

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**КУРСОВА РОБОТА**

з дисципліни «Проектування корпоративних інформаційних систем»

на тему:

«**Розроблення корпоративної інформаційної системи СТО**»

Виконав студент

групи КП-61с

Серватович Павло Іванович

Залікова книжка КП-6110с

Керівник роботи

Павловський В.І.

До захисту допущено

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата, підпис)

Захищено з оцінкою

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата, підпис)

Київ 2017

**АНОТАЦІЯ**

Курсового проекту з дисципліни "Проектування корпоративних інформаційних систем"

студента гр. КП-61с Серватовича П.І.

Курсовий проект 37 с., 28 рисунків, 2 лістинга, 7 джерел.

Об’єктом розробки є корпоративна інформаційна система (КІС) СТО.

Метою розробки є створення програмного забезпечення, що надає можливість організувати роботу СТО. Система дозволяє одночасно візуалізувати хід виконання робіт на кожному етапі, формувати електроні документи для контролю ремонту. Залежно від наданих прав користувачеві він може здійснювати ті чи інші дії в системі.

Результатом розробки є розподілена КІС СТО.

Основні методи проектування – структурне та об’єктне проектування.

Здійснено аналіз предметної області та побудовані модель бізнес-процесів (IDEF0), діаграма потоків даних (DFD), модель логіки взаємодії потоків данных (IDEF3), концептуальна, логічна та фізична моделі даних (IDEF1x) СТО. Cтворено ОО-модель КІС, побудовано UML-діаграми прецедентів, класів та послідовностей. Спроектовано базу даних та розроблено програмне забезпечення КІС, що реалізоване як окремий настільний додаток мовою програмування C#.

КІС може функціонувати під управлінням ОС Windows. КІС може функціонувати під управлінням ОС Windows. Програмні інструменти, що використані при проектуванні КІС – ERWin, yEd GD і Enterprise Architect, та для реалізації КІС – C# (WPF, DOCX), MySQL.

**C#, WPF, CIS, КОРПОРАТИВНА, ІНФОРМАЦІЙНА, СИСТЕМА, DFD, IDEF, UML, СТО, Enterprise Architect, xaml.**

**АННОТАЦИЯ**

Курсового проекта по дисциплине "Проектирование корпоративных информационных систем"

студента гр. КП-61с Серватовича П.И.

Курсовой проект 37 с., 28 рисунков, 2 листинга, 7 источников.

Объектом разработки является корпоративная информационная система (КИС) СТО.

Целью разработки является создание программного обеспечения, позволяет организовать работу СТО. Система позволяет одновременно визуализировать ход выполнения работ на каждом этапе, формировать электронные документы для контроля ремонта. В зависимости от предоставленных прав пользователю он может совершать те или иные действия в системе.

Результатом разработки является распределенная КИС СТО.

Основные методы проектирования – структурное и объектно проектирования.

Осуществлен анализ предметной области и построены модель бизнес-процессов (IDEF0), диаграмма потоков данных (DFD), модель логики взаимодействия потоков данных (IDEF3), концептуальная, логическая и физическая модели данных (IDEF1x) СТО. Создан ОО-модель КИС, построено UML-диаграммы прецедентов, классов и последовательностей. Спроектирован базу данных и разработано программное обеспечение КИС, что реализовано как отдельный настольное приложение на языке программирования C #.

КИС может функционировать под управлением ОС Windows. КИС может функционировать под управлением ОС Windows. Программные инструменты, используемые при проектировании КИС - ERWin, yEd GD и Enterprise Architect, и для реализации КИС - C # (WPF, DOCX), MySQL.

**C#, WPF, CIS, Корпоративная, ИНФОРМАЦИОННАЯ, СИСТЕМА, DFD, IDEF, UML, СТО, Enterprise Architect, xaml**.

**ANNOTATION**

Course project on discipline "Design of corporate information systems"

student gr. KP-61s Servatovych P.I.

Course project 37 pp., 28 illustrations, 2 Listing, 7 sources.

The object is developing a corporate information system (CIS) stations.

The purpose of development is to create software that allows you to organize the station. The system can simultaneously visualize the progress of the work at each stage, to form electron document to control repair. Depending on the rights granted to the user, he can perform certain actions on the system.

The result is the development of distributed CBS stations.

Basic design methods - structured and object design.

The analysis of the subject area and built model business process (IDEF0), data flow diagram (DFD), interaction logic model data streams (IDEF3), conceptual, logical and physical data model (IDEF1x) stations. Ctvoreno OO model CIS, built UML-diagrams precedents, forms and sequences. Designed and developed a database of software CIS, which is implemented as a standalone desktop application programming language C#.

CIS can function running Windows. CIS can function running Windows. Software instruments used in the design of the CIS - ERWin, yEd GD and Enterprise Architect, and to implement CBS - C # (WPF, DOCX), MySQL.

**C#, WPF, CIS, corporate, INFORMATION, SYSTEM, DFD, IDEF, UML, stations, Enterprise Architect, xaml.**

**ЗМІСТ**

[ВСТУП 6](#_Toc484558893)

[1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ СИСТЕМИ 7](#_Toc484558894)

[1.1. Опис предметної області СТО 7](#_Toc484558895)

[1.2. Постановка завдання на розробку КІС СТО 8](#_Toc484558896)

[1.3. Опис засобів розробки програмного забезпечення 6](#_Toc484558897)

[2. ПРОЕКТУВАННЯ КОРПОРАТИВНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ СТО 12](#_Toc484558898)

[2.1. Опис засобів та інструментів проектування 12](#_Toc484558899)

[2.2. Проектування бізнес процесів системи 13](#_Toc484558900)

[2.3. Проектування моделі даних системи 20](#_Toc484558901)

[3. ПРОГРАМНА РЕАЛІЗАЦІЯ](#_Toc484558902) 24

[3.1. Загальна структура проекту 2](#_Toc484558903)4

[3.2. Підсистема роботи з даними 2](#_Toc484558903)5

[3.3. Архітектура системи 2](#_Toc484558903)8

[3.4. Функціональні модулі програми 2](#_Toc484558903)9

[3.5. Рекомендації щодо вдосконалення системи](#_Toc484558905) 35

[ВИСНОВКИ 39](#_Toc484558906)

[СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ 40](#_Toc484558907)

ВСТУП

В теперішній час впровадження інформаційних технологій відбувається в усі сфери людського життя. Станції технічного обслуговування, як приклад не великого виробництва, не є винятком, так як вести облік всіх видів діяльності значно зручніше в електронному вигляд для контролю і аналізу.

До складу СТО входять робітники, прилади, автомобілі над якими здійснюються ремонтні роботи. Ремонтні роботи над автомобілем в СТО є задокументованим і розділеним на визначені етапи. Процес інформатизації СТО є наступним етапом організації його роботи, адже саме так можна підвищити ефективність і покращення якості роботи робітників.

Створення корпоративної інформаційної системи СТО є актуальною задачею. Вона дозволить ефективно організувати його роботу, ввести контроль над виконанням, відслідковування поточного стану результатів роботи робітників. Також це підвищить рівень прозорості у всіх процесах на СТО для клієнта.

В даному курсовому проекті розроблено комп'ютерну інформаційну систему (КІС) для СТО. Дана КІС дозволить одночасно візуалізувати хід виконання робіт на кожному етапі, формувати електроні документи для контролю виконання ремонтних робіт. КІС повинна забезпечити доступ до певної інформації певним особам.

1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ СИСТЕМИ
   1. Опис предметної області СТО

СТО – підприємство, яке займається ремонтом (діагностикою) автомобільного транспорту на замовлення. СТО – невелике підприємство для обслуговування клієнтів, надаючи їм послуги ремонтних робіт транспортних засобів.

Для розширення пакету послуг на СТО, власники СТО створюють комплекси по обслуговуванню автомобілів. В ці комплекси входять на відміну від СТО: мийки, шиномонтажі та інше. Оскільки, чисельність населення зростає, так і зростає чисельність автомобілів, відбувається збільшення дорожньо-транспортних пригод (ДТП), зношування деталей приводять до необхідності звернення на підприємства автосервісу, що і супроводжується зростанням необхідності створення різних СТО, а також розширеного переліку послуг, що надаються .

Зараз в Україні багато різних пунктів, що реалізуються як автосервіс, але при цьому багато в чому вони не відповідає тим параметрам, необхідним при розробці і проектуванні СТО.

Відповідні послуги надають люди, які не мають спеціальної освіти, використовуються неякісні запасні частини. [2].

Коротко опишемо загальну структуру і роботи підприємства СТО. Весь процес роботи розпочинається з оформлення замовлень від клієнтів (власників автомобілів) через працівника СТО. Працівник оформлює замовлення від клієнтів і заздалегідь складає попередній план виконання ремонтних робіт для конкретного автомобіля. Після оформлення плану він надає завдання для спеціалістів у паперових документах, де вказано вид роботи, автомобіль, деталі автомобіля над якими потрібно провести роботу і запланований час початку і завершення роботи. В свою чергу працівники СТО приступає до виконання поставлених завдань. Після виконання завдання робітником, він фіксує завершення виконання роботи, вказуючи коментар після виконання. Підчас процесу виконання поставлених завдань голова зміни контролює своєчасність і якість виконання ремонтних робіт, по завершенню автомобіль готується до здачі клієнту.

Користувач системи – робітник майстерні, що безпосередньо працює з системою і таким чином, має можливість: слідкувати за станом виконання запланованого ходу, формувати і видавати завдання робітникам у паперовому вигляді, переглядати інформацію про результати роботи робітників.

