МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологий

Кафедра Программной инженерии

Специальность 1-40 01 01 Программное обеспечение информационных технологий

Направление специальности 1-40 01 01 10 Программное обеспечение информационных технологий (программирование интернет-изданий)

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**КУРСОВОГО ПРОЕКТА:**

по дисциплине «Объектно-ориентированные технологии программирования и стандарты проектирования»

Тема Программное средство «Управление интернами»

Исполнитель

студент 2 курса группы 4 Городилов Михаил Петрович

(Ф.И.О.)

Руководитель работы ? А.А. Рауба

(учен. степень, звание, должность, подпись, Ф.И.О.)

Курсовой проект защищен с оценкой

Председатель Пацей Н.В.

(подпись)

Минск 2020

**Содержание**

[ВВЕДЕНИЕ 3](#_Toc41148751)

[1 Анализ прототипов, литературных источников и формирование требований к проектируемому программному средству 4](#_Toc41148752)

[2 Анализ требований к программному средству и разработка функциональных требований 5](#_Toc41148753)

[3 Проектирование программного средства 6](#_Toc41148754)

[3.1 Структуры и взаимоотношения между классами 6](#_Toc41148755)

ВВЕДЕНИЕ

Приложение «Управление интернами» – это рабочий инструмент для организации стажировок в компании. В частности, крупные IT компании имеют потребность в подобном ПО, которое позволило бы автоматизировать процессы отслеживания стажировок.

Цель данного курсового проекта – проектирование и разработка данного приложения.

При выполнении курсового проекта будут использованы принципы и приемы ООП. Для разработки графической составляющей данного продукта используется технология Windows Presentation Foundation (WPF).

В первом разделе представлен анализ прототипов и литературных источников.

1 Анализ прототипов, литературных источников и формирование требований к проектируемому программному средству

Немаловажным этапом в разработке программного продукта является аналитический обзор прототипов и литературных источников.

На сегодняшний день можно встретить очень малое количество доступных программных решений, разработанных для управления стажировками, т.к. в основном компании имеют отдельное ПО для учёта сотрудников, без функционала для интернов, т.к. стажировки в IT компаниях раньше не были востребованы. За неимением аналогов специализированного ПО, рассмотрим программу «Connecteam», рассчитанную на сотрудников:

«Connecteam» ⎯ это ПО для администрирования персонала. Программа может быть использована администраторами и самими сотрудниками.

Проанализировав часть программы, рассчитанную на повышение квалификации сотрудников, можно выделить её основные плюсы и минусы.

Основные плюсы:

* Возможность создавать базу знаний компании;
* Наличие интерактивных опросов для проверки знаний;
* Возможность получить результаты развития сотрудников в графиках.

Основные минусы:

* Обобщённость продукта по сферам деятельности;
* Нет возможности ограничить доступность для интернов;
* Отсутствие реализации стажировок как таковых;

На основе проведенного анализа и с учетом требований, указанных в задании

на курсовое проектирование, сформированы следующие требования:

* Разработать бизнес логику приложения;
* Реализовать программный интерфейс;
* Реализовать нотификацию пользователей.

Для разработки данного ПС были выбраны следующие средства разработки:

* Интегрированная среда разработки Microsoft Visual Studio 2017;
* Программная платформа .NET Framework 4.6;
* Язык программирования C#;
* Технология Windows Presentation Foundation (WPF);
* Расширяемый язык разметки XAML;
* Технология ADO.NET Entity Framework.

2 Анализ требований к программному средству и разработка функциональных требований

Описание функциональности ПС:

* Возможность авторизации и регистрации;
* Наличие интерфейса для управления заявками на создание аккаунта;
* Возможность просматривать список стажировок и поиск по нему;
* Возможность менять картинку профиля;
* Возможность выставления временных рамок стажировки;
* Наличие интерфейса для частичного редактирования стажировки;
* Возможность назначать встречу со студентом;
* Возможность завершения стажировки;
* Наличие оповещения в программе;
* Наличие оповещения через email;

Вариант использования: в компании появляется нужда в интернах, данное ПС могло бы предоставить необходимые для менеджера и интерна данные о стажировке, упростило работу менеджеров с интернами, если основное ПО компании сложно масштабировать под интернатуры.

