

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки  
Кафедра інформаційних систем та технологій

### **Лабораторна робота №3**

по дисципліні «Технології розроблення програмного  
забезпечення»

Тема: «Діаграма розгортання. Діаграма компонентів. Діаграма  
взаємодій та послідовностей»

Виконав:  
студент групи ІА-01  
Мартюк М.К.

Перевірив:  
вик. кафедри ІСТ  
Колеснік В. М.

Дата здачі: 03.11.2022

Захищено з балом \_\_\_\_\_

Київ 2022

**Тема:** Діаграма розгортання. Діаграма компонентів. Діаграма взаємодій та послідовностей.

**Завдання:**

1. Ознайомитися з короткими теоретичними відомостями.
2. Розробити діаграму розгортання для проектованої системи.
3. Розробити діаграму компонентів для проектованої системи.
4. Розробити діаграму послідовностей для проектованої системи.
5. Скласти звіт про виконану роботу.

**Хід роботи:**

1. Реалізувати частину функціональності системи, описану діаграмами з попередньої лабораторної роботи згідно обраної теми.

```
@RestController
@RequestMapping("api/v1/avatars")
@RequiredArgsConstructor
@Slf4j
public class ProfileAvatarController {

    4 usages
    private final ProfileAvatarService profileAvatarService;

    @GetMapping(
        value = "{profileId}",
        produces = {
            MediaType.IMAGE_JPEG_VALUE,
            MediaType.IMAGE_PNG_VALUE
        }
    )
    public ResponseEntity<byte[]> downloadAvatar(@PathVariable("profileId") Integer profileId) {
        final var avatar : byte[] = profileAvatarService.downloadAvatar(profileId);
        return ResponseEntity.ok(avatar);
    }

    @PostMapping("{profileId}")
    public ResponseEntity<String> uploadFile(@PathVariable("profileId") Integer profileId, MultipartFile file) {
        profileAvatarService.uploadAvatar(file, profileId);
        return ResponseEntity.ok( body: "Uploaded!");
    }

    @PutMapping("{profileId}")
    public ResponseEntity<String> updateAvatar(@PathVariable("profileId") Integer profileId, MultipartFile file) {
        profileAvatarService.updateAvatar(file, profileId);
        return ResponseEntity.ok( body: "Update avatar!");
    }

    @DeleteMapping("{profileId}")
    public ResponseEntity<String> deleteAvatar(@PathVariable("profileId") Integer profileId) {
        profileAvatarService.deleteAvatar(profileId);
        return ResponseEntity.ok( body: "Delete avatar!");
    }
}
```

⌵ @Service

⌵ @RequiredArgsConstructor

```
public class S3ProfileAvatarService implements ProfileAvatarService {
```

4 usages

```
private final ObjectStorageService objectStorageService;
```

2 usages

```
private final ProfileRepository profileRepository;
```

4 usages

```
@Value("${avatars.bucket-name}")
```

```
private String avatarsBucketName;
```

2 usages

⌵ @Override

```
public void uploadAvatar(MultipartFile file, Integer profileId) {
```

```
    if (file.isEmpty()) {
```

```
        throw new ResourceWriteException("File cannot be empty");
```

```
    }
```

```
    final var fileName :String = file.getOriginalFilename();
```

```
    final var objectName = UUID.randomUUID().toString().replaceAll(regex: "-", replacement: "")  
        + requireNonNull(fileName).substring(fileName.lastIndexOf(str: "."));
```

```
    if (isNull(FileTypeUtility.getFileType(file))) {
```

```
        throw new ResourceWriteException("Can write only jpg or png");
```

```
    }
```

```
    var profile : Profile = extractProfile(profileId);
```

