

Bundes Report

1. Purpose

Today, we continue to have special communication that shares our daily lives, empathizes with each other, and creates new things that we need. Our communication connects each other, creates new values, and supports you who love Germany and Korea. So that the excitement of starting a new life on the other side of the globe and the joy of free communication continue every single day,

"Bundes Report is in your daily life today."

2. Period

November 10, 2020 to June December 02, 2020

3. Position

We started our project during K-Move(Germany) IT School. This project team consisted of three developers. I was the only Front-End developer on this team. I had full responsibility for the view and layout. In addition, to increase user revisit rates and to help learning German, I created game corners.

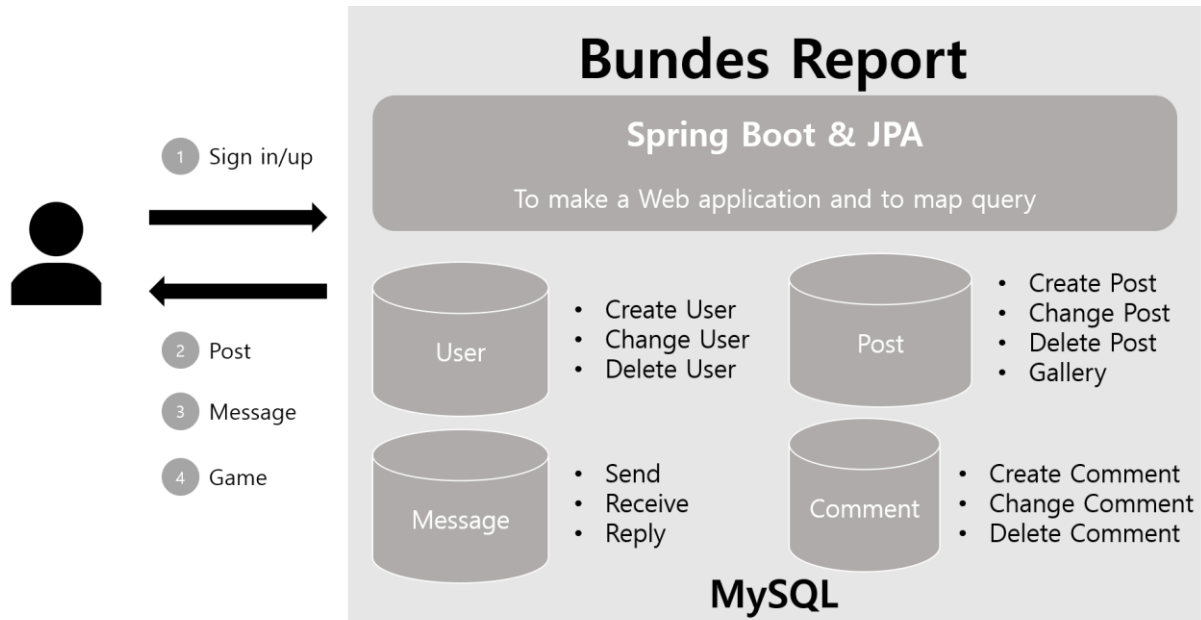
4. Skills

Our web project started from the Java project and this project based on Java Spring Boot and JPA. JPA was main data processing technology of our team. For Game Function, I used also JPA to map our query. Thanks to JPA, we didn't need to make a SQL query.