Робітник – особа, що безпосередньо працює в майстерні. Виконує отриманні завдання, фіксуючи початок та завершення роботи.

* 1. Постановка завдання на розробку КІС СТО

Розробити комп’ютерну інформаційну систему СТО з використанням Web та/або Forms інтерфейсу користувача.

Основними задачами системи є:

* надання можливості користувачу спостерігати за станом виконанням робіт над усіма автомобілями, які знаходяться у СТО;
* створення умов для фіксування робітниками факту початку і завершення роботи над певним процесом, за допомогою особистого посвідчення робітника і документу зі завданням, шляхом сканування QR-коду камерою;
* формування електронних документів;
* відображення інформації про результати роботи робітника за вказаний період часу.

Система повинна надавити доступ двох рівнів: користувач системи, робітник СТО. Доступ до тієї чи іншої інформації чи до тих чи інших дій варто узгодити з замовником даної КІС.

Система повинна забезпечувати цілісність інформації, що зберігається в базі даних.

* 1. Опис засобів розробки програмного забезпечення

Для реалізації поставленої задачі необхідно було визначити мову програмування, на якій буде написаний код реалізації алгоритмів, систему управління базами даних для зберігання необхідної інформації в базі даних, а також зручний спосіб взаємодії цих двох інструментів розробки.

Було вирішено використовувати мову програмування C#, орієнтовану на платформу Microsoft .Net, у якості СКБД використати MySQL, а також підсистема для розробки інтерфейсу WPF.

**C#** – об'єктно-орієнтована мова програмування з безпечною системою типізації для платформи .NET. Розроблена Андерсом Гейлсбергом, Скотом Вілтамутом та Пітером Гольде під егідою Microsoft Research (при фірмі Microsoft).

Синтаксис C# близький до С++ і Java. Мова має строгу статичну типізацію, підтримує поліморфізм, перевантаження операторів, вказівники на функції-члени класів, атрибути, події, властивості, винятки, коментарі у форматі XML. Перейнявши багато що від своїх попередників – мов С++, Delphi, Модула і Smalltalk – С#, спираючись на практику їхнього використання, виключає деякі моделі, що зарекомендували себе як проблематичні при розробці програмних систем, наприклад множинне спадкування класів (на відміну від C++) [3].

**Windows Presentation Foundation** (WPF, кодова назва – Avalon) – графічна (презентаційна) підсистема в складі .NET Framework 3.0, що має пряме відношення до XAML.

Це перше реальне оновлення технологічного середовища призначеного для користувача інтерфейсу з часу випуску Windows 95. Воно включає нове ядро, яке повинне замінити GDI і GDI+, використовувані на нинішній Windows-платформі. WPF є високорівневим об'єктно-орієнтованим функціональним шаром, що дозволяє створювати двовимірні та тривимірні інтерфейси [4].

Однією з важливих особливостей є використання мови декларативною розмітки інтерфейсу XAML, заснованого на XML: ви можете створювати насичений графічний інтерфейс, використовуючи або декларативне оголошення інтерфейсу, або код на керованих мовах C# і VB.NET, або поєднувати і те, і інше.

**Model-View-ViewModel** – це шаблон проектування, що застосовується під час проектування архітектури застосунків (додатків). Публічно вперше був представлений Джоном Госсманом (John Gossman) у 2005 році як модифікація шаблону Presentation Model. MVVM орієнтований на такі сучасні платформи розробки, як Windows Presentation Foundation та Silverlight від компанії Microsoft.

MVVM полегшує відокремлення розбробки графічного інтерфейсу від розробки бізнес логіки (бек-енд логіки), відомої як модель (можна також сказати, що це відокремлення представлення від моделі). Модель представлення являє собою частину, яка відповідає за перетворення даних для їх подальшої підтримки і використання. З цієї точки зору, модель представлення більше схожа на модель, ніж на представлення і оброблює більшість, якщо не всю, логіку відображення даних. Модель представлення може також реалізовувати патерн медіатор, організовуючи доступ до бек-енд логіки навколо множини правил використання, які підтримуються представленням.

MVVM зручно використовувати замість класичного MVC та йому подібних у тих випадках, коли на платформі, де ведеться розробка, присутнє «зв'язування даних». [5].

**MySQL** – вільна система керування реляційними базами даних.

MySQL був розроблений компанією «ТсХ» для підвищення швидкодії обробки великих баз даних. Ця система керування базами даних (СКБД) з відкритим кодом була створена як альтернатива комерційним системам. MySQL з самого початку була дуже схожою на mSQL, проте з часом вона все розширювалася і зараз MySQL – одна з найпоширеніших систем керування базами даних. Вона використовується, в першу чергу, для створення динамічних веб-сторінок, оскільки має чудову підтримку з боку різноманітних мов програмування [6].

1. ПРОЕКТУВАННЯ КОРПОРАТИВНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ СТО
   1. Опис засобів та інструментів проектування

Для проектування моделі бізнес процесів використовуються три методології: IDEF0, IDEF3, DFD.

IDEF0 – Function Modeling – методологія функціонального моделювання і графічного описання процесів, призначена для формалізації і опису бізнес-процесів. Особливістю IDEF0 є її акцент на ієрархічне представлення об'єктів, що значно полегшує розуміння предметної області.

IDEF3 (англ. Integrated DEFinition for Process Description Capture Method) – методологія моделювання і стандарт документування процесів, що відбуваються в системі. Метод документування технологічних процесів являє собою механізм документування та збору інформації про процеси. IDEF3 показує причинно-наслідкові зв'язки між ситуаціями і подіями в зрозумілій експерту формі, використовуючи структурний метод вираження знань про те, як функціонує система, процес або підприємство.

Діаграма потоків даних (англ. Data Flow Diagram) – модель проектування, графічне представлення «потоків» даних в інформаційній системі. Діаграма потоків даних також може використовуватись для представлення процесів обробки даних (структурна розробка).

Для розробки діаграм використовується ERWin7, які містять інструменти для робот з вище переліченими моделями.

Для побудови ER діаграм використовується інструмент MySQL Workbench, що містить інструменти не тільки для моделювання концептуальних та фізичних моделей даних, але й для генерації фізичної бази даних.

При проектуванні об’єктно-орієнтованої структури проекту використовується UML. Дана мова моделювання дозволяє спроектувати велику кількість аспектів майбутньої системи за рахунок гнучкості, потужності та великої кількості різних типів діаграм. В даному проекті побудовані такі з них: діаграма класів, діаграма станів. Для побудови даних діаграм використовуємо програмний засіб – Enterprise Architect та yEd Graph Editor.

* 1. Проектування бізнес процесів системи
     1. **Побудова IDEF0 моделі**

На рис. 1 відображено діаграму першого рівня нотації IDEF0 діяльності СТО по ремонту автомобілів.

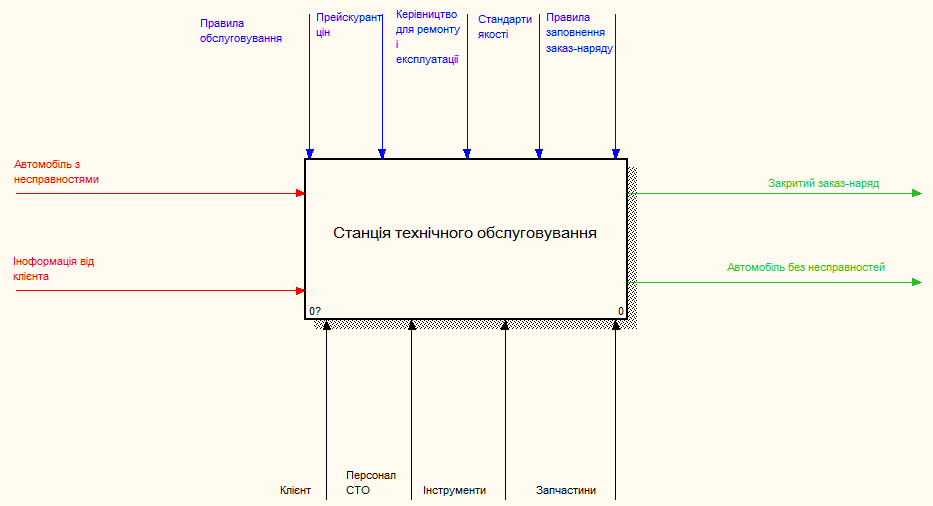


Рис. 1. Перший рівень IDEF0

Дана робота включає в себе отримання від клієнта замовлення на ремонт автомобіля, сам процес ремонту автомобілів на СТО.

На рис. 2 зображено діаграму декомпозиції другого рівня в нотації IDEF0. На даній діаграмі відбувається декомпозиція головної роботи підприємства на чотири складові: діагностика, ремонт, демонстрація клієнту, завершення.

