**Функциональная часть приложения ShpetsBailifDesktop**

**1. Введение**

Приложение **ShpetsBailifDesktop** предназначено для управления деятельностью приставов внутри компании. Оно предоставляет возможность управления данными о должниках, кредиторах, исполнительных производствах, сотрудниках, отделах и других сущностях, связанных с деятельностью приставов. Приложение будет иметь разграничение доступа по ролям, что позволит контролировать доступ к определенным функциям и данным.

**2. Структура приложения**

**2.1. Модули приложения**

Приложение будет разделено на несколько модулей, каждый из которых отвечает за определенный функционал:

* **Модуль аутентификации и авторизации**: Отвечает за вход пользователей в систему и управление ролями.
* **Модуль управления сотрудниками**: Позволяет управлять данными о сотрудниках, их должностях и отделах.
* **Модуль управления должниками и кредиторами**: Позволяет управлять данными о должниках и кредиторах.
* **Модуль управления исполнительными производствами**: Позволяет управлять данными об исполнительных производствах, конфискованном имуществе и документах.
* **Модуль отчетности**: Позволяет генерировать отчеты по различным параметрам.

**2.2. Структура классов**

Приложение будет реализовано с использованием паттерна MVVM (Model-View-ViewModel), что обеспечит разделение логики представления и бизнес-логики.

* **Модели (Models)**: Классы, представляющие сущности базы данных (например, Employee, Debtor, EnforcementProceeding и т.д.).
* **Представления (Views)**: XAML-файлы, отвечающие за отображение данных пользователю (например, MainWindow.xaml, EmployeeView.xaml и т.д.).
* **ViewModel**: Классы, отвечающие за бизнес-логику и связывание данных между View и Model (например, EmployeeViewModel, DebtorViewModel и т.д.).

**2.3. Структура проекта**

Проект будет организован следующим образом:

* **ShpetsBailifDesktop.Core**: Содержит основные модели и интерфейсы.
* **ShpetsBailifDesktop.Data**: Содержит классы для работы с базой данных (репозитории, контекст базы данных).
* **ShpetsBailifDesktop.Services**: Содержит сервисы, отвечающие за бизнес-логику (например, сервис аутентификации, сервис управления сотрудниками).
* **ShpetsBailifDesktop.ViewModels**: Содержит ViewModel-классы.
* **ShpetsBailifDesktop.Views**: Содержит XAML-файлы представлений.
* **ShpetsBailifDesktop.App**: Основной проект, содержащий файлы приложения (например, App.xaml, MainWindow.xaml).

**3. Реализация приложения**

**3.1. Аутентификация и авторизация**

* **Модель**: User (содержит поля UserID, Username, PasswordHash, Role).
* **ViewModel**: LoginViewModel (отвечает за логику входа в систему).
* **View**: LoginView (форма входа в систему).
* **Сервис**: AuthenticationService (отвечает за проверку учетных данных и авторизацию пользователя).

**3.2. Управление сотрудниками**

* **Модель**: Employee, Position, Department.
* **ViewModel**: EmployeeViewModel (отвечает за логику управления сотрудниками).
* **View**: EmployeeView (форма для управления сотрудниками).
* **Сервис**: EmployeeService (отвечает за операции с базой данных, связанные с сотрудниками).

**3.3. Управление должниками и кредиторами**

* **Модель**: Debtor, Creditor.
* **ViewModel**: DebtorViewModel, CreditorViewModel.
* **View**: DebtorView, CreditorView.
* **Сервис**: DebtorService, CreditorService.

**3.4. Управление исполнительными производствами**

* **Модель**: EnforcementProceeding, SeizedProperty, Document, Court.
* **ViewModel**: EnforcementProceedingViewModel, SeizedPropertyViewModel, DocumentViewModel.
* **View**: EnforcementProceedingView, SeizedPropertyView, DocumentView.
* **Сервис**: EnforcementProceedingService, SeizedPropertyService, DocumentService.

**3.5. Отчетность**

* **ViewModel**: ReportViewModel.
* **View**: ReportView.
* **Сервис**: ReportService (отвечает за генерацию отчетов).

**4. Работа с проектом**

**4.1. Добавление новой функциональности**

1. **Создание модели**: Добавьте новый класс в папку ShpetsBailifDesktop.Core/Models.
2. **Создание ViewModel**: Добавьте новый класс в папку ShpetsBailifDesktop.ViewModels.
3. **Создание View**: Добавьте новый XAML-файл в папку ShpetsBailifDesktop.Views.
4. **Создание сервиса**: Добавьте новый класс в папку ShpetsBailifDesktop.Services.
5. **Регистрация сервиса**: Зарегистрируйте сервис в App.xaml.cs или другом подходящем месте.

**4.2. Изменение существующей функциональности**

1. **Найдите соответствующий ViewModel**: Измените логику в ViewModel.
2. **Обновите View**: Если необходимо, обновите XAML-файл представления.
3. **Измените сервис**: Если необходимо, обновите сервис, отвечающий за бизнес-логику.

**4.3. Чтение кода**

1. **Модели**: Ознакомьтесь с классами в папке ShpetsBailifDesktop.Core/Models.
2. **ViewModel**: Ознакомьтесь с классами в папке ShpetsBailifDesktop.ViewModels.
3. **View**: Ознакомьтесь с XAML-файлами в папке ShpetsBailifDesktop.Views.
4. **Сервисы**: Ознакомьтесь с классами в папке ShpetsBailifDesktop.Services.

**5. Заключение**

Приложение **ShpetsBailifDesktop** предоставляет удобный интерфейс для управления деятельностью приставов внутри компании. Разделение функционала по ролям обеспечивает безопасность и контроль доступа. Структура проекта организована таким образом, чтобы упростить добавление новой функциональности и изменение существующей.