**Descrizione del progetto: sito web di un'istituzione educativa**

Il progetto per l'anno accademico 2020/2021 consiste nella realizzazione di un sito web per un'istituzione educativa.

1. Il sito web deve avere una homepage che fornisca informazioni per l'università, i dipartimenti e le informazioni di contatto (numero di telefono e indirizzo) accessibili a chiunque lo visiti.
2. Il sito web deve avere un portale accessibile da studenti, docenti, segreteria dell'università e amministratore del sito.
3. Se uno studente accede al portale, può:
4. Visualizzare la lista dei corsi disponibili.
5. Iscriversi a un corso.
6. Registrarsi per un esame di un corso.
7. Visualizzare le informazioni personali (nome, cognome, email, corsi ed esami a cui sono iscritti).
8. Visualizzare i voti degli esami svolti.
9. Se un docente accede al portale, può:
10. ***Creare/Modificare/Cancellare un corso.*** Per creare un corso, il docente deve inserire il nome del corso, la data di inizio e fine del corso, una descrizione del corso, e opzionale un co-docente o un esame. Un corso può avere solo un esame.
11. ***Aggiungere/Modificare/Cancellare un esame.*** Per creare un esame, il docente deve aggiungere la data, l'ora e durezza dell'esame e l’aula dove si svolgerà l'esame. In un esame, un studente può registrarsi solo una volta.
12. Visualizzare le proprie informazioni personali (nome, cognome, corsi che insegna con relativo esame)
13. Inviare i voti dell'esame alla segreteria
14. Se la Segreteria effettua il login al portale, può:
15. Visualizzare i dati personali degli studenti
16. Inserire voti d'esame per gli studenti
17. Pubblicare i voti degli esami.
18. Se l'amministratore accede al portale, può:
19. Creare account per studenti/docenti/segreteria utilizzando il loro indirizzo email
20. Aggiungere/Modificare/Visualizzare le informazioni personali degli studenti e dei docenti.

**Modalità di realizzazione**

* Il modello di processo suggerito è un ibrido strutturato-agile composto da due fasi: *inception*e *construction*.
* In inception vengono prodotti come artefatti una modellazione dei casi d’uso, un modello di dominio, un glossario (più eventualmente altri artefatti accessori).
* In costruction si seguono le pratiche di gestione di processo di Scrum, due componenti del gruppo giocano il doppio ruolo di membri del team e di scrum master e product owner (i ruoli si ridistribuiscono al termine di ogni sprint e bisogna documentare l’assegnazione dei ruoli nel Diario). Sono rispettati gli eventi standard (Sprint planning, Daily scrum, Sprint review e Sprint retrospective) e vengono prodotti e gestiti gli artefatti standard (Product backlog, Sprint backlog, Burn down chart).
* I casi d’uso identificati in inception vengono posti inizialmente nel backlog e raffinati in user stories (tipicamente una per scenario).

**Prodotti da realizzare**

Prima di consegnare, controllare nuovamente questa lista e verificare che tutta la documentazione richiesta in seguito sia stata prodotta. Rileggere più volte attentamente tutte le indicazioni!!

Ogni gruppo dovrà realizzare:

* Artefatti obbligatori della fase di inception
* Modello dei casi d’uso (con la descrizione dettagliata di almeno un paio di casi d’uso).
* Modello di dominio
* **Glossario**
* **Diario del progetto:**questo documento deve riportare in modo sintetico e schematico lo stato di avanzamento del progetto e le attività del gruppo. In particolare all’inizio di ogni sprint devono essere specificati lo stato dei backlog e i criteri seguiti per la popolazione dello sprint backlog attraverso l’assegnazione delle priorità e dell’effort agli elementi del product backlog. Deve inoltre essere presente un dettaglio delle attività del gruppo dove vengono riportate in modo sintetico e schematico: tutte le attività svolte dal gruppo giorno per giorno, inizio/fine iterazioni e sprint, eventi di scrum, ruoli assegnati tramite scrum, versioni intermedie o primordiali degli artefatti, link ai commit delle release di codice ed eventualmente screenshot delle board per il tracking del progetto. Alla consegna del progetto condividere il link al documento permettendo la possibilità di modificare (scegliere l’opzione corretta da google doc).
* **Descrizione del metodo adottato**, questa deve riportare: numero di iterazioni, milestone,  issue, criteri per l’assegnazione di priorità ai task, user stories e artefatti scrum.
* **Relazione su scelte progettuali e di interfaccia.**
* **Manuale**. Il manuale utente deve contenere le istruzioni su come installare e lanciare su un nuovo computer il software sviluppato condiviso tramite il repository Bitbucket. Inoltre dovrà contenere una panoramica che documenti come sono state realizzate le funzionalità richieste.

