# **Вариант 1**

**Задание**

На основе технического задания, написать графический интерфейс для добавления и редактирования заявок.

Используйте уместные элементы ввода в формах в соответствии типа *параметра*.

Заполнение и изменение данных оформите в отдельных окнах (страницах по желанию)

Дополнительно: реализовать валидацию данных и обработки ошибок

**Техническое задание**

**Общие сведения**

*Наименование проекта:* Разработка программного модуля для учета заявок на ремонт автомобилей.

*Заказчик:* Автосервис «АвтоТранс».

*Исполнитель:* Сервисный центр «ТехноПолис».

**Функциональные требования**

**Возможность добавления заявок** в базу данных с указанием следующих параметров:

- Номер заявки;

- Дата добавления;

- Вид авто;

- Модель авто;

- Описание проблемы;

- ФИО клиента;

- Номер телефона;

- Статус заявки (новая заявка, в процессе ремонта, завершена).

**Возможность редактирования заявок:**

- Изменение этапа выполнения (готова к выдаче, в процессе ремонта, ожидание автозапчастей);

- Изменение описания проблемы;

- Изменение ответственного за выполнение работ.

**Возможность отслеживания статуса заявки:**

- Отображение списка заявок;

- Получение уведомлений о смене статуса заявки;

- Поиск заявки по номеру или по параметрам.

**Возможность назначения ответственных за выполнение работ:**

- Добавление автомеханика к заявке;

- Отслеживание состояния работы и получение уведомлений о ее завершении;

- Автомеханик может добавлять комментарии на форме заявки и фиксировать информацию о заказанных автозапчастях и материалах.

**Расчет статистики работы отдела обслуживания:**

- Количество выполненных заявок;

- Среднее время выполнения заявки;

- Статистика по типам неисправностей.

**Нефункциональные требования**

**Безопасность:**

- Логин и пароль для доступа к приложению;

- Доступ к данным должен быть ограничен в зависимости от роли пользователя.

**Удобство использования:**

- Простой и интуитивный интерфейс;

- Информативные уведомления и подсказки.

# **Вариант 2**

**Задание**

На основе технического задания, написать графический интерфейс для добавления и редактирования заявок.

Используйте уместные элементы ввода в формах в соответствии с типом параметра.

Заполнение и изменение данных оформите в отдельных окнах (страницах по желанию).

Дополнительно: реализовать валидацию данных и обработку ошибок.

**Техническое задание**

**Общие сведения**

Наименование проекта: Разработка программного модуля для учета заявок на ремонт бытовой техники.

Заказчик: Сервисный центр «БытТех».

Исполнитель: Компания «ТехноСервис».

**Функциональные требования**

Возможность добавления заявок в базу данных с указанием следующих параметров:

* Номер заявки;
* Дата добавления;
* Тип техники (холодильник, стиральная машина, телевизор и т.д.);
* Модель техники;
* Описание проблемы;
* ФИО клиента;
* Номер телефона;
* Статус заявки (новая заявка, в процессе ремонта, завершена).

**Возможность редактирования заявок:**

* Изменение этапа выполнения (ожидание запчастей, в процессе ремонта, готово к выдаче);
* Изменение описания проблемы;
* Изменение ответственного за выполнение работ.

**Возможность отслеживания статуса заявки:**

* Отображение списка заявок;
* Получение уведомлений о смене статуса заявки;
* Поиск заявки по номеру или по параметрам.

**Возможность назначения ответственных за выполнение работ:**

* Добавление мастера к заявке;
* Отслеживание состояния работы и получение уведомлений о ее завершении;
* Мастер может добавлять комментарии на форме заявки и фиксировать информацию о заказанных запчастях.

**Расчет статистики работы отдела обслуживания:**

* Количество выполненных заявок;
* Среднее время выполнения заявки;
* Статистика по типам неисправностей.

**Нефункциональные требования**

**Безопасность:**

* Логин и пароль для доступа к приложению;
* Доступ к данным должен быть ограничен в зависимости от роли пользователя.

**Удобство использования:**

* Простой и интуитивный интерфейс;
* Информативные уведомления и подсказки.

# **Вариант 3**

На основе технического задания, написать графический интерфейс для добавления и редактирования заявок.

Используйте уместные элементы ввода в формах в соответствии с типом параметра.

Заполнение и изменение данных оформите в отдельных окнах (страницах по желанию).

