2020

Курсовой проект

ВАРИАНТ 2. ТЕХОБСЛУЖИВАНИЕ

DANIL SHAPOVALENKO

КОМПЬЮТЕРНАЯ АКАДЕМИЯ «ШАГ»

ЗАДАНИЕ КУРСОВОГО ПРОЕКТА

Требуется создать программную систему, предназначенную для диспетчера станции техобслуживания. Такая система должна обеспечивать хранение сведений об услугах, оказываемых станцией и их стоимости, о клиентах станции, о работниках станции и об автомобилях, которые они ремонтируют в текущий момент.

Клиент станции – это человек, который хотя бы раз воспользовался услугами станции. О клиенте должны хранится следующие сведения: паспортные данные, включая фамилию, имя, отчество, дату рождения, прописку, а также даты обращения на станцию техобслуживания с указанием автомобилей, которые он сдавал в ремонт.

Клиент сдает в ремонт не обязательно автомобиль, владельцем которого он является.

Сведения об автомобилях включают в себя марку автомобиля, его цвет, год выпуска, номер государственной регистрации, перечень неисправностей и данные о владельце.

Сведения о работнике – это его фамилия, имя, отчество, специальность, разряд, стаж работы.

Диспетчер заносит в БД сведения об автомобиле и о клиенте, если клиент обращается на станцию впервые. После этого диспетчер определяет рабочих, которые будут устранять имеющиеся в автомобиле неисправности.

Оставляя автомобиль на станции техобслуживания, клиент получает расписку, в которой указано, когда автомобиль был поставлен на ремонт, какие он имеет неисправности, когда станция обязуется возвратить отремонтированный автомобиль.

После возвращения автомобиля клиенту данные о произведенном ремонте помещаются в архив, клиент получает счет, в котором содержится перечень устраненных неисправностей с указанием времени работы, стоимости работы и стоимости запчастей.

Возможно увольнение и прием на работу работников станции, изменение сведений о клиенте (клиент может поменять паспорт, права, адрес, телефон), номера государственной регистрации и цвета автомобиля.

Диспетчеру могут потребоваться следующие сведения:

1. Фамилия, имя, отчество и адрес владельца автомобиля с данным номером государственной регистрации?
2. Марка и год выпуска автомобиля данного владельца?
3. Перечень устраненных неисправностей в автомобиле данного владельца?
4. Фамилия, имя, отчество работника станции, устранявшего данную неисправность в автомобиле данного клиента, и время ее устранения?
5. Фамилия, имя, отчество клиентов, сдавших в ремонт автомобили с указанным типом неисправности?
6. Самая распространенная неисправность в автомобилях указанной марки?
7. Количество рабочих каждой специальности на станции?

Необходимо предусмотреть возможность выдачи справки о количестве автомобилей в ремонте на текущий момент, количестве незанятых рабочих на текущий момент.

Требуется также выдача месячного отчета о работе станции техобслуживания. В отчет должны войти данные о количестве устраненных неисправностей каждого вида и о доходе,

полученном станцией, а также перечень отремонтированных за прошедший месяц и находящихся в ремонте автомобилей, время ремонта каждого автомобиля, список его неисправностей, сведения о работниках, осуществлявших ремонт.

С-ТРЕБОВАНИЯ ПРОЕКТА

1. Создать программную систему, предназначенную для работы диспетчера станции техобслуживания.

2. Система должна обеспечить хранение следующих данных:
    a. Хранение данных об **услугах,** оказываемых станцией и их стоимости.
    b. Хранение данных о **клиентах** станции.
    c. Хранение данных о **работниках** станции.
    d. Хранение данных об **автомобилях**.