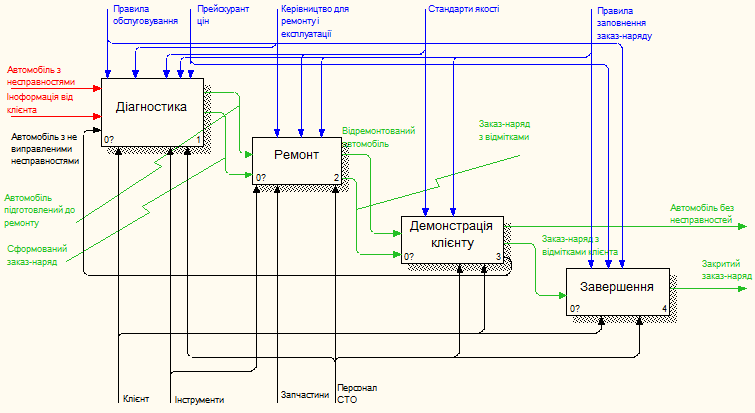


Рис. 2. Діаграма декомпозиції другого рівня в нотації IDEF0

**Діагностика**. Дана робота є важливою складовою для діяльності СТО. Під час діагностики проводиться огляд автомобіля для виявлення несправностей, узгодження списку несправностей з інформацією наданою клієнтом, а також визначається ціна і термін ремонту. Керується прейскурантом цін, правилами обслуговування клієнтів, керівництвом для ремонту і експлуатації, стандартами якості, правилами заповнення заказ-наряду. Даною роботою керує хтось з персоналу СТО, частіше за все майстер. На виході маємо сформований документ заказ-наряд і автомобіль підготовлений до ремонту. Діаграма представлена на рис. 3.

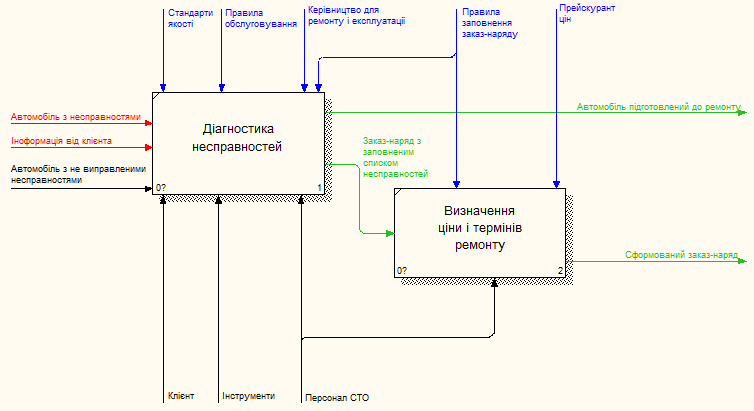


Рис. 3. Діаграма декомпозиції роботи Діагностика

**Ремонт**. Для проведення даної роботи залучаються необхідний персонал СТО, який вказаний в документі заказ-наряд. Ремонт здійснюється поетапно:

1. Складання плану ремонту – визначення виконавця і час/дату початку і закінчення виконання певної роботи по ремонту;

2. Виправлення несправностей – процес, в якому відбувається роботи над автомобілем, тобто ремонт автомобіля;

3. Контроль якості ремонту – перевірка якості проведених ремонтних робіт над автомобілем.

Керуємося при роботі правилами заповнення заказ-наряду, керівництвом для ремонту і експлуатації, стандартами якостями. На виході маємо документ заказ-наряд з відмітками результатів ремонтних робіт, а також відремонтований автомобіль. Діаграма відображена рис. 4.

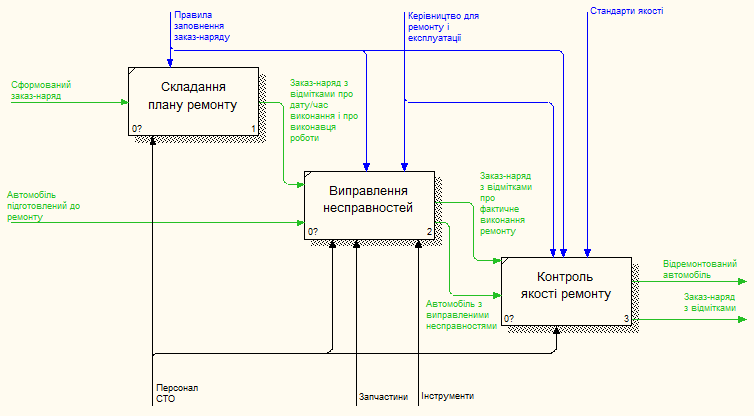


Рис. 4. Діаграма декомпозиції роботи Ремонт

**Демонстрація клієнту**. В даній роботі беруть участь як і людина з персоналу СТО, також і клієнт. Суть даної роботи полягає в тому, що клієнт повинний побачити чи всі необхідні несправності виправленні. Робота складається з двох підробіт: демонстрація виправлених несправностей, перегляд претензій клієнта. На вхід отримуємо документ заказ-наряд з всіма відмітками по ремонту, відремонтований автомобіль. Керуватися потрібно стандартами якості ремонту автомобілів, правилами заповнення документу заказ-наряд. На виході отримуємо, в разі відсутності необхідності проведення повторного процесу ремонту : автомобіль без несправностей, документ заказ-наряд з відмітками клієнта. В противному випадку – автомобіль повертається на етап діагностики ремонту. Діаграма відображена рис. 5.

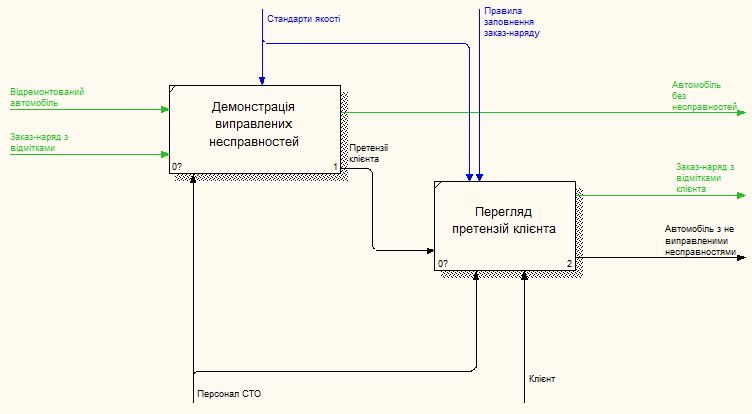


Рис. 5. Діаграма декомпозиції роботи Демонстрація клієнту

**Завершення**. Відбувається визначення кінцевої ціни ремонту, реєстрація оплати клієнтом, перевірка і закриття документу заказ-наряд. На вході отримуємо документ-наряд з відмітками клієнта. На виході – закритий цей документ. Дану роботу проводить персонал СТО, а оплату – клієнт. Керуватися потрібно прейскурантом цін, правилами обслуговування, правилами заповнення документу заказ-наряду. Діаграма відображена рис. 6.

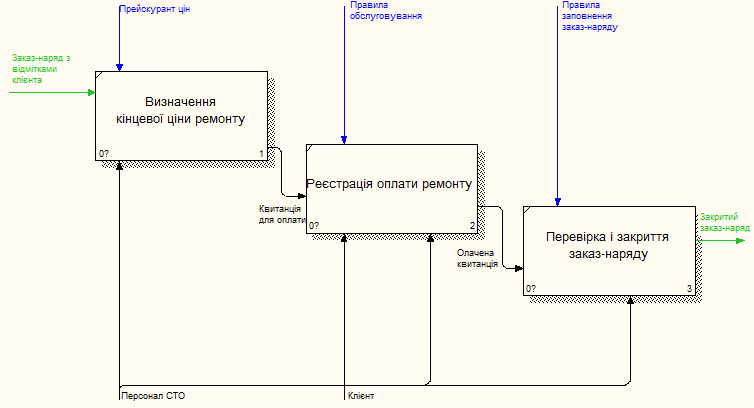


Рис. 6. Діаграма декомпозиції роботи Завершення

* + 1. **Побудова IDEF3 моделі**

Для побудови діаграми декомпозиції в нотації IDEF3 виберемо роботу з нотації IDEF0 – Діагностика несправностей. Діаграма декомпозиції в нотації IDEF3 представлена на рис. 7. Діагностика несправностей складається з наступних робіт:

* попередній огляд майстром;
* огляд спеціалістом;
* видача спеціалістом списку на перевірку за допомогою електроприборів;
* видача майстром списку на перевірку за допомогою електроприборів;
* обробка результатів перевірки2;
* обробка результатів перевірки1;
* складання висновку спеціалістом;
* складання висновку майстром.

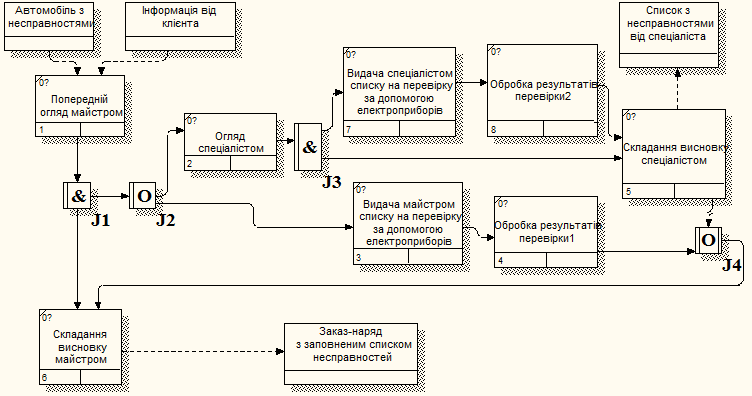


Рис. 7 Декомпозиція IDEF3 роботи Діагностика несправностей

Після отримання інформації про несправності від клієнта відбувається попередній огляд майстром. Дальше можливо два варіанти : 1) Якщо він вирішує, що потрібно оглянути автомобіль спеціалістом в окремій області, тоді автомобіль проходить огляд в цього спеціаліста; 2) Якщо майстер вирішує, що справиться сам з оглядом – він його проводить. Потім після 2-го варіанту подій складається висновок майстром. А після першого спеціаліст проводить огляд, робить свій висновок і на основі цього висновку, після цього, майстер робить кінцевий висновок. В результаті отримуємо заповнений майстром документ заказ-наряд з заповненим списком несправностей.

