

Ministerul Educației și Cercetării al Republicii Moldova  
IP Centrul de Excelență în Informatică și Tehnologii Informaționale  
Catedra Informatică II

# ПРАКТИЧЕСКАЯ РАБОТА

## P2424-R

Uzun Timur  
Билет №15

24.12.2025

# Документация к прототипу настольного приложения

## «Управление медицинской клиникой»

### 1. Общая информация о проекте

**Название проекта:**

Управление медицинской клиникой (Medical Clinic Management)

**Тип приложения:**

Настольное (Desktop Application)

**Среда прототипирования:**

Figma

**Назначение приложения:**

Прототип предназначен для автоматизации работы медицинской клиники. Приложение позволяет управлять данными пациентов, врачей, консультаций, а также просматривать статистику и аналитические данные.

## **2. Цель и задачи приложения**

### **2.1 Цель проекта**

Создание интерактивного прототипа настольного приложения для медицинской клиники с поддержкой аутентификации пользователей, управления пациентами и визуализации статистики.

### **2.2 Основные задачи:**

- Аутентификация пользователей
- Управление списком пациентов
- Добавление, редактирование и удаление данных пациентов
- Просмотр статистики консультаций
- Обеспечение удобной навигации между экранами

## **3. Роли пользователей**

В приложении предусмотрены следующие роли:

### **3.1 Врач**

- Просмотр списка пациентов
- Редактирование информации о пациенте
- Просмотр статистики консультаций

### 3.2 Регистратор

- Добавление новых пациентов
- Редактирование и удаление пациентов
- Просмотр списка пациентов

## 4. Описание экранов прототипа

### 4.1 Экран входа (Login)

#### Назначение:

Аутентификация пользователя в системе.

#### Элементы интерфейса:

- Поле ввода имени (Name)
- Поле ввода IDNP
- Поле ввода номера телефона
- Кнопка «Sign In»
- Ссылка «I am not registered yet»

#### Функциональность:

- Проверка введённых данных
- Переход к главному меню после успешного входа

### 4.2 Главное меню (Main Menu)

#### Назначение:

Центральная панель управления приложением.

### **Элементы интерфейса:**

- Боковое меню:
  - Clients List
  - Edit
  - Statistics
  - Exit
- Список пациентов с указанием возраста
- Информационный блок о лечении

### **Функциональность:**

- Навигация между основными экранами
- Быстрый доступ к списку пациентов

## **4.3 Экран списка пациентов (All Clients List)**

### **Назначение:**

Отображение всех зарегистрированных пациентов.

### **Элементы интерфейса:**

- Таблица пациентов:
  - Номер
  - Имя
  - Возраст
  - Назначенный врач
- Кнопка редактирования для каждого пациента

- Кнопка Exit

#### **Функциональность:**

- Просмотр информации о пациентах
- Переход к экрану редактирования пациента

### **4.4 Экран редактирования пациента (Edit)**

#### **Назначение:**

Изменение данных пациента.

#### **Поля формы:**

- Имя
- Новый имя
- Врач
- Новый врач
- Диагноз
- Новый диагноз
- Номер телефона
- Новый номер

#### **Кнопки:**

- Save & Exit

#### **Функциональность:**

- Обновление информации пациента

- Сохранение изменений

#### **4.5 Экран статистики (Statistics)**

##### **Назначение:**

Визуализация статистических данных.

##### **Элементы интерфейса:**

- Круговые диаграммы:
  - Возраст пациентов
  - Длительность консультаций
- Процентное распределение данных
- Кнопка Exit

##### **Функциональность:**

- Анализ активности пациентов
- Наглядное представление статистики

### **5. Функциональные требования**

1. Экран входа с аутентификацией и проверкой данных
2. Главная панель управления
3. Табличный список пациентов с поиском
4. Возможность добавления, редактирования и удаления пациентов
5. Экран редактирования пациента с формой

6. Экран статистики с графиками
7. Навигация между экранами через меню и кнопки

## **6. Нефункциональные требования**

1. Чёткий визуальный стиль и единая цветовая схема
2. Интуитивно понятный пользовательский интерфейс
3. Логичная структура экранов
4. Полная интерактивность прототипа в Figma
5. Удобство навигации

