

Università degli Studi di Salerno
Corso di Ingegneria del Software

WeCook
Object Design Document
Versione 1.0



WeCook

Data: 15/12/2024

| | |
|-----------------------------------|------------------|
| Progetto: WeCook | Versione: 1.0 |
| Documento: Object Design Document | Data: 15/12/2024 |

Coordinatore del progetto:

| Nome | Matricola |
|------|-----------|
| | |

Partecipanti:

| Nome | Matricola |
|-------------------------|------------|
| Simone Francesco Albino | 0512116140 |
| Valentina Ferrentino | 0512116842 |
| Giovanni Semioli | 0512117412 |

| | |
|--------------------|---|
| Scritto da: | Simone Francesco Albino - Valentina Ferrentino - Giovanni Semioli |
|--------------------|---|

Revision History

| Data | Versione | Descrizione | Autore |
|------------|----------|---------------|--------|
| 15/12/2024 | 1.0 | Prima stesura | Team |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

| | |
|-----------------------------------|------------------|
| Progetto: WeCook | Versione: 1.0 |
| Documento: Object Design Document | Data: 15/12/2024 |

Indice

| | |
|---|----------|
| 1. Introduzione | 3 |
| 1.1 Object design goals | 3 |
| 1.2 Interface documentation guidelines | 4 |
| 1.3 Definizioni, acronimi e abbreviazioni | 4 |
| 1.4 Riferimenti | 4 |
| 2. Packages | 5 |
| 3. Interfacce delle classi | 6 |

| | |
|-----------------------------------|------------------|
| Progetto: WeCook | Versione: 1.0 |
| Documento: Object Design Document | Data: 15/12/2024 |

1. Introduzione

We Cook nasce con l'idea di connettere persone di tutto il mondo che condividono una passione comune: la cucina. La piattaforma ha l'obiettivo di mettere in contatto tradizioni culinarie, sapori e storie personali, abbattendo le barriere culturali e geografiche, offrendo uno spazio dove chiunque possa condividere le proprie radici, la propria idea di cucina, in modo da permettere ad ogni ingrediente di essere valorizzato al meglio.

1.1 Object design goals

Affidabilità: Il sistema deve garantire stabilità operativa, protezione dei dati e gestione sicura di eventuali errori, assicurando il corretto funzionamento anche in condizioni di input non validi o inattesi.

Efficienza: La progettazione deve ottimizzare il sistema per tempi di risposta rapidi, garantendo che le operazioni come il caricamento delle pagine, delle immagini e dei contenuti avvengano entro limiti prestabiliti, minimizzando l'uso delle risorse.

User-Friendliness: Il design deve essere intuitivo e orientato all'utente, con notifiche chiare per fornire un feedback immediato e utile, contribuendo a migliorare l'esperienza e la comprensione del sistema.

Robustezza: La progettazione deve prevedere soluzioni che garantiscano una gestione solida degli errori e un comportamento consistente anche in caso di malfunzionamenti o input non previsti.

Flessibilità: Il sistema deve essere progettato per adattarsi facilmente a nuove esigenze o aggiornamenti futuri, riducendo al minimo gli sforzi necessari per introdurre miglioramenti o estensioni.

Riusabilità: Il sistema deve essere progettato con componenti modulari e generici, come notifiche, layout responsive, logging e validazione centralizzati, che possano essere facilmente riutilizzati in altre parti del sistema.

| | |
|-----------------------------------|------------------|
| Progetto: WeCook | Versione: 1.0 |
| Documento: Object Design Document | Data: 15/12/2024 |

1.2 Interface documentation guidelines

Questo sottoparagrafo sarà completato con la stesura del codice.

1.3 Riferimenti

Come ispirazione per la realizzazione di WeCook sono stati presi come riferimento diverse piattaforme, analizzandone il design e le principali funzioni social, come il feed delle notizie, la gestione dei profili utente e le interazioni tra utenti (Instagram, Facebook, ecc).

Il progetto si basa, inoltre, su metodologie descritte nei testi di elencati di seguito:

- Bruegge, Bernd, e Dutoit, Allen H., Object-Oriented Software Engineering Using UML, Patterns, and Java™ (Third Edition), Pearson;
- Requirements Analysis Document;
- System Design Document.

| | |
|-----------------------------------|------------------|
| Progetto: WeCook | Versione: 1.0 |
| Documento: Object Design Document | Data: 15/12/2024 |

2. Packages

Questo paragrafo sarà completato con la stesura del codice

| | |
|-----------------------------------|------------------|
| Progetto: WeCook | Versione: 1.0 |
| Documento: Object Design Document | Data: 15/12/2024 |

3. Interfacce delle classi


| | |
|--|---|
| <p>context PostService::changeStato(s) post: getStato() = s</p> <p>context PostService::publish()</p> <p>context PostService::addLike(l) pre: not contains(l) post: contains(l)</p> <p>context PostService::removeLike(l) pre: contains(l) post: not contains(l)</p> <p>context PostService::getActivePosts()</p> | <div> <div> <div></div> <div>PostService</div> </div> <div> + aggiungiCommento(c: Commento) + aggiungiLike(l: Like) + rimuoviLike(l: Like) + cambiaStato(s: Stato) </div> </div> |
|--|---|

| | |
|---|--|
| <p>context SavedPostService::getSavedPost()</p> <p>context SavedPostService::addSavedPost(p) pre: not contains(p) post: contains(p)</p> | <div> <div> <div></div> <div>SavedPostService</div> </div> <div> + getSavedPost(): List<Post> + addSavedPost(p: Post) </div> </div> |
|---|--|

| | |
|---|---|
| <p>context AuthenticationService::login(username, pwd) pre: contains(username)</p> <p>context AuthenticationService::register(username, pwd) pre: not contains(username) post: contains(username)</p> | <div> <div> <div></div> <div>AuthenticationService</div> </div> <div> +login(username: String, password: String) +register(username: String, password: String) </div> </div> |
|---|---|

| | |
|-----------------------------------|------------------|
| Progetto: WeCook | Versione: 1.0 |
| Documento: Object Design Document | Data: 15/12/2024 |

| | |
|--|---|
| context CommentService::changeState(s) post: getStato() = s context CommentService::publish() | <div> CommentService</div> <div>+ changeState(s: Stato) + publish()</div> |
|--|---|

| | |
|--|---|
| context ReportService::repostPost(p) post: p.getState = "active" context ReportService::repostComment(c) post: c.getState = "active" context ReportService::getOpenPostsReports() context ReportService::getOpenCommentsReports() | <div> ReportService</div> <div>+ reportPost(p: Post) + reportComment(c: Comment) + getOpenPostsReports() + getOpenCommentsReports()</div> |
|--|---|

| | |
|---|--|
| context RoleService::hasRole(u) pre: contains(u) post: getUtente() = u | <div> RoleService</div> <div>+hasRole(u: Utente)</div> |
|---|--|