* + 1. **Побудова DFD моделі на основі IDEF0**

Для декомпозиції IDEF0 до DFD було обрано роботу Визначення ціни і терміну ремонту. Суть цієї роботи полягає в тому що персонал СТО визначає ціну і термін кожної окремої ремонтної роботи і заповнює документ заказ-наряд, записавши ціну і термін. DFD модель відображена на рис. 8.

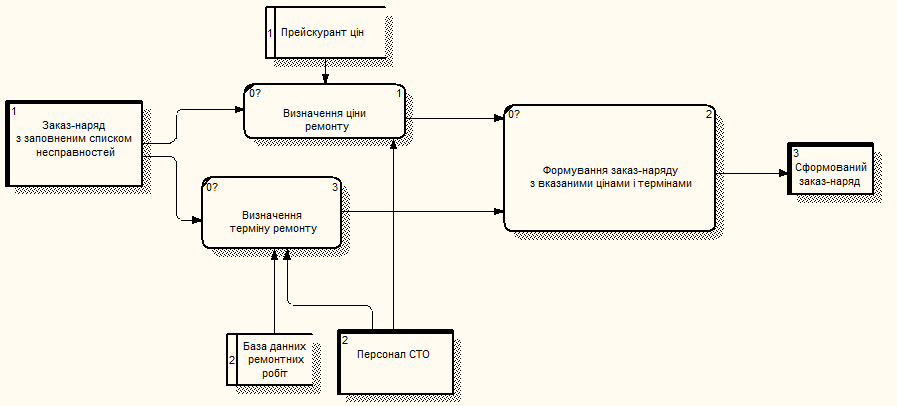


Рис. 8 DFD роботи Визначення ціни і терміну ремонту

* 1. Проектування моделі даних системи

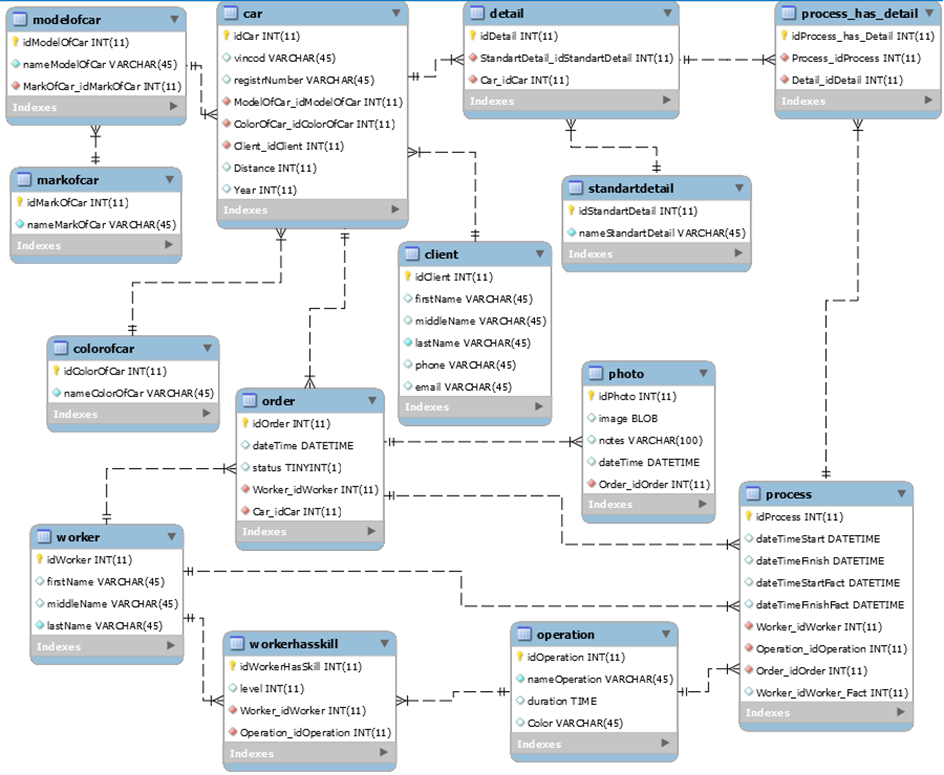


Рис. 9 EER діаграма КІС «СТО»

Для створення КІС «СТО» було спроектовану таку базу даних:

Car – таблиця машин, що задіяні в замовленнях. Таблиця пов’язана з власником автомобіля.

Detail – таблиця, що містить деталі конкретного автомобіля із списку стандартних деталей.

ColorOfCar – таблиця можливих кольорів автомобіля, що задаються рядковим типом даних.

MarkOfCar – таблиця марок автомобілів.

ModelOfCar – таблиця множин моделей автомобіля, заданої марки з таблиці MarkOfCar.

Photo – таблиця фотографій, що періодично робляться з веб-камери в автомайстерні. Ці фотографії пов’язуються з замовленням, що виконуються в даний час та заданому місці.

StandartDetail – список стандартних деталей, над якими можливий ремонт.

Operation – список операцій, які виконуються в майстерні.

WorkerHasSkill – множина навичок, якими володіє конкретний робочий зі зазначеним рівнем володіння.

Всі атрибути та первинні ключі таблиць, а також залежності між таблицями наведено на схемі бази даних системи візуалізації виконання виробничих процесів та маршрутизації виробничого документообігу. Схема бази даних представлена на рис. 9.

Важливим етапом проектування будь якої системи є розроблення Use Case діаграм. При роботі з даною інформаційною системою передбачається два актори:

* користувач – особа, яка безпосередньо працює з ІС;
* робітник – особа, яка працює на СТО.

Користувач системи – актор, який може створювати нові замовлення, створювати план робіт для замовлення, також слідкувати за станом виконання робіт згідно плану. Детальніше про можливості користувача, описують його прецеденти:

* створення замовлення, що розширюється додаванням нового автомобіля чи існуючого вже в базі даних;
* складання/редагування плану робіт, що включає роботу з процесами;
* додавання/редагування/видалення робітника з бази даних;
* формування документів за заданими формами;
* перегляд завантаженості робітника;
* перегляд стану виконання процесів над усіма замовленнями;
* перегляд інформації про результати роботи робітника;
* формування електронного текстового документу.

Діаграму варіантів використання «ІС СТО» для користувача наведемо на рис. 10 і рис. 11.

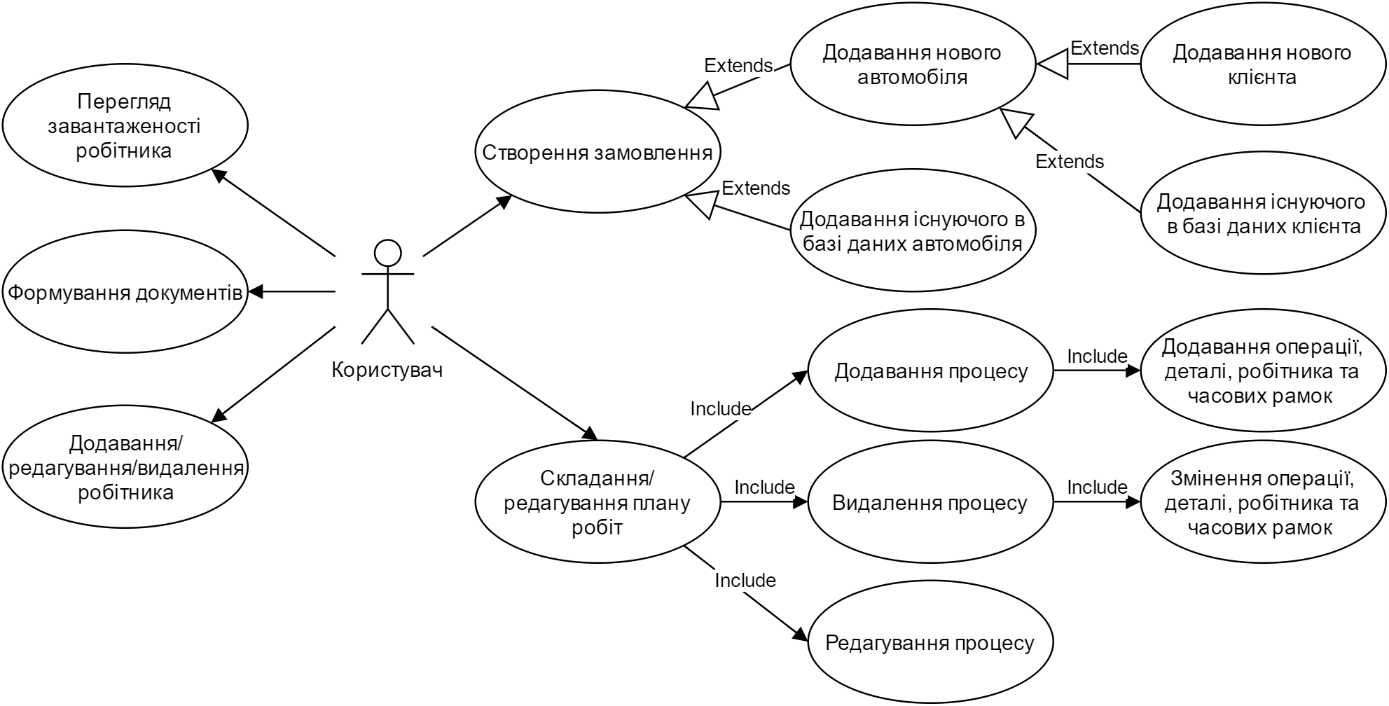


Рис. 10 Діаграма варіантів використання для користувача (Частина 1)