## **7. UML-диаграммы**

### **7.1 Диаграмма вариантов использования (Use Case)**

#### **Акторы:**

- Врач
- Регистратор

#### **Основные варианты использования:**

- Вход в систему
- Просмотр списка пациентов
- Добавление пациента
- Редактирование пациента
- Удаление пациента
- Просмотр статистики



## 7.2 Диаграмма деятельности (Activity Diagram)

### Процесс: добавление пациента

1. Пользователь входит в систему
2. Переходит в раздел Clients List
3. Нажимает кнопку добавления
4. Заполняет форму
5. Сохраняет данные
6. Пациент добавляется в базу данных

## 8. Технические требования и Git

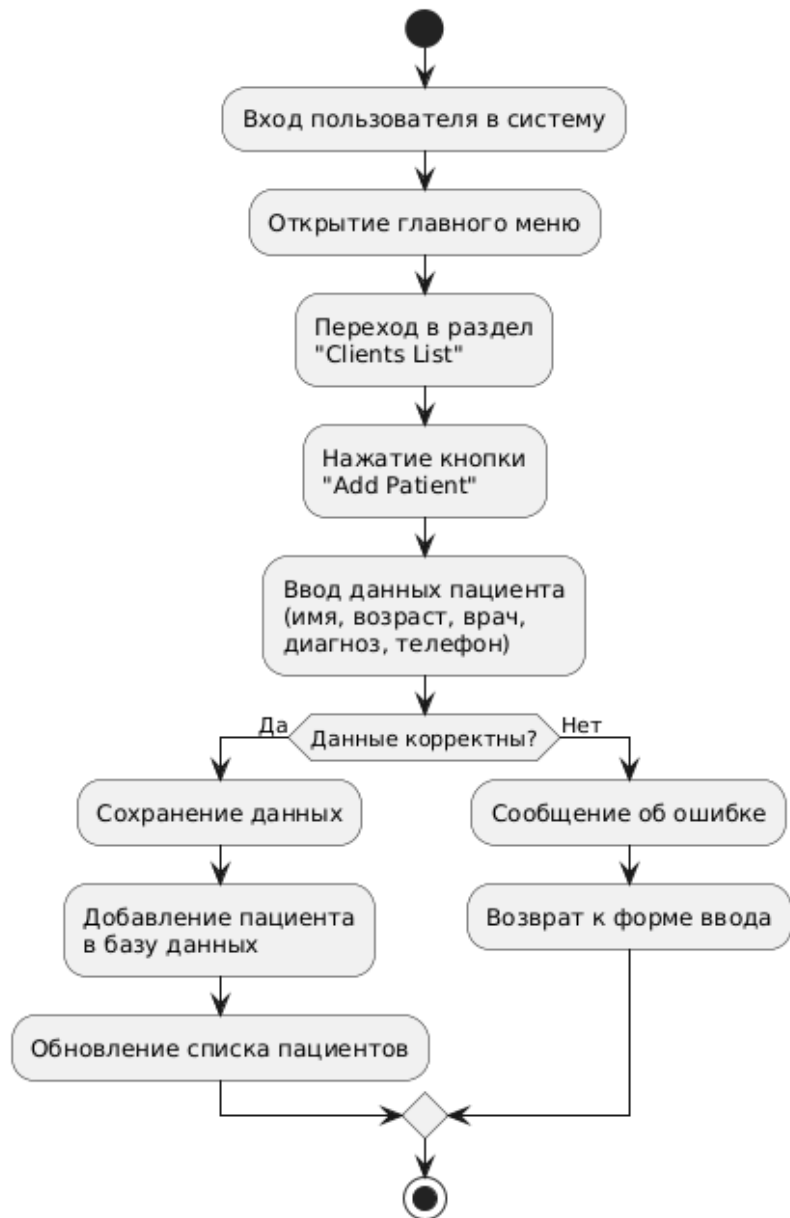
- Создан Git-репозиторий с именем **Medical Clinic**
- Экспортированы все экраны из Figma (PNG/SVG)
- Коммиты по этапам:
  - Login
  - Dashboard
  - Form
  - Statistics
- Репозиторий опубликован на GitHub

## 9. Заключение

В ходе работы был разработан интерактивный прототип настольного приложения для управления медицинской

клиникой. Прототип полностью соответствует функциональным и нефункциональным требованиям экзаменационного задания, демонстрирует логичную навигацию, удобный интерфейс и наглядную визуализацию данных.





