

**Московский авиационный институт
(Национальный исследовательский университет)**

Институт: «Компьютерные науки и прикладная математика»
Кафедра: 806 «Вычислительная математика и программирование»

**Лабораторная работа № 1
по курсу «Программная инженерия»**

Студенты: Чернобаев А. А.
Группа: М8О-114М-23
Преподаватели: Дзюба Д.В.
Миронов Е.С.

Москва
2024

1. Постановка задачи.

1. Установить инструменты из списка
 - Клиент Git
 - Текстовый редактор (рекомендуется Visual Studio Code)
 - Плагины к Visual Studio Code C4 DSL
2. Зарегистрироваться на github.com (если еще нет учетной записи)
3. Создать публичный репозиторий для выполнения практической работы у себя в аккаунте
4. Скопировать репозиторий https://github.com/DVDDemon/hl_mai_lab_00 с примерами задания
5. Создать файлы с описанием «архитектуры» согласно вашему варианту задания в Structurizr Lite.
6. Требования к диаграммам:
 - Должна быть контекстная диаграмма
 - Должна быть диаграмма контейнеров
 - Должна быть диаграмма развертывания
 - Должно быть несколько динамических диаграмм

Вариант задания - 2 (Магазин)

Приложение должно содержать следующие данные:

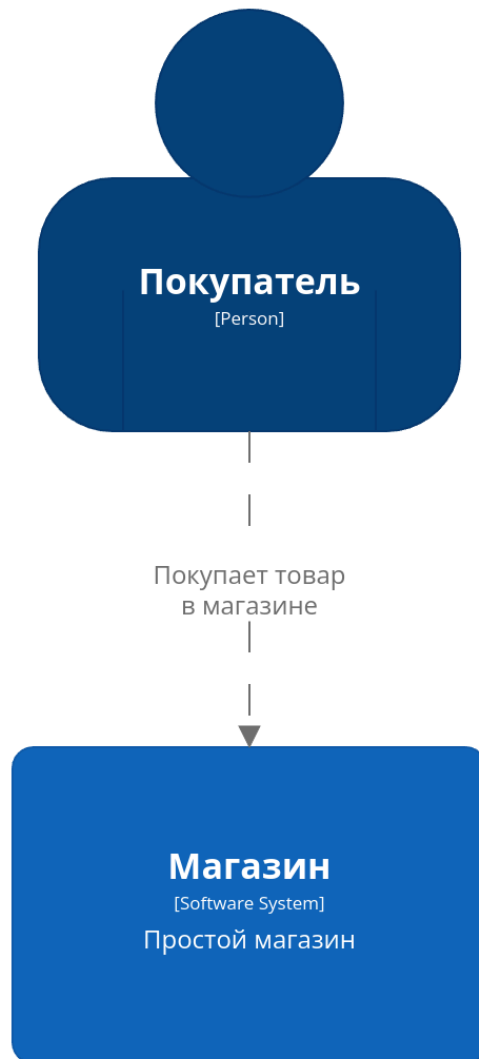
- Пользователь
- Товар
- Корзина

Реализовать API:

- Создание нового пользователя
- Поиск пользователя по логину
- Поиск пользователя по маске имя и фамилии
- Создание товара
- Получение списка товаров
- Добавление товара в корзину
- Получение корзины для пользователя

2. Диаграммы.