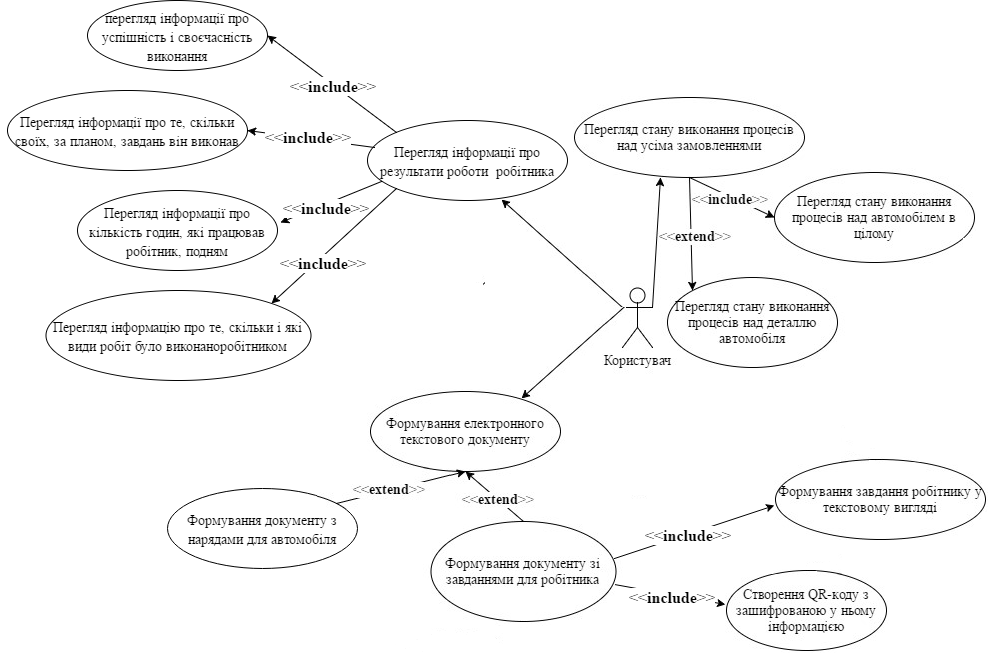


Рис. 11 Діаграма варіантів використання для користувача (Частина 2)

Для даної КІС розроблено спрощену діаграму класів. Більш розширений її варіант буде наведено далій в роботі. Для створення діаграми класів скористаємося ПЗ Enterprise Architect 8.0.

Після проведення роботи над створення діаграми класів розроблюваної системи отримали наступну статичну діаграму класів «ІС СТО», що зображена на рис. 12.

Створена діаграма класів «ІС СТО» містить в собі одинадцять класів:

* Worker – представляє робітника СТО;
* Client – клас, що описує клієнта СТО;
* Car – клас, що описує автомобіль;
* MarkOfCar –клас, що представляє марку автомобіля;
* ModelOfCar –клас, що представляє модель марки автомобіля;
* Color – колір автомобіля;
* Detail – клас, який представляє деталі автомобіля;
* Skill – навики, якими може володіти робітник СТО;
* Operation – клас, що представляє види робіт на СТО;
* Order – клас для представлення замовлення на ремонт автомобіля у СТО;
* Process – клас для представлення процесів ремонтних робіт з деталями автомобіля на СТО.

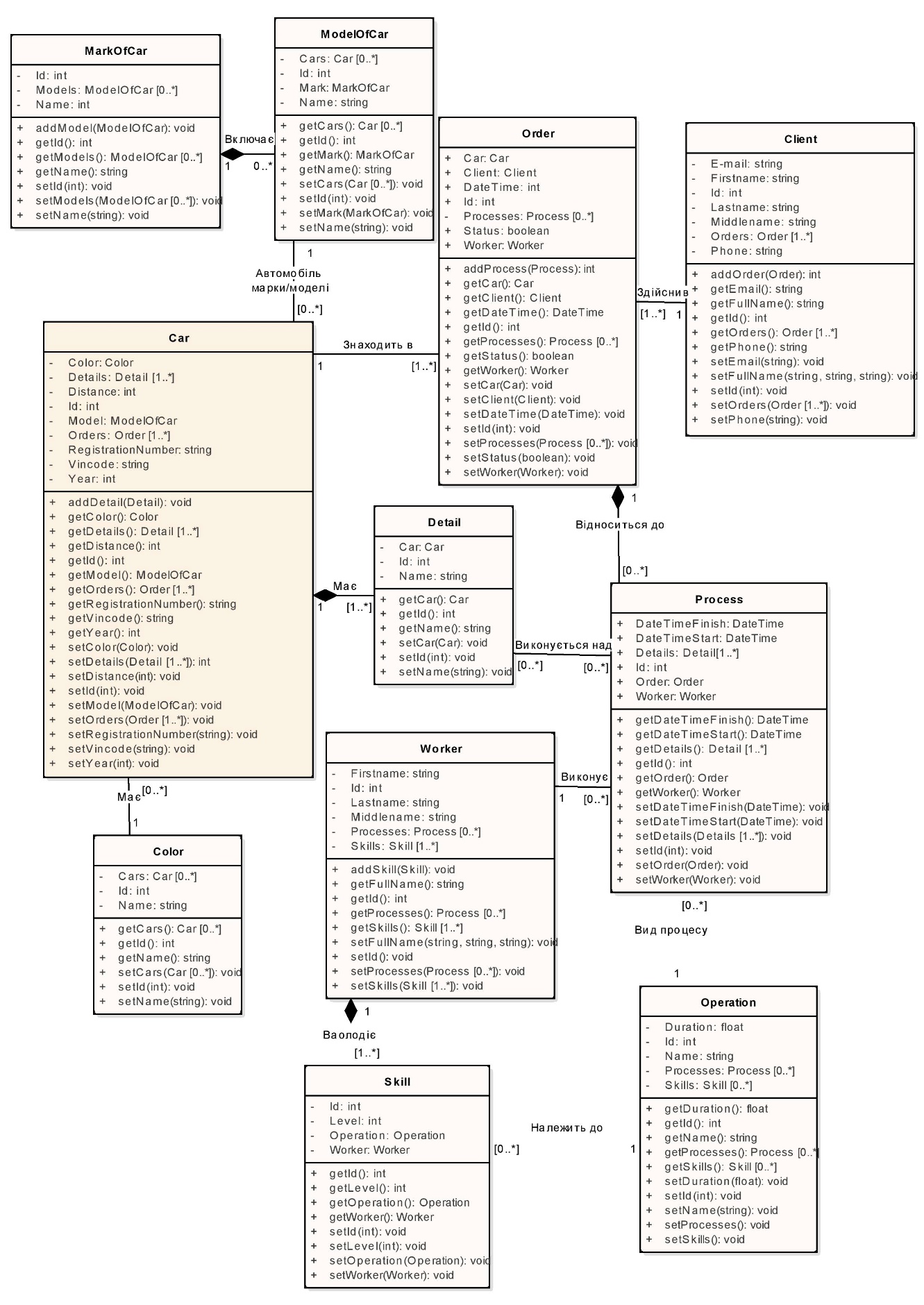


Рис. 12 Діаграма класів «ІС СТО»

Для даної КІС було створено 3 діаграми послідовності: «Створення замовлення», «Створення та робота з планом робіт», «Отримання документу зі завданням».

Діаграма послідовності створення замовлення в системі представлена на рис. 13. Ця діаграма описує процес створення користувачем замовлення, що може створюватися на основі нового клієнта або клієнта, що вже звертався до СТО, також на основі нового автомобіля або того, що вже проходив ремонт в даному СТО.

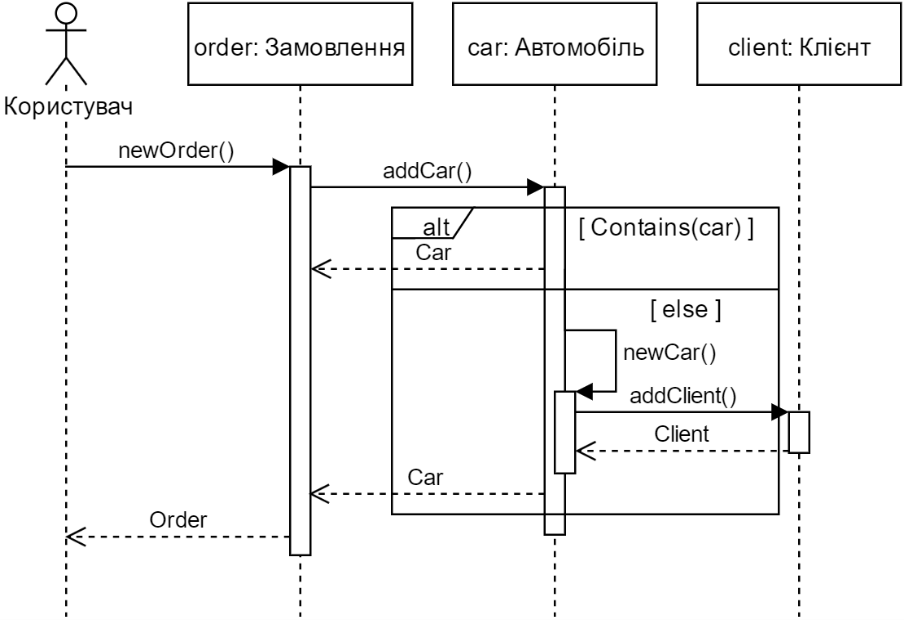


Рис. 13. Діаграма послідовності створення замовлення

Діаграма послідовності створення та робота з планом робіт в системі представлена на рис. 14. Дана діаграма описує роботу, яку здійснює користувач в даній системі, коли було здійснення реєстрація замовлення. Після реєстрації замовлення відбувається створення плану по ремонту автомобіля. Створення плану – це процес планування, розподілу робіт над автомобілем для успішного здійснення ремонту.