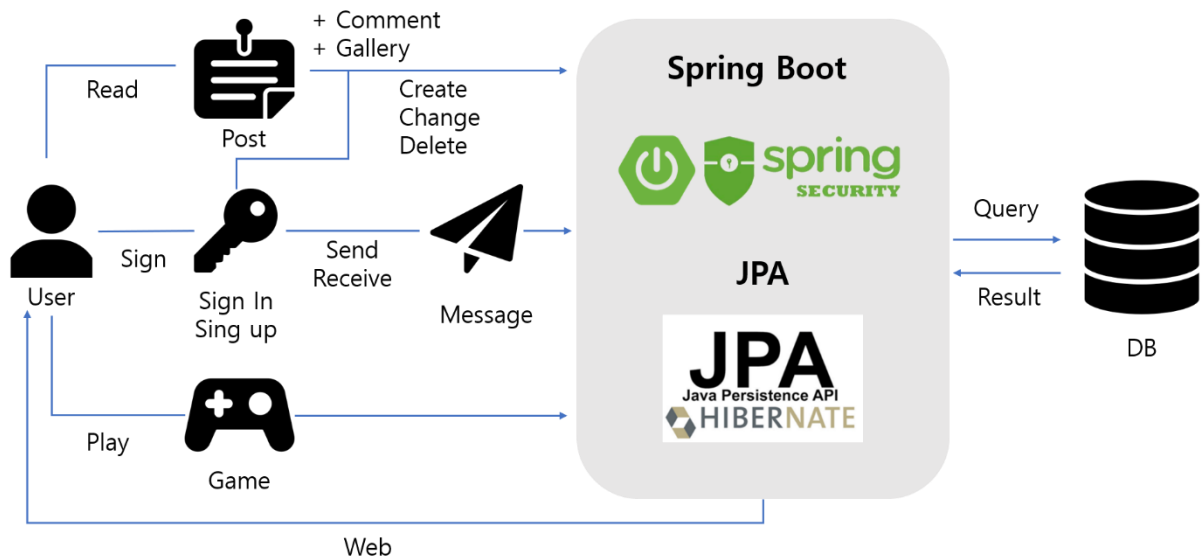
My main job was view and layout. We used Thymeleaf as a template engine. On the client side, code was written in HTML format and was used to dynamically draw DOMs. It was responsible for the process of receiving data and dynamically drawing that on DOM objects.