3. Характеристики услуг:
    a. Индекс
    b. Название
    c. Стоимость

4. Характеристики клиентов:
    a. данные **Человека** (Ф.И.О., паспорт).
    b. **Адрес** проживания (Улица, дом, квартира).
    c. Номер телефона
    d. Дата рождения
    e. Список дат обращения на сервис

5. Характеристики работников:
    a. данные **Человека** (Ф.И.О., паспорт).
    b. **Специальность** работника.
    c. **Статус работника.** (уволен/работает/свободен/в отпуске).
    d. Разряд
    e. Стаж работы

6. Характеристики автомобилей:
    a. **Марка** авто (Наименование марки, модель авто).
    b. Владелец авто (Данные человека. См. выше).
    c. Регистрационный номер.
    d. Цвет авто.
    e. Год выпуска авто.
7. Клиент – человек, который хотя бы 1 раз оформлял заказ на ремонт. Клиент может оформить заказ на ремонт авто, которым не владеет.

8. При оформлении заказа на ремонт, диспетчер заносит в БД сведения об автомобиле и о клиенте, если данных в БД ранее не было. После добавления авто и клиента, Диспетчер определяет работника, который будет ремонтировать неисправности.

9. После оформления заказа на ремонт, клиент получает **расписку** (чек), в котором содержится данные об дате оформления заказа, неисправности автомобиля, дата завершения ремонта (*доп. Информация содержит работника, который занимался ремонтом, данные об автомобиле, данные о клиенте*).

10. Диспетчер может принимать на работу новых работников, а также увольнять их.

11. Диспетчер может выполнять список запросов к базе данных, для поиска информации. Список запросов:
    a. Фамилия, имя, отчество и адрес владельца автомобиля с данным номером государственной регистрации?
    b. Марка и год выпуска автомобиля данного владельца?
    c. Перечень устраненных неисправностей в автомобиле данного владельца?
    d. Фамилия, имя, отчество работника станции, устранявшего данную неисправность в автомобиле данного клиента, и время ее устранения?
    e. Фамилия, имя, отчество клиентов, сдавших в ремонт автомобили с указанным типом неисправности?
    f. Самая распространенная неисправность в автомобилях указанной марки?
    g. Количество рабочих каждой специальности на станции?

12. Диспетчер может оформлять справки о кол-ве автомобилей на ремонте в текущий момент, количестве незанятых рабочих на текущий момент.

13. Диспетчер может оформлять отчеты о работе стации. Оформление месячных отчетов о работе станции:

    a. Первый Отчет:
        i. Количество устранённых неисправностей каждого вида
        ii. Доход от устраненных неисправностей
    b. Второй отчет:
        i. Перечень отремонтированных за прошлый месяц и находящихся в ремонте автомобилей
        ii. Время ремонта текущих автомобилей
        iii. Список неисправностей
        iv. Сведения о работниках, которые устраняли неисправности.

D-ТРЕБОВАНИЯ К ПРИЛОЖЕНИЮ

Используемые при разработке технологии:

1) Windows Presentation Foundation (WPF)
2) Entity Framework (Code-first)
3) Шаблон проектирования «Model-View-ViewModel» (MVVM)
4) Использование Git/Github

ЖУРНАЛ СПРИНТА

СПРИНТ №1

Задача 1: Создание сущностей баз данных.

Сущность «Персона»:

- Id
- Фамилия
- Имя
- Отчество
- Паспорт персоны

Сущность «Адрес»:

- Id
- Улица
- Дом
- Квартира

Сущность «Клиент»:

- Id
- Id персоны
- Id адреса
- Дата рождения
- Номер телефона

Сущность «Марка авто»:

- Id
- Название марки
- Модель

Сущность «Автомобиль»:

- Id
- Id модели
- Id владельца (персона)
- Регистрационный номер автомобиля
- Цвет автомобиля
- Год выпуска

Сущность «Специальность»:

- Id
- Название специальности

Сущность «Статус работника»:

- Id
- Статус работника в данный момент

Сущность «Работник»:

- Id
- Id персоны
- Id специальности
- Id статус работника
- Разряд работника
- Стаж работы

Сущность «Неисправности»:

- Id
- Название неисправности
- Стоимость починки
- Примерное количество времени, затрачиваемое на устранение неисправности

Сущность «Заявка на ремонт»:

- Id
- Id клиента
- Id работника
- Id автомобиля
- Список неисправностей (список id неисправностей)
- Дата оформления заявки
- Статус готовности авто (готово/не готово)
- Стоимость ремонта (ВЫЧЕСЛЯЕМОЕ СВОЙСТВО)
- Дата завершения ремонта (ВЫЧЕСЛЯЕМОЕ СВОЙСТВО)

Задача 1.1. Инициализация данных.