Спецификация функциональных требований:

1. Авторизация должна быть безопасна для пользователей, т.е. пароли будут храниться в БД в зашифрованном виде;
2. Основная часть функций будет реализовывать работу с БД, чтобы данные функции не блокировали пользовательский интерфейс при отработке, требуется сделать их асинхронными;
3. Функциональность оповещений по Email будет нуждаться в секретной информации (логин и пароль для smtp сервера), поэтому данная информация будет храниться в конфигурационных файлах;

3 Проектирование программного средства

3.1 Структуры и взаимоотношения между классами

Приложение построено на основе паттерна MVVM. Паттерн MVVM (Model-View-ViewModel) позволяет отделить логику приложения от визуальной части (представления). Данный паттерн является архитектурным, то есть он задает общую архитектуру приложения [<https://metanit.com/sharp/wpf/22.1.php>].



Рисунок 3.1 – Визуальное представление MVVM

Ниже приведены диаграммы классов ПС. Более подробные диаграммы приведены в графических материалах.

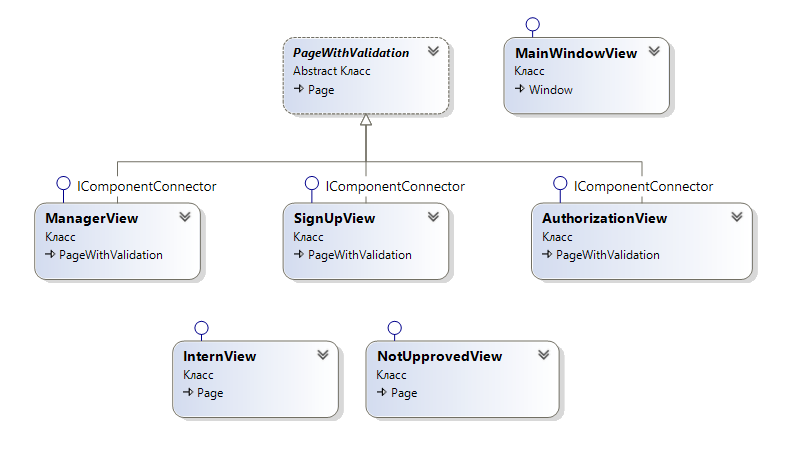


Рисунок 3.2 – Диаграмма классов представлений

Класс MainView наследуется от встроенного класса Window и представляет основное окно приложения.

Остальные классы наследуются от встроенного класса Page, который представляет страницы приложения, либо от реализованного класса PageWithValidation, который служит для валидации полей страницы.

Класс AuthorizationView реализует страницу авторицации.

Класс SignUpView реализует страницу регистрации.

Класс NotUpprovedView реализует страницу для неподтвержённых пользователей.

Класс ManagerView реализует страницу администрирования.

Класс InternView реализует страницу конкретного интерна.

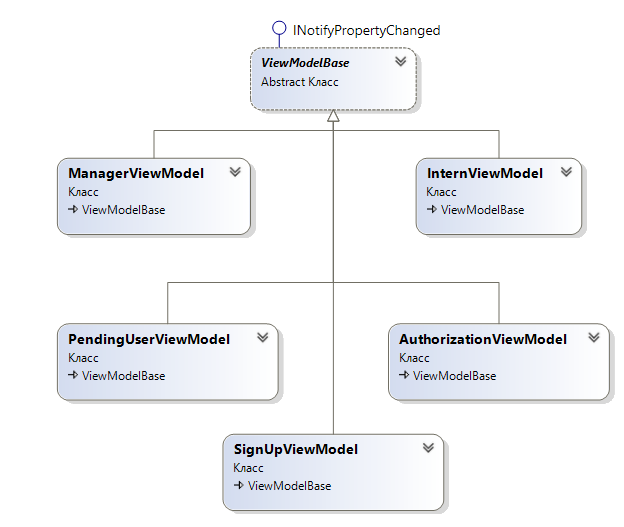


Рисунок 3.3 – Диаграмма классов моделей представлений

Класс ViewModelBase реализует интерфейс INotifyPropertyChanged, и является родительским для каждого класса моделей представлений.