5. Implement method and Algorithm

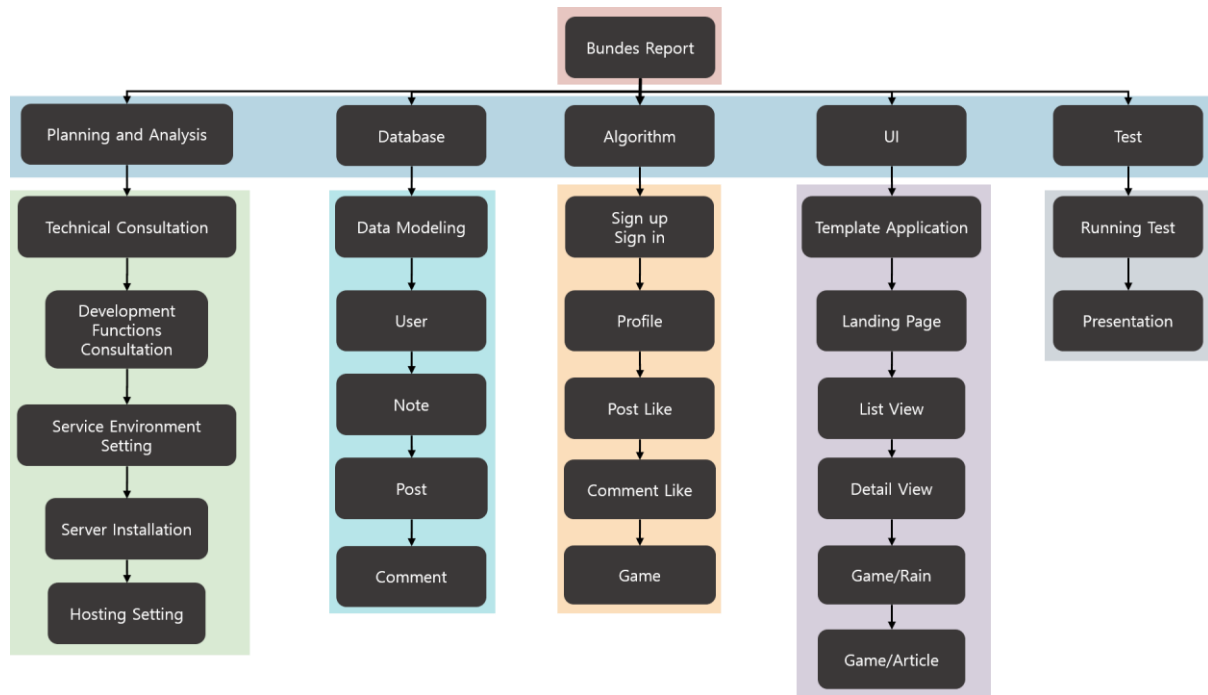
1) System Scenario



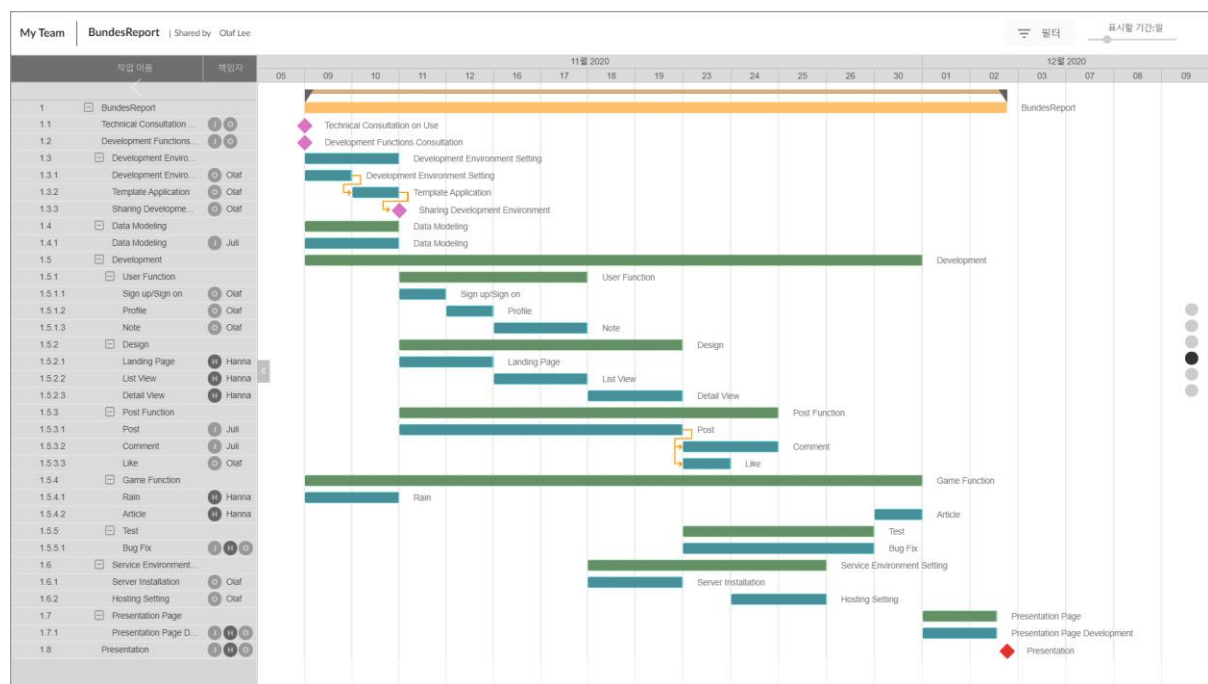
2) System Structure



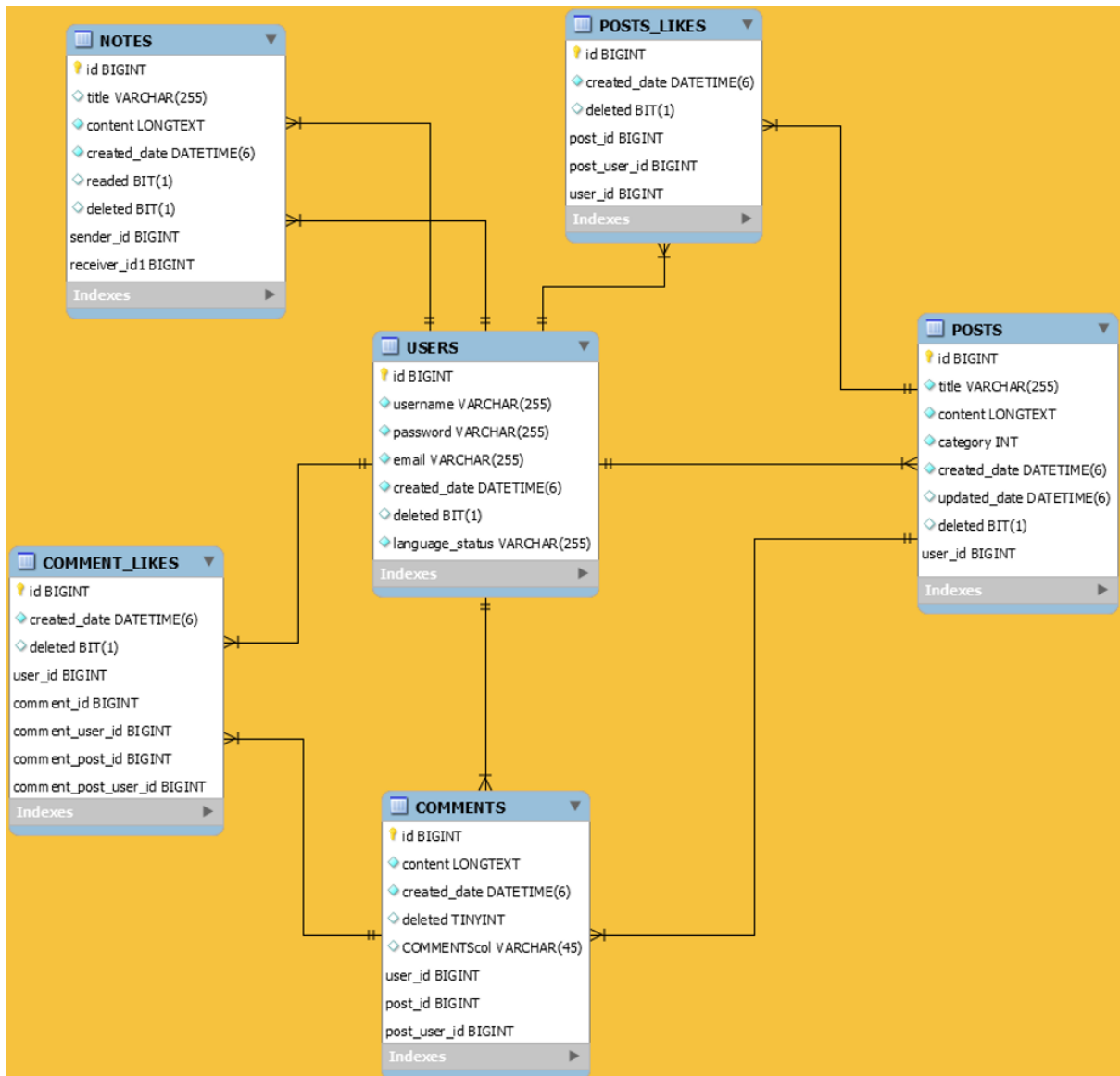
3) WBS



4) Gant Chart



5) UML Diagram



(1) Class Specification

① Class User

Class Name	User
Class Type	Class
Class Overview	Create User Model
Parent Class	
Attributes	

```
- serialVersionUID: long

// Serialization

- id: Long

// Primary Key

- deleted: String

- username: String

- password: String

- email: String

- createDate: LocalDateTime

// Save time automatedly when insert

- languageStatus: LanguageStatus

// Enum [ KO, DE ]

- posts: List<Post>

// Mapped by User

- postLikes: List<PostLike>

// Mapped by User

- comments: List<Comment>

// Mapped by User

- commentLikes: List<CommentLike>

// Mapped by User

- sendNotes: List<Note>

// Mapped by User

- receiveNotes: List<Note>
```

Operations

```

+ User(Long id, boolean deleted, String username, String password, String email,
LocalDateTime createdAt, LanguageStatus languageStatus)

// Create Entity Object

+ toUserForm(): UserForm

// Create User insert form

+ getAuthorities(): Collection<? extends GrantedAuthority>

// Spring Security Authority

+ getUsername(): String

+ isAccountNonExpired(): boolean

+ isCredentialsNonExpired(): boolean

+ isEnabled(): boolean

