**Київський національний університет імені тараса Шевченка**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Факультет комп’ютерних наук та технологій

Кафедра інформаційних систем

Звіт-ретроспектива

до лабораторної роботи №3

Фреймворк Java Spring

**Підготували:**

студенти групи ПІ21

Волков Олександр,

Кліщ Дмитро

Київ – 2022

**Звіт-ретроспектива**

1. **Які конкретні задачі планували вирішувати за допомогою цієї бібліотеки?**

За допомогою фреймворка Spring ставилось на меті створення веб-застосунку, який є аналогом соціального фото-сервісу Pinterest.

1. **Чому було обрано саме цю бібліотеку, а не аналоги?**

Даний фреймворк було використано, бо він має ряд можливостей: доступ до бази даних, проксі, реалізується аспектно-орієнтоване програмування, RPC та веб-інфраструктура MVC. Ці аспекти фреймворку Spring дозволяють створювати сучасні та зручні Java додатки. На сьогоднішній день Spring є найпоширенішим та має найбільш докладну документацію, тому і був обраний нами.

1. **Наскільки просто та зрозуміло було отримати, встановити, налаштувати та почати використовувати цю бібліотеку?**

Встановлення фреймворку не викликало особливих труднощів, оскільки нашою командою було обрано для розробки проєкту середовище Intellij Idea від JetBrains. Дане інтнегроване середовище містить зручну функцію встановлення додаткових фреймворків, в переліку яких також був і Spring. При налаштуванні дуже допомогла докладна документація, яка значно спростила цей процес. Після цих дій можна було одразу використовувати всі можливості Spring.

1. **Наскільки зрозумілою та корисною була документація бібліотеки?**

Документація бібліотеки повної мірою задовольнила всі наші потреби. Переважно використовувалась інформація з офіційного сайту. Для вирішення певних задач, наприклад, інтеграція GCS, були задіяні сторонні ресурси, бо запропоновані шляхи вирішення проблем, які постали перед нами з офіційного джерела здалися не найкращим рішенням наших задач.

1. **Наскільки було зрозуміло, як саме використовувати бібліотеку, які класи/методи/функції використовувати для вирішення поставлених задач?**

Оскільки, перед написанням коду для нашого проєкту ми чітко визначили завдання, що мають бути розв’язані за допомогою даного фреймворку, тому було повне розуміння, які можливості та функції Spring нам потрібні. Найбільш активно використовувались контейнери Spring IOC/Dependency Injection, що забезпечувало залежності між об’єктами та Spring MVC.

1. **Наскільки зручно було використовувати бібліотеку, чи не треба було писати багато надлишкового коду?**

Однією із головних причин вибору Spring була саме його простота та зручність. При написанні коду не виникало труднощів з його розумінням, оскільки лаконічний синтаксис дозволяв легко вирішувати поставлені задачі.

1. **Наскільки зрозумілою була поведінка класів/методів/функцій з бібліотеки?**

Дані питання були чітко розписані в офіційній документації, тому поведінка була врахована при проєктуванні архітектури програми. Зручний синтаксис підкреслював зрозумілість класів/методів/функцій.

1. **Наскільки зрозумілою була взаємодія між різними класами/ методами/ функціями цієї бібліотеки, а також взаємодія між бібліотекою та власним кодом?**

Взаємодія між класами/методами/функціями Spring є гарно продуманою та реалізованою, тому проблем не виникало. Але при створенні проєкту ми використовували сторонні технології, тому виникали проблеми з підключенням їхніх можливостей до Spring. Доводилося випробовувати багато варіантів, запропонованих на різних ресурсах.

1. **Чи виникали якісь проблеми з використанням бібліотеки? Чи вдалось їх вирішити, як саме?**

При використанні фреймворку найголовнішою проблемою була інтеграція сторонніх технологій. Виникала проблема з підключенням Google Cloud Storage. Інформація з офіційного сайту Spring є застарілою, тому допомогла документація GCS. Вирішення проблеми полягало в обході Spring і прямого підключення до проєкту даного сховища.

1. **Що хорошого можна сказати про цю бібліотеку, які були позитивні аспекти використання бібліотеки?**

* Простий та зрозумілий синтаксис.
* Докладна документація та багато ресурсів з інформацією.
* Можливість інтеграції сучасних технологій для роботи з даними.
* Простота підключення до проєкту.
* Простий та зручний модуль для налаштування безпеки застосунку Spring Security.

1. **Що поганого можна сказати про цю бібліотеку, які були негативні аспекти використання бібліотеки?**

Найголовніший мінус при використанні даного фреймворку – це його відсутність у загальнодоступній версії Intellij Idea. Потрібно було використовувати платну версію з повним доступом. Також негативним аспектом можна вважати те, що фреймворк має багато версій. Як наслідок в джерелах дуже багато застарілої інформації.

1. **Якби довелось вирішувати аналогічну задачу, але вже враховуючи досвід використання в цій лабораторній роботі, що варто було б робити так само, а що змінити? Можливо, використати інші бібліотеки, чи використати інші можливості цієї бібліотеки, чи інакше організувати код, чи ще щось?**

По-перше, після певної практики в даному фреймворці, з’явилися перевірені ресурси, які дають достовірну та оновлену інформацію щодо Spring. Це суттєво спрощує та пришвидшує написання якісного продукту. Також були визначені кращі методи для тестування додатку, що пришвидшує розробку і якість додатку.

**Використані технології у проєкті**

* Server side: Java Spring, Docker, Google Cloud Storage;
* Client side: JavaScript HTML CSS Bootstrap.

**Джерела**

1. Лонг Джош, Бастини Кеннет. Java в облаке. Spring Boot, Spring Cloud, Cloud Foundry. – СПб.: Питер, 2019 – 624 с.
2. <https://spring.io/>
3. <https://aws.amazon.com/ru/s3/>
4. <https://cloud.google.com/storage/docs>
5. <https://spring.io/projects/spring-cloud-gcp>
6. <https://spring.io/projects/spring-restdocs>
7. <https://www.youtube.com/watch?v=2tpLIcnf6zA>
8. <https://www.youtube.com/watch?v=tlshVRtbS_c>
9. <https://www.youtube.com/watch?v=5D27ZTih8ss>
10. <https://www.baeldung.com/>
11. <https://stackoverflow.com>