- Создание данных для работы приложения. Формирования базовых значений для работы с приложением.
- При инициализации базы данных использовать асинхронный метод для избегания паузы после запуска приложения.

Важность: 100
Сложность: 2.3 ч.
Затрачено времени: 3 ч.

Задача 2: Формирование базовых запросов к БД:

1. Формирование запроса для получения данных по всем персонам
2. Формирование запроса для получения данных по всем адресам
3. Формирование запроса для получения данных по всем клиентам
4. Формирование запроса для получения данных по всем маркам автомобиля
5. Формирование запроса для получения данных по всем автомобилям
6. Формирование запроса для получения данных по всем специальностям
7. Формирование запроса для получения данных по всем статусам работников
8. Формирование запроса для получения данных по всем работникам
9. Формирование запроса для получения данных по всем неисправностям
10. Формирование запроса для получения данных по всем заявкам на ремонт

Важность: 100
Сложность: 0.3 ч.

Затрачено времени: 0.2 ч.

Задача 3: Оформление MainWindow:

Дополнительным стилем оформления служит MaterialDesign (http://materialdesigninxaml.net/)

Главное окно приложения делиться на 3 столбца.

1. В первом столбце находится основные кнопки приложения:
    a. Оформление заявки на ремонт (открытие окна для оформления заявки)
    b. Запросы к базе данных (открытие окна для запросов к БД по ТЗ)
    c. О приложении (открытие окна «о приложении»)
    d. Изменение статуса готовности заявки (изменение статуса заявки на ремонт, становится активной только если пользователь выберет карточку заявки и если ремонт не был завершен)
    e. Выход из приложения (закрытие приложения)

Меню Дискорда вдохновило на оформление такого дизайна.

2. Во втором столбце находятся «Карточки», которые являются отображением Заявок на ремонт. В данную карточку входят все значения сущности.
3. В третьем столбце находятся данные по дополнительным сущностям и их обработкам
    a. Список клиентов
        i. Фамилия клиента
        ii. Имя клиента
        iii. Отчество клиента
    b. Кнопка добавления клиента
    c. Кнопка изменения клиента (становится активной только если пользователь выберет клиента для изменения его данных)
    d. Список работников
        i. Фамилия работника
        ii. Имя работника
        iii. Отчество работника
        iv. Статус работника в данный момент
    e. Кнопка добавление работника
    f. Кнопка увольнения работника (становится активной только если пользователь выбирает работника для увольнения)
    g. Список автомобилей
        i. Регистрационный номер автомобиля
        ii. Марка авто
        iii. Модель авто
    h. Кнопка добавления автомобиля
    i. Кнопка изменения автомобиля (становится активной только если пользователь выбирает авто для изменения данных)

Важность: 100
Сложность: 4 ч.
Затрачено времени: 5 ч.

СПРИНТ №2

Задача №1. Добавление данных для отображения данных в UI:

Прежде чем формировать контейнеры или переменные для выбранных пользователей данных, необходимо создать ViewModel для главного окна который будет реализовывать интерфейс INotifyPropertyChanged

1. Контейнер для хранения данных о заявках на ремонт.
2. Контейнер для хранения данных о клиентах.
3. Контейнер для хранения данных о работниках.
4. Контейнер для хранения данных о автомобилях.
5. Переменная для выбранной заявки на ремонт.
6. Переменная для выбранного клиент.
7. Переменная для выбранного работника.
8. Переменная для выбранного автомобиля.

Важность: 100
Сложность: 0.4 ч.
Затрачено времени: 0.3 ч.