# UML-диаграммы

## Диаграмма вариантов использования (Use Case Diagram)

### *Назначение диаграммы*

Диаграмма вариантов использования предназначена для описания функциональных возможностей системы и взаимодействия пользователей с приложением «Управление медицинской клиникой». Диаграмма отражает основные действия, которые могут выполнять пользователи системы в зависимости от их роли.

### Акторы системы

В системе предусмотрены два основных актора:

#### **1. Регистратор**

Отвечает за управление данными пациентов и первичную работу с системой.

#### **2. Врач**

Использует систему для работы с медицинской информацией пациентов и анализа статистики.

### Варианты использования для роли «Регистратор»

Регистратор имеет доступ к следующим функциям системы:

- Вход в систему
- Просмотр списка пациентов
- Добавление нового пациента

- Редактирование данных пациента
- Удаление пациента
- Поиск пациента в списке

Регистратор взаимодействует с системой для обеспечения актуальности и корректности данных пациентов.

## **Варианты использования для роли «Врач»**

Врач имеет доступ к следующим функциям системы:

- Вход в систему
- Просмотр списка пациентов
- Редактирование медицинских данных пациента
- Просмотр статистики консультаций и пациентов

Врач использует систему для анализа информации о пациентах и контроля лечебного процесса.

## **Общие варианты использования**

Некоторые функции доступны обоим акторам:

- Аутентификация (вход в систему)
- Просмотр списка пациентов
- Выход из системы

## **Вывод по диаграмме вариантов использования**

Диаграмма вариантов использования наглядно демонстрирует разграничение прав доступа между ролями «Регистратор» и «Врач», а также основные функции системы. Это позволяет обеспечить безопасность данных и логичную организацию работы пользователей в приложении.

# Диаграмма действий (Activity Diagram)

## Процесс «Добавление пациента в базу данных»

### Назначение диаграммы

Диаграмма действий предназначена для описания пошагового процесса добавления нового пациента в систему. Она отображает последовательность действий пользователя и реакцию системы на каждом этапе.

### Участник процесса

- Регистратор (основной исполнитель процесса)

### Описание процесса добавления пациента

Процесс добавления пациента включает следующие этапы:

#### 1. Начало процесса

Регистратор успешно входит в систему.

#### 2. Переход в раздел списка пациентов

Пользователь открывает экран «Clients List» через главное меню.

#### 3. Инициация добавления пациента

Регистратор выбирает действие «Добавить пациента».

#### 4. Отображение формы добавления

Система открывает форму ввода данных пациента.

## **5. Ввод данных пациента**

Регистратор вводит необходимые данные:

- a. Имя пациента
- b. Возраст
- c. Номер телефона
- d. Назначенного врача
- e. Диагноз

## **6. Проверка корректности данных**

Система проверяет заполненность и корректность введенных данных.

## **7. Принятие решения**

- a. Если данные введены корректно — процесс продолжается
- b. Если обнаружена ошибка — пользователь возвращается к редактированию формы

## **8. Сохранение данных**

При успешной проверке система сохраняет информацию о пациенте в базе данных.

## **9. Обновление списка пациентов**

Новый пациент отображается в общем списке.

## **10. Завершение процесса**

Процесс добавления пациента завершается.

## **Вывод по диаграмме действий**

Диаграмма действий демонстрирует логичную и последовательную структуру процесса добавления пациента в базу данных. Она позволяет наглядно представить взаимодействие пользователя и системы, а также точки проверки данных, что повышает надёжность и удобство использования приложения.



## Общий вывод по UML-диаграммам

Использование UML-диаграмм в проекте позволяет:

- Формализовать функциональные требования системы
- Чётко определить роли пользователей
- Визуализировать ключевые бизнес-процессы
- Повысить качество проектирования приложения