



Nouvelles de rêve

Projet JEE - Groupe 2 - 11/10/2024

Cheffe de projet : Caroline

Développeurs : Florine / Julien / Sid

Sommaire

- Présentation du Projet
- Répartition du Travail
- Conception
- Développement
- Démonstration
- Difficultés
- Conclusion

Projet Nouvelles de Rêve

Site Web communautaire permettant de partager des nouvelles avec les utilisateurs, de les lire, les commenter ou encore les noter.

Possibilité de s'inscrire afin de publier les nouvelles, se connecter / déconnecter et enfin se désinscrire si nécessaire.

Espace administrateur pour une gestion des utilisateurs, nouvelles et commentaires.

Répartition

- Caroline (cheffe de projet) : Conception / Git / Utilisateurs
- Sid : Conception / Story / CSS / PPT / Test
- Florine : Conception / BDD / Commentaires / Profil
- Julien : Conception / Accueils / Profil / CSS

Documents nécessaires

- Maquette
- Diagramme de classe (UML)
- Modèle conceptuel de données (MCD)
- GANTT

[profil](#)



page d'accueil

[inscription / connexion](#)

Voici notre top 3 nouvelles de la semaine !

score

[voir la suite](#)

score

[voir la suite](#)

score

[voir la suite](#)

Ci-dessous les 10 dernières nouvelles sorties :

score

[voir la suite](#)

score

[voir la suite](#)⁶

formulaire inscription

[inscription / connexion](#)

pseudo

mail

mot de passe

confirmation mot de passe

envoyer

formulaire connexion

inscription / connexion

mail

mot de passe

envoyer

formulaire story

déconnexion

Titre :

Nouvelle :

envoyer

story

inscription / connexion

Titre	score
	+
	-
com1	*
com2	*
com3	*
com4	*
	+

Profile Person (Connecté)

header
(non connecté)

Home [/home]
Inscription [/signup]
Se connecter [/login]

header
(connecté)

Home [/home]
Profile [/profile]
Se désinscrire [lien : DisablePerson]
Se déconnecter [lien : Logout]
Ecrire une nouvelle [/createStory]

Vos données personnelles :

ID interne, adresse mail, pseudo, pwd (debug)

Liste de ses 3 nouvelles avec score [/personStories]
puis lien vers l'entièreté de sa bibliothèque de ses nouvelles

Modifier votre mot de passe :

Formulaire de changement de pwd
(oldPwd, newPwd, Confirm => modifier)

Supprimer mon compte [lien : DeletePerson]

Home Admin

adminHeader

Home [back/adminHome]

Se connecter [/login] ou Se déconnecter [lien : logout]

Liste de toutes les nouvelles [back/story]

Choix : Visualiser une nouvelle

Choix : Supprimer une nouvelle

Liste des inscrits [back/person]