Класс AuthorizationViewModel реализует логику авторицазии пользователя приложения.

Класс SignUpViewModel реализует логику регистрации пользователя.

Класс ManagerViewModel реализует логику управления интернами: добавление пользователей, поиск по списку интернов, переход на страницу интерна.

Класс PendingUserViewModel реализует модель представления пользователя, ожидающего подтверждение аккаунта.

Класс InternViewModel реализует логику для страницы интерна: изменение даты окончания, окончание, назначение встреч.

Класс NotUpprovedViewModel реализует единственную функцию – возврат на страницу авторизации.

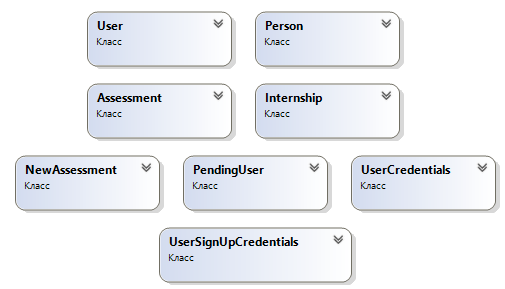


Рисунок 3.4 – Диаграмма классов моделей

Классы User, Person, Assessment, Internship являются представлениями Entity Framework, но их не достаточно, поэтому введены дополнительные классы для некоторых моделей представления. Подробное содержание классов так же находится в графических материалах.

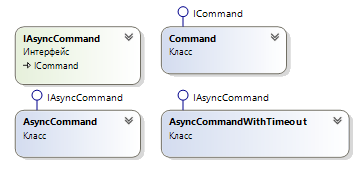


Рисунок 3.5 – Диаграмма классов моделей

Интерфейс ICommand является встроенным, требует реализации ивента CanExecuteChanged, и функций CanExecute, Execute. Т.е. паттерна комманда[].

Интерфейс IAsyncCommand наследуется от ICommand и добавляет метод ExecuteAsync.

Ниже представленны вспомогательные классы приложения.

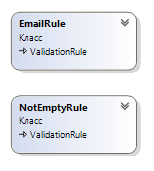


Рисунок 3.6 – Диаграмма классов валидации

Данные классы служат для валидации полей на таких страницах как SignUpView, AuthorizationView и ManagerView.



Рисунок 3.7 – Диаграмма классов валидации

Класс InternshipExtension содержит метод расширения GetSearchData для класса Internship.

Класс PasswordEncoder содержит логику хеширования и проверки пароля.

Класс AppNavHelper содержит методы, для отображения загрузочной панели, поля для хранения текущего пользователя и агрегирует NavigationService класса MainWindowView, для поддержки переходов на других страницах.

Класс ImageEncodingHelper реализует конвертацию png картинки в массив байт, для сохранения ранее упомянутой картинки в базе данных.

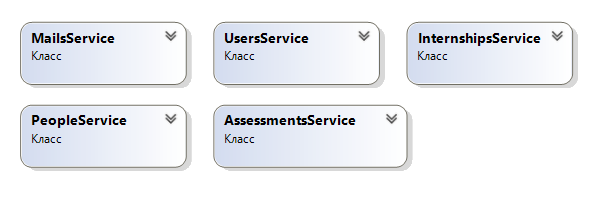


Рисунок 3.8 – Диаграмма классов сервисов

Класс MailsService содержит метод для отправки email сообщений.

Остальные классы представляют собой репозитории по работе с классом DataBaseContext и реализую методы добавления, редактирования, удаления и чтения.