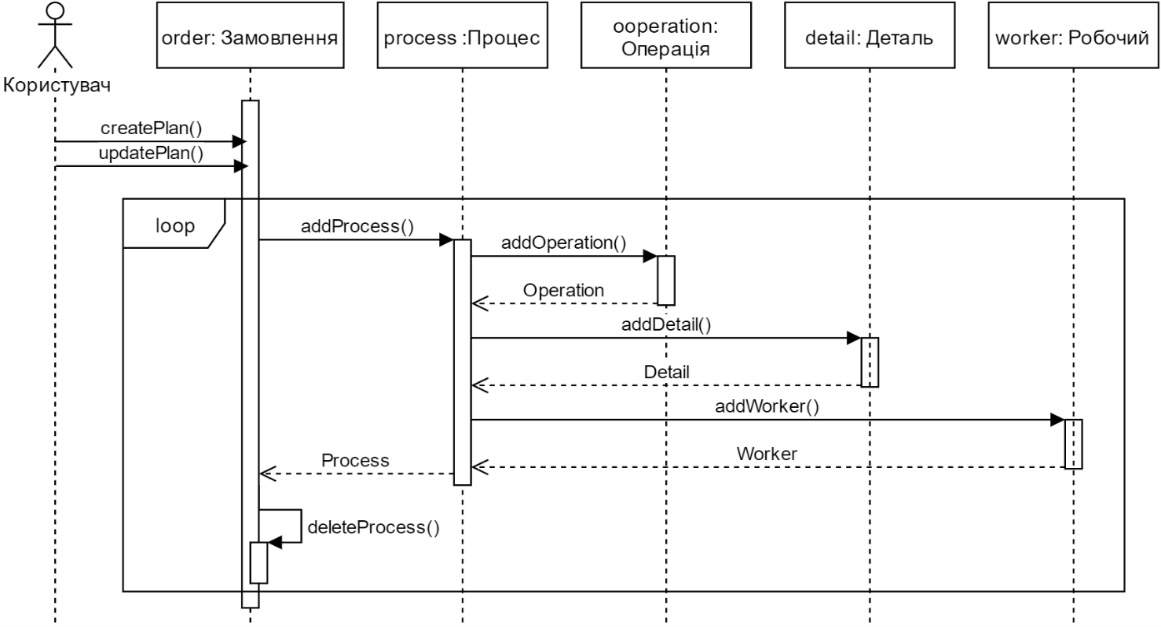


Рис. 14. Діаграма послідовності створення та робота з планом робіт

Третя діаграма описує процес формування електронного текстового документу зі завданням для робітника. Дане завдання формується відповідно до створеного раніше плану. В документі зі завданням вказано тип операції, яку необхідно зробити, запланований час початку і кінця, автомобіль і його деталі над якими виконуватиметься робота, а також інформації про робітника для якого заплановане це завдання. Діаграма послідовності отримання документу зі завданням представлена на рис. 15.

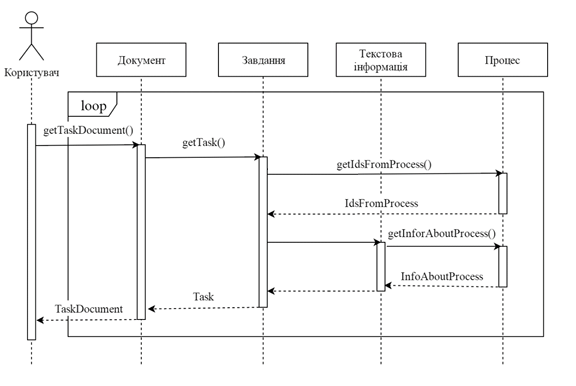


Рис. 15. Діаграма послідовності отримання документу зі завданням

1. ПРОГРАМНА РЕАЛІЗАЦІЯ
   1. Загальна структура проекту Архітектура системи

Загальна структура проекту, що розробляється мовою С# в середовищі розробки Visual Studio 2015, зображено на рис. 16.

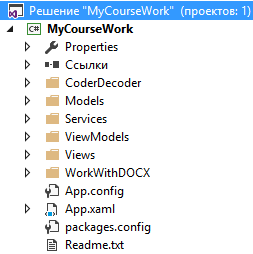


Рис. 16. Загальна структура проекту

Проект складається з 3 складових частин: модель, представлення і модель вигляду. Далі наведено короткий опис кожної частини.

Модель **–** містить описи класів та роботу з вмістом бази даних, складові частини зображено на рис. 17.

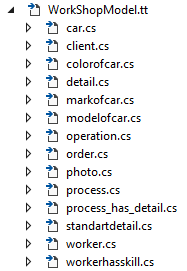


Рис. 17. Модель

Представлення **–** відповідає за користувацький інтерфейс, а саме це – набір представлень сторінок у форматі .xaml, зображено на рис. 18.

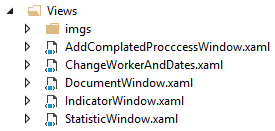


Рис. 18. Представлення

Модель вигляду **–** відповідає за взаємодію моделі та представлень, зображено на рис. 19.

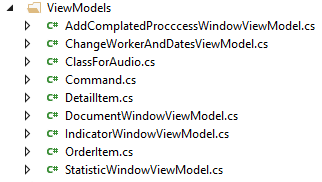


Рис. 19. Моделі вигляду

* 1. Підсистема роботи з даними

Підсистема роботи з даними включає в себе використання ADO.NET.

Для того, щоб не прив’язуватися до конкретної реалізації бази даних, було застосовано ADO.NET Entity Framework. Адже на цьому рівні не важлива конкретна реалізація бази даних або її драйвера.

Підключення до бази даних та бібліотеки, що використовуються, конфігуруєтся в файлі App.config, який наведено на лістингу 1.

|  |
| --- |
| **Web.config** |
| <?xml version="1.0" encoding="utf-8"?>  <configuration>  <configSections>  <!-- For more information on Entity Framework configuration, visit http://go.microsoft.com/fwlink/?LinkID=237468 -->  <section name="entityFramework" type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework, Version=5.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" requirePermission="false" />  </configSections>  <startup>  <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.0" />  </startup>  <connectionStrings>  <add name="workshopEntities" connectionString="metadata=res://\*/Models.WorkShopModel.csdl|res://\*/Models.WorkShopModel.ssdl|res://\*/Models.WorkShopModel.msl;provider=MySql.Data.MySqlClient;provider connection string=&quot;server=localhost;password=1111;user id=root;persistsecurityinfo=False;database=workshop&quot;" providerName="System.Data.EntityClient" />  </connectionStrings>  <entityFramework>  <defaultConnectionFactory type="System.Data.Entity.Infrastructure.LocalDbConnectionFactory, EntityFramework">  <parameters>  <parameter value="v13.0" />  </parameters>  </defaultConnectionFactory>  </entityFramework>  </configuration |

Лістинг 1. Налаштування підключення EntityFramework

Для таблиць бази даних створено класи, що їх описують. Приклад такого класу наведено на лістингу 2.

|  |
| --- |
| **Car.cs** |
| public partial class car  {  public car()  {  this.detail = new HashSet<detail>();  this.order = new HashSet<order>();  }    public int idCar { get; set; }  public string vincod { get; set; }  public string registrNumber { get; set; }  public int ModelOfCar\_idModelOfCar { get; set; }  public int ColorOfCar\_idColorOfCar { get; set; }  public int Client\_idClient { get; set; }  public Nullable<int> Distance { get; set; }  public Nullable<int> Year { get; set; }    public virtual client client { get; set; }  public virtual colorofcar colorofcar { get; set; }  public virtual modelofcar modelofcar { get; set; }  public virtual ICollection<detail> detail { get; set; }  public virtual ICollection<order> order { get; set; }  } |

Лістинг 2. Клас сутність Car

* 1. Архітектура системи

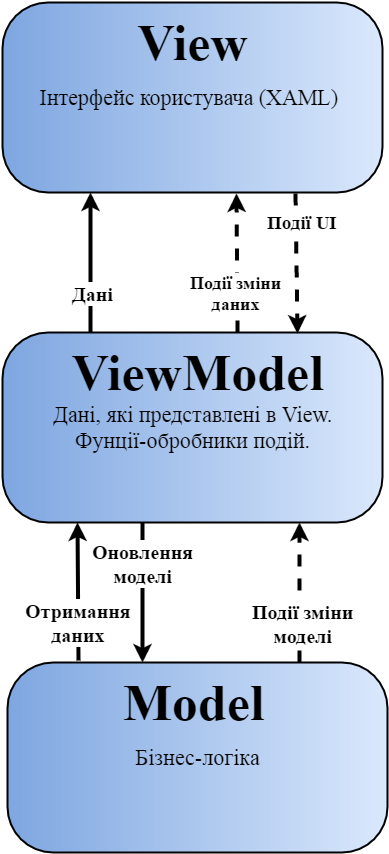


Рис. 20. Шаблон проектування MVVM

Структура архітектури програми для КІС «СТО» представлена на рис. 20. Вона складається з наступних частин:

* View – віконний графічний інтерфейс;
* ViewModel – модель вигляду, відіграє роль зв’язку між інтерфейсом і моделлю, тобто між View і Model;
* Model – модель, об'єктне-представлення бази даних.
  1. Функціональні модулі програми

Після запуску програми користувачу демонструється основним робоче вікно. Це вікно відіграє роль табло, яке представлена на рис. 21. На ньому відображається стан кожного процесу для кожного замовлення, нижче відображається стан процесів для деталей автомобіля вибраного користувачем замовлення. Таким чином користувач програми може візуально контролювати за виконанням процесів і своєчасно реагувати при необхідності. Також на головному вікні розміщені три кнопки, які дозволяють використовувати додаткові функціональні можливості програми. Це функціональні можливості для фіксування початку і завершення операції за допомогою сканування QR-коду, для формування різних видів електронних документів, для відображення інформації про результати роботи робітника.