```

② Class Note

Class Name	Note
Class Type	Class
Class Overview	Create Note Model
Parent Class	
Attributes	
- id: Long // Primary Key - deleted: boolean - title: String - content: String - createdAt: LocalDateTime // Save time automatically when insert	

- readed: Boolean - sender: User // Join with sender_id - receiver: User // Join with receiver_id
Operations
+ Note(Long id, boolean deleted, String title, String content, LocalDateTime createdDate, boolean readed, User sender, User receiver) // Create Entity Object

③ Class Post

Class Name	Post
Class Type	Class
Class Overview	Create Post Model
Parent Class	
Attributes	
- id: Long // Primary Key - deleted: boolean - title: String - content: String - category: CategoryType - createdDate: LocalDateTime // Save time automatedly when insert - user: User	

<pre>// Join with user_id - comments: List<Comment> // Mapped by post - likes: List<PostLike> // Mapped by post - viewCount: int</pre>
Operations
<pre>+ Post(Long id, boolean deleted, String title, String content, CategoryType category, LocalDateTime createdAt, User user, int viewCount) // Create Entity Object + toPostForm(): PostForm // Create Post insert form + getUpdateModel(PostForm form): Post // Get Title and Content from form + getLikeCount(): int + getCommentCount(): int + getFirstImage(): String // show no-image when there is no image in gallery</pre>

④ Class PostLike

Class Name	PostLike
Class Type	Class
Class Overview	Count Like in Post
Parent Class	
Attributes	

<pre> - id: Long // Primary Key - deleted: boolean - user: User // Join with user_id - post: Post // Join with post_id - createdAt: LocalDateTime // Save time automatically when insert </pre>
Operations
<pre> + PostLike(Long id, boolean deleted, User user, Post post, LocalDateTime createdAt) // Create Entity Object </pre>

⑤ Class Comment

Class Name	Comment
Class Type	Class
Class Overview	Create Comment model
Parent Class	
Attributes	
<pre> - id: Long // Primary Key - deleted: boolean - content: String - createdAt: LocalDateTime // Save time automatically when insert </pre>	

<pre> - user: User // Join with user_id - post: Post // Join with post_id - likes: List<CommentLike> // Mapped by comment </pre>
Operations
<pre> + Comment (Long id, boolean deleted, String content, LocalDateTime createdDate, User user, Post post) // Create Entity Object + toCommentForm(): CommentForm // Create Comment insert form + getUpdateModel(CommentForm form): Comment // Get Content from form + hasMineLike(User user): boolean // Check pressed Like + getLikeCount(): int </pre>

⑥ Class CommentLike

Class Name	CommentLike
Class Type	Class
Class Overview	Count Like in Comment
Parent Class	
Attributes	
	- id: Long

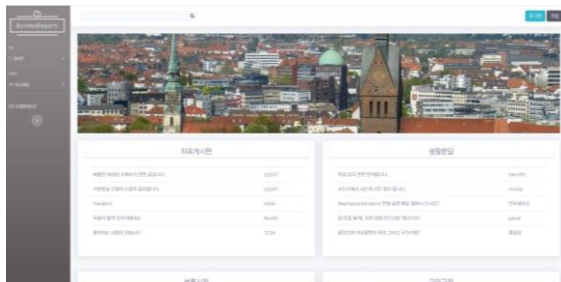
```
// Primary Key  
  
- deleted: Boolean  
  
- user: User  
  
// Join with user_id  
  
- comment: Comment  
  
// Join with comment_id  
  
- createdAt: LocalDateTime  
  
// Save time automatically when insert
```

Operations

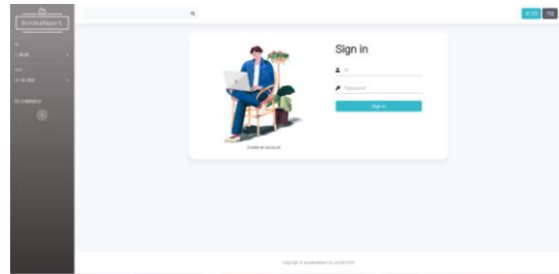
```
+ CommentLike (Long id, boolean deleted, User user, String content, LocalDateTime  
createdAt)  
  
// Create Entity Object
```