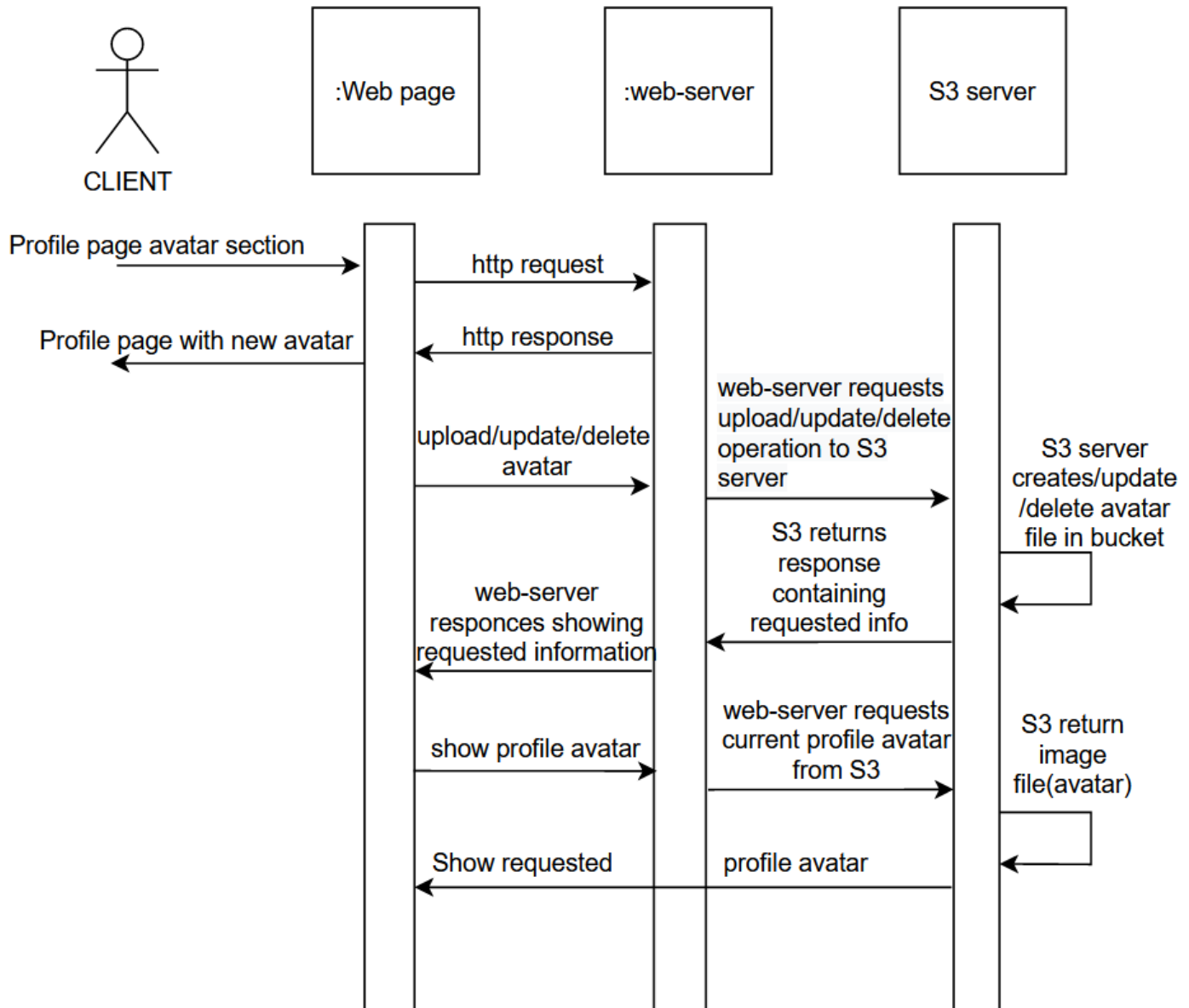
```
    profile.setAvatar(objectName);
```

```
    profileRepository.save(profile);
```