Модуль, який відповідає за фіксування початку і завершення процесу, дозволяє робітнику швидко ввести інформацію до програми за допомогою сканування QR-коду, який знаходиться у документі зі завданням, а також на особистому посвідченні робітника . В цьому вікні відбувається сканування піднесеного до камери QR-коду, після чого з'являється інформацію про процес чи робітника, якщо QR-код пройшов перевірку. Модуль з успішним результатом сканування представлений на рис. 22.

Якщо інформація, яка була отримана в результаті сканування QR-коду є не вірною, тоді відображається відповідне повідомлення. Модуль з не успішним результатом сканування представлений на рис. 23. Для фіксування закінчення виконання процесу робітнику потрібно натиснути кнопку «ОК».

Перевірка отриманої інформації з QR-коду, необхідна для того, щоб забезпечити неможливість втрутитися в роботу системи сторонніх осіб, які згенерували власний QR-код. Кожен робітник має унікальний номер, який є зашифрований і відображений, за допомогою QR-коду на особистому посвідченні робітника.



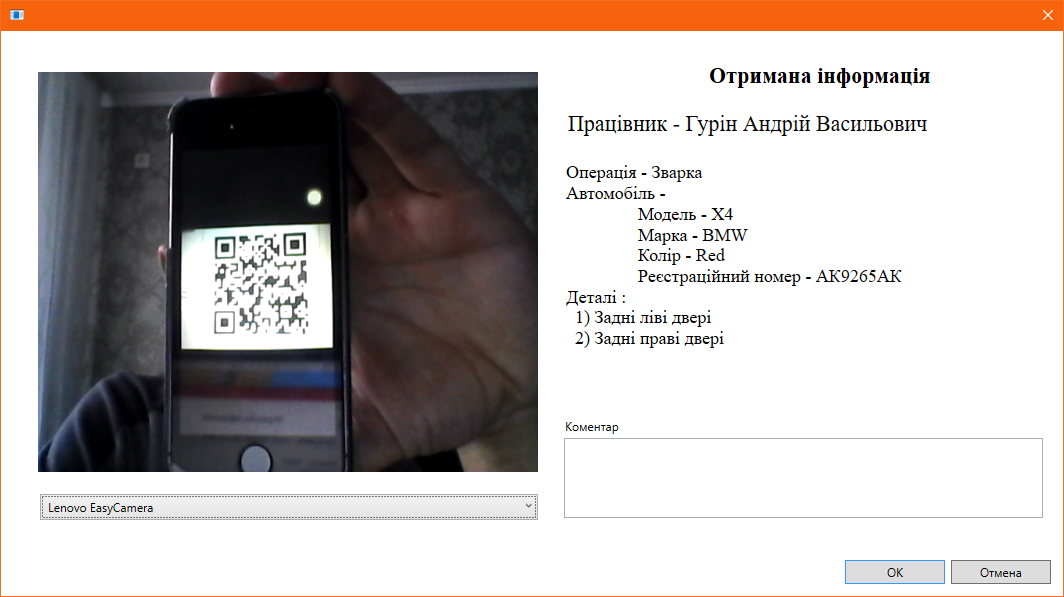


Рис. 22. Вікно сканування з успішним результатом

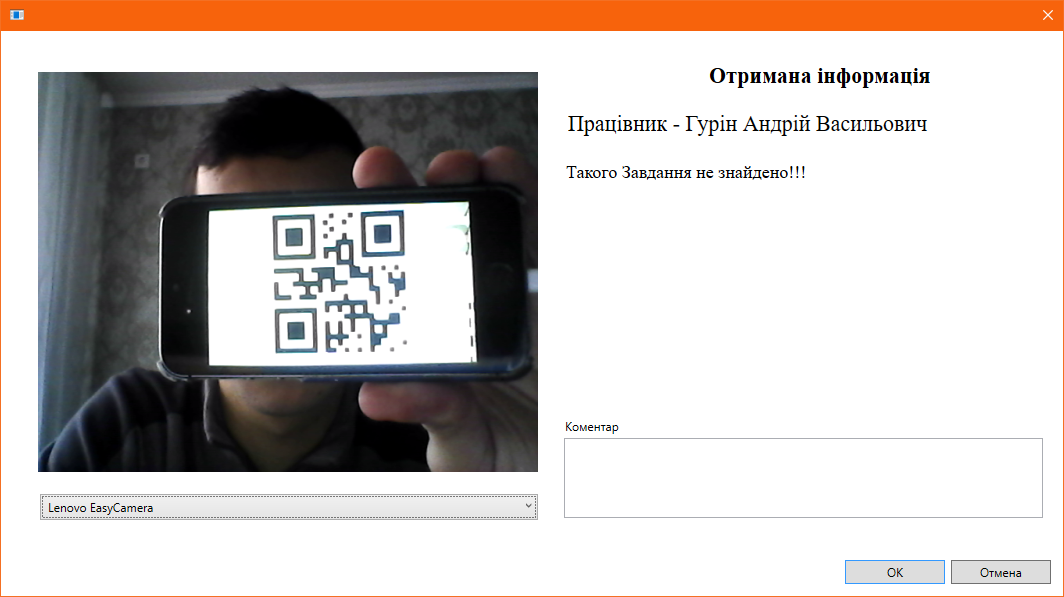


Рис. 23. Вікно сканування з не успішним результатом

Модуль, який відповідає за формування документів візуально представлений вікном програми, де користувач обирає, тип документу, який йому потрібний.

Вікно, де представлена вкладка «Завдання» на рис. 24, де відбувається формування документа зі завданням для робітника.

Вікно, де представлено вкладку «Наряди» на рис. 25, тут відбувається формування документу, в якому міститься інформація про всі процеси над автомобілем.

Формуючи документ зі завданням користувач отримує електронний текстовий документ формату DOC, він представлений на рис. 26.

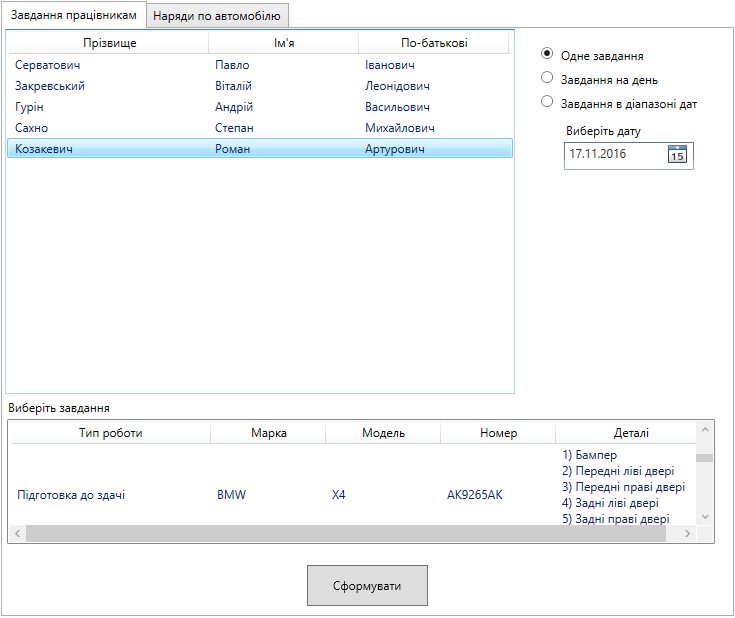


Рис. 24. Вікно формування документа вкладка «Завдання»

У разі якщо користувач сформував документ з нарядами, він отримає такий документ на рис. 27.

Модуль для відображення інформації про результати роботи робітника являє собою вікно з різними діаграмами і одним графіком. Вікно, де відображається інформація про результати роботи робітника представлене на рис. 28.