6) UI

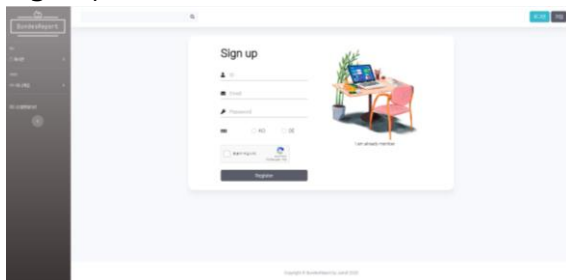
Main Page



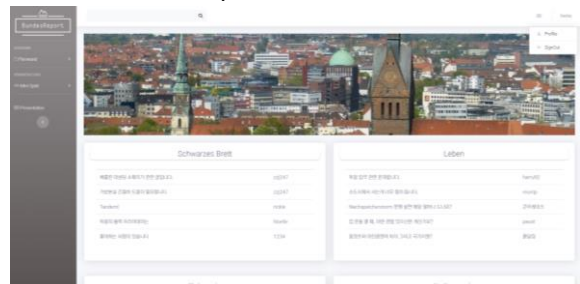
Sign in



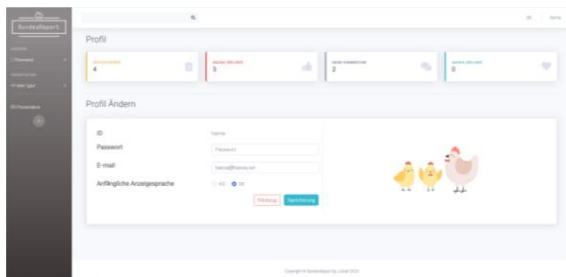
Sign up



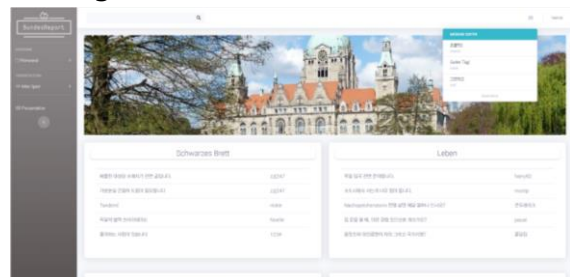
Profile Menu Expand



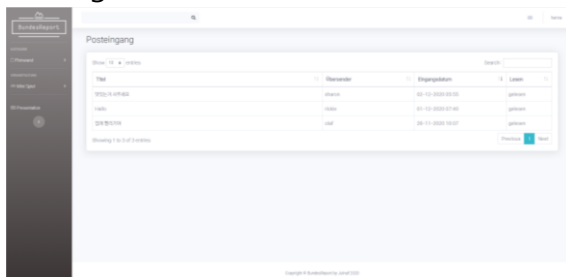
Profile



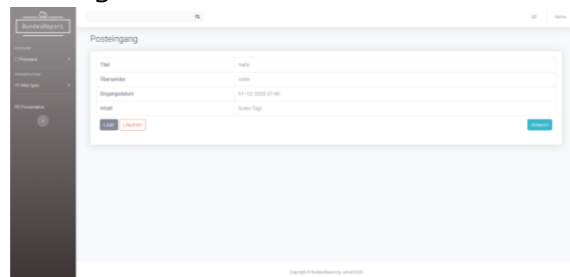
Message Center



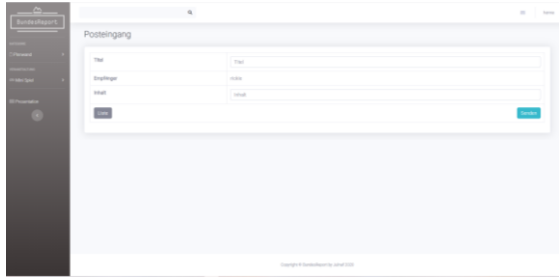
Message List



Message View