Задача №2. Написание основных команд для главного окна приложения:

1. Команда для открытия окна «Оформление заявки на ремонт».
2. Команда для открытия окна «Заявки».
3. Команда для открытия окна «О приложении».
4. Команда для завершения ремонта.
    a. Добавить валидацию: если пользователь не выбрал заявку или в выбранной заявке ремонт уже завершен, то кнопка неактивна.
5. Команда для выхода из приложения.
6. Команда для добавления нового клиента.
7. Команда для изменения выбранного клиента.
    a. Добавить валидацию: если пользователь не выбрал клиента для изменения, то кнопка неактивна.
8. Команда для добавления нового работника.
9. Команда для увольнения выбранного работника.
    a. Добавить валидацию: если пользователь не выбрал работника для увольнения, то кнопка неактивна.
10. Команда для добавления нового автомобиля.
11. Команда для изменения выбранного автомобиля.
    a. Добавить валидацию: если пользователь не выбрал автомобиль для изменения, то кнопка неактивна.

Важность: 100
Сложность: 0.45 ч.
Затрачено времени: 1 ч.

Задача №3. Написание разметки для окна «Добавление/Изменение клиента»:

Параметры, которые будет заполнять пользователь:

1. Имя (заполнение данных через - TextBox)
2. Фамилия (заполнение данных через - TextBox)
3. Отчество (заполнение данных через - TextBox)
4. Паспорт (заполнение данных через - TextBox)
5. Номер телефона (заполнение данных через - TextBox)
6. Дата рождения (заполнение данных через - DatePicker)
7. Улица проживания (заполнение данных через - TextBox)
8. Дом (заполнение данных через - TextBox)
9. Квартира (заполнение данных через - TextBox)
10. Кнопка добавить.

Примечание: у всех TextBox написать hint для подсказки пользователю

Важность: 100
Сложность: 1 ч.
Затрачено времени: 1 ч.

Задача №4. Написание ViewModel для окна «Добавление/Изменение клиента»:

Прежде чем формировать переменную для клиента, необходимо реализовывать интерфейс INotifyPropertyChanged.

1. Переменная «Клиент» для привязки данных с окном разметки.
2. Команда закрытия окна.

При запуске окна передается переменная типа BOOL которая обозначает тип окна, изменение или добавление. Этот флаг меняет название окна и контент кнопки.

Важность: 100
Сложность: 0.45 ч.
Затрачено времени: 1 ч.

СПРИНТ №3

Задача №1. Написание разметки для окна «Формирования заявки на ремонт»:

Контейнеры для ввода/отображения данных:

1. Контейнер для выбора клиента. (ListView)
    a. Есть возможность добавлять клиентов (кнопка под контейнером)
2. Контейнер для выбора авто. (ListView)
    a. Есть возможность добавлять авто (кнопка под контейнером)
3. Контейнер для ввода даты подачи заявки (DatePicker)
    a. Изначально стоит «сегодняшняя» дата
4. Контейнер для выбора свободного работника (ComboBox)
5. Контейнер для выбора неисправностей (DataGrid)
    a. Есть возможность выбора нескольких неисправностей

Важность: 100
Сложность: 0.3 ч.
Затрачено времени: 0.2 ч.

Задача №2. Написание VeiwModel для окна «Формирование заявки на ремонт»:

Прежде чем формировать переменные для окна, необходимо реализовать интерфейс INotifyPropertyChanged.

1. Переменная «Заявка на ремонт» для получения результата.
2. Переменная «Клиент» для выбора клиента.
3. Список клиентов для выбора клиента.
4. Команда добавления нового клиента.

Добавление происходит сразу в БД и в контейнер для отображения. В случае если клиент не ввел все необходимые данные, то выдавать ошибку MessageBox.

5. Переменная «Автомобиль» для выбора автомобиля.
6. Список автомобилей для выбора автомобиля.
7. Команда добавления нового автомобиля.

Добавление происходит сразу в БД и в контейнер для отображения. В случае если клиент не ввел все необходимые данные, то выдавать ошибку MessageBox.

8. Список работников свободный на данный момент.
9. Переменная «Работник» для выбора работника.
10. Список неисправностей для выбора неисправностей.
11. Команда добавления заявки.