Дополнительно: реализовать валидацию данных и обработку ошибок.

**Техническое задание**  
**Общие сведения**  
 Наименование проекта: Разработка программного модуля для учета заявок на обслуживание IT-оборудования.

Заказчик: Компания «IT-Сервис».

Исполнитель: Компания «ТехноСофт».

**Функциональные требования**

**Возможность добавления заявок в базу данных с указанием следующих параметров:**

* + Номер заявки;
  + Дата добавления;
  + Тип оборудования (компьютер, сервер, принтер и т.д.);
  + Модель оборудования;
  + Описание проблемы;
  + ФИО клиента;
  + Номер телефона;
  + Статус заявки (новая заявка, в процессе ремонта, завершена).

**Возможность редактирования заявок:**

* + Изменение этапа выполнения (ожидание запчастей, в процессе ремонта, готово к выдаче);
  + Изменение описания проблемы;
  + Изменение ответственного за выполнение работ.

**Возможность отслеживания статуса заявки:**

* + Отображение списка заявок;
  + Получение уведомлений о смене статуса заявки;
  + Поиск заявки по номеру или по параметрам.

**Возможность назначения ответственных за выполнение работ:**

* + Добавление инженера к заявке;
  + Отслеживание состояния работы и получение уведомлений о ее завершении;
  + Инженер может добавлять комментарии на форме заявки и фиксировать информацию о заказанных запчастях.

**Расчет статистики работы отдела обслуживания:**

* + Количество выполненных заявок;
  + Среднее время выполнения заявки;
  + Статистика по типам неисправностей.

**Нефункциональные требования**

**Безопасность:**

* Логин и пароль для доступа к приложению;
* Доступ к данным должен быть ограничен в зависимости от роли пользователя.

**Удобство использования:**

* Простой и интуитивный интерфейс;
* Информативные уведомления и подсказки.

# **Вариант 4**

На основе технического задания, написать графический интерфейс для добавления и редактирования заявок.

Используйте уместные элементы ввода в формах в соответствии с типом параметра.

Заполнение и изменение данных оформите в отдельных окнах (страницах по желанию).

Дополнительно: реализовать валидацию данных и обработку ошибок.

**Техническое задание**  
**Общие сведения**  
Наименование проекта: Разработка программного модуля для учета заявок на ремонт медицинского оборудования.  
Заказчик: Медицинский центр «Здоровье».  
Исполнитель: Компания «МедТехСервис».

**Функциональные требования**

**Возможность добавления заявок в базу данных с указанием следующих параметров:**

* + Номер заявки;
  + Дата добавления;
  + Тип оборудования (рентген, УЗИ, кардиограф и т.д.);
  + Модель оборудования;
  + Описание проблемы;
  + ФИО клиента;
  + Номер телефона;
  + Статус заявки (новая заявка, в процессе ремонта, завершена).

**Возможность редактирования заявок:**

* + Изменение этапа выполнения (ожидание запчастей, в процессе ремонта, готово к выдаче);
  + Изменение описания проблемы;
  + Изменение ответственного за выполнение работ.

**Возможность отслеживания статуса заявки:**

* + Отображение списка заявок;
  + Получение уведомлений о смене статуса заявки;
  + Поиск заявки по номеру или по параметрам.

**Возможность назначения ответственных за выполнение работ:**

* + Добавление инженера к заявке;
  + Отслеживание состояния работы и получение уведомлений о ее завершении;
  + Инженер может добавлять комментарии на форме заявки и фиксировать информацию о заказанных запчастях.

**Расчет статистики работы отдела обслуживания:**

* + Количество выполненных заявок;
  + Среднее время выполнения заявки;
  + Статистика по типам неисправностей.

**Нефункциональные требования**

**Безопасность:**

* Логин и пароль для доступа к приложению;
* Доступ к данным должен быть ограничен в зависимости от роли пользователя.

**Удобство использования:**

* Простой и интуитивный интерфейс;
* Информативные уведомления и подсказки.

# **Вариант 5**

**Задание**

На основе технического задания, написать графический интерфейс для добавления и редактирования заявок.

Используйте уместные элементы ввода в формах в соответствии типа *параметра*.

Заполнение и изменение данных оформите в отдельных окнах (страницах по желанию)

Дополнительно: реализовать валидацию данных и обработки ошибок

**Техническое задание**

**Общие сведения**

*Наименование проекта*: Система управления логистикой онлайн-магазина.