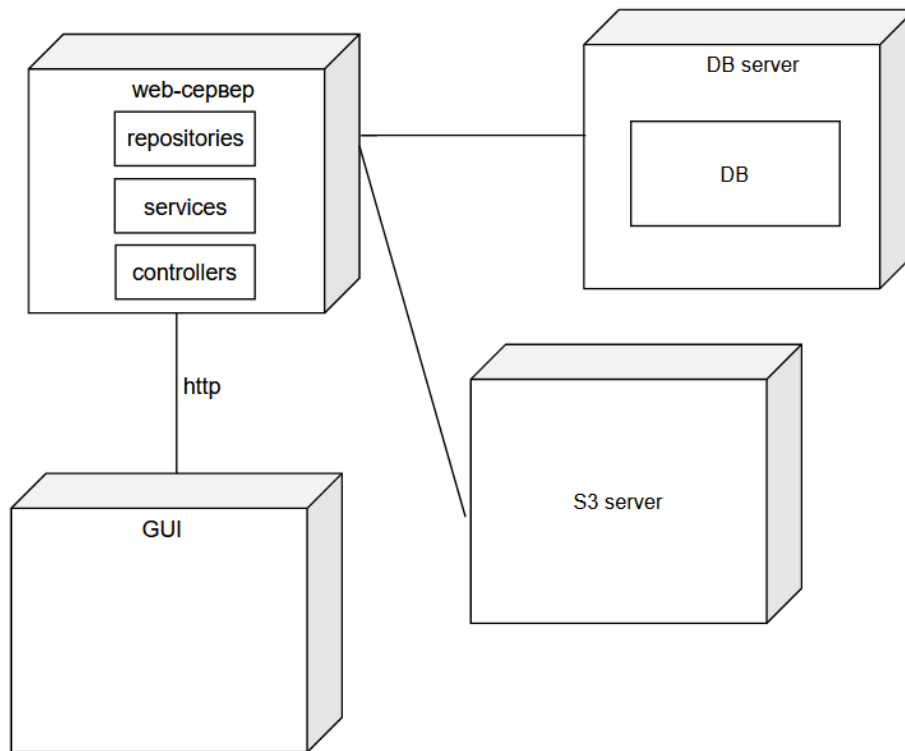
```
    objectStorageService.uploadFile(file, avatarsBucketName, objectName);
```

```
}
```

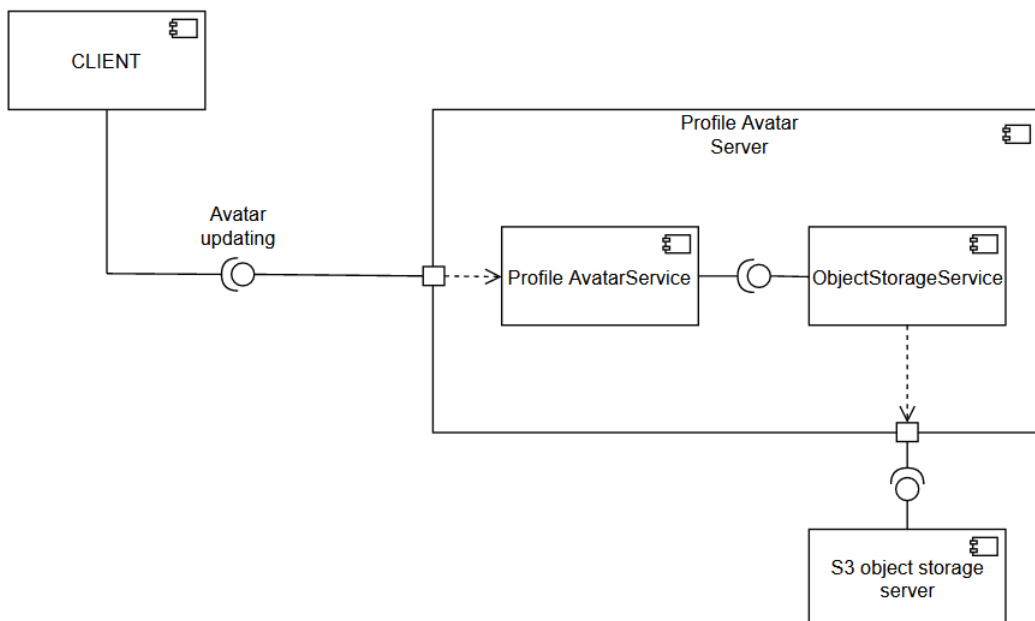
2. Спроекувати діаграму послідовностей для одного із процесів розроблюваної системи.



3. Спроекувати діаграму розгортання для розроблюваної системи.



4. Спроекувати діаграму компонентів для розроблюваної системи.



**Висновок:** на цій лабораторній роботі було розроблено діаграму розгортання для проектованої системи, діаграму компонентів для проектованої системи та діаграму послідовностей для проектованої системи.