**Руководство пользователя системы «…»**

[***Введение*** 3](#_Toc199701490)

[***1. Назначение системы*** 3](#_Toc199701491)

[***2. Технические требования*** 4](#_Toc199701492)

[**2.1. Системные требования** 4](#_Toc199701493)

[**2.2. Структура проекта** 4](#_Toc199701494)

[***3. Инструкция по использованию*** 5](#_Toc199701495)

[**3.1. Запуск системы** 5](#_Toc199701496)

[**3.2. Основные функции** 5](#_Toc199701497)

[***4. Используемые методы*** 8](#_Toc199701498)

[***5. Обработка ошибок*** 8](#_Toc199701499)

[***6. Техподдержка*** 9](#_Toc199701500)

**Руководство пользователя и проектная документация системы «[Название системы]»**

# ***Введение***

**Название системы**: [Например, «HotelManager» для информационной системы гостиницы]

**Версия**: 1.0

**Предметная область**: [Например, «Управление гостиничным бизнесом»]

**Описание**: Документ содержит описание функционального назначения системы, инструкции по её использованию, перечень основных методов с параметрами и технические требования. Он предназначен для пользователей (администраторов, сотрудников, гостей) и разработчиков, поддерживающих систему.

# ***1. Назначение системы***

Система предназначена для автоматизации процессов в предметной области [указать область]. Она предоставляет следующие ключевые возможности:

* [Функция 1]: [Краткое описание]. Авторизация пользователей: Обеспечивает безопасный вход для пользователей с ролями «Администратор», «Руководитель», «Пользователь» и «Гость».
* [Функция 2]: [Краткое описание].
* [Функция 3]: [Краткое описание].

# ***2. Технические требования***

## **2.1. Системные требования**

* **Операционная система**: [Например, Windows 10 или выше].
* **Среда разработки**: [Например, Visual Studio 2022].
* **СУБД**: [Например, SQLite, MySQL].
* **Дополнительные библиотеки**: [Например, .NET Framework 4.8, ADO.NET].

## **2.2. Структура проекта**

Система организована в соответствии с принципом единственной ответственности: каждая сущность представлена отдельным классом или компонентом. Основные директории и файлы:

* **Forms/Auth.cs**: Форма авторизации пользователей.
* **Forms/ChangePassword.cs**: Форма для смены пароля.
* **Forms/MainForm.cs**: Главная форма с меню для разных ролей.
* **UserControls/UserControl1.cs**: Компонент для управления учетными записями.
* **HotelDataSet.xsd**: Определение структуры базы данных и адаптеров.

# ***3. Инструкция по использованию***

## **3.1. Запуск системы**

1. Установите Visual Studio 2022 и .NET Framework 4.8.
2. Откройте файл HotelManager.sln в Visual Studio.
3. Нажмите «Пуск» для запуска приложения.
4. Откроется форма Auth, где нужно ввести логин и пароль или выбрать режим «Гость».

## **3.2. Основные функции**

**Функция 1: Авторизация**

* **Описание**: Позволяет пользователю войти в систему, указав логин и пароль. Проверяется корректность данных, статус блокировки и дата последнего входа.
* **Как выполнить**:
  1. В форме авторизации введите логин в поле «Логин» и пароль в поле «Пароль».
  2. Нажмите кнопку «Войти».
  3. При неверных данных отобразится сообщение об ошибке (например, «Неверный пароль!»).
  4. После 3 неверных попыток учетная запись блокируется.

**Функция 2: Смена пароля**

* **Описание**: Позволяет пользователю изменить пароль, если он использует стандартный пароль или по необходимости.
* **Как выполнить**:
  1. При первом входе с паролем «new\_password» автоматически открывается форма смены пароля.
  2. Введите текущий пароль, новый пароль и подтверждение нового пароля.
  3. Нажмите кнопку «Изменить».
  4. При успешной смене отобразится сообщение: «пароль успешно изменен!».

**Функция 3: Управление пользователями (для администратора)**

* **Описание**: Позволяет администратору просматривать, редактировать и добавлять учетные записи пользователей.
* **Как выполнить**:
  1. Войдите как администратор (role\_id = 1).
  2. В меню главной формы выберите «users».
  3. Откроется таблица с данными пользователей.
  4. Отредактируйте данные (логин, пароль, роль, статус блокировки) и нажмите «Сохранить».
  5. Для добавления нового пользователя [указать шаги, если реализовано].

# ***4. Используемые методы***

**Класс Auth (Форма авторизации)**

1. **loginButton\_Click**
   * **Описание**: Проверяет логин, пароль, статус блокировки и дату последнего входа. Управляет попытками входа и открывает форму смены пароля или главную форму.
   * **Параметры**:
     + sender (object): Объект, вызвавший событие (кнопка).
     + e (EventArgs): Аргументы события.
   * **Возвращаемое значение**: Нет.
   * **Пример вызова**: Автоматически вызывается при нажатии кнопки «Войти».
2. **guestLabel\_Click**
   * **Описание**: Открывает главную форму для роли «Гость» (role\_id = 4).
   * **Параметры**:
     + sender (object): Объект, вызвавший событие (метка).
     + e (EventArgs): Аргументы события.
   * **Возвращаемое значение**: Нет.
   * **Пример вызова**: Автоматически вызывается при клике на метку «Гость».

# ***5. Обработка ошибок***

* Все формы и компоненты используют MessageBox для отображения ошибок и уведомлений.
* Ошибки ввода данных (например, пустые поля, неверный пароль) сопровождаются информативными сообщениями.
* Системные ошибки обрабатываются через try-catch, предотвращая аварийное завершение приложения.

*Пример*:

* При неверном пароле: «Неверный пароль! осталось попыток: {число}».
* При ошибке сохранения данных: «Ошибка при обновлении: {описание ошибки}».

# ***6. Техподдержка***

По вопросам использования системы или ошибок обращайтесь к администратору системы:

* Электронная почта: [[email@example.com](mailto:email@example.com)].
* Телефон: [указать номер].