analisi dei requisiti*
  + *Documentazione di progetto (architetturale e non):* bisogna documentare solo i componenti prodotti o anche quelli offerti dai framework utilizzati?
  + *Documentazione dell’implementazione*
  + *Descrizione dei test*
* Differenza tra epic e storia utente.

## Framework utilizzati

Si è scelto di utilizzare diversi framework di sviluppo:

* (Backend) Spring boot: Spring Security, Spring Data JPA
* (Frontend) Vaadin

## Architettura del sistema

Lo stile architetturale del sistema è di tipo client-server.

* Il **client** è un client realizzato attraverso Vaadin.
* Il **server**, scritto in Java, rispetta il pattern architetturale MVC:
  + Model: persistenza gestita tramite Spring Data JPA (e Hibernate)
  + View: realizzata tramite i componenti offerti da Vaadin.
  + Controller: realizzato attraverso Spring Boot.

### Dubbi

* Come possiamo mappare service e repository nel pattern MVC?
  + Service -> controller
  + Repository -> tra model e database

# Ricevimento 09/09/2020

Sprint 1 terminato 🡪 sprint review necessaria

## Requisiti

* Requisiti funzionali:
  + Definizione dei requisiti in maniera testuale.
  + Specifica dei requisiti a mezzo di storie utente su Jira Software.
  + Dubbi:
    - è possibile portare un solo esempio sul pdf di documentazione?
    - Le storie utente sono sufficienti a descrivere i requisiti oppure sono necessari anche i casi d’uso?
* Requisiti non funzionali:
  + Descritti in modo testuale.

## Diagrammi

* Component diagram di analisi:
  + Dubbi:
    - è possibile descrivere l’architettura del sistema utilizzando un semplice diagramma (ad es. package diagram) che rappresenti esclusivamente i livelli in cui è suddiviso il sistema?
    - Il modello di programmazione dei framework non consente di realizzare una architettura strict layered.
    - come possiamo collegare il backend al livello di database.
* System Domain Model
* System Model: Include le classi sviluppate per la UI?
* Sequence diagram di analisi: è necessario sviluppare anche i diagrammi delle funzionalità più elementari (ad es. aggiunta di un docente al sistema).

## Feedback sull’applicazione

* Deployment su piattaforma AWS

# Ricevimento Amalfitano 15/09/2020

## Analisi

* Context diagram
* System domain model con responsabilità

## Progettazione

* System model GRASP
  + **Problema:** non è coerente con l’implementazione
* Se va bene il system model:
  + Note nel diagramma
  + Sequence GRASP
* System model (parziale) che rispetta l’implementazione
  + Si può organizzare in MVC? Il model non notifica gli aggiornamenti alla view.
* Architettura: component diagram

## Implementazione

* Deployment diagram

# Ricevimento 17/09/2020

## Documentazione

* Nella sezione Avvio come possiamo esprimere “requisiti generali e di qualità, vincoli”. È necessario utilizzare un formato standard?

## Analisi

* Context diagram:
  + È corretta la notazione che abbiamo utilizzato?
  + Va inserito il database?

## Progettazione

* Suddivisione progettazione in RDD ed FOD + “Matrice tracciabilità” + sequence diagram di entrambi
* Component diagram, domande sul vpp
* Package diagram FOD: Problema del pattern architetturale MVP