**Indicazioni di carattere generale da seguire obbligatoriamente**

* Il progetto dovrà essere realizzato in gruppo, ogni gruppo dovrà essere composto da massimo 3 persone.
* Effettuare unit testing (utilizzando JUnit).
* Utilizzare il linguaggio Java e GWT.
* Il codice dovrà essere opportunamente commentato e indentato.
* Utilizzare in maniera appropriata gli strumenti di versionamento del codice (Git), bug tracking e code review (creazione di branch, pull request ed attuazione dei merge);
* Si dovrà utilizzare [Github](https://www.google.com/url?q=http://github.com&sa=D&source=editors&ust=1639490901715000&usg=AOvVaw3rsN8cM7xxno0FFv1fGPED) per gli strumenti a supporto del progetto (gestione delle revisioni, issue/project tracking). Il repository di GitHub deve essere privato.
* Usate Github branches per lo sviluppo dei feature del sito.
* Per la documentazione, [create una directory](https://www.google.com/url?q=https://github.community/t/add-a-folder/2304&sa=D&source=editors&ust=1639490901716000&usg=AOvVaw15ZiGurENx7EGEQrAuunrY) GitHub denominata "Documentazione" e caricate lì tutti i vostri file. E’ obbligatorio usare il linguaggio [Markdown](https://www.google.com/url?q=https://www.markdownguide.org/basic-syntax/&sa=D&source=editors&ust=1639490901717000&usg=AOvVaw091KiIRhotjOOSam55HRkr) per la documentazione, se create file direttamente in Github.
* Il gruppo deve obbligatoriamente utilizzare Trello e Jira. Il gruppo deciderà in seguito come utilizzare questi due strumenti. Si consiglia di usare Trello come product backlog e Jira come sprint backlog.
* Con Jira si intende l’issue tracker gratuito integrato in GitHub (non il servizio a pagamento). Guida per abilitare Jira in Github: [guida](https://www.google.com/url?q=https://support.atlassian.com/jira-cloud-administration/docs/integrate-with-github/&sa=D&source=editors&ust=1639490901718000&usg=AOvVaw1FNIRzvI0AClTEnz1pcMyR).
* Guida per abilitare Trello in GitHub:
* Creare una nuova carta (attività) a Trello.
* Apri la carta.
* Nella sezione Power-up, clic su Add power-ups.
* Nella nuova finestra che si apre, clic su Developers tools e cerca GitHub. Fare clic su aggiungi e concedere le autorizzazioni necessarie per l'integrazione.
* Ora, Github è integrato. Vai di nuovo alla card e nella sezione di Power-ups, allega la tua card a un issue, commit, branch o pull request in GitHub.
* Tutte le informazioni gestite dal sito dovranno essere accessibili anche dopo il riavvio del server. Per gestire la persistenza è obbligatorio utilizzare MapDB (MapDB). Non si accetterà la consegna di progetti che usano altri meccanismi di gestione della persistenza.
* L’utente amministratore dovrà avere come username “admin” e password “admin”. La sua registrazione non potrà avvenire attraverso interfaccia web.

