Geekbrains

**Дипломный проект: Веб-приложение "Арматурная карта учета"**

Программа: Java-разработчик

Специалист

Камешев Ануар Ерденович

Акколь

2025

### Введение

#### Актуальность темы

В современном мире информационные технологии играют ключевую роль в автоматизации бизнес-процессов. Организации стремятся внедрять цифровые решения для учета, планирования и управления ресурсами. В данном контексте автоматизация учета инвентаря становится важной задачей для повышения эффективности работы учреждений.

Медико-социальные учреждения, оказывающие услуги лицам с ограниченными возможностями, сталкиваются с необходимостью ведения точного учета инвентаря. Ведение записей вручную или в таблицах Excel приводит к ошибкам, потере данных и неудобствам в управлении ресурсами.

#### Цель проекта

Разработка веб-приложения "Арматурная карта", предназначенного для автоматизированного учета инвентаря в медико-социальном учреждении, с возможностью хранения информации о выданных предметах и формированием соответствующих отчетов.

#### Задачи проекта

1. Проанализировать существующие подходы к разработке веб-приложений.
2. Определить требования к системе.
3. Разработать архитектуру веб-приложения.
4. Реализовать систему с использованием Spring Boot.
5. Провести тестирование и развернуть приложение.

#### Методы исследования

* Анализ литературных источников и интернет-ресурсов.
* Использование современных технологий веб-разработки.
* Практическая реализация проекта с помощью Java Spring Boot.

#### Инструменты и технологии

* **Язык программирования**: Java
* **Фреймворк**: Spring Boot
* **Шаблонизатор**: Thymeleaf
* **База данных**: H2
* **Система управления версиями**: Git
* **Среда разработки**: IntelliJ IDEA

### Глава 1. Обзор веб-технологий и их эволюция

С момента появления интернета веб-приложения претерпели значительные изменения. Первые веб-страницы были статическими, использовали HTML и не имели динамического взаимодействия с пользователем. Позже появились серверные технологии, такие как CGI, PHP, JSP и ASP, позволяющие генерировать контент на лету.

Современные веб-приложения разделены на клиентскую и серверную части. Серверная часть обрабатывает запросы, управляет бизнес-логикой и базами данных, а клиентская часть отвечает за отображение информации. Одним из популярных решений для разработки серверной части является фреймворк Spring Boot.

В настоящее время активно применяются облачные технологии и микросервисные архитектуры. Однако для небольших проектов монолитная архитектура остается предпочтительной, так как она упрощает разработку и сопровождение.

### Глава 2. Архитектура Spring Boot

Spring Boot – это фреймворк для создания веб-приложений с минимальной конфигурацией. Он построен на основе Spring Framework и предоставляет удобные инструменты для быстрой разработки.

#### Основные принципы Spring Boot:

* **Inversion of Control (IoC)** – управление зависимостями передается контейнеру Spring.
* **Dependency Injection (DI)** – внедрение зависимостей позволяет гибко управлять объектами.
* **Auto-Configuration** – автоматическая настройка компонентов упрощает конфигурацию проекта.
* **Spring MVC** – модель "Контроллер-Представление-Модель" (Model-View-Controller) позволяет разделять логику приложения.

В проекте "Арматурная карта" используется **монолитная архитектура**, так как она лучше подходит для небольших веб-приложений. Все компоненты системы работают в одном приложении, что упрощает поддержку и развертывание.

### Глава 3. Разработка веб-приложения "Арматурная карта"

#### 3.1 Структура проекта

Проект состоит из следующих основных модулей:

* **Контроллеры** – обработка HTTP-запросов.
* **Сервисный слой** – бизнес-логика приложения.
* **Репозитории** – взаимодействие с базой данных.
* **Шаблоны Thymeleaf** – представления для отображения информации.

#### 3.2 Работа с базой данных

Используется встроенная база данных H2. Основные сущности:

* Inventory – инвентарь.
* Card – учетная карта.

Пример SQL-запроса для создания таблицы:

CREATE TABLE inventory (

id BIGINT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(255),

description TEXT,

quantity INT

);

#### 3.3 Реализация CRUD-операций

Приложение поддерживает операции создания, чтения, обновления и удаления (CRUD) записей инвентаря. Пример метода для получения списка всех элементов:

@GetMapping("/inventories/all")

public String getAllInventories(Model model) {

List<InventoryItem> items = inventoryService.findAll();

model.addAttribute("items", items);

return "inventory-list";

}

### Заключение

В ходе выполнения дипломного проекта было разработано веб-приложение "Арматурная карта" для учета инвентаря. В процессе работы были достигнуты следующие результаты:

* Проведен анализ веб-технологий и архитектуры Spring Boot.
* Разработана архитектура системы.
* Реализованы основные функции приложения.
* Проведено тестирование и отладка.

Разработанное приложение позволяет автоматизировать процесс учета инвентаря и может быть внедрено в медико-социальные учреждения. В дальнейшем возможно расширение функционала, интеграция с внешними системами и улучшение пользовательского интерфейса.

### Список используемой литературы

1. Крейг Уоллс. "Spring в действии". 5-е издание.
2. Документация Spring Boot: https://spring.io/projects/spring-boot
3. Java Persistence API (JPA): https://docs.oracle.com/javaee/7/tutorial/persistence-intro.htm
4. Thymeleaf: https://www.thymeleaf.org/documentation.html

### Приложения

(Скриншоты интерфейса, ER-диаграмма базы данных, примеры кода)