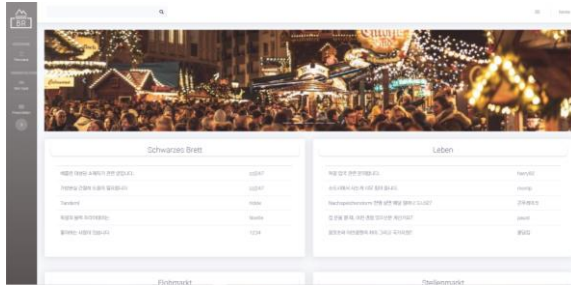
Message Send Form



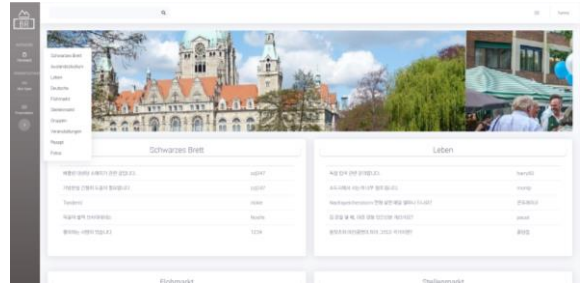
Menu Expand



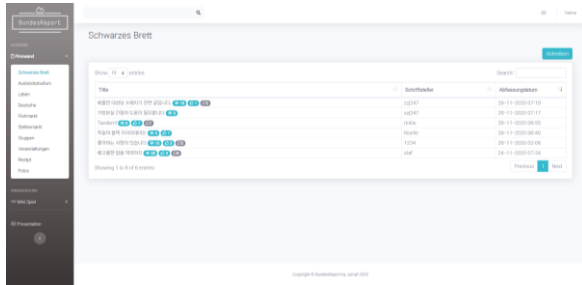
Small Menu



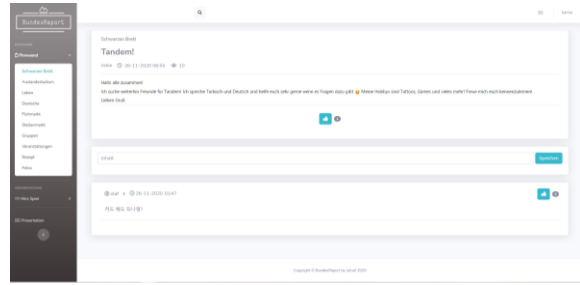
Small Menu Expand



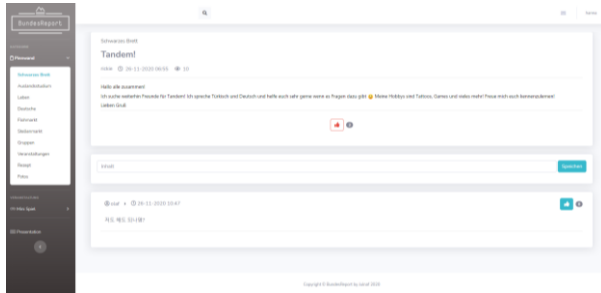
Post List



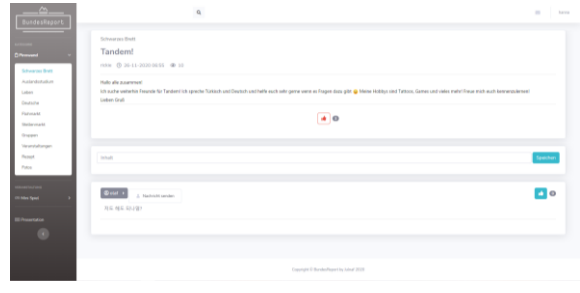
Post View



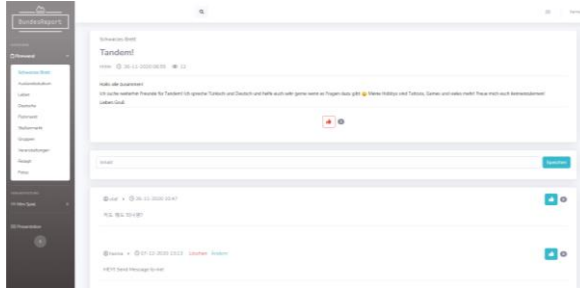
Pressed Like



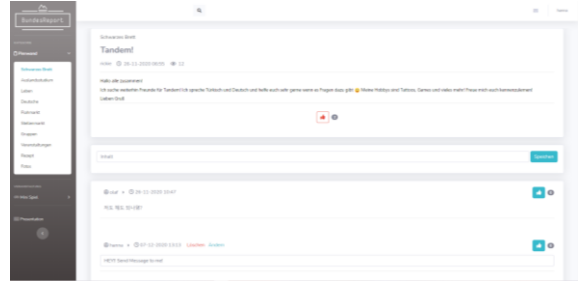