**Consigli**

* Le slide mostrate durante la lezione di presentazione del progetto sono disponibili al seguente [link](https://www.google.com/url?q=https://docs.google.com/presentation/d/1oOFuf8deYrsGPArwDKdKHHhO3EPZgNkD/edit?usp%3Dsharing%26ouid%3D113864870076967859609%26rtpof%3Dtrue%26sd%3Dtrue&sa=D&source=editors&ust=1639490901721000&usg=AOvVaw2-gJQxng0hz4cqJoy8JMvg).
* Si consiglia di utilizzare Eclipse come IDE.
* E’ apprezzato l’uso di Google Doc per la stesura del diario e delle relazioni. Per i diagrammi si consiglia l’utilizzo di [draw.io](https://www.google.com/url?q=https://draw.io/&sa=D&source=editors&ust=1639490901722000&usg=AOvVaw04GaZV8fsn70Tcor4EIVFw). Draw.Io è uno strumento che permette l’editing collaborativo di diagrammi UML. Si occupa di gestire le varie revisioni dei diagrammi ed è integrato con Google Drive (salva i diagrammi su una directory di Google Drive). Inoltre si occupa di fare l’export dei diagrammi in formato pdf o immagine. In alternativa ci sono anche altri strumenti che permettono l’editing dei diagrammi offline (es. [yEd](https://www.google.com/url?q=https://www.yworks.com/products/yed&sa=D&source=editors&ust=1639490901723000&usg=AOvVaw1YFRZhw21oGx6ANAT3E4o6)).
* Per prendere confidenza con GWT e MapDB si consiglia di seguire i relativi tutorial:
* [Tutorial MapDB](https://www.google.com/url?q=https://jankotek.gitbooks.io/mapdb/content/&sa=D&source=editors&ust=1639490901724000&usg=AOvVaw2kTSIG9F7KcM9NqLeV-mC-)
* [Tutorial GWT](https://www.google.com/url?q=http://www.gwtproject.org/doc/latest/tutorial/gettingstarted.html&sa=D&source=editors&ust=1639490901725000&usg=AOvVaw1MApMICYrOZd4fXci7bTc7)
* Come punto di partenza per l’implementazione di può prendere ad esempio [questo](https://www.google.com/url?q=https://github.com/morcilab/StockWatch&sa=D&source=editors&ust=1639490901725000&usg=AOvVaw36QRcqPG428Dkb2uXJRWDx) repository.
* Se si desidera creare un progetto più generale e meno legato ad Eclipse si può passare a Maven. Un esempio di utilizzo di maven e map db potete trovarlo [qui](https://www.google.com/url?q=https://github.com/morcilab/mapdb&sa=D&source=editors&ust=1639490901726000&usg=AOvVaw0tM9L1A-pXfFsybgvRqG2T).
* [Guida Utilizzo GIT](https://www.google.com/url?q=http://rogerdudler.github.io/git-guide/index.it.html&sa=D&source=editors&ust=1639490901726000&usg=AOvVaw1ggOJujLcSP3cn0Iir1faV).

**Modalità e tempi di consegna**

La consegna avverrà inviando una mail a fiorela(punto)ciroku2(chiocciola)unibo(punto)it e invitando al repository di Bitbucket l’utente “fiorelaciroku”. Successivamente ogni gruppo verrà convocato per discutere il progetto nelle date che verranno rese note di volta in volta nel sito del corso. Ci sarà orientativamente una data di discussione al mese e nel caso di necessità particolari potranno essere concordate date aggiuntive.

* **Nota**Quando il gruppo invia la mail per consegnare il progetto devono essere in CC tutti i componenti del gruppo!!

**Valutazione**

La valutazione del progetto avverrà tenendo conto dei seguenti punti:

* La realizzazione delle specifiche funzionali;
* L’organizzazione e la leggibilità del codice;
* L’adozione di una metodologia e l’utilizzo appropriato degli strumenti di testing, versioning, bug tracking, code reviewing e project tracking;
* L’adeguatezza della documentazione allegata al progetto e in particolare della descrizione del processo di sviluppo adottato;
* La discussione del progetto.

Per qualsiasi dubbio o chiarimento riguardanti le specifiche del progetto non esitate a contattarmi all’indirizzo email fiorela(punto)ciroku2(chiocciola)unibo(punto)it o utilizzate il forum dell’insegnamento.

**Verifiche finali prima della consegna!**

Prima della consegna definitiva rileggere attentamente, più volte specifiche di progetto e procedere con le seguenti verifiche:

* Le specifiche sono state rilette?
* E’ stato consegnato tutto quello che era richiesto nelle specifiche di progetto?
* I diagramma sono completi? Nel diagramma delle classi tutte le associazioni hanno un nome e cardinalità? Le classi hanno gli attributi necessari?
* I documenti google doc sono stati condivisi in modalità modifica?
* Il codice rispecchia la documentazione prodotta?
* Per eseguire il codice basta fare checkout del repository e seguire le istruzioni nel manuale di installazione?
* La board Trello è accessibile dal repository?