*Заказчик*: Интернет-магазин «QuickBuy».

*Исполнитель*: Компания «LogiSoft».

**Функциональные требования**

**Возможность добавления заказов** в базу данных с указанием следующих параметров:

- Номер заказа;

- Дата добавления;

- Тип товара;

- Вес 1 единицы;

- Объём 1 единицы;

- Расстояние (от склада до заказчика);

- Количество товара;

- ФИО заказчика;

- Номер телефона;

- Статус заказа (в пути, на складе, доставлен).

1. Управление заказами:
   * Добавление/редактирование заказов;
   * Автоматическое резервирование товаров на складе;
2. Учет товаров:
   * Отслеживание остатков на складе;
   * Уведомления о низком запасе;
   * Автоматический заказ у поставщиков.
3. Расчет стоимости доставки:
   * Учет веса, габаритов, расстояния;
4. Аналитика:
   * Отчеты по продажам;
   * Расчет рентабельности товаров;

**Нефункциональные требования**

**Безопасность:**

- Логин и пароль для доступа к приложению;

- Доступ к данным должен быть ограничен в зависимости от роли пользователя.

**Удобство использования:**

- Простой и интуитивный интерфейс;

- Информативные уведомления и подсказки.

# **Вариант 6**

**Задание**

На основе технического задания, написать графический интерфейс для добавления и редактирования студентов в систему.

Используйте уместные элементы ввода в формах в соответствии типа *параметра*.

Заполнение и изменение данных оформите в отдельных окнах (страницах по желанию)

Дополнительно: реализовать валидацию данных и обработки ошибок

**Техническое задание**

**Общие сведения**

*Наименование проекта*: Система управления учебными курсами.

*Заказчик*: Университет «Инновация».

*Исполнитель*: IT-компания «EduTech Solutions».

**Функциональные требования**

**Возможность создания курсов** с параметрами и сохранение их в базу данных:

* Название курса;
* Преподаватель;
* Время проведения;
* Список студентов;
* Максимальное количество студентов;

**Функционал программы**.

* Регистрация студентов:
* Запись на курсы;
* Автоматическое распределение по группам при превышении лимита.
* Формирование расписания:
* Контроль пересечений времени занятий.

**Аналитика:**

* Расчет нагрузки преподавателей;
* Отчеты по заполняемости групп.

**Нефункциональные требования**

**Безопасность:**

- Логин и пароль для доступа к приложению;

- Доступ к данным должен быть ограничен в зависимости от роли пользователя.

**Удобство использования:**

- Простой и интуитивный интерфейс;

- Информативные уведомления и подсказки.

# **Вариант 7**

**Задание**

На основе технического задания, написать графический интерфейс для добавления, бронирования и выдачи спортивного инвентаря.

Используйте уместные элементы ввода в формах в соответствии типа *параметра*.

Заполнение и изменение данных оформите в отдельных окнах (страницах по желанию)

Дополнительно: реализовать валидацию данных и обработки ошибок

**Техническое задание**

**Общие сведения**

* Наименование проекта: Платформа «SportRent».
* Заказчик: Сеть спортивных магазинов «ActiveLife».
* Исполнитель: Команда «DevSport».

**Функциональные требования**

**Возможность создания каталога инвентаря** с параметрами и сохранение их в базу данных:

* Вид инвентаря;
* Описание инвентаря;
* Размер инвентаря
* Цена бронирования (на час);
* Наличие инвентаря;
* Наличие скидок;

1. Каталог инвентаря:
   * Добавление позиций (велосипеды, лыжи, снаряжение) с описанием;
   * Фильтрация по типу, размеру, цене.
2. Бронирование:
   * Выбор даты и времени аренды;
   * Автоматическая проверка доступности;
   * Формирование электронного договора.
3. Расчет стоимости:
   * Учет сезонных наценок (например, +20% зимой);
   * Скидки для постоянных клиентов.
4. Уведомления:
   * Напоминание о возврате инвентаря;
   * Оповещение о повреждениях.

**Нефункциональные требования**

**Безопасность:**

- Логин и пароль для доступа к приложению;

- Доступ к данным должен быть ограничен в зависимости от роли пользователя.

**Удобство использования:**

- Простой и интуитивный интерфейс;

- Информативные уведомления и подсказки.