Формирование заявки из выбранных данных (в случае если нет привязки).
После выбрасывания значения в главное окно, заявка добавляется в БД, после чего появляется в контейнере для отображения и статус работника которого выбрал пользователь меняется на «Занят в данный момент»

12. Команда отмены.

Выбрасывать null значение и ничего не происходит. Просто прерывать операцию.

Важность: 100
Сложность: 2 ч.
Затрачено времени: 2.1 ч.

Задача №3. Написание разметки для окна «Добавление/Изменение авто»:

Контейнеры для ввода/отображения данных:

1. Контейнер для ввода марки (TextBox)
2. Контейнер для ввода модели (TextBox)
3. Контейнер для ввода гос. номера авто (TextBox)
4. Контейнер для ввода цвета авто (TextBox)
5. Контейнер для ввода года выпуска авто (TextBox)
6. Контейнер для выбора владельца (ListView)

Примечание: у всех TextBox написать hint для подсказки пользователю

Важность: 100
Сложность: 0.3 ч.
Затрачено времени: 0.35 ч.

Задача №4. Написание ViewModel для окна «Добавление/Изменение авто»:

Прежде чем формировать переменные для окна, необходимо реализовать интерфейс INotifyPropertyChanged.

1. Переменная авто
2. Переменная «Владелец» (Персона)
3. Список владельцев для выбора владельца (список Персон)
4. Команда закрытия окна

При запуске окна передается переменная типа BOOL которая обозначает тип окна, изменение или добавление. Этот флаг меняет название окна и контент кнопки.

Важность: 100
Сложность: 0.45 ч.
Затрачено времени: 1 ч.

СПРИНТ №4

Задача №1. Написать разметку для окна «Добавление работника»:

Контейнеры для ввода/вывода данных:

1. Контейнер для ввода имени (TextBox).
2. Контейнер для ввода фамилии (TextBox).
3. Контейнер для ввода отчества (TextBox).
4. Контейнер для ввода паспорта (TextBox).
5. Контейнер для выбора специальности работника (ComboBox).
6. Контейнер для выбора разряда работника (ComboBox).
7. Контейнер для ввода стажа работы (TextBox).
8. Кнопка для добавления работника.

Примечание: у всех TextBox написать hint для подсказки пользователю

Важность: 100
Сложность: 0.3 ч.
Затрачено времени: 0.35 ч.

Задача №2. Написание ViewModel для «Добавление работника»:

Прежде чем формировать переменные для окна, необходимо реализовать интерфейс INotifyPropertyChanged.

1. Переменная работника.
2. Список специальностей для выбора
3. Список разрядов для работника
4. Команда закрытия окна

Для списка специальностей и разрядов использовать тип string.

Важность: 100
Сложность: 0.45 ч.
Затрачено времени: 1 ч.

Задача №3. Написание разметку для окна «О приложении»:

В данном окне в FlowDocument будет располагаться разметка, состоящая из Paragraph и List в которой будет описано задание курсового проекта.

В данном окне ViewModel не нужна

Важность: 20
Сложность: 1 ч.
Затрачено времени: 1 ч.

Задача №4. Написание разметки для окна «Чек заявки на ремонт»:

В данном окне располагается только 1 элемент – TextBox для отображения чека по заявке.

Важность: 20
Сложность: 0.1 ч.
Затрачено времени: 0.1 ч.

Задача №5. Написание ViewModel для окна «Чек заявки на ремонт»:

Прежде чем формировать переменные для окна, необходимо реализовать интерфейс INotifyPropertyChanged.

1. Строка для отображения данных чека по заявке на ремонт.

Прежде чем писать данный класс необходимо написать Controller в котором будет формироваться чек для заявки на ремонт, записать данных в .txt файл. Во ViewModel будет происходить чтение данных из файла и перемещение этих данных в контейнер, описанный выше.

Важность: 50
Сложность: 0.3 ч.
Затрачено времени: 0.45 ч.

СПРИНТ №5