ECOMENTOR

PROJECTE D'ENGINYERIA DEL SOFTWARE, Q2 2024-2025
INCEPCIÓ 1 i 2
GRUP 22

Víctor Díez Serrano, Backend Developer
Dídac Dalmases Valcárcel, Scrum Master 1 & Frontend Developer
David Mas Escudé, Scrum Master 2 & Backend Developer
Rubén Palà Vacas, Frontend Developer & Architectural designer
David Sanz Martínez, Backend Developer
Neptune Christoper Lumayag Cartalla, Frontend Developer

(Preliminary note: this document represents the current state of the project. Please keep all sections updated and coherent with each other.)

1. Requirements

1.1. General conception of the project

See inception document

1.2. "NOT" list

See inception document

1.3. Conceptual model

UML conceptual model (in the form of class diagram) of the project

1.4. Summary of product backlog in the document

Fill the table below:

Epic 1	US1,1(x, y)	US1,2(x, y)	…		
Epic 2	US2,1(x, y)	…			
…	…				

(x, y): x means in which sprint it entered the sprint backlog, y means in which sprint it was done. Alternatively, you can show the epics and user stories from Taiga.

1.5. Non-functional requirements

Using Volere template (ref. ER course)

1.6. Treatment of transversal aspects

Clearly indicate final target and current state

Aspect	Description	Current status
Geolocalització		
Xarxes socials		
Xat		
Gamificació		
Stakeholders reals		
Refutació		
Calendari		
Web-app admin		

Multiidioma		
Add others as required…		

1.7. Third-party services

Describe both the service offered, and the service to be integrated, as mentioned in the course slides (Chapter 3).

Describe how are you communicating with the other teams

2. Methodology

Durant el desenvolupament del projecte, hem seguit un enfocament àgil, utilitzant Scrum com a metodologia principal per organitzar el treball i gestionar els lliuraments de manera iterativa. Aquest enfocament ens ha permès adaptar-nos als canvis, validar progressivament les funcionalitats i mantenir una comunicació fluida dins de l'equip.

2.1. Project management

- Utilitzem **Taiga** per gestionar el backlog i fer seguiment de les tasques amb **story points** per prioritzar i estimar l'esforç de cada història d'usuari.
- Els sprints es defineixen en base als objectius clau i les tasques es tanquen mitjançant revisió col·laborativa.
- Les reunions de revisió permeten detectar bloquejos i ajustar el roadmap.

2.2. Repository management

- Implementem **GitHub** com a sistema de control de versions, seguint un model **Git Flow** amb branques separades per funcionalitats, desenvolupament i producció.
- Cada canvi passa per un procés de revisió via **pull requests** abans de ser fusionat a la branca principal.
- Es fan commits freqüents per assegurar la traçabilitat dels canvis.

2.3. Communication within the team

- Per a la comunicació diària, fem servir un **grup de WhatsApp** per missatges ràpids i un **servidor de Discord** per reunions més extenses.
- Es realitzen **dailies informals** per sincronitzar l'equip i resoldre possibles bloquejos.
- Taiga facilita la documentació i assignació clara de responsabilitats.

2.4. Quality management

- Hem establert **estàndards de codi** per garantir coherència i mantenibilitat.
- Les revisions de codi són una part essencial del procés, assegurant bones pràctiques i identificant possibles errors abans del merge.

2.5. Testing strategy

- Utilitzem proves unitàries i integració contínua per validar les funcionalitats abans del desplegament.
- Els tests es desenvolupen amb eines adequades segons la tecnologia seleccionada.
- Els resultats de les proves es documenten per a millorar iterativament la qualitat del programari.

2.6. Management of configurations

- Les configuracions es gestionen mitjançant **fitxers d'entorn** i variables per adaptar el projecte a diferents entorns sense comprometre la seguretat.

2.7. Interaction with colleagues

- La col·laboració es fomenta a través de revisions de codi i sessions de treball conjuntes per garantir un alineament tècnic.
- Les funcionalitats es discuteixen en equip abans de la seva implementació per assegurar una comprensió comuna.

2.8. Bug management

- Els errors es registren a Taiga com a **incidències**, prioritzant-se segons l'impacte en el projecte.
- S'apliquen estratègies de **debugging** per identificar i resoldre problemes de manera eficient.

2.9. NFRs treatment

- Els requisits no funcionals, com ara rendiment, seguretat i accessibilitat, s'han tingut en compte des de la fase d'arquitectura i disseny.
- Es realitzen proves per assegurar que l'aplicació compleix amb aquests requisits abans de la seva entrega.

2.10. Coding assistants

- Hem fet ús d'eines d'ajuda al desenvolupament, com **linters** per garantir la qualitat del codi i assistents d'IA per suggeriments i optimitzacions.

3. Technical description

3.1. Overall conception of the architecture

3.1.1. Physical architecture

Graphical description of the different elements that configure the layout of the system

3.1.2. Architectural pattern(s) applied

Justification of need

3.2. Domain layer

3.2.1. Domain model diagram (optional)

In case you think it helps

3.2.2. Design patterns applied

Justification of need
Explanation of application (eventually including some sequence diagram or code for illustration purposes)

3.3. Database diagram (UML)

UML class diagram describing the data base (only of the part implemented so far)

3.4. Instrumentation and list of technologies

Llenguatges de programació i frameworks:

- **React Native + Expo (JavaScript):** Utilitzat en la part **frontend** de l'aplicació.
- **SpringBoot (Java):** Extensió de l'*Spring Framework* que facilita molt la gestió i creació d'aplicacions **API REST**, a més que proporicona eines com Spring Data JBDC o Spring Data JPA que faciliten la creació de **queries SQL.**

Servidors:

- **Amazon EC2 + (Docker / Amazon RDS):** Utilitzarem un servidor Amazon EC2 per allotjar el nostre backend. Inicialment, s'utlitzarà també **Docker** per crear

un contenidor on s'allotjarà la base de dades. En etapes més avançades del desenvolupament no es descarta fer una migració a Amazon RDS.

Base de dades:

- **PostgreSQL:** Solució amb suport i documentació àmplia de bases de dades relacional.

3.5. APIs

Present any API you may be offering
Explain which external APIs are you using in your program
Explain service consumption and service supply to your teammates
Open data consumption: how is it done?

3.6. Development tools and working environment

Use of Frameworks
Continuous Integration
Deployment