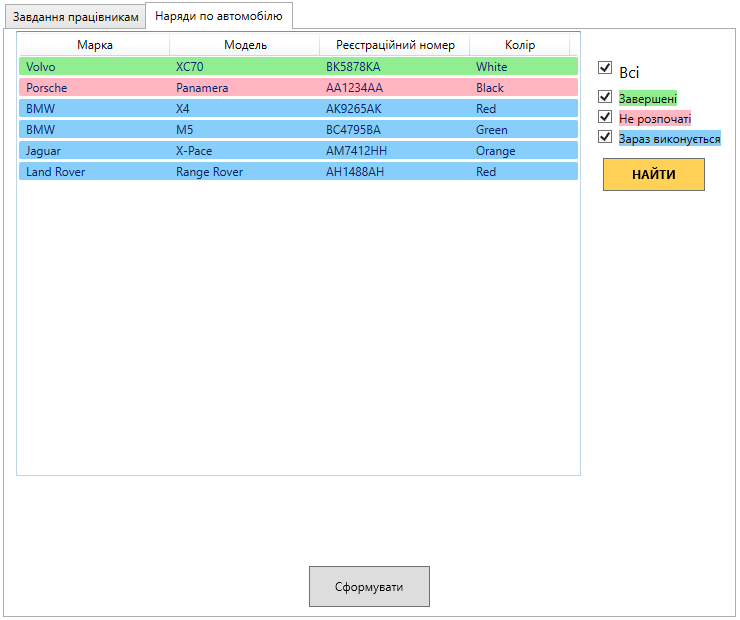


Рис. 25. Вікно формування документу вкладка «Наряди»

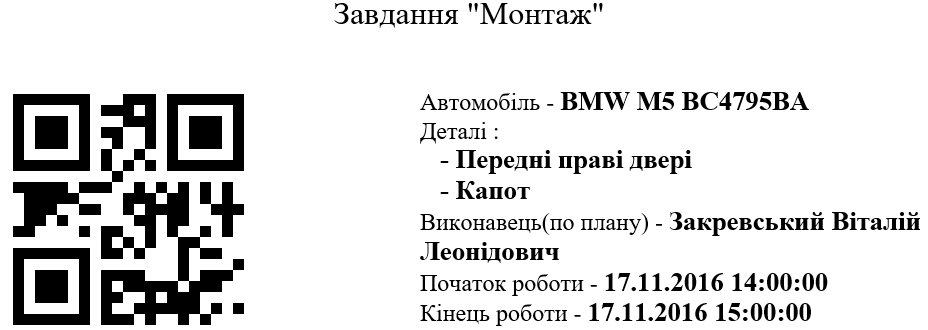


Рис. 26. Документ зі завданням

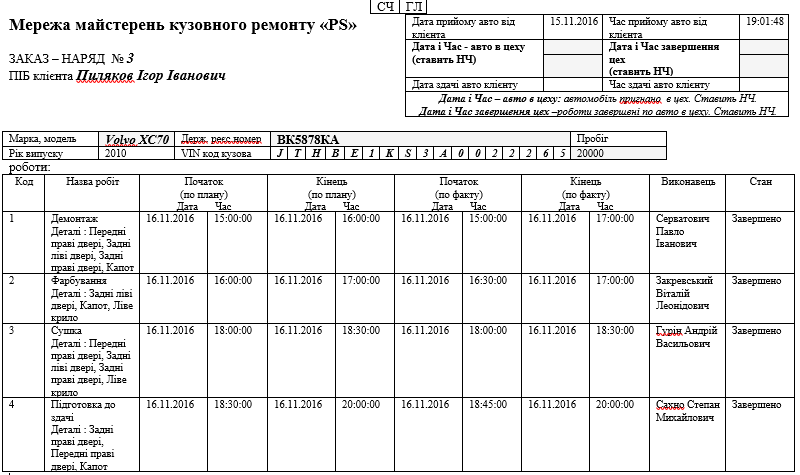


Рис. 27. Документ з нарядами для обраного автомобіля

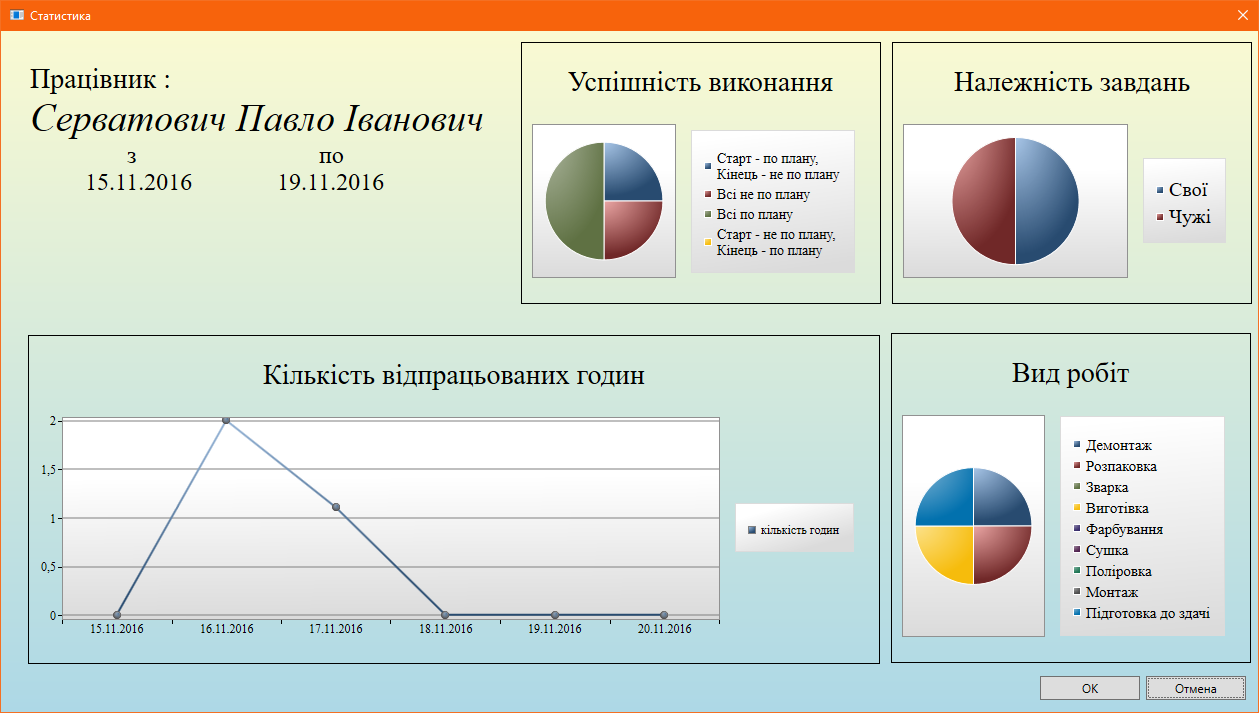


Рис. 28. Результати роботи робітника

* 1. Рекомендації щодо вдосконалення системи

Так як час і можливості виконання курсового проекту були обмежені, розроблена програма потребує покращення та вдосконалення. Проте всі вимоги поставленої задачі було реалізовано. На сьогодні, мною було реалізовано програму для візуалізації виконання виробничих процесів, тобто було створено табло, на якому відображаються стани всіх процесів з автомобілем та його деталями. Також реалізовано фіксацію початку та завершення виконання процесу, формування двох типів документів.

Список задач для майбутнього розвитку та вдосконалення програми:

1. Можливість фіксувати початок та завершення за допомогою спеціалізованого сканера;
2. Розробити web-ресурс для клієнтів, які б самі могли здійснювати замовлення відслідковувати стан ремонту свого автомобіля;

2. Можливість користувачу автоматично формувати більше різних типів документів;

3. Розробка оповіщення про відхилення чи не заплановані ситуації.

ВИСНОВКИ

В даному курсовому проекті було розроблено комп'ютерну інформаційну систему (КІС) для СТО. Дана КІС повинна забезпечити контроль за своєчасністю виконання всіх робіт на СТО, ввести прозорість в усіх процесах управління СТО.

Щоб вирішити всі поставленні завдання, я вирішив розробити настільний додаток. Для реалізації такого настільного додатку було обрано інструментарій: мова програмування C#, WPF .NET Framework, AForge .Net Framework, DocX .NET Library, ZXing .NET Library, СКБД MySQL, шаблон проектування MVVM, середовище розробки Microsoft Visual Studio 2012.

Розроблена програма для відображення стану виконання робіт на СТО з використанням QR-коду та веб-камери дозволяє:

* переглядати стан виконання всіх процесів для всіх авто;
* фіксувати, за допомогою сканування QR-коду, початок і завершення виконання процесів;
* переглядати інформацію про результати роботи робітників на певному часовому проміжку;
* формувати електронні текстові документи зі завданнями для робітників;
* формувати електронні текстові документи з нарядами для автомобілів.

Інтерфейс розробленої системи є зрозумілим і зручним.

Тестування продукту виконано у відповідності до затвердженої програми та методики тестування.

Застосування україномовне. Розроблена система підвищить якість контролю за виконанням попереднього плану і забезпечить швидке реагування на не заплановані ситуації.

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Информационная система [Електронний ресурс]. – Режим доступу: https://ru.wikipedia.org/wiki/Информационная\_система – Дата доступу: 6 червня 2017. – Назва з екрану.
2. СТО [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/СТО – Дата доступу: 6 червня 2017. – Назва з екрану.
3. C Sharp [Електронний ресурс]. — Режим доступу : https://uk.wikipedia.org/wiki/C\_Sharp. — Дата доступу : Березень 2017. — Назва з екрана.
4. Windows Presentation Foundation [Електронний ресурс]. — Режим доступу : https://uk.wikipedia.org/wiki/Windows\_Presentation\_Foundation. — Дата доступу : Березень 2017. — Назва з екрана.
5. Model-View-ViewModel [Електронний ресурс]. — Режим доступу : https://uk.wikipedia.org/wiki/Model-View-ViewModel. — Дата доступу : Березень 2017. — Назва з екрана
6. MySQL [Електронний ресурс]. — Режим доступу : https://uk.wikipedia.org/wiki/MySQL. — Дата доступу : Березень 2016. — Назва з екрана