Контекстная диаграмма

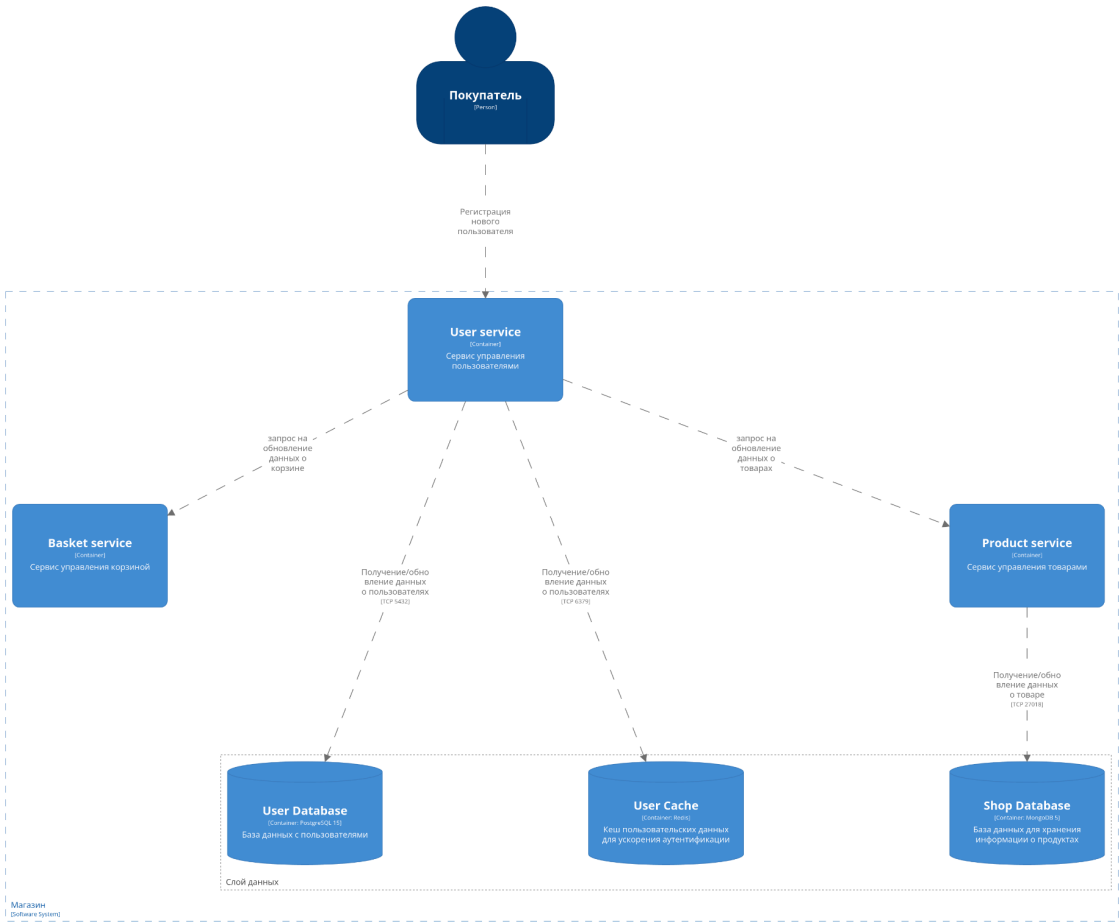


[System Landscape]

Monday, March 11, 2024 at 7:10 PM Moscow Standard Time

Рисунок 1. Контекстная диаграмма

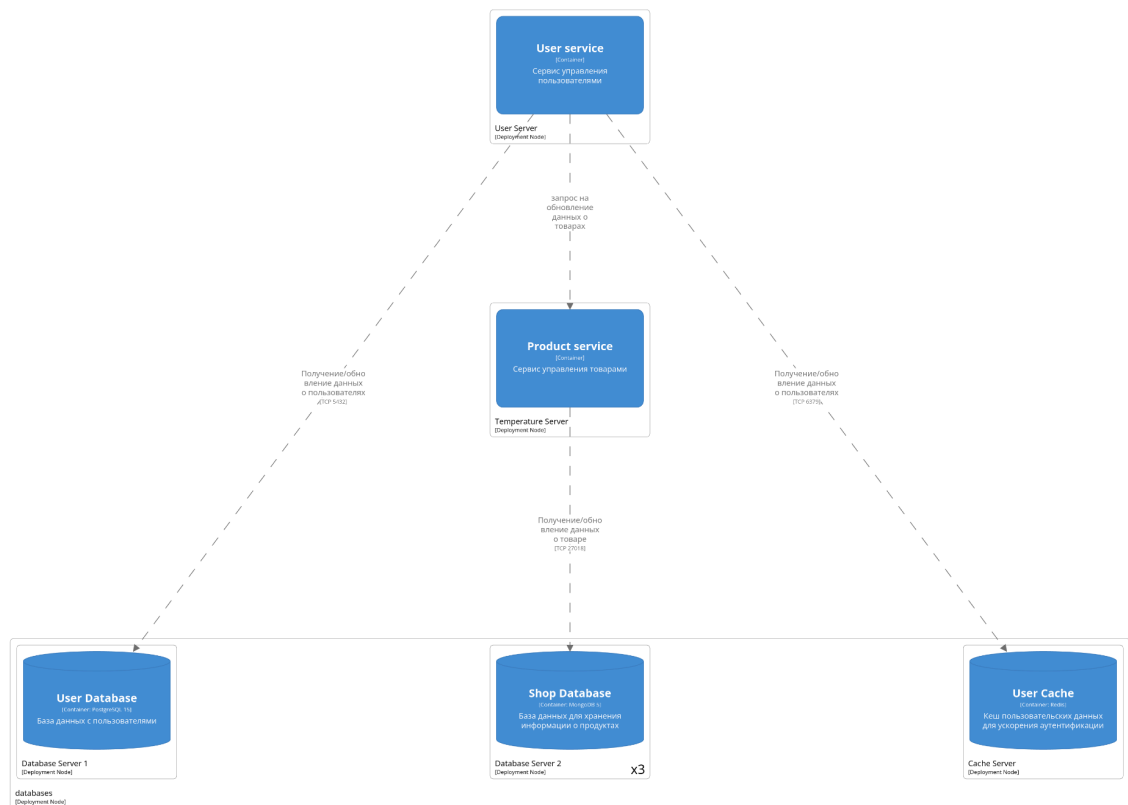
Диаграмма контейнеров



[Container] Магазин
Monday, March 11, 2024 at 7:10 PM Moscow Standard Time

Рисунок 2. Диаграмма контейнеров

Диаграмма развертывания



[Deployment] Магазин - Production
Monday, March 11, 2025 at 7:10 PM Moscow Standard time