Choix : Désactiver un utilisateur

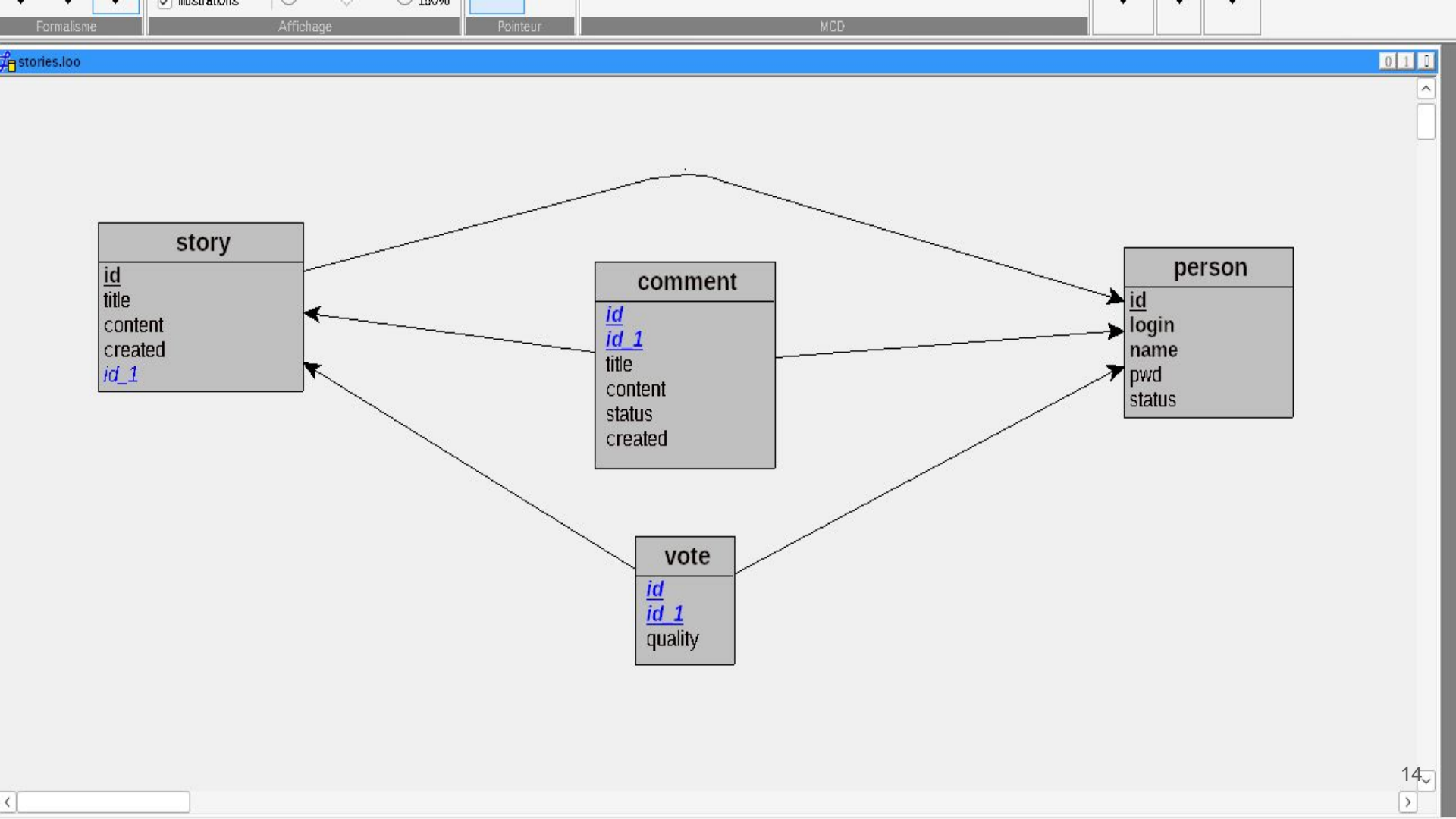
Choix : Supprimer un utilisateur

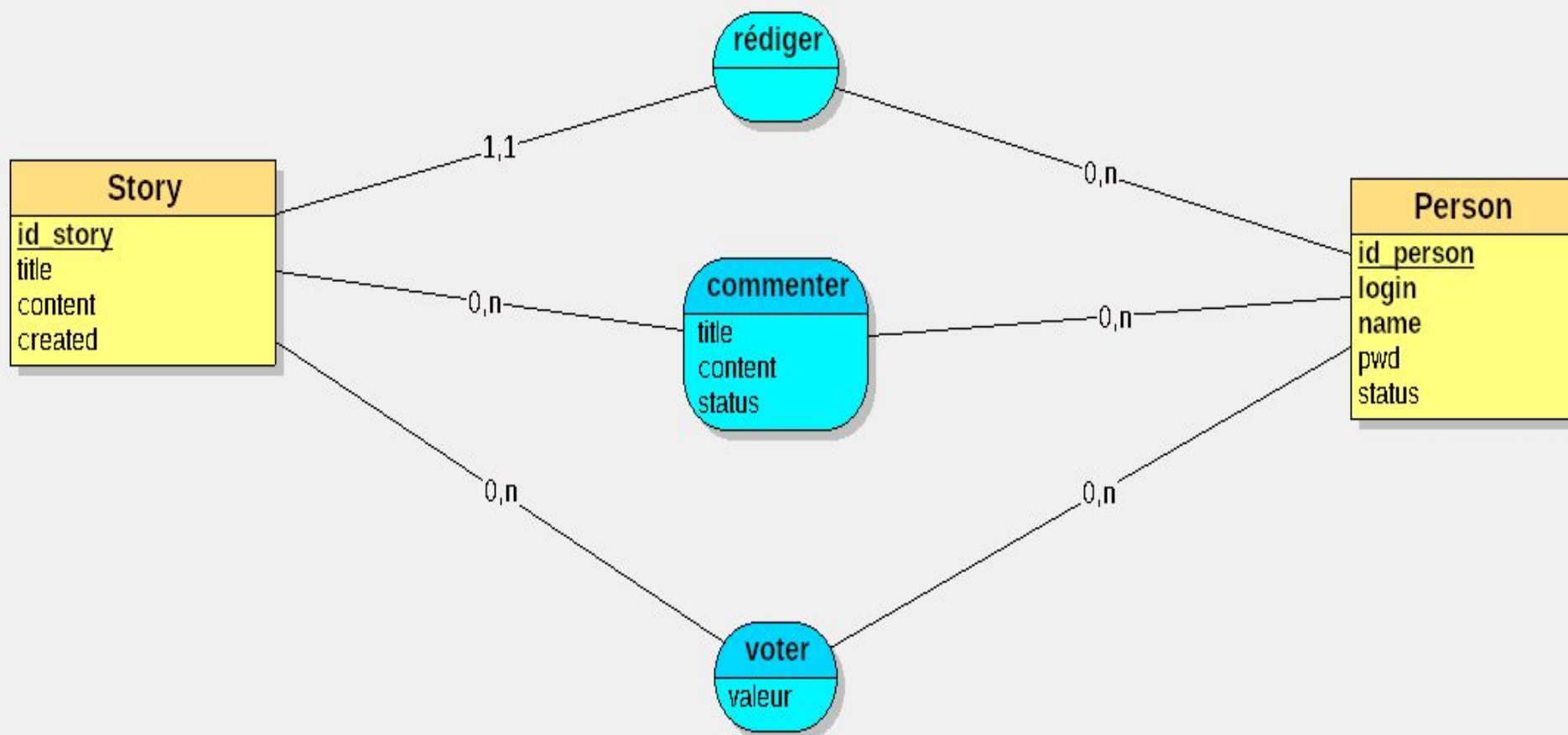
Commentaires signalés [back/comment]

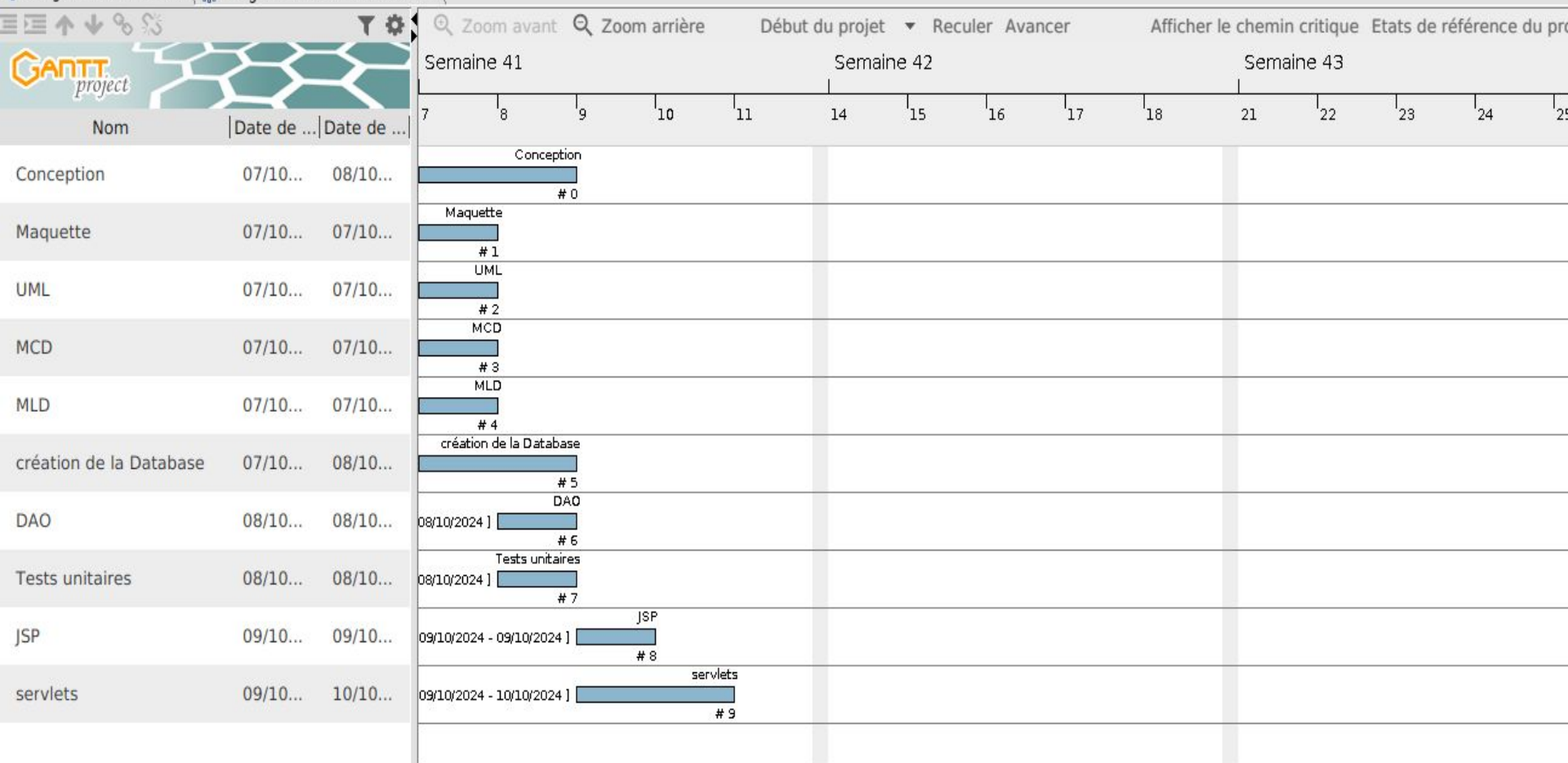
Choix : Visualiser un commentaire signalé

Choix : Valider un commentaire signalé

Choix : Supprimer un commentaire signalé







Base de Données

```
MariaDB [cda202401_je_e_g2]> describe comment;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
id_story	int(11)	YES	MUL	NULL	
id_person	int(11)	YES	MUL	NULL	
title	varchar(50)	NO		NULL	
content	varchar(200)	NO		NULL	
status	tinyint(4)	NO		NULL	
created	date	NO		NULL	

```
7 rows in set (0,025 sec)
```

```
MariaDB [cda202401_je_e_g2]> describe person;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
login	varchar(50)	NO	UNI	NULL	
name	varchar(50)	NO	UNI	NULL	
pwd	varchar(50)	NO		NULL	
status	tinyint(4)	NO		NULL	

```
5 rows in set (0,025 sec)
```

5 rows in set (0,025 sec)

MariaDB [cda202401_jeeg2]> describe story;

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
title	varchar(50)	NO		NULL	
content	varchar(200)	NO		NULL	
created	date	NO		NULL	
id_person	int(11)	NO	MUL	NULL	

5 rows in set (0,027 sec)

MariaDB [cda202401_jeeg2]> describe vote;

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
id_story	int(11)	YES	MUL	NULL	
id_person	int(11)	YES	MUL	NULL	
quality	tinyint(4)	NO		NULL	

4 rows in set (0,084 sec)

Développement

Story

```
@WebServlet(urlPatterns = "/story/create")
public class CreateStory extends HttpServlet {

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        request.getRequestDispatcher(string: "/WEB-INF/jsp/createStory.jsp").forward(sr: request, sr1: response);
    }

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        // Validation du formulaire
        CreateStoryFormChecker formChecker = new CreateStoryFormChecker();
        Map<String, String> errors = formChecker.validateForm(request);

        // Si des erreurs sont présentes, on renvoie au formulaire avec les erreurs
        if (!errors.isEmpty()) {
            request.setAttribute(string: "errors", o: errors);
            request.getRequestDispatcher(string: "/WEB-INF/jsp/createStory.jsp").forward(sr: request, sr1: response);
            return;
        }
    }
}
```

Story

```
// Récupération des paramètres du formulaire
String title = request.getParameter(string: "title");
String content = request.getParameter(string: "content");

// Création d'une nouvelle instance de Story
Story story = new Story();
story.setTitle(title);
story.setContent(content);
story.setCreated(created: LocalDate.now()); // Définit la date de création à aujourd'hui
story.setId_person(id_person: 1L); // Remplacez par l'ID réel de l'utilisateur connecté

// Enregistrement de l'histoire dans la base de données
DAOFactory.getDAOStory().create(story);

// Redirection vers la liste des histoires après création
response.sendRedirect(getServletContext().getContextPath() + "/home");
}
```

```
public class CreateStoryFormChecker {
```

```
/** Vérifie les données du formulaire de création d'une histoire ...6 lines */
```

```
public Map<String, String> validateForm(HttpServletRequest request) {  
    Map<String, String> errors = new HashMap<>();  
  
    // Vérifier le titre  
    String title = request.getParameter(string: "title");  
    if (title == null || title.trim().isEmpty()) {  
        errors.put(k: "title", v: "Le titre ne peut pas être vide.");  
    } else if (title.length() > 100) { // Limite de 100 caractères pour le titre  
        errors.put(k: "title", v: "Le titre ne doit pas dépasser 100 caractères.");  
    }  
  
    // Vérifier le contenu  
    String content = request.getParameter(string: "content");  
    if (content == null || content.trim().isEmpty()) {  
        errors.put(k: "content", v: "Le contenu ne peut pas être vide.");  
    } else if (content.length() > 2000) { // Limite de 2000 caractères pour le contenu  
        errors.put(k: "content", v: "Le contenu ne doit pas dépasser 2000 caractères.");  
    }  
  
    return errors;  
}
```


Accueil

```
//servlet : Home.java
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {

    request.setAttribute(string: "stories3", o: DAOFactory.getDAOStory().listBests3());
    request.setAttribute(string: "stories10", o: DAOFactory.getDAOStory().listLasts10());
    getServletContext() ServletContext
        .getRequestDispatcher(string: VIEW) RequestDispatcher
        .forward(sr: request, sr1: response);
}
```

Accueil

```
// Méthode pour récupérer les 10 dernières stories
public Collection<Story> listLasts10() {
    ArrayList<Story> list = new ArrayList<>();
    String sql = "SELECT * FROM " + table
        + " ORDER BY created DESC LIMIT 10";
    try (PreparedStatement pstmt = this.connection.prepareStatement(string: sql)) {
        ResultSet rs = pstmt.executeQuery();
        while (rs.next()) {
            Story obj = createObject(rs);
            list.add(obj);
        }
    } catch (SQLException ex) {
        throw new RuntimeException("Liste introuvable dans la table story. Erreur :\n" + ex.getMessage());
    }
    return list;
}
```


Accueil

```
<section id="homeStories10">
  <h2>Voici le résumé des 10 dernières nouvelles :</h2>
  <!--afficher que les 200 premiers caractères-->
  <div id="div10" class="grid-layout">
    <c:forEach items="${requestScope.stories10}" var="story">
      <article>
        <h3>${story.title}</h3>
        <!-- Troncature du contenu à 200 caractères -->
        <div class='texte-italique-petit'>Nouvelle écrite le ${story.created} par ${story.name}</div>
```

Florine

```
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    //convertir la chaîne de caractère idParam en Long
    String idParam = request.getParameter(string: "id");
    try {
        id = Long.valueOf(s: idParam);
    } catch (NumberFormatException ex) {
        id = null;
    }
    if (id != null) {
        //récupère le commentaire dans la DB grâce à la DAO
        request.setAttribute(
            string: "comment",
            o: DAOFactory.getDAOComment().findByStatus(status: 0));
    }
    getServletContext().getRequestDispatcher(string: VIEW) RequestDispatcher
        .forward(sr: request, srl: response);
}
```

```

@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    String idParam = request.getParameter(string: "status");
    CreateCommentFormChecker form = new CreateCommentFormChecker(request);
    Comment comment = form.checkForm();
    comment.setStatus(status: 0);
    if (form.getErrors().isEmpty()) {
        DAOFactory.getDAOComment().persist(obj: comment);
        response.sendRedirect(
            getServletContext().getContextPath() + "/back/comments");
    } else {
        DAOFactory.getDAOComment().updateStatus(id, newStatus: 1);
        request.setAttribute(string: "msg", o: "Le commentaire est supprimé");
        getServletContext() ServletContext
            .getRequestDispatcher(string: VIEW) RequestDispatcher
            .forward(sr: request, srl: response);
    }
}

```

Utilisateur

```
public void updateStatus(Long id, int newStatus) {  
    try {  
        String req = "UPDATE " + table + " SET status=? WHERE id=?";  
        PreparedStatement pstmt = this.connection.prepareStatement(string: req);  
        pstmt.setInt(i: 1, i1: newStatus);  
        pstmt.setLong(i: 2, l: id);  
        pstmt.executeUpdate();  
    } catch (SQLException ex) {  
        Logger.getLogger(name: DAOPerson.class.getName()).log(level: Level.SEVERE, msg: null, thrown: ex);  
    }  
}
```

Utilisateur

```
public Person findByLogin(String login){
    Person obj = null;
    try {
        String req = "SELECT * FROM " + table
            + " WHERE login=?";
        PreparedStatement pstmt = this.connection.prepareStatement(string: req);
        pstmt.setString(i: 1, string: login);
        ResultSet result = pstmt.executeQuery();
        if (result.next()) {
            obj = createObject(rs: result);
        }
    } catch (SQLException ex) {
        Logger.getLogger(name: DAOPerson.class.getName()).log(level: Level.SEVERE, msg: null, thrown: ex);
    }
    return obj;
}
```

Attentes auxquelles nous avons répondu

- Accueil visible par tous les visiteurs (“Top3” et résumés des 10 dernières nouvelles)
- Accueil spécifique à l’administrateur (gestion des nouvelles, inscrits et commentaires)
- Gestion de l’authentification
- Formulaire d’inscription, de connexion et lien de déconnexion
- Page de profil avec liste de ses nouvelles, formulaire de changement de pwd et lien pour se désinscrire
- Formulaire de création de nouvelle

Git Graph X		
	Merge branch 'sid' into 'master'	9 Oct 2024 08:29 Casteras Caroline
	Merge branch 'master' into 'sid'	9 Oct 2024 08:29 sid ksouri
	Merge branch 'Florine' into 'master'	8 Oct 2024 16:14 Casteras Caroline
	Update vote DAO vote	8 Oct 2024 16:08 Perabout
	Merge master into Florine	8 Oct 2024 16:01 Perabout
	résolution des soucis	8 Oct 2024 15:52 sid
	résolution des soucis	8 Oct 2024 15:49 sid
	Merge branch 'Julien' into 'master'	8 Oct 2024 15:35 Casteras Caroline
	DAOComment OK	8 Oct 2024 15:33 Perabout
	Avant merge	8 Oct 2024 15:31 Caroline Casteras
	useless	8 Oct 2024 15:30 sid
	useless	8 Oct 2024 14:39 sid
	Merge master into Julien	8 Oct 2024 14:31 Anglade
	test dao Story	8 Oct 2024 14:28 sid
	Ajout Crudable<T> + modification find(Long id) (long->Long) dans DAO	8 Oct 2024 14:27 Caroline Casteras
	ajout dsb.properties, MariaDbConnection et DAOVote/Vote	8 Oct 2024 14:21 Anglade
	Merge master into V1Caro	8 Oct 2024 14:06 Caroline Casteras
	modif Story, DAO, DAOStory, Crudable	8 Oct 2024 14:00 sid
	modif Story, DAO, DAOStory, Crudable	8 Oct 2024 13:54 sid
	DAOPerson, et update des long de clés étrangères dans les beans	8 Oct 2024 13:45 Caroline Casteras
	daoStory	8 Oct 2024 12:03 sid
	test branch	8 Oct 2024 10:06 sid
	Merge branch 'Florine' into 'master'	8 Oct 2024 13:32 Casteras Caroline
	DAO et DAOFactory OK	8 Oct 2024 10:44 Perabout
	First Commit	8 Oct 2024 09:45 Caroline Casteras

Démonstration

Conclusion

Questions ?