Message Send Menu



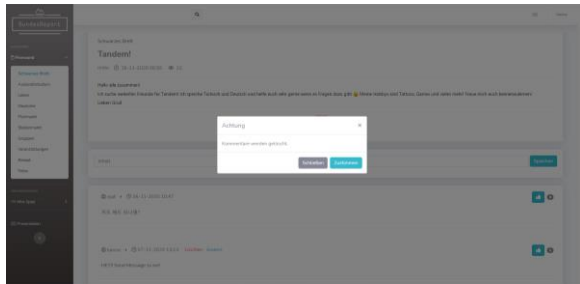
Writted Comment



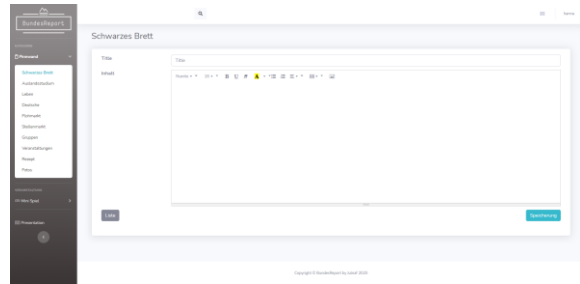
Change Comment



Delete Comment Modal



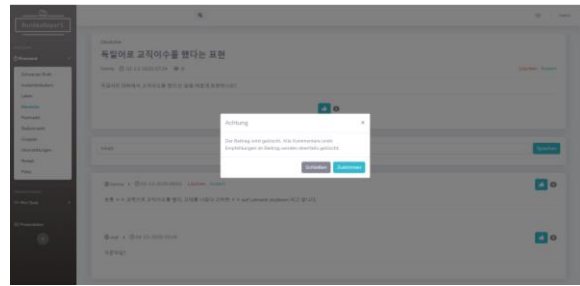
Write Post Form



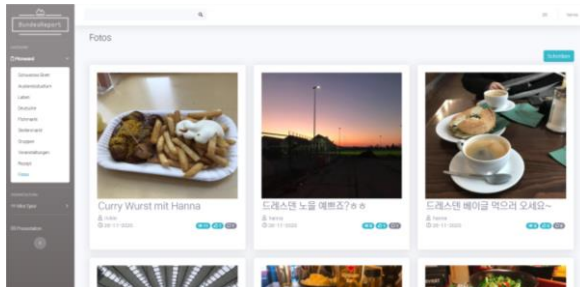
Writted Post



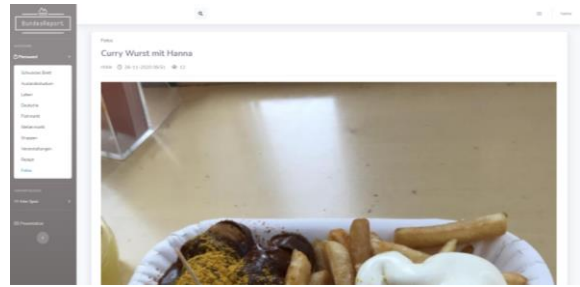
Delete Post Form



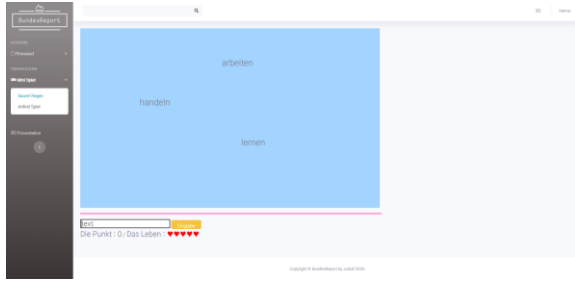
Gallery List



Gallery View



Rain Game



Article Game