Рисунок 3. Диаграмма развертывания

Динамические диаграммы



[Dynamic] Магазин

Добавление нового пользователя

Monday, March 11, 2024 at 7:10 PM Moscow Standard Time

Рисунок 4. Динамическая диаграмма 1



[Dynamic] Магазин

Удаление пользователя

Monday, March 11, 2024 at 7:10 PM Moscow Standard Time

Рисунок 5. Динамическая диаграмма 2



[Dynamic] Магазин

Сохранить данные о товарах

Monday, March 11, 2024 at 7:10 PM Moscow Standard Time

Рисунок 6. Динамическая диаграмма 3



[Dynamic] Магазин
Сохранить данные о корзине
Monday, March 11, 2024 at 7:10 PM Moscow Standard Time

Рисунок 7. Динамическая диаграмма 4

3. Код

```
workspace {  
  name "Магазин"  
  description "Простая система магазина с базовыми функциями для покупателей"  
  
  # включаем режим с иерархической системой идентификаторов  
  !identifiers hierarchical  
  
  !docs documentation  
  !adrs decisions  
  # Модель архитектуры  
  model {  
  
    # Настраиваем возможность создания вложенных групп  
    properties {  
      structurizr.groupSeparator "/"  
    }  
  
    # Описание компонент модели  
    user = person "Покупатель"  
    shop = softwareSystem "Магазин" {  
      description "Простой магазин"  
  
      user_service = container "User service" {  
        description "Сервис управления пользователями"  
      }  
  
      basket_service = container "Basket service" {  
        description "Сервис управления корзиной"  
      }  
  
      product_service = container "Product service" {
```

```

    description "Сервис управления товарами"
}

group "Слой данных" {
    user_database = container "User Database" {
        description "База данных с пользователями"
        technology "PostgreSQL 15"
        tags "database"
    }

    user_cache = container "User Cache" {
        description "Кеш пользовательских данных для ускорения аутентификации"
        technology "Redis"
        tags "database"
    }

    shop_database = container "Shop Database" {
        description "База данных для хранения информации о продуктах"
        technology "MongoDB 5"
        tags "database"
    }
}

user_service -> user_database "Получение/обновление данных о пользователях"
"TCP 5432"

user_service -> user_cache "Получение/обновление данных о пользователях" "TCP
6379"

user_service -> basket_service "запрос на обновление данных о корзине"

user_service -> product_service "запрос на обновление данных о товарах"

product_service -> shop_database "Получение/обновление данных о товаре" "TCP
27018"
}

```

user -> shop "Покупает товар в магазине"

user -> shop.user_service "Регистрация нового пользователя"

```
deploymentEnvironment "Production" {  
  deploymentNode "User Server" {  
    containerInstance shop.user_service  
  }  
}
```

```
deploymentNode "Product Server" {  
  containerInstance shop.product_service  
  properties {  
    "cpu" "4"  
    "ram" "256Gb"  
    "hdd" "4Tb"  
  }  
}
```

```
deploymentNode "databases" {  
  
  deploymentNode "Database Server 1" {  
    containerInstance shop.user_database  
  }  
  
  deploymentNode "Database Server 2" {  
    containerInstance shop.shop_database  
    instances 3  
  }  
  
  deploymentNode "Cache Server" {  
    containerInstance shop.user_cache  
  }  
}
```

```

    }

}

}

views {
    themes default

    properties {
        structurizr.tooltips true
    }

    !script groovy {
        workspace.views.createDefaultViews()

        workspace.views.views.findAll { it instanceof com.structurizr.view.ModelView }.each {
            it.enableAutomaticLayout() }
    }

    dynamic shop "UC01" "Добавление нового пользователя" {
        autoLayout
        user -> shop.user_service "Создать нового пользователя (POST /user)"
        shop.user_service -> shop.user_database "Сохранить данные о пользователе"
    }

    dynamic shop "UC02" "Удаление пользователя" {
        autoLayout
        user -> shop.user_service "Удалить нового пользователя (DELETE /user)"
        shop.user_service -> shop.user_database "Удалить данные о пользователе"
    }

    dynamic shop "UC03" "Сохранить данные о товарах" {
        autoLayout

```

```
shop.product_service -> shop.shop_database "Сохранить данные о товарах"  
}
```

```
dynamic shop "UC04" "Сохранить данные о корзине" {  
  autoLayout  
  shop.basket_service -> shop.user_service "Отправить данные о корзине"  
  shop.user_service -> shop.product_service "Отправить данные о корзине"  
  shop.product_service -> shop.shop_database "Сохранить данные о корзине"  
}
```

```
styles {  
  element "database" {  
    shape cylinder  
  }  